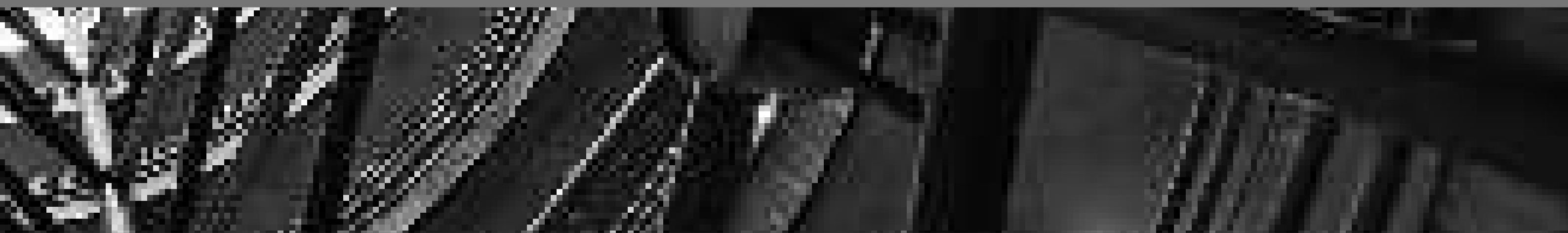
Navigating in Complex Process Model Collections

Markus Hipp



Dissertation

Ulm University
Institute of Databases and Information Systems

2015

Titelbild:
©Benjamin Erb (www.benjamin-erb.de)

Fakultät für
Ingenieurwissenschaften,
Informatik und Psychologie
Institut für Datenbanken
und Informationssysteme
Leiter: Prof. Dr. Manfred Reichert

Navigating in Complex Process Model Collections

DISSERTATION

Dissertation zur Erlangung des Doktorgrades Dr. rer. nat. der Fakultät für Ingenieurwissenschaften und Informatik der Universität Ulm

Vorgelegt von
Markus Hipp

aus Füssen, Deutschland
markus.hipp@uni-ulm.de

2015

Amtierende Dekanin:
Prof. Dr. Tina Seufert

Gutachter:
Prof. Dr. Manfred Reichert
Prof. Dr. Tina Seufert
Prof. Dr. Bela Mutschler

Tag der Promotion:
29. Juli 2015

Vorwort

Die vorliegende Arbeit ist das Resultat meiner Promotion, welche ich im Förderprojekt niPRO, in Zusammenarbeit mit der Daimler AG, der Universität Ulm und der Hochschule Ravensburg-Weingarten absolviert habe. Während der gesamten Zeit hatten verschiedene Personen maßgeblichen Anteil am Gelingen dieses Schriftstückes. Einigen davon möchte ich an dieser Stelle persönlich danken.

Mein erster Dank gilt meinem Doktorvater Prof. Dr. Manfred Reichert. Die fachlichen Gespräche und die zahllosen detaillierten Korrekturen und Anregungen haben maßgeblich zum Erfolg dieser Arbeit und aller dazugehörigen Veröffentlichungen und Vorträge beigetragen. Auch den Austausch zu fachfremden Themen habe ich immer sehr genossen.

Ganz besonders bedanken möchte ich mich bei Prof. Dr. Bela Mutschler, der diese Arbeit über den gesamten Zeitraum mit unermüdlichem Einsatz unterstützt hat. Der enge Austausch zu fachlichen Themen sowie seine Bereitschaft in kürzester Zeit Schriftstücke zu lesen, machten diese Betreuung einzigartig. Ohne seine unerschöpfliche Motivation, jedes meiner Arbeitsergebnisse immer noch besser machen zu wollen, wäre die Arbeit in dieser Form sicher nicht so entstanden.

Frau Prof. Dr. Tina Seufert danke ich herzlich für ihr Interesse an meinem Thema und die Bereitschaft, die Zweitbegutachtung zu übernehmen. Vielen Dank auch für die wichtigen Tipps und die Hilfestellung zu empirischen Fragestellungen.

Ich danke dem Team um Michael Offergeld für die angenehmen vier Jahre bei der Daimler AG und den nötigen Freiraum, den ich bekommen habe, um diese Arbeit erfolgreich zu beenden.

Allen DBIS-Mitarbeitern möchte ich für die schöne Zeit danken. Obwohl ich nur selten vor Ort sein konnte, so habe ich mich doch immer außerordentlich wohl und willkommen gefühlt. Speziell möchte ich mich bei Bernd Michelberger bedanken, der als direkter Kollege im gleichen Forschungsprojekt immer ein offenes Ohr für alle Probleme hatte und dessen Anregungen mir immer eine große Hilfe waren. Ich möchte mich auch bei allen Studenten bedanken, die mit ihren Abschlussarbeiten einen wichtigen Teil zu dieser Arbeit beitragen konnten.

Zuletzt bedanke ich mich bei meiner Familie und meiner Freundin Nicola, ohne deren Unterstützung ich sicher nicht so weit gekommen wäre.

Abstract

The increasing adoption of process-aware information systems (PAIS) has led to the emergence of large process model collections. In the automotive and healthcare domains, for example, such collections may comprise hundreds or thousands of process models, each consisting of numerous process elements (e.g., process tasks or data objects). In existing modeling environments, process models are presented to users in a rather static manner; i.e., as image maps not allowing for any context-specific user interactions. As process participants have different needs and thus require specific presentations of available process information, such static approaches are usually not sufficient to assist them in their daily work. For example, a business manager only requires an abstract overview of a process model collection, whereas a knowledge worker (e.g., a requirements engineer) needs detailed information on specific process tasks.

In general, a more flexible navigation and visualization approach is needed, which allows process participants to flexibly interact with process model collections in order to navigate from a standard (i.e., default) visualization of a process model collection to a context-specific one. With the *Process Navigation and Visualization (ProNaVis)* framework, this thesis provides such a flexible navigation approach for large and complex process model collections. Specifically, ProNaVis enables the flexible navigation within process model collections along three navigation dimensions. First, the *geographic dimension* allows zooming in and out of the process models. Second, the *semantic dimension* may be utilized to increase or decrease the level of detail. Third, the *view dimension* allows switching between different visualizations. All three navigation dimensions have been addressed in an isolated fashion in existing navigation approaches so far, but only ProNaVis provides an integrated support for all three dimensions.

The concepts developed in this thesis were validated using various methods. First, they were implemented in the process navigation tool *Compass*, which has been used by several departments of an automotive OEM (Original Equipment Manufacturer). Second, ProNaVis concepts were evaluated in two experiments, investigating both navigation and visualization aspects. Third, the developed concepts were successfully applied to process-oriented information logistics (POIL). Experimental as well as empirical results have provided evidence that ProNaVis will enable a much more flexible navigation in process model repositories compared to existing approaches.

Parts of this thesis have been published in the following referred papers:

Markus Hipp, Bela Mutschler, and Manfred Reichert. *On the Context-aware, Personalized Delivery of Process Information: Viewpoints, Problems, and Requirements.* in: Proc 6th Int'l Conf on Availability, Reliability and Security (ARES'11), LNCS 6908, pp. 390–397, Springer, 2011

Markus Hipp, Bela Mutschler, and Manfred Reichert. *Navigating in Process Model Collections: A new Approach Inspired by Google Earth.* in: Proc 1st Int'l Workshop on Process Model Collections (PMC'11), LNBIP 100, pp. 87–98, Springer, 2011

Markus Hipp, Bela Mutschler, and Manfred Reichert. *Navigating in Complex Business Processes.* in: Proc 23rd Int'l Conf on Database and Expert Systems Applications (DEXA'12), LNCS 7447, pp. 466–480, Springer, 2012

Markus Hipp, Bernd Michelberger, Bela Mutschler, and Manfred Reichert. *A Framework for the Intelligent Delivery and User-adequate Visualization of Process Information.* in: Proc 28th Symposium on Applied Computing (SAC'13), pp. 1383–1390, ACM, 2013

Bernd Michelberger, Armin Reisch, Bela Mutschler, Jörg Wurzer, Markus Hipp, and Manfred Reichert. *iCare: Intelligent Medical Information Logistics.* in: Proc 15th Int'l Conf on Information Integration and Web-based Applications & Services (iiWAS'13), pp. 396–399, ACM, 2013

Bernd Michelberger, Bela Mutschler, Markus Hipp, and Manfred Reichert. *Determining the Link and Rate Popularity of Enterprise Process Information.* in: Proc 21st Int'l Conf on Cooperative Information Systems (CoopIS'13), LNCS 8185, pp. 112–129, Springer, 2013

Markus Hipp, Bernd Michelberger, Bela Mutschler, and Manfred Reichert. *Navigating in Process Model Repositories and Enterprise Process Information.* in: Proc 8th Int'l Conf on Research Challenges in Information Science (RCIS'14), IEEE, pp. 1–12, 2014

Bernd Michelberger, Bela Mutschler, Daniel Binder, Jan Meurer, and Markus Hipp. *iGraph: Intelligent Enterprise Information Logistics.* in: Proc 10th Int'l Conf on Semantic Systems (SEMANTiCS'14), Posters & Demonstrations Track, 1224, pp. 27–30, 2014

Markus Hipp, Achim Strauss, Bernd Michelberger, Bela Mutschler, and Manfred Reichert. *Enabling a User-Friendly Visualization of Business Process Models.* in: Proc 3rd Int'l Workshop on Theory and Applications of Process Visualization (TaProViz'14), pp. 395-407, 2014

B. Michelberger, M. Hipp, and B. Mutschler. *Process-oriented Information Logistics: Requirements, Techniques, Application.* in: Advances in Intelligent Process-Aware Information Systems, 2015. Accepted for Publication

Table of Contents

Part I

Introduction

1 Motivation

In response to continuously increasing competitive pressure, shorter growing product lifecycles, and rising quality and cost demands, new ways of supporting business processes are needed. To the same extent, business processes are becoming increasingly complex and may comprise hundreds or thousands of process tasks. As examples consider distributed engineering processes [MHHR06, GOR12, GOR13], patient treatment processes in integrated healthcare networks [LR07], data collection processes in a supply chain [GMS$^+$13], or transportation and logistics processes [BKK04, Bas05]. Managing this complexity necessitates the introduction of process-aware information systems (PAIS) in enterprises [RW12]. Respective, PAIS usually rely on process models describing process logic and hence providing the schema for process execution [WRRM08]. In general, process models are managed by process management systems [MRB08], providing generic functions for modeling [Hav05], executing [WRWRM09, RRMD09, RW12], and monitoring processes [Men08]. This allows for a separation of concerns, which is a well established principle in computer science in order to increase maintainability and reduce complexity [Dij76].

In practice, business processes are stored and maintained in large process model repositories. They are created by process modelers using tools like ARIS Toolset, TIBCO Staffware Process Suite, Websphere Process Server, Bizagi, or Signavio Process Editor. The created models, in turn, are then distributed to process participants providing guidance for their daily work (cf. Figure 1.1).

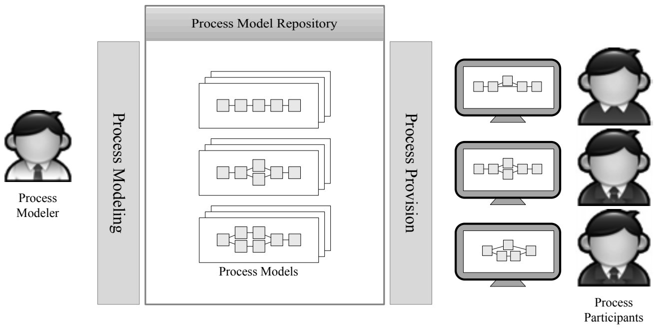


Figure 1.1: Process models in practice.

An example of a process model is depicted in Figure 1.2. This model reflects a *general specification* process from the automotive domain. More precisely, it deals with a part of the development of electric/electronic (E/E) components in a modern car. The model describes the preparation and creation of a *general specification* document describing the functions of a car control unit. The process model involves five roles: *E/E development (R1)*, *Component Responsible (R2)*, *Expert (R3)*, *Project Responsible (R4)*, and *Decision Maker (R5).* Furthermore, the process model comprises 11 tasks (i.e., *T1-T11*), which are related to the preparation, creation and validation of the general specification of a car control unit. The process tasks, in turn, refer to 12 data objects D1-D12. Note that in current practice respective process models are delivered to process participants in sheets or pdf files.

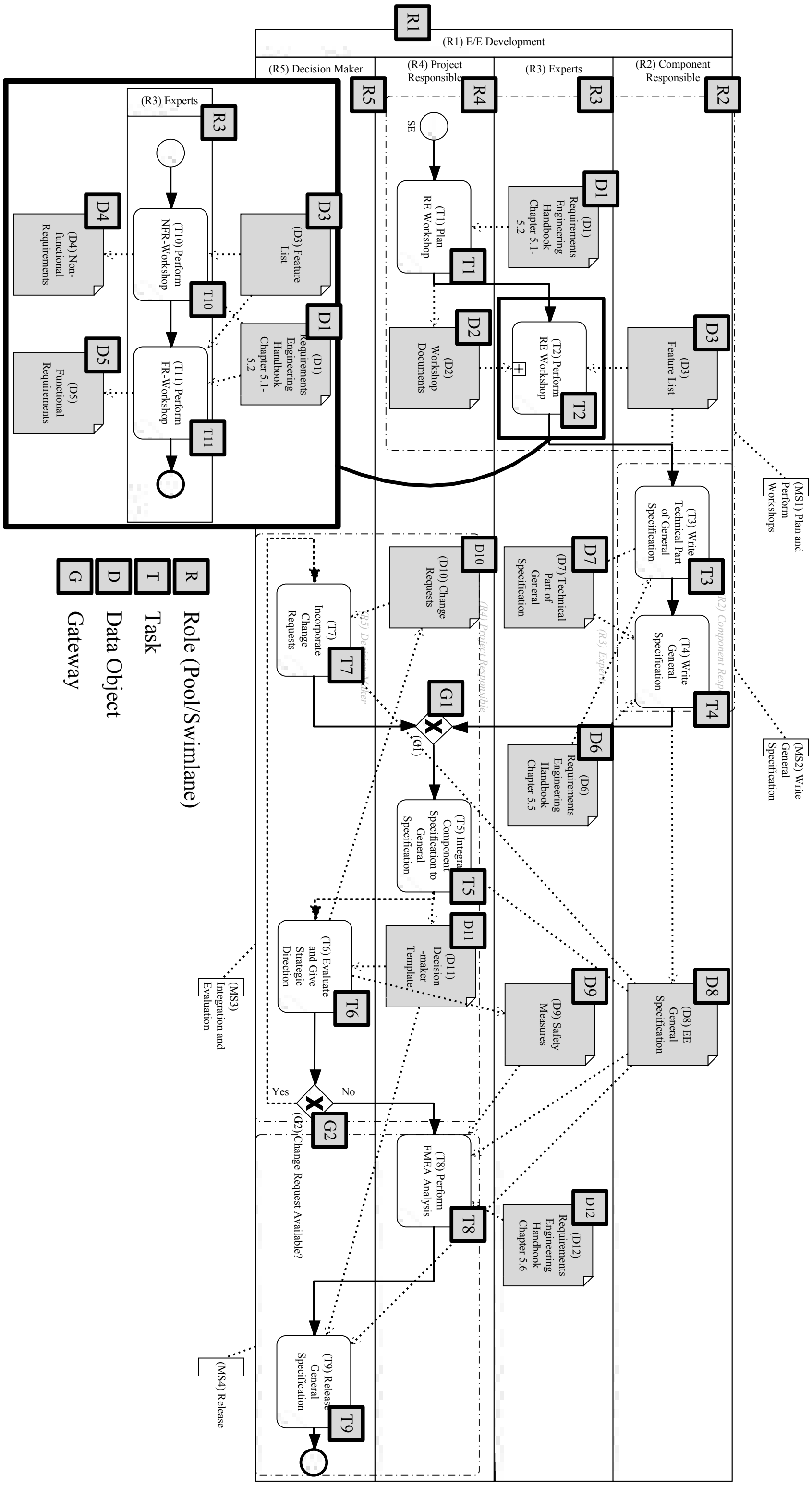


Figure 1.2: General specification process (partial view).

A well-defined set of process models is denoted as *process model collection* [WRMR11]. Usually, the models of such a collection are distributed across multiple departments [Som12, SZ10, MHHR06]. Furthermore, there exist model collections comprising dozens or hundreds of process models [Ger05, Rei11, LPR12].

To manage process model collections, (web-based) process portals, acting as model repositories, have been introduced in practice. An example of a screen of a process portal from the automotive domain is depicted in Figure 1.3. This portal aims to support knowledge workers, involved in the engineering of E/E[1] components for vehicles, i.e., the portal manages a collection of process models required for the development of E/E components.

In the top of Figure 1.3A, an abstract visualization of the entire process model collection is presented. Specifically, each box represents a *process area*, i.e., a set of process models related to a particular topic. In turn, process areas are manually defined by a process administrator and aim to assist knowledge workers in finding the right process models within a collection. In this context, an image map is provided for enabling user interactions, i.e., the user may click on a certain process area. Then, the document list at the bottom of Figure 1.3, which includes topic-related process models as pdf files (cf. Figure 1.3B), is adapted accordingly. This way, the list of displayed process models (i.e., pdf documents) may be reduced, enabling the user to quicker find the right process model, for example, if the user is interested in the process model for reviewing a specification document he will find the respective pdf document within the requirements engineering process area. However, this navigation approach is hard-wired, i.e., it is not generic, but only provides a solution with rudimentary navigation support for a particular application environment. In turn, this confirms the need for a generic approach enabling flexible navigation in process model collections.

1.1 Problem Statement

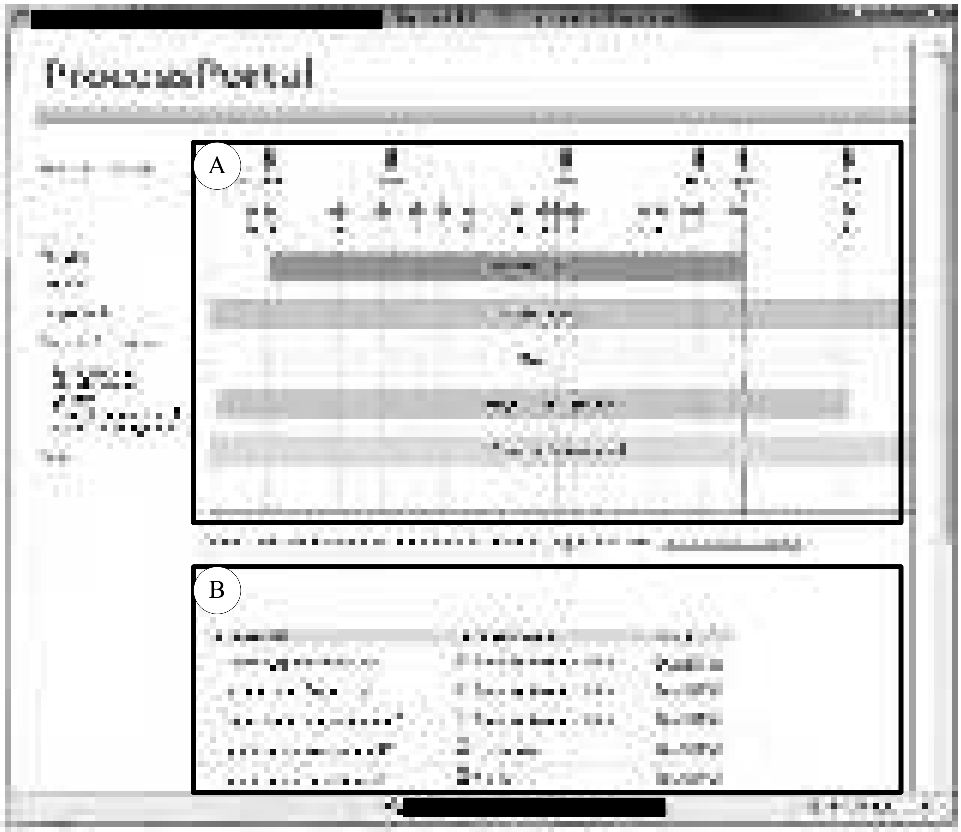
In general, a process portal integrates process models from different sources (e.g., departments) and provides central access to the resulting process model collection. However, the way such a model collection is presented to the process participants reveals several drawbacks

1.1.1 Visualization

In current approaches, process models are presented to process participants in a rather technical and non-intuitive manner [RRv$^+$11, FRS$^+$10]. In particular, the models are often displayed in exactly the same form as initially created by the process modeler [BRB07]. Neither contemporary process modeling tools (e.g., ARIS Architect) nor process portals offer a more sophisticated visualization approach in this context [BBR06, HMR11b]. However, as process participants (i.e., domain experts) are unfamiliar with technical process modeling notations, more user-oriented ways of visualizing process model collections are needed.

Drawback 1 (Visualization) *The visualization of process model collections in contemporary approaches is too static and at a too technical level for process participants.*

[1] E/E: electric/electronic



A: Process Model Collection; B: Process Information

Figure 1.3: Process portal from the automotive industry.

1.1.2 Interaction

As emphasized, process model collections are presented to the user in a static manner, e.g., as images or documents. As a consequence, there only exist rudimental ways for process participants to interact with a process model collection. In most cases, there is only one abstract image of the process model collection (cf. Figure 1.3A). Single process models are then represented in terms of simple pdf files (cf. Figure 1.3B). Regarding the aforementioned process portal, for example, it is not possible to flexibly switch between different process models (i.e., documents or images). Thereby, interaction is limited to hard-wired links between the images and documents. Process modeling tools often do not even consider process model collections, i.e., single process models are handled separately and, therefore, no interaction is possible at all.

Drawback 2 (Interaction) *The interaction within process model collections is limited to static links between images and documents.*

1.1.3 Navigation

The flexible navigation within a process model collection, e.g., to navigate from an abstract to a more detailed visualization of a process model collection or from the visualization of a particular process model to another one, is not considered by existing process modeling tools at all. Usually, only single process models are considered for navigation, and the combination of multiple process models must be realized by sub processes. The presented process portal, however, only provides the rigit navigation from an abstract visualization of the entire process model collection to a detailed visualization of single process models.

Drawback 3 (Navigation) *Process participants cannot flexibly navigate within process model collections.*

1.2 Use Cases

Due to the described drawbacks, current process portals are unable to support process participants in accessing process model collections [BBR06]. Only hard-wired and limited navigation possibilities are provided. Instead, a flexible navigation approach is needed that allows for a user-driven way of intuitively navigating within process model collections. Figure 1.4 illustrates the relations between *visualization*, *interaction* and *navigation*.

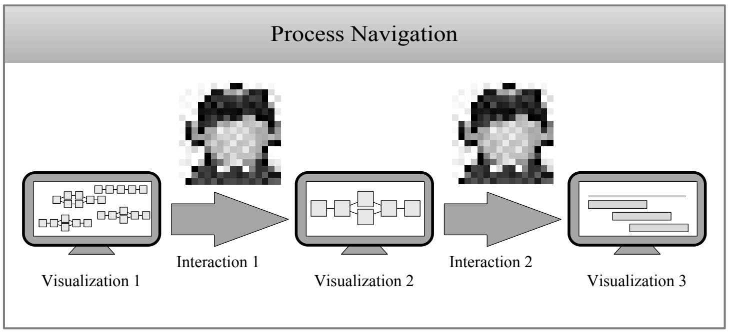


Figure 1.4: Basic process navigation approach.

To illustrate what kind of process navigation and visualization approach is actually needed, characteristic use cases are provided in the following. In particular, these are related to the development of a car control unit. The use cases allow us to illustrate the diversity of the requirements existing in the context of handling process model collections. Note that similar use cases can be found in other domains, like healthcare or finance, as well.

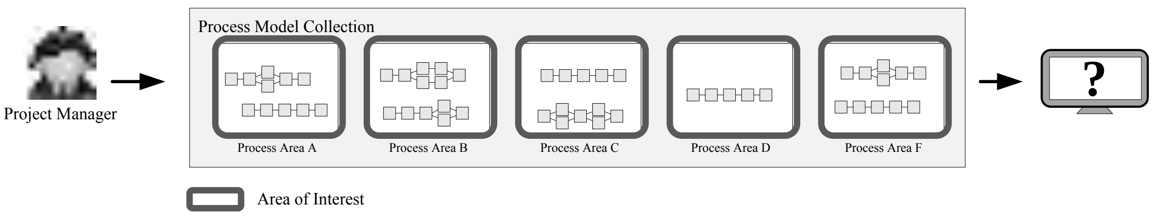


Figure 1.5: Use Case 1 - Project Manager.

- **Use Case 1 - Project Manager:** A project manager is responsible for the development of a car control unit and, hence, for the entire process model collection related to this task. To gather information about overall project status, for example, he should be able to get a quick overview on all relevant process areas (cf. Figure 1.5).

- **Use Case 2 - Business Unit Manager:** A business unit manager is responsible for a specific process area. For example, a *requirements manager* is responsible for process area *requirements engineering*. Unlike a project manager, he needs a more detailed visualization of the various process models of this area, e.g., to monitor process execution (cf. Figure 1.6). If there are delays during process execution, the business unit manager must be able to quickly identify that process task causing the delay as well as to interact with the person being responsible for this task.

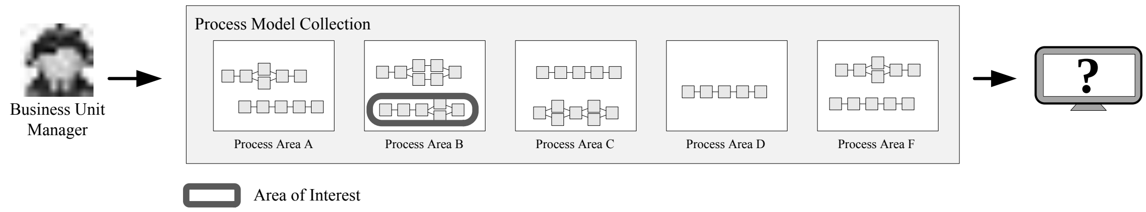


Figure 1.6: Use Case 2 - Business Unit Manager.

- **Use Case 3 - Requirements Engineer:** A requirements engineer creates specification documents for specific control units, e.g., the anti-lock breaking system (ABS) control unit. Accordingly, he must perform various tasks of the specification process. In this context, he needs access to technical instructions like guidelines, templates, or checklists. Finally, detailed task descriptions are required (cf. Figure 1.7).

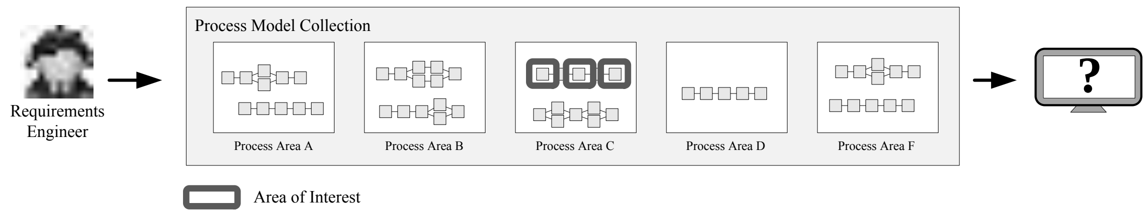


Figure 1.7: Use Case 3 - Requirements Engineer.

- **Use Case 4 - New Employee:** New employees need an overview on all process tasks they are responsible for. For example, an unexperienced requirements engineer needs an overview on all process tasks relevant in the process area *requirements engineering* (cf. Figure 1.8). Moreover, he needs detailed instructions for each of these tasks.

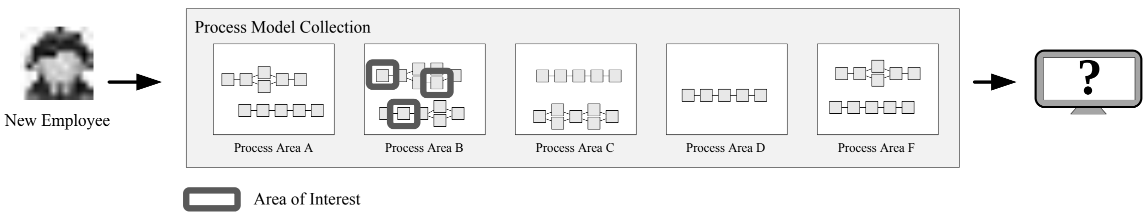


Figure 1.8: Use Case 4 - New Employee.

- **Use Case 5 - Quality Manager:** A quality manager is involved in various processes from different process areas. In particular, he is responsible for quality issues related to process execution, e.g., the quality of the specification documents created, test documents, or review documents. As these documents emerge from processes corresponding to different process areas, a quality manager needs an overview on all process models and tasks from the process model collection (cf. Figure 1.9).

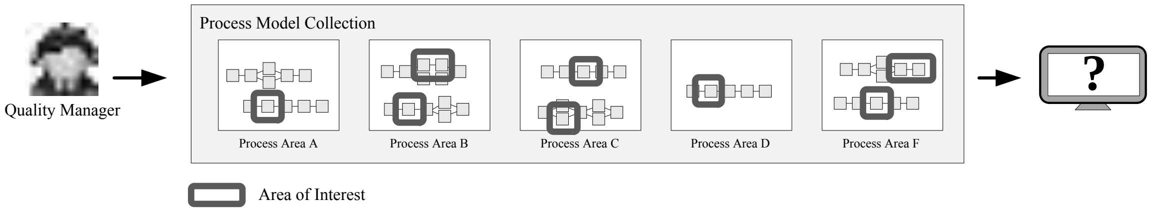


Figure 1.9: Use Case 5 - Quality Manager.

- **Use Case 6 - Quality Engineer:** A quality engineer must assure that process outcomes meet predefined quality standards. Unlike the quality manager, a quality engineer must consider deadlines (e.g., a quality gate). In this context, he must check all documents required to pass a specific quality gate. For this purpose, he needs access to information about all process tasks related to the creation of a document (cf. Figure 1.10).

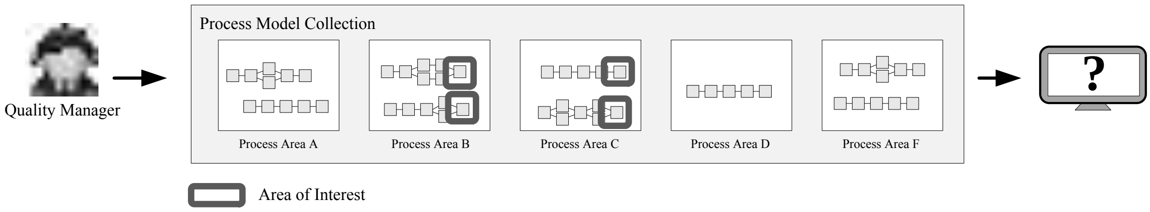


Figure 1.10: Use Case 6 - Quality Engineer.

- **Use Case 7 - Test Engineer:** A test engineer must define tests for a developed car control unit. Corresponding process models can be found in the process area *testing* (e.g., process area F in Figure 1.10). Furthermore, testing depends on the results produced by another process area dealing with “implementation" (process area D). Test engineers need to know which functions have been implemented in a specific car component in order to properly prepare the test cases.

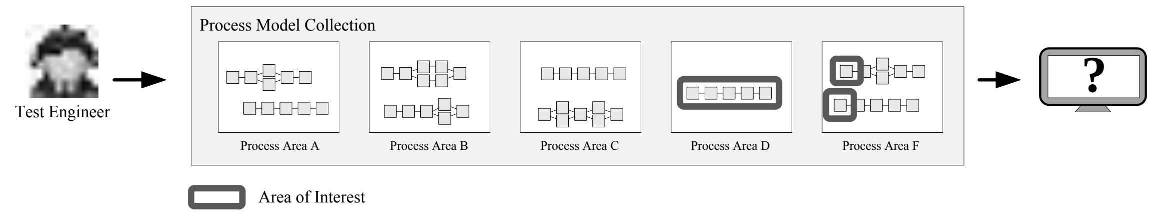


Figure 1.11: Use Case 7 - Test Engineer.

The use cases emphasize the need for enabling navigation within process model collections as well as for providing proper visualizations in this context. In particular, three major challenges need to be tackled:

1. Navigating on different levels of detail. For example, a project manager needs abstract information on the entire process model collection, whereas a business unit manager requires detailed information about a specific process model of a certain process area. In turn, a requirements engineer needs detailed information on single process tasks (e.g., task descriptions or documents created or consumed during task execution). Accordingly, navigation in process model collections is required on different levels of detail.

2. Navigation by zooming. The presented use cases demonstrate that process participants need to be able to navigate to different objects within a process model collection (e.g., process areas, process models, and process tasks). In turn, these objects may be spread across the entire process model collection. Think of a quality manager being interested in all process tasks he is responsible for. In turn, a requirements engineer might be only interested in a single process task. The second challenge for navigating in process model collections, therefore, is to be able to zoom to specific objects (i.e., to a specific part of the process model collection).

3. Navigation between different visualizations. Process participants require various visualizations of a process model collection. For example, the quality manager may want to focus on temporal aspects (e.g., deadlines), whereas a business unit manager may need more detailed information about process participants. Finally, a requirements engineer needs textual descriptions of specific process tasks to properly understand them. Thus, as a third challenge, it must be possible to switch between different visualizations on the same information.

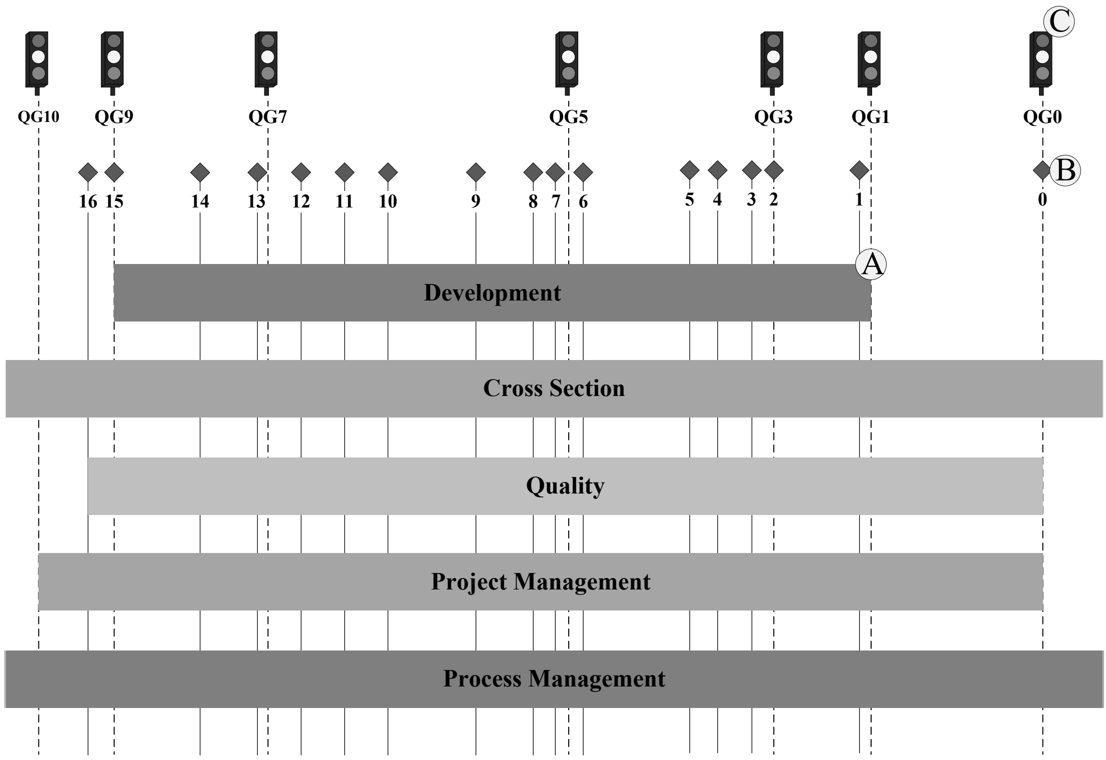


Figure 1.12: Time-based visualization of a process model collection.

In order to enable such flexible ways of navigating within process model collections, today's enterprises manually add static visualizations (i.e., images) to process portals. In turn, this allows users to navigate between the visualizations (i.e., switch from one image to another). For example, Figure 1.12 shows a time-based visualization of the process model collection maintained by the process portal from Figure 1.3, i.e., a visualization focusing on temporal aspects. More precisely, rectangular boxes are used to represent process areas (cf. Figure 1.12A). Furthermore, all process areas are displayed, i.e., the zoom refers to the entire process model collection, and they are aligned with a grid that visualizes deadlines (e.g., milestones (cf. Figure 1.12B) and quality gates (cf. Figure 1.12C)). In particular, this visualization allows users to quickly scan the temporal properties and dependencies of the various process areas. For example, process managers can use this visualization to get quick overview on the entire process model collection.

Figure 1.13 shows a *logic-based* visualization of a single process model, i.e., the zoom is on one particular process model. The visualization is based on the *Business Process Management and Notation (BPMN)* standard [MW08]. As can be seen, swimlanes are used to represent the role responsibilities for the various process tasks. Thereby, different shades of grey are applied to assign single roles to organizational units, such as *management*, *development* or *testing* (cf. Figure 1.13A). In general, this visualization focuses on the causal relations between process tasks (i.e., predecessor and successor relations). However, temporal dependencies may be also considered by picking up milestones and quality gates from the time-based visualization (cf. Figure 1.13B+C). This visualization is used, for example, by business unit managers to gather information about single process models and responsible roles.

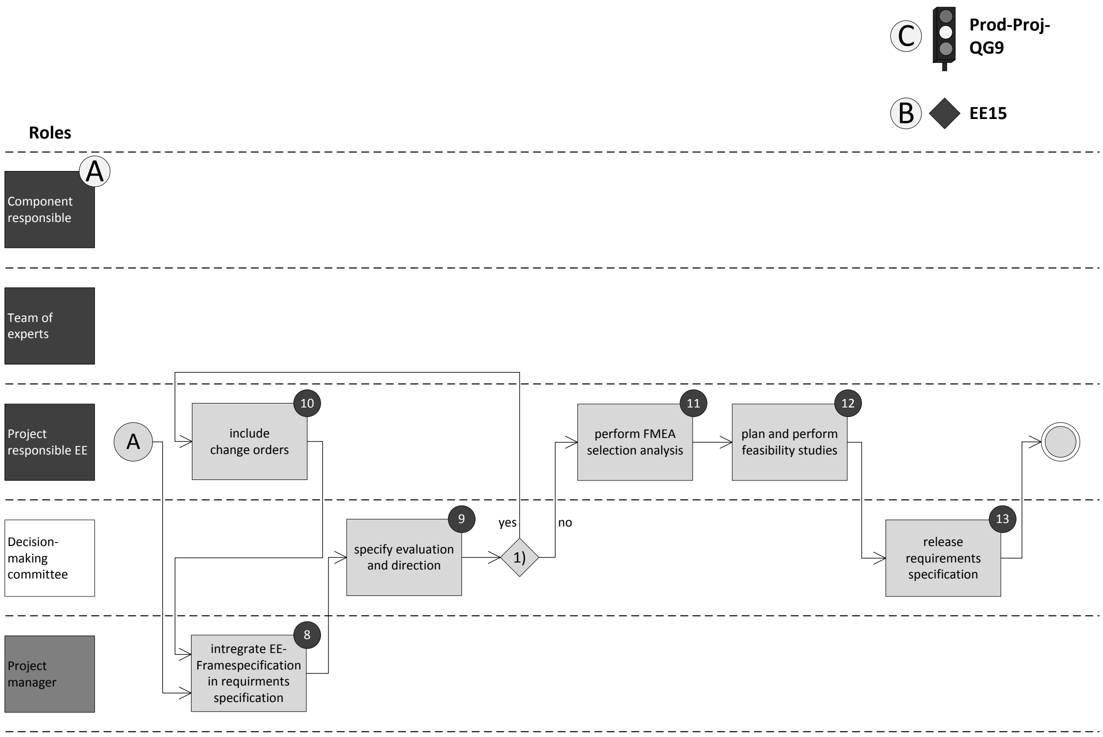


Figure 1.13: Logic-based visualization of a process model.

Figure 1.14 shows a *text-based* visualization which provides a detailed description of a single process task (i.e., the zoom is on this task solely). Such visualization might be helpful, for example, for requirements engineers or new employees, since process tasks are described in a detailed manner (e.g., using a bullet list). Additional information may displayed as well, e.g., the process participant may get informed about inputs and outputs, tools, and guidelines supporting the task execution.

In practice, more dynamic ways to navigate within process model collections need to be provided to users. Furthermore, alternative ways of visualizing particular process information are required. In response to these needs this thesis introduces the ProNaVis framework. The latter enables process navigation based on a 3-dimensional navigation space. In particular, this navigation space allows process participants

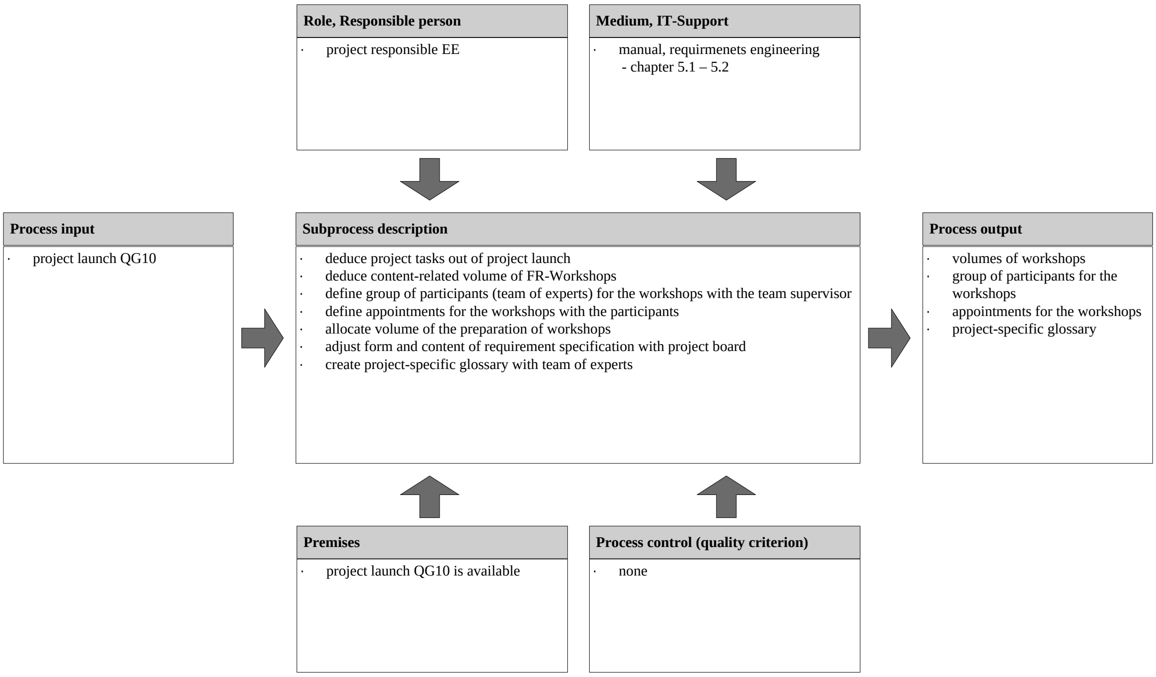


Figure 1.14: Text-based visualization of a process task.

to flexibly navigate within process model collections as well as to dynamically select the visualizations preferred by them.

1.3 Research Questions

The following research challenges need to be tackled: *first*, we must understand the practical problems from a process participant's point of view, i.e., the problems encountered when working with process model collections. Based on this, requirements concerning the navigation and visualization of process model collections can be derived. *Second*, user-driven process navigation must be enabled, i.e., various user interactions with process model collections must be supported. *Third*, the visualization of process model collections should be personalized, i.e., we need to address the way information is presented to the user. *Fourth* and *fifth*, we must investigate which factors need to be considered to evaluate different navigation approaches and different visualizations respectively. *Sixth*, we have to evaluate, how process navigation effectively influences and supports process participants in their day-to-day work.

Based on these challenges, we derive six research questions guiding the research addressed by this thesis. Thereby, we distinguish between *knowledge problems* (KP) and *world problems* (WP) [WMMR05, WH06]. A *knowledge problem* is a difference between what we know about the world and what we would like to know [WH06]. Knowledge problems can be solved by asking others, by reviewing the literature, or by doing research. Knowledge problems have stakeholders, namely the people who would like to acquire the desired knowledge. Research problems typically constitute knowledge problems in which we search for true propositions. In turn, *world problems* are engineering problems, in which we search for an improvement of the world with respect to some goals [WMMR05]. The evaluation criteria for answering both kinds of problems are quite different: *truth* in case of knowledge problems and *goal achievement* in case of world problems.

- **Research Question 1 (KP):** What are existing problems and requirements regarding the navigation within process model collections as well as the visualization of the latter from the perspective of the end user?
- **Research Question 2 (WP):** How should a navigation concept for process model collections be approached?
- **Research Question 3 (WP):** How may process model collections be visualized in a comprehensible manner?
- **Research Question 4 (WP):** How can the benefit of a user-driven navigation concept be measured?
- **Research Question 5 (WP):** How can comprehensibility and aesthetic appearance of process visualizations be measured?
- **Research Question 6 (WP):** How does the navigation concept support process participants in their daily work?

These research questions constitute the foundation of this thesis.

1.4 Research Methodology

Figure 1.15 shows the research methodology underlying the thesis according to [HMPR04, WH06]. It comprises four phases: (1) *problem analysis*, (2) *requirements analysis*, (3) *solution design*, and (4) *solution validation*.

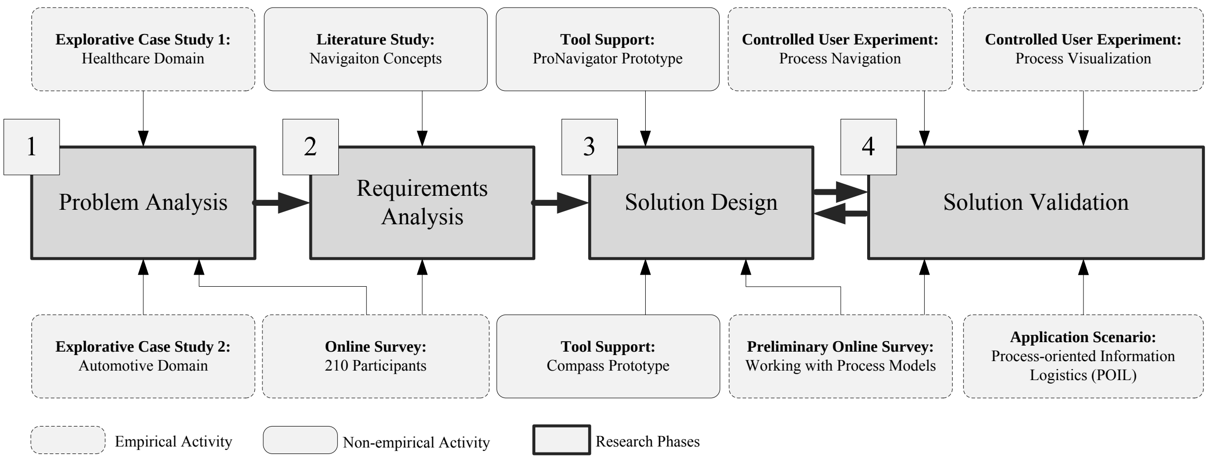


Figure 1.15: Research methodology.

In two preliminary case studies we investigate how process model collections are managed and how they are made available to process participants. Expert interviews are performed with employees possessing different roles. Further, we consider different domains, i.e., automotive engineering and healthcare. The insights we gathered from these studies shall allow us to better understand the problem to be investigated

(*Phase 1*). An additional online survey, a literature study, and practical experiences gathered in the automotive domain will further allow us to derive requirements regarding the navigation within process model collections and their visualization (*Phase 2*). Specifically, Research Question 1 is addressed in these first two phases. In *Phase 3*, sophisticated navigation and visualization concepts are developed based on the results of Phases 1 and 2. To illustrate the applicability of these concepts, *ProNavigator* is provided as proof-of-concept prototype. In turn, the *Compass* tool has been developed in cooperation with an industrial partner to apply the concepts in the automotive domain (Phase 3). In particular, Compass supports knowledge workers dealing with complex E/E process model collections. This phase addresses Research Question 3. Finally, the navigation concepts are evaluated based on controlled user experiments (*Phase 4*). One of these experiments focuses on user interactions, i.e., process navigation, whereas the other deals with the visualization of process model collections. The results obtained will then be used to answer Research Questions 4 to 6.

1.5 Contribution

The contributions of this thesis are as follows:

- We present requirements regarding the navigation within process model collections as well as their visualization from the perspective of process participants. These requirements are derived from practical experiences, case studies, and an online survey.
- We identify existing navigation and visualization approaches for complex information spaces and compare them in respect to the requirements we identified.
- We develop the ProNaVis framework that enables sophisticated navigation possibilities and visualization approaches in respect to process model collections. Specifically, we introduce a 3-dimensional navigation space supporting users in navigating within process model collections. It provides a more flexible navigation concept compared to existing approaches.
- The approach is implemented in a proof-of-concept prototype as well as in a software tool developed with an industrial partner.
- We present results of user experiments and an online survey to validate the approach.

1.6 Outline

The thesis is organized as follows (cf. Figure 1.16). *Part I* introduces the topic. While *Chapter 1* motivates the thesis, *Chapter 2* elicitates the requirements for a flexible process navigation and visualization.

Part II presents the ProNaVis framework: *Chapter 3* sketches the ProNaVis approach in a nutshell. *Chapter 4* then describes the chosen navigation space in more detail, whereas *Chapter 5* presents its formalization. *Chapter 6* introduces visualization concepts and *Chapter 7* shows how the process space can be applied in practice.

Part III validates the ProNaVis framework. *Chapter 8* discusses work related to process navigation. *Chapter 9* presents proof-of-concept prototypes. *Chapters 10* and *11* deal with two controlled user experiments, evaluating the navigation and visualization concepts of the developed approach. *Chapter*

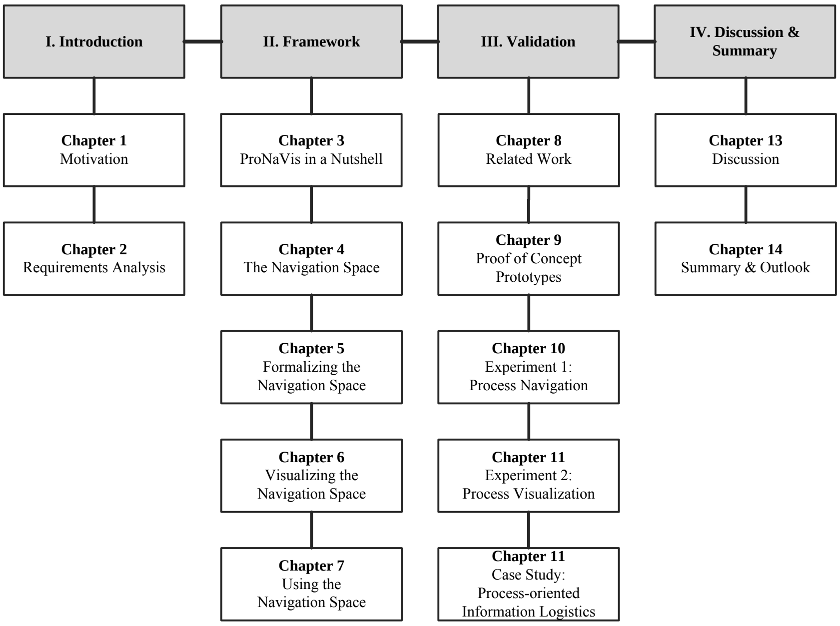


Figure 1.16: Outline of the thesis.

12 demonstrates how the ProNaVis framework can be applied to enable process-oriented information logistics.

Part IV discusses (*Chapter 13*) and summarizes (*Chapter 14*) the main contributions of the thesis.

Figure 1.17 indicates which research question is addressed by which chapters.

	I		II					III					IV	
Research Questions / **Chapters**	Chapter 1	Chapter 2	Chapter 3	Chapter 4	Chapter 5	Chapter 6	Chapter 7	Chapter 8	Chapter 9	Chapter 10	Chapter 11	Chapter 12	Chapter 13*	Chapter 14*
Research Question 1	●	●						●						
Research Question 2			●	●	●		●		●	●	●	●		
Research Question 3						●			●	●	●	●		
Research Question 4					●		●		●	●	●	●		
Research Question 5						●			●	●	●	●		
Research Question 6									●	●	●	●		

* not relevant with respect to the defined research questions

Figure 1.17: Answering the research questions.

2 Requirements Analysis

This chapter[1] presents results from three empirical studies we performed to investigate the issue of navigating in large process model collections and their visualization: two exploratory case studies from the healthcare and automotive domains as well as an online survey with 219 participants. In a *first* step, we identify and describe problem areas with respect to process model collections in general as well as the navigation within process model collections and their visualization in particular. In this context, we adopt a strict end user perspective, i.e., we perform interviews with various process participants. In a *second* step, we derive requirements related to the user-driven navigation within process model collections and the proper visualization. Altogether, Chapter 2 addresses Research Question 1 (cf. Section 1.3):

What are existing problems and requirements regarding the navigation within process model collections and their visualization from a user's perspective?

As specific goal, the two case studies shall identify *problem areas* hampering the effective handling and use of process model collections from an end user perspective. Thereby, each problem area is investigated by tackling two *viewpoints* (cf. Figure 2.1).

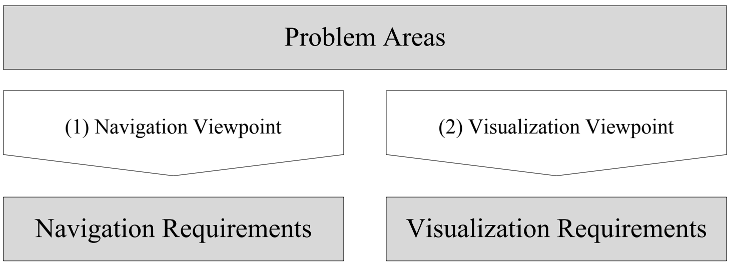


Figure 2.1: Deriving requirements from problem areas.

The *navigation viewpoint* deals with problems and challenges related to the navigation within large process model collections. Problems may be caused by different sources:

- Management requirements (e.g., requirements for documenting process models)
- Organizational structures (e.g., departments or business units)
- Governance rules (e.g., rules dealing with the access to process models)
- Compliance rules (e.g., rules addressing the protection and archiving of process information)

[1] The chapter is based on the following referred paper[HMR11b]:
Markus Hipp, Bela Mutschler, and Manfred Reichert. *On the Context-aware, Personalized Delivery of Process Information: Viewpoints, Problems, and Requirements.* in: Proc 6th Int'l Conf on Availability, Reliability and Security (ARES'11), LNCS 6908, pp. 390–397, Springer, 2011

In turn, the *visualization viewpoint* deals with issues related to the end user presentation and visualization of process model collections. Usually, such problems are related to user interface design. When displaying too much information, for example, process participants are rather disturbed [vWN04].

Based on the derived problem areas, we consider both viewpoints to identify more specific problems, which then can be used to derive specific requirements for enabling a navigation and visualization support for process model collections. Depending on the considered viewpoints, requirements are either categorized as *navigation requirement (NavReq)* or as *visualization requirement (VisReq)*.

The remainder of this chapter is organized as follows. Sections 2.1 and 2.2 present problem areas and requirements related to the two case studies. Section 2.3 then discusses results from the conducted online survey. Section 2.4 summarizes the derived requirements and Section 2.5 concludes this chapter.

2.1 Case Study 1: Clinical Domain

The first case study took place in a large hospital in Southern Germany [HMR11b]. Eight interviews were performed in five different departments, taking about 45 minutes on average. The sequence of the interviews followed a characteristic patient treatment process starting with patient admission and ending with the invoicing. We were able to interview all stakeholders (doctors, nurses, administrative staff etc.) involved in the process, i.e., all process participants.

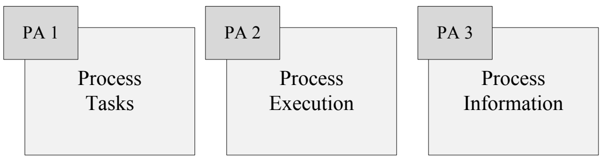


Figure 2.2: Problem areas from case study 1 (healthcare domain).

In this case study we identified Problem Areas 1-3 (cf. Figure 2.2):

2.1.1 Problem Area 1: Process Tasks

In hospitals, the proper execution of process tasks related to patient treatment is crucial. However, the definition and documentation of process tasks is often not sufficient to support clinical staff in executing tasks in the best possible way. We can consider this issue both from the navigation and visualization viewpoint:

Navigation Viewpoint: Interviewees state that many process tasks are not defined properly. As example consider the task of patient admission for which (executive) guidelines only exist in paper form and are thus hard to find. Furthermore, this task is usually performed by the admission department. However, in emergency cases, patients may be admitted by nurses in a ward. Consequently, this task is performed by experienced clinical staff in the first case, and by non-experienced one in the second. We can conclude:

NavReq #1: Depending on a process participant's experience, the level of detail regarding a process task should be adjustable.

In addition, the distributed execution of process tasks might be critical. As a task may be performed by different departments (i.e., it may be documented in different process models), it is hard to identify these tasks across different process models. In this context, accessing only one single process model at once hampers process participants in obtaining an overview on tasks being executed by different departments (i.e., documented in different process models), and thus the entire process model collection. Thus, we can conclude:

NavReq #2: Process participants should be able to adjust the level of detail regarding process model collection in order to obtain a quick overview on a specific task that is currently executed.

Visualization Viewpoint: Especially, non-experienced staff complained about missing task descriptions, directly accessible in the context of process models. Usually, finding documents including this information is time-consuming, and might thereby affect the patient treatment. Moreover, only small parts of these task descriptions are needed. Therefore, documents comprising up to hundreds of pages in total must be manually searched. However, interviewees explained that information has to be intuitively and quickly understandable. Thus, we can conclude:

VisReq #1: Task descriptions must be documented in a well understandable manner.

2.1.2 Problem Area 2: Process Execution

During patient treatment, a patient passes through different departments, e.g., admission, radiology, and surgery. In this context, the documentation of single process tasks are only available in the departments where the task is executed.This might affect the seamless execution of the entire patient treatment process.

Navigation Viewpoint: From the navigation viewpoint, a specific problem is the missing linkage of process tasks (in the patient treatment) across different departments, i.e., process tasks (and their descriptions) should be accessible across different departments. In particular, medical departments are often unaware of the current status of process tasks corresponding to processes from other areas. In turn, the seamless execution of cross-departmental patient treatment processes can not be guaranteed. Thus, we can conclude:

NavReq #3: Users should be enabled to access process tasks in other process areas.

Visualization Viewpoint: Participants stated that communication between departments was suboptimal. This is of particular importance when taking temporal constraints into account. Think of a notification of the operation theatre when patients need to be transferred back to their ward. In particular, temporal relations between process tasks need to be explicitly visualized, e.g., the relation of a process task (e.g., an x-ray examination) to its preceding and subsequent tasks. Thus, we can conclude:

VisReq #2: Temporal and logical dependencies must be considered when visualizing processes.

2.1.3 Problem Area 3: Process Information

The patient record represents process information needed treatment. Interviewees stated that such records are often managed in paper-based form. Thus, the record can only be used in one single process task at the same time. Hence, several problems occur.

Navigation Viewpoint: In the context of paper-based medical records, both the access to patient information (e.g., findings from an x-ray examination) and the retrieval of needed information (e.g., on medical problems of the patient) constitute delaying and time-consuming tasks for process participants. In turn, this leads to another problem: if needed patient information is not complete, process participants must search for it. Figure 2.3b summarizes answers of interviewees on the question whether important information related to a specific process task can be quickly found. As can be seen, the *median* is neither agree nor disagree. However, some interviewees seem to have problems with finding information needed.

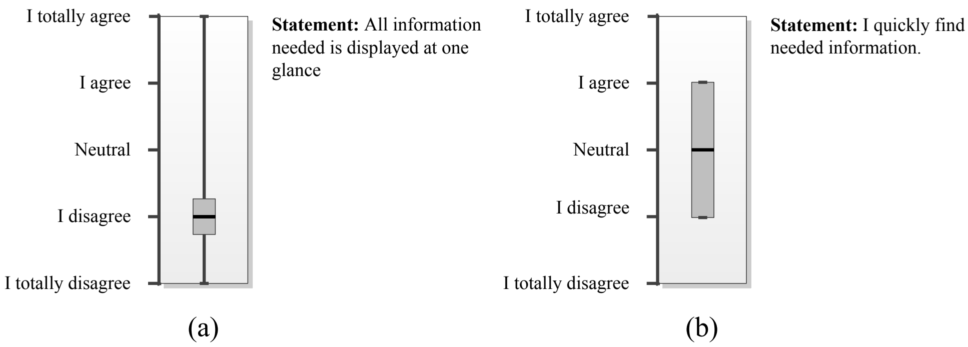


Figure 2.3: Handling of information 1.

All participants argued that quickly finding information is easier for experienced staff. New employees, in turn, confirmed to have difficulties with this. Thus, we can conclude:

NavReq #4: Relevant process information must be accessible at the level of single process models from the process model collection.

Visualization Viewpoint: Since medical records may become large during patient treatment, the visualization of process-related information must be adapted (e.g., only specific views on this data should be presented to users, depending on the executed process task). Figure 2.3a shows that the participants disagreed with the statement that the exact information needed shall be displayed at a glance. For example, earlier medication of the patient has to be identified within hundreds of handwritten sheets. Additionally, the way of presenting data to users must be adopted. For example, temperature curves should be visualized as graphs, whereas the actual medication should be displayed as a table. Thus, we can conclude:

VisReq #3: Complex process information must be visualized in a comprehensible manner.

Table 2.1 summarizes the requirements derived from case study 1.

Req #	Requirement	Source		
		PA 1	PA 2	PA 3
NavReq #1	Depending on a process participant's experience, the level of detail regarding a process task should be adjustable.	●		
NavReq #2	Process participants should be able to adjust the level of detail regarding process model collection in order to obtain a quick overview on a specific task that is currently executed.	●		
NavReq #3	Users should be enabled to access process tasks in other process areas.		●	
NavReq #4	Relevant process information must be accessible at the level of single process models from the process model collection.			●
VisReq #1	Task descriptions must be documented in a well understandable manner.	●		
VisReq #2	Temporal and logical dependencies must be considered when visualizing processes.		●	
VisReq #3	Complex process information must be visualized in a comprehensible manner.			●

Table 2.1: Requirements derived from case study 1.

2.2 Case Study 2: Automotive Domain

The second case study was conducted in the automotive domain [HMR11b]. Nine interviews with seven knowledge workers and two decision makers were conducted. Each interview lasted about 60 minutes. Like in the first case study, participants were selected based on a typical development process for car control units.

In addition to the problem areas identified in case study 1, two additional problem areas (PA) were identified (cf. Figure 2.4). Note that problem areas 1-3 are also applicable in the automotive domain:

2.2.1 Problem Area 4: Roles

The responsibility for process tasks and process models is managed by different roles. Employees are assigned to specific roles based on their skills and competencies. Examples of such roles include *requirements engineer*, *test engineer*, *quality manager*, and *process owner*. In particular, respective roles are required to execute process tasks. The proper definition of roles is important for other employees, for example, to be able to quickly identify contact persons if needed.

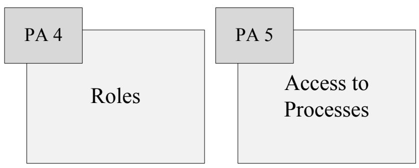


Figure 2.4: Problem areas from case study 2 (automotive domain).

Navigation Viewpoint: A major problem concerns the insufficient definition of roles across departmental borders. On one hand, roles are often not completely defined (as important information on tasks and competencies is missing). Role *process owner* was considered as a typical example of incompletely defined roles by most interviewees. According to its definition, a process owner is responsible for an entire process. In practice, however, several people may be responsible for different process tasks. On the other hand, role definitions are not consistently used across departmental borders, e.g., process owners may have different responsibilities depending on the different business units. Thus, we can conclude:

NavReq #5: Roles must be globally defined in a detailed manner.

Visualization Viewpoint: From the visualization viewpoint, it is hard for process participants to identify role affiliations (e.g., which role is responsible for process models or process tasks) as documentation is inconsistent in this respect. Thus, we can conclude:

VisReq #4: Information about roles must be intuitively identifiable.

2.2.2 Problem Area 5: Access to Processes

Interviewees reported on needs regarding the access to processes. These needs may even vary for single process participants due to a continuously changing work context.

Navigation Viewpoint: From the navigation viewpoint, participants argued that needed information is not provided at an appropriate level of detail (e.g., depending on the user's role). A knowledge worker, for example, requires detailed information on single process tasks (e.g., on guidelines, checklists or tools he uses). Managers, in turn, need more abstract information, e.g., on an entire process as well as its dependencies on other processes. Thus, processes need to be aggregated and provided on different levels of details to fit the needs of process participants with different roles. Thus, we can conclude:

NavReq #6: Process participants must be able to access process models on different levels of detail.

Visualization Viewpoint: From a visualization viewpoint, process participants stated that accessing and executing a process task often resulted in an information overload (cf. Figure 2.5a). In this context, five out of nine participants rated the amount of available information as too high. Moreover, the same number of participants totally disagreed that needed information is displayed at a glance (cf. Figure 2.5b).

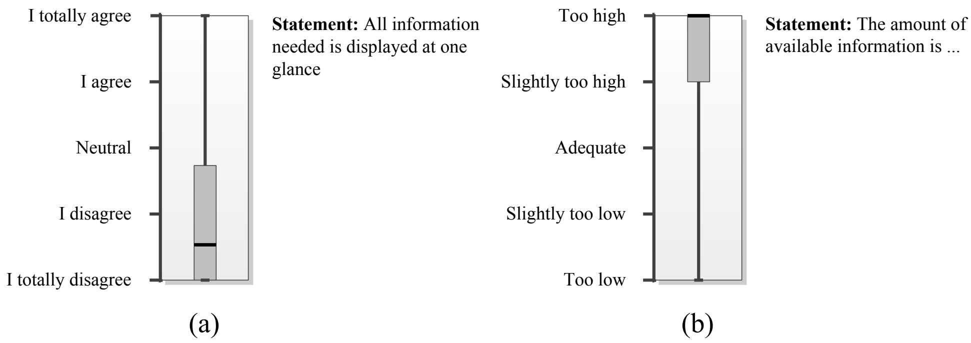


Figure 2.5: Handling of information 2.

As major reason for that, process participants have to find the right information from distributed data sources, managed by different departments. In this context, information should be visualized in a way suitable to support process participants in their resprctive working contexts. Users should not feel overtaxed by the amount of information provided. Thus, we can conclude:

VisReq #5: The amount of visualized information should not overload process participants.

Table 2.2 sums up the requirements derived from our second case study.

Req #	Requirement	Source	
		PA 4	PA 5
NavReq #5	Roles must be globally defined in a detailed manner.	●	
NavReq #6	Process participants must be able to access process models on different levels of detail.		●
VisReq #4	Information about roles must be intuitively identifiable.	●	
VisReq #5	The amount of visualized information should not overload process participants.		●

Table 2.2: Requirements derived from case study 2.

2.3 Online Survey

To further validate case study results, we performed an additional online survey. 219 people (73% male, 27% female) from more than 100 companies participated. The majority of them (96%) was located in Germany. 57% of the participants are knowledge workers, 26% are decision makers and 17% provided no information about their position.

First, we asked participants about the benefits of process portals (cf. Figure 2.6, Statement 1). 85.85% of them totally or somewhat agree that central access to process information would help them in their daily work (cf. *NavReq #3*). More specifically, 18.72% totally agree that step-by-step guidance regarding past, current and future process tasks would be benefical for them, too (cf. Figure 2.6, Statement 2). 39.66% somewhat agree with that statement.

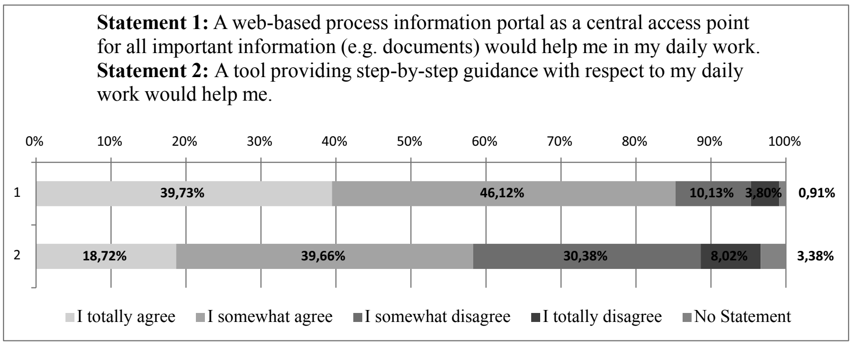


Figure 2.6: Online survey results 1.

Second, we addressed the context-sensitive provision of process information. As depicted in Figure 2.7 (Statement 1), the majority of respondents (76.71%) totally or somewhat agree with the statement that it would be helpful to automatically get relevant information depending on the current process context (cf. *NavReq #2*). Only 5.48% totally disagree. We further ask for the relevance of a continuously available process overview (cf. Figure 2.7, Statement 2). 30.59% totally agree that such an overview would be helpful. 42.92% somewhat agree (cf. *NavReq #6*).

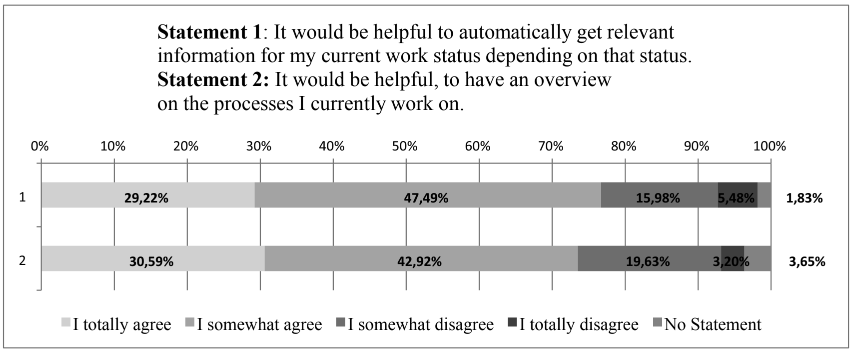


Figure 2.7: Online survey results 2.

Finally, we asked for user preferences when retrieving information. In our case study interviews, the use of search functions was mentioned very often. Our online survey (cf. Figure 2.8) confirms this. Specifically, we asked for the most common way to retrieve information. While 40.18% of the respondents use search functions, 40.65% of them prefer navigating along existing structures, e.g., along folder structures in file explorers (cf. *NavReq #1* and *#6*).

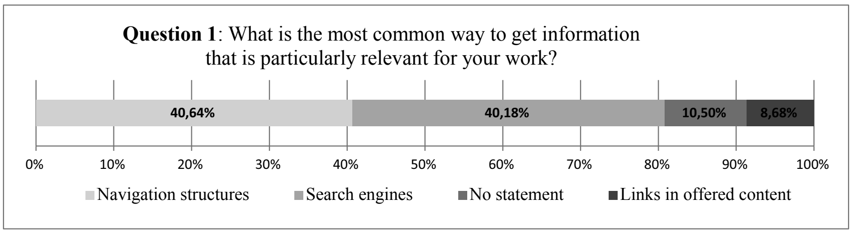


Figure 2.8: Online survey results 3.

In particular, providing processes on different detail levels is important (cf. *NavReq #1* and *#6*). Therefore, process participants must be able to interact with process model collections, i.e., to navigate to the right process representations on the right detail level (cf. Figure 2.9).

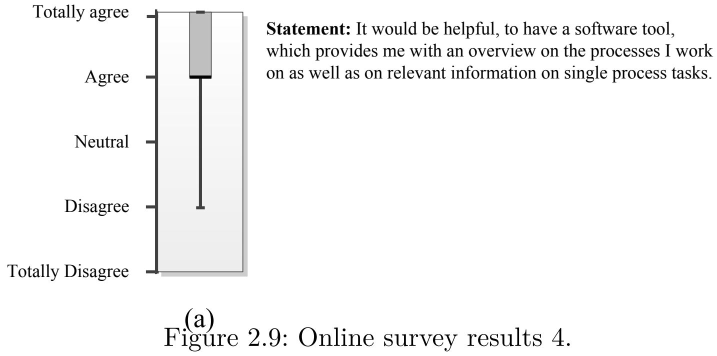


Figure 2.9: Online survey results 4.

2.4 Requirements at a Glance

Table 2.3 lists all derived requirements from both case studies.

Req #	Requirement	Source				
		PA 1	PA 2	PA 3	PA 4	PA 5
NavReq #1	Depending on a process participant's experience, the level of detail regarding a process task should be adjustable.	●				
NavReq #2	Process participants should be able to adjust the level of detail regarding process model collection in order to obtain a quick overview on a specific task that is currently executed.	●				
NavReq #3	Users should be enabled to access process tasks in other process areas.		●			
NavReq #4	Relevant process information must be accessible at the level of single process models from the process model collection.			●		
NavReq #5	Roles must be globally defined in a detailed manner.				●	
NavReq #6	Process participants must be able to access process models on different levels of detail.					●
VisReq #1	Task descriptions must be documented in a well understandable manner.	●				
VisReq #2	Temporal and logical dependencies must be considered when visualizing processes.		●			
VisReq #3	Complex process information must be visualized in a comprehensible manner.			●		
VisReq #4	Information about roles must be intuitively identifiable.				●	
VisReq #5	The amount of visualized information should not overload process participants.					●

Table 2.3: Overview on derived requirements.

Requirements regarding the *navigation viewpoint* mainly emphasize the need for a user-driven navigation in order to be able to consider processes on different levels of detail. Especially, an overview of processes from different areas is required by interviewees from the management level. Thereby, they are able to estimate the dependencies between processes, being executed in different areas. In turn, knowledge workers need specific support for executing a single process task, i.e., detailed task descriptions and additional documents such as guidelines or checklists.

Requirements from the *visualization viewpoint* identify the need to present processes in different ways, i.e., needed information shall be quickly and intuitively identifiable. For example, when managers view different processes at once (i.e., on an abstract detail level), graphical visualizations may be better for identifying the dependencies between process tasks from different processes. In turn, task descriptions provide better information when presented in a structured, textual manner, e.g., organized as a bullet point list.

2.5 Summary

This chapter presented results from two exploratory case studies and an online survey. Detailed insights have been given into the work routines emerging in the context of business processes (e.g., patient treatment processes). Based on interviews with process participants possessing different roles, we were able to identify major problem areas that emerge when working with process model collections: *Process*

Tasks, Process Execution, Process Information, Roles, and *Access to Process Models.* Tackling these problem areas, we were able to derive requirements on the navigation in and visualization of process model collections. All requirements have been considered as basis for developing the ProNaVis framework. Considering the similar observations we made in the two different domains, we may assume that the requirements are applicable to other domains as well.

Part II

Framework

3 ProNaVis in a Nutshell

Chapters 1 and 2 have revealed that process participants in different roles need specific perspectives on the same process model collection. A business manager, for example, is mainly interested in an abstract visualization of process models to obtain a quick overview of currently running tasks, whereas requirements engineers need more detailed information about the process tasks they are working on.

To support process participants in accessing process model collections in a flexible way, a user-driven process navigation and visualization approach is required. In particular, users should be enabled to flexibly interact with process model collections. More specifically, process navigation shall allow users to navigate across different levels of detail as well as alternative visualizations of a process model collection.

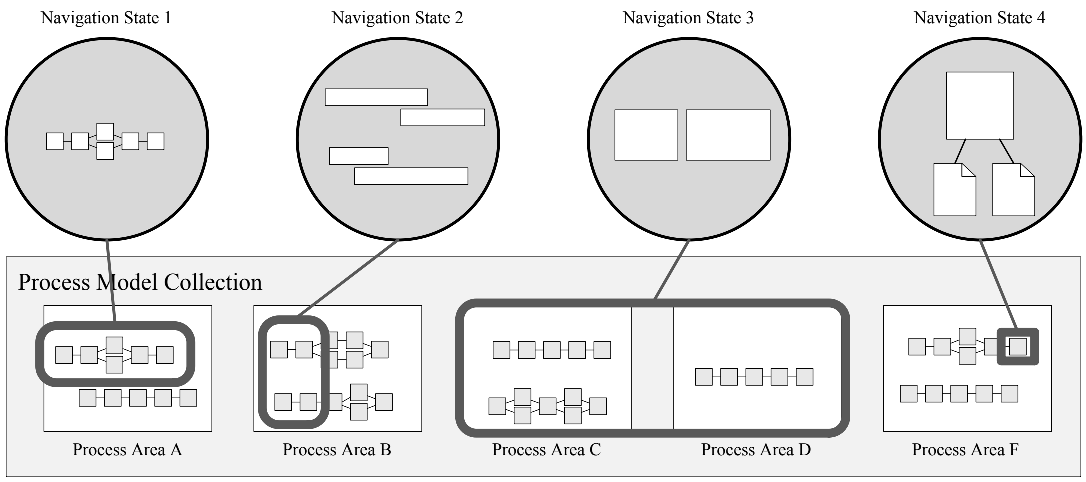


Figure 3.1: Examples of four different navigation states in a process model collection.

For this purpose we introduce the notion of *navigation states*. A navigation state defines the current position of a process participant within a process model collection, including the detail level, zoom level, and type of visualization. Navigating within a process model collection then means that the user switches between different navigation states. Figure 3.1 exemplarily shows four navigation states of a process model collection. Navigation state 1 focuses on a single process model. The latter could be visualized, for example, based on BPMN. Navigation state 2, in turn, focuses on the first two process tasks of two different process models from the same process area. In this case, the tasks are visualized using a Gantt Chart. Furthermore, navigation state 3 focuses on two process areas on a more abstract detail level: only the process areas are visualized, whereas the detailed process models are not displayed. Finally, navigation state 4 focuses on a single process task on a detailed level, i.e., the task is visualized along with the related process information (e.g., data-objects).

This chapter[1] introduces the ProNaVis framework, a flexible navigation and visualization framework allowing process participants to flexibly navigate within process model collections on different detail levels, zooming levels, and visualization forms. Thereby, we use process states as basis elements for process navigation.

The remainder of this chapter is structured as follows. Section 3.1 discusses basic issues regarding process navigation. Section 3.2 presents a running example and Section 3.3 introduces basic ideas underlying the ProNaVis framework. In Section 3.4, these concepts are applied to the running example. Section 3.5 summarizes the chapter.

3.1 Basic Issues

Figure 3.2 illustrates *process navigation* based on different navigation states. The process participant starts from an initial navigation state 1, which corresponds to a default representation of the process model collection (e.g., process areas on an abstract level). By zooming into a specific part of the process model collection, for example, the user changes the level of detail, switching to navigation state 2. In turn, the latter includes more detailed information on process areas C and D. Through another zooming interaction, navigation state 3 is reached. The focus of this state is on one particular process model from area D. Finally, users might change the view of this process model to a Gantt Chart, i.e., they might change its visualization. This interaction leads to a transition to navigation state 4.

Process interaction is an activity allowing process participants to move from one navigation state to another one based on user-triggered operations. For example, a user may adjust the level of detail or the way of visualization. The navigation state then changes accordingly. Process navigation comprises a sequence of interactions and allows process participants to navigate within a process model collection, e.g., from a default navigation state (navigation state 1) to a more specific one (navigation state 4).

To enable process navigation in model collections, approaches from other domains could be adopted (cf. Chapter 8). Especially in the area of *geographic information systems*, complex navigation concepts have been established. The ProNaVis framework is particularly inspired by navigation concepts known from Google Maps.[2]

Generally, process models and process model collections constitute complex information spaces. In turn, Google Maps provides a navigation concept for one of the most complex existing information spaces, namely the global geographical information space of the earth. Of course, there exist significant differences between process models and global geographical information. Hence, we consider the Google Maps navigation approach as the starting point of our approach.

Google Maps is a virtual globe, map and geographic information system. It displays satellite images of varying resolution of the earth's surface, allowing users to browse items like cities and houses looking perpendicularly down or at an oblique angle [Ray10]. Google Maps allows users to search for addresses of certain countries, to enter coordinates, or to simply use the mouse to navigate to a particular location. Specifically, user interaction is enabled within the information space via two independent navigation dimensions.

[1] The chapter is based on the following referred paper [HMR11a]:
Markus Hipp, Bela Mutschler, and Manfred Reichert. *Navigating in Process Model Collections: A new Approach Inspired by Google Earth.* in: Proc 1st Int'l Workshop on Process Model Collections (PMC'11), LNBIP 100, pp. 87–98, Springer, 2011

[2] http://maps.google.com

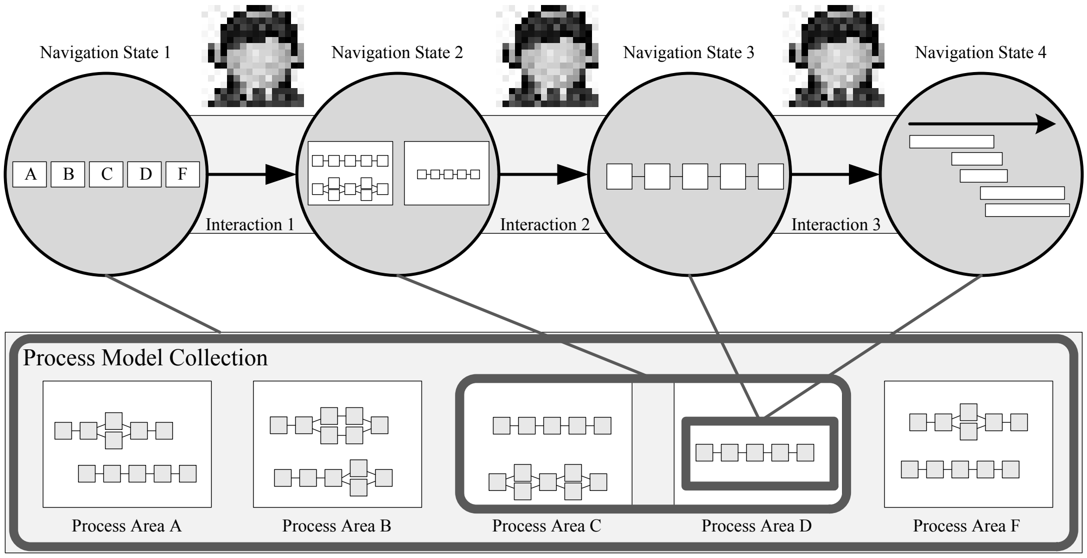


Figure 3.2: Process navigation.

The *zooming dimension* allows user to zoom into a certain part of a map in order to focus on the area of interest. The smaller this area becomes, the more detailed information is presented on the map (e.g., villages become visible only when a small area of the map is focused). Thus, the level of zooming is related to the level of detail, and determines the information to be presented on the screen. The zooming dimension addresses the *process navigation* aspect of the ProNaVis framework, as user-driven actions lead to changing navigation states of the underlying information space. In turn, the *visualization dimension* allows users to switch between different visualizations, i.e., different ways of displaying information (e.g., the *satellite visualization* uses real world pictures of the earth's surface, whereas the *map visualization* focuses more on structural elements) (cf. Figure 3.4). Note that both visualizations are based on the same navigation state, i.e., the same objects are visualized in a different way. Thereby, the zooming dimension is not affected when switching between different visualizations. The visualization dimension picks up the *process visualization* aspect of the ProNaVis framework.

Both navigation dimensions are independently adjustable by users in Google Maps, i.e., the user may switch the visualization independently from the current detail level on the map. We pick up these navigation dimensions as a basoc pillar of the ProNaVis framework in order to allow the flexible navigation in complex process model collections.

To provide a better understanding of how the Google navigation approach can be applied to process model collections, we present an illustrating example from a process model collection from the automotive domain.

3.2 Running Example

Our example comprises process models dealing with the electric/electronic (E/E) development of car control units. Currently, limited process navigation possibilities are provided to the user in a process portal (cf. Chapter 1). In the existing portal, all process models are documented in terms of process diagrams captured in documents (e.g., pdf files). Furthermore, they are categorized into process areas based on

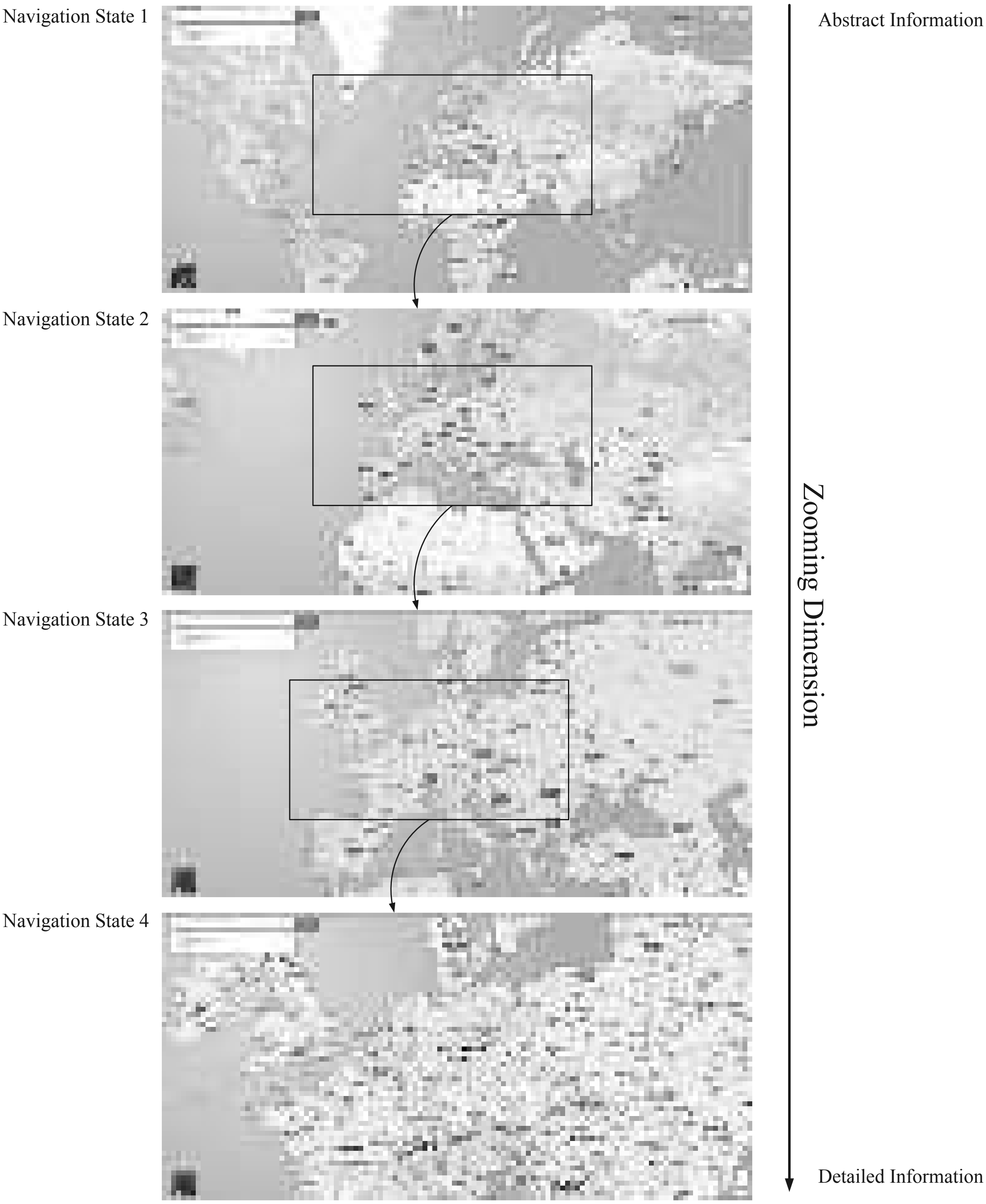


Figure 3.3: Zooming dimension of Google Maps [Ray10].
[Map data: ©2014 Google, INEGI, 2014 Basarsoft, Mapa GISrael, basado en BCN IGN Espana, ORION-ME, 2015 GeoBasis-DE/BKG (©2009), Google]

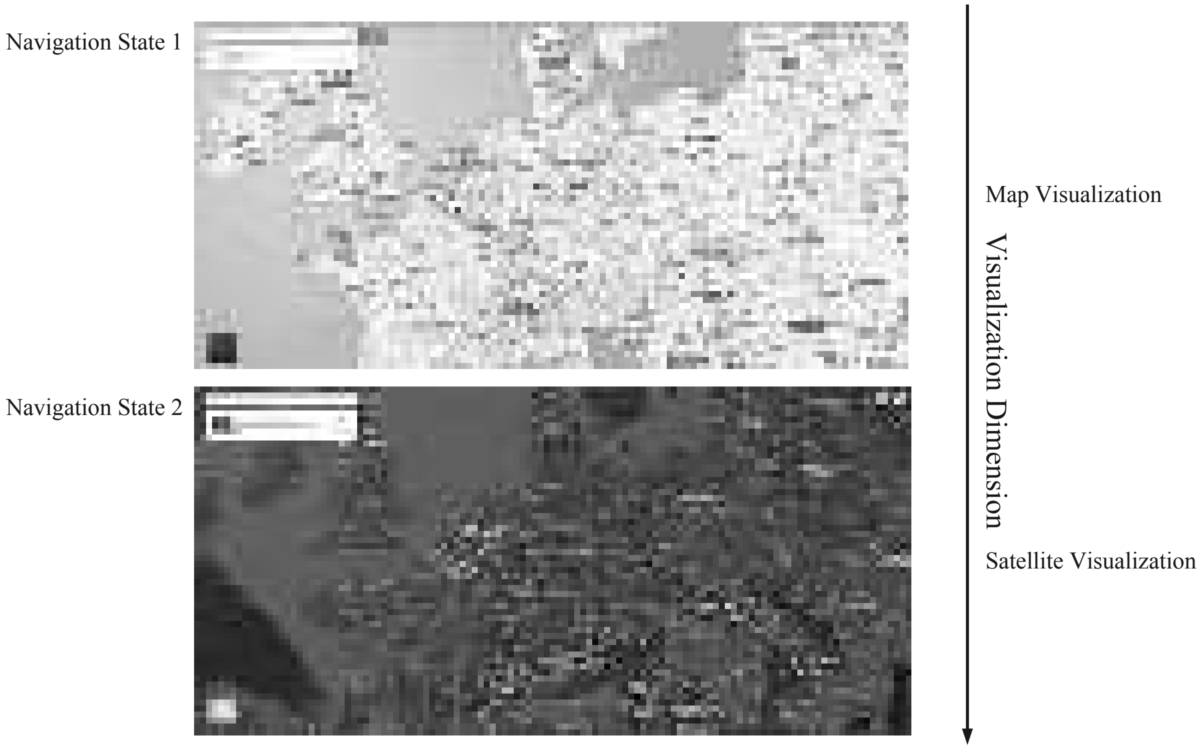
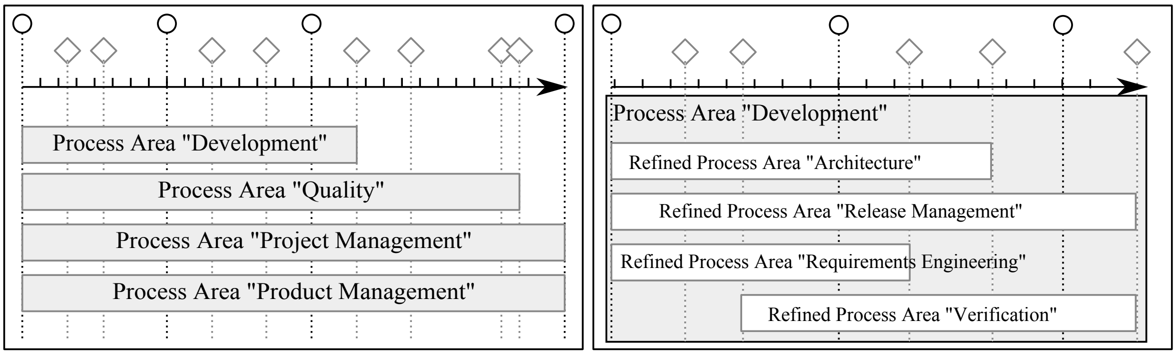


Figure 3.4: Visualization dimension of Google Maps [Ray10].
[Map data: ©2014 Basarsoft, GeoBasis-DE/BKG (©2009), basado en BCN IGN Espana, 2014 Data SIO, NOAA, U.S. Navy, NGA, GEBCO, Landsat, Kartendaten ©2014, Google]

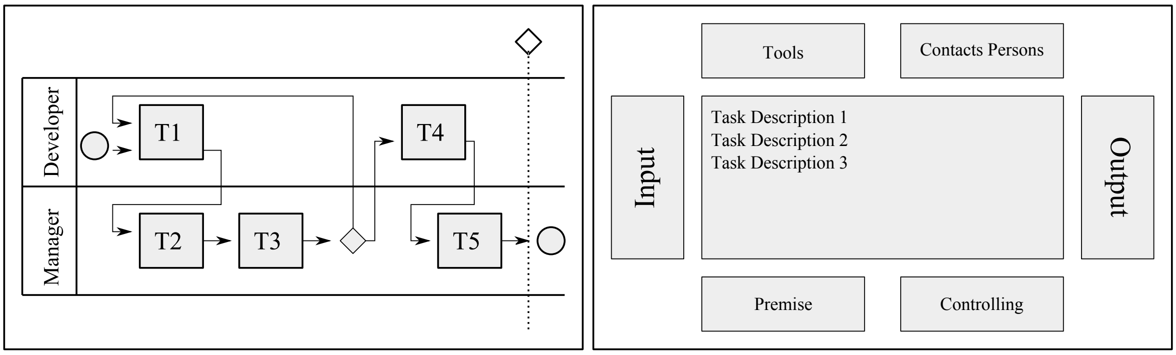
their topics. Moreover, each process area is depicted as image map. Hence, all available navigation states within the process model collection are manually created with the visualization dimension. Therefore, they are hard-wired to the level of detail. Altogether, the entire process model collection comprises various process models and process areas on different levels of detail as well as in different visualizations (cf. Figure 3.5).

Navigation state 1 (cf. Figure 3.5a) covers the entire process model collection, including abstract process areas. As displaying single process models would be too complex at this point, only abstract process areas are depicted. The respective visualization is *time-based*, i.e., the length of rectangles corresponds to the duration of process areas. Navigation state 1 provides a starting point for the process navigation for process participants entering the process portal. Based on it, a process participant may select the process area including the needed process task, e.g., for example, by choosing process area *Development* (by clicking on the according image map), the user navigates to a more detailed, but still time-based visualization of this process area in navigation state 2 (cf. Figure 3.5b). The contents of single process models may then be displayed at Level 3 (cf. Figure 3.5c).

In our example (cf. Figure 3.5), the *Requirements Engineering* process is depicted in terms of a process diagram, in which single process tasks (T1...T5) are linked through a sequence flow to indicate logical relations. Furthermore, roles are introduced on this level and displayed as swimlanes. As opposed to navigation states 1 and 2, the visualization in navigation state 3 is *logic-based*, e.g., it allows modelling



(a) Navigation State 1 - time-based visualization (b) Navigation State 2 - time-based visualization



(c) Navigation State 3 - logic-based visualization (d) Navigation State 4 - text-based visualization

Figure 3.5: Real-world example from the automotive industry.

feedback loops (e.g., to jump back from T3 to T1) in case a certain condition is not met. Each process task is further refined in navigation state 4, which provides a *text-based* visualization neither having time nor logic restrictions. This visualization only contains information about a single process task, i.e., a detailed textual description as well as additional information (e.g., tools or contact persons). The latter navigation state is the most detailed visualization and thus represents an important destination (e.g., for knowledge workers) when searching for specific information needed. Note that for a manager, navigation state 2 (see the time-based visualization in Figure 3.5b) might already be sufficient to meet his specific needs.

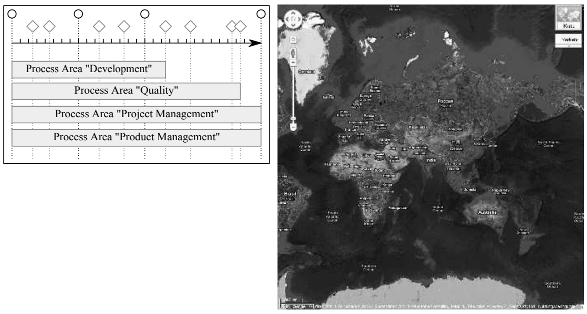
We apply the Google Maps navigation concept and adopt it to the presented example. Table 3.1 shows the four different zooming levels of the previously described process model collection. The goal is to map these levels to the Google Maps navigation approach.

Zooming Level	Process Model Collection	Google Maps
Level 1	Process Model Collection	Globe
Level 2	Process Area	Continent
Level 3	Process Model	Country
Level 4	Process Task	City

Table 3.1: Mapping of terms.

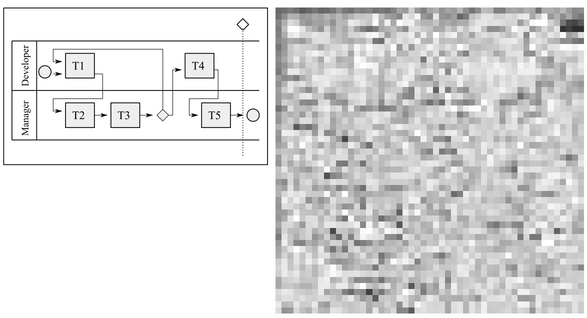
As can be seen in Figure 3.6a, our process model collection (i.e., zooming level 1 of our scenario) corresponds to the entire globe in Google Maps. Process areas, in turn, may be considered as continents (cf. Figure 3.6b). Note that both the globe and the continents are depicted using the same visualization (i.e., the satellite visualization). On zooming level 3 (cf. Figure 3.6c), Google Maps switches to another visualization, namely a map visualization. On this level, Google Maps shows single countries. Picking up again our scenario, a single country corresponds to a single process model from the process model collection. Finally, single process tasks (zooming level 4) correspond to single cities in Figure 3.6d. The visualization has changed again, now to a terrain visualization in Google Maps (i.e., to a text-based visualization in our example).

Obviously, Google Maps can be applied to our process model collection and to its different zooming levels and visualizations. In particular, process navigation as well as process visualization aspects can be reflected in the Google Maps approach.

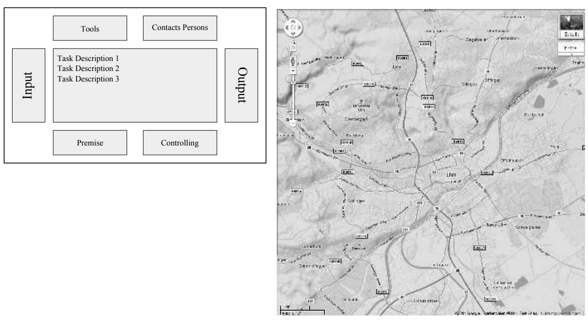


(a) Navigation State 1 - satellite visualization

(b) Navigation State 2 - satellite visualization



(c) Navigation State 3 - map visualization



(d) Navigation State 4 - terrain visualization

Figure 3.6: Mapping a process model collection to Google Maps.
[Map data: ©2011 TerraMetrics, NASA, Kartendaten ©20011 Geocentre Consulting, MapLink, Tele Atlas, Whereis(R), Sensis Pty Ltd, ©2011 Europa Technologies, PPWK, Transnavicom, Barasoft, Google]

However, due to the static navigation states available in the presented process model collection, process navigation within process portals still remains a challenge. Process participants, for example, cannot adjust the zoom levels and visualizations independently, since they are hard-wired and manually defined for each navigation state. Navigation state 3, for instance, is always depicted as a logic-based visualization. In fact, the user may adjust the zooming level (i.e., one dimension, the dimension x in Figure 3.7a), but each visualization obtained is still hard-wired.

The presented example reveals two drawbacks:

1. The representation of the different zooming levels is inconsistent. While navigation states 1 and 2 provide static image maps, navigation states 3 and 4 are represented as pdf files. Navigation from navigation state 3 to navigation state 4 then corresponds to a simple scrolling action in the pdf file.

2. There are missing relations between different processes models. As all process models are documented in pdf files, visualizing multiple process models is not possible.

The Google Maps approach, which is based on the geographic navigation space, in turn, supports navigation in two independent navigation dimensions. The first dimension is the *zooming dimension* (x) (i.e., zooming hard-wired with the level of detail). The second one subsumes different *visualizations* (y). We can depict these two dimensions as a matrix (cf. Figure 3.7b). As we can identify four information levels and three visualizations in our running example, the applied Google Maps navigation concept can be depicted as 4×3 matrix (cf. Figure 3.7c). Thus, twelve navigation states would be possible from a theoretical part of view when using two independent navigation dimensions. Note that this significantly differs from the four static navigation states of the process model collection from Figure 3.5.

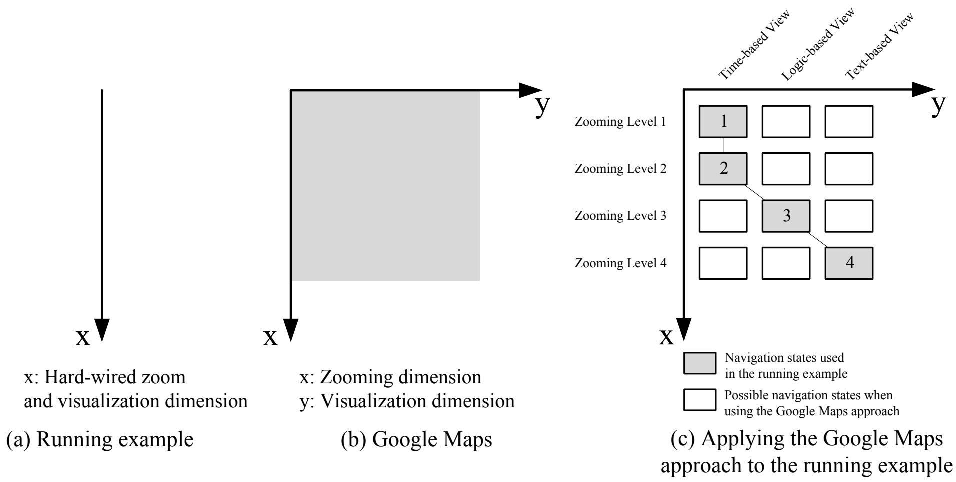


Figure 3.7: Different navigation approaches.

However, the Google Maps navigation concept (with its two navigation dimensions) is still not able to cover all use cases presented in Chapter 1. As example consider use case 5. A quality manager must have an overview over process tasks from distributed process models. Using the Google Maps metaphor, this scenario may be described as follows: The user wants to see certain villages across different countries, but also wants to see all these villages at the same time (possibly spread across the entire globe). The Google Maps navigation concept cannot solve this problem. The user may either zoom in (i.e., the area of interest is limited to one single village, but then looses the overview on the globe at the same time) or zoom out (i.e., the area of interest covers the entire globe, but single villages are not displayed, as the level of detail is too abstract).

In the following, we show how the ProNaVis framework tackles this challenge. Specifically, we show how the Google Maps approach must be enhanced in detail in order to fit all user requirements.

3.3 The ProNaVis Framework

In this section, we present the ProNaVis framework. It enables process participants to navigate within process model collections in different navigation dimensions. In particular, the ProNaVis framework provides access to process model collections on different levels of detail, focusing on specific areas of interest as well as providing different visualizations. Thereby, a major challenge concerns the zooming dimension. Figure 3.8 illustrates how the zooming dimension may be applied to a process model collection. In this example, a process model collection with three process areas is given (*General Specification*, *System Specification*, *Component Specification*). Applying the zooming dimension, focus is on a particular area of the process model collection (*General Specification* in the example). At the same time, information is presented on a higher level of detail (i.e., process models nested within the process area). As a problem the Google Maps navigation concept is unable to display detailed information on an abstract zooming level since the level of detail is hard-wired to the area of interest (i.e., the zooming level).

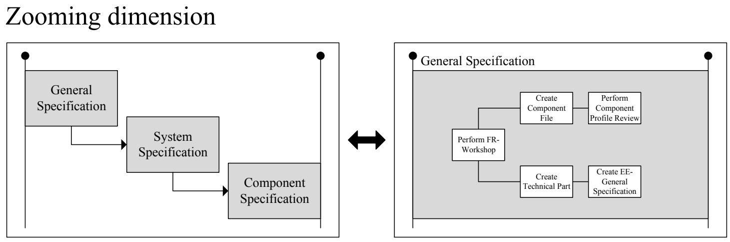


Figure 3.8: The zooming dimension.

To enable flexible process navigation on both different zooming levels and on different levels of detail, the zooming dimension is not sufficient. Thus, we split up the zooming dimension into two independent navigation dimensions: the *semantic* and *geographic dimensions*. The semantic dimension allows distinguishing information on different levels of detail, whereas the geographic dimension only allows for the visual focusing (i.e., magnification) of a certain area of the screen (i.e., the area of interest).

Putting this together, a 3-dimensional *navigation space* results. It comprises the following navigation dimensions: semantic (x), geographic (y), and visualization dimension (z) (cf. Figure 3.9).

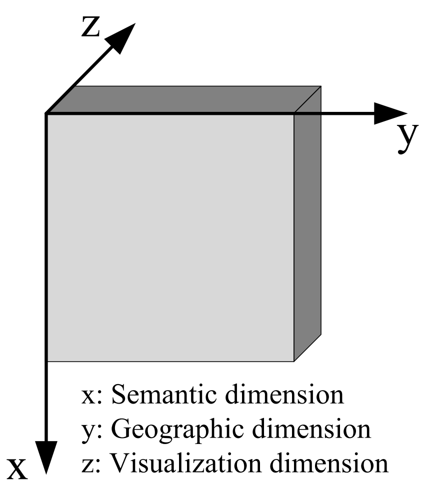


Figure 3.9: The three dimensional navigation space.

In the *semantic dimension*, process model collections may be displayed in different levels of detail (cf. Figure 3.10). On a high semantic level, for example, only the names of the process areas shall be shown. If the semantic level of the respective process area becomes more detailed, additional details (e.g., duration, responsible roles, and contact persons) may be shown as well. Note that similar concepts have been used in the area of zoomable user interfaces as well [RB09].

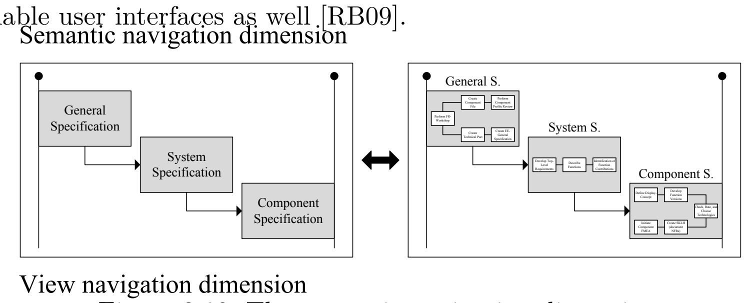


Figure 3.10: The semantic navigation dimension.

The *geographic dimension* allows for a visual zooming without changing the level of detail (cf. Figure 3.11). Think of a magnifier while reading a newspaper. To set different zooming levels, scales can be used. In the area of user interface design, Wijk et al. [vWN03] have already introduced a similar technique.

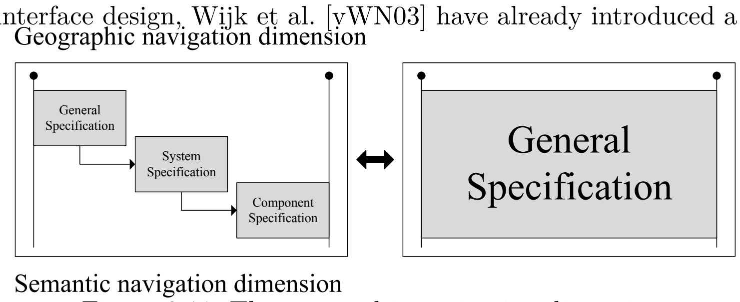


Figure 3.11: The geographic navigation dimension.

The *visualization dimension* enables the user to select different types of process information, such as time aspects, documents, contact persons, or logical relationships with other information (cf Figure 3.12). As opposed to the semantic dimension, information displayed remains on a constant level of detail, i.e., only the point of view is changed. In Figure 3.5, three dimensions have already been introduced. The time-based visualization (cf. Figure 3.5a) emphasises time aspects and uses a time line. The logic-based visualization accentuates logic relations between process steps (cf. Figure 3.5c). Finally, the text-based visualization represents task descriptions (cf. Figure 3.5d). The visualization of process models has been discussed in detail by different authors [KKR12, BRB07, LKR13, KFKF12].

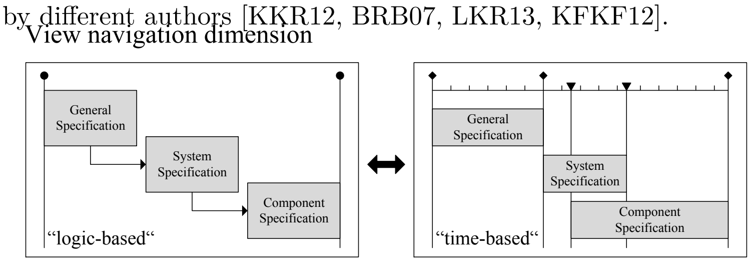


Figure 3.12: The visualization navigation dimension.

Previous research has only considered a single navigation dimension or the combination of two of them (cf. Chapter 8). ProNaVis, in turn, provides three independent navigation dimensions, enabling a user-driven navigation in complex process model collections. Thus, the ProNaVis framework represents a new generation of process repository support.

3.4 Applying the Approach

This section describes how ProNaVis may be applied. Table 3.2 shows the values of the three navigation dimensions used in the example. User interaction is enabled by providing separate adjustment possibilities for each navigation dimension. For this reason, Figure 3.13 depicts a schematic *navigation element*, i.e., a user interface element, providing user interaction elements. In particular, a slider is shown to change the geographic dimension (G). Different semantic levels in the semantic dimension (S) may be chosen using check boxes. Finally, radio buttons may be used to switch between different visualizations (V).

Geographic Dimension	Semantic Dimension	Visualization Dimension
1	Process Model Collection	Time-based Visualization
2	Process Area	Logic-based Visualization
3	Process Models	Text-based Visualization
4	Process Task	

Table 3.2: The used dimensions in our example.

Process navigation starts with a representation of the entire process model collection (cf. Figure 3.13a) in navigation state 1. In this state, the process models are visualized as grey boxes. The geographic level corresponds to level 1, i.e., all process models of the process model collection are shown. The semantic dimension provides process models as abstract grey boxes (semantic dimension level 3). In particular, the visualization is a time-based visualization, i.e., process model durations are represented through the lengths of each box.

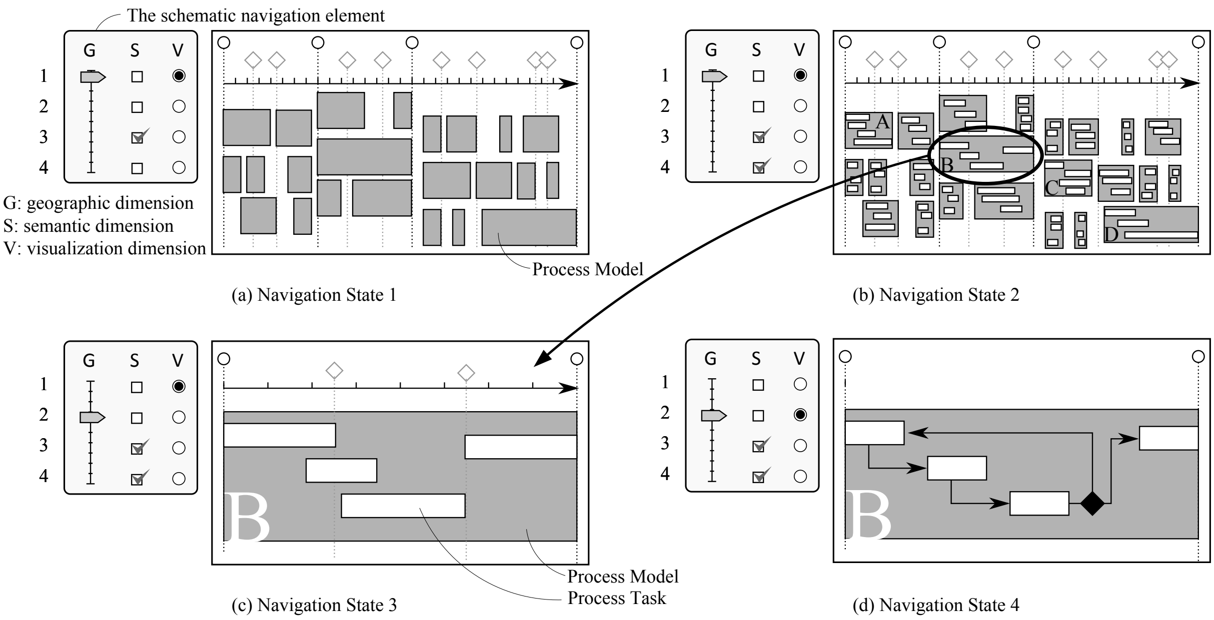


Figure 3.13: Example of process navigation in three navigation dimensions.

Assuming that a requirements engineer is solely interested in the current process task, he may select semantic level 4 to visualize all included process tasks. This interaction results in navigation state 2 (cf. Figure 3.13b), which displays all process tasks (semantic level 4) in combination with the associated process models (semantic level 3). As the engineer is interested in a specific process task within process

B, he applies the geographic dimension to process B reaching navigation state 3 (cf. Figure 3.13c). Note that all interactions are user-driven, i.e., triggered based on user interaction with the navigation element.

Finally, assume that the requirements engineer is less interested in time aspects, but more in what he has to do next when finishing the current process task. Therefore, he switches to the logic-based visualization in navigation state 4 (as depicted in Figure 3.13d). Using this visualization, he can quickly identify predecessor and successor relations of all involved process tasks.

The presented example sketches the ProNaVis navigation concept and how process participants may navigate in complex process model collections based on a flexible process navigation concept that makes use of three independent navigation dimensions.

3.5 Summary

This chapter presented basic ideas of the ProNaVis framework. We first discussed basic issues and provided a common understanding of process navigation and visualization. An illustrating example showed how a process portal from the automotive domain is currently used to enable a 1-dimensional process navigation in process model collections. Then, we investigated the navigation approach from Google Maps supporting two independent navigation dimensions (i.e., zooming dimension and visualization dimension). These concepts, however, are still unable to cope with all use cases presented in Chapter 1. Main reason is the zooming dimension (i.e., the hard-wired semantic and geographic dimension). It allows changing the area of interest by zooming on a certain area on the map. However, the level of detail is automatically adjusted depending on the focused area. In particular, this dimension does not allow displaying detailed information on an abstract zooming level (i.e., area of interest). Based on these observations, the zooming dimension is not sufficient for a flexible navigation support in complex process model collections. Therefore, ProNaVis divides the *zooming dimension* into a *semantic* and a *geographic dimension*. In combination with the *visualization dimension*, these three navigation dimensions form the 3-dimensional *navigation space* as the core of the ProNaVis framework.

The following chapters introduces the ProNaVis navigation concepts in a detailed manner, including the technical realization, formalizations and further use cases.

4 The Navigation Space

The three navigation dimensions introduced in Chapter 3 represent the *navigation space*. This chapter[1] introduces the major steps to construct the latter for a given process model collection. The chapter is structured as follows. Section 4.1 sketches main challenges for constructing the navigation space. Section 4.2 then briefly summarizes the two major steps required to construct the navigation space. Then, Sections 4.3 and 4.4 describe these steps in detail. Finally, Section 4.5 summarizes the chapter.

4.1 Motivation

To support process participants in accessing process model collections in a flexible manner, it becomes necessary to integrate all process models of the model collection into our 3-dimensional navigation space [HMMR14].

This integration, however, is far from being a trivial task. In practice, process models are typically documented inconsistently across different departments. While process management technology used in some departments, for example, can represent process models as XML files, other departments document their process models with PowerPoint and pdf files.

However, to construct the navigation space, all process models of a collection need to be available in a homogeneous, machine-readable form. This is the prerequisite to create a logical representation of the collection's process models and to combine and transfer them to a navigation space. Thus, in order realize a 3-dimensional navigation space, the following challenges need to be addressed:

- Process models must be extracted from heterogeneous sources and must be transferred to a homogeneous, machine-readable representation.
- Process models must be integrated to enable cross-model navigation.
- Process models must be transformed into an integrated hierarchical structure serving as the basis to derive the navigation space, i.e., the three navigation dimensions.

Note that this work focuses on integrating process models which are already available as BPMN XML files. For further information on integrating process models from other sources, please refer to [MUG+14, MMR12a, MMR12b].

4.2 Main Construction Steps

The navigation space is constructed in two consecutive steps taking a process model collection as input:

[1] The chapter is based on the following referred paper [HMMR14]:
Markus Hipp, Bernd Michelberger, Bela Mutschler, and Manfred Reichert. *Navigating in Process Model Repositories and Enterprise Process Information.* in: Proc 8th Int'l Conf on Research Challenges in Information Science (RCIS'14), IEEE, pp. 1–12, 2014

Step 1 (Process Space). First, the *process space* is constructed (cf. Figure 4.1). It represents a harmonized, but preliminary data structure that can be used to construct the actual navigation space. To derive the process space from a process model collection, first of all, we represent each model of the collection as a hierarchical structure called *process object model (POM)* (cf. Step 1.1 in Figure 4.1). Then, we organize and categorize POMs (i.e., process models) according to their topical similarity. In this context, we apply the idea of process areas (cf. Chapter 1). As example consider process models documenting specification tasks (e.g., writing a general specification, system specification, and component specification). The respective process models might then be subsumed under process area *requirements engineering* (cf. Step 1.2). Note that process areas might be further combined to more abstract process areas (e.g., *planning* or *development*). Finally, all POMs are organized under one root process area (e.g., *product development*).

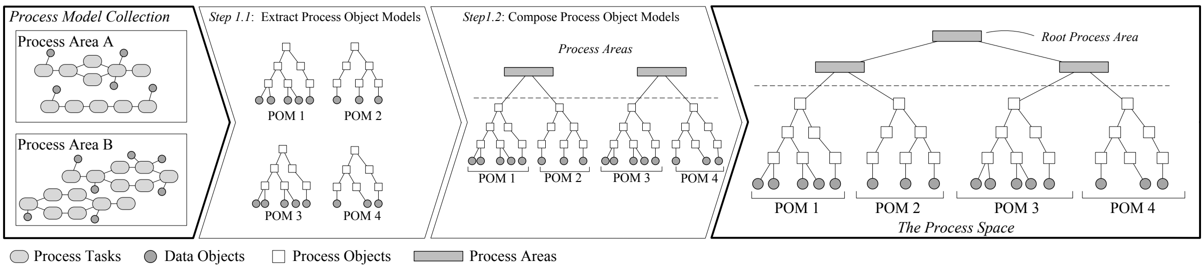


Figure 4.1: Constructing the process space (Step 1).

In general, we differentiate between three kinds of elements in a process space:

- *Process Areas*: Process areas represent logical elements. In turn, a process area comprises other process areas and process models (POMs), respectively.
- *Process Objects*: Process objects represent process model elements such as *pools*, *swimlanes*, *events*, *gateways*, and *tasks*.
- *Data Objects*: Data objects represent documents related to single process tasks. Examples include checklists, guidelines, and best practices.

The construction of the process space is described in detail in Section 4.3.

Step 2 (Navigation Space). Taking the process space derived in Step 1 as input, the *navigation space* is constructed in Step 2 (cf. Figure 4.2). In particular, the three navigation dimensions are created. First, the *semantic dimension* is constructed based on the process space. Thereby, all objects (i.e., process areas, process objects, and data objects) belonging to the same hierarchical level (also denoted as *detail level*) constitute one *navigation state* (cf. Step 2.1 in Figure 4.2) along the semantic dimension. Second, the *geographic dimension* extends the semantic one by adding zooming functionality (cf. Step 2.2). Third, the *visualization dimension* allows displaying a navigation state in different ways (cf. Step 2.3). By combining the three navigation dimensions, we obtain the navigation space.

Section 4.4 presents details regarding the construction of the navigation space.

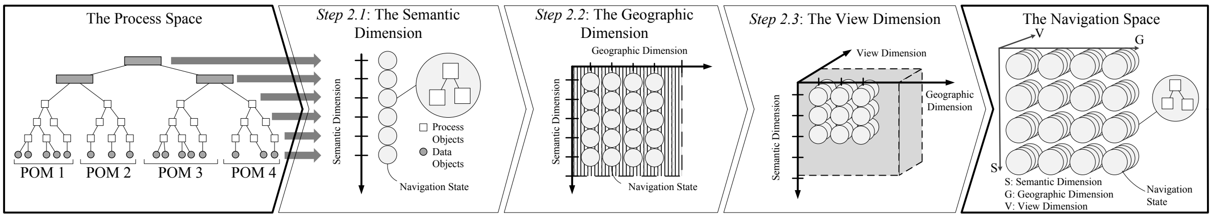


Figure 4.2: Constructing the navigation space (Step 2).

4.3 Step 1: Constructing the Process Space

The process space constitutes a harmonized data structure that provides the basis for deriving the navigation space. The construction of the process space comprises two steps: *Extracting process objects models* (Step 1.1) and *Combining process models* (Step 1.2).

4.3.1 Step 1.1: Extracting Process Object Models

First of all, each process model of a process model collection is represented by a *process object model (POM)*. Note that this will later enable us to construct the semantic dimension.

Different approaches can be applied to transform process models into POMs [SKGM12, Shn91, MY12]. Since, none of them (explicitly) fits our requirements to directly derive the semantic dimension, we introduce a more appropriate approach in the following. Thereby, we assume that process models are available as extensible markup language (XML) representations following the XML Schema Definition for BPMN 2.0 as provided by the Object Management Group[2] (cf. Figure 4.3a).

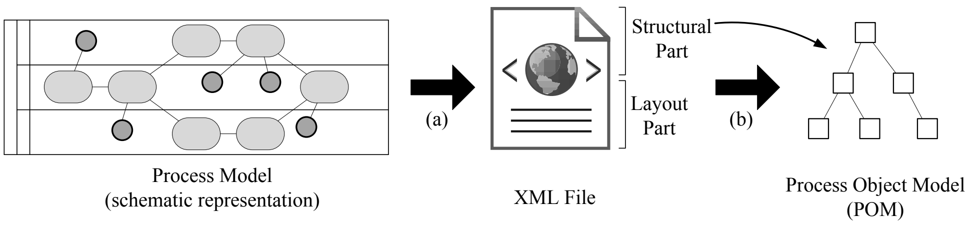


Figure 4.3: Extracting a POM from a process model.

Each process model is represented by one XML file. In turn, each of these XML files is divided into two basic parts, i.e., a structural and a layout part. The *structural* part describes the structure of the process model, i.e., single process objects and their relations. In the *layout part*, in turn, the layout of the process model is defined, i.e., the width and height of process elements or their positions (in terms of x and y coordinates). Regarding the derivation of the POM of a process model, only the structural part is of interest (cf. Figure 4.3b).

For the sake of readability, we illustrate the derivation of a POM along an example, i.e., the process model of the *general specification* process introduced in Chapter 1 (cf. Figure 4.4). The process model

[2]http://www.omg.org/spec/BPMN/2.0/

has been created with Signavio Process Editor[3] and exported as BPMN 2.0 XML file.[4] In particular, all information needed for constructing the POM can be derived from this file.

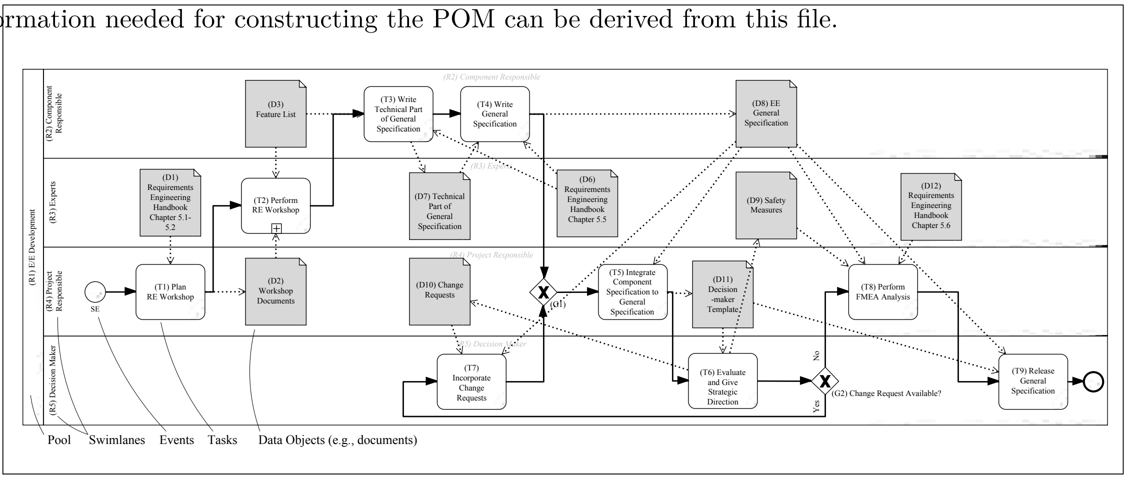


Figure 4.4: The process model.

In the following, we illustrate how the POM is created based on the given input.

1. Root Node

A POM is a tree structure [Shn96]. Consequently, each POM has a *root node*, which is automatically created when exporting the Signavio process model. Logically, the root node corresponds to the tags *<definitions>* and *<collaboration>* in the input XML file (cf. Figure 4.5). Further, it provides information regarding the namespaces used (e.g., *xmlns:bpmndi* and *xmlns:omgdc*) as well as a unique identifier *id* of the process model; *id* is also used as identifier for the POM, allowing us to differentiate between different POMs. Together the two tags include all information needed to derive the root node of a POM (cf. Figure 4.5). Specifically, the *root node* is labeled as “Root” in the POM.

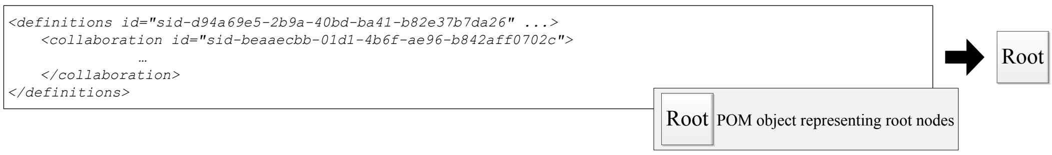


Figure 4.5: The root node.

2. Pools

A Pool is the graphical representation of a participant in a process collaboration (e.g., an organization). Pools are represented in the XML file by means of the *<participant>* and *<process>* tags (according to the BPMN 2.0 standard[5]). Both tags have an *id* and a *name* attribute. Thereby, tag *<participant>* is a true child node of *<collaboration>*. It further provides a reference (*processRef*) to the respective *<process>* (cf. Figure 4.6). In a POM, we label pools with “P”.

[3]Signavio Process Editor: http://www.signavio.com/
[4]The corresponding file can be find in Figure A.1.
[5]Object Management Group (OMG) BPMN 2.0 definition: http://www.omg.org/spec/BPMN/2.0/

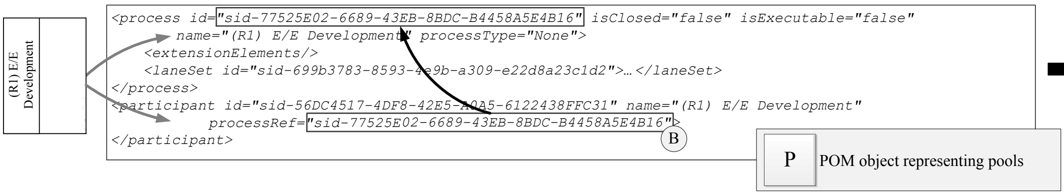


Figure 4.6: Pools.

3. Swimlanes

A swimlane presents the tasks of a particular user role; swimlanes may be nested within a pool. In the XML file (and therewith in POMs) swimlanes are represented by *<lane>* tags. Each lane has an *id* and a *name*. All swimlanes are aggregated within the *laneSet* tag (cf. Figure 4.7). In turn, the *<laneSet>* tag is a child tag of *<process>*. In a POM, lanes are labeled with “L” .

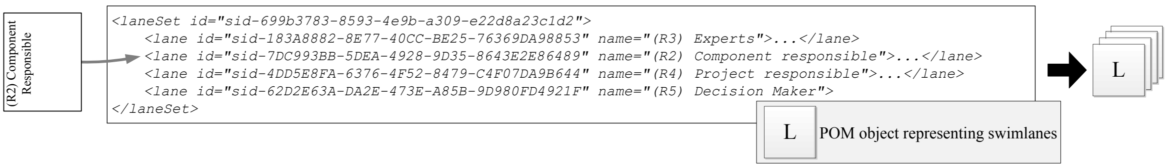


Figure 4.7: Swimlanes.

4. Events, Gateways, Tasks

Events are used to trigger certain tasks within a process model. In turn, tasks represents a single unit of work that has to be processed by the user. Gateways may be used for forking and merging the sequence flow (i.e., the logical sequence of process tasks). Events, gateways, and tasks are represented by self-explaining tags. Examples include *<StartEvent>*, *<EndEvent>*, *<exclusiveGateway>*, and *<task>* (cf. Figure 4.8). Events, gateways, and tasks are related to specific swimlanes through references (*flowNodeRef*). In a POM, events are represented as objects labeled with “SE” (start event) or “EE” (end event), tasks with “T”, and gateways with “G”.

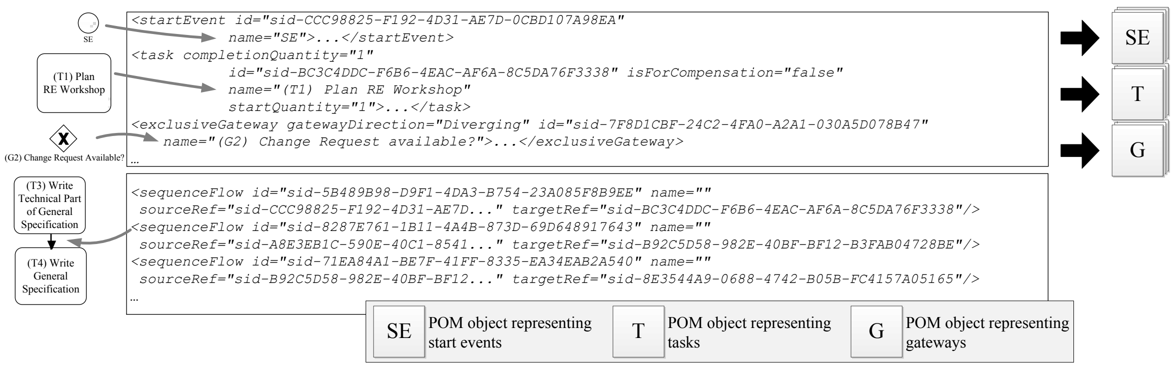


Figure 4.8: Events, Gateways, and Tasks.

A sequence flow is defined by the *<sequenceFlow>* tag. More precisely, each sequence flow between two elements is defined by one tag including references to the source (*sourceRef*) as well as target (*targetRef*) object (cf. Figure 4.4).

5. Data Objects

Data objects provide the user with data required to execute a process task (e.g., documents). Data objects are represented by *<dataObject>* tags in the XML file (cf. Figure 4.9). Furthermore, they can be related to multiple process tasks. Respective relations between data objects and tasks are expressed as directed data flows. Figure 4.9 shows an example of a data flow between process task *T6* and data object *D10*. Each task may have data inputs and outputs. Regarding the presented example *T6* only has *<dataOutputAssociation>* tags since it has no data inputs). In a POM, data objects are represented as grey circles labeled with "D".

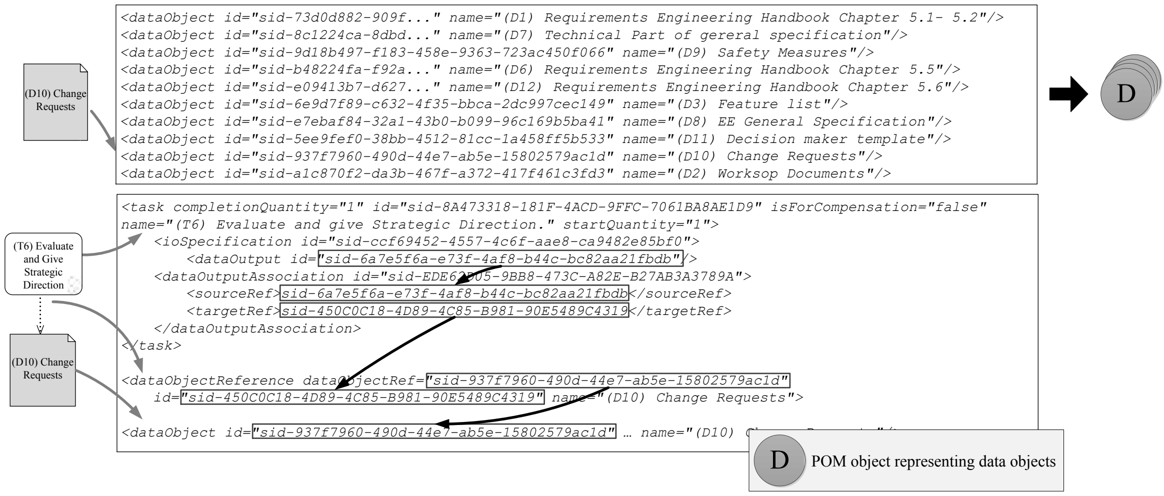


Figure 4.9: Data objects.

Process Models

Figure 4.10 illustrates how a POM can be derived from a XML file.

The root node constitutes the most abstract process object (*Root*). We define this level of detail as 0. Pools (P) represent more detailed information and are therefore assigned to detail level 1. In turn, swimlanes (L) may be nested within a pool; hence they are assigned to detail level 2. Events, gateways and tasks are contained in swimlanes and are assigned to detail level 3. Finally, data objects (D) are considered as the most detailed objects. Consequently, they are assigned to detail level 4.

Consider the *general specification* process from the running example (cf. Figure 4.4). First, we identify the root node that corresponds to detail level 0. Following the structure of the POM, pool $P1$ (i.e., *E/E development*) is related to the root node. Thus, we assign $P1$ to detail level 1. In turn, the swimlanes contained in $P1$, i.e., $L2$ (*component responsible*), $L3$ (*expert*), $L4$ (*project responsible*), and $L5$ (*decision maker*) are assigned to detail level 2. Furthermore, all process events, gateways, and tasks are assigned to detail level 3. Finally, data objects $D1 - D3$ and $D6 - D12$ are assigned to detail level 4 (cf. Fig. 4.11).

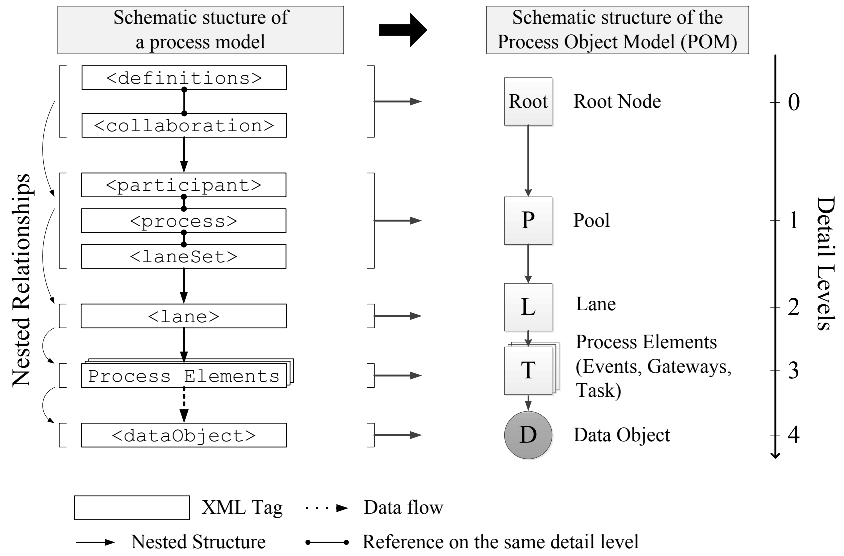


Figure 4.10: Deriving a POM from a process model (represented as XML).

Using the POM, navigation on different levels of detail becomes possible. For example, one may gather information from a particular level of detail, while hiding the objects from other levels of details. In this context, a user may start with the root node and then navigate to information on a more detailed level. For example, he may navigate to the level of swimlanes (i.e., detail level 2) to display the used roles involved in the process. For instance, if a manager wants to know whether a requirements engineer is needed in this process model, it will be sufficient to take a look at detail level 2, i.e., swimlanes.

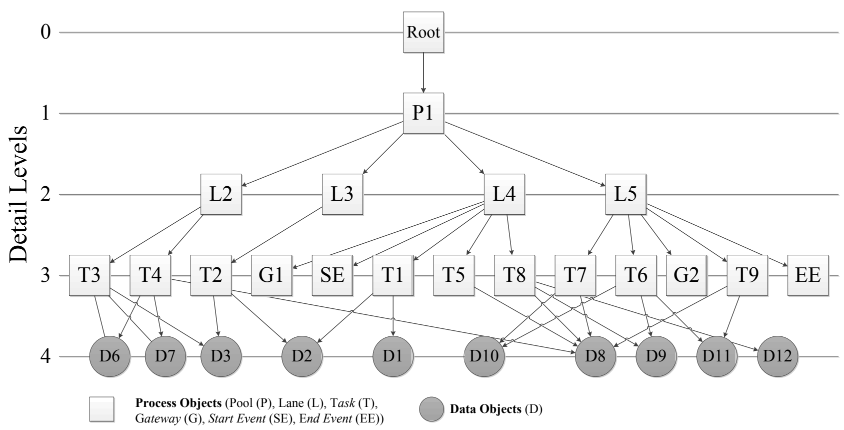


Figure 4.11: The POM related to the running example.

Note that a POM allows navigating within a single process model. The presented POMs, therefore, already provide a flexible way to navigate within a single process model and to interact with it. However, to also enable navigation across process models in a given process model collection, POMs related to the one and same collection need to be combined as well.

4.3.2 Step 1.2: Combining Process Models

In order to allow for the navigation within an entire process model collection, we pick up the idea of process areas as outlined in Chapter 1. Thereby, a process area combine several POMs related to the same topics. More precisely, topics are represented by manually created *process areas*. In turn, each process area is assigned to detail level -1. As another means of abstraction, multiple process areas may be combined to an aggregated process area. In turn, the latter is then assigned to a further decreasing detail level (cf. Figure 4.12).

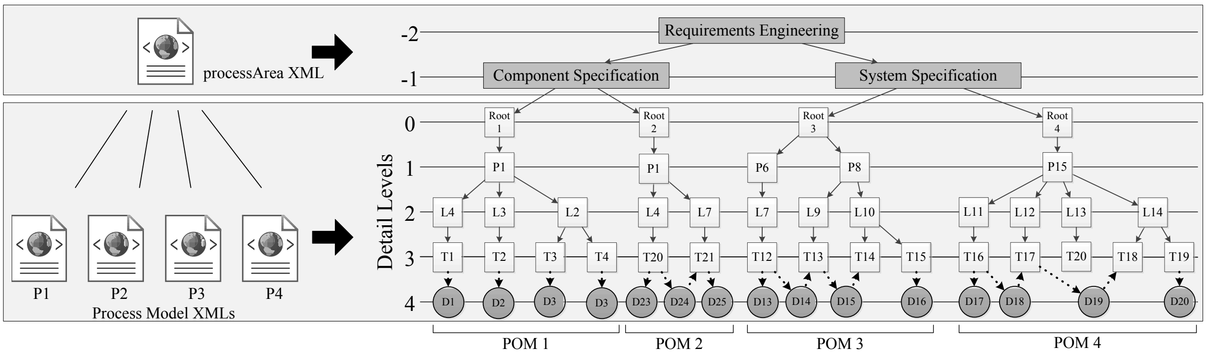


Figure 4.12: The process space (Step 1.2).

For illustrating our approach, we use a schematic representation of a process space comprising four POMs (i.e., POM 1 - POM 4 in Figure 4.12). Remember that each POM represents one process model. In the example, two relevant topics are identified (cf. Figure 4.12). Both are represented as process areas and are assigned to detail level -1. More precisely, process models POM 1 and POM 2 are assigned to process area *component specification*, whereas process models POM 3 and POM 4 are assigned to process area *system specification*. Finally, both process areas are connected through an additional process area on detail level -2, which represents the root process area of the process model collection. Starting with this abstract process area on detail level -2, a user may navigate to all four process models of the collection.

The identification of topical similarities is a difficult task to accomplish, which cannot be fully automated. Accordingly, the definition of process areas as well as the assignment of POMs to them has not yet been automated. Instead, we use an XML file for manually defining this assignment (cf. *processArea.xml* in Figure 4.12).

```
<processArea name="Requirements Engineering" id="1111">
    <processArea name="Component Specification" id="2222" parentRef="1111">
      <root name="Manage Workshop" src="C:/processes/workshop.bpmn"\>
      <root name="Prepare Specification" src="C:/processes/prepareTemp.bpmn"\>
    </processArea>
    <processArea name="System Specification" id="3333" parentRef="123456">
      <root name="General Specification" src="C:/processes/general.bpmn"\>
      <root name="System Specification" src="C:/processes/system.bpmn"\>
    </processArea>
    <processArea name="Implementation" id="3333" parentRef="1111"></processArea>
</processArea>
```

Listing 4.1: Definition of process areas.

As an example consider Listing 4.1. It shows the *processArea.xml* file representing the process space from Figure 4.12. Each process area is defined by a *<processArea>* tag, comprising attributes *<id>*, *<name>*, and *<parentRef>*. The latter allows referring to the parent process area. In turn, connections between process areas and single POMs can be established by using the *<root>* tag.

Using a separate XML file to define process areas and their assignment to POMs reveals two advantages. First, process areas can be easily maintained, e.g., new process areas may be introduced at a later stage. Second, when changing process models (e.g., replacing an old process model by a new one), only the reference to the respective process area needs to be updated. Adding a process model to the given process model collection can be easily accomplished as well. In this case, the new process model must be assigned to a given process area by inserting a *<root>* tag.

Alltogether, by associating POMs with process areas, an integrated *process space* results. In turn, the latter allows for the flexible navigation within the entire process model collection.

4.3.3 Concluding Remarks

So far, we have shown how a process model collection can be transformed into a process space. In particular, we described how process models can be represented as POMs. Furthermore, we showed how process areas can be utilized to combine POMs. Finally, process areas are defined in a separate XML structure, which allows defining process areas as well as their relations to POMs.

The following section describes the construction of the navigation space; i.e., it shows how the three navigation dimensions can be derived based on a given process space. In particular, we pick up the detail levels of the process space to construct the semantic dimension first. Based on the resulting structure, we then introduce the geographic and visualization dimensions.

4.4 Step 2: Constructing the Navigation Space

Taking a process space (cf. Section 4.3) as input, the navigation space can now be derived by consecutively constructing the three navigation dimensions.

4.4.1 Step 2.1: The Semantic Dimension

The semantic dimension has been originally introduced as *semantic zooming* in the area of zoomable user interface (ZUI) [RB09]; a detailed discussion can be found in Chapter 8. Semantic zooming is defined as *"a more sophisticated concept, in which objects change their appearance as the amount of screen real estate available to them changes."* [RB09]. We adopt this definition to process model collection. Note that the latter may change its appearance based on the varying objects on the different levels of detail of the respective process space.

As described in Section 4.3, all process areas, process objects, and data objects from the process space are assigned to a particular detail level. To derive the semantic dimension, we pick up objects from the same detail level and assign them to a so called *navigation state* NS(s), where s corresponds to the semantic detail level. Figure 4.13 illustrates how navigation states can be derived along the semantic dimension based on the process space we constructed in Section 4.3. In this context, navigation states NS(2) and NS(5) are presented in more detail (cf. Figure 4.13B). More precisely, NS(2) comprises all

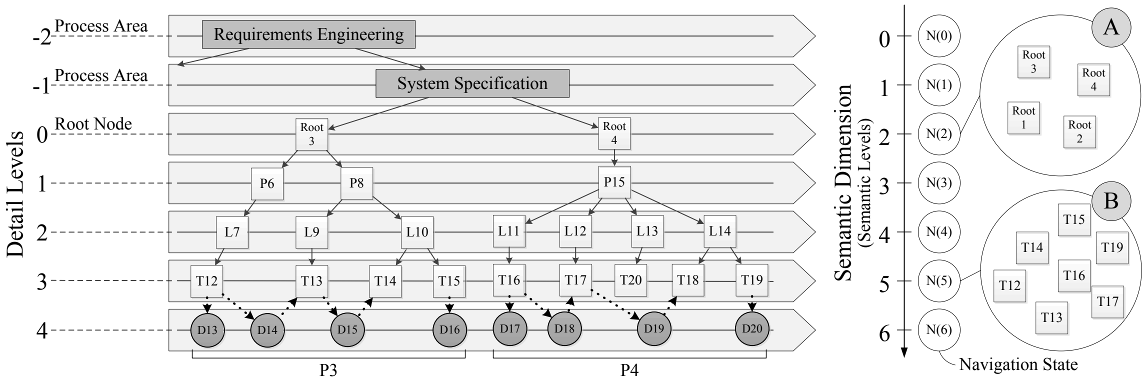


Figure 4.13: Deriving the semantic dimension.

process model root nodes (cf. Figure 4.13A), whereas NS(5) comprises all events, tasks, and gateways (cf. Figure 4.13B). As introduced in Chapter 3, a navigation state defines the current "position" of a process participant within a process model collection, i.e., a navigation state comprises a well-defined set of objects from the process space.

Semantic zooming becomes possible by changing the desired semantic level, i.e., by traversing different navigation states. However, along the semantic dimension different navigation states may not always be sufficient to support process participants, as they might include too many objects at once. As example consider a business unit manager (cf. Use Case 2 from Chapter 1) who is mainly interested in process tasks related to a specific process model. However, navigating to navigation state NS(5) along the semantic dimension reveals all process elements (i.e., events, gateways, and tasks). In particular, this navigation state does not only include the needed process tasks of the considered process model, but the ones of the entire process model collection as well. Therefore, users should be able to focus on a particular set of objects in order to tailor the information needed. For example, the business unit manager should be able to zoom on process tasks assigned to a particular process model. To enable this, the geographic dimension is introduced in the following section.

4.4.2 Step 2.2: The Geographic Dimension

As discussed in Section 4.4.1, navigation states might comprise objects not relevant for a particular process participant. For example, navigation state NS(5) (cf. Figure 4.13) comprises all process tasks of the process model collection. However, if a process participant is only interested in the process tasks of a particular process model, NS(5) does not constitute a proper state. Instead, a specific focus on required objects within a navigation state should be enabled. From the user's point of view, this can be achieved by zooming into a specific part of the process model collection. In literature, for example, zooming concepts have been investigated by van Wijk et al. [vWN03, vWN04], who considered zooming and panning concepts in information spaces (cf. Chapter 8 for further detail). In the following, we apply corresponding concepts to construct the geographic dimension.

Based on the navigation states obtained in the context of the semantic dimension, the geographic dimension enables geographic zooming. Thereby, geographic zooming logically corresponds to the selection of subtrees in the process space, i.e., to focus on a specific part of the process space. In turn, a subtree is defined by its corresponding root node, also denoted as *reference object*. Furthermore, the level of the geographic dimension is defined by the detail level of this reference object (cf. Figure 4.14). Navigation states, both the semantic and the geographic dimension into account, can be defined as tuples NS(s, g), where s represents the semantic level of the desired objects (i.e., semantic dimension) and g the level of detail of the reference object (geographic dimension). For example, navigation state NS(5,2) includes all process elements (semantic level 2) from process model P3, i.e., the subtree defined by reference object *Root 3* (geographic level 2).

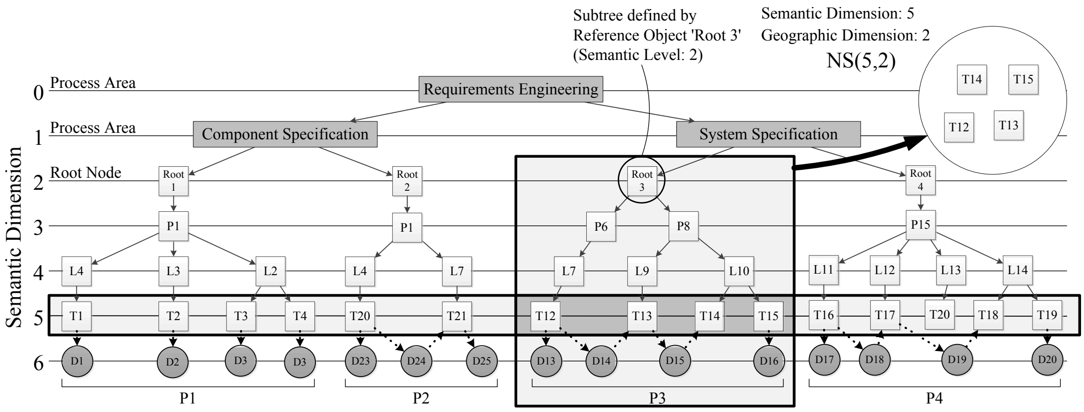


Figure 4.14: Deriving a navigation state from a process space.

In general, the geographic dimension extends the semantic one to a 2-dimensional navigation space (cf. Figure 4.15). This implies that the navigation states derived from the semantic dimension (cf. Section 4.4.1) refer to the entire process space on geographic level 0. In this case, the considered subtree corresponds to the process space itself, i.e., the reference object is the root of the process space. Consequently, NS$(5, 0)$ corresponds to the former navigation state NS(5) presented in Section 4.4.1.

Changing the reference object now corresponds to navigating along the geographic dimension. For example, navigation state NS$(5, 4)$ includes only process elements belonging to the same swimlane, e.g., assigned to a subtree defined by the reference object $L10$ on geographic level 4 (cf. Figure 4.15). From the user's point of view, navigating along the geographic dimension logically corresponds to zooming into the reference object (e.g., swimlane $L10$).

The geographic dimension introduces a concept to define navigation states not only based on the detail level of the semantic dimension, but on subtrees of the process space as well. To illustrate how a user might navigate along the geographic dimension, Figure 4.16 presents a navigation scenario. Assume that a process participant is interested in specific process tasks, i.e., process elements on detail level 5. Therefore, the semantic dimension is set to level 5. Navigation along the geographic dimension, in turn, starts on an abstract level as the initial reference object is the root process area. For the given example, let us assume that in Figure 4.16 navigation starts with navigation state NS$(5, 0)$, which includes a set of all process elements (semantic level 5) within the subtree defined by the process area *requirements engineering* (geographic level 0). Figure 4.16 further illustrates how the navigation state might look like on a user screen (i.e., the *image space*[6] [vWN03, vWN04]). Regarding navigation state NS$(5, 0)$, a large number of process elements need to be visualized. Starting from this navigation state, the user might select another reference object along the structure of the process space (as indicated in Figure 4.16). For example, this means that the user might zoom on process area *system specification*. For this purpose,

[6]The image space represents what the user experiences when navigating along the geographic dimension.

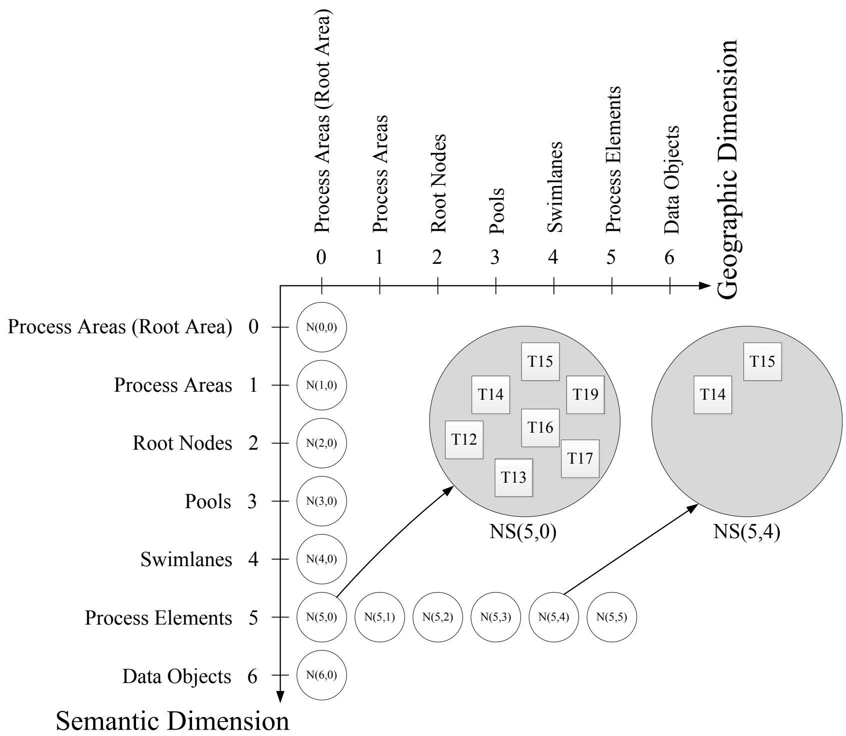


Figure 4.15: Navigation along the geographic dimension.

he has to navigate from navigation state NS(5, 0) to navigation state NS(5, 1) as process area *system specification* is located on geographic level 1. NS(5, 1) then only contains process tasks corresponding to any process model from this process area. The user might further zoom on *Root 3*, which corresponds to a single process model. This zoom can be realized by navigating to navigation state NS(5, 2). From the perspective of the user, the latter zooms into the process space step-by-step. To reach navigation state NS(5, 3), the user navigates to pool $P8$. Therewith, he reaches the desired navigation state NS(5, 4) by choosing lane $L10$ on detail level 4 as reference object. Finally, $NS(5, 4)$ only constitutes $T14$ and $T15$.

The geographic dimension can be followed for every navigation state of the semantic dimension. As the semantic and geographic dimensions may be adjusted independent from each other, each navigation state corresponds to a point within a 2-dimensional navigation space (cf. Figure 4.17). As discussed, this navigation space can be derived from the process space introduced in Section 4.3 when applying both the semantic and the geographic dimension to a given process space, we can tailor the set of objects associated with navigation states.

For users, however, it will be crucial that the selected objects are visualized in a user-friendly manner. To ensure this, we add the *visualization dimension* as the third dimension to our navigation space.

4.4.3 Step 2.3: The Visualization Dimension

The visualization dimension deals with the actual visualization of single navigation states as defined by the semantic and geographic dimension. In particular, this dimension shall allow transforming navigation states together with the objects they comprise, into various representations. Unlike the semantic and

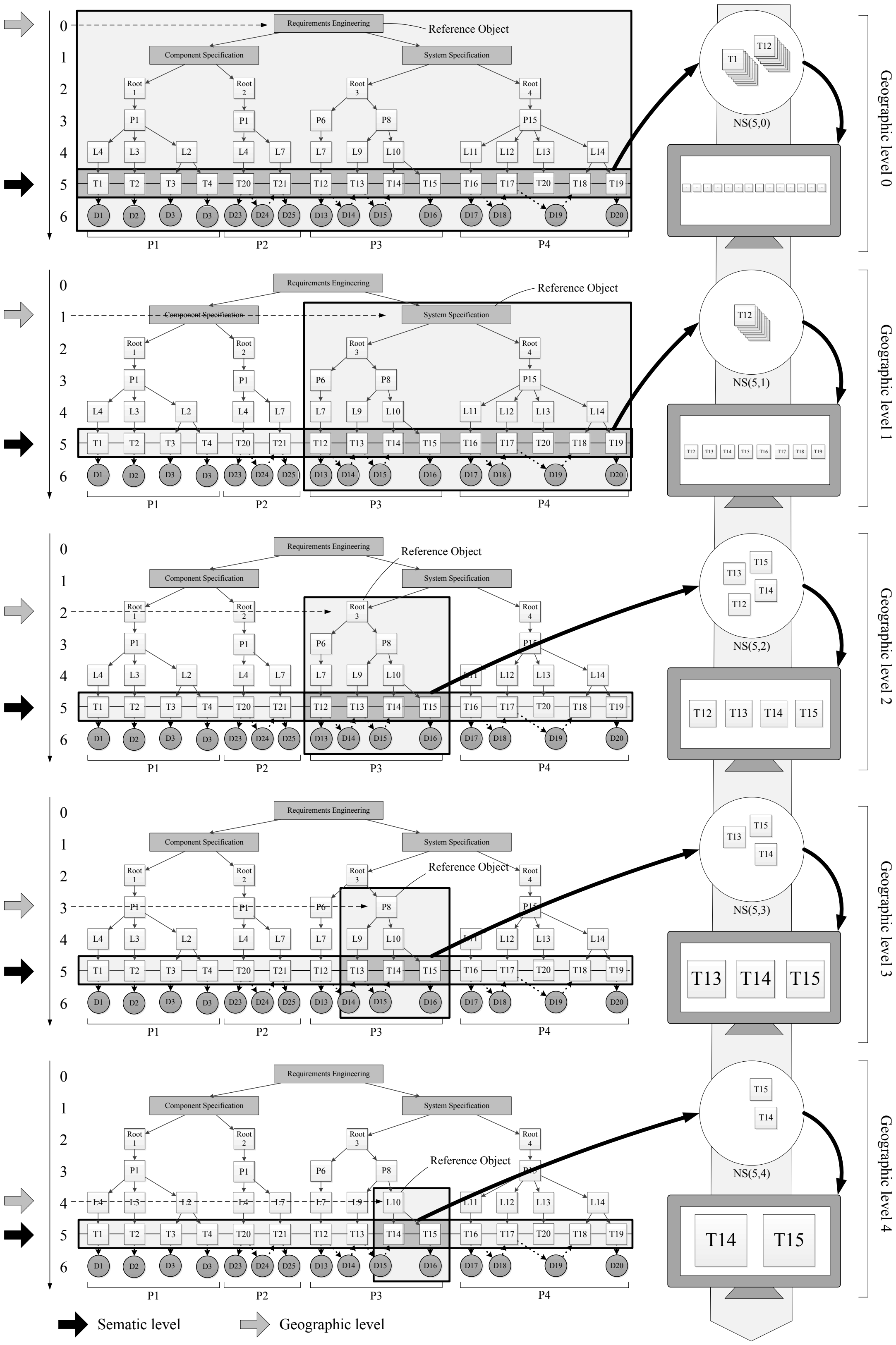


Figure 4.16: Navigation along the geographic dimension.

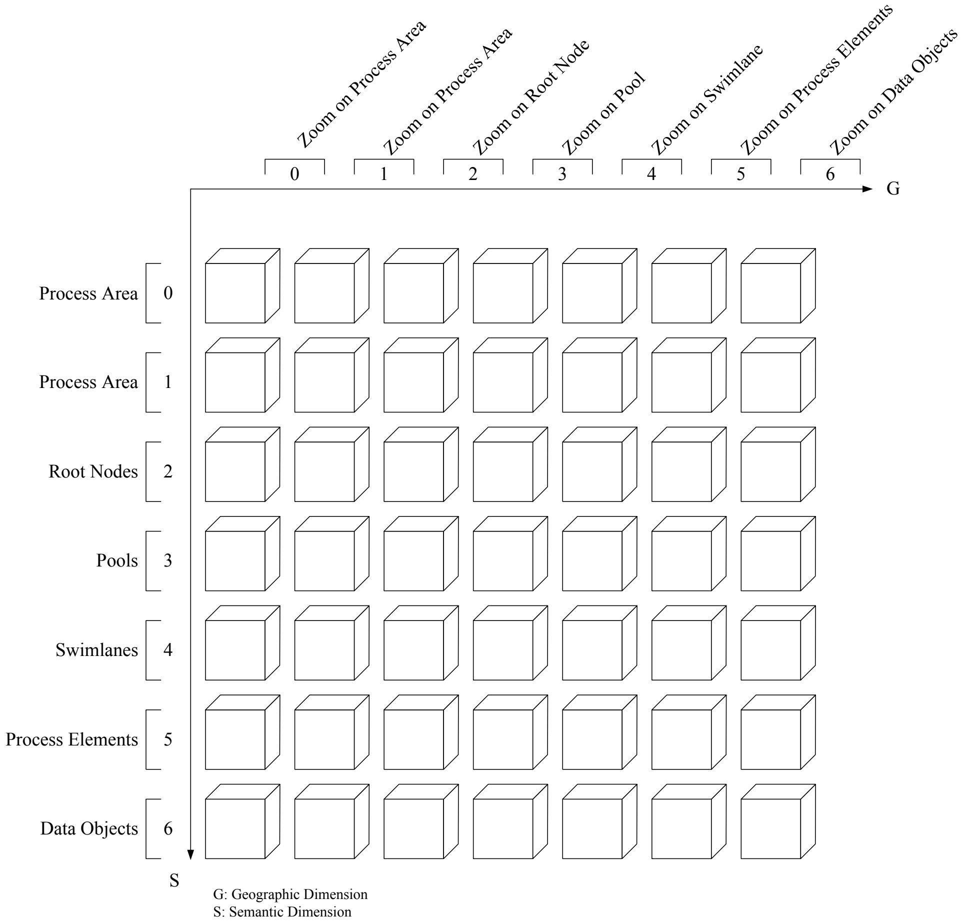


Figure 4.17: 2-dimensional navigation space.

geographic dimensions, however, the visualization dimension cannot be directly derived from the given process model collection.

As example of a basic representation of a navigation state, consider a *logic-based* visualization; i.e., a BPMN-like visualization of a process model. Note that this thesis does not focus on different visualization techniques, which have already been addressed, for example, by Bobrik et al. [BRB07] and Kolb et al. [KR13a, KR13b] (see Chapter 8 for a detailed discussion). Instead, we focus on conceptual visualization approaches of navigation states from the user perspective. These concepts are described in detail in Chapter 6, whereas this section introduces the visualization dimension on an abstract level solely.

To illustrate how the visualization dimension might be integrated with the semantic and geographic dimensions, we refer to three basic types of visualization types already introduced in Chapter 1: *time-based* (1), *logic-based* (2), and *text-based* (3). In general, other types can be applied to the navigation space as well.

The *time-based* visualization is used to visualize temporal aspects. For example, tasks may be represented by rectangles, which then reflect the duration of the respective tasks. In turn, a *logic-based* visualization may be used to emphasize logical relations between tasks, e.g., predecessor and successor relations between them. Finally, a *text-based* visualization might be used in order to provide textual descriptions, e.g., textual process descriptions instead of logic-based process models [KLMR13].

As example consider navigation state NS(5,4) (cf. Section 4.4.2). Events, tasks, and gateways on semantic level 5 are considered. Further, swimlane L14 is used as reference object (geographic level 4). The resulting navigation state, which comprises process tasks T14 and T15, might then be visualized as shown in Figure 4.18. To enable navigation between the different visualization types, navigation state NS(5,4) needs to be replayed by one of the navigation states NS(5,4,0), NS(5,4,1), or NS(5,4,2).

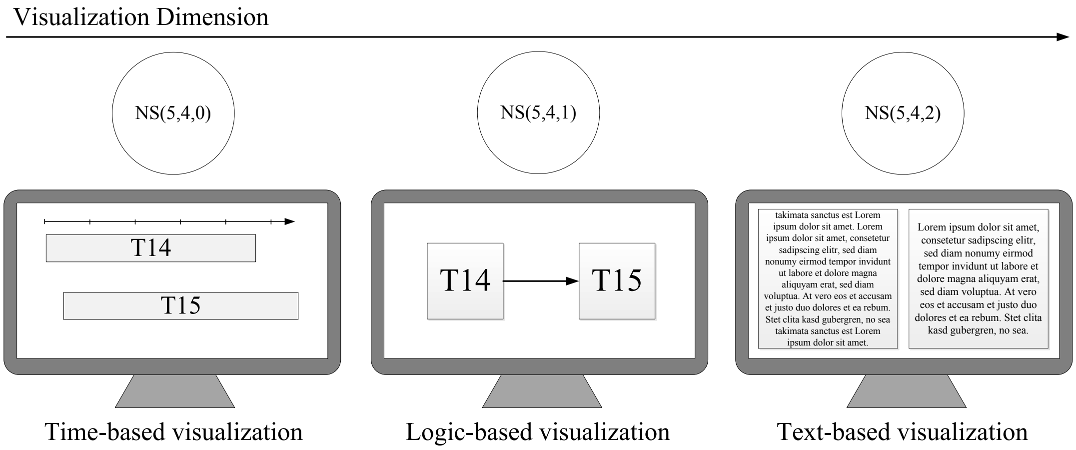


Figure 4.18: The visualization dimension.

Generally, we define the visualization dimension as the third dimension in addition to the semantic and geographic ones. Consequently, navigation states need to be defined as triples NS(s, g, v), where s represents the level of detail along the semantic dimension, g the zooming level along the geographic dimension, and v the applied visualization. The resulting navigation space, including all possible navigation states, is shown in Figure 4.19.

4.4.4 Enhancing Process Navigation

This chapter has introduced the navigation space with its three navigation dimensions. In particular, the navigation space allows navigating between navigation states through state transitions. Thereby, a state transition is triggered by the user manipulating the navigation dimensions. For example, increasing the detail level triggers a state transition along the semantic dimension. Zooming into a part of the process model collection, in turn, triggers a state transition along the geographic dimension. Finally, switching between different visualization types triggers a state transition along the visualization dimension.

In general, however, this kind of navigation is not yet sufficient to cover all relevant use cases. In the following, we introduce two additional concepts supporting advanced navigation within the navigation space.

Filter Mechanisms

In certain scenarios, very detailed information might be required. For example, consider Use Case 5 as described in Chapter 1. A quality manager is involved in multiple process tasks of a process model collection, i.e., he needs to consider different process models to get an overview on all process tasks he is responsible for. Using the presented navigation space, for example, he may navigate to navigation

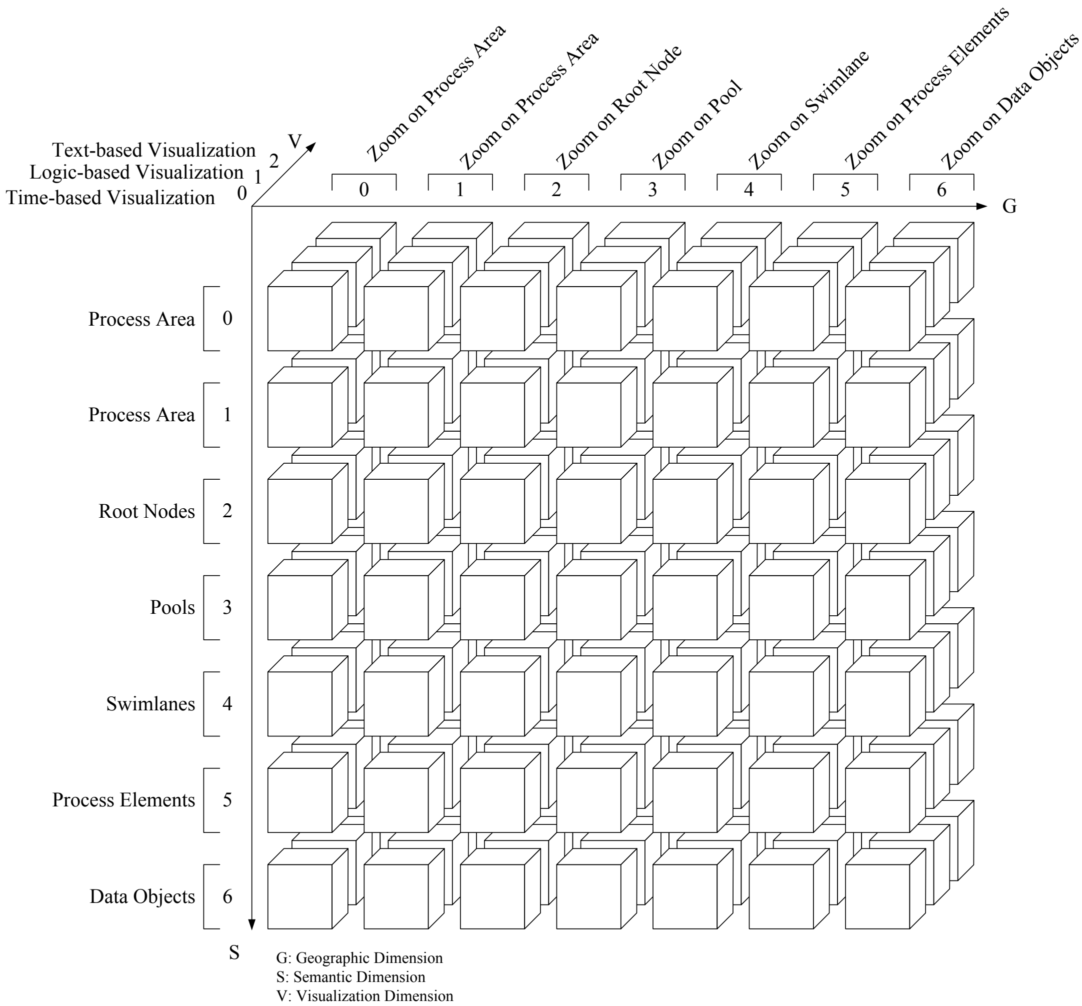


Figure 4.19: The 3-dimensional navigation space.

state NS(5,0,0) if he wants to see his role-specific process tasks within the entire process model collection (geographic level 0) in a time based visualization (visualization type 0). Note that the resulting navigation state already hides unnecessary information such as pools, swimlanes, or data objects as these elements are assigned to navigation states on another semantic level. However, the navigation state still contains process tasks not relevant for the quality manager. In fact, visualization respective navigation states is only useful for a process participant if additional filter criteria may be applied to exclude selected objects of a navigation state from being displayed. In the context of the scenario considered (i.e., Use Case 5), navigation state NS(5,0,0) should be filtered as follows:

$$NS(5,0,0).filter(simlane.name = \text{“}QualityManager'') \tag{4.1}$$

Figure 4.20 shows the result of filtering NS(5,0,0) this way. Only T21 and T12 are displayed based on the applied filter for “Quality Manager"[7].

[7] We further illustrate the application of such filter mechanisms in Chapters 7 and 9.

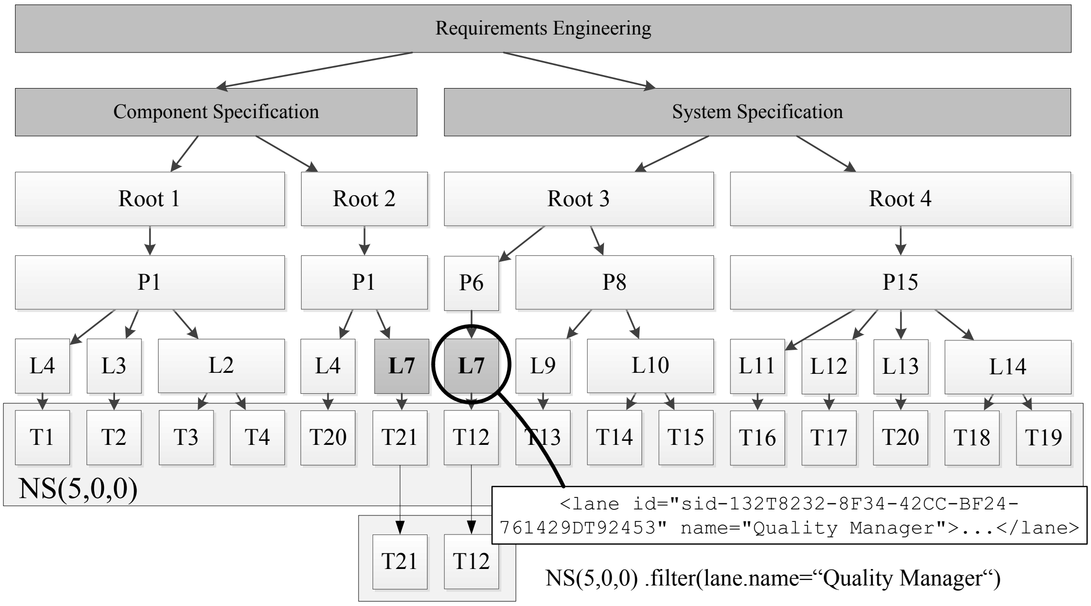


Figure 4.20: Filter mechanism applied to the visualization of a navigation state.

Visualizing Multiple Navigation States

In certain scenarios, visualizing solely one navigation state at once might be difficult to understand for process participants, especially since a navigation state only provides objects on one detail level. Therefore, users might loose orientation when navigation to these navigation states [WLS98]. As example reconsider again the Quality Manager from Use Case 5. Further, assume that he is now interested in tasks he is responsible for and that belong to a particular process area (cf. Figure 4.21A). Therefore, he navigates to navigation state NS(5,1,0). Visualizing this navigation state means to display process tasks from different process models and assigned to different swimlanes. Note that swimlanes are not visible to the user in this navigation state. To increase orientation for this particulat navigation state, swimlanes may be additionally provided, by additionally visualizing the respective navigation state ($NS(4,1,0)$), on a more abstract detail level (cf. Figure 4.21B). The result of combining NS(5,1,0) and NS(4,1,0) can be seen in Figure 4.21C.

$$NS(5,1,0).combine(NS(4,1,0)) \tag{4.2}$$

As another example, consider the visualization of an entire process model. According to the navigation space, a model includes process objects on different detail levels, e.g., pools on level 3, swimlanes on level 4, process elements on level 5, and data objects on level 6. Hence, visualizing an entire model requires the combination of all navigation states on these detail levels. In order to visualize an entire process model, therefore, additional navigation states on more abstract semantic levels must be combined:

$$NS(6,2,0).combine(NS(5,2,0), NS(4,2,0), NS(3,2,0)) \tag{4.3}$$

The application of this concept is discussed in Chapter 7, whereas its implementation is described in Chapter 9.

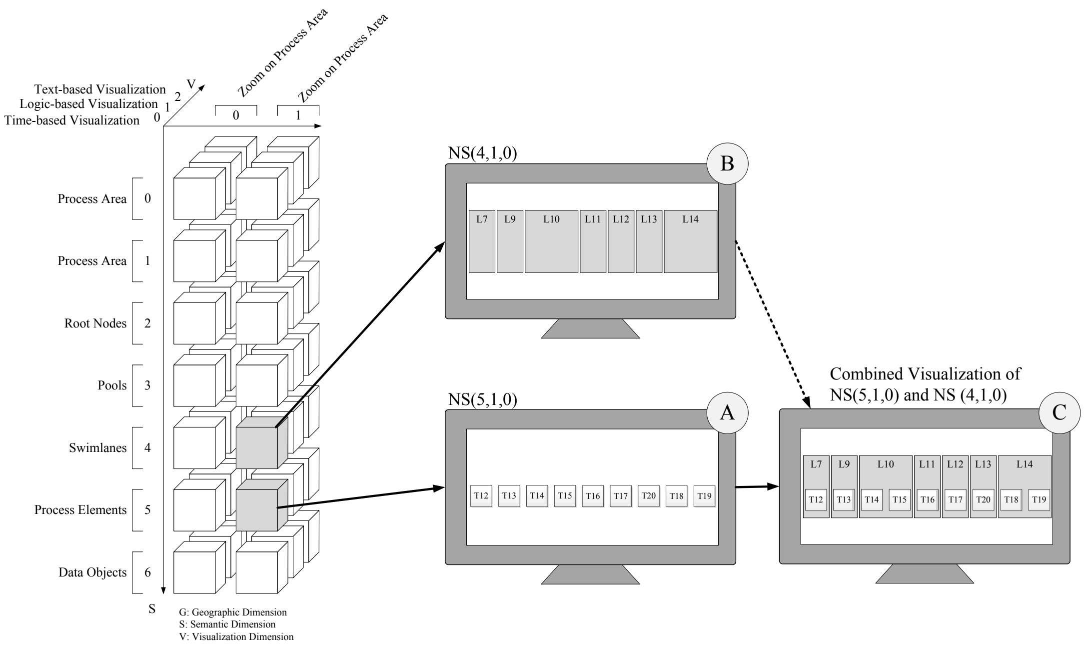


Figure 4.21: Visualization of multiple navigation states along the semantic dimension.

4.4.5 Concluding Remarks

This section presented the navigation space in detail. The latter allows users to navigate along three independent navigation dimensions. First, we introduced the semantic dimension, which assigns objects from the process space to detail levels. In particular, the semantic dimension allows users to navigate within the process space on different levels of detail. The geographic dimension, in turn, allows focusing on specific objects based on reference objects; i.e., it allows decreasing the number of objects to be visualized. From a perspective of a user, this corresponds to zooming on certain parts of the process space. Finally, the visualization dimension deals with the presentation of navigation states to end users.

A process participant may interact with the navigation space using one or more of the three navigation dimensions, i.e., interacting with a navigation dimension triggers a state transition between two navigation states within the navigation space.

Finally, we introduced two other concepts that enable a more effective process navigation and foster comprehensibility of the information displayed. More specifically, filter mechanisms allow decreasing the number of objects for a navigation state and, hence, the number of objects to be displayed on the screen. We also presented an approach that allows visualizing multiple navigation states.

4.5 Summary

This chapter presented an approach to construct the navigation space based on a given process model collection. In particular, we illustrated how the three navigation dimensions can be derived when building the navigation space. Table 4.1 summarizes how these navigation dimensions meet the requirements from Chapter 2.

The next chapter presents a formalization of the navigation space.

Req #	**Requirement**	**Navigation Dimensions**		
		Semantic	Geographic	Visualization
NavReq #1	Depending on a process participant's experience, the level of detail regarding a process task should be adjustable.	●	●	
NavReq #2	Process participants should be able to adjust the level of detail regarding process model collection in order to obtain a quick overview on a specific task that is currently executed.	●	●	
NavReq #3	Users should be enabled to access process tasks in other process areas.		●	
NavReq #4	Relevant process information must be accessible at the level of single process models from the process model collection.		●	●
NavReq #5	Roles must be globally defined in a detailed manner.	●		●
NavReq #6	Process participants must be able to access process models on different levels of detail.	●	●	
VisReq #1	Task descriptions must be documented in a well understandable manner.		●	●
VisReq #2	Temporal and logical dependencies must be considered when visualizing processes.			●
VisReq #3	Complex process information must be visualized in a comprehensible manner.			●
VisReq #4	Information about roles must be intuitively identifiable.			●
VisReq #5	The amount of visualized information should not overload process participants.	●	●	●

● The requirement is met.

Table 4.1: Requirements met by the navigation dimensions.

5 Formalizing the Navigation Space

This chapter[1] provides a formalization of the navigation space introduced in Chapter 4. We use concepts from Linear Algebra for this purpose. As the three navigation dimensions can be adjusted independently, the navigation space corresponds to a 3-dimensional Cartesian system based on three perpendicular axes [DO01]. Consequently, navigation states correspond to single points within this system. In particular, the provided formalization allows reasoning about navigation paths, e.g., on whether a particular path is optimal in a given context or whether it is valid at all.

This chapter is organized as follows: Section 5.1 introduces basic definitions. Section 5.2 presents a running example. Section 5.3 introduces advanced formalizations. Section 5.4 then shows how the formalizations can be applied. Alternative formalization approaches are discussed in Section 5.5. Section 5.6 concludes the chapter with a summary.

5.1 Basic Definitions

This section introduces basic definitions required in the context of the formalizations.

Navigation State (NS). A navigation state corresponds to a specific point within the 3-dimensional navigation space. Thereby, the (discrete) levels of the three navigation dimensions are represented on an absolute scale. For the sake of simplicity, we use natural numbers for this purpose. Hence, in our context, we can define a navigation state as a triple. Let s be the value of the semantic dimension, g be the value of the geographic dimension, and v be the value of the view dimension. Then, a specific navigation state NS can be represented as follows:

$$NS = (s, g, v) \text{ with } s, g, v \in \mathbb{N} \tag{5.1}$$

Note that s, g and v may be manually selected by the user. Accordingly, the set of all potential navigation states NS_{total} is as follows:

$$NS_{total} = \{(g, s, v) | g, s, v \in \mathbb{N}\} \tag{5.2}$$

Some of the navigation states make no sense from a semantic point of view, i.e., they disturb the user (as they are not relevant) or they are forbidden by definition (cf. Section 4.4.4). Reconsider the Google Maps metaphor (cf. Chapter 3) and assume the user wants to see all city names at the same time (semantic dimension) on the entire globe (geographic dimension). In such a navigation state, labels

[1] The chapter is based on the following referred paper [HMR12]:
Markus Hipp, Bela Mutschler, and Manfred Reichert. *Navigating in Complex Business Processes.* in: Proc 23rd Int'l Conf on Database and Expert Systems Applications (DEXA'12), LNCS 7447, pp. 466–480, Springer, 2012

would significantly overlap due to limited screen space. Hence, such a navigation state should be not reachable and be added to the set of forbidden navigation states $NS_{forbidden}$. In turn, we denote the set of allowed navigation states as *basis model* BM.

Basis Model (BM). The basis model corresponds to the set of allowed navigation states within the given navigation space:

$$BM = NS_{total} \backslash NS_{forbidden} \tag{5.3}$$

Process Interaction. Changing the values of the three navigation dimensions in a given navigaiton state results in a state transition within the navigation space. Since respective state transitions are *user-driven*, we denote them as *process interactions*. In our navigation framework, process interactions are represented by vectors. Changing the view from 'logic-based' to 'time-based' constitutes an example of such an interaction.

A *1-dimensional* process interaction constitutes an activity transforming a given navigation state into another one by changing the value of exactly one navigation dimension. In general, a one-dimensional process navigation Int_{oneDim} can be defined as follows:

$$Int_{oneDim} = \{\vec{e} = (\tilde{e_1}, \tilde{e_2}, \tilde{e_3}) | \tilde{e_1}, \tilde{e_2}, \tilde{e_3} \in \{0, 1, -1\} \text{ and } \|\vec{e}\| = 1\} \tag{5.4}$$

In turn, a *multi-dimensional* process interaction can be defined as an interaction transforming a navigation state into another one by changing the value of multiple navigation dimensions at the same time (e.g., both the geographic and the semantic dimension may be changed at once). Google Maps, for example, implicitly uses multi-dimensional interactions when the user applies the scroll wheel to zoom (see the *zooming dimension* described in Section 3.3). If the geographic dimension is changed, the semantic one will be changed accordingly. Since such behavior is well known and accepted by users, we apply it to process navigation as well. We define multi-dimensional process interaction as follows:

$$Int_{multiDim} = \{(\tilde{e_1}, \tilde{e_2}, \tilde{e_3}) | \tilde{e_1}, \tilde{e_2}, \tilde{e_3} \in \{0, 1, -1\}\} \tag{5.5}$$

Navigation Model (NM). A navigation model (NM) corresponds to a pre-defined set of allowed process interactions. This set may contain 1-dimensional as well as multi-dimensional process interactions. According to Formula (5.4) and (5.5), and due to the fact that 1-dimensional interactions constitute a subset of multi-dimensional process interactions (5.6a), the set of all possible process interactions Int_{total} can be defined as follows:

$$Int_{oneDim} \subset Int_{multiDim} \tag{5.6a}$$

$$Int_{total} = Int_{multiDim} \tag{5.6b}$$

The set of allowed process interactions may be further reduced by manually eliminating all elements from the set of forbidden process interactions $Int_{forbidden}$. Thus, NM can be defined as follows:

$$NM = Int_{total} \backslash Int_{forbidden} \tag{5.7}$$

Navigation Sequence (NavSeq). A navigation sequence corresponds to a sequence of process interactions. More precisely describes the path along which the user navigates from a start navigation state NS_0 to an end navigation state NS_n:

$$\begin{gathered} NavSeq = (a_1, \dots, a_n, NS_0, NS_n) \\ \text{with } a_1, \dots, a_n \in NM \land NS_0, NS_n \in BM \end{gathered} \tag{5.8}$$

Process Navigation (PN). In general, process navigation can be defined as 4-tuple consisting of the basis model, the navigation model, a start state NS_0, and a navigation sequence defined by the user:

$$PN(BM, NM, NS_0, NavSeq) \tag{5.9}$$

5.2 Running Example

We use a running example to illustrate the introduced definitions, i.e., an automotive requirements engineering process (that is based on Use Case 3 as presented in Section 1.1). The corresponding navigation space is shown in Figure 5.1. The schematic representation of the navigation space, which is based on the three navigation dimensions introduced in Chapter 4, is depicted in the center of Figure 5.1. We assume that the requirements engineer is currently working on process task *Create Component Profile* within process *General Specification*. Assume further that the requirements engineer needs to know the process task succeeding the current one in order to find the right contact person for handing over the specification document resulting from his work. For this purpose, he needs to navigate from a default start state $(0, 0, 0)$, to navigation state $(1, 1, 0)$ in which he may access the information needed.

In this simple example, we define $s, g, v \in \{0, 1\}$, i.e., every navigation dimension may be only scaled in two values. Consequently, the overall number of possible navigation states is $2^3 = 8$.

In the following, NS_{total} is manually restricted by excluding two states: $(1, 0, 0)$ and $(1, 0, 1)$. These two states provide too many information items on the screen and would thus confuse the user. Consider again of the Google Maps scenario, where all city names might be shown in the semantic dimension, but the entire globe be shown in the geographic dimension at the same time. Considering Formula (5.10) and (5.11), the basis model BM can then be defined as shown in (5.12):

$$NS_{total} = \{(0, 0, 0), (0, 0, 1), \dots, (1, 1, 1)\} \tag{5.10}$$

$$NS_{forbidden} = \{(1, 0, 0), (1, 0, 1)\} \tag{5.11}$$

$$BM = \{(0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 1, 0), (1, 1, 1)\} \tag{5.12}$$

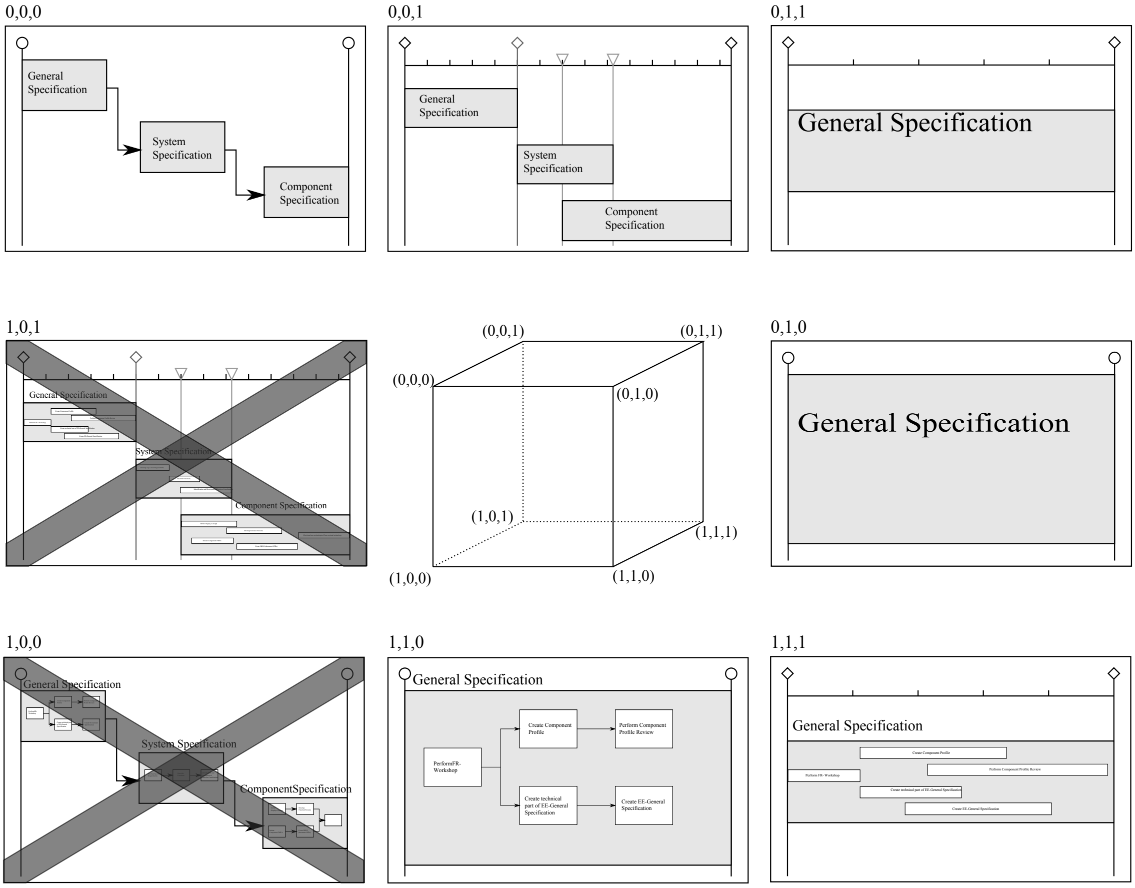


Figure 5.1: Running example illustrating a navigation space with 8 navigation states.

In this simple example, only 1-dimensional process interactions shall be allowed. Therefore, we restrict Int_{total} by excluding all other possible process interactions (i.e., $Int_{forbidden}$):

$$NM = \left\{ a \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, a \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, a \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}, \ a \in \{1, -1\} \tag{5.13}$$

Based on the definition of process navigation (cf. Formula 5.9), we can now investigate user-driven navigation sequences. For each process interaction, we can calculate whether or not the requirement engineer leaves the BM (i.e., he reaches a navigation state not being an element of BM). For example, assume that he applies the following navigation sequence:

$$NavSeq = \left(i_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, i_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \right) \tag{5.14}$$

$NavSeq$ comprises two process interactions. More precisely, i_1 corresponds to a geographical zooming without changing the level of information detail, whereas i_2 corresponds to an increase of the level of information detail. In the following, we apply both navigation interactions to the given BM.

Step 1: We first calculate navigation state NS_1 (i.e., the requirements engineer adjusts the geographic dimension to zoom into the *General Specification* process, cf. Fig. 5.1). Therefore, we add the first vector i_1 to start state NS_0:

$$NS_0 + i_1 = NS_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \tag{5.15}$$

As a result, we obtain navigation state $(0, 1, 0) \in BM$. Hence, Step 1 constitutes a valid process interaction.

Step 2: From the newly obtained state NS_1 (i.e., the new start state) the requirements engineer now wants to increase the level of information detail, i.e., the value of the semantic dimension is increased to display the activities within process step *General Specification*. This process interaction i_2 can be performed similarly to Step 1:

$$NS_1 + i_2 = NS_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \tag{5.16}$$

Since NS_2 also constitutes an element of BM, $NavSeq$ corresponds to an allowed navigation sequence.

If the user chooses another navigation sequence to reach the preferred end state $(1, 1, 0)$, the result might be different. For example, a navigation sequence may start by increasing the value of the semantic dimension, i.e., by applying process interaction $(0, 1, 0)$. Then, the resulting state will be $(0, 1, 0)$, which is not an element of BM; i.e., $(0, 1, 0)$ constitutes a forbidden state and hence user must not navigate to this state.

By calculating allowed navigation options in advance, i.e., before the user action takes place, the framework can guide the user such that he does not follow a forbidden path during navigation. In turn, this increases navigation efficiency.

5.3 Advanced Formalizations

5.3.1 Reachability

Taking the running example (cf. Fig. 5.1), we investigate possibilities to navigate from a given navigation state to other states. Such consideration is useful to effectively support users in navigating within the navigation space. Think of a scenario in which a user is initially situated in navigation state $(0, 0, 0)$. As navigation spaces could become much more complex than the one presented in the running example, the user does not always know how the basis model BM looks like in detail, i.e., he does not know to which navigation state(s) he may navigate.

To avoid invalid navigation, like the one from navigation state $(0,0,0)$ to forbidden state $(1,0,0)$, it is important to provide users with recommendations regarding the allowed navigation options (i.e., process interactions) in a given state. In particular, it is important to identify allowed neighboring navigation states.

The *neighbor* concept describes two navigation states P_1 and P_2 that my be reached from each other by applying exactly one single process interaction. Since we differentiate between 1- and multi-dimensional process interactions, we distinguish between 1- and multi-dimensional neighbors as well.

1-dimensional Neighbors. Two navigation states P_1 and P_2 are 1-dimensional neighbors if a user may navigate from P_1 to P_2 (or vice versa) by applying exactly one 1-dimensional process interaction. If solely 1-dimensional process interactions are allowed, the user may only navigate to 1-dimensional neighbors of the current state:

$$\begin{gathered} P_1 \text{ is a 1-dimensional neighbor of } P_2 \text{ iff} \\ P_1, P_2 \in BM \wedge \exists \vec{e} \in Int_{oneDim} : P_1 + \vec{e} = P_2 \end{gathered} \tag{5.17}$$

Multi-dimensional Neighbors. Reconsider the running example (cf. Fig. 5.1) and assume that a user wants to navigate from $(0,0,0)$ to $(1,1,0)$. This could be accomplished by two consecutive one-dimensional process interactions. Generally, two states P_1 and P_2 are multi-dimensional neighbors, if P_2 is reachable from P_1 through a multi-dimensional process interaction:

$$\begin{gathered} P_1 \text{ is multi-dimensional neighbor of } P_2 \text{ iff} \\ P_1, P_2 \in BM \wedge \exists \vec{e} \in Int_{multiDim} : P_1 + \vec{e} = P_2 \end{gathered} \tag{5.18}$$

Reachable Navigation States. A state P_2 is reachable from a state P_1 if there exists a navigation sequence that allows the user to navigate from P_1 to P_2. Thereby, the neighbor concept may be applied in every process navigation step. As precondition, both P_1 and P_2 must be elements of BM:

$$\begin{aligned} & P_1 \text{ is reachable from } P_2 \text{ iff} \\ & P_1, P_2 \in BM \wedge \exists (n_1, \ldots, n_z) \text{ with } n_1, \ldots, n_z \in Int_{multiDim} \\ & \wedge\ P_1 + \textstyle\sum_{i=1}^{z} n_i = P_2 \wedge P_1 + \sum_{i=1}^{m} n_i \in BM\ \forall m \in \{1, \ldots, z\} \end{aligned} \tag{5.19}$$

Knowing neighbors and reachable navigation states allows determining the navigation options a user has. If a user is currently in a certain navigation state, he can be guided by recommending only those process interactions to him that result in allowed neighbors. Note that this prohibits any trial-and-error navigation.

5.3.2 Distance

A navigation sequence applied by a user also reflects the number of conducted state transitions between two navigation states. In turn, state transitions may require several user interactions (e.g., mouse clicks in an Intranet portal). Assuming that a user only applies 1-dimensional process interactions, the number of user interactions corresponds to the number of mouse clicks. To decrease the latter (i.e., to enable more efficient process navigation), the length of the chosen navigation sequence from a start state to a desired target state should be minimized.

As mentioned in Section 5.1, in general, we assume that the values of each navigation dimension correspond to natural numbers. Accordingly, the distance between two arbitrary navigation states P_1 and P_2 can be calculated as follows:

$$
\begin{aligned}
DIST(P_1, P_2) = \sqrt{(s_1 - s_2)^2 + (g_1 - g_2)^2 + (v_1 - v_2)^2} \\
\text{with } P_i = (s_i, g_i, v_i) \text{ ; } i = 1, 2
\end{aligned}
\tag{5.20}
$$

Note that this metric can be applied to arbitrary states of the navigation space, i.e., the two states do not necessarily have to be 1- or multi-dimensional neighbors. Furthermore, we can measure the overall length of a navigation path chosen by a user to navigate within the navigation space. This distance corresponds to the sum of 1- and multi-dimensional process interactions:

$$NAVDIST(NavSeq) = \sum_{i=1}^{n} \|a_i\| \text{ where } a_1, \ldots, a_n \in NavSeq \tag{5.21}$$

5.3.3 Quality

To obtain information about the quality of a chosen navigation sequence, we can measure its effectiveness. This means that we calculate how quickly the user reaches his navigation goal when applying a navigation sequence. For this purpose, we consider the ratio of the distance between the start and end point of the navigation sequence on the one hand and the length of the applied navigation sequence on the other. Note that this not only allows us to compare different navigation sequences, but also allows for better user assistance, e.g., based on recommendations about shorter navigation sequences. Thus, a more effective navigation path might be provided, when the process participant wants to revisit a particular navigation state later:

$$Eff(P_1, P_2, NavSeq) = \frac{DIST(P_1, P_2)}{NAVDIST(NavSeq)} \tag{5.22}$$

5.4 Applying the Navigation Space

We apply the navigation framework to a scenario characterized by a larger number of navigation states. Figure 5.2a shows a snippet of the navigation space introduced in Chapter 4. White cubes represent the basis model BM, i.e., the set of allowed navigation states. In turn, grey cubes represent navigation states on the navigation sequence of the user. Finally, dark grey cubes represent forbidden navigation states from set $NS_{forbidden}$.

We assume that a user wants to navigate from start state $(0, 0, 0)$ to end state $(6, 1, 0)$. This corresponds to Use Case 1 from Section 1.1: A project manager tries to identify project delays. Therefore, he needs detailed information about due dates, durations, and data objects (along the semantic dimension). Additionally, he requires an overview of all process steps of the project, i.e., on all process models within a process area (along the geographic dimension).

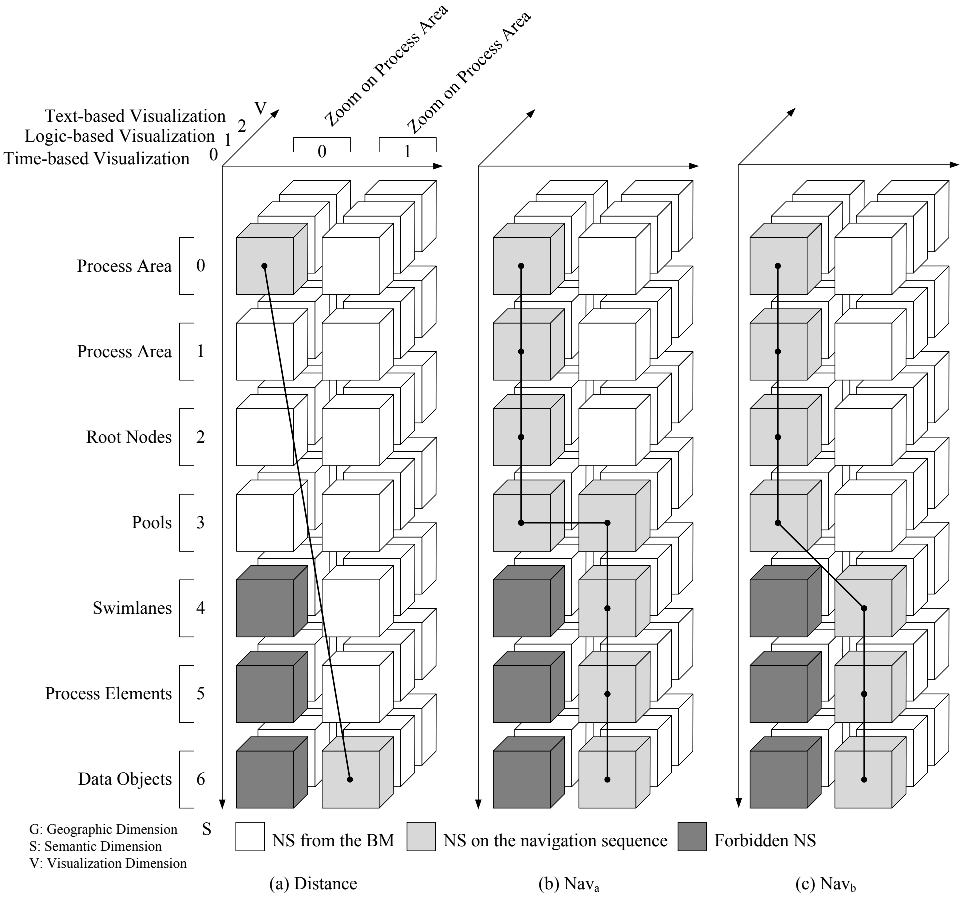


Figure 5.2: Example of calculating the quality of navigation sequences.

First, we check for the reachability of the end state from the start state based on the given basis model. Thereby, we can check whether the needed navigation state may be displayed at the desired semantic and geographic level and whether the user may navigate to this state based on the given navigation model.

In navigation state $(0,3,0)$, for example, a further increase of the semantic dimension would result in an information overflow, i.e., in a forbidden navigation state. Consequently, the project manager has to change the geographic level, focusing on a more specific process area, before he might further increase the level of detail along the semantic dimension.

Second, we measure the distance between start and end navigation state as metric to investigate the user's navigation sequence (cf. Fig. 5.2a):

$$DIST(Start, End) = \sqrt{6^2 + 1^2} \approx 6{,}08 \tag{5.23}$$

We now investigate the manager's navigation sequence, while navigating within the navigation space, i.e., navigation sequence Nav_a from Fig. 5.2b. The manager applies seven 1-dimensional process interactions to reach the end state. Hence, the distance can be calculated as follows:

$$DIST(Nav_a) = \sum_{i=0}^{n-1} DIST(P_{i+1}, P_i) = \sum_{0}^{6} 1 = 7 \tag{5.24}$$

Regarding the considered scenario, the project manager might only be interested in adjusting the semantic dimension as his main goal is to obtain these data objects being independent from the applied geographic dimension. In particular, the geographic dimension could be adjusted accordingly (from navigation state $(3, 0, 0)$ to state $(4, 1, 0)$) in order to avoid an information overflow. In this context, a multi-dimensional process interaction could be applied automatically as soon as semantic zooming would result in a forbidden navigation state. Additionally, applying a multi-dimensional process interaction reduces the user path by one interaction (cf. Fig. 5.2c). The distance of Nav_b can then be calculated as follows:

$$DIST(Nav_b) = \sum_{i=0}^{n-1} DIST(P_{i+1}, P_i) = 1 + 1 + 1 + \sqrt{2} + 1 + 1 \approx 6{,}41 \tag{5.25}$$

Using the ratio to calculate the effectivity of a navigation sequence Eff, the following effectiveness ratios can be calculated for Nav_a and Nav_b respectively:

$$Eff(Start, End, Nav_a) = \frac{6{,}08}{7} \approx 86{,}86\% \tag{5.26a}$$

$$Eff(Start, End, Nav_b) = \frac{6{,}08}{6{,}41} \approx 94{,}85\% \tag{5.26b}$$

As can be seen in Formula 5.26a and 5.26b, suggesting navigation shortcuts can result in a more effective navigation path in Nav_b as indicated by the effectiveness ratios 94,85% and 86,86%, respectively. This effect increases with the number of shortcuts. If typical navigation sequences can be assigned to specific roles, further path suggestions could already be made before the user starts navigating.

Finally, the example indicates how the process navigation framework can be applied to Use Case 1. Again, we use neighbors to measure distances as well as to calculate the effectiveness of navigation sequences. In particular, more efficient navigation becomes possible when eliminating unnecessary process interactions.

5.5 Related Approaches

Besides Linear Algebra, other formalization approaches might be applicable to create a formal model for process navigation based on the presented navigation space. This section compares four alternative approaches and explains why we used Linear Algebra: Finit State Machines (FSM), process navigation (PN), State Transition Systems (STS), and Linear Algebra (LA).

Each approach is evaluated based on seven criteria:

1. **Ease of modeling:** *Ease of modeling* corresponds to the difficulty and the effort required to develop a comprehensible formal model.

2. **Bidirectionality:** *Bidirectionality* corresponds to the ability to reflect process navigation along all navigation dimensions in both directions.

3. **Extensibility:** *Extensibility* corresponds to the effort to add additional navigation states or navigation dimensions to the formal model.

4. **Complexity:** *Complexity* refers to the increasing complexity of a formal model if the navigation space increases.

5. **Comprehensibility:** *Comprehensibility* refers to the difficulty to comprehend a formal model.

6. **Ease of use:** *Ease of use* reflects the effort to map or apply a formalization approach to process navigation.

7. **Memory usage:** *Memory usage* refers to the effort to save and maintain a formal model.

Creating Finit State Machines (FSM) [WSWW06, Ped13] is time consuming as each related state and respective transitions must then be modeled separately (*ease of modeling*). However, bidirectional transitions can be expressed (*bidirectionality*). Extending the model by new navigation states will be complex as state transitions to other states must be considered (*extensibility*). Using FSM, *complexity* increases as the number of states in the formal model increases exponentially in multi-dimensional navigation spaces. In turn, FSM are understandable as only few different modeling elements are required (*comprehensibility*). However, *ease of use* is limited due to the complexity of FSM. Finally, *memory usage* is considerably high as the respective formal model must be predefined and states as well as state transitions must be maintained separately.

process navigation (PN) [Rei13] provide different elements and rules. In general, the effort to formalize the navigation space using PN would be considerably high (*ease of modeling*). Bidirectional navigation sequences may be realized by modeling two separate transitions (*bidirectionality*). The *extensibility* of a PN formal model, however, is limited as the number of states exponentially grows when adding navigation dimensions (*complexity*). Furthermore, *comprehensibility* and *ease of use* of PN are rather low. In turn, the *memory usage* is rather high as each navigation state must be maintained separately.

The STS [BK08] is complex as states and transitions must be modeled separately as well (*ease of modeling*). An STS can be realized as simple table, thus designing a formal model is optional. *Bidirectionality* is supported using directed transitions or respective table entries. The realization as a table also allows for simple *extensibility*. In turn, the *complexity* of the formalization approach can be compared to the one from FSM as for each new state all transitions to other states must be newly created. Considering the table visualization, *comprehensibility* of an STS is good. The formalization approach is easy to apply, as the set of allowed transitions is predefined and stored in the table. Included information can be easily extracted (*ease of use*). Finally, storing the respective table is less space consuming compared to storing the formal model (*memory usage*).

Using the Linear Algebra (LA) approach, the formal model of the navigation space can be represented by the Cartesian System (*ease of modeling*), which makes it easy to create a formal model. As state transitions can be represented by vectors, *bidirectionality* is supported as well. The navigation space can be easily extended as the size of the Cartesian System is infinite (*extensibility*). Thus, enlarging the navigation space only leads to an increase of *complexity* when dimensions are added. Moreover, the LA approach can deal with multi-dimensional navigation spaces without an exponentially increasing *complexity*. LA is a lean approach that can be easily understood by modelers (*comprehensibility* and *ease* *of use*). The *memory usage* is comparatively low, also due to the fact that no explicit model needs to be stored.

Figure 5.1 summarizes findings. PN provide the worst results as they provide complex elements and sets of rules, which needs to be taken into consideration while creating a formal model. Thus, PN are too complex in our context. Most criteria (except for one) received a negative rating.

	Approach			
Criteria	FSM	PM	STS	LA
Ease of modeling	-	- -	-	n.a.
Bidirectionality	o	-	+	++
Extensibility	-	-	++	++
Complexity	-	- -	-	++
Comprehensibility	o	- -	++	+
Ease of use	-	- -	++	++
Memory usage	- -	- -	-	++

Table 5.1: Comparison of different formalization approaches.

FSM do not adequately allow formalizing process navigation as its application is rather complex in our context. Within a process navigation scenario, for example, each state needs to be considered as final state, and state transitions must be manually created for each new state. Therefore, the concept of FSM does not match the requirements for realizing process navigation.

STS, in turn, show better results. As a matter of fact, STS provide a limited set of elements and rules and might therefore be applied to process navigation more intuitively. Modeling navigation states and state transitions could be applied to process navigation. STS are easy to use, extensible, and are able to cope with bidirectional navigation.

Altogether, LA shows the best results among the evaluated formalization approaches. In particular, the navigation space corresponds to a multidimensional Cartesian System. Accordingly, a navigation space can be defined easily using LA. Furthermore, extending the navigation space does not implicate additional efforts, as new states can be simply defined by adding points to the Cartesian System. LA is a lean, but powerful approach to formalize process navigation.

5.6 Summary

This chapter illustrated how process navigation within a process space can be formalized using Linear Algebra. This formalization might be used as basis to support the user when navigating within a navigation space. The basis model constitutes the foundation of the navigation approach. It defines the allowed navigation states during process navigation. Within the basis model, the user may navigate without any other limitations. The basis model dismisses navigation states within the navigation space, which are not suitable for process participants. For example, states including too much or too little information can be forbidden. The navigation model, in turn, defines allowed interactions, i.e., allowed state transitions between navigation states. In this context, 1-dimensional process interaction is introduced as basic interaction concept. In turn, multi-dimensional process interactions allow for a more complex process interaction along multiple navigation dimensions at once.

Combining the basis model and the navigation model, the formalization approach is able to support and guide users when navigating within the navigation space. For illustration purposes, a running example was introduced to show how the basis model and the navigation model can be used within a given navigation space and how a user can be guided, while interacting with the navigation space. Finally, we presented selected approaches to formalize process navigation.

6 Visualizing the Navigation Space

After formalizing the navigation space, this chapter[1] introduces concepts for visualizing navigation states along the visualization dimension. The overall goal is to visualize single navigation states in a user-adequate manner (cf. Figure 6.1). Thereby, different visualization types should be used to emphasize specific process information (e.g., temporal aspects), while hiding non-relevant [BBR06, Bob08]. The concepts introduced in this chapter were derived from the case studies (cf. Chapter 2).

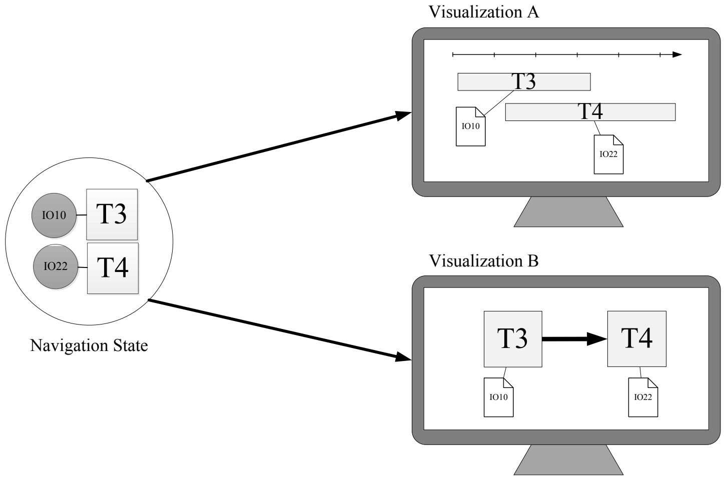


Figure 6.1: Visualizing a navigation state.

In order to properly visualize a particular navigation state, we need to consider all objects from the process space (cf. Section 4.3), i.e., process areas, process objects (root nodes, pools, swimlanes, events, gateways, tasks, sequence flow, data flow), and data objects.

This chapter is structured as follows. Section 6.1 presents background information. Section 6.2 introduces four different visualization types. Section 6.3 then presents three specific visualization approaches with respect to BPMN, which is the most popular and widespread business process modeling language. Section 6.4 discusses related visualization approaches and Section 6.5 summarizes the chapter.

6.1 Background Information

Process model collections can become very large and complex [OS08, WRMR11]. Despite the complexity of a process, process participants need to quickly understand process models in order to perform their

[1]The chapter is based on the following referred paper [HSM+14]:
Markus Hipp, Achim Strauss, Bernd Michelberger, Bela Mutschler, and Manfred Reichert. *Enabling a User-Friendly Visualization of Business Process Models.* in: Proc 3rd Int'l Workshop on Theory and Applications of Process Visualization (TaProViz'14), pp. 395-407, 2014

work in the best possible way [MKR12]. In this context, the visualization of process models adopts a key role [Ras00, JGH$^+$08]. In particular, it has significant effects on the *understandability* [MRC07], *aesthetic appearance* [Nor88], and *clarity* [RMD11] of process models. In other terms, a non-adequate visualization of process models negatively affects user acceptance [ISO95, ISO98, May99, RC01, SBH$^+$05].

There exists a lot of research in the area of information visualization [Spe00, CRM91, CMS99]. However, looking at the visualization of process models from a user's perspective has been neglected so far with few exceptions (e.g., [BBR06, KLMR13]). Process modeling notations (such as BPMN or event-driven process chain (EPC)) are typically used to visualize process models. Existing process modeling tools, like WBI Modeller [IBM06], Signavio Process Modeler[2] or ARIS WebPublisher [ARI07], typically, do not provide alternative visualization approaches; i.e., the same symbols are used for both modeling and visualization, i.e., process models are visualized to end users in the same way they were drawn by the modelers [BRB05, Rei12]. Unfortunately, existing notations do often not allow for user-adequate visualizations as they might be hard to understand by inexperienced process participants (e.g., think of a nurse in a hospital).

We pick up this weakness and introduce novel visualization types for process model collections. Logically, these visualizations correspond to specific navigation states (cf. Chapter 4). In particular, users might switch between visualizations depending on their information demands.

6.2 Visualization Types

Existing approaches for generating user-specific visualizations of process models [BBR06, BRB07, BRW11, KKR12] show that the complexity of process models may be reduced, for example, by applying *aggregation* and *reduction* techniques (e.g., aggregating different process task to one abstract task, or reducing the number of process tasks by hiding selected tasks [BRB07, KKR12, SRW11]). In our context, this corresponds to a combination of the visualization and semantic dimensions. Our ambition, however, is to derive visualization types based on specific user needs as process information visualization directly affects user acceptance [Bir33].

We have already shown how the semantic and geographic dimension support different abstraction and zooming levels (cf. Chapter 4). Both navigation dimensions enable users to tailor a process model collection on the desired semantic and geographic level. We have further shown how users may benefit from visualizing multiple navigation states at once.

Req #	**Requirement**
VisReq #1	Task descriptions must be documented in a well understandable manner.
VisReq #2	Temporal and logical dependencies must be considered when visualizing processes.
VisReq #3	Complex process information must be visualized in a comprehensible manner.
VisReq #4	Information about roles must be intuitively identifiable.
VisReq #5	The amount of visualized information should not overload process participants.

Table 6.1: Overview of all visualization requirements.

This chapter presents four basic concepts for visualizing one or multiple navigation states. The presented visualization types rely on the visualization requirements discussed in Chapter 2 (cf. Table 6.1).

[2]Signavio Process Editor: www.signavio.de

6.2.1 Time-based Visualization

In many domains, the proper visualization of temporal constraints in process models (see [LWR14]) is crucial in order to successfully perform a process (e.g., flight planning, patient treatment, and automotive engineering) [EPR99, CGJ+07]. This has been confirmed by interviewees in the context of our case studies (cf. Chapter 2). Especially, managers require temporal information when asking for an overview on process models.

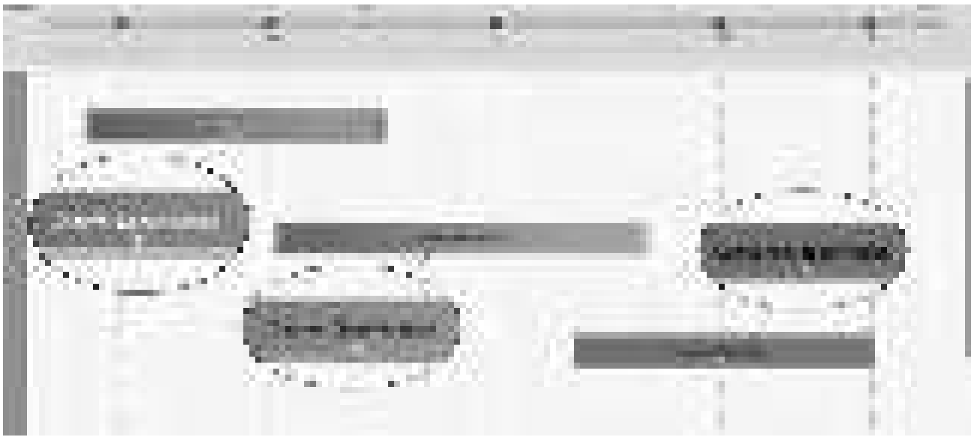


Figure 6.2: Time-based visualization ($NS(3, 1, 0).combine(NS(2, 1, 0))$).

This section introduces a *time-based* visualization type (cf.Figure 6.2). It has been inspired by existing approaches using Gantt Charts [Cla22, May01, SGL12, KRM12, LKR13]. Table 6.2 shows which objects from a navigation state are considered by this visualization type and how these objects are visualized. The time-based visualization emphasizes objects providing explicit temporal information. In turn, non relevant objects are hidden. Thus, events, gateways, and sequence flows are not considered for this type of visualization. Indeed, process areas, process root nodes (representing entire process models), and process tasks are visualized. In particular, their duration (from the start to the end time) is visually represented by their length, i.e., the width of the rectangles representing the process tasks.

Object	Considered	Visualized as
process area	✓	rectangle
process root node	✓	rectangle
pool	✓	color
swimlane	✓	color
event	✗	
gateway	✗	
task	✓	rectangle
sequence flow	✗	
data flow	✓	straight arrow
data object	✓	document (container) icon

Table 6.2: Considered objects in the time-based visualization.

There exist objects that do not provide temporal information, but constitute a better structuring of information visualized: swimlanes and pools. In turn, these objects are represented in different colors. Finally, sequence flows, gateways, and events are factored out as this information is not required for a time-based visualization.

An example of the time-based visualization is shown in Figure 6.2. It is based on the navigation space defined in Chapter 4. The time-based visualization is applied to the combined navigation states NS$(3, 1, 0)$ and NS$(2, 1, 0)$, representing three process root nodes subsumed within a process area (*component specification*, *system specification*, and *general specification*).

To further increase the simplicity of the visualization, data objects and data flows are only shown on demand, i.e., the respective navigation state on semantic level 6 (NS$(6, 1, 0)$) may be added to the visualization in case the data flow between objects (cf Figure 6.3) should be followed. Thin straight arrows are used to visualize data flow. In turn, document icons are used to represent data objects.

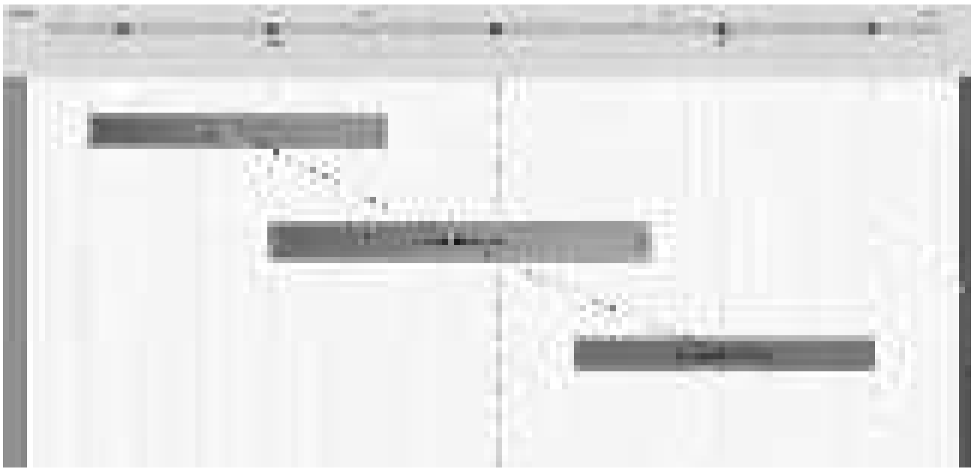


Figure 6.3: Time-based visualization with data flow ($NS(6, 1, 0).combine(NS(3, 1, 0), NS(2, 1, 0))$).

The time-based visualization focuses on temporal dependencies. Therefore, it omits all information not related to any time-depending aspects.

6.2.2 Logic-based Visualization

The *logic-based* visualization allows visualizing logic relations between objects, i.e., predecessor and successor relations. Table 6.3 shows which objects are considered in the logic-based visualization and how they are visualized.

Object	Considered	Visualized as
process area	✓	standardized rectangle
process root node	✓	standardized rectangle
pool	✓	lane
swimlane	✓	lan
event	✓	event
gateway	✓	gateway
task	✓	standardized rectangle
sequence flow	✓	sequence flow
data flow	✓	arrow
data object	✓	document icon

Table 6.3: Considered objects in the logic-based visualization.

As an example consider Figure 6.4. It shows a logic-based visualization of the general specification process model introduced in Chapter 1. Based on the navigation space presented in Chapter 4, the logic-based visualization is applied to a combination of navigation states NS(6,2,0), NS(5,2,0), and NS(4,2,0), i.e., data objects, tasks, events, gateways, and swimlanes are considered (corresponding to semantic levels 4, 5, and 6). Geographic level 2 indicates the zoom on a certain process model. Furthermore, swimlanes are provided by colored stripes on their left border (*VisReq #4*). Process tasks are visualized as rectangular boxes within the lanes including its title. All boxes have similar lengths. Logic relations, i.e., the sequence flow, are visualized by arrows between objects. Events and gateways are presented as circles and diamonds. Furthermore, document icons are used to visualize data objects, the corresponding data flow is represented by dotted arrows. Finally, the logic-based visualization focuses on logic relations between objects, taking common process model notation standards, such as BPMN, into account as well.

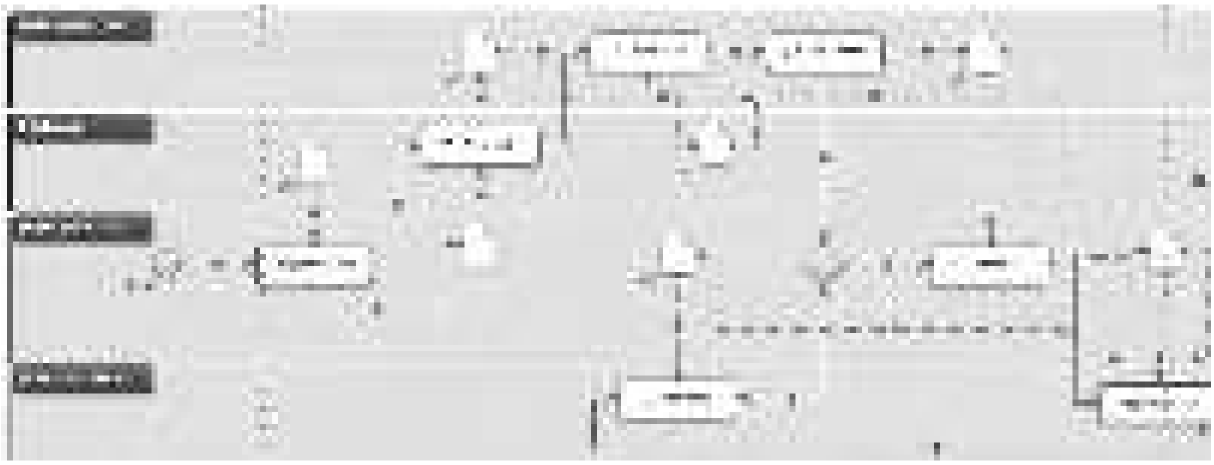


Figure 6.4: Logic-based visualization ($NS(6,2,0).combine(NS(5,2,0), NS(4,2,0))$).

6.2.3 Text-based Visualization

Employees working on knowledge-intensive process models need access to detailed descriptions about process tasks. Providing only task labels as in the logic-based visualization (cf. Figure 6.4) is not sufficient. Instead, users should be provided with detailed textual descriptions of single process objects (*VisReq #1*). Note that this is crucial when complex tasks must be processed or decisions must be made. Table 6.4 shows the objects considered in the *text-based* visualization type.

Object	Considered	Visualized as
process area	✓	textual description
process root node	✓	textual description
pool	✗	
swimlane	✗	
event	✗	
gateway	✗	
task	✓	textual description
sequence flow	✓	partially to predecessor and successor
data flow	✓	implicitly by the link to the data object
data object	✓	clickable link

Table 6.4: Considered objects in the text-based visualization.

Note that we distinguish between two different text-based visualizations. The *turtle* visualization on the one hand, and the *content* visualization on the other.

Turtle Visualization

Figure 6.5 presents the turtle visualization approach. The latter was developed to support employees working on knowledge-intensive process tasks. More precisely single steps of a process task are described as item list in the center of the visualization (i.e., in the *Task Description* field). In addition to this task description, the turtle visualization offers further information (*VisReq #3*). For example, data objects are presented depending on the specific data flow in the process model either as task input in the box on the left or as task output in the box on the right. Furthermore, two boxes are aligned on top and two at the bottom of the process description. The two boxes on the top display the roles the actor processing the task must have (left; linked to detailed role descriptions) and support documents (right; data objects, such as manuals or guidelines). The box on the bottom present preconditions.

Figure 6.5: Turtle visualization.

The turtle visualization might be used to visualize single process tasks. It is well structured assists users on performing single process tasks. However, it might be applied to more abstract objects such as process root nodes and process areas.

Content Visualization

For new employees, in turn, task descriptions in terms of item lists are not suitable. Respective users typically need more detailed information in textual form (*VisReq #1*). For this purpose, we introduce the content visualization (cf. Figure 6.6) which provides verbalized textual information on process subjects in a less structured way. The layouting of the content visualization was inspired by the one of *Wikipedia*, i.e., a box containing major information is provided in the top right corner (*Further Information*), whereas all other information is provided in different boxes.

Like the turtle visualization, the content visualization might be applied to more abstract objects such as process root nodes or process areas. For example, this might help managers in getting basic information on a given process model or entire process area.

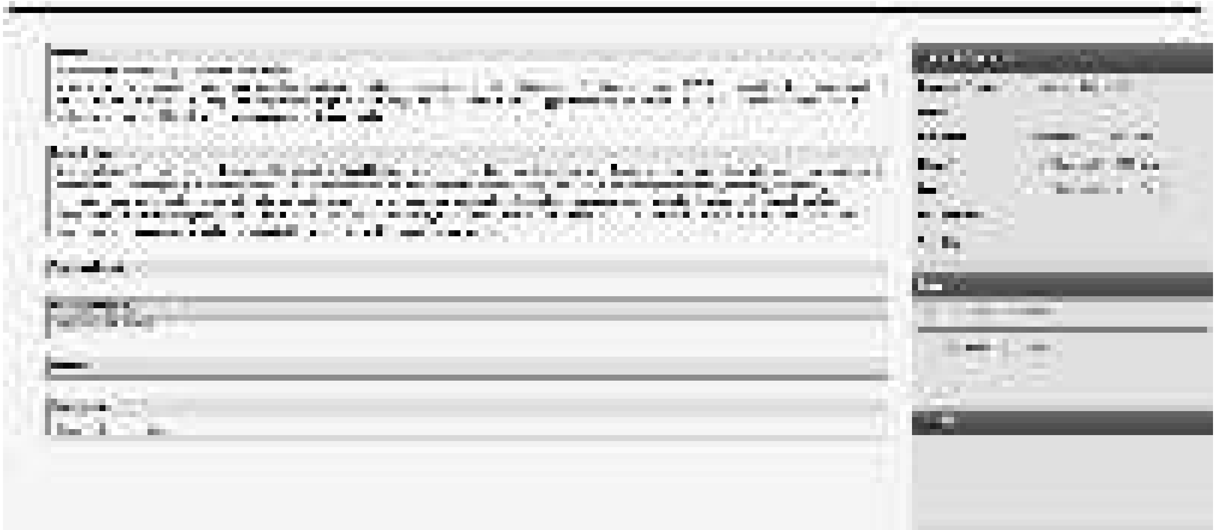


Figure 6.6: Content visualization.

6.2.4 List-based Visualization

Another visualization type is the *list* visualization. It provides a simple, but very structured visualization of a navigation state as a list of entries. Table 6.5 shows the objects considered by this visualization type.

Object	Considered	Visualized as
process area	✓	entry (process)
process root node	✓	entry (process)
pool	✓	entry (role)
swimlane	✓	entry (role)
event	✗	
gateway	✗	
task	✓	entry (process)
sequence flow	✗	
data flow	✗	
data object	✓	entry (artifact)

Table 6.5: Considered objects in the list-based visualization.

The list visualization allows for the structured listing of all objects corresponding to one or several navigation states (cf Figure 6.7). More precisely, objects are organized by different types, i.e., *process*, *role*, and *data objects* (called *artifacts*). In turn, the respective types are visualized by different icons (on the left side of the list). The list may be further filtered according to these types.

6.2.5 Discussion

The presented visualization approaches meet the visualization requirements set out in Chapter 2 (cf. Table 6.6). Furthermore, they are based on a user-centered design approach [ND86], i.e., domain experts were involved during the design phase.

All presented visualization approaches may be accessed through the visualization dimension of the navigation space. In particular, it becomes possible to provide different visualizations types for a specific

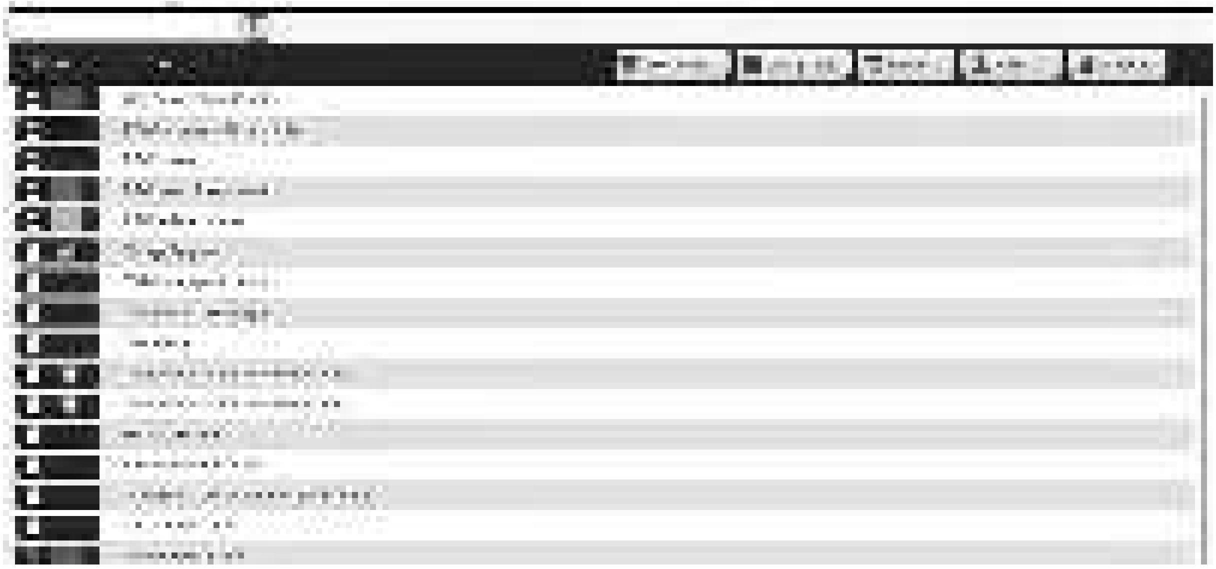


Figure 6.7: List-based visualization.

navigation state, i.e., to present different perspectives on one and the same subject matter. However, as a problem, not every visualization considers all objects of the respective navigation state.

Req #	**Requirement**	time-based	logic-based	text-based	list-based
VisReq #1	Task descriptions must be documented in a well understandable manner.			✓	
VisReq #2	Temporal and logical dependencies must be considered when visualizing processes.	✓	✓		
VisReq #3	Complex process information must be visualized in a comprehensible manner.	✓	✓	✓	
VisReq #4	Information about roles must be intuitively identifiable.	✓	✓		✓
VisReq #5	The amount of visualized information should not overload process participants.	✓	✓	✓	✓

Table 6.6: Requirements met by the visualization types.

Altogether, various visualization types may represent the same navigation state, i.e., the user is able to create a coherent mental representation of the navigation state by summing up the visualization types [Seu03a, Seu03b]. Note that such coherent information is crucial for the processing of information by the users [CS91], i.e., for a profound understanding of the subject matter. With the presented visualization types, the framework offers multiple ways of representing a subject matter, providing redundant as well as complementary information that may be applied for building an elaborated knowledge structure [GRF08].

The following section investigates the logic-based visualization type in a more detailed manner as it is the most widespread visualization type for representing process models. Existing process modeling tools use the BPMN notation in order to visualize process models in a logic-based manner. The following section refines the *logic-based* visualization type by providing four specific visualization approaches based on the BPMN notation.

6.3 Logic-based Visualization Approachess

Typically, complex process models are modeled and visualized using the BPMN language, i.e., in a *logic-based* manner. As example, consider the BPMN-based process model depicted in Figure 6.8, which corresponds to a simplified requirements engineering process from the automotive domain.

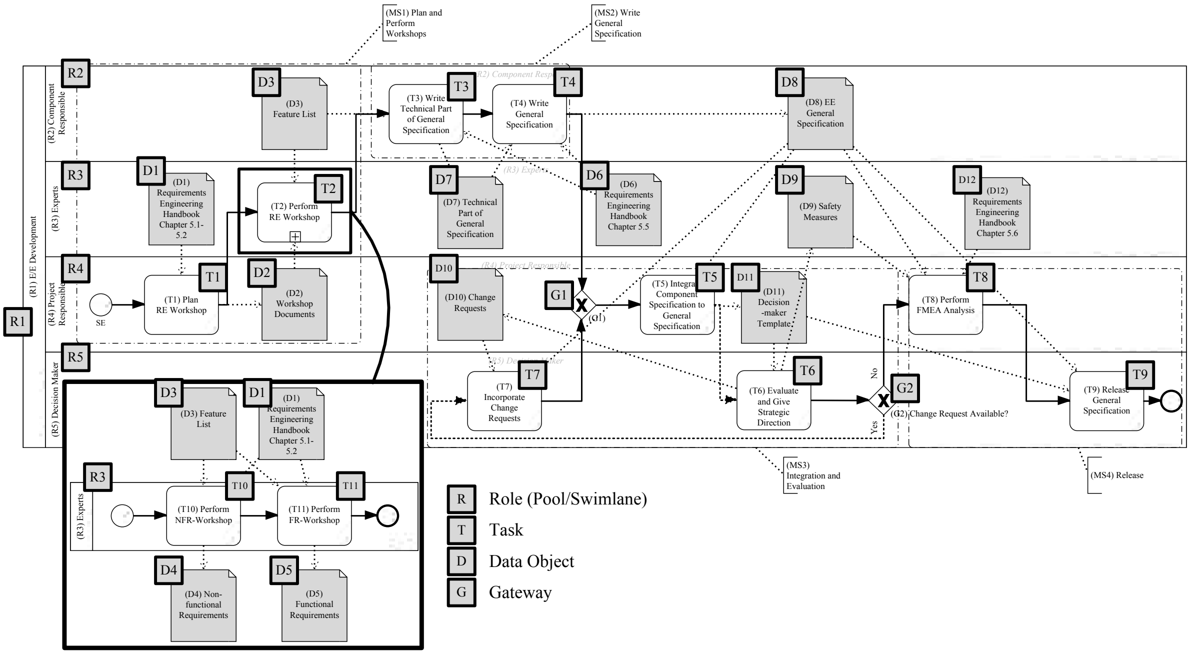


Figure 6.8: Visualization weaknesses in the general specification process.

Note that even this simplified process model reveals significant weaknesses regarding its visualization:

- **Positioning of data objects:** Usually, data objects are positioned right next to process tasks or between them [Rec10]. However, such positioning might be misleading for users; e.g., $D7$ is positioned within swimlane $R3$ although $D7$ is not related to $R3$. Note that $D7$ is solely linked with tasks $T3$ and $T4$ contained in $R2$.
- **Data object relations:** Data objects may be related with more than one process task. In turn, this might lead to "long distance" data relations (i.e., dotted arrows) decreasing model comprehensibility [MW08]. For example, $D8$ is related to five process tasks, resulting in five data relations.
- **Intersections:** Sequence and data flows might overlap. Furthermore, data objects and process tasks might be crossed by data relations (see $D11$ in Fig. 6.8). Usually, such intersections affect the model's comprehensibility [MRC07].

In the context of large process models [RKBB12], corresponding drawbacks significantly affect both the *comprehensibility* [MRC07] and *aesthetic appearance* [Nor88] of process models. To remedy these drawbacks, we present four alternative visualization approaches aiming at a user-friendly, logic-based visualization of process models. Before, we discuss specific requirements for logic-based visualizations of process models.

6.3.1 Visualization Requirements

This section summarizes major requirements regarding the comprehensibility as well as aesthetic appearance of a logic-based visualization of process models. The requirements were derived in the context of two case studies in the automotive and healthcare domains [HMR11b, MMR11a]. In turn, the generalizability of case study results was confirmed by a literature study [MAGM13].

Process model quality is crucial in respect to the comprehensibility of process models [MRC07]. Important factors influencing the comprehensibility of process models include its size as well as the degree of sequencing, concurrency, density, and structure [MMN$^+$06, RFME11, RM11]. Regarding large process models, two requirements are particularly relevant.

Req #1 (Sequence Flow). The sequence flow determines the order of process tasks in a process model and should be visualized in a comprehensible manner.

Req #2 (Clarity). Users should be able to get a quick overview of a process model. In particular, its visualization should enhance the clarity of process models.

Humans are confronted with a continuously growing amount of visual information and, therefore, tend to become more intolerant to non-aesthetic one. Hence, *aesthetic appearance* significantly influences the acceptance of user interfaces [Bir33]. The case studies and literature study have confirmed the importance of aesthetic process model visualizations, especially with respect to two issues:

Req #3 (Interest). To increase their aesthetic appearance, process models must be visualized in an interesting manner as humans are more attracted to visualizations being different from what they already know [Wri03].

Req #4 (Stimulation). People always crave at developing personal knowledge and skills [Wri03]. The aesthetic appearance of process models should stimulate these goals.

In addition to these requirements, related to process model *comprehensibility* and *aesthetic appearance*, the following requirements must be met:

Req #5 (Simplicity). The complexity of a process model has a significant negative influence on its comprehensibility [MMN$^+$06] as well as its aesthetic appearance [Bir33]. Therefore, the visualization of process models should be intuitive and simple.

Req #6 (Appeal). The graphical representation of a process model should support the user's perception of the entire process. In particular, users should feel comfortable when working with process models in order to foster their willingness to reuse the models later on [Wri03]. To achieve this goal, the visualization of process models should be appealing.

Req #	**Name**	**Requirement**
Req #1	Sequence Flow	The sequence flow of a process model must be *comprehensible.*
Req #2	Clarity	The visualization of a process model must be *clear.*
Req #3	Interest	The visualization of a process model must be *interesting.*
Req #4	Stimulation	The visualization of a process model must be *stimulating.*
Req #5	Simplicity	The visualization of a process model must be *simple.*
Req #6	Appeal	The visualization of a process model must be *appealing.*
Req #7	Structure	The visualization of a process model must be *structured.*

Table 6.7: Overview on requirements.

Req #7 (Structure). Mendling et al. [MRC07] state that small variations in process models might lead to significant differences in respect to their comprehensibility. Amongst others, the structuring and sequencing of a process model was identified as a factor positively influencing comprehensibility and aesthetic appearance [Nor88].

Table 6.7 summarizes the derived requirements.

In the following, we present four different concepts for visualizing process models: the *Bubble*, *BPMN3D*, *Network*, and *Thin Line* approaches. In order to ensure comparability as well as to foster readability, the visualization approaches are presented along an abstract process model (cf. Fig. 6.9) including nine tasks (*A-I*) and eight data objects (*D1-D8*).

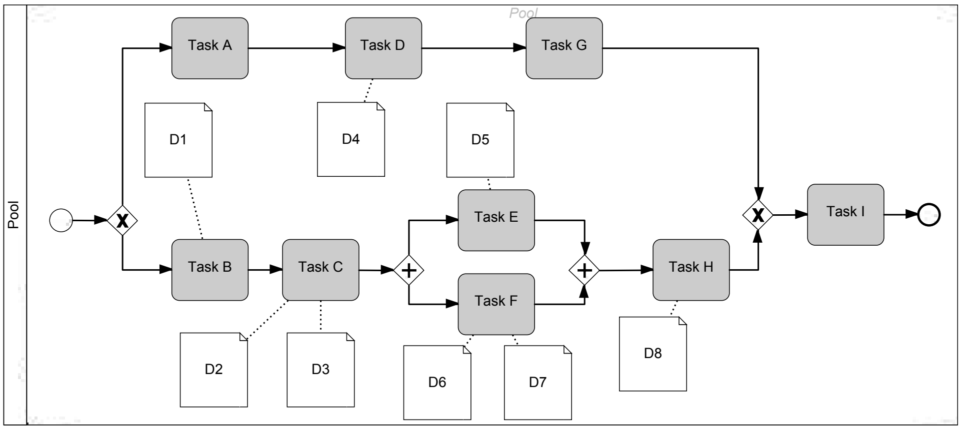


Figure 6.9: Running example.

6.3.2 Bubble Approach

The first visualization approach, called *Bubble*, does not use common shapes like rectangles and hexagons. Instead, it is inspired by a node-oriented network representation. Figure 6.10 shows the application of the *Bubble* concept to our running example. Circles are used to represent process tasks in an appealing, but simple manner (*Req #6*). Thereby, circles have a standardized size, i.e., they do not differ from each another.[3] In particular, circles are graphically better distinguishable from rectangular icons representing data objects [Nor88]. Thus, data objects can be easier identified and more intuitively identified in the

[3]Note that the use of different sizes could indicate an unintended semantic meaning, e.g., bigger circles might be considered as being more important than smaller ones. However, this idea can be picked up in future work, resulting in another dimension for information presentation.

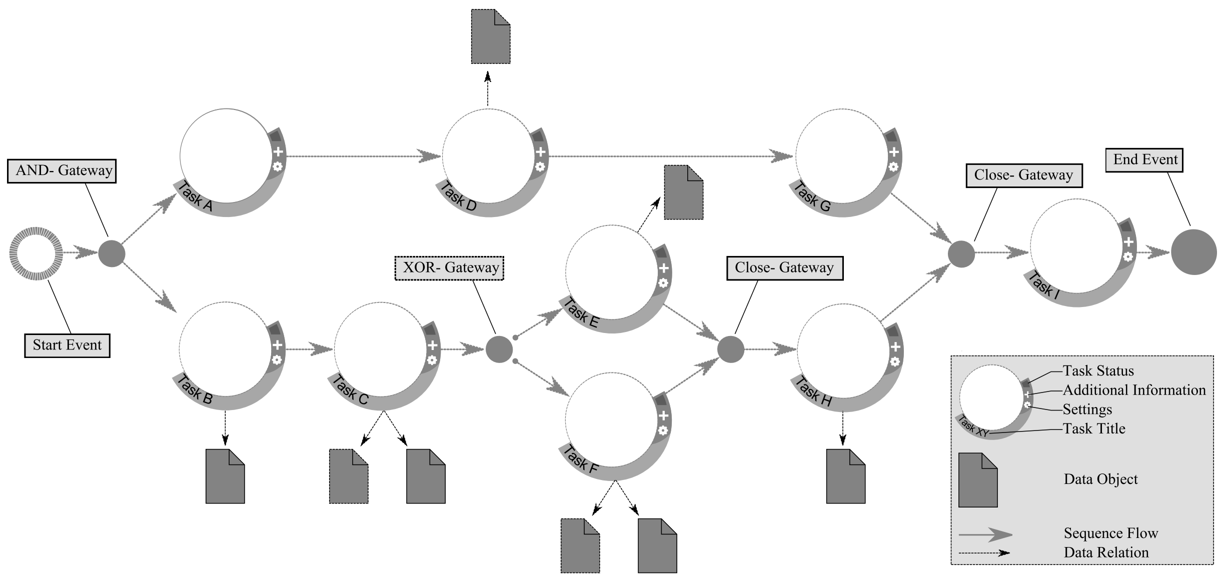


Figure 6.10: Bubble visualization approach.

process model, providing a better overview (*Req #2*) and structure (*Req #7*). In turn, data objects are presented using document icons. Arrows are used to model both the control flow (i.e., the sequence of tasks) and data flow (*Req #1*). The concept uses symbols for gateways and events being similar to the ones known from BPMN. Task labels are added to the task's edge. Finally, additional information may be accessed using the plus and gearwheel buttons, e.g., to detail task descriptions.

6.3.3 BPMN3D Approach

BPMN3D aims to use standard BPMN elements, but "outsources" the visualization of data objects to a third dimension (cf. Fig. 6.11). Process tasks, events and sequence flows are represented through common BPMN elements on a common two-dimensional plain, whereas the presentation of data objects is realized through a third dimension. More precisely, *BPMN3D* extends every process task with a *pole*, pointing to the third dimension, which is then mapped to the 2-dimensional visualization. This idea has been inspired by concepts from Effinger [ES10] and Bobrik [BBR06]. Data objects are aligned to these poles in terms of circles. In turn, icons indicate the type of the data objects (e.g., pdf files, office files, or images). Applying this concept, data objects appear to be more independent from the actual sequence flow. This improves the structure of the process model (*Req #7*) and enables a quick overview on the latter (*Req #2*).

6.3.4 Network Approach

Like *Bubble*, the *Network* concept constitutes a network representation (cf. Fig. 6.12) (cf. *Reqs #3* and *#4*). Each process task is represented through a *node* and comprises a small, centered circle (called *core*) as well as the *galaxy*. The latter offers space for *references*, which may be used to connect a node with other nodes, data objects, or roles.

To reduce the complexity of the visualized process as well as the mental load of the user, this concept focuses on single process tasks, i.e., single nodes. In particular, always one node is dynamically emphasized

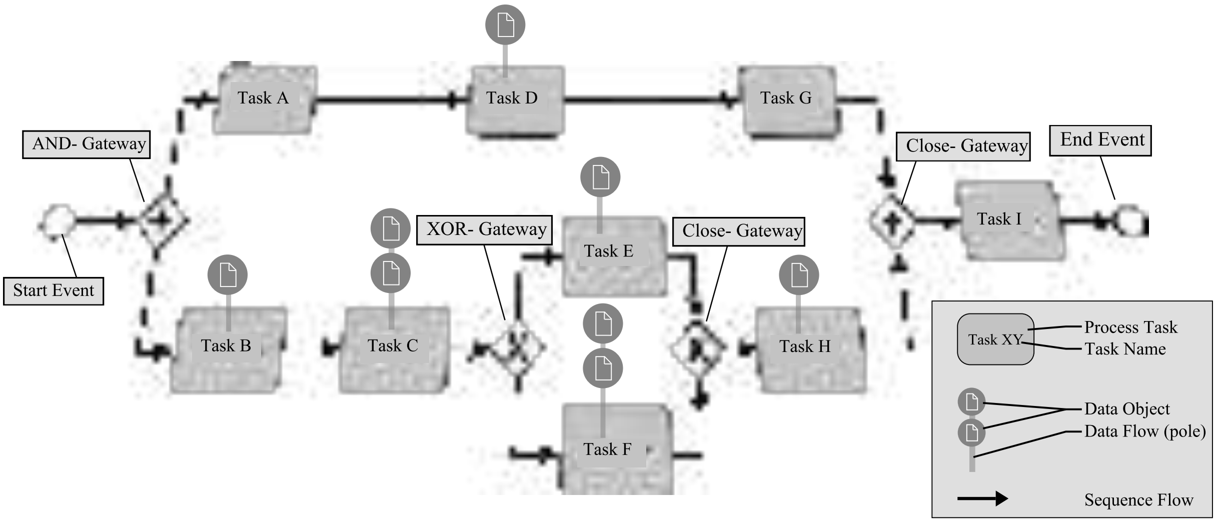


Figure 6.11: BPMN3D visualization approach.

as shown in Fig. 6.12 (Task E in the example). Other nodes and corresponding references, data objects and roles are greyed out. Overall, *Network* provides a new way of visualizing process models (*Req #8*).

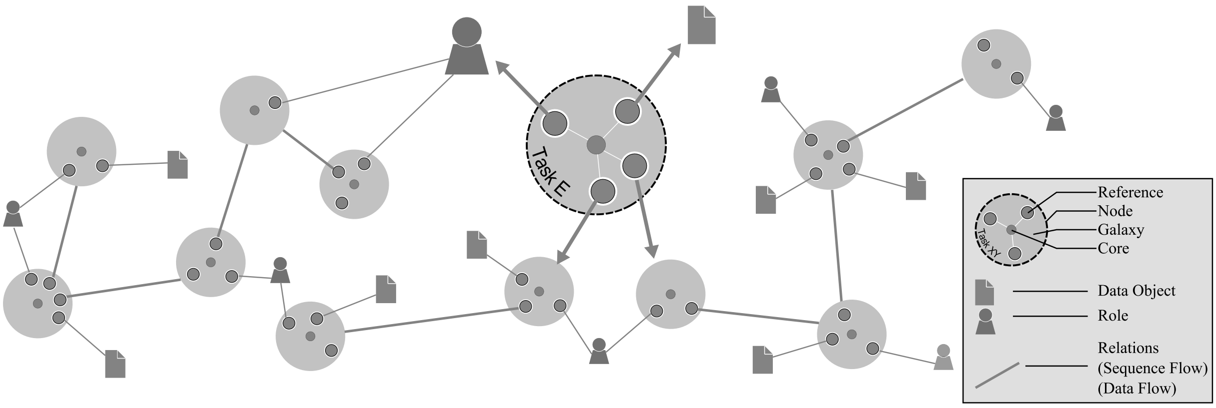


Figure 6.12: Network visualization approach.

6.3.5 ThinLine Approach

The goal of *ThinLine* is to better structure the information displayed. The basic idea is to separate process tasks and sequence flows from data objects. This increases the overview of the process model and facilitates its comprehensibility (cf. Reqs #2 and #7). This approach is inspired by critical path method (CMP) concepts [NM02]. On one hand, users can focus on the sequence flow of the model. On the other, data objects are easily accessible in an explicit area below the sequence flow visualization (cf. Fig. 6.13).

This approach can be considered as a minimalistic one with respect to process visualization. Both process tasks and sequence flow are represented through arrows, which results in a significant reduction of the amount of information displayed (Req #5). The title of a process task is displayed on top of each arrow.

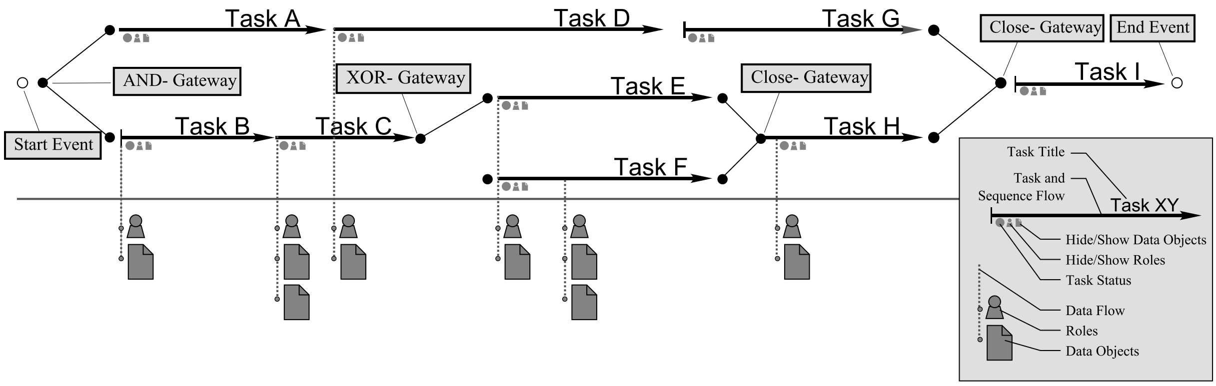


Figure 6.13: ThinLine visualization approach.

Furthermore, additional elements for gateways and events are introduced. Finally, vertical lines guide the user to the area the related data objects are displayed.

6.3.6 Discussion

A detailed presentation of the four visualization types, together with illustrating examples, can be found in [Str12]. Table 6.8 shows the specific visualization requirements and how they are addressed by each of the four logic-based visualization approaches.

Req #	**Name**	**Bubble**	**BPMN3D**	**Network**	**ThinLine**
Req #1	Sequence Flow	✓	✓		✓
Req #2	Clarity	✓	✓		✓
Req #3	Interest	✓	✓	✓	✓
Req #4	Stimulation	✓		✓	✓
Req #5	Simplicity	✓	✓		✓
Req #6	Appeal	✓	✓	✓	✓
Req #7	Structure	✓	✓		✓

Table 6.8: Requirements considered by the visualization approaches.

Sequence flows occur as structural element in all approaches except *Network*. As familiar symbols are used to represent the sequence flow (i.e., arrows), the latter is comprehensible in all three visualization approaches (Req #1). For the same three approaches, we consider clarity, simplicity, and structure as crucial characteristic (Req #2, #5, and #7).

All presented approaches have used new forms of elements, making them more interesting and appealing (Req #3 and #6). This should stimulate users to work with these concepts. We do not expect a stimulation effect with the *BPMN3D* approach, since it is pretty close related to the well-known BPMN standard (Req #4).

Evidence for these requirements and the presented visualization approaches is provided in a user experiment, whose results are presented in detail in Chapter 11.

6.4 Related Visualization Approaches

In literature, there are other visualization approaches, e.g., for managing large business process models through views with reduced complexity [SPB05, BRB07]. However, these approaches focus on technical issues whereas issues related to the graphical representation of process artifacts (e.g., process tasks or data objects) are not addressed. However, there are a few approaches related to the *time-based* and *logic-based* visualization approaches:

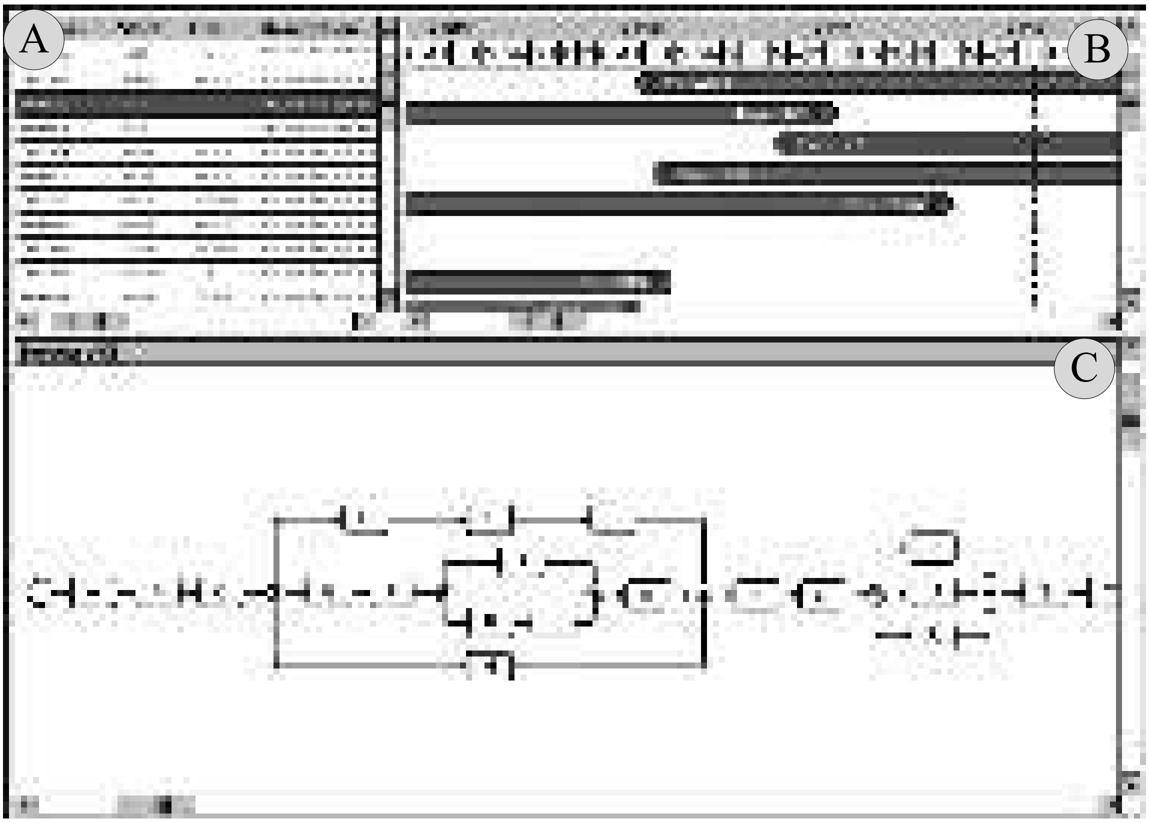


Figure 6.14: Visualizing process change documentation. [KFKF12].

Time-based visualization approaches are provided in [GRRv06, GRMR+08, KFKF12, KRM12, KWRM13], which focus on visualizing document change operations on process models (cf. Figure 6.14). For a better understanding, the authors combine different visualizations to present a common subject matter. The main purpose of this work is to expand the understanding on how to visualize process change information and to explore the concept of timeline visualization. Multiple visualizations are used in order to support the presentation of change information from different perspectives: *list visualization* (cf. Figure 6.14A), *timeline visualization* (cf. Figure 6.14B), and *process model visualization* (cf. Figure 6.14C). A similar approach, which is based on existing process mining techniques is presented in [GRRv06, GRMR+08]. More precisely, the latter approach allows for the visualization of process change logs. However, unlike our time-based visualization, this timeline visualization only provides information on change documentation of a given process model and not on the process model itself.

Lanz et al. [LKR13], in turn, present explicit visualization approaches for time-aware process models [LWR14]. The authors introduce characteristic time patterns for specifying the temporal perspective of process models [BLW+12]. In addition, they present an approach for transforming time-aware process models into enhanced Gantt Charts (cf. Figure 6.15). This approach focuses on temporal dependencies to be obeyed during process execution. For example, minimum and maximum task durations and minimum and maximum time lags between tasks must be considered to predict minimum, maximum, and average execution durations for entire processes [LPCR13, LR14]. Therefore, the authors introduce *eGantt*, an extended Gantt visualization that allows to visualizing the needed information. However, the presented visualization approaches are only based on the introduced time patterns of the approach [LWR14]. Our visualization approaches, in turn, are derived from a user's perspective, i.e., based on strict user requirements.

Other approaches from the area of temporal workflows [CGPP12, EPPR99, BWJ02] either rely on traditional process notations (e.g., BPMN) for visualizing time-aware processes or do not consider visualization issues at all.

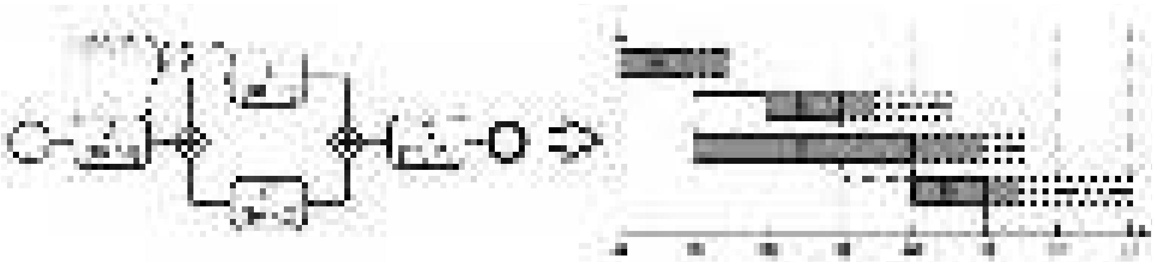


Figure 6.15: Time-aware process visualization. [LKR13].

There also exists research in the area of *logic-based* process visualization. An approach for visualizing event-driven process chains is presented in [MBN04]. In [SAtDL04] and [BEL+07] an approach for embedding process visualizations in larger enterprise architecture models is discussed. In turn, [WW96] describes an approach for a qualitative visualization of processes, i.e., using graph layout and focusing techniques. Another approach is introduced by the Poviado framework [BRB07, BBR06, Bob08]. The latter enables the flexible, configurable visualization of complex processes (cf. Figure 6.16). A template mechanism enables the support of different graphical process notations using different shapes or colors [BBR06].

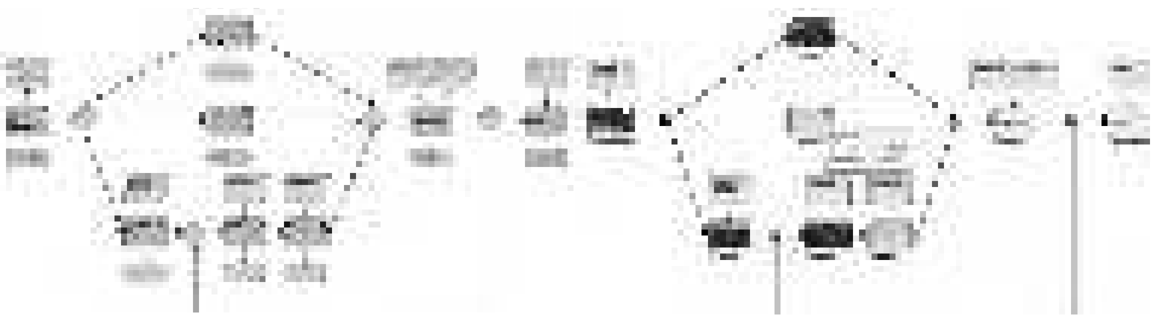


Figure 6.16: Two different visualization approaches. [BRB07].

Seyfang et al. [SKGM12] and Shneiderman et al. [Shn91] both make use of process hierarchies in order to efficiently visualize complex process models on small canvas. Their approach allows displaying very large process hierarchies in their entirety in a compact manner and thus facilitates the presentation of information on different semantic levels. Misue et al. [MY12] discuss the representation of detailed information about a single activity without loosing the overview on the global structure of an organization. Further Misue et al. provide a representation technique embedding charts which express activities into cells of a tree map. Schoenhage et al. [SvBE00] and Effinger [Eff12], in tun, investigate business visualization in 3D. They pick up a 2D visualization of a business process as a starting point, for which they subsequently provide a 3D visualization (cf Figure 6.17). With this approach, data visualization in multiple dimensions (e.g., past, present and simulated data) becomes possible. Note that we apply this idea in the context of BPMN3D to a certain extend as well.

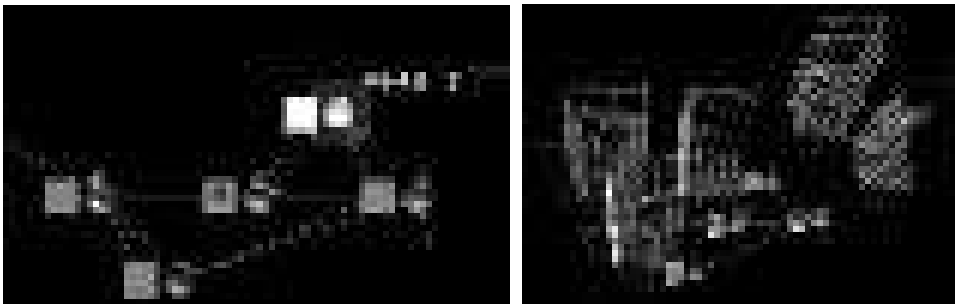


Figure 6.17: Process models visualized in a 3D environment [SvBE00].

The 3D visualization of process models is addressed by [PRJB13] and [BRW11] as well, which both enable collaborative process modeling in a 3D environment based on 3D avatars. In order to combine different views on one process model, Jablonski et al. [JG07] present a meta model, providing different visualizations for business process models applied to different perspectives, e.g., an organizational perspective or operational perspective.

6.5 Summary

Visualizing process model collections in a user-adequate manner constitutes a key factor for enterprises when being confronted with large and complex process model collections. However, existing visualization approaches based on common process modeling notations do not fully meet all requirements (e.g., regarding textual descriptions or temporal dependencies).

On one hand, this chapter introduced four visualization types for process model collections. A *time-based* visualization is used to focus on temporal aspects. A *logic-based* visualization, in turn, visualizes the execution logic of the tasks of a process model as known from BPMN. A *text-based* visualization allows for the documentation of detailed information. Finally, a *list-based* visualization type allows listing all objects from a navigation state. On the other hand, we investigated the *logic-based* visualization type in more detail. We presented four different approaches for alternative *logic-based* visualizations.

After having introduced the navigation concept as well as its formalization and visualization, the following chapter deals with the issue how the navigation framework is used by process participants. Further, it shows how the framework might be applied to different use cases.

7 Using the Navigation Space

This chapter illustrates the practical application of the ProNaVis concepts along the use cases introduced in Chapter 1. In particular, we show how ProNaVis contributes to evaluate and optimize navigation sequences.

Section 7.1 introduces preliminaries, whereas the use cases are presented in Sections 7.2 - 7.8. Section 7.9 provides a discussion. Finally, Section 7.10 summarizes the chapter.

7.1 Preliminaries

Figure 7.1 shows the navigation space we use for illustrating the use cases. It comprises seven semantic levels, seven geographic levels, and three visualization types (i.e., a time-based (1), a logic-based (2), and a text-based one (3)). Hence, there are 147 ($7 \times 7 \times 3$) different navigation states.

Each navigation state comprises a set of objects taken from the process space. Thereby, different navigation states might include various numbers of objects, depending to the according levels of the semantic and geographic dimensions. In certain cases, navigation states might comprise too many or too less objects. As visualizing these navigation states might confuse process participants, these *forbidden* navigation states should not be accessible during navigation. Therefore, the navigation space depicted in Figure 7.1 must be transferred to a basis model (BM), solely comprising *allowed* navigation states. Thereby, two kinds of forbidden navigation states are distinguished:

1. Navigation states with too few objects.
2. Navigation states with too many objects.

For a better understanding, we reconsider the process space from Chapter 4.3. For example, navigation state NS[1]$= (0, 5, 0)$ provides too little information, i.e., no information at all (cf. Figure 7.2a). According to the geographic level 5, the zoom is on a specific task (e.g., $T15$). At the same time, only the root process area should be considered within the navigation state (semantic level 0). From the users perspective, he zooms into a blank area somewhere within the root process area. This phenomenon is called “desert fog”. It describes a state, in which a user zooms to a small area on the screen that does not provide any information [JF98]. In turn, navigation state $(5, 0, 0)$ (cf. Figure 7.2b) provides too many objects, including all process elements (e.g., process tasks) on semantic level 5 across the entire process model collection (geographic level 0). However, this might lead to a visualization comprising hundreds or thousands of objects at the same time.

To identify forbidden navigation states we introduce *information density* – a metric indicating the number of objects in a navigation state. Various studies (e.g., [WLS98]) showed that information density significantly affects user navigation in applications (see *Principle of Constant Information Density* [FT94, TP66]). In general, the amount of information displayed should more or less remain constant

[1]Navigation state NS(S,G,V): S – semantic dimension; G – geographic dimension; V – visualization dimension

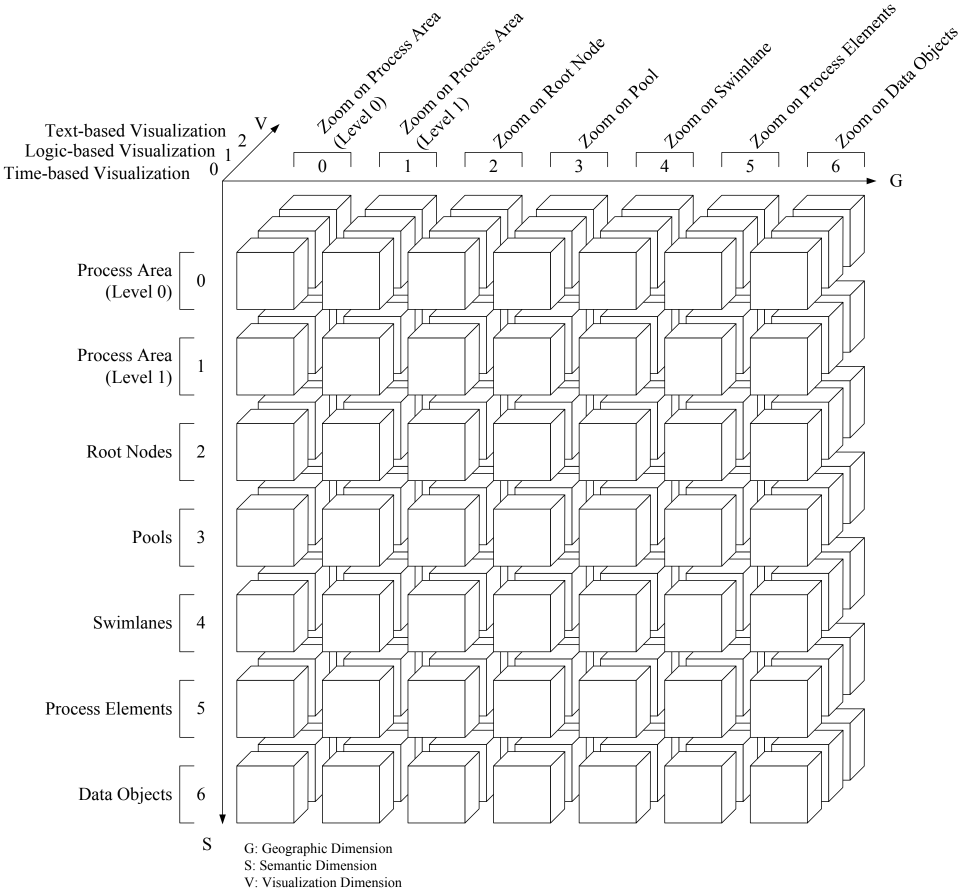


Figure 7.1: The navigation space used for illustration purpose.

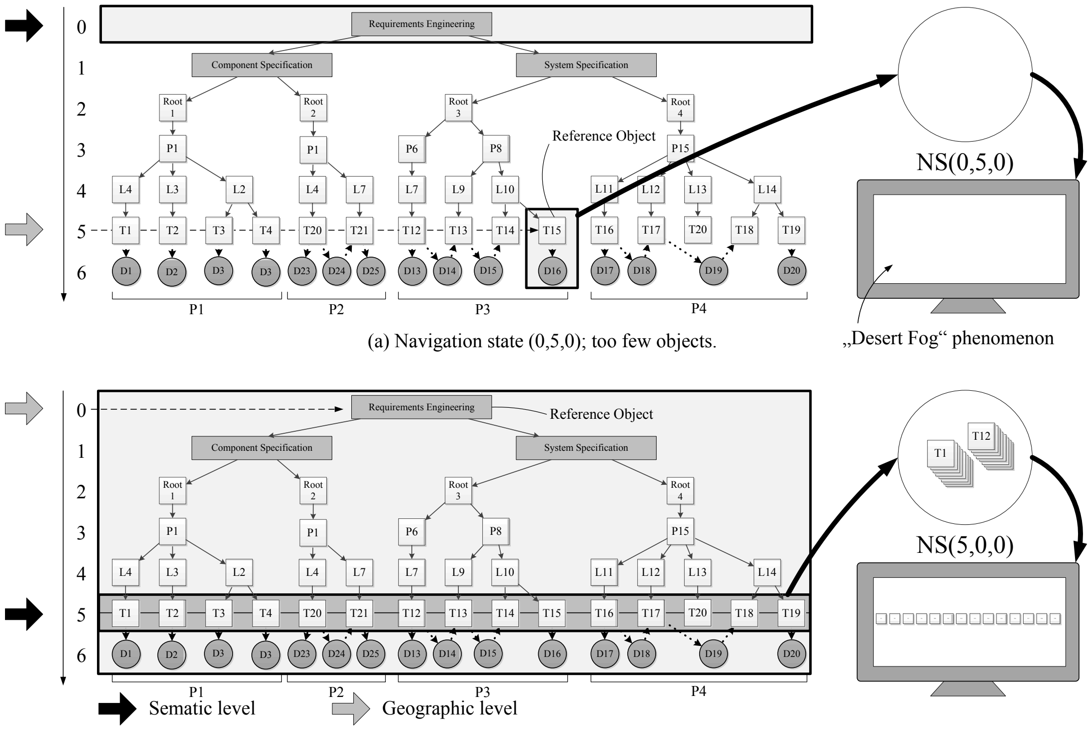
while panning and zooming. In turn, constant information density can be achieved either by visualizing objects at a higher level of detail when the user gets closer to them or by showing more objects when the user zooms into the canvas [FT94].

We consider information density when creating the BM. The geographic dimension indicates the size of the area provided on the screen, whereas the semantic dimension indicates the number of objects to be displayed. Note that the visualization dimension does not influence information density as it visualizes the same amount of information in different ways.

As the number of objects that may be displayed along the semantic dimension depends on the given process model collection and its corresponding process models, respectively, an exact calculation of the information density is difficult. Therefore, we use the different levels of the geographic and the semantic dimension as an indicator instead. Specifically, we assume that navigation states on a higher semantic level provide a higher number of objects. Likewise, we assume that a higher geographic level refers to a smaller area on the screen. Thus, a simplified density ratio dr can be calculated as follows:

$$dr = semantic\ level - geographic\ level \tag{7.1}$$

As example of a navigation state with a high dr, reconsider navigation state $NS(5, 0, 0)$ (cf. Figure 7.2b). Its dr corresponds to $5 = 5 - 0$, which indicates that too many objects are displayed on the screen, as



(a) Navigation state (0,5,0); too few objects.

(b) Navigation state (5,0,0); too many objects.

Figure 7.2: Examples for forbidden navigation states.

detailed information is shown on a big area on the screen (i.e., on an abstract geographic level). In turn, navigation state $NS(0, 5, 0)$ (cf. Figure 7.2a) has a negative *dr*, i.e., $dr = -5 = 0 - 5$, which indicates that there are no objects on the screen.

7.1.1 Handling Navigation States with too few Objects

In general, navigation state with a negative *dr* can be considered as providing too few objects on the screen. In such a case, the user might lose orientation [RB09] as he is zooming on objects (geographic dimension), not considered by the semantic dimension (i.e., *desert fog* phenomenon [JF98]). Consequently, all navigation states with a negative *dr* (cf. Figure 7.3) are removed from the navigation space.

We illustrate a navigation state with a negative *dr*, by considering a user zooming to a root node (i.e., geographic level 2). Then the displayed area on the screen would only cover (i.e., visualize) the root node itself as well as all objects nested within the root node. Thus, at least the root node object must be considered for visualization (i.e., at least semantic level 2) in order to be a valid navigation state. For example, navigation state $(2, 2, x)$ can be considered as valid navigation state, as focus is on the root node (geographic level 2). At the same time, the root node is considered for visualization (semantic level 2). As a result, the root node is visualized on the screen. In turn, if the user had further zoomed to a particular swimlane (i.e., geographic level 4), he would have zoomed to a blank area on the screen, as swimlanes would have not been considered on the semantic dimension ($NS(2, 4, 0)$ with $dr = -2$).

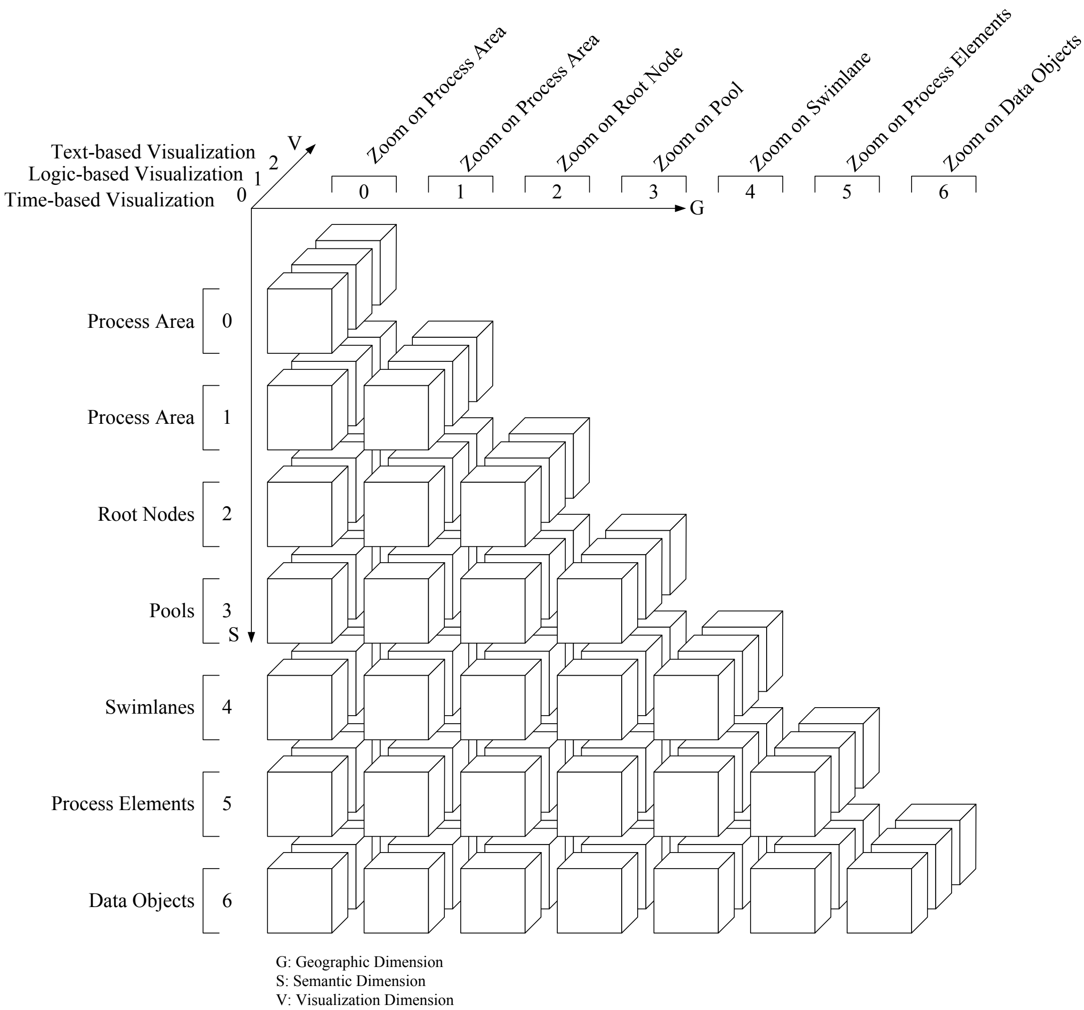


Figure 7.3: The reduced navigation space.

7.1.2 Handling Navigation States with too many Objects

Navigation states comprising too many objects confuse users as well. However, removing respective navigation states might cause a loss of relevant information. Therefore, this type of navigation states is treated differently by not removing them. Instead, they are marked in the BM.

Marked navigation states can be considered as intermediate navigation states in a navigation sequence, supporting the maintenance of the navigational context for process participants. In particular, they allow for 1-dimensional process interactions. This kind of interaction facilitates recognizing similar objects corresponding to different navigation states in a navigation sequence. In turn, this fosters the user's coherence between different representations of objects [SJB07]. First, an object becomes enlarged when the user navigates along the geographic dimension. Second, an object is presented in greater detail when the user navigates along the semantic dimension. Third, an object is visualized in different ways, when the user navigates along the visualization dimension. Indeed, objects change their representation in a navigation sequence. However, the changes made should be limited when navigating between two states. Only then the user will always be able to recognize the objects along a navigation sequence. In summary, 1-dimensional interactions constitute the easiest way to navigate within a navigation space. In particular, this fosters coherence as well as a decrease of the user's the mental load [SB06].

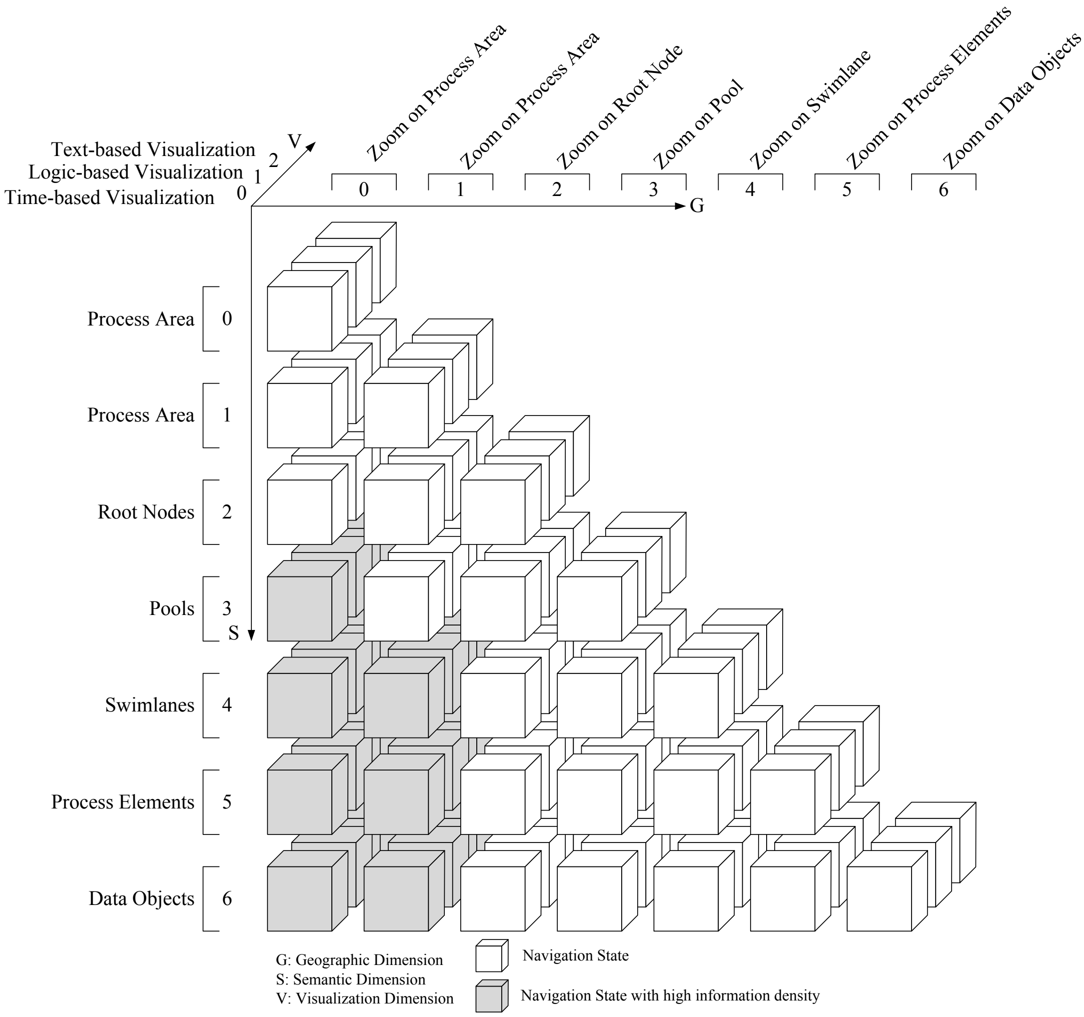


Figure 7.4: Applying the basis model used in our use cases.

As *dr* solely indicates information density, navigation states should be marked manually, which could be accomplished by, for example, a process modeler. Note that certain navigation states on the semantic levels of swimlanes (4), process elements (5), and data objects (6) will not be marked, even if the information density indicates a high number of objects (i.e., navigation states (5,2,x), (6,2,x), and (6,3,x) with x being any level of the visualization dimension). These navigation states address the visualization of an entire process model by combining various navigation states on different semantic levels (cf. Section 4.4.4). Hence, a higher information density can be accepted for selected navigation states (e.g.,$dr > 3$) when considering the visualization of entire process models being more important than dismissing navigation states based on their high information density.

The BM resulting after the removal of forbidden navigation states is shown in Figure 7.4. This BM comprises 84 navigation states.

7.1.3 Methodology

In the following, we refer to the use cases presented in Chapter 1 (cf. Table 7.1). In particular, we investigate navigation sequences required to realize the use cases.

#	Name	Title
1	Project Manager	A project manager needs an overview on the entire process model collection.
2	Business Unit Manager	A business unit manager needs information about the process models refering to his business unit.
3	Requirements Engineer	A requirements engineer needs detailed descriptions of a certain process task.
4	New Employee	A new employee wants to get an overview of all process steps, he must perform.
5	Quality Manager	A quality manager shall ensure the quality of all documents corresponding to a process model collection.
6	Quality Engineer	A quality engineer shall identify the process tasks related to a certain deadline.
7	Test Engineer	A test engineer needs access to process models from different process areas.

Table 7.1: Considered use cases.

For describing these navigation sequences, we apply the structure proposed by Fowler [Fow04]. In the following subsections, each use case is structured as follows:

- **Title:** The use case is associated with a title.
- **Description:** The main goal of the use case is briefly described (based on the use case descriptions introduced in Chapter 1).
- **Main Success Scenario (Navigation Sequence):** Based on the given BM, we present a default navigation sequence for the use case. This sequence consists of 1-dimensional process interactions and can be directly derived from the use case descriptions. Note that $NS(0,0,0)$ is used as starting state for the navigation sequence.

  We structure the navigation sequence along the three navigation dimensions. In the first step, we illustrate which navigation steps are required with respect to the semantic dimension. In the second step, navigation steps required for the geographic dimension are considered. The third step deals with navigation steps required in the context of the visualization dimension. Finally, in a fourth step, we show how filter mechanisms can be applied to better support the use case.
- **Analysis:** We analyze the default navigation sequence by calculating the linear distance $DIST$ between the start and the end point of the sequence. Further, we calculate the length of the given navigation sequence, i.e., $NAVDIST_{NavSeq}$ (cf. Section 5.3).
- **Improvement:** We illustrate how the default navigation sequence can be replaced by a better alternative, e.g., considering multi-dimensional process interactions (cf. Section 5.4). We calculate the effectivity Eff for both the old and the new navigation sequence, and compare them with each other.

7.2 Use Case 1

Title: A project manager needs an overview on the entire process model collection.

Description: A project manager wants to have a quick look on all process models within the process model collection in order to determine the already finished, the currently running, or the not yet started processes. Note that this is helpful to estimate overall project progress. In this context, the manager needs illustrating information on temporal dependencies between different process models across the entire process model collection.

Main Success Scenario (Navigation Sequence):

- Step 1 (Semantic Dimension): As the project manager wants to see different process models at a glance, as detail level he chooses process model root nodes, which represent entire process models (cf. Section 4.3.1).
- Step 2 (Geographic Dimension): To get an overview, the project manager sets the geographic dimension to a low level. Thus, the entire process model collection shall be visible on the screen.
- Step 3 (Visualization Dimension): To intuitively identify temporal dependencies, a time-based visualization is needed, i.e., process objects shall be displayed as graphical representations.
- Step 4 (Filter Settings): No filters are required, as all objects are of interest.

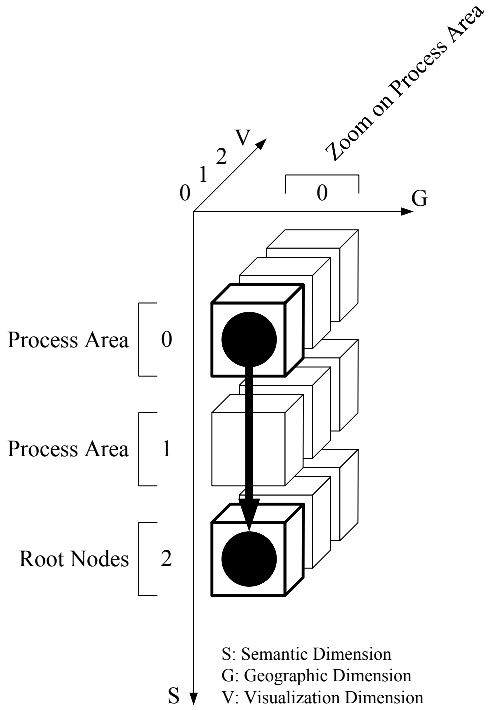


Figure 7.5: Use Case 1 - The navigation sequence within the navigation space.

Figure 7.5 illustrates the navigation sequence of the project manager following the main success scenario. Initial navigation state NS_0 is $(0, 0, 0)$. In this case, the navigation sequence is simple. The semantic dimension has to be adjusted to the level of root nodes (semantic level 2). As the project manager wants to see all root nodes, the geographic level remains 0, i.e., the default value is kept, i.e., the geographic dimension is unchanged compared to the default navigation state. Since the time-based visualization also corresponds to the default level, there is also no need to change the visualization dimension.

Analysis: Based on starting state $NS_0 = \{0,0,0\}$, the project manager's destination is $NS_n = (2,0,0)$. The coresponding navigation sequence can be defined as follows:

$$NavSeq = (i_1, i_2) = \left((1,0,0)^T, (1,0,0)^T\right) \tag{7.2}$$

Process navigation along this sequence can be defined as follows:

$$NS_n = NS_0 + i_1 + i_2 = (0,0,0)^T + (1,0,0)^T + (1,0,0)^T = (2,0,0)^T \tag{7.3}$$

Since $(2,0,0) \in BM$ holds, the navigation sequence from $NS_0 = (0,0,0)$ to $NS_n = (2,0,0)$ is an allowed one (cf. Definition 5.9 from Chapter 5).

Improvement: Since the project manager solely adjusts a single navigation dimension, effectiveness of the navigation sequence corresponds to 100%. In turn, distance $DIST((0,0,0),(2,0,0))$ corresponds to the length of the navigation sequence.

$$Eff(P1, P2, NavSeq) = \frac{DIST(P_1, P_2)}{NAVDIST(NavSeq)} = \frac{\sqrt{2^2 + 0^2 + 0^2}}{\sqrt{2^2 + 0^2 + 0^2}} = 100\% \tag{7.4}$$

In this particular use case, therefore, the navigation sequence cannot be improved. A wireframe, visualizing navigation state $(2,0,0)$, is depicted in Figure 7.6.

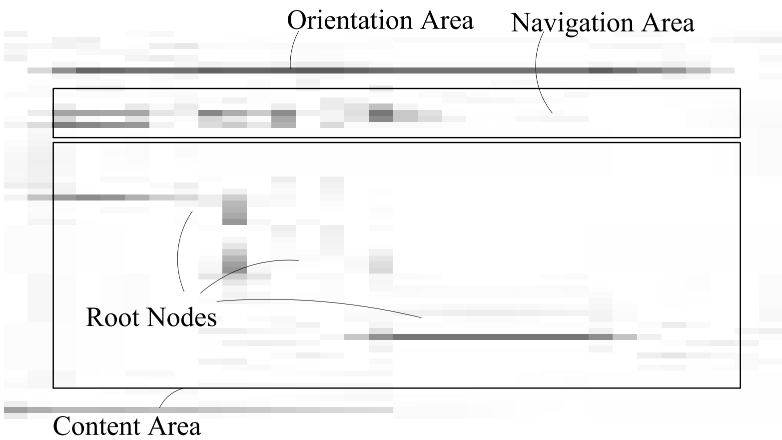


Figure 7.6: Use Case 1 - Wireframe of the visualized navigation state $(2,0,0)$.

As the semantic dimension focuses on root nodes, the latter are presented to the user in the content area of the presented wireframe. Thereby, each root node corresponds to an abstract representation of a process model. The time-based visualization visualizes each root node as rectangular box. The length of a box corresponds to the duration of the underlying process model. Thus, temporal dependencies can be quickly identified using the orientation area, where a timeline indicates a temporal scale. In turn, the navigation area indicates the current level of detail in the semantic dimension based on a simple breadcrumb navigation concept.

7.3 Use Case 2

Title: A business unit manager needs information about the process models refering to his business unit.

Description: A business unit manager is responsible for process models related to a specific process area. Unlike the project manager, he needs more detailed information about single process models and their corresponding process objects (e.g., swimlanes or tasks).

Main Success Scenario (Navigation Sequence):

- Step 1 (Semantic Dimension): The business unit manager is interested in roles and tasks of a certain process model in order to monitor the execution of corresponding instances. Therefore, the semantic level of swimlanes and tasks is of interest.
- Step 2 (Geographic Dimension): The business unit manager is only interested in a certain area of the process model collection. Therefore, he may use the geographic dimension to focus a certain area.
- Step 3 (Visualization Dimension): In the given use case, the most suitable visualization is a logic-based visualization, i.e., relying on process swimlanes, process tasks, and sequence flow.
- Step 4 (Filter Settings): One might filter the resulting navigation state for a certain role (i.e., name of a role), if the business unit manager wants to monitor tasks of a specific employee.

Following the main success scenario, the business unit manager may apply the navigation sequence depicted in Figure 7.7. Thus, the semantic dimension is set to level 5; at the same time the geographic level is set to 2, i.e., focus is on a single root node, and the visualization has to be set from a time- to a logic-based visualization.

Analysis: Based on the concepts of the navigation space, we can calculate the distance between start state $(0,0,0)$ and end state $(5,2,1)$.

$$DIST((0,0,0),(5,2,1)) = \sqrt{5^2+2^2+1^2} \approx 5{,}48 \tag{7.5}$$

Based on the main success scenario, the user may follow a navigation sequence starting at $NS_0 = (0,0,0)$, i.e., $Nav_a = (i_1,i_2,i_3) = \big((5,0,0)^T,(0,2,0)^T,(0,0,1)^T\big)$. Based on this, the distance of this navigation sequence Nav_a can be calculated:

$$NAVDIST(Nav_a) = \sum_{i=0}^{n-1} DIST(P_{i+1},P_i) = \sum_{0}^{7} 1 = 8 \tag{7.6}$$

Improvement: In order to maintain the user's coherence during navigation, we present an alternative navigation sequence applying 2-dimensional process interactions. More precisely, we recommend the following navigation sequence Nav_b:

$$Nav_b = (i_1,i_2,i_3,i_4) = \big((1,1,0)^T,(1,1,0)^T,(3,0,0)^T,(1,0,1)\big) \tag{7.7}$$

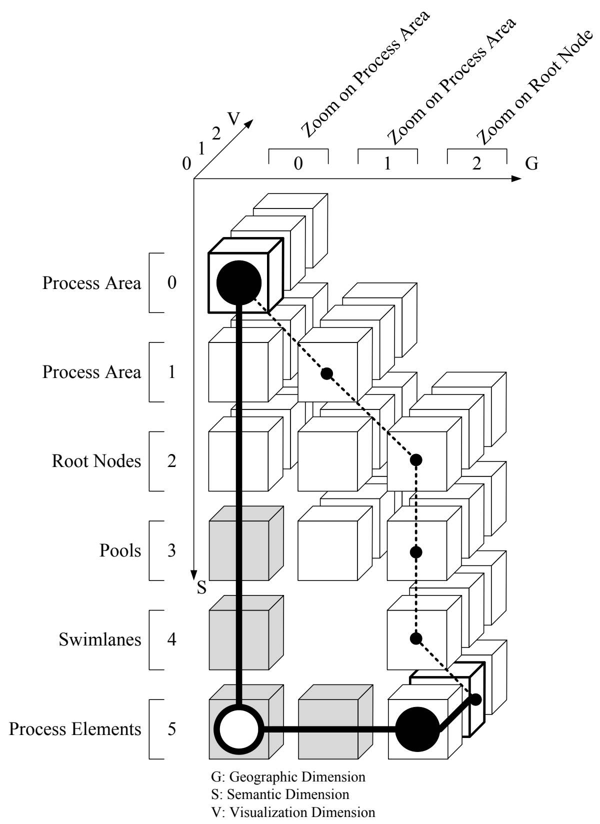


Figure 7.7: Use Case 2 - The navigation sequence within the navigation space.

Its distance can be calculated as follows:

$$NAVDIST(Nav_b) = \sum_{i=0}^{n-1} DIST(P_{i+1}, P_i) = \sqrt{2} + \sqrt{2} + 3 + \sqrt{2} \approx 7{,}24 \tag{7.8}$$

Picking up Nav_a and Nav_b, we can calculate the navigation effectiveness.

$$Eff(Start, End, Nav_a) = \frac{5{,}48}{8} \approx 68{,}5\% \tag{7.9a}$$

$$Eff(Start, End, Nav_b) = \frac{5{,}48}{7{,}24} \approx 75{,}69\% \tag{7.9b}$$

As can be seen, the alternative navigation sequence which solely comprises 2-dimensional interactions is more effective than the 1-dimensional navigation sequence applied to the main success scenario.

Finally, Figure 7.8 depicts a wireframe, visualizing the final navigation state $(5, 2, 1)$.

The navigation area indicates the semantic levels selected for the combined visualization (cf. Chapter 6) of navigation states $(5, 2, 1)$ and $(4, 2, 1)$ (swimlanes and process elements). Therefore, swimlanes and their corresponding process elements are visualized within the content area.

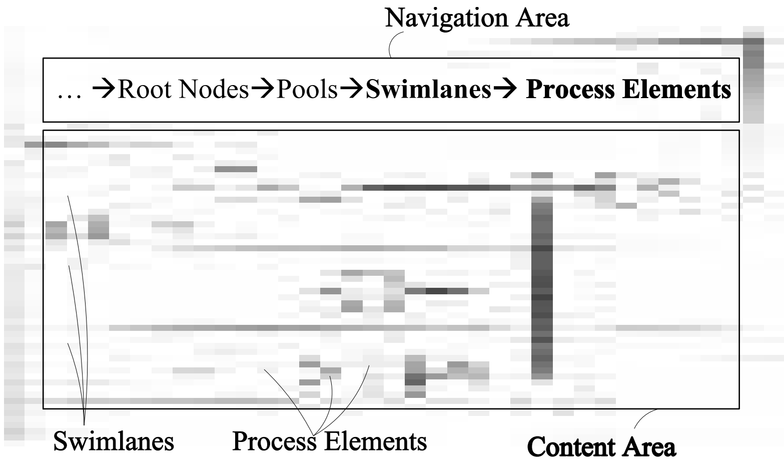


Figure 7.8: Use Case 2 - Wireframe of the visualized navigation state $(5, 2, 1)$.

7.4 Use Case 3

Title: A requirements engineer needs detailed descriptions of a certain process task.

Description: A requirements engineer writes specification documents, e.g., for an anti-lock breaking system (ABS) control unit. In this context, he must execute several process tasks of the respective process. In this context, he requires technical instructions such as specification guidelines, templates, and checklists when performing specific tasks.

Main Success Scenario (Navigation Sequence):

- Step 1 (Semantic Dimension): Detailed information on data objects related to a specific process task is required. Therefore, the detail level of data objects is selected.
- Step 2 (Geographic Dimension): Since the requirements engineer wants to work on a particular process tasks, the geographic dimension shall focus on this task solely.
- Step 3 (Visualization Dimension): The visualization shall provide detailed task descriptions as well as access to related data objects (i.e., documents).
- Step 4 (Filter Settings): No filters are required.

Figure 7.9 shows the navigation sequence corresponding to the main success scenario. As the requirements engineer needs detailed information on data objects, the semantic level is set to 6. The engineer is interested in data objects corresponding to a particular process task, i.e., the geographic level is set to 5. As the engineer needs detailed task descriptions, the visualization shall be text-based (2). Accordingly, as desired navigation state we obtain $NS_n = (6, 5, 2)$.

Analysis: Based on the described navigation space concept (cf. Chapter 4), we can calculate the distance between start state $(0, 0, 0)$ and end state $(6, 5, 2)$.

$$DIST((0,0,0),(6,5,2)) = \sqrt{6^2 + 5^2 + 2^2} \approx 8{,}06 \tag{7.10}$$

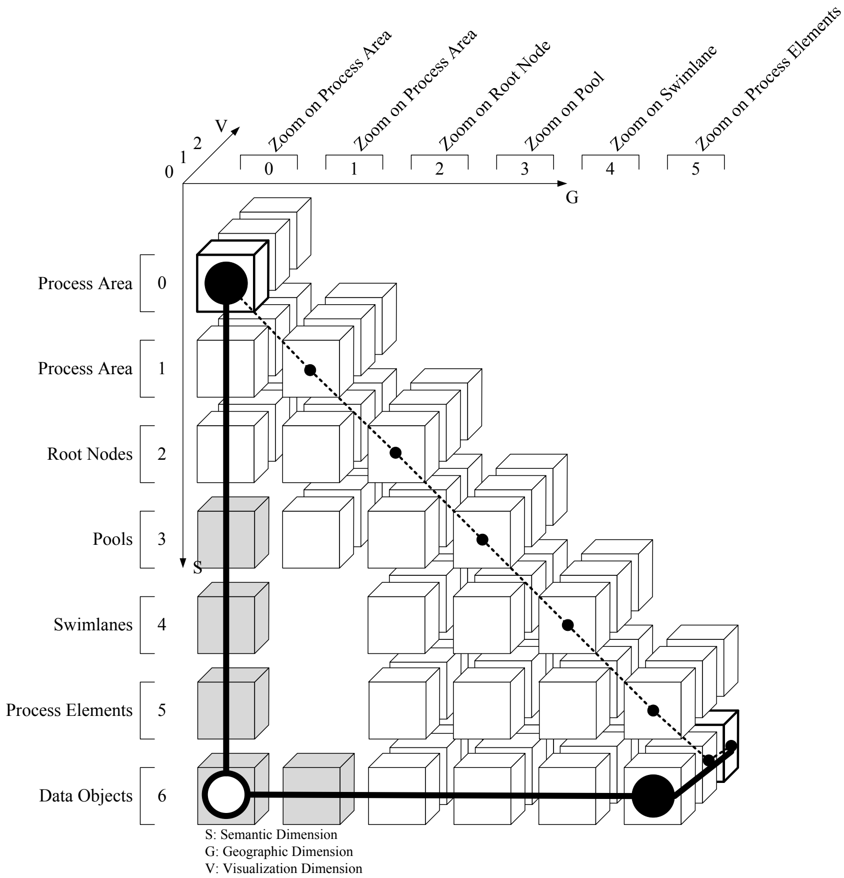


Figure 7.9: Use Case 3 - The navigation sequence within the navigation space.

Based on starting state $NS_0 = (0, 0, 0)$, the navigation sequence Nav_a following the main success scenario can be defined as follows.

$$Nav_a = (i_1, i_2, i_3) = \left((6, 0, 0)^T, (0, 5, 0)^T, (0, 0, 2)^T\right) \tag{7.11}$$

The length of this navigation sequence can be calculated as follows:

$$NAVDIST(Nav_a) = \sum_{i=0}^{n-1} DIST(P_{i+1}, P_i) = \sum_{0}^{12} 1 = 13 \tag{7.12}$$

Improvement: In the following, an alternative navigation sequence Nav_b, which also includes 2-dimensional process interactions, is described. Its length is calculated afterwards:

$$\begin{gathered} Nav_b = (i_1, i_2, i_3, i_4, i_5, i_5, i_6, i_7) = \\ \left((1, 1, 0)^T, (1, 1, 0)^T, (1, 1, 0)^T, (1, 1, 0)^T, (1, 1, 0)^T, (0, 1, 1)^T, (0, 0, 1)^T\right) \end{gathered} \tag{7.13}$$

$$DIST(Nav_b) = \sum_{i=0}^{n-1} DIST(P_{i+1}, P_i) = 6 * (\sqrt{2}) + 1 \approx 9{,}48 \tag{7.14}$$

We further calculate and compare the effectiveness of the two navigation sequences presented.

$$Eff(Start, End, Nav_a) = \frac{8{,}06}{13} \approx 62\% \tag{7.15a}$$

$$Eff(Start, End, Nav_b) = \frac{8{,}06}{9{,}48} \approx 85{,}02\% \tag{7.15b}$$

As can be seen, Nav_b turns out to be more effective compared to Nav_a. Figure 7.10 shows a wireframe depicting the calculated navigation state. Thereby, a text-based visualization is used, i.e., detailed textual task descriptions are provided in the content area. In turn, related data objects are accessible in a separate area.

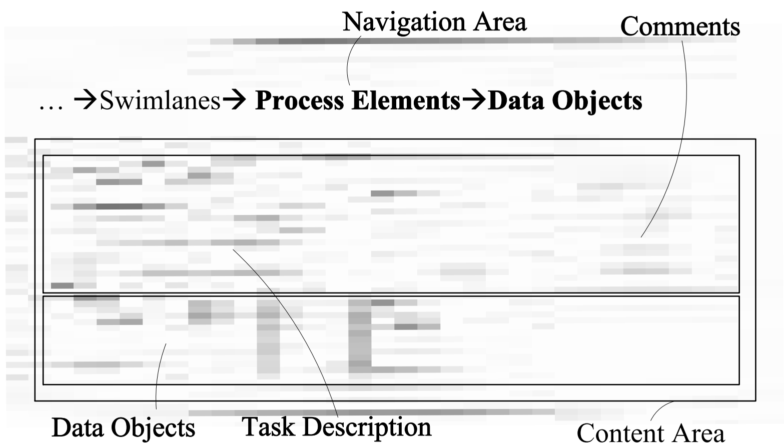


Figure 7.10: Use Case 3 - Wireframe of the visualized navigation state $(6, 5, 2)$.

7.5 Use Case 4

Title: A new employee wants to get an overview of all process steps, he must perform.

Description: A new employee (e.g., a requirements engineer) shall obtain an overview on all tasks he must perform to enable him to properly prepare each task. In this context, he needs a quick overview on all tasks from process area of requirements engineering for which he is responsible.

Main Success Scenario (Navigation Sequence):

- Step 1 (Semantic Dimension): The process participant wants to identify single process tasks. Accordingly, the semantic level of process tasks (i.e., process elements) must be selected.
- Step 2 (Geographic Dimension): In order to get an overview on a specific process area (e.g., requirements engineering), the focus of the geographic dimension needs to be on a process area.

- Step 3 (Visualization Dimension): As the requirements engineer is interested in quickly identifying process tasks, a graphical representation of the latter is of interest, i.e., a logic-based visualization (or alternatively a time-based one).
- Step 4 (Filter Settings): Only tasks assigned to the requirements engineer shall be visualized.

Regarding the main success scenario, the applied navigation sequence can be defined as shown in Figure 7.11. Accordingly, the desired navigation state corresponds to $NS_n = (5, 1, 0)$. In this context, NS_n exhibits high information density. Note that displaying process elements on a large area (i.e., on a process area in the given case) results in a high amount of information to be displayed, i.e., density ratio dr would be very high. Therefore, this navigation state shall only be reasonable, when a filter criterion is applied that reduces the number of displayed objects. In this context, filtering process tasks assigned to a specific role (i.e., swimlane) is useful.

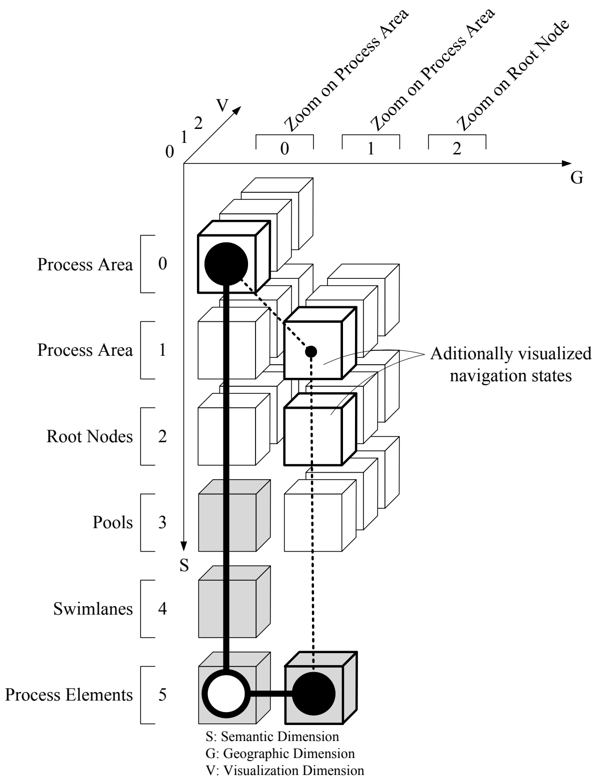


Figure 7.11: Use Case 4 - The navigation path within the navigation space.

Analysis: Based on the navigation space concepts, we can calculate the distance between start state $(0, 0, 0)$ and end state $(5, 1, 0)$:

$$DIST((0, 0, 0), (5, 1, 0)) = \sqrt{5^2 + 1^2 + 0^2} \approx 5{,}09 \tag{7.16}$$

Based on starting state $NS_0 = (0, 0, 0)$, the navigation sequence corresponding to the main success scenario is:

$$Nav_a = (i_1, i_2) = \left((5, 0, 0)^T, (0, 1, 0)^T\right) \tag{7.17}$$

Further, the length of navigation sequence Nav_a can be calculated as follows.

$$NAVDIST(Nav_a) = \sum_{i=0}^{n-1} DIST(P_{i+1}, P_i) = \sum_{0}^{5} 1 = 6 \tag{7.18}$$

Improvement: An alternative navigation sequence comprising 2-dimensional process interactions is as follows:

$$Nav_b = (i_1, i_2, i_3, i_4, i_5) = \left((1,1,0)^T, (1,0,0)^T, (1,0,0)^T, (1,0,0)^T, (1,0,0)^T\right) \tag{7.19}$$

The distance of this navigation sequence is calculated as follows:

$$NAVDIST(Nav_b) = \sum_{i=0}^{n-1} DIST(P_{i+1}, P_i) = (\sqrt{2}) + 4 \approx 5{,}41 \tag{7.20}$$

The effectiveness of the presented navigation sequences can be calculated as follows:

$$Eff(Start, End, Nav_a) = \frac{5{,}09}{6} \approx 84{,}83\% \tag{7.21a}$$

$$Eff(Start, End, Nav_b) = \frac{5{,}09}{5{,}41} \approx 94{,}08\% \tag{7.21b}$$

Navigation sequence Nav_b provides more effective process navigation compared to navigation sequence Nav_a.

The desired navigation state $(5, 1, 0)$ is illustrated in the wireframe shown in Figure 7.12. In order to increase user orientation, the visualization is combined with navigation states $(1, 1, 0)$ and $(2, 1, 0)$ (cf. Section 4.4.4). Different objects are displayed as nested rectangles in a logic-based visualization. In particular, the user is enabled to figure out, which process tasks are assigned to which process model and process area respectively. Note that the presented wireframe already shows the filtered visualization on process tasks. These tasks are assigned to the role of the new employee, therefore only few objects are displayed.

7.6 Use Case 5

Title: A quality manager shall ensure the quality of all documents corresponding to a process model collection.

Description: A quality manager is involved in different processes across process areas. He is responsible for the overall quality of process execution, e.g., the quality of documents such as specification documents, test documents, or review documents. As these documents are created by different processes in various process areas, the quality manager needs a quick overview on different process models and process tasks across the entire process model collection.

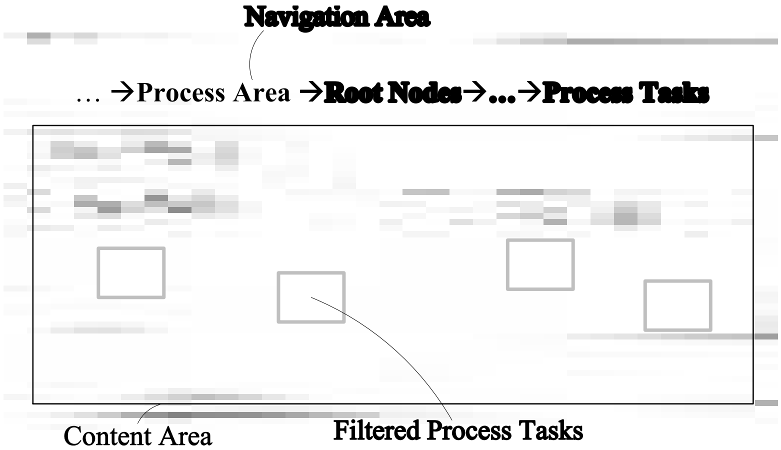


Figure 7.12: Use Case 4 - Wireframe of the visualized navigation state $(5, 1, 0)$.

Main Success Scenario (Navigation Sequence):

- Step 1 (Semantic Dimension): The quality manager must access single data objects, e.g., documents assigned to process tasks. Therefore, as detail level the level of detail of process tasks is chosen.
- Step 2 (Geographic Dimension): The quality manager is involved in process tasks spread over the entire process model collection, i.e., focus is on the entire process model collection.
- Step 3 (Visualization Dimension): To identify data objects and related process tasks, a logic-based visualization is chosen.
- Step 4 (Filter Settings): The presented process tasks have to be filtered for for the assigned role (i.e., quality manager).

The main success scenario describes a navigation sequence ending in navigation state $NS_n = (5, 0, 1)$. Again, NS_n exhibits high information density. Thus, filter criteria must be applied to reduce the amount of information displayed. The navigation sequence is shown in Figure 7.13.

Analysis: Based on the navigation space concepts, we can calculate the distance between start state $(0, 0, 0)$ and end state $(5, 0, 1)$.

$$DIST((0, 0, 0), (5, 0, 1)) = \sqrt{5^2 + 0^2 + 1^2} \approx 5{,}09 \tag{7.22}$$

Tackling start state $NS_0 = (0, 0, 0)$, the navigation sequence following the main success scenario and its length can be calculated as follows:

$$Nav_a = (i_1, i_2) = \big((5, 0, 0)^T, (0, 0, 1)^T\big) \tag{7.23}$$

$$NAVDIST(Nav_a) = \sum_{i=0}^{n-1} DIST(P_{i+1}, P_i) = \sum_{0}^{5} 1 = 6 \tag{7.24}$$

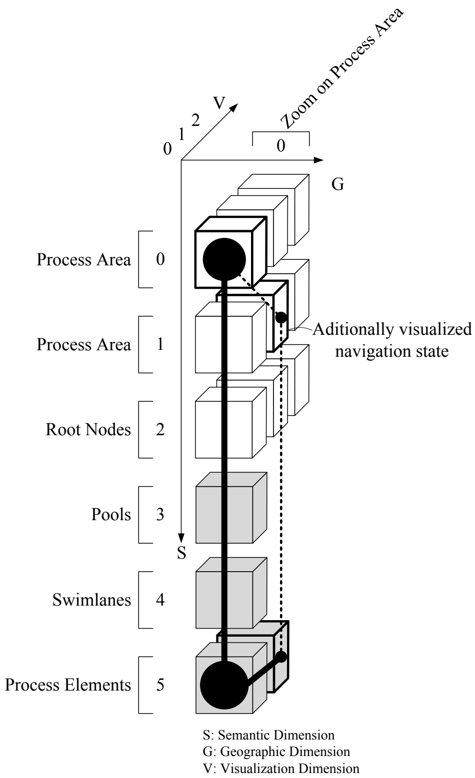


Figure 7.13: Use Case 5 - The navigation path within the navigation space.

Improvement: When also considering 2-dimensional process interactions, the following alternative navigation sequence can be applied:

$$Nav_b = (i_1, i_2, i_3, i_4, i_5) = \left((1,0,1)^T, (1,0,0)^T, (1,0,0)^T, (1,0,0)^T, (1,0,0)^T\right) \tag{7.25}$$

$$NAVDIST(Nav_b) = \sum_{i=0}^{n-1} DIST(P_{i+1}, P_i) = (\sqrt{2}) + 4 \approx 5{,}41 \tag{7.26}$$

The effectiveness of both navigation sequences can be calculated as follows:

$$Eff(Start, End, Nav_a) = \frac{5{,}09}{6} \approx 84{,}83\% \tag{7.27a}$$

$$Eff(Start, End, Nav_b) = \frac{5{,}09}{5{,}41} \approx 94{,}08\% \tag{7.27b}$$

As can be seen, navigation sequence Nav_b is more effective compared to navigation sequence Nav_a.

As can be further seen from Figure 7.14, the wireframe provides process areas as well as process tasks in a logic-based visualization, i.e., a combined visualization of navigation states $(5,0,1)$ and $(1,0,1)$. Finally, the visualized process tasks have already been filtered for process tasks assigned to the role quality manager.

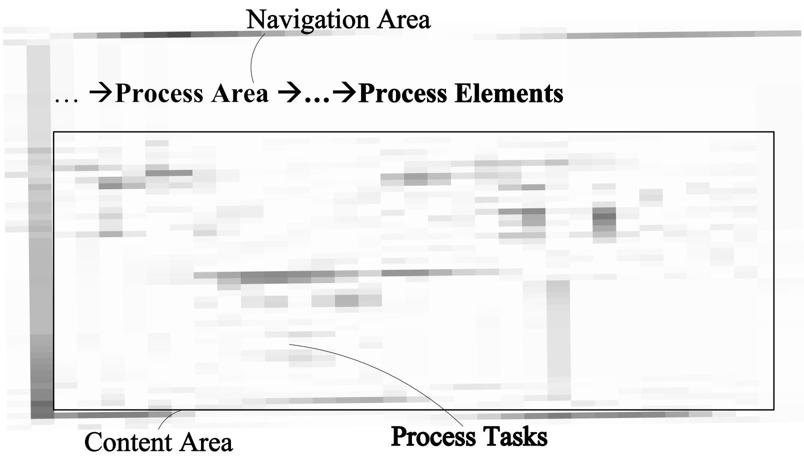


Figure 7.14: Use Case 5 - Wireframe of the visualized navigation state $(5, 0, 1)$.

7.7 Use Case 6

Title: A quality engineer shall identify the process tasks related to a certain deadline.

Description: A quality engineer must assure that business process results fulfil predefined quality standards. Unlike the quality manager, the quality engineer must consider certain deadlines. In particular, a quality engineer must check all resulting documents necessary to pass a certain deadline. Accordingly, he needs information about all process tasks related to the creation of a document and to be completed until a specific deadline.

Main Success Scenario (Navigation Sequence):

- Step 1 (Semantic Dimension): As the quality engineer must check process tasks, the semantic dimension must be set to the detail level of process tasks.
- Step 2 (Geographic Dimension): The geographic focus shall be on process areas, to enable an overview on process tasks aligned to a deadline.
- Step 3 (Visualization Dimension): To visualize process tasks referring to a certain deadline, a time-based visualization is of need.
- Step 4 (Filter Settings): Only those process tasks shall be visualized, which are associated with a deadline (i.e., tasks to be completed until a certain point in time).

Figure 7.15 describes the navigation sequence for the main success scenario. The quality manager needs an overview on multiple process models of a particular process area. Then, he must identify process tasks in this area. Furthermore, the desired navigation state $NS_n = (5, 1, 0)$ shows high information density. Thus, filter criteria should be applied to reduce the number of displayed objects. In the given case, a temporal filter can be applied, solely visualizing those process tasks finished unitl a certain deadline.

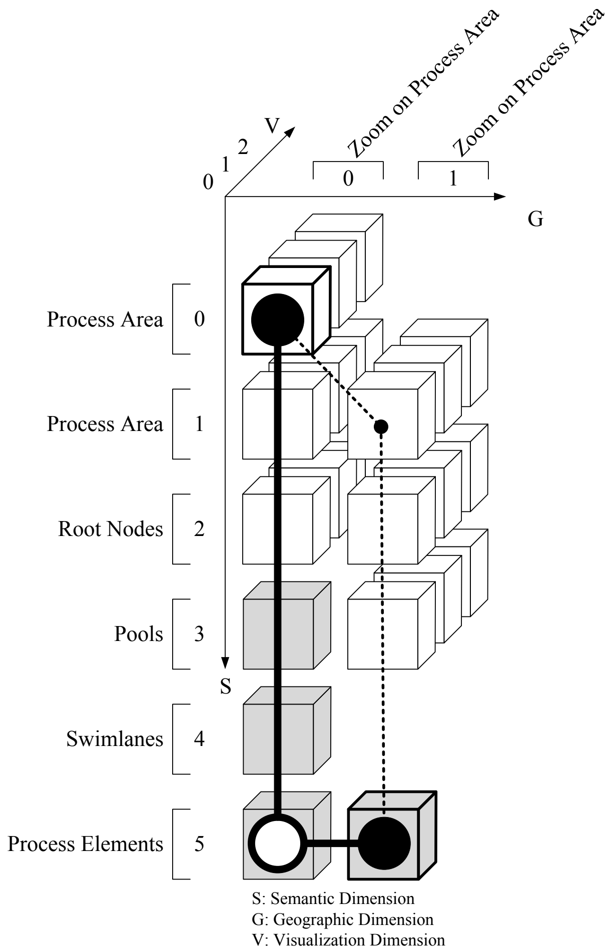


Figure 7.15: Use Case 6 - The navigation path within the navigation space.

Analysis: Based on the navigation space concepts, we can calculate the distance between start state $(0,0,0)$ and end state $(5,1,0)$.

$$DIST((0,0,0),(5,1,0)) = \sqrt{5^2+1^2+0^2} \approx 5{,}09 \tag{7.28}$$

Tackling start state $NS_0 = (0,0,0)$, the navigation sequence following the main success scenario can be calculated as follows:

$$Nav_a = (i_1, i_2) = \left((5,0,0)^T, (0,1,0)^T\right) \tag{7.29}$$

Accordingly, we can calculate the length of Nav_a:

$$NAVDIST(Nav_a) = \sum_{i=0}^{n-1} DIST(P_{i+1}, P_i) = \sum_{0}^{5} 1 = 6 \tag{7.30}$$

Improvement: An alternative navigation sequence, which also considers 2-dimensional process interactions is defined in the following:

$$Nav_b = (i_1, i_2, i_3, i_4, i_5) = \big((1,1,0)^T, (1,0,0)^T, (1,0,0)^T, (1,0,0)^T, (1,0,0)^T\big) \tag{7.31}$$

$$NAVDIST(Nav_b) = \sum_{i=0}^{n-1} DIST(P_{i+1}, P_i) = (\sqrt{2}) + 4 \approx 5{,}41 \tag{7.32}$$

The effectiveness of both navigation sequences can be calculated as follows:

$$Eff(Start, End, Nav_a) = \frac{5{,}09}{6} \approx 84{,}83\% \tag{7.33a}$$

$$Eff(Start, End, Nav_b) = \frac{5{,}09}{5{,}41} \approx 94{,}08\% \tag{7.33b}$$

The wireframe depicted in Figure 7.16 shows the visualization of navigation state $(5, 1, 0)$. Note that use case 6 exhibits the same navigation state as seen in the context of use case 4. However, applying a different filter results in a different visualization.

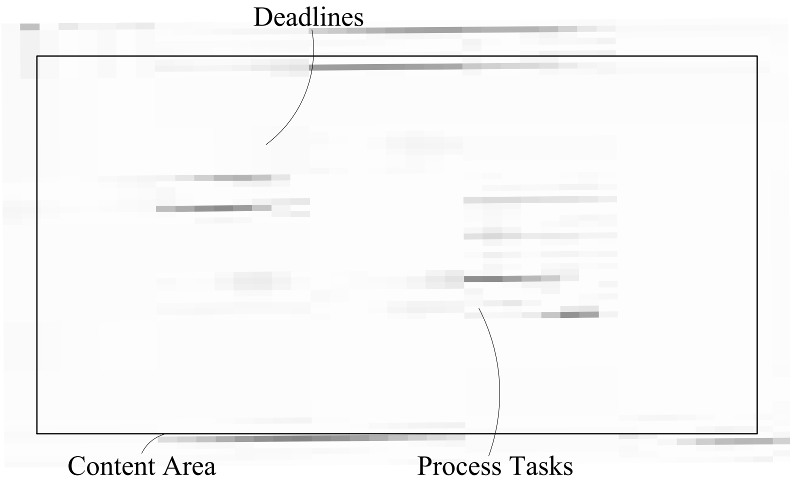


Figure 7.16: Use Case 6 - Wireframe of the visualized navigation state $(5, 1, 0)$.

7.8 Use Case 7

Title: A test engineer needs access to process models from different process areas.

Description: A test engineer shall prepare tests for a developed car control unit. The respective process corresponds to a process area dealing with the topic “testing”. Furthermore, this task depends on results from the previous process area dealing with “implementation”. In particular, the test engineer needs to know what functions are implemented in the car component in order to prepare the test cases.

Main Success Scenario (Navigation Sequence):

- Step 1 (Semantic Dimension): The test engineer is interested in identifying process models that deliver input for the testing tasks. Accordingly, as detail level he first selects the level of detail of root nodes.
- Step 2 (Geographic Dimension): The geographic dimension needs to focus on process areas, since an overview an different process models is required
- Step 3 (Visualization Dimension): As temporal aspects are required, the visualization shall be time-based.
- Step 4 (Filter Settings): Temporal filters need to be applied based on given deadlines.

A test engineer shall identify process models that deliver results (e.g., documents) until a certain deadline. The engineer has to use these results to trigger other process executions. Accordingly, a time-based visualization is needed to identify temporal dependencies. Further, it is sufficient to identify root nodes within a process area. The desired navigation state ($NS_n = (2, 1, 0)$) and the according navigation space based on the main success scenario are depicted in Figure 7.17.

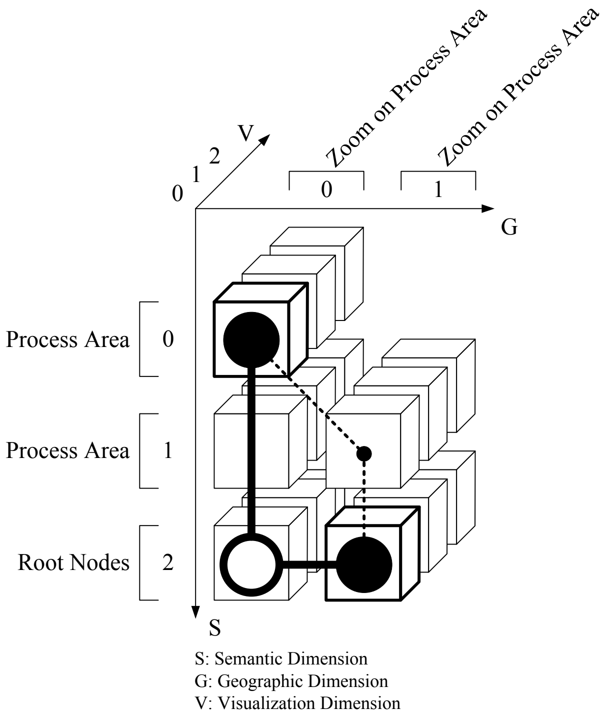


Figure 7.17: Use Case 7 - The navigation path within the navigation space.

Analysis: Based on the navigation space concepts, we can calculate the distance between start state $(0, 0, 0)$ and end state $(2, 1, 0)$.

$$DIST((0,0,0),(2,1,0)) = \sqrt{2^2 + 1^2 + 0^2} \approx 2{,}24 \tag{7.34}$$

Tackling start state $NS_0 = (0,0,0)$ the navigation sequence to $NS(2,1,0)$ can be defined and calculated as follows:

$$Nav_a = (i_1, i_2) = \left((2,0,0)^T, (0,1,0)^T\right) \tag{7.35}$$

$$NAVDIST(Nav_a) = \sum_{i=0}^{n-1} DIST(P_{i+1}, P_i) = \sum_{0}^{2} 1 = 3 \tag{7.36}$$

Improvement: We consider an alternative navigation sequence by taking 2-dimensional process interactions into account. For example, the following navigation sequence can be defined:

$$Nav_b = (i_1, i_2) = \left((1,1,0)^T, (1,0,0)^T\right) \tag{7.37}$$

$$NAVDIST(Nav_b) = \sum_{i=0}^{n-1} DIST(P_{i+1}, P_i) = (\sqrt{2}) + 1 \approx 2{,}41 \tag{7.38}$$

Its effectiveness can be calculated as follows:

$$Eff(Start, End, Nav_a) = \frac{2{,}24}{3} \approx 74{,}67\% \tag{7.39a}$$

$$Eff(Start, End, Nav_b) = \frac{2{,}24}{2{,}41} \approx 92{,}94\% \tag{7.39b}$$

A wireframe, visualizing the desired navigation state $(2,1,0)$, is depicted in Figure 7.18. Process models are visualized as process boxes in a time-based view. In turn, deadlines are integrated and serve as filter criteria. Thus, only those process models are displayed that end until these deadlines are reached.

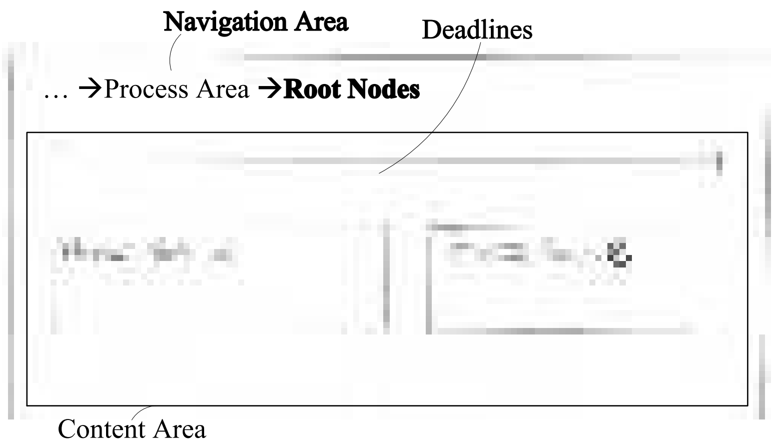


Figure 7.18: Use Case 7 - Wireframe of the visualized navigation state $(2,1,0)$.

7.9 Discussion

A 3-dimensional navigation space allows for numerous navigation possibilities that may be applied by users. In the present approach, as main benefit the semantic and the geographic dimensions are supported. In particular, this enables more sophisticated navigation options, e.g., navigating to states with a high information density. Available filter mechanisms allow for better handling these navigation states despite the high amount of displayed information.

Figure 7.19 compares the ProNaVis navigation concept with both the Google Maps and the process portal presented in Chapter 1. As can be seen in Figure 7.19a, the Google approach allows for static navigation along the semantic and geographic dimension. However, both dimensions are hard-wired, i.e., zooming to a certain level of detail automatically increases the level of the semantic dimension. However, the visualization dimension is independently adjustable from this zooming dimension on each detail level.

The navigation space applied in the context of process portal from the automotive domain is created manually (cf. Figure 7.19b). Thereby, navigation states are manually constructed (e.g., as images). In turn, navigation sequences have been created using manual links and image maps. In particular, process navigation is limited and the maintenance effort in case of changes becomes very high.

Finally, the ProNaVis concept provides a navigation space allowing users to navigate within three different navigation dimensions (cf. Figure 7.19c). In particular, independently navigating along the semantic and geographic dimension, in combination with the applied filter mechanisms, allows for complex navigation opportunities, not considered by common navigation concepts so far (see Chapter 8 for details).

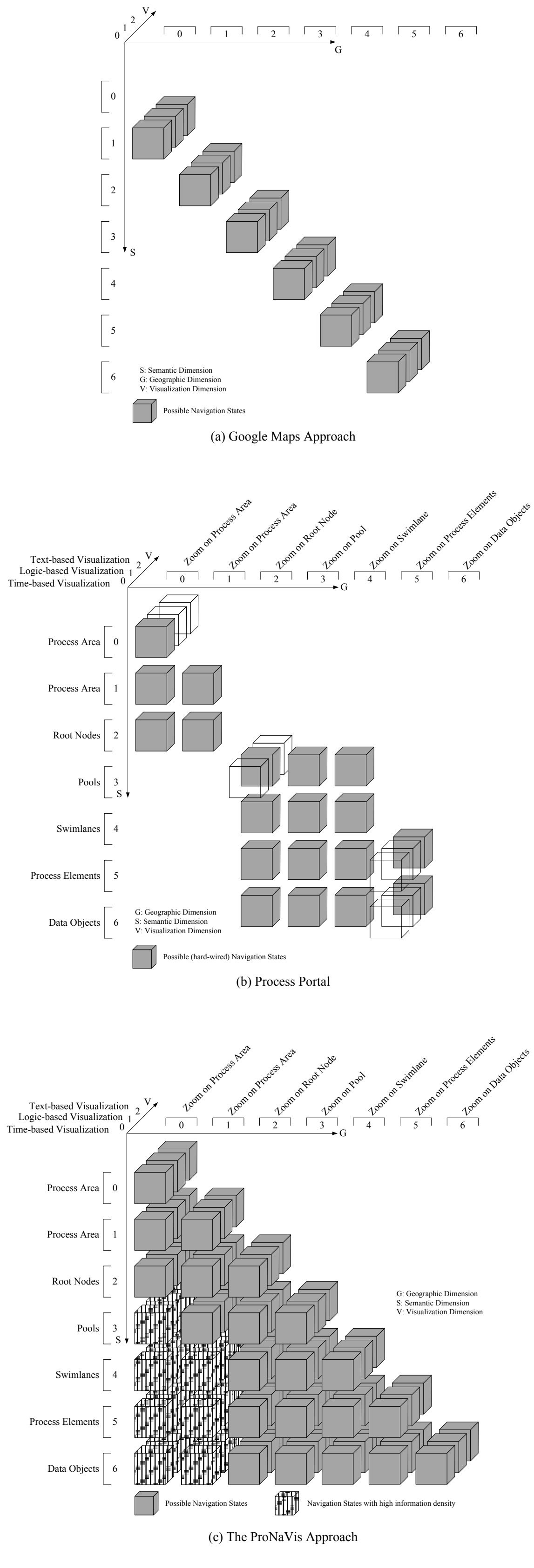
Table 7.2 shows how the different use cases can be supported by existing navigation approach. Note that only ProNaVis is able to support all use cases. Further note that the ProNaVis concept constitutes a generic concept for navigating in process model collections. In particular it might be applied to other domains as well. Consequently, other use cases not presented in this thesis can be supported as well.

Use Cases	Navigation State	Process Portal	Google Maps	ProNaVis
1 - Project Manager	(2,0,0)	✓		✓
2 - Business Unit Manager	(5,2,1)			✓
3 - Requirements Engineer	(6,5,2)	✓	✓	✓
4 - New Employee	(5,1,0)			✓
5 - Quality Manager	(5,0,1)			✓
6 - Quality Engineer	(5,0,1)			✓
7 - Test Engineer	(2,1,0)	✓	✓	✓

Table 7.2: How navigation approaches support the use cases.

7.10 Summary

This chapter demonstrated the applicability of the ProNaVis concept along characteristic use cases from the automotive industry. First, the used navigation space is introduced. Second, the basis model is created by omitting forbidden navigation states (i.e., navigation states with too few or too much information). Furthermore, we showed how a navigation space can be used as basis for applying the presented formalizations introduced in Chapter 5 to the use cases. Each use case was described in detail and different navigation sequences were analyzed and investigated to improve navigation effectiveness. Finally, we discussed the benefits of ProNaVis and compared it with the Google Maps approach as well as the



(a) Google Maps Approach

(b) Process Portal

(c) The ProNaVis Approach

Figure 7.19: Comparison of navigation spaces provided by different navigation approaches.

process portal from the automotive domain. In this context, the separation of semantic and geographic dimension is a key factor to successfully support process navigation along different navigation dimensions.

Part III

Validation

8 Related Work

There exists a variety of approaches in different research areas dealing with navigation in complex information spaces. Like ProNaVis, these approaches consider different navigation dimensions. However, existing approaches significantly differ from ProNaVis as they do not support three independent navigation dimensions. Instead, they only support one or two navigation dimensions (cf. Figure 8.1).

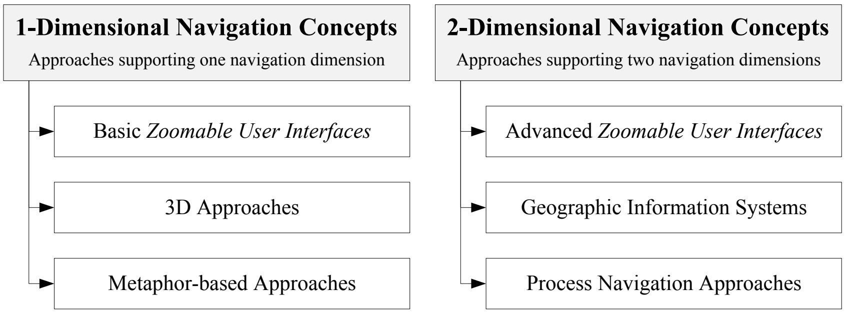


Figure 8.1: Two main categories of related work.

There exist few concepts for navigating in process model collections. Most existing navigation concepts are based on *information spaces*. However, these concepts can be mapped to process model collections as well [New96]. The major problem to be solved is to find the exact information needed [HMR11b]. For this purpose, i.e., to hide the complexity and structure of these information spaces as well as to offer access to the information, interaction techniques are used [Cha93].

The remainder of this chapter is structured as follows. Navigation approaches supporting only one navigation dimension are presented in Section 8.1. Section 8.2 then discusses navigation concepts supporting two navigation dimensions. Finally, Section 8.3 summarizes the chapter.

8.1 1-Dimensional Navigation Concepts

This section describes approaches that enable navigation along a single navigation dimension. Note that a second navigation dimension may be considered as well, but is not manually adjustable by the user. We first introduce early navigation concepts from the area of zoomable user interface (ZUI) (cf. Section 8.1.1). Second, concepts related to the area of 3D environments (cf .Section 8.1.2) are presented. Third, Section 8.1.3 deals with metaphor-based navigation concepts. Finally, Section 8.1.4 discusses our findings.

8.1.1 Basic Zoomable User Interfaces

This section describes early, but fundamental navigation approaches from the area of ZUI, focusing on zooming functionality, i.e., the geographic dimension. The semantic dimension is partially considered, but is then hard-wired with the geographic dimension. The visualization dimension is not explicitly addressed by these approaches.

Pad++

Bederson and Hollan [?, BWS93, BH94, BM98] introduce *Pad++*–a framework applying the concepts from ZUI [PF93]. Pad++ uses zooming as a basic interaction concept to navigate in complex information spaces. In particular, it represents an alternative to traditional window and icon-based user interface design approaches. The major goal is to ease the search for specific information in large information spaces. As a particular challenge, effective access to a large information space on a much smaller display needs to be provided.

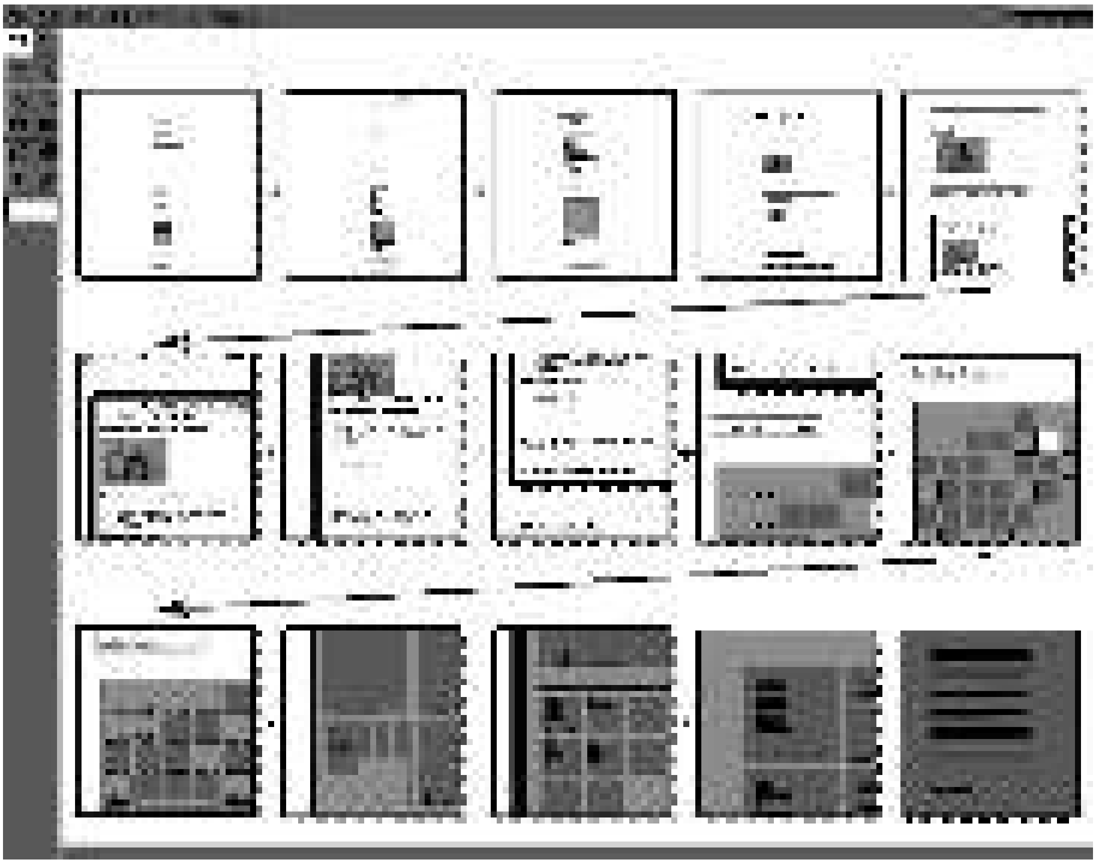


Figure 8.2: A sequence of views during zooming. [BHP+96].

Pad++ allows viewing information at different levels of detail by using the natural spatial way of thinking, i.e., zoom in to get more detailed information, and zoom out to get a better overview (cf Figure 8.2).

Unlike traditional approaches [DDF+90, Hil94], which rather recommend filtering in most cases, Pad++ structures information by providing the most highly rated information largest on the screen, whereas related, but lower rated information is presented nearby and smaller.

Animations are used for the intuitive navigation through the information space [BHP+96]. Animations combine panning and zooming to emphasize the specified location. If the end point is more than one screen width away from the starting point, the animation zooms out to a point midway between the starting and ending points such that both points becomes visible. The animation then smoothly zooms into the destination. This maintains the viewer's context and the speed of animation since most of the panning is performed when zoomed out. Note that this covers much more ground than panning while being zoomed in.

Pad++ fully supports the geographic dimension as introduced in ProNaVis. The semantic dimension is implicitly considered, but hard-wired with the geographic dimension, i.e., the level of detail of the displayed information changes depending on the selected zooming level. Navigation solely along the semantic dimension is not possible.

JAZZ/Piccolo

With JAZZ [BMG00][1] and PICCOLO [BGM04][2], Bederson et al. present an advancement of Pad++. In particular, JAZZ constitutes a basic toolkit for creating zoomable applications based on 2D graphics. JAZZ further provides efficient zooming animation. By using a hierarchical scene graph model with cameras, JAZZ is able to directly support a variety of common interface mechanisms [FR01]. This includes hierarchical groups of objects providing transformation, translation, scale, rotation, zooming, and multiple representations.

Figure 8.3: Screenshot of PhotoMesa written in JAZZ [BGM04].

Based on Pad++ concepts, JAZZ supports the geographic dimension of the provided zooming functionality. The semantic dimension is considered, but is still hard-wired with the geographic dimension, even if the used hierarchical scene graph provides a technical basis for navigating along the semantic dimension. Finally, there is no support of the visualization dimension, i.e., only static visualizations are provided.

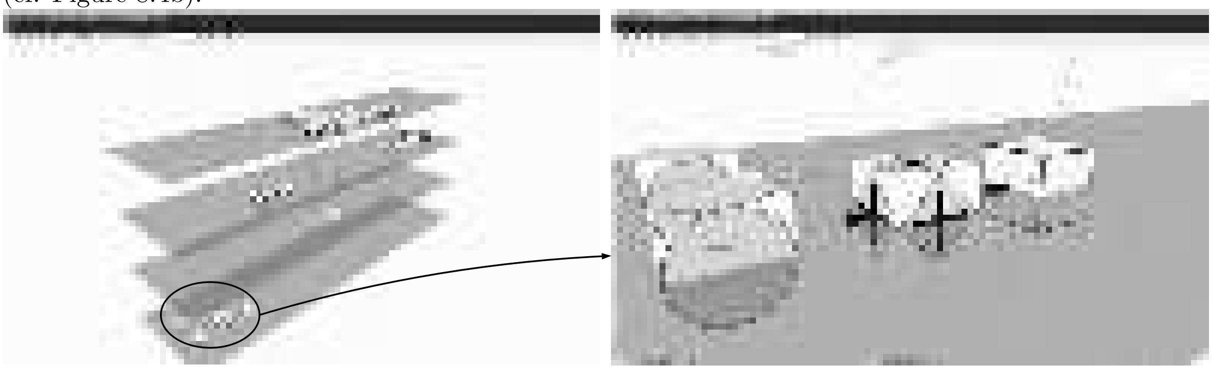
[1] http://www.cs.umd.edu/hcil/jazz/play
[2] https://code.google.com/p/piccolo2d/

8.1.2 3D Approaches

Various approaches deal with 3D graph representations [HMM00, Hon05, Mun97] and 3D environments [BEH$^+$08, BR09, vPD08]. All of them focus on navigation along the geographic dimension as they only allow for zooming in and out of a given 3D environment. The only available visualization (in terms of the visualization dimension) is a 3D representation of process models. Note that, in this context, 3D only describes the way of spatial information visualization–independent of the supported navigation dimensions. In the following, two interesting concepts, dealing with navigating in 3D environments in more detail, are presented. A broader overview on 3D visualization approaches for general information visualization can be found in [TC09].

Flight Navigator

Zooming and panning in a 3D environment is realized by the *Flight Navigator* concept [Eff12]. It supports numerous interaction paradigms that enable the user to present, inspect and analyze models in a 3D-environment (cf. Figure 8.4). In particular, it offers navigational support to users when browsing models. Thus, the geographic dimension is addressed as zooming is a crucial aspect in this context. Figure 8.4 depicts two screenshots of the Flight Navigator concept. As can bee seen, the user can zoom from an overview showing the entire model (cf. Figure 8.4a) to a more specific area within a specific swimlane (cf. Figure 8.4b).



(a) (b)

Figure 8.4: The Flight Navigator Tool [Eff12].

In turn, the semantic dimension is static; i.e., all available information is shown at the same time and the level of detail is not adjustable. The same applies to the visualization dimension, i.e., no alternative visualizations are available.

Virtual Worlds for Process Modeling

Navigation approaches inspired by 3D virtual worlds are presented in [WBR10, BRW11, PRJB13]. In particular, they extend previous 3D approaches with the aspect of collaborative process modeling. Specifically, avatars, as used in *third-person* games, are used to support the collaborative modeling of process models. As these avatars can be freely moved within the virtual environment, the geographic navigation dimension is addressed. A process participant might either move his avatar away from the created process model (cf. Figure 8.5a) to view a bigger part of it or move closer to a process object (e.g., a single task) in order to edit this object (cf. Figure 8.5b). Navigation corresponds to moving the avatar within the 3D environment while, at the same time, changing the representation of the process model along the geographic dimension.

Figure 8.5: Collaborative process modeling in a virtual world [BRW11].

Again, the semantic dimension is static, i.e., the detail level remains constant. The visualization dimension is limited to one single visualization, i.e., the 3D visualization of process models.

8.1.3 Metaphor-based Approaches

There exist other navigation approaches that are based on real-world metaphors for navigating in information spaces, e.g., landscapes and cities. The use of metaphors facilitates the understanding of the approaches as well as their use during navigation [AB05, Ben01]. Unlike the previously presented approaches, these navigation concepts take the semantic dimension into account as well. However, this dimension is still hard-wired for the geographic dimension, i.e., the level of detail is automatically increased when the user zooms on a specific area.

Landscape Metaphor

With *Bead* Chalmers [Cha93] presents a spatial landscape metaphor providing a navigation concept for a collection of documents. Bead is a prototypical system for the graphical exploration of information. For example, spatial proximity is used to represent similarity in a quick and comprehensible way [LJ08]. In this context, similar documents are placed close to each another and further from dissimilar documents. The emerging structure is then represented as a landscape or map of the information within the document set. The goal is to enable geographical interaction with a database of information, and to move away from interaction styles requiring knowledge of query languages and the database itself. In turn, this allows people to move from cognitive problem-solving to more natural strategies, such as zooming and panning, and to support more exploratory modes of use.

Figure 8.6a shows a map-like structure of scientific articles. Users are free to move over the landscape. Landmarks and borders are used for orientation purpose. Individual documents are shown as colored triangles placed within the landscape, producing collective patterns of density and locality. Again, the semantic dimension is statically linked to the geographic one. Thus, information regarding single documents are not provided on this abstract level of detail. Users may then zoom into a specific area such that the detail level of the information displayed is adapted as well. As can be seen in Figure 8.6b, titles and authors are additionally visualized then due to the increased level of detail.

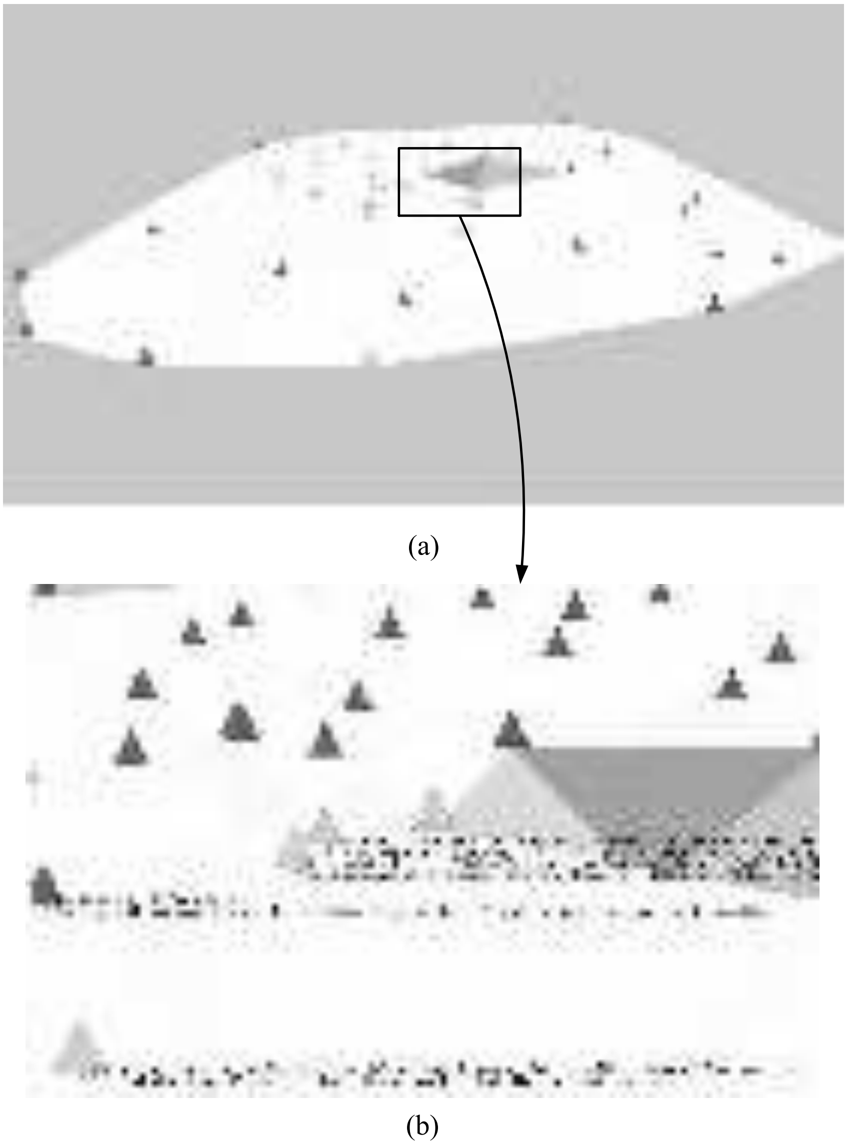


Figure 8.6: *Bead*: A landscape metaphor [Cha93].

The presented approach enables access to a complex information space based on the model or metaphor of a landscape. Accordingly, the display design is directed towards a more exploratory and dynamic style of use compared to most traditional information retrieval systems. Further, it tries to take advantage of our natural spatial experience by presenting a set information as a mostly open landscape. The geographic dimension can be used to zoom in and out, while adjusting the semantic dimension. However, the visualization remains static.

Information City

Dieberger and Frank [DF98] propose a city metaphor to support navigation in complex information spaces. In particular, they present a navigation concept based on the structure of a city, denoted as *Information City*. In particular, Information City is a conceptual spatial user interface metaphor for large information spaces, which is based on structures found in real cities as well as, on knowledge about city-planning and on how people move in such environments.

A city constitutes a familiar environment for humans and, hence, is an excellent metaphor, which can be easily extended. Generally, any spatial user interface metaphor has navigational as well as organizational advantages [AB05]. Furthermore, there is a strong relationship between spatial metaphors and information visualization, i.e., the visualization communicates the structure of the information space to enable easy navigation for users.

This concept addresses the visualization dimension as well. Depending on the level of detail, information visualization is changed. Figure 8.7a, for example, shows the area of *computer technology* visualized as a city map. The user may first move over the city for some time and study its layout, before deciding to go for the *computer graphics* district. When entering this district, i.e., zooming into this area, information is then visualized in a different way (cf. Figure 8.7b).

(a) (b)

Figure 8.7: The city metaphor [DF98].

City metaphors define a conceptual spatial metaphor for navigating in complex information spaces. Besides the semantic dimension, the visualization dimension is explicitly considered by this approach. However, the visualization dimension is still hard-wired to the geographic dimension, i.e., there exists a visualization for the abstract representation of the information space and another one for a detailed representation.

8.1.4 Discussion

Table 8.1 summarizes how the presented concepts cover the navigation dimensions provided by ProNaVis. As can be seen, the geographic dimension is supported by each of the presented concepts. It enables zooming into and out of a given information space and hence corresponds to the natural spatial way of human thinking [BWS93].

The semantic dimension, however, is not supported by all concepts. The flight navigator concept and the virtual world concepts, for example, do not provide a semantic dimension at all; i.e., information is always made available at the same level of detail. Thus, it cannot be abstracted, i.e., the level of detail cannot be decreased.

Concept	Sem. Dim	Geo. Dim	Vis. Dim
Flight Navigator [Eff12]	✗	✓	✗
Virtual Worlds [WBR10, BRW11, PRJB13]	✗	✓	✗
Pad/Pad++ [BH94, BHP+96, BM98]	❍	✓	✗
JAZZ/PICCOLO [BMG00, BGM04]	❍	✓	✗
Landscape Metaphor [Cha93]	❍	✓	✗
Information City [DF98]	❍	✓	❍

✓: totally supported; ❍: considered, but not manually adjustable; ✗: not considered

Table 8.1: Support of navigation dimensions.

If the semantic dimension is supported, it is always automatically linked to the geographic one, i.e., when the user zooms into an information space, the level of detail is increased as well. This facilitates navigation on an abstract level as the detail level of information is always adopted to the level of zooming.

The following section describes navigation concepts, supporting two navigation dimensions.

8.2 2-Dimensional Navigation Concepts

This section presents concepts that allow navigating within information spaces along two dimensions; i.e., two of the three navigation dimensions proposed by ProNaVis can be manually adjusted by the user. The third navigation dimension may be addressed as well, but is not manually adjustable. Specifically, we present advanced ZUI concepts (cf. Section 8.2.1) and concepts from geographic information systems (GIS) (cf. Section 8.2.2).

8.2.1 Advanced Zoomable User Interfaces

Advanced ZUI concepts replace conventional windows, icons, menus, and pointers (WIMP) concepts [GMR07]. The goal is to facilitate data presentation on limited screen sizes by allowing the user to alter the scale of the viewpoint such that it shows a decreasing fraction of the information space with an increasing magnification [RB09]. A ZUI displays graphical information on virtual canvas, which can be seen by a virtual "camera" panning and zooming over the surface (i.e., geographic dimension) [BMG00]. For example, a global overview of an information space may be presented to the user for the sake of orientation. Based on this, users may re-allocate the screen space according to the information they are interested in. A ZUI allows users to dynamically change views on information spaces.

A ZUI can be categorized as natural user interface (NUI) as it builds upon the user's knowledge and understanding of real-world spacial concepts and, hence, leads to a more natural and reality-based interaction [JGH+08]. Navigation approaches applying these concepts are presented in the following. They consider a separate handling of the semantic and geographic dimension, but still lack independent visualizations.

ZEUS

With ZEUS Gundesweiler et al. [GMR07] present a zoomable explorative user interface. In particular, ZEUS is a web application allowing for browsing, searching and object presentation in complex navigation spaces. Usually, this functions are addressed separately in software systems [CDT00]. However, ZEUS hierarchically structures the information space and, hence, is able to present information on different levels of detail (along the semantic dimension).

In particular, combining search and filtering with zooming interaction techniques allows tailoring search results. Navigation through animated panning and zooming supports natural orientation capabilities of users in the best way when searching for information needed (i.e., along the geographic dimension). In general, it is easier for users to visually move through an information space to explore its contents than to navigate through a hyperlinked collection of objects [GMR07].

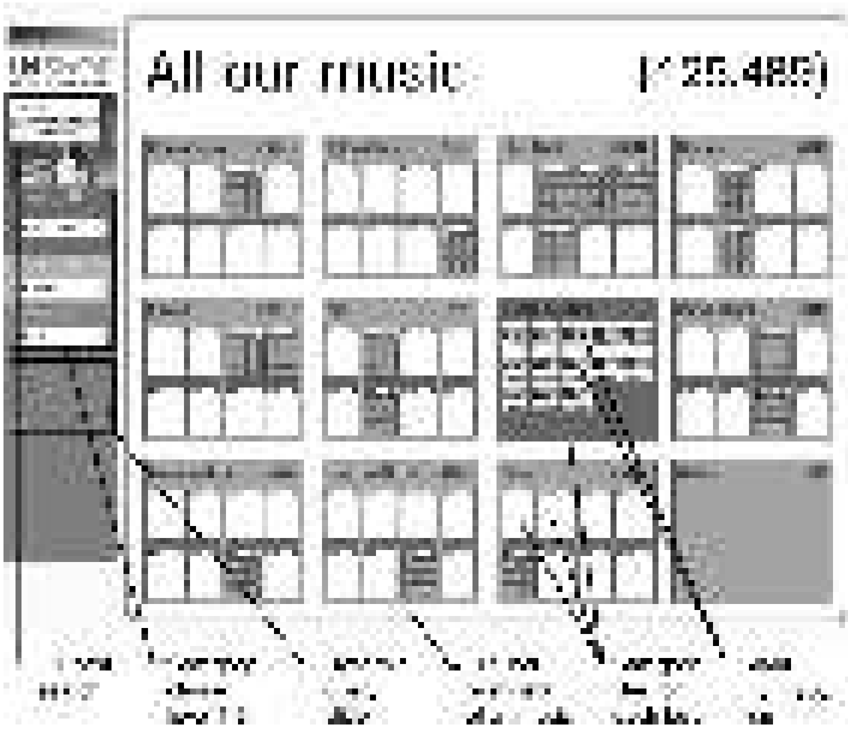


Figure 8.8: The ZEUS framework applied to a virtual music store called iNShOP [GMR07].

As an example consider Figure 8.8. It depicts *iNShOP*, which is a virtual music store [GMR07]. The application consists of a main area visualizing the objects and categories as well as a filter area for searching and selecting music categories. In turn, filter and category operations are triggered by selecting category attributes in the combo boxes. Selecting “Music type” as first category level, for example, organizes the results in different tiles. The latter constitute the main visual components used to organize the information space and to visualize the data items. There exist two different kinds of tiles. Category tiles organize the information space as groups on different hierarchy levels. They may include other category or information tiles to visualize the data items in the respective level as well. An information tile visualizes one item and may include text, images and multimedia objects (e.g., video and sound). Selecting a category in a combo box, in turn, initiates a recalculation of the tile organization as well as redrawing of tiles [GMR07]. Detailed information tiles indicate how the semantic dimension may be applied independent from the geographic dimension.

By clicking on a tile, zooming operations are triggered and the selected tile is enlarged to fit to the screen size. Furthermore, panning operations may be applied to switch between neighbored tiles. Both zooming and panning operations are presented to the user through an animation to make the actions visually traceable [vWN03, vWN04]. Based on the enlarged tile, the readability of information is improved. By clicking on the tile area in the background, in turn, the user may zoom out again.

Gundesweiler et al. show that the combination of searching and hierarchical information structuring enables an effective and efficient navigation approach. In particular, an independent navigation along the geographic and semantic dimension can be realized. Compared to the ProNaVis framework, however, the visualization dimension is not considered, i.e., only one static kind of visualization is provided.

ZOIL

ZOIL is both a design paradigm and software framework for post-WIMP concepts [JKGR08, ZJR11]. The provided interaction concept follows basic ZUI principles [Ras00, PF93]. In particular, the information space is not limited to the visible screen size, but resembles a virtual canvas of infinite size for persistent visual-spatial information. Items in the information space may be directly accessed by panning to the right spot and zooming in [Ras00].

Figure 8.9: Semantic zooming in ZOIL [JKGR08].

In particular, ZOIL is used for document management. It applies semantic zooming [PF93] to all documents, which means that geographic growth in display space is not only used to render more details, but to reveal additional, semantically different content (cf. Figure 8.9). This transition between iconic representation, meta-data, and full-text/full-functionality prevents the problem of information overload and disorientation, typically caused by traditional WIMP approaches with multiple overlaying windows or occluding renderings of details [JKGR08].

Compared to ProNaVis, ZOIL provides different visualizations for documents as well. These visualizations are further combined within the ZOIL work environment (cf. Figure 8.10a); e.g., a *calendar* visualization (on top), where documents are aligned on a timeline according to their creation date. Other visualizations may be added as modular plug-ins. Furthermore, documents may be visualized according to their size, their location, or the project they are assigned to. The user may zoom into a specific area of the work environment in order to obtain more detailed information about a document (cf. Figure 8.10b). Finally, navigation along the semantic dimension is limited in ZEUS as this dimension is hard-wired to the geographic one.

Squidy

Squidy constitutes an interaction library easing the design of natural user interfaces by unifying relevant frameworks and toolkits in a common library [KRR09]. Squidy provides a central design environment based on high-level visual data flow programming and combined with zoomable user interface concepts.

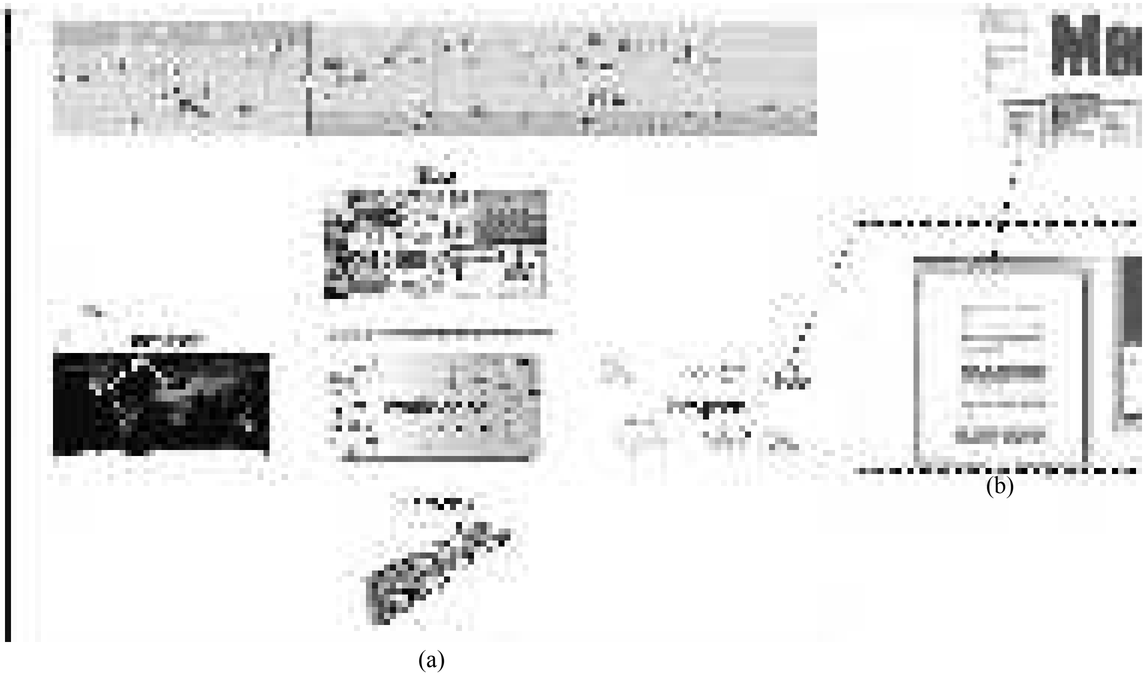


Figure 8.10: The ZOIL workspace [JKGR08].

In particular, semantic zooming is used to enable on-demand-access to more advanced functions, i.e., to change the level of detail. Consequently, the complexity of the user interface can be adjusted to the exact needs of the user.

Figure 8.12a provides a high-level visualization of the data flow between an input and output device. This visualization is called *pipeline* visualization. It constitutes a simple, yet powerful visual language to design the interaction logic. Thereby, the user models the data flow using nodes for input and output devices, as well as for filter or data processing tasks [KRR09].

Figure 8.11: Different visualizations of Squidy [KRR09].

Squidy applies a zoomable user interface concept to navigate within the design environment. A node may be focused, and semantic zooming may be applied to get more information about it (cf Figure 8.12a). Different visualizations are used to provide information in different ways. To obtain a more detailed description of a single node, *information* visualization is used (cf. Figure 8.12b). To change the properties of this specific node, a *table* visualization may be used (cf. Figure 8.12). All properties listed within this table are directly editable.

As a unique feature of Squidy one may semantically zoom into edges in order to obtain visual information about the data flow between two nodes.

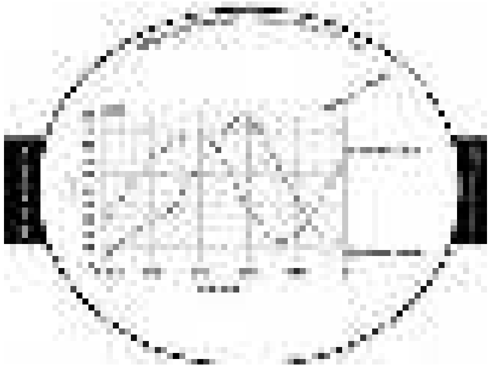


Figure 8.12: Data flow visualization with Squidy [KRR09].

Squidy explicitly uses semantic zooming to allow for information visualization on different levels of detail in a ZUI. However, the geographic dimension is hard-wired to the semantic one. Squidy further provides various kinds of visualizations. The *pipeline* visualization, for example, is related to the *logic-based* visualization described in Chapter 6. Further, the *information* visualization provides similar information as the *text-based* visualizations presented in Chapter 6. However, Squidy does not provide a *time-based* visualization.

8.2.2 Geographic Information Systems

Geographic information systems deal with the presentation of all types of geographical data [HCC12]. Map applications, such as Google Maps[3], Microsoft Bing Maps[4] and OpenStreetMap[5], have developed sophisticated concepts for navigating in complex information spaces. In particular, they combine the geographic dimension with the semantic one. Additionally, they provide a visualization dimension, i.e., the visualization of the presented information can be displayed in a different manner independent from the current level of detail.

Geographic information systems allow users to scale the information space into different levels of detail (cf. Figure 8.13). Figure 8.13a, for example, shows an entire country on one screen. Hence, only abstract information is displayed, e.g., the names of countries, big cities, and main highways. The user may then zoom into a certain area (i.e., along the geographic dimension), which, in turn, results in an increased level of detail (along the semantic dimension) (cf. Figure 8.13b). Accompanying to this, the infrastructure of a specific city will be displayed, including names of smaller streets and specific points of interest.

Besides the geographic and semantic levels, different visualizations are provided, e.g., a map visualization and a satellite visualization (cf. Figure 8.14). In particular, these views can be manually applied independent of the level of detail. Accordingly, geographic information systems can be classified as 2-dimensional navigation approaches, since the geographic and semantic dimensions are combined to one dimension.

[3]Google Maps: http://maps.google.com
[4]Bing Maps: http://www.bing.com/maps/
[5]OpenStreetMap: http://www.openstreetmap.org/

Figure 8.13: The combined geographic and semantic dimension of Google Maps [Ray10].
[Map data: ©2015 GeoBasis-DE/BKG (©2009), Google]

8.2.3 Process Navigation Approaches

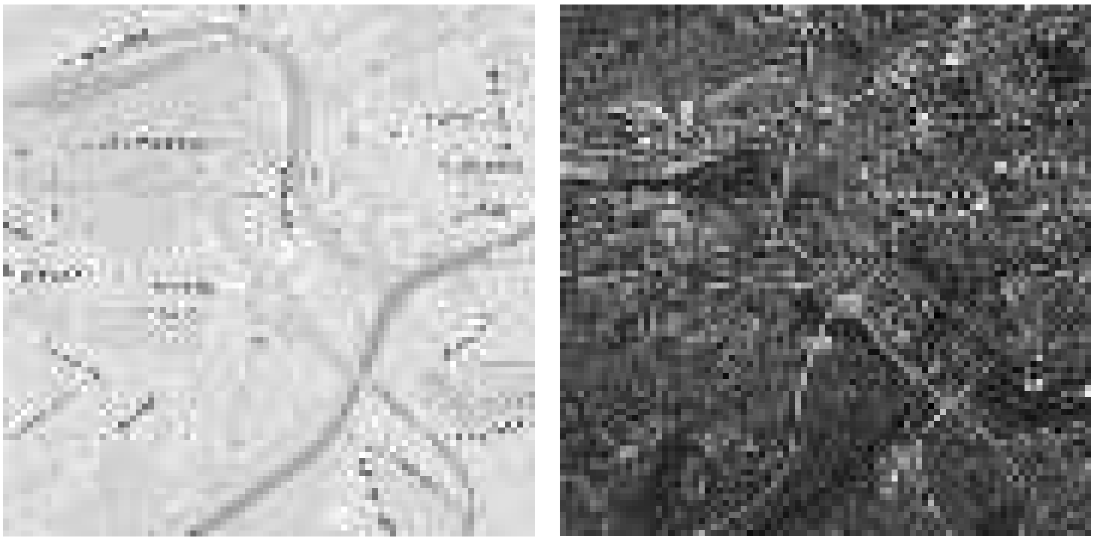
The *Proviado* [RKBB12, BRB07] as well as the *proView* frameworks [KRW12, KKR12, KR13a, Kol15] deal with the creation of different views on single business process models, i.e., they both allow for process model abstractions. Specifically, both frameworks apply *aggregation* and *reduction* techniques to create flexible views on complex business process models (cf. Figure 8.15).

Different hierarchical structures can be created for a process model during run time, by applying different graph reduction techniques. This allows for the navigation along the semantic dimension. Both frameworks provide a powerful set of techniques to aggregate or reduce a set of elements (e.g., tasks) in a process model. This allows aggregating different fragments within a single process model. In turn, ProNaVis only allows aggregating (i.e., abstracting) entire process models to process areas. Note that we would suggest to model smaller process models and combine them to a process model collection rather than to apply complex graph reduction techniques on a complex single process model.

Proviado supports different stakeholders having different roles. For example, managers need a more abstract view, whereas knowledge worker may want to hide (reduce) uninteresting tasks from the process model. *Proviado* further deals with different visualizations of process models although theses are only handled at a very abstract level. In particular, these visualizations are limited to changing the forms and colors of process tasks applying different cascading style sheets (CSS) [BBR06].

In turn, *proView* allows for personal views on a process model. Thereby, the synchronization and maintenance of multiple views on one process model is addressed. As opposed to other abstraction approaches [Bob08, SRW11], in addition, users may introduce process changes based on their particular process views [KKR12, KR13c].

Altogether, the presented concepts address the semantic dimension by providing aggregation and reduction techniques as well as the visualization dimension by enabling different styles of visualizing to process models. As opposed to ProNaVis, the geographic dimension is not explicitly considered.



(a) (b)

Figure 8.14: The visualization dimension of Google Maps [Ray10].
[Map data: ©2015 GeoBasis-DE/BKG (©2009), Google]

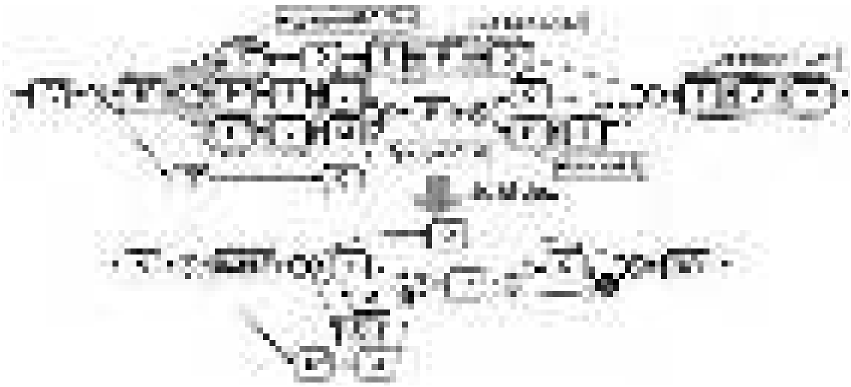


Figure 8.15: Creation of an abstract view on a process model [BRB07].

8.2.4 Discussion

This section introduced navigation concepts addressing the flexible navigation along two of the three ProNaVis navigation dimensions. Table 8.2 summarizes the presented approaches. GIS provide sophisticated navigation concepts and support the geographic dimension in combination with the hard-wired semantic dimension. Additionally, a GIS provides a flexible visualization dimension for the user. Regarding ZUI, we presented concepts supporting navigation in two navigation dimensions and a hard-wiring third dimension to one of the other two. *ZEUS* allows for a flexible navigation along the semantic and geographic dimensions. Furthermore, *ZOIL* enables the navigation along the geographic and visualization dimensions, whereas *Squidy* addresses the semantic and visualization dimensions. *Proviado* and *proView*, dealing with navigation in process models, only address the semantic and visualization dimension.

Concept	Sem. Dim	Geo. Dim	Vis. Dim
GIS	❍	✓	✓
ZEUS [GMR07]	✓	✓	❍
ZOIL [JKGR08, ZJR11]	❍	✓	✓
Squidy [KRR09]	✓	❍	✓
Proviado [BRB07]	✓	✗	✓
proView [KKR12]	✓	✗	✓

✓: totally supported; ❍: considered, but not manually adjustable; ✗: not considered

Table 8.2: Support of navigation dimensions.

8.3 Summary

This chapter discussed different approaches for navigating in complex information spaces. They all address at least one of the presented three navigation dimensions. However, none of the presented concepts supports all three dimensions independently from each other. Compared to ProNaVis, all presented concepts lack flexibility as certain dimensions are not considered or hard-wired to other ones. Table 8.3 summarizes the discussions of this chapter.

Area	Concept	Sem.	Geo.	Vis.
3D Approaches	**Flight Navigator** [Eff12]	✗	✓	✗
	Virtual Worlds [WBR10, BRW11, PRJB13]	✗	✓	✗
Basic ZUI	**Pad/Pad++** [BH94, BHP+96, BM98]	❍	✓	✗
	JAZZ/PICCOLO [BMG00, BGM04]	❍	✓	✗
Metaphor-based Approaches	**Landscape Metaphor** [Cha93]	❍	✓	✗
	Information City [DF98]	❍	✓	❍
Geographic Information Systems	**GIS (Google Maps, Bing Maps,...)**	❍	✓	✓
Advanced ZUI	**ZEUS** [GMR07]	✓	✓	❍
	ZOIL [JKGR08, ZJR11]	❍	✓	✓
	Squidy [KRR09]	✓	❍	✓
Process Navigation Approaches	**Proviado** [BRB07]	✓	✗	✓
	proView [KKR12]	✓	✗	✓
	ProNaVis	✓	✓	✓

Sem.: semantic dimension; geo.: geographic dimension; vis.: visualization dimension
✓: totally supported; ❍: considered, but not manually adjustable; ✗: not considered

Table 8.3: Support of navigation dimensions.

The presented navigation approaches for 3D environments (*Flight Navigator* and *Virtual Worlds*) focus on zooming along the geographic dimension. They solely provide a fixed semantic dimension (i.e., the level of detail is not adjustable) and only one complex visualization.

Basic ZUI approaches (*Pad*, *Pad++*, *JAZZ*, and *Piccolo*) focus on the navigation along the geographic dimension as this constitutes the natural way of human spatial thinking. However, they also consider the semantic dimension, that, in turn, is hard-wired to the geographic one. Compared to the 3D approaches, this facilitates user experience as the level of detail of the visualized information is always adopted to the current level of geographic zooming.

The same is applied to metaphor-based approaches (*Landscape Metaphor* and *Information City*), which also support the geographic dimension, again with a hard-wired semantic dimension. They further try to increase user experience by visualizing information based on well-known metaphors, such as landscapes or city maps in order to address the natural orientation capabilities of users. Additionally, for the first time,

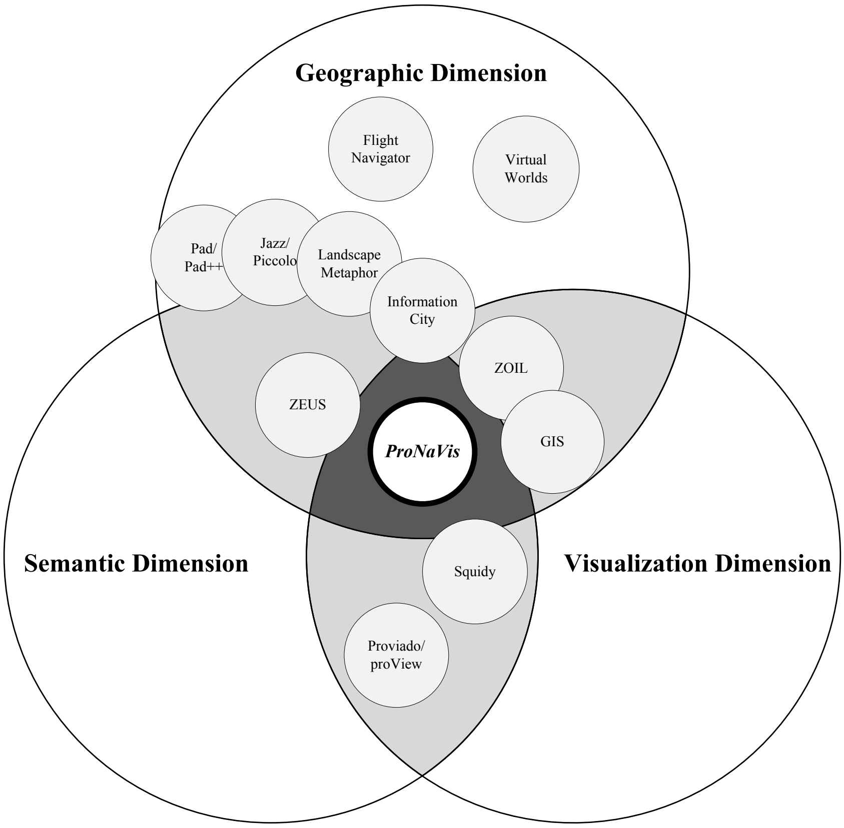


Figure 8.16: Visual classification of the presented approaches.

the *Information City* concept addresses the visualization dimension, as it provides different visualizations of information on different zooming levels.

GIS combine semantic as well as geographic navigation dimensions. However, GIS extend the *Information City* concept by a flexible and independent visualization dimension, i.e., the visualization may be changed independently from the current semantic and geographic level of detail.

Finally, advanced approaches from the ZUI area enable independent navigation in two dimensions. *ZEUS* allows for the navigation along the semantic and geographic dimension, whereas the visualization dimension is adjusted automatically. *ZOIL*, in turn, picks up the GIS approach and enables navigation along the geographic and visualization dimensions. In turn, the semantic dimension is automatically adjusted. Finally, *Squidy* emphasizes the semantic and visualization dimension, whereas the geographic dimension is adopted automatically.

As opposed to ProNaVis, neither GIS nor ZUI navigation approaches provide the freedom to navigate within three independent navigation dimensions.

Figure 8.16 classifies the presented navigation approaches based on the number of supported navigation dimensions. If a concept is drawn on the edge of a navigation dimension, the latter is at least considered by the concept, but is not adjustable by the user; i.e., it is automatically adjusted depending on another navigation dimension.

9 Proof-of-Concept Prototypes

To be able to demonstrate the concepts developed in the context of ProNaVis and to discuss them with users, we implemented two different prototypes. *ProNavigator* was created to illustrate the holistic ProNaVis functionality, i.e., the 3-dimensional navigation space. In particular ProNaVis constitutes the basis for a user experiment addressing the navigation functionality (cf. Chapter 10). In turn, *Compass* was developed as process navigation tool to be used by an industrial partner as process navigation tool supporting process participants when developing E/E car components. Both tools have taken the requirements presented in Chapter 2 into account (cf. Table 9.1).

Req #	**Requirement**
NavReq #1	Depending on a process participant's experience, the level of detail regarding a process task should be adjustable.
NavReq #2	Process participants should be able to adjust the level of detail regarding process model collection in order to obtain a quick overview on a specific task that is currently executed.
NavReq #3	Users should be enabled to access process tasks in other process areas.
NavReq #4	Relevant process information must be accessible at the level of single process models from the process model collection.
NavReq #5	Roles must be globally defined in a detailed manner.
NavReq #6	Process participants must be able to access process models on different levels of detail.
VisReq #1	Task descriptions must be documented in a well understandable manner.
VisReq #2	Temporal and logical dependencies must be considered when visualizing processes.
VisReq #3	Complex process information must be visualized in a comprehensible manner.
VisReq #4	Information about roles must be intuitively identifiable.
VisReq #5	The amount of visualized information should not overload process participants.

Table 9.1: Overview on the derived requirements.

The remainder of the chapter is structured as follows. Section 9.1 presents the *ProNavigator* prototype. This click-prototype demonstrates different interaction concepts for navigating in a 3-dimensional navigation space. In turn, Section 9.2 presents *Compass*, a powerful process navigation and visualization tool, developed in collaboration with an industrial partner. In particular, Compass supports knowledge workers dealing with the engineering of E/E components for cars, trucks, and buses. Section 9.3 discusses the presented applications and Section 9.4 concludes the chapter with a summary.

9.1 *ProNavigator*

The ProNavigator click-prototype deals with the navigation within a particular process space, i.e., navigating within a process model collection. In particular, *ProNavigator* focuses on the user interactions when navigating along the three navigation dimensions.

Figure 9.1: Basic user interface areas of ProNavigator.

Figure 9.1 shows a screenshot of the ProNavigator prototype. First, the *management area* (A) provides general functions. In the *navigation area* (B), a breadcrumb navigation concept is provided indicating the semantic level of the current navigation state. Clicking on breadcrumb elements, the user may navigate along the semantic dimension. Second, the *orientation area* (C) displays specific information depending on the current visualization, e.g., a timeline is presented in the time-based visualization. Finally, the *content area* (D) provides space for visualizing the content of navigation states. For this area, a navigation element (E) is provided in the upper right corner, which which allows for interaction possibilities with the three navigation dimensions.

In the following, the navigation element (E) and the application of the three navigation dimensions are described in more detail.

9.1.1 The Navigation Element

Regarding the three navigation dimensions, user interactions are enabled based on the *navigation element* shown in Figure 9.2. It allows manipulating the three navigation dimensions separately.

The navigation element provides a spherical shape, which is divided into clickable fragments grouped in three columns. Fragments in the middle column represent different visualizations and allow navigating along the visualization dimension. The currently applied visualization is represented by the fragment in the middle, which is highlighted in blue. In the example (cf. Figure 9.2), four visualizations are provided: a *time-based*, *logic-based*, *content*, and *turtle* visualization. Each visualization is represented by a specific icon. When changing visualizations, the navigation element is rotating around its horizontal axis, ensuring the current visualization is always in front (cf. Figure 9.3).

The semantic dimension, in turn, can be changed by clicking either on fragment A or fragment B of the navigation element (cf. Figure 9.2) . Fragment A, which is on the left, decreases the semantic level.

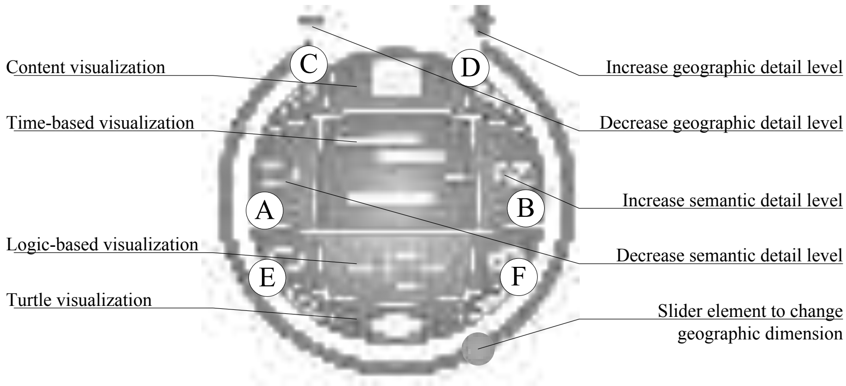


Figure 9.2: The navigation element.

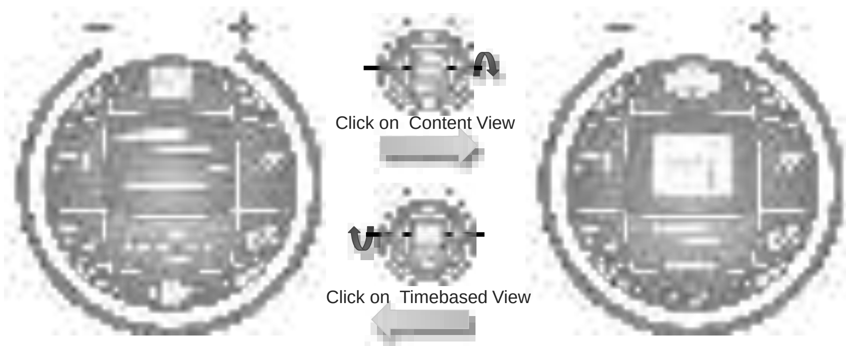


Figure 9.3: The rotation of the navigation element.

In turn, fragment B, which is on the right increases the semantic level. Fragments C, D, E, and F trigger a 2-dimensional process interaction, i.e., a click changes both the semantic and the visualization dimension(cf. Section 5.3).

Finally, a slider is surrounding the navigation element. By dragging the slider or clicking the "+/-" icons in the navigation element, the geographic level can be adjusted.

9.1.2 Example

We refer to a simplified view on a navigation space to illustrate ProNavigator.

The used navigation space (cf. Figure 9.4) comprises four levels along each navigation dimension. The semantic and geographic dimensions consist of levels 0 and 1 for process areas. In turn, process root nodes are represented on level 2 and process elements are provided on level 3. The visualization dimension provides four visualizations (time-based, logic-based, turtle and content visualization).

Figure 9.5 shows an example of navigating along the semantic dimension. Starting with navigation state $(1, 0, 0)$, a user navigates to state $NS(2, 0, 0)$. In particular, navigation state $NS(1, 0, 0)$ provides three process areas of a process model collection (cf. Figure 9.5a): *Planning*, *Holiday*, and *Post-processing*. The process areas are represented as rectangles, whose length corresponds to the respective duration. When increasing the semantic level using the navigation element, objects corresponding to semantic level 2 are displayed (cf. Figure 9.5b). In particular, process models represented by their root nodes are presented

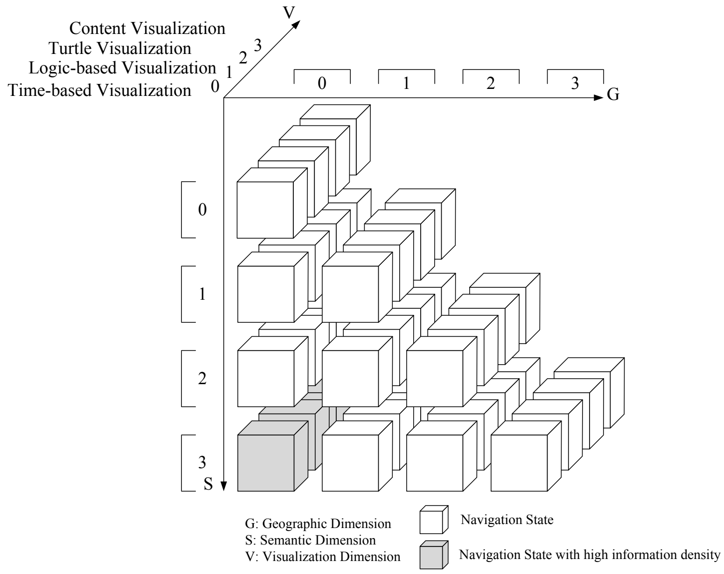


Figure 9.4: Exemplary navigation space.

within the three process areas. In the navigation space, this corresponds to a 1-dimensional interaction along the semantic dimension to navigation state $(2, 0, 0)$ (*NavReq #6)*).

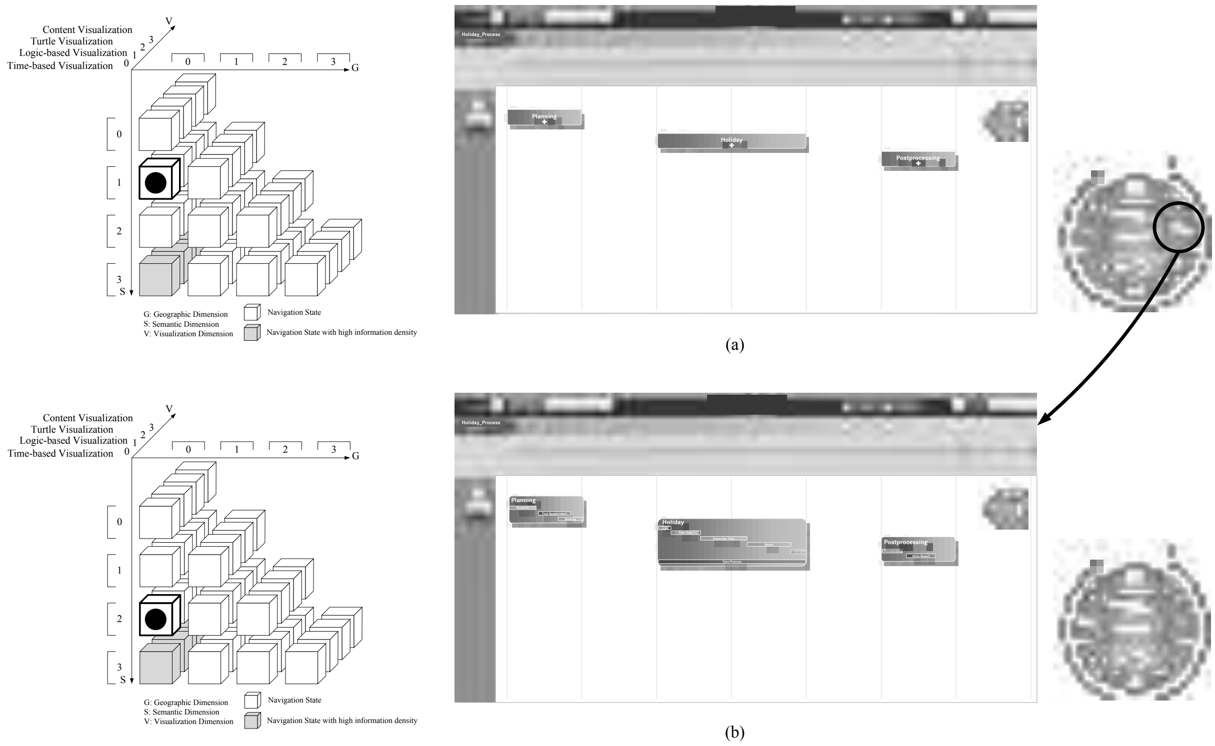


Figure 9.5: Navigating along the semantic dimension.

The geographic dimension can be adjusted using the "+/-" icons of the navigation element. Again we consider navigation state $(1, 0, 0)$ as starting point. Figure 9.6 illustrates how to navigate from this start state to a navigation state $NS(1,1,0)$. In particular, the user navigates to geographic level 1, i.e., he zooms to process area *Planning* (cf. Section 4.4.2). The resulting navigation state $(1,1,0)$ is depicted in Figure 9.6b. In general, zooming out of a specific process model collection reveals a better overview on multiple process areas (*NavReq #2*).

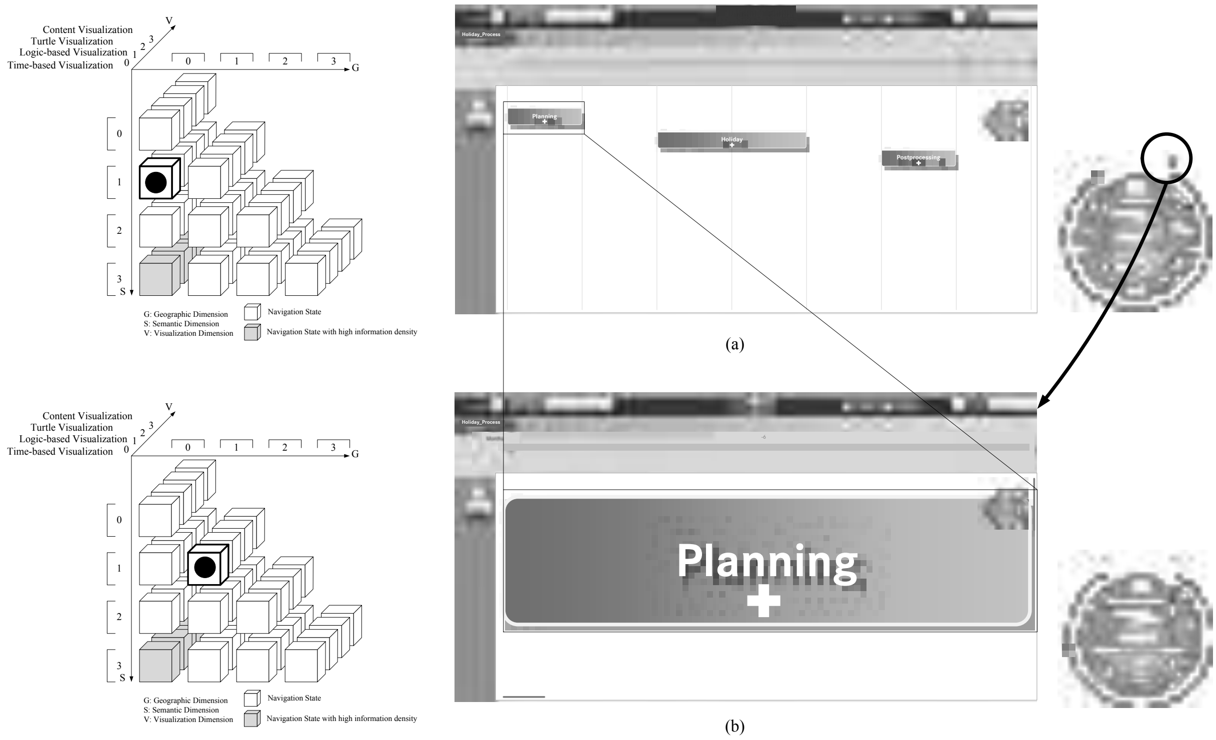


Figure 9.6: Navigating along the geographic dimension.

Different visualizations can be selected using the middle part of the navigation element. Figure 9.7 illustrates the state transition from a time to a logic-based visualization, i.e., from $NS(1,0,0)$ to $NS(1,0,1)$. Figure 9.7a shows navigation state $(1,0,0)$ once again, providing a time-based visualization of three process areas. Switching to a logic-based visualization allows visualizing process areas as boxes. Thereby, the logical execution order is illustrated through arrows indicating predecessor/successor relations. Note that the user may completely focus on either logical or temporal dependencies (*VisReq #2*). Instead, the turtle and content visualizations (cf. Chapter 6) allow for detailed textual descriptions, e.g., detailed information about single tasks (*VisReq #1*).

9.1.3 Combined Navigation Possibilities

Advanced ProNaVis concepts can be demonstrates with ProNavigator as well. In particular, ProNavigator allows for 2-dimensional process interactions (cf. Section 5.3). Figure 9.8 shows an example of combining the semantic with the geographic dimension. The 2-dimensional interaction facilitates the navigation from an abstract to a detailed part of a process model collection as it combines the zooming on a specific area with an increase of the semantic level (*NavReq #3*). Note that this navigational concept is used by the Google Maps approach as well (cf. Chapter 8).

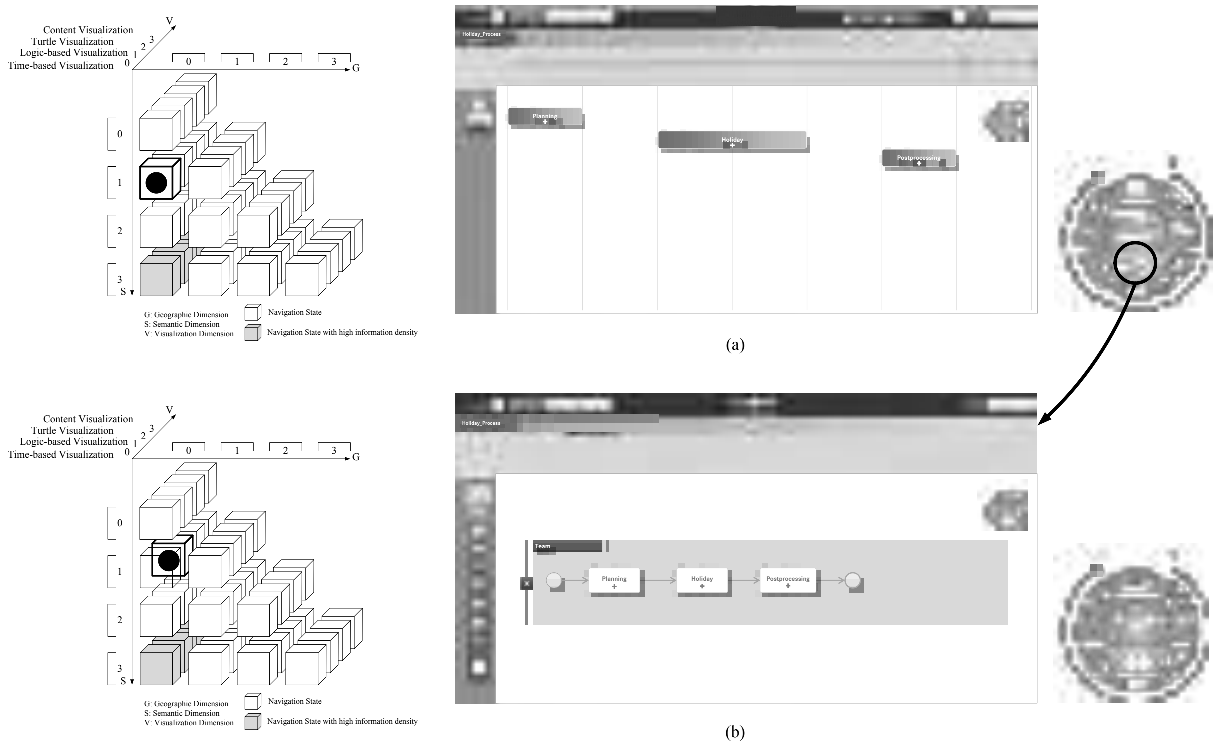


Figure 9.7: Navigating along the visualization dimension.

In ProNavigator, 2-dimensional interaction is triggered by double clicking on the desired reference object (process area *planning* in Figure 9.8a). ProNavigator then zooms to this specific process area (geographic level 1), while providing more detailed information (semantic level 2) at the same time (cf. Figure 9.8b). Thereby, the detail level is automatically adjusted (*NavReq #1*). Note that the breadcrumb navigation indicates the changed semantic level, while the slider at the navigation element indicates the changed geographic level.

ProNavigator provides another possibility for two dimensional interactions. Thereby, the fragments C, D, E, and F of the navigation element can be applied (cf. Figure 9.2) to change the detail level along the semantic dimension and the visualization at once. Based on the navigation state $(1, 0, 0)$, such an interaction results in navigation state $(2, 0, 1)$ (cf. Figure 9.9).

9.1.4 Conclusion

ProNavigator is a click-prototype illustrating the ProNaVis navigation concepts. The main goal of ProNavigator is to provide users with a realistic impression of three dimensional process navigation. Therefore, it implements concepts to navigate in a given navigation space. A navigation element is introduced as a central element enabling user interactions. Both 1- and 2-dimensional process interactions are considered by ProNavigator.

ProNavigator further applies already known user interaction concepts as well. For example, a breadcrumb navigation indicates the current detail level of the semantic dimension. Additionally, the user might directly interact with objects from the content area. Double clicking on an object, for example, triggers

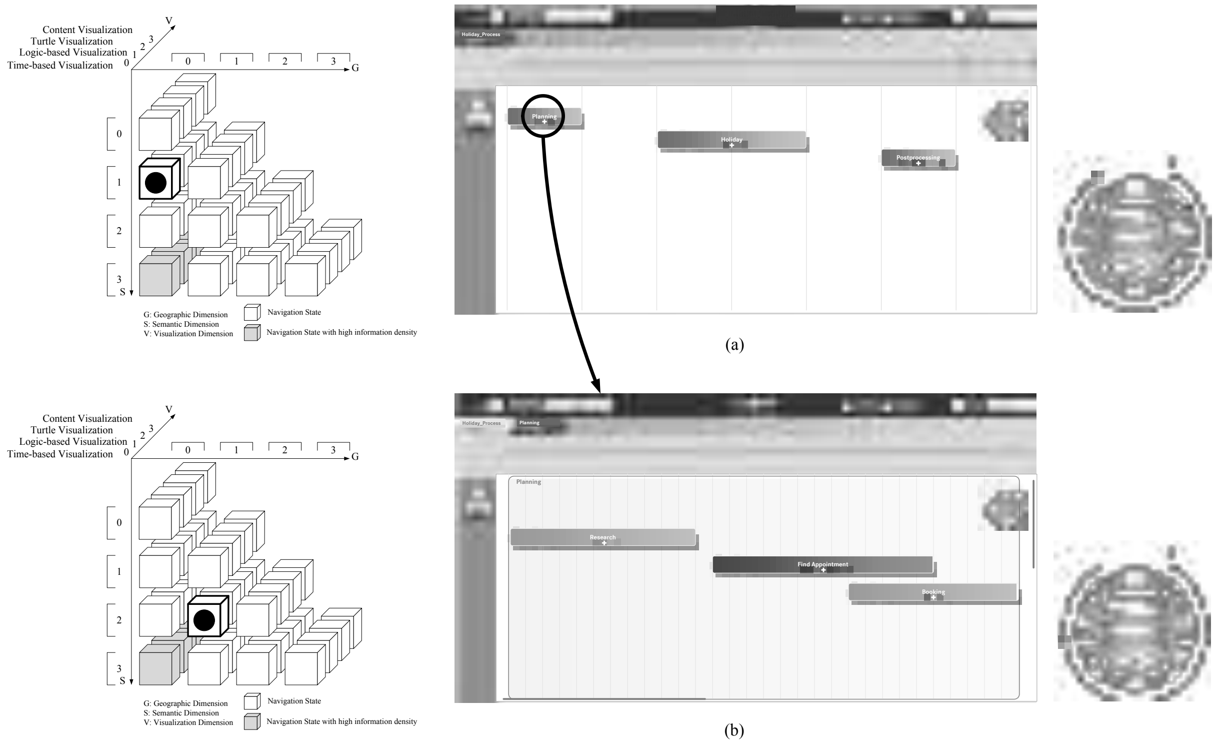


Figure 9.8: Combining the semantic and geographic dimension.

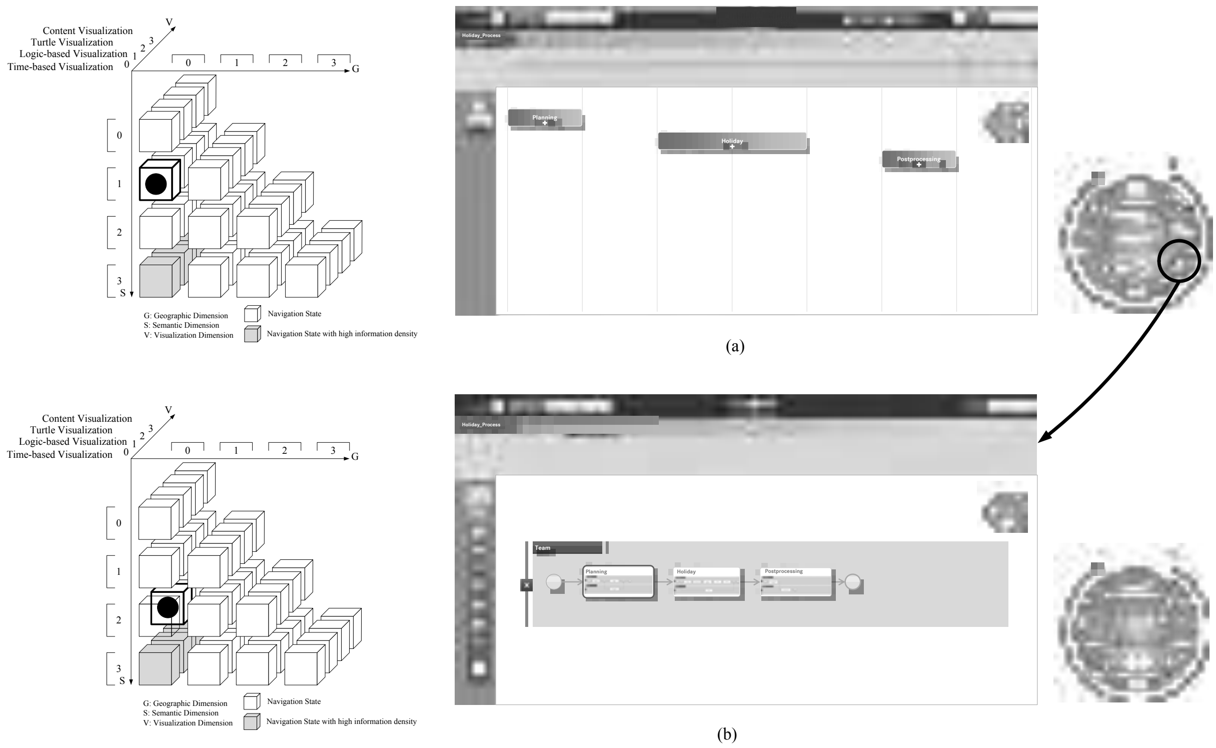


Figure 9.9: Combining the semantic and visualization dimension.

a 2-dimensional process interaction combining the semantic and geographic dimension. ProNavigator is further used to evaluate ProNaVis concepts (results can be found in Chapter 10).

9.2 Compass

This section introduces *Compass*, a tool for modeling process landscapes and navigating within them. Compass has been developed in cooperation with the research and development department of a large automotive Original Equipment Manufacturer (OEM). Specifically, it implements ProNaVis navigation concepts and applies them to engineering processes in the area of E/E car components [MHHR06]. Compass picks up user interface concepts developed for ProNavigator, focusing on the feasibility and ease of use of the developed approach. Hence, to reduce complexity, navigation options are restricted to a certain extent.

Department	Employees	Process Models	Documents	Area
Business Unit A	257	50	290	Bus
Business Unit B	47	15	60	Truck
Business Unit C	37	23	30	Car
Business Unit D	23	4	10	Car

Table 9.2: Details on the use of Compass.

Compass is currently used by 4 business units (cf. Table 9.2). 364 employees are working with the tool. The process model collections maintained by Compass comprise between 4 and 50 process models; thereby, the models have between 8 and 37 process tasks depending on the business unit. In total, 390 documents (i.e., data objects), such as guidelines, checklists and handbooks, are considered.

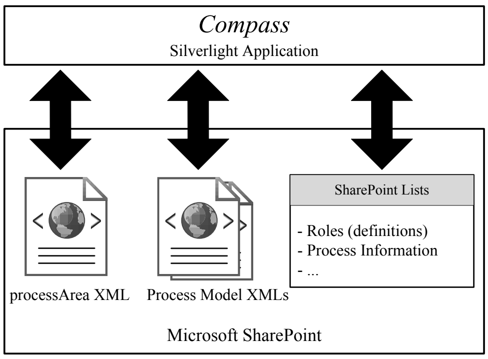


Figure 9.10: Conceptual architecture of Compass.

Compass is a Silverlight[1] application running in a SharePoint[2] environment (cf. Figure 9.10). Process models are integrated using XML files (cf. Chapter 4). Additionally, Compass makes use of SharePoint lists for storing global information (e.g., role descriptions or process information). Amongst others, this allows for unique role definitions across the entire process model collection. Moreover, global information might be created, read and edited directly using SharePoint, i.e., Compass is not needed for this purpose.

[1] Microsoft Silverlight: http://www.microsoft.com/silverlight/
[2] Microsoft SharePoint: http://office.microsoft.com/en-us/sharepoint/

For integrating process models, Compass provides sophisticated features. In particular, the entire navigation space can be modeled with Compass. This is useful if process models cannot be integrated automatically, e.g., due to a paper-based documentation (cf. Chapter 1). Besides, process areas can be explicitly modeled as well. Further note that in Compass the modeled navigation space is not limited by a fixed number of semantic levels. Instead, the user may dynamically add additional process areas at any point in time, i.e., the semantic dimension may be extended.

Figure 9.11: Compass user interface.

9.2.1 User Interface Design

Compass comprises five major areas (cf. Figure 9.11): First, the *process management area* (A) provides general management functions. Second, the *navigation area* (B) features navigation support, such as a breadcrumb navigation concept. Third, the *orientation area* (C) provides visualization-specific information, e.g., a timeline for a time-based visualization. Fourth, the *tool area* (D) comprises functions for modeling process models. Content, i.e., process models and process information, is provided in the *content area* (E).

Unlike for ProNavigator, we do not provide a navigation element. Instead, main interaction concepts are directly integrated with Compass and users interact with the visualized objects on the screen.

9.2.2 Example

The process model collection we use to illustrate Compass stems from the software development process in the E/E area of the automotive OEM. The navigation space from Figure 9.12 has been created using Compass. It comprises two levels for two process areas (0 and 1), one level for root nodes (2), one level for the swimlanes (3), and one level for single process events, tasks and gateways (4) (cf. Figure 9.12).

In the following, we explain the navigation functionality of Compass in more detail.

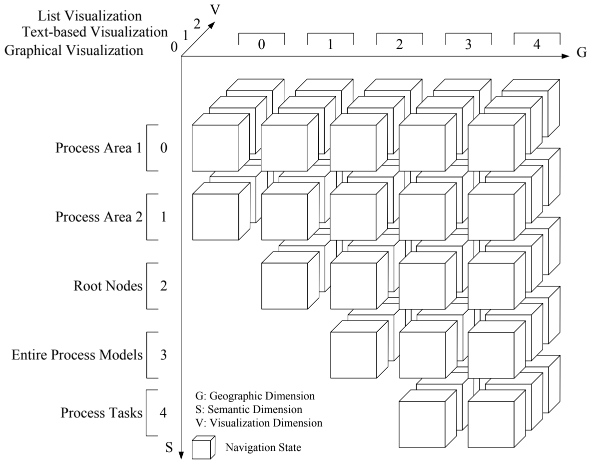


Figure 9.12: Exemplary part of a process space.

9.2.3 The Semantic Dimension

Based on experiences we gathered with ProNavigator (cf. Chapter 10), navigation along the semantic dimension can be considered as complex for users. Therefore, Compass allows for 2-dimensional interactions along the semantic and geographic dimension. In this context, it omits the navigation element introduced in ProNavigator. Instead, it enables direct interaction with the process contents, e.g., visualized objects. For example, when double clicking on an object, a 2-dimensional interaction, changing the semantic and geographic dimension at the same time, is triggered (*NavReq #6*). Figure 9.13 illustrates this kind of interaction. Starting with navigation state $(1, 0, 0)$ (cf. Figure 9.13a), a 2-dimensional navigation to process area *requirements engineering* results in navigation state $(2, 1, 0)$ (cf. Figure 9.13b).

In particular, Figure 9.13a shows a process model collection with four process areas: *Preparation*, *Requirements Engineering*, *Development*, and *Testing*. Double clicking on the requirements engineering process area results in a transition to navigation state $NS(2, 1, 0)$. In turn, this state provides process root nodes (*General Specification*, *System Specification*, and *Component Specification*). At the same, Compass zooms on the requirements engineering process area. Furthermore, the breadcrumb navigation indicates that the user moved into the requirements engineering process area. In particular, a new element, representing the process area requirements engineering, has been added.

9.2.4 The Geographic Dimension

The geographic dimension may be separately adjusted in Compass as well. Users may manually define the areas to be displayed. For this purpose, a popup dialog with two sliders is provided. Figure 9.14a shows navigation state $(2, 1, 0)$ providing process root nodes in process area *Requirements Engineering*. For example, the user might be interested in one particular root node (i.e., *System Specification* in our example), and hence might want to zoom on the area, including the desired root node. Figure 9.14b shows the dialog that may be used to limit the area displayed. Two sliders allow defining the start and end point of the zoomed area. As orientation, a schematic representation of the timeline is provided. The result is presented in Figure 9.14c ($NS(2, 2, 0)$).

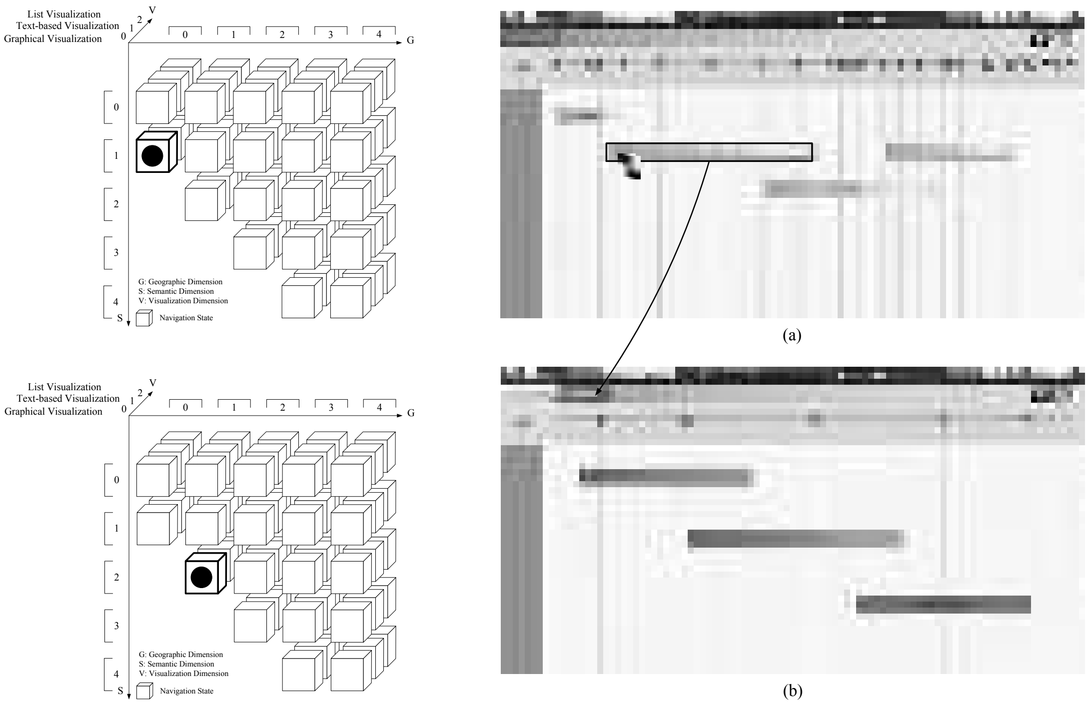


Figure 9.13: Semantic zooming in Compass.

9.2.5 The Visualization Dimension

Compass provides three visualizations. A logic-based one, which is called *graphical visualization* (corresponding to navigation state $(3, 2, 0)$) is shown in Figure 9.15a (*VisReq #2*). Figure 9.15b presents a *text-based visualization* focusing on textual descriptions (*VisReq #1*). Finally, Figure 9.15c provides a *list visualization* listing all objects of a navigation state. To switch between the different visualizations, three buttons are provided by the navigation area. By clicking on one of them, the visualization switches accordingly, i.e., the user might navigate to one of the following navigation states: $(3, 2, 0)$, $(3, 2, 1)$ or $(3, 2, 2)$.

Figure 9.15a shows a graphical visualization of the *General Specification* process model that corresponds to the requirements engineering process area (as indicated by the breadcrumb navigation). Note that data objects and data flow can be manually hidden from the users to keep the presented information as comprehensible as possible (*VisReq #5*). Switching to a text-based visualization provides the same information about the general specification process in textual manner (cf. Figure 9.15b) (*NavReq #1*). Thereby, the text is clustered into different areas, such as *Target* or *Description*, to improve readability. Finally, important meta data about the process model is presented within a box on the right side of the content area.

The visualization can be switched to a list visualization as well (cf. Figure 9.15c). In this case, all elements related to the general specification process, e.g., roles, process tasks, or data objects, are presented in one list. The latter can be filtered by the user according to the element type. For example, he may want to access related data objects and then gets a list of all documents, used in the general specification process.

Finally, the breadcrumb navigation shows that changes of the visualization do not affect the semantic or geographic navigation dimension.

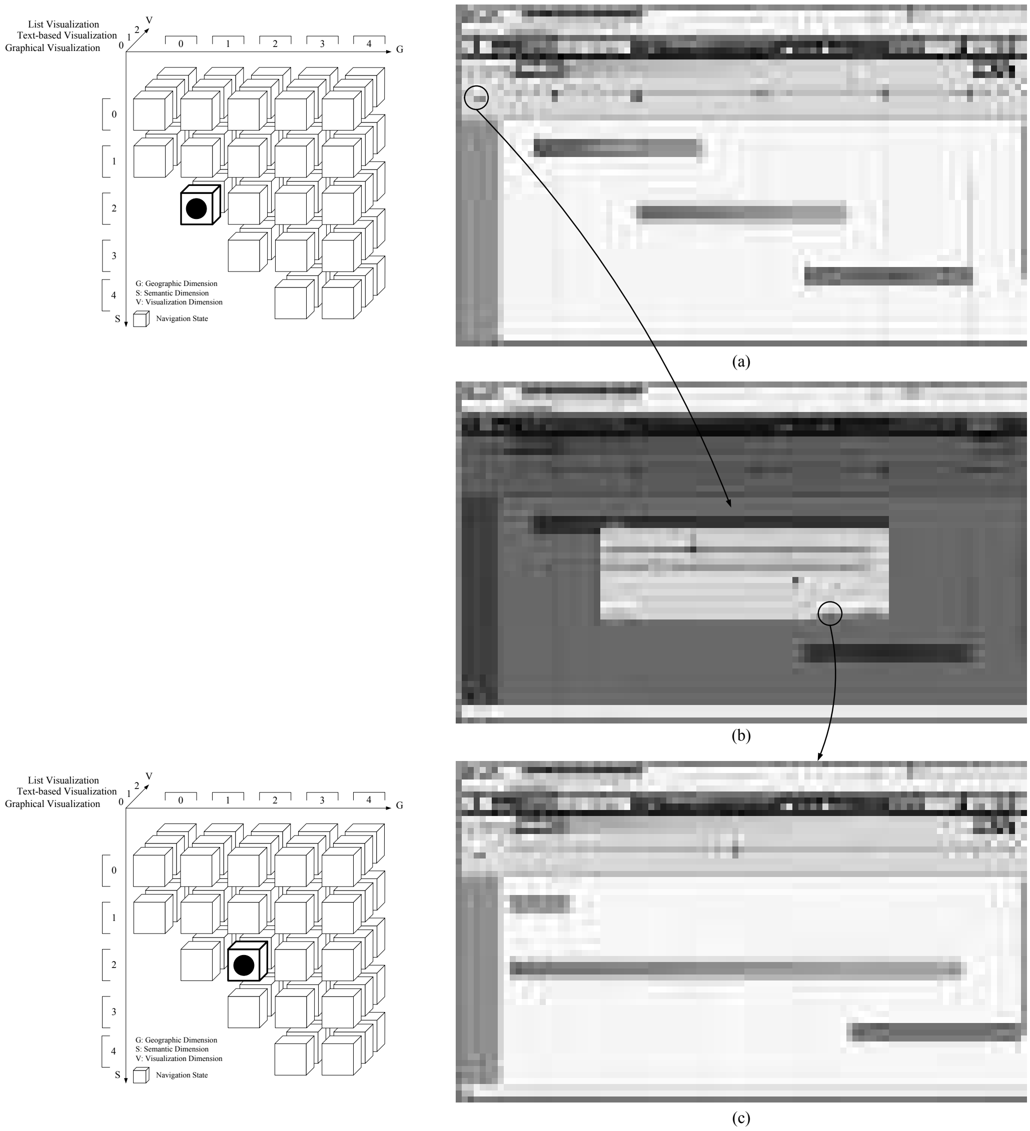


Figure 9.14: Geographic zooming in Compass.

Filter Mechanisms

In addition to the navigation space, Compass implements the filter concept introduced in Section 4.4.4, i.e., it allows filtering the objects corresponding to a particular navigation state. Objects might be filtered by various attributes. On one hand, default attributes can be used (e.g., *roles* or *participants*). On the other, attributes, manually defined by an administrator, may be applied as well (e.g., *milestones* or *project affiliations*).

Figure 9.16a shows the dialog used to adjust the filter criteria. The user may pre-specify certain filter adjustments, i.e., filter settings frequently used during his or her daily work (A). The main area of the filter dialog (B) allows adjusting each filter criterion separately. Alternatively, all filter criteria may be reset as well (B). Finally, the user chooses how the filtered objects shall be displayed (D). *Grey out* visualizes objects not matching the filter as greyed out objects, whereas *hide* completely hides the

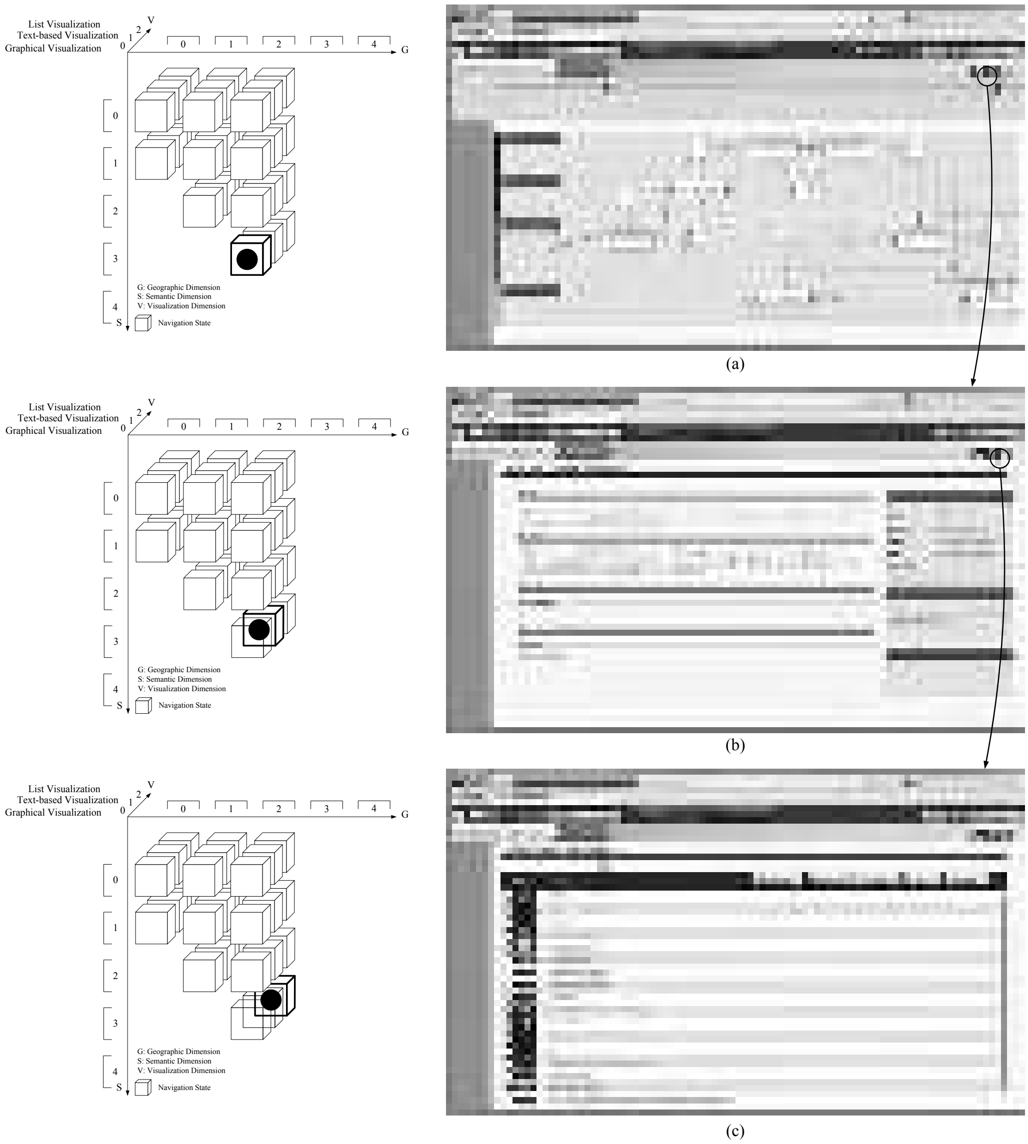


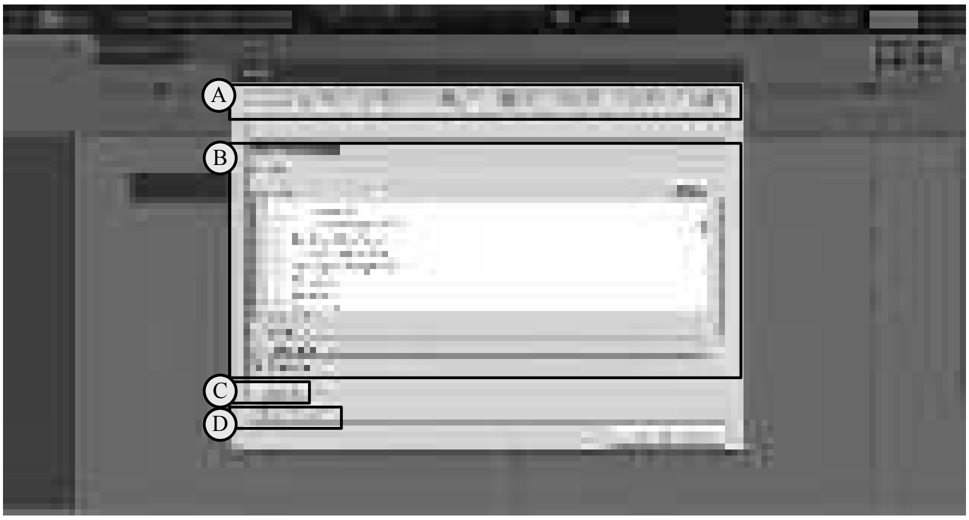
Figure 9.15: Switching between visualizations in Compass.

respective objects. Which filters are actually applied is indicated by blue stripes on the right of the filter attributes (cf. Figures 9.16b).

Figure 9.17a shows the result after filtering by *role*. As can be seen, only the process root node *general specification* matches the filter criterion. Thus, the other two process root nodes are greyed out. In tun, Figure 9.17b shows the result when applying visualization option *hide*. In this case, the two other process root nodes are completely hidden. Accordingly, this option reduces the number of displayed objects and hence information density.

9.2.6 Conclusion

Section 9.2 introduced Compass–a process navigation and modeling tool. It transfers several of the concepts developed in this thesis to industrial practice. In particular, it implements concepts of the



(a)

(b)

Figure 9.16: Different filter options in Compass.

ProNaVis navigation concept and applies it to process model collections from the automotive domain. Compass allows users to navigate in a 3-dimensional navigation space. Additionally, filter functionality have been added to Compass. Unlike ProNavigator, Compass constitutes a fully functional process navigation and modeling tool used by several hundred engineers in practice.

9.3 Discussion

We compare the functionality of the two prototypes and discuss it in more detail, referring to the navigation requirements presented in Chapter 2. Table 9.3 summarizes our conclusions.

NavReq#1 The level of detail of task descriptions is adjustable in both prototypes, either by using different semantic levels or applying different visualization types to a navigation state.

(a) (b)

Figure 9.17: Visualizing filter results.

NavReq#2 Presenting various process tasks on a detailed semantic level at a glance on an abstract geographic level is only possible when separating the semantic and geographic dimension. This concept is implemented in both prototypes. For the sake of usability, Compass limits it by providing 2-dimensional interactions along the semantic and geographic dimensions as well as 1-dimensional interactions along the geographic one.

NavReq#3 As all process models of a model collection are integrated in one navigation space, users can navigate from a given root node to every process area and, therefore, to each process model of the model collection. The navigation space is implemented by both prototypes.

NavReq#4 Compass organizes process information in lists. corresponding items of this list then represent documents, links, or plain text. Further, they may be manually linked to any process model from the model collection in terms of data objects. Thereby, the list is globally accessible from every process model of the collection.

NavReq#5 Compass provides a sophisticated role management concept. Roles are globally defined, including information about tasks, competencies and responsibilities.

NavReq#6 The ProNaVis concept allows navigating on different levels of detail, using the semantic and geographic dimensions. The separation of the two dimensions allows for the flexible navigation on any level of detail regarding the given process model collection. In turn, the combination of the two dimensions facilitates user navigation. Thus, both implementations meet this requirement.

VisReq#1 Both prototypes provide text-based visualization types, i.e., a turtle and content visualization. In combination with a high semantic level, in turn, these visualizations allows for very detailed textual task descriptions.

VisReq#2 Both, temporal and logical relations have been taken into account in the context of different visualizations. The time-based visualization solely focuses on temporal aspects, whereas the logic-based visualization focuses on logic relations. Both prototypes realize these visualizations.

VisReq#3 In Compass, process information (e.g., documents) can be enriched with meta data (e.g., *author*, *description* or *comments*). This information can be accessed with the right side bar and may be used to identify a certain document.

VisReq#4 In Compass, each role description includes contact persons, e.g., *experts*, *participants*, or *responsible* persons. These persons are directly accessible through their phone number and email address. To intuitively identify roles, a color concept is used, assigning a color to each role, which is globally used across the entire process model collection.

VisReq#5 Compass limits the navigation along the semantic dimension to avoid navigating to states with high information density. Therefore, 2-dimensional interactions are supported, i.e., when navigating along the semantic dimension, the geographic level is adjusted accordingly.

Req #	**Requirement**	ProNavigator	Compass
NavReq #1	Depending on a process participant's experience, the level of detail regarding a process task should be adjustable.	✓	✓
NavReq #2	Process participants should be able to adjust the level of detail regarding process model collection in order to obtain a quick overview on a specific task that is currently executed.	✓	
NavReq #3	Users should be enabled to access process tasks in other process areas.	✓	✓
NavReq #4	Relevant process information must be accessible at the level of single process models from the process model collection.		✓
NavReq #5	Roles must be globally defined in a detailed manner.		✓
NavReq #6	Process participants must be able to access process models on different levels of detail.	✓	✓
VisReq #1	Task descriptions must be documented in a well understandable manner.	✓	✓
VisReq #2	Temporal and logical dependencies must be considered when visualizing processes.	✓	✓
VisReq #3	Complex process information must be visualized in a comprehensible manner.		✓
VisReq #4	Information about roles must be intuitively identifiable.		✓
VisReq #5	The amount of visualized information should not overload process participants.		✓

Table 9.3: Overview on how the implementations meet the derived requirements.

9.4 Summary

This chapter introduced two proof-of-concept prototypes. First, we introduced *ProNavigator*, a click-prototype, implementing the navigation space with all three navigation dimensions. The overall goal was to create a realistic navigation feeling for process participants. Second, *Compass* was introduced, which supports knowledge workers in accessing complex process model collections during E/E component development. Compass implements the ProNaVis framework, including 2-dimensional interactions and filter functions. Compass was used by 4 different business units in the automotive domain.

Both prototypes constitute the basis for two user experiments described in Chapters 10 and 11. The first experiment investigates the 3-dimensional navigation approach to a static, 1-dimensional one. The second experiment focuses on the visualization of process models in the logic-based visualization, as this is the most common notation for process models.

10 Experiment 1: Process Navigation

10.1 Motivation

To validate the ProNaVis framework, this chapter[1] presents results from a controlled user experiment involving 27 subjects from the automotive domain. As main goal we want to investigate the benefit of the 3-dimensional navigation concept compared to a static, 1-dimensional navigation concept as used in common process portals. Additionally, we investigate how initial support of subjects working with the 3-dimensional navigation concept effects the experiment results. The research question of this experiment is as follows:

Is 3-dimensional process navigation concept (with and without initial support during introduction) more suitable for navigating in process model collections compared to a static, 1-dimensional navigation concept? If 'yes', how strong is this difference?

On one hand, we assume that providing three navigation dimensions makes navigation more difficult to learn and less intuitive, since the number of navigation options increases. On the other, more sophisticated navigation options arise, allowing for more precise navigation. Therefore, we assume best results for subjects working with the 3-dimensional navigation concept in conjunction with an additional support during introduction.

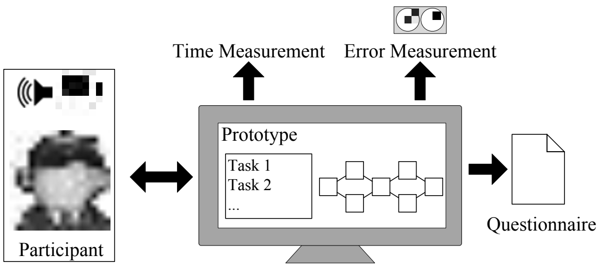


Figure 10.1: Experiment setup.

Subjects are asked to perform navigational tasks using the *ProNavigator* prototype (cf. Figure 10.1). In particular, they shall navigate to different navigation states within a given navigation space. Thereby, execution times of the navigational tasks are measured. Additionally, the number of errors is measured (e.g., a subject has not properly finished a navigational task). At the end, subjects fill out a questionnaire rating their subjective impressions regarding the tested prototype. The subjects have been tested separately, and the sessions have been recorded on video.

[1]The chapter is based on the following referred paper [HMMR14]:
Markus Hipp, Bernd Michelberger, Bela Mutschler, and Manfred Reichert. *Navigating in Process Model Repositories and Enterprise Process Information.* in: Proc 8th Int'l Conf on Research Challenges in Information Science (RCIS'14), IEEE, pp. 1–12, 2014

The remainder of this chapter is structured as follows. Section 10.2 describes the used design of the experiment. In Section 10.3, the hypotheses to be investigated are presented. Section 10.4 describes how the experiment is performed. Section 10.5 shows how experiment data is prepared, whereas Section 10.6 presents experimental results. Section 10.7 discusses threats of validity. Finally, Section 10.8 discusses results and Section 10.9 concludes the chapter.

10.2 Experimental Design

When designing the experiment, we took the following criteria into account [BRWSH86]:

- The design of the experiment shall allow for the collection of as much data as possible with respect to the major goals of the experiment.
- The collected data shall be unambiguous.
- The experiment shall be feasible for a given setting.

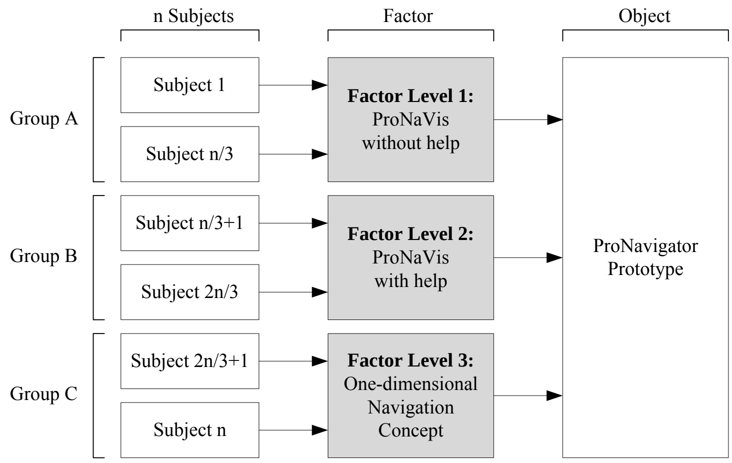


Figure 10.2: The experiment design.

Following these design criteria, we conduct a *controlled single factor experiment* [JM01, WRH+12] (cf. Figure 10.2). Subjects are randomly divided into 3 groups consisting of 9 members each. There are two *experimental groups* (groups A and B) and one *control group* (group C). Both *experimental groups* work with ProNavigator (cf. Section 9.1) and thus with the 3-dimensional ProNaVis navigation concept (*experimental system*), whereas the *control group* works with a different implementation of ProNavigator providing only a static, 1-dimensional navigation concept (*control system*). In the control system, navigation is limited to the geographic dimension; i.e., both the semantic and the visualization dimension are "hard-wired" with the geographic dimension. On an abstract geographic level, for example, contents are always presented using a time-based visualization. On a more detailed geographic level, in turn, the visualization switches to a logic-based one. Note that this functionality exactly corresponds to the functionality of the process portal presented in Chapter 1.

The subjects, object, and selected variables of the experiment are as follows:

Subjects: The subjects are 27 engineers from the automotive domain. Subjects are divided into 3 groups, of which each comprises 9 members. Subjects are randomly assigned to the groups prior to the start of each experiment.

Object: The object to be evaluated by each subject is the process navigation prototype ProNavigator (cf. Section 9.1).

Factor and Factor Levels: The factor is the *process navigation concept* applied to ProNavigator. The considered factor levels include *3-dimensional navigation* and *1-dimensional navigation*. They are realized by two different implementations of ProNavigator. Groups A and B work with the ProNavigator version that allows for 3-dimensional navigation (cf. Section 9.1). In turn, group C works with the ProNavigator version only allowing for 1-dimensional navigation.

Dependent Variables: The following dependent variables are considered to investigate how users perceive the prototype [MRC07, Men08, Nor88]: *comprehensibility*, *traceability*, *simplicity*, *intuitiveness*, *interest*, and *stimulation*. Additionally, execution times for navigational tasks are logged during the experiment to investigate how fast the different tasks are accomplished [HFL12]. Finally, the number of errors made during the experiment is considered as a measurement for subjects' performance [ZPR+12].

Instrumentation: To collect data, we use an online tool[2] providing an implemented stop watch as well as automated error recognition functions. The tool further allows collecting qualitative feedback, i.e., a structured questionnaire can be provided. For video and audio recording, CamStudio[3] is used.

Data Analysis Procedure: For performing data analysis, well-established statistical methods and standard metrics are applied, i.e., *Mann-Whitney U test*, *Kolmogorov-Smirnov test*, and *Shapiro-Wilk test* (cf. Section 10.4).

10.3 Hypothesis Formulation

To address the research question of the experiment, we present 3 hypotheses clustering response variables (cf. Figure 10.3): (H1) *Understandability*, (H2) *Usability*, and (H3) *Process Navigation Speed*. We consider *understandability* of the navigation concept, which is a crucial factor when evaluating process models [MRC07, RM11]. Furthermore, *usability* aspects play a role when designing user interaction concepts [HH93, ND86]. The conducted case studies [HMR11b] have confirmed that quickly finding information is crucial for process participants. Therefore, execution times of experiment tasks are measured as indicator for process model understandability [MMRS09, MS08]. Therefore, *process navigation speed* is considered as hypothesis as well. All response variables have been selected based on their relation to the respective hypothesis.

[2] http://onlineumfrage.com
[3] CamStudio - Open Source: http://camstudio.org/

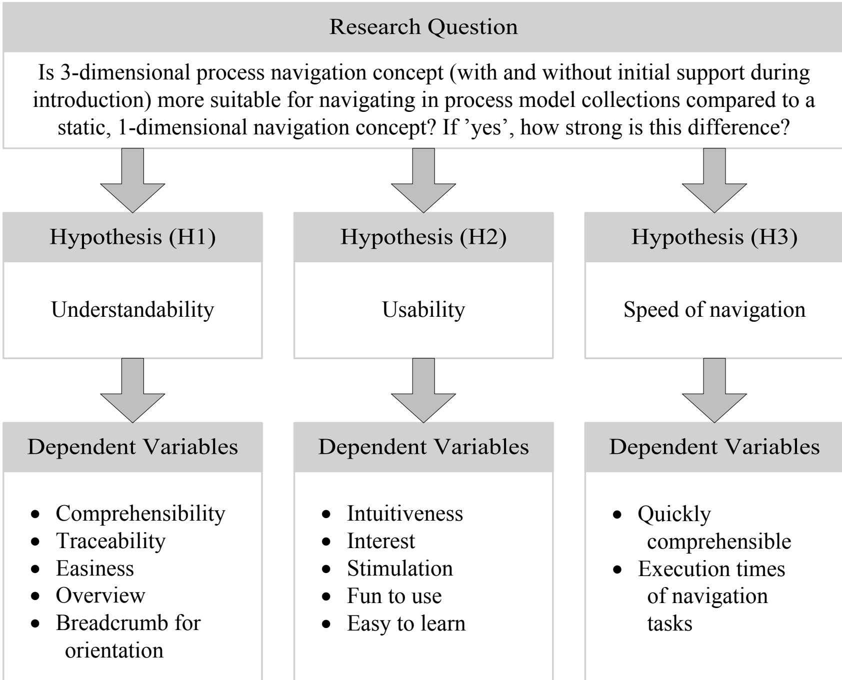


Figure 10.3: Deriving the Response Variables.

H1: Understandability We investigate and compare the understandability of the 3-dimensional navigation concept with and without initial support and the 1-dimensional navigation concept:

- **0-Hypothesis** $H_{0,1}$**:** There is no significant difference in the understandability of a 3-dimensional navigation concept (with and without initial support) and a 1-dimensional navigation concept.
- **Alternative Hypothesis 1** $H_{1,1,1}$**:** The 3-dimensional navigation concept with initial support is significantly better understandable than the 3-dimensional navigation concept without initial support.
- **Alternative Hypothesis 2** $H_{1,1,2}$**:** The 3-dimensional navigation concept with initial support is significantly better understandable than the control concept.
- **Alternative Hypothesis 3** $H_{1,1,3}$**:** The 3-dimensional navigation concept without initial support is significantly better understandable than the control concept.

H2: Usability We investigate and compare the usability of a 3-dimensional navigation concept (with and without initial support) and a 1-dimensional one:

- **0-Hypothesis** $H_{0,2}$**:** There is no significant difference regarding the usability of a 3-dimensional navigation concept (with and without initial support) and a 1-dimensional navigation concept.
- **Alternative-Hypothesis 1** $H_{1,2,1}$**:** The 3-dimensional navigation concept with initial support is significantly better usable than the 3-dimensional navigation concept without initial support.
- **Alternative-Hypothesis 2** $H_{1,2,2}$**:** The 3-dimensional navigation concept with initial support is significantly better usable than the control concept.

- **Alternative-Hypothesis 3** $H_{1,2,3}$**:** The 3-dimensional navigation concept without initial support is significantly better usable than the control concept.

H3: Process Navigation Speed We investigate whether or not a task can be accomplished faster using a 3-dimensional navigation concept (with and without initial support) compared to a 1-dimensional navigation concept:

- **0-Hypothesis** $H_{0,3}$**:** There is no significant difference in how fast a task can be performed with a 3-dimensional navigation concept (with and without initial support) compared to a 1-dimensional navigation concept.
- **Alternative Hypothesis** $H_{1,3,1}$**:** With the 3-dimensional navigation concept with initial support, a task can be solved significantly faster than with the 3-dimensional navigation concept without initial support.
- **Alternative Hypothesis** $H_{1,3,2}$**:** With the 3-dimensional navigation concept with initial support, a task can be solved significantly faster than with the control concept.
- **Alternative Hypothesis** $H_{1,3,3}$**:** With the 3-dimensional navigation concept without initial support, a task can be solved significantly faster than with the control concept.

10.4 Experiment Execution

Part 1 of the experiment (cf Figure 10.4) introduces experiment goals and procedures to the subjects. Afterwards, the subjects must perform three introductory navigation tasks in order to become familiar with the respective navigation concept. In part 2, in turn, the subjects must answer demographic questions regarding their age, gender, experience with process navigation, and experience with process modeling notations.

The third part of the experiment comprises 14 process navigation tasks. For example, subjects must navigate to a specific process and search for a related document. The navigational tasks have been chosen based on typical use cases (cf. Chapter 1). In order to allow participants to focus on navigation, the used process model collection was about the planing and executing a holiday trip and was easy to understand [SB06]. Exemplary experiment tasks were:

- Which processes are related to role *team leader*?
- Which processes are overlapping in time within process area *planning*?
- Which process task needs document *flight schedule* as input?

Approximately, the experiment session takes 45 minutes per subject. When performing the experiment, subjects are captured on video. For seven specific navigation tasks, the execution times are recorded as well. After finishing all tasks, the subjects must fill out a questionnaire regarding their subjective impressions on the navigation concept (cf. Figure A.2.3). Specifically, they use a 5-step Likert scale ranging from 1 (*I totally disagree*) to 5 (*I totally agree*) in order to rate the according response variables.

During the introduction phase of the experiment (part 1), group A receives a paper-based introduction to the experimental system (cf. Figure A.2.2). The ProNavigator functions are explained to group A, using textual descriptions and illustrating pictures. Group B, in turn, receives the same introduction, but

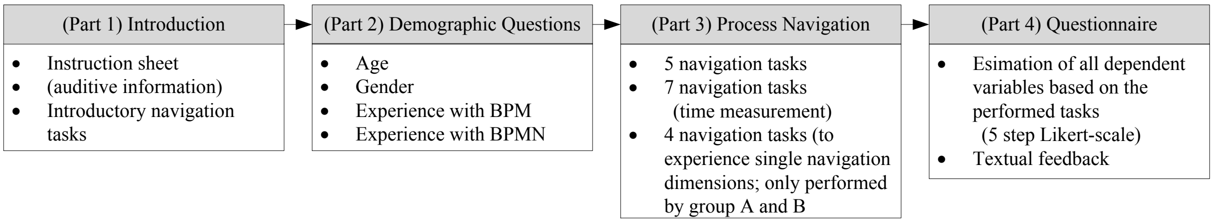


Figure 10.4: Execution of the single factor experiment.

with additional auditive information given by the experimenter. The introduction of the control group (group C) is accomplished also in text-based style (cf. Figure A.2.1). However, since the functions of the control concept are very limited, we assume that all subjects fully understand them. Thus, we assume that Group B exhibits the same level of knowledge regarding the ProNaVis functions, as group C has on the control concept. Table 10.1 provides an overview on how the experiment introduction is accomplished for the three groups.

	Group A	**Group B**	**Group C**
Introduction	text-based	text-based and auditive	text-based
Prototype Functions	3-dimensional: Semantic Dimension Geographic Dimension Visualization Dimension	3-dimensional: Semantic Dimension Geographic Dimension Visualization Dimension	1-dimensional: Hard-wired navigation dimensions

Table 10.1: The experimental groups.

10.5 Preparation of Data

Before presenting experiment results, experiment data is analyzed and prepared in several steps.

10.5.1 Data Validation and Analysis

First of all, experiment data is *collected* and validated in respect to its *plausibility.* The experiment data is collected by the used online tool. Collected data comprises the time to perform the navigation tasks as well as the subjects' evaluations in terms of the questionnaire results.

Data plausibility is analyzed using *box-wisker-plot diagrams* [Coo09]. Such diagrams visualize the distribution of a sample showing outliers. Thereby, a low number of outliers indicates plausible data (cf. Figure 10.5). Overall, the experiment data is plausible since only few (negligible) outliers can be observed.

However, we discard one data set, since the measured answer times seem to be too high (e.g., >300 seconds for a task performed in less than 40 seconds on average). In this particular case, the captured video shows that the subject did not properly follow the instructions when performing the respective navigation task, i.e., the subject was externally influenced during task execution. Note that the subject shows average execution times regarding other navigation tasks.

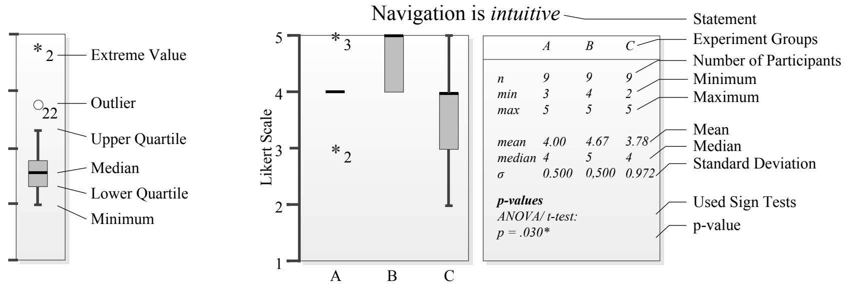


Figure 10.5: Box-Wisker-Plot diagram.

10.5.2 Developing Scales

In this section, we develop scales for each of our hypotheses. A scale combines a group of response variables (items) into a single, more aggregated variable [Mic90]. To do so, a prerequisite is that all items show high reliability [Kli99], i.e., all items measure the same general topic. Therefore, *Cronbach's α* is calculated.[4] Table 10.2 shows the scales used in the experiment.

Hypothesis	**Scale**	**Items**	**Cronbach's α**
H1	*Understandability*	Comprehensibility	.71
		Traceability	
		Easiness	
		Overview	
		Breadcrumb for orientation	
H2	*Clarity*	Intuitiveness	.78
		Interest	
		Stimulation	
		Fun to use	
		Easy to learn	
H3	*Speed of navigation*	Quickly comprehensible	only 1 item

Table 10.2: Scales used in experiment 1.

As hypothesis 3 only comprises one item which is measured by subjects, no scale is used. To investigate speed of navigation, we also rely on time measurements for seven single experiment tasks. For aggregation purposes, we also consider the overall execution times (i.e., the sum of execution times of all seven experiment tasks).

10.5.3 Control Variables and Correlations

First of all, we investigate, whether the control variables reveal significant differences between the three groups. The applied *t-tests* between all combinations of groups do not reveal any significant differences (cf. Table 11.4). Therefore, none of the independent variables has to be considered in the following significance tests. Note that control variables #3 and #4 are not combined to a scale due to a low reliability (*Cronbach's α=.57*).

Second, we investigate, whether the dependent variables correlate with the control variables. In case of a correlation, we have to consider the dependent variables as *covariant* in the significance tests. As can be seen in Table 11.5, none of the dependent variables shows a significant correlation to one of the control variables. Therefore, the significant tests can be performed without considering a covariant.

[4] According to [Kli99], α>0.6 indicates acceptable and 0.7< α<0.9 indicates good reliability.

#	Control Variable	Group	N	M	SD	t-test
1	How old are you?	A	9	32.56	8.96	A/B: $p_2 = .71$
		B	9	30.78	10.58	B/C: $p_2 = .92$
		C	9	30.44	4.64	A/C: $p_2 = .54$
2	Are you experienced in process modeling? (1=yes, 2=no)	A	9	1.33	0.50	A/B: $p_2 = .63$
		B	9	1.22	0.44	B/C: $p_2 = .56$
		C	9	1.11	0.333	A/C: $p_2 = .29$
3	Please estimate your experience with process modeling? (5=very experienced, 1=no experience)	A	6	3.67	0.52	A/B: $p_2 = .87$
		B	7	3,57	1.27	B/C: $p_2 = .44$
		C	8	4.00	0.76	A/C: $p_2 = .37$
4	How well do you know BPMN? (5=very well, 1=no at all)	A	4	3.00	0.82	A/B: $p_2 = .87$
		B	7	2.86	1.57	B/C: $p_2 = .30$
		C	8	3.62	1.19	A/C: $p_2 = .37$

Table 10.3: Differences of control variables between groups.

#	Control Variable		Scale H1	Scale H2	Item H3
1	How old are you?	Sig. (2-tailed)	A/B: $p_2 = .93$	A/B: $p_2 = .25$	A/B: $p_2 = .32$
			B/C: $p_2 = .62$	B/C: $p_2 = .77$	B/C: $p_2 = .50$
			A/C: $p_2 = .77$	A/C: $p_2 = .29$	A/C: $p_2 = .20$
2	Are you experienced in process modeling?	Sig. (2-tailed)	A/B: $p_2 = .22$	A/B: $p_2 = .52$	A/B: $p_2 = .80$
			B/C: $p_2 = .75$	B/C: $p_2 = .11$	B/C: $p_2 = .30$
			A/C: $p_2 = .92$	A/C: $p_2 = .58$	A/C: $p_2 = .92$
3	Please estimate your experience with process modeling?	Sig. (2-tailed)	A/B: $p_2 = .66$	A/B: $p_2 = .87$	A/B: $p_2 = .84$
			B/C: $p_2 = .33$	B/C: $p_2 = .97$	B/C: $p_2 = .93$
			A/C: $p_2 = .11$	A/C: $p_2 = .20$	A/C: $p_2 = .83$
4	How well do you know BPMN?	Sig. (2-tailed)	A/B: $p_2 = .66$	A/B: $p_2 = .99$	A/B: $p_2 = .63$
			B/C: $p_2 = .86$	B/C: $p_2 = .60$	B/C: $p_2 = .73$
			A/C: $p_2 = .52$	A/C: $p_2 = .51$	A/C: $p_2 = .77$

Table 10.4: Correlations between control variables and dependent variables.

10.5.4 Data Analysis

Main goal of the experiment is to investigate whether or not there is a difference between the experiment results of the three groups. More specifically, we analyze the three hypotheses. Initially, the respective 0-hypotheses are considered as correct. By applying significance tests (e.g., *t-test* or an additional sign test if the t-test fails) we are able to assess whether the means of two samples statistically differ from each other [Coo09]. A successful test rejects the 0-hypothesis. Specifically, the tests are executed based on a 5% significance level (α=0.05). All used tests are explained in detail in the following.

Explorative Data Analysis: A *Kolmogorov-Smirnov test* and a *Shapiro-Wilk test* are used to analyse whether a data set is well-modeled by a normal distribution (*test of normality*). Not all data sets in our experiment show normal distribution. The used significance tests are described in the following.

Significance Tests for Data Sets with Normal Distribution: Data samples from normally distributed data are analyzed using a *t-test*. With this test, the statistical difference between different data samples is measured.

Significance Tests for Data Sets not showing Normal Distribution: We use the *Mann-Whitney U test* and the *Kruskal-Wallis test* to analyze significances in non-normally distributed data sets.

Statistical Measures: For all significance tests, we provide descriptive statistics of the samples (number n, the *mean*, the *median*, the biggest (*max*) and smallest (*min*) value, and the *standard deviation* σ). For reporting results from significance tests we provide the p-values[5] and respective values according to the APA style [Fie13].[6]

10.6 Results

This section presents results of the experiment in respect to the three hypotheses.

10.6.1 Understandability

The developed scale (cf. Figure 10.6) shows that all navigation concepts are very *understandable* (*mean* group A: $M = 4.13$, *standard deviation*: $SD = 0.36$, group B: $M = 4.6, SD = 0.39$, and group C: $M = 4.22, SD = 0.73$). Comparing the two experimental groups with the control group, only group B shows a significantly higher result (B/C: $U = 22.00, z = -1.65, p_1 = .049^*, r = -0.39$)[7]. Group A, however, shows even worse results compared to group C. Considering both experimental groups, subjects with initial support (group B) rate the 3-dimensional navigation concept significantly higher compared to subjects using the 3-dimensional navigation concept without initial support ($U = 16.50, z = -2.15, p_1 = .02^*, r = -0.51$).

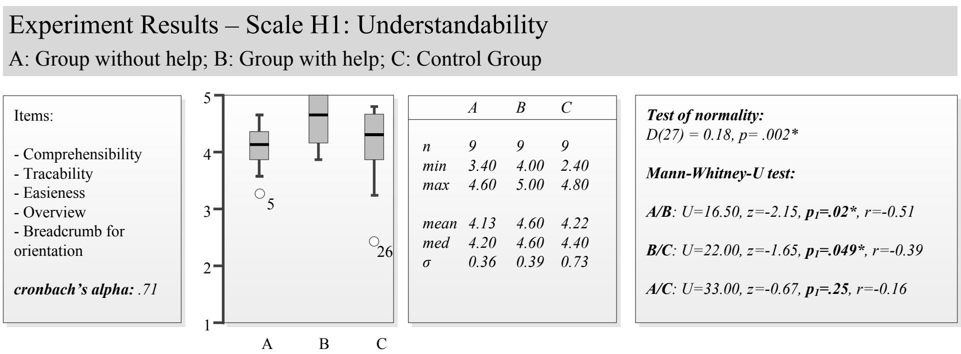


Figure 10.6: Scale for hypothesis H1.

Despite the more complex navigation concept provided to groups A and B, the latter shows the highest ratings regarding understandability. Group A, which only received a paper-based introduction, almost

[5] p_2 represents the p-value for 2-tailed tests, and p_1 for 1-tailed tests.
[6] APA Style: http://www.apastyle.org/
[7] Using directed hypotheses, we can use 1-tailed significance tests. Therefore, p_1 represents the halved p_2 value [Kli99].

shows identical results compared to control group C. We may conclude that the 3-dimensional navigation concept is only understandable if users receive a detailed introduction on the provided functions. However, if the latter applies, the results are significantly better. Results show that the understandability is perceived significantly better by group B compared to the other groups. Based on the presented results, we can reject 0-hypothesis $H_{0,1}$. Therefore, two alternative hypotheses can be accepted ($H_{1,1,1}$, and $H_{1,1,2}$).

> $H_{1,1,1}$: The 3-dimensional navigation concept with initial support is significantly better understandable than the 3-dimensional navigation concept without initial support.✓
> $H_{1,1,2}$: The 3-dimensional navigation concept with initial support is significantly better understandable than the control concept.✓
> $H_{1,1,3}$: The 3-dimensional navigation concept without initial support is significantly better understandable than the control concept.✗

10.6.2 Usability

Results from the developed scale (cf. Figure 10.7) show that all navigation concepts provide high *usability* (group A: $M = 4.02, SD = 0.46$, group B: $M = 4.56, SD = 0.41$, and group C: $M = 3.87, SD = 0.76$). Combining the two experimental groups with the control group, only group B shows a significantly higher result compared to group C ($U = 19.00, z = -1.92, p_1 = .03^*, r = -0.45$). Additionally, subjects from group B rate the usability of the experimental concept significantly higher than subjects from group A ($U = 14.50, z = -2.33, p_1 = .01^*, r = -0.55$).

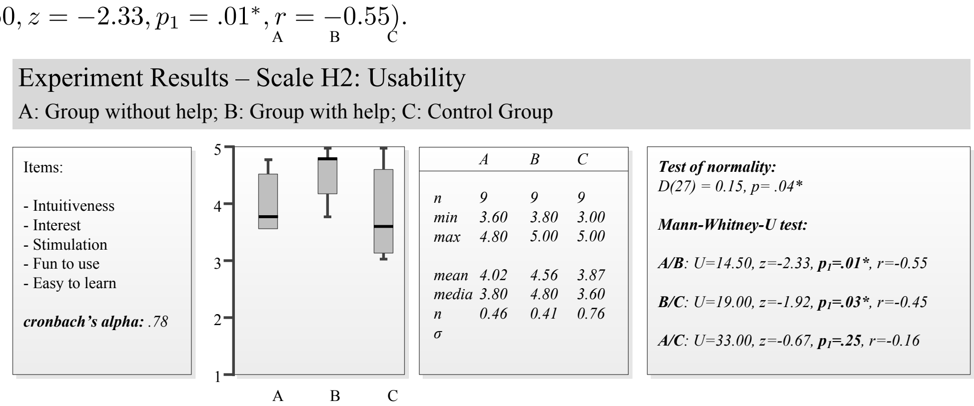


Figure 10.7: Scale for hypothesis H2.

Based on the results, including two significantly better results for group B, we can reject 0-hypothesis $H_{0,2}$. In turn, alternative hypotheses $H_{1,2,1}$ and $H_{1,2,2}$ can be accepted.

> $H_{1,2,1}$: The 3-dimensional navigation concept with initial support is significantly better usable than the 3-dimensional navigation concept without initial support.✓
> $H_{1,2,2}$: The 3-dimensional navigation concept with initial support is significantly better usable than the control concept.✓
> $H_{1,2,3}$: The 3-dimensional navigation concept without initial support is significantly better usable than the control concept.✗

10.6.3 Process Navigation Speed

In this section, results regarding the third hypothesis are investigated. The results are mainly based on time measurements of the navigation tasks the subjects have to perform during the experiment. Additionally, subjects are asked about their subjective impressions on how fast they were able to perform a particular navigation task. Figure 10.8 shows the results.

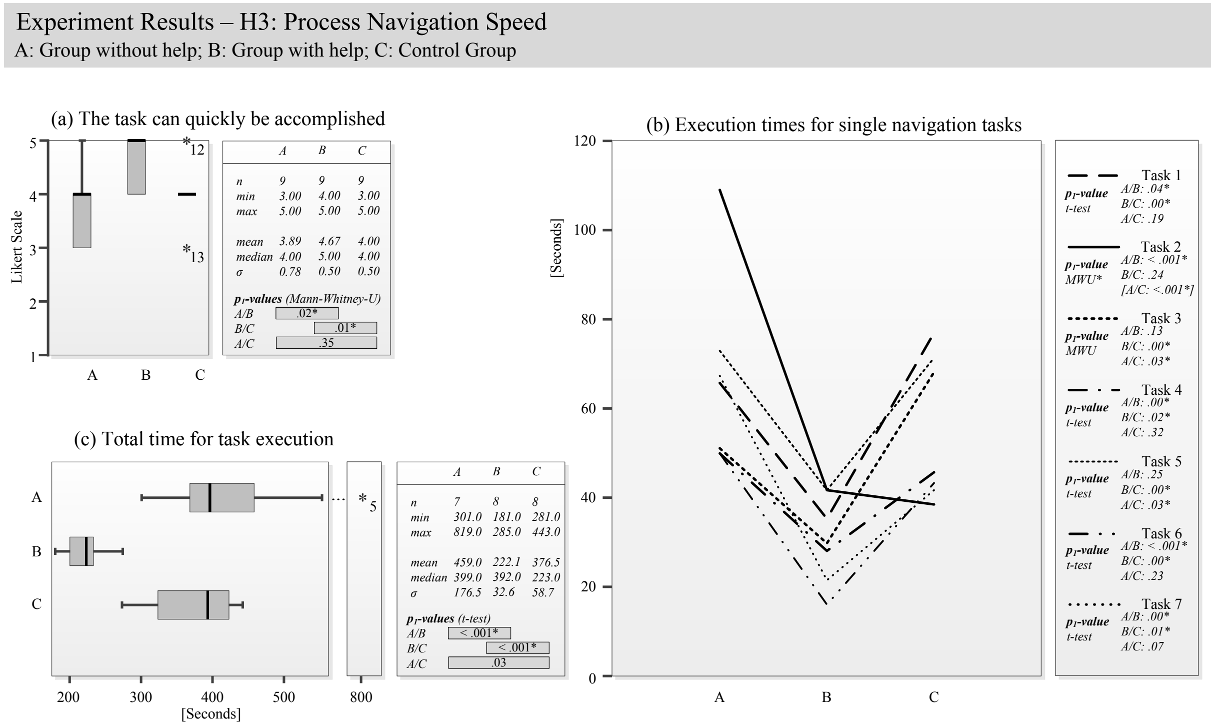


Figure 10.8: Results for hypothesis H3.

As can be seen in Figure 10.8a, group B has the subjective impression of being able to quickly accomplish the given tasks ($M = 4.67, SD = 0.50$). Groups A and C further have the impression of accomplishing their tasks pretty fast (group A: $M = 3.89, SD = 0.78$; group C: $M = 4.00, SD = 0.50$). However, group B provides a significantly higher rating compared to the other two groups (B/C: $U = 16.50, z = -2.41, p_1 = .01^*, r = -0.57$ and B/A: $U = 18.00, z = -2.15, p_1 = .02^*, r = -0.51$).

Interestingly, results from this subjective impression slightly correspond with time measurements ($r = 0.48, n = 16, p_2 = .06$), i.e., subjects having the feeling to perform tasks fast tend to actually perform them faster than other subjects. Except for task 2, all other tasks are performed significantly faster by subjects from group B compared to subjects from group C (cf. Figure 10.8b). Task 2 deals with the identification of different input documents of a single process task. As the control concept (i.e., the 1-dimensional navigation concept) only provides one static visualization for process tasks (including a list of all input and output documents), it is easy for subjects to identify the right documents. In turn, the ProNaVis navigation concept implemented in the experimental system provides three independent visualizations of process tasks, including visualizations, that do not comprise input documents as they focus on other information (e.g., temporal dependencies). Thus, few subjects, especially from group A, get stuck in these visualizations and are unable to find the required documents. In turn, subjects from group B are explicitly taught to switch visualizations in order to get specific information. As can be seen, subjects from group B perform significantly better in the context of this task compared to subjects from group A.

Combining all seven navigation tasks, the overall execution time of the three groups is shown in Figure 10.8c. As can be seen, subjects from group B perform all tasks significantly faster compared to the other groups (B/C: $F = 0.78, t(14) = -8.92, p_1 =< .001^*, r = 0.92$ and B/A: $F = 24.61, t(14) = 4.918, p_1 =< .001^*, r = 0.80$). In combination with the results from the subjective impression of the participants, we can reject 0-hypothesis $H_{0,2}$. In turn, we accept alternative hypotheses $H_{1,3,1}$ and $H_{1,3,2}$.

$H_{1,3,1}$: With the 3-dimensional navigation concept with initial support, a task can be solved significantly faster than with the 3-dimensional navigation concept without initial support.✓
$H_{1,3,2}$: With the 3-dimensional navigation concept with initial support, a task can be solved significantly faster than with the control concept.✓
$H_{1,3,3}$: With the 3-dimensional navigation concept without initial support, a task can be solved significantly faster than with the control concept.✗

10.6.4 Navigation Dimensions

We separately investigate the three navigation dimensions of the ProNaVis navigation concept. Note that only subjects who used ProNaVis during the experiment are considered (groups A and B).[8] Figure 10.9 shows that subjects agree or even totally agree that the geographic dimension is *easy to learn*, *intuitive*, *easy*, *important*, and *helpful* for users navigating in process model collections. Subjects further agree that

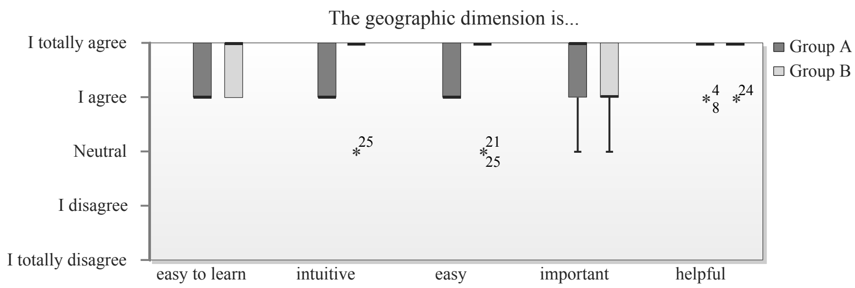


Figure 10.9: The geographic dimension.

the semantic dimension is *intuitive*, *important*, and *helpful* for process participants (cf. Figure 10.10). Only one out of the 18 subjects disagrees that the navigation concept is *easy to learn*. Experiment results related to the view dimension are presented in Figure 10.11. As can be seen, this dimension is considered as being very *important* and *helpful* as well. Subjects further agree that the geographic dimension is *intuitive* and *easy to learn*. The presented results confirm our assumption that each navigation dimension supports users in quickly accomplishing a specific navigation task. In particular, the combination of the three navigation dimensions allows for very useful navigation options.

[8]Descriptive statistics can be found in Figure A.1.

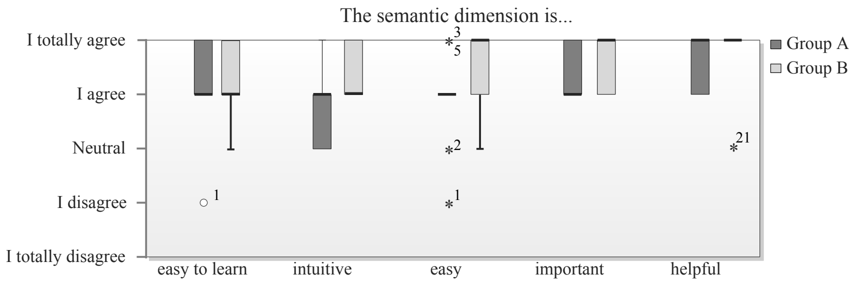


Figure 10.10: The semantic dimension.

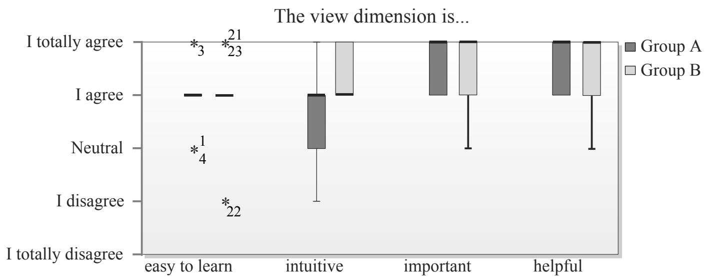


Figure 10.11: The view dimension.

10.7 Threats to Validity

Generally, there are risks when performing experimental research. Hence, factors threatening the *internal* and *external validity* of the experiment need to be considered. Regarding the described experiment, threats of internal validity are as follows:

- **Subjects:** Different experience levels of subjects constitute a crucial factor threatening internal validity. To limit this threat, we exclusively choose subjects from the industrial sector, i.e., process experts working in the area of E/E development processes. This way we want to guarantee same conditions among the subjects. This fact, together with the separated execution of the experiment, also explains the rather small number of subjects. Note that such rare experts are hard to recruit. Finally, we randomly assign subjects to experiment groups in order to achieve a uniform distribution among them.
- **Object:** The investigated objects should not differ in more than one factor in order to make results traceable to this origin. Note that in the context of the experiment, ProNavigator is used for all three groups, i.e., all groups are confronted with similar user interfaces. Only the applied navigation concepts differ (3-dimensional vs. 1-dimensional). Additionally, exactly the same process model collection is used for all groups.
- **Training:** As the complexity of both navigation concepts differs, we distinguish between subjects that receive a standard introduction and subjects that receive a detailed introduction (including the support of the experimenter). Moreover, the training of experiment group B assures that subjects have the same level of knowledge about ProNaVis, as subjects from group C have about the control concept.

Threats of external validity are as follows:

- **Experience:** In order to guarantee a similar level of experience, we select subjects with familiar knowledge. However, this might have a negative impact on the external validity since all subjects are experts in the area of business process management (BPM).
- **Process Models:** Difficult process models, i.e., comprising complex content, could negatively affect subjects. To not falsify results due to comprehensibility issues regarding the used process models, we only consider process models that are semantically easy to understand. They are about planing, executing and post-processing a holiday trip.

10.8 Discussion

The navigation concepts used by participants during the experiment strongly differ in respect to complexity (3-dimensional vs. 1-dimensional). Therefore, it is difficult to instruct subjects in a way that they exhibit equal knowledge about the navigation concepts they have to use. In case of group A, the same amount of time has been invested for introducing the ProNaVis navigation concept as in the case of group C to whom we introduced the 1-dimensional control concept. Results show that, due to the different complexity of the concepts, both groups show different levels of knowledge in respect to the according concepts. To avoid this bias, we introduce group B, whose members received a much more detailed introduction, including auditive support from the experimenter during system introduction. We assume that group B has the same amount of knowledge about the ProNaVis concept as group C has about the control concept after the introductions.

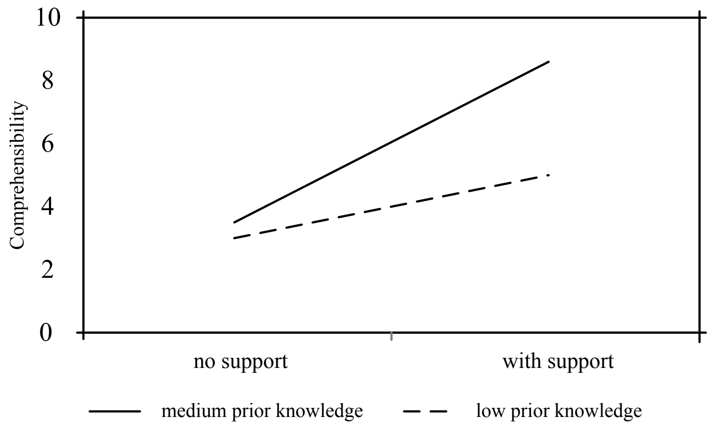


Figure 10.12: Learning effect affecting comprehensibility [Seu03a].

Results are significantly better for group B compared to group A for all three hypotheses. Probably, this effect is caused by the detailed instructions provided by the experimenter prior to the experiment. Seufert et al. [Seu03a] have shown this effect in the area of multimedia learning as well. A learner understands concepts much quicker, when he gets support during introduction. Especially, this applies when he has only low or medium prior knowledge about the topic (cf. Figure 10.12). Note that this applies in our experiment since subjects have process modeling knowledge, but have never applied the ProNaVis navigation concepts before.

Furthermore, group B shows significant results compared to the control concept in each presented scale. This result indicates that the increased functional complexity of ProNaVis does not negatively affect the understandability and usability of the system. In turn, the increased navigation possibilities allow participants to faster navigate to the information needed.

The main lessons learned from the experiment as well as the feedback directly obtained from the subjects are as follows:

- The provision of a separated geographic dimension allows for a better overview of the process model collection.
- The possibility to either decrease or increase the number of displayed information objects along the semantic dimension facilitates tool usage.
- Navigating across process models allows for a better understanding of the relations that exist between single process models.
- The provision of different visualizations allow supporting specific demands of users having different roles (e.g., engineers and managers).

10.9 Summary

The experiment results confirm that the 3-dimensional ProNaVis navigation concept is better suitable for navigating in process model collections than a 1-dimensional navigation concept, if the 3-dimensional navigation concept is introduced in a detailed manner. Though the experiment did not always reveal significant differences regarding single response variables, it shows significant differences regarding the calculated scales for each hypothesis in favor of group B, i.e., the subjects using ProNavigator with the 3-dimensional ProNaVis navigation concept in conjunction with initial support during system introduction. Despite its higher complexity, the developed navigation concept does not negatively bias the subjects' performance. By contrast, subjects perform tasks significantly faster.

0-hypothesis	rejected
H1: There is no significant difference in the understandability of a 3-dimensional navigation concept (with and without initial support) and a 1-dimensional navigation concept.	✓
H2: There is no significant difference regarding the usability of a 3-dimensional navigation concept (with and without initial support) and a 1-dimensional navigation concept.	✓
H3: There is no significant difference in how fast a task can be performed with a 3-dimensional navigation concept (with and without initial support) compared to a 1-dimensional navigation concept.	✓

Table 10.5: Overview on the investigated hypotheses.

As summarized in Table 10.5 all three presented 0-hypotheses can be rejected. Hence, the research question can be answered with reasonable certainty; 3-dimensional process navigation is more suitable for navigating in process model collections compared to a static, 1-dimensional navigation concept, if users are introduced to system functions in a detailed manner.

11 Experiment 2: Process Visualization

11.1 Motivation

This chapter[1] presents a second experiment, investigating visualization concepts for process models. In particular, the experiment deals with the visualization concepts presented in Section 6.3. Thereby, three aspects adopt a key role: *comprehensibility* [MRC07], *aesthetic appearance* [Bir33], and *clarity* [RM11]. The research question of the experiment is as follows:

> *Regarding comprehensibility, aesthetic appearance and clarity, is there a difference between alternative ways of visualizing the logic of process models and–if 'yes'–how strong is this difference?*

To answer this question, we compare the *BPMN3D* and *Bubble* visualization concepts presented in Section 6.3. These are chosen based on an experiment pretest we performed among all four developed visualization concepts. Moreover, we compare the two chosen visualization concepts with a *control concept*, which corresponds to a BPMN-based visualization concept as used by a large automotive manufacturer. 66 participants are involved in the experiment. Its results contribute to better understand the requirements on the visualization of complex business processes.

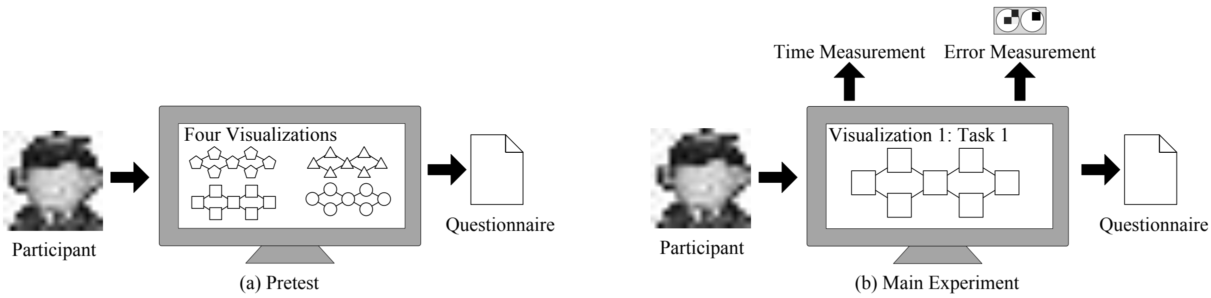


Figure 11.1: Experiment setup.

Figure 11.1 illustrates the setup of the experiment. In the pretest, subjects rate the four visualizations in a *within-subjects* experiment, i.e., subjects rate all four visualization concepts based on their subjective perception. The top two visualizations are then investigated in the main experiment. It is executed as a *between-subjects* experiment. The latter comprises three groups, comparing the two experimental concepts with a control concepts. Subjects have to rate the visualizations based on process-related tasks

[1] The chapter is based on the following referred paper [HSM^{+}14]:
Markus Hipp, Achim Strauss, Bernd Michelberger, Bela Mutschler, and Manfred Reichert. *Enabling a User-Friendly Visualization of Business Process Models.* in: Proc 3rd Int'l Workshop on Theory and Applications of Process Visualization (TaProViz'14), pp. 395-407, 2014

they should perform. Additionally, we measure the execution times and the number of errors made during task execution, e.g., not identifying included syntactical errors in the models (cf. Figure 11.1b).

The remainder of this chapter is organized as follows: Results of the experiment pretest are presented in Section 11.2. Section 11.3 introduces the control concept in detail. Section 11.4 illustrates the experimental design. The hypotheses to be investigated are presented in Section 11.5. Section 11.6 then describes how the experiment was conducted. Section 11.7 presents how experiment data is prepared, whereas Section 11.8 presents experiment results. Section 11.9 discusses threats to validity. Finally, Section 11.10 discusses the results and Section 11.11 concludes the chapter.

11.2 Pretest

In a pretest among the four visualization concepts (cf. Section 6.3), we perform a lightweight controlled experiment involving 22 subjects;[2] 9 of them are students, 5 are academic staff, and 8 stem from industry. The goal is to limit the number of concepts to be tested in the main experiment. The two concepts performing best in this pretest are then further investigated in the main experiment.

A questionnaire is used to collect data about the perception of the four concepts. Part 1 of this questionnaire includes demographic questions, e.g., related to the subjects' modeling experience. In part 2, the subjects must rate each concept in respect to several items using a five step Likert-scale. Possible answers range from *"I totally agree (5)"* to *"I totally disagree(1)"*. The items to be measured are: *Comprehensibility, Comprehensibility of the sequence flow, Clarity, Interest, Stimulation, Simplicity, Appeal, Structure.*

Finally, in part 3 the subjects must evaluate each concept with an overall rating between 0 and 10.

11.2.1 Results

In this section we present results of the pretest. Note that we are only interested in identifying the two best performing concepts. Therefore, results are only presented on a descriptive level, i.e, we do not report or discuss significance of the results.

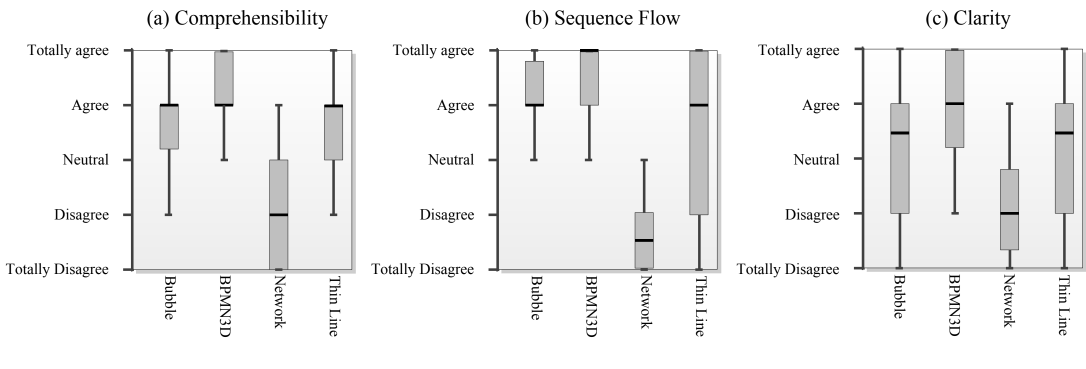


Figure 11.2: Pretest results on comprehensibility, sequence flow and clarity.

In the pretest, BPMN3D is perceived as the most *understandable* concept (*mean* $M = 4.14$; *std dev* $SD = 0.83$) (cf. Figure 11.2a), followed by the Bubble concept ($M = 3.64, SD = 0.79$) and the Thin

[2]To avoid any bias effects, none of the 22 subjects participating in the pretest were involved in the main experiment.

Line concept ($M = 3.36, SD = 1.18$). With ($M = 2.00, SD = 0.93$), the Network concept performs worst. According to [MRC07], this result is reasonable since BPMN3D is similar to BPMN, whereas the Network concept introduces new ideas of visualizing process models.

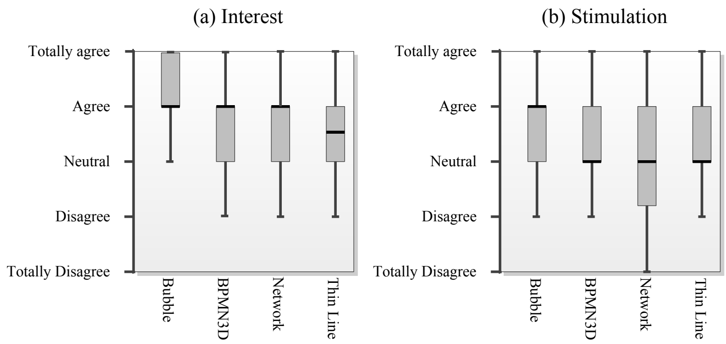


Figure 11.3: Pretest results on interest and stimulation.

Concerning the *sequence flow* (cf. Figure 11.2b), again, BPMN3D is perceived as most comprehensible ($M = 4.59, SD = 0.59$). In turn, Bubble is rated with ($M = 3.91, SD = 1.07$), followed by Thin Line ($M = 3.36, SD = 1.50$) and Network ($M = 1.68, SD = 0.95$).

The *clarity* of the visualization concepts shows similar results (cf. Figure 11.2c). Again, BPMN3D obtains best ratings ($M = 4.05, SD = 0.09$) compared to Bubble ($M = 3.18, SD = 1.05$) and Thin Line ($M = 3.18, SD = 1.18$). Again, Network ($M = 2.09, SD = 1.92$) performs worst.

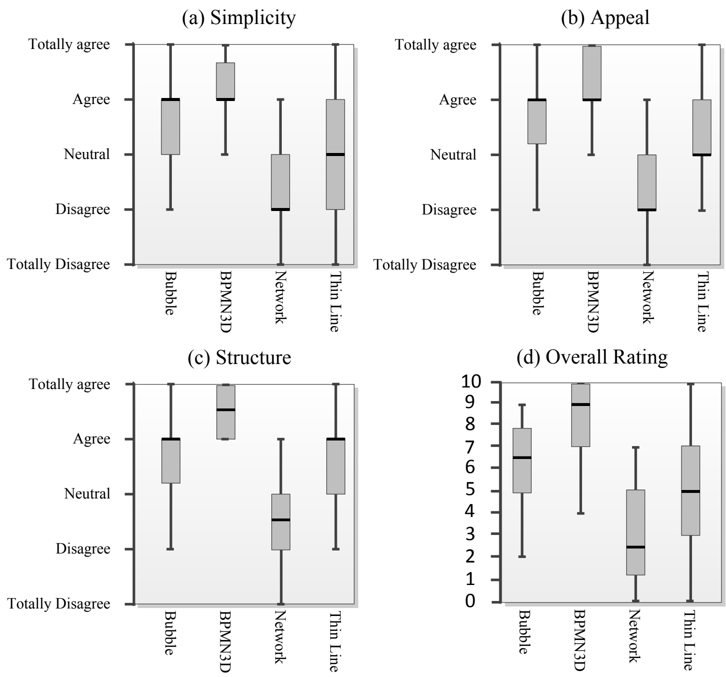


Figure 11.4: Experiment results on simplicity, appeal, structure and overall impression.

Bubble is perceived as the most *interesting* concept ($M = 4.14, SD = 0.83$) (cf. Figure 11.3a). BPMN3D receives the second highest rating ($M = 3.73, SD = 0.83$). Whether a visualization concept *stimulates* the subjects has been answered with similar ratings (cf. Figure 11.3b).

As can be seen in Figure 11.4a, BPMN3D is perceived as the most *simple* concept ($M = 4.00, SD = 0.87$), followed by Bubble ($M = 3.55, SD = 0.86$). Concerning *appeal*, we receive similar results (cf. Figure 11.4b). Again, BPMN3D is perceived as the most appealing concept ($M = 4.18, SD = 0.50$), followed by Bubble ($M = 3.86, SD = 0.77$). BPMN3D is further perceived as the best *structured* concept ($M = 4.41, SD = 0.73$) (cf. Figure 11.4c), while Bubble ($M = 3.59, SD = 0.91$) and Thin Line ($SD = 3.55, SD = 1.06$) are rated second and third best in this category.

11.2.2 Overall Rating

Besides the presented items, subjects are asked to rate their overall impression on each concept (cf. Figure 11.3d). BPMN3D is rated best with $M = 9.18$ out of 10 points ($SD = 1.87$), followed by Bubble, ThinLine and Network.

To identify the two best concepts, 4 points are assigned to each concept for being rated best regarding one item, 3 points for the second highest rating and so on. Finally, the overall rating is considered with doubled points (e.g., 8,6,4, and 2). Table 11.1 summarizes the results of the pretest.

Variable	Bubble	BPMN3D	Network	ThinLine
Comprehensibility	3	4	1	2
Sequence Flow	3	4	1	2
Clarity*	3	4	1	3
Interest	4	3	2	1
Stimulation	4	3	1	2
Simplicity	3	4	1	2
Appeal	3	4	1	2
Structure	3	4	1	2
Overall Rating	6	8	2	4
Total	**32**	**38**	**11**	**20**

* Bubble and ThinLine showed exactly equal ratings.

Table 11.1: Results.

11.2.3 Discussion

Subjects have all been very familiar with BPMN ($M = 4.41, SD = 1.05$). Therefore, the described results of the pretest allow concluding that the subjects' expertise might influence their opinion, as BPMN3D is inspired by BPMN. For example, [MRC07] confirms that the amount of theoretical modeling knowledge influences the comprehensibility of process models. As Bubble also uses BPMN-like structures, its second highest overall rating fosters this assumption. Thus, visualization concepts for process models should combine well-known elements and structures from process model notations with rather few new ideas.

The pretest further confirms that distinguishing process tasks from data objects results in an increased process model comprehensibility. BPMN3D uses a third dimension to visualize data objects. In turn, ThinLine displays process tasks and data objects in different areas. Finally, Bubble uses different visualizations for process tasks and data objects. All three concepts are being considered as comprehensible. Altogether, we selected *BPMN3D* and *Bubble* for our main experiment based on the pretest results.

To improve the *internal validity* of the pretest, all concepts are applied to the same process model, i.e., the resulting visualizations represent the same amount of information. Thus, the visualization itself is the only varying factor. Finally, the visualization concepts are introduced in the same way to all subjects.

Regarding *external validity*, the chosen process models might be considered as being too small. Furthermore, the fact that all subjects had experiences with BPMN might have influenced results as well.

11.3 Control Concept

The *control* concept used in the main experiment consists of a BPMN visualization concept (cf. Figure 11.5) practically used at an industrial partner. More ot less it corresponds to the BPMN standard, i.e., tasks are represented by rectangles, gateways by diamonds, and sequence flows by arrows.

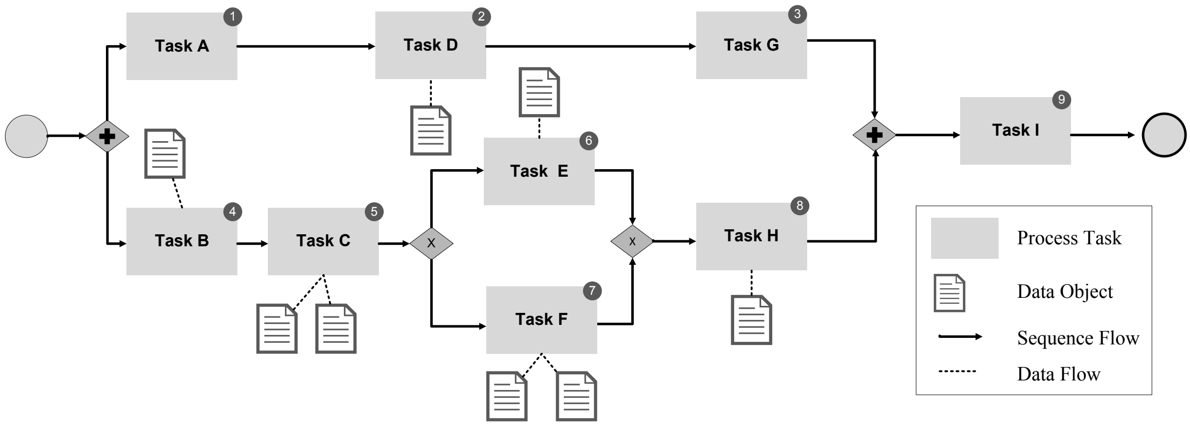


Figure 11.5: The control concept.

11.4 Experimental Design

The main experiment compares Bubble and BPMN3D with the control concept. It corresponds to a *controlled single factor experiment* (cf. Figure 11.6), since it investigates the effects of solely one factor, i.e., visualization. The experiment was conducted as part of a Master Thesis project [Rot13].

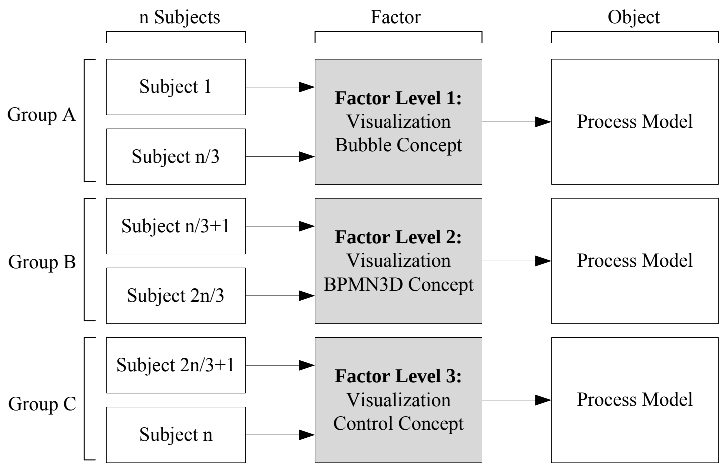


Figure 11.6: Design of our single factor experiment.

Its subjects, object, and selected variables are described in the following:

Subjects: Subjects are 66 students from Ulm University and the University of Applied Sciences Ravensburg-Weingarten. Subjects are divided into 3 groups consisting of 22 members each. Subjects were randomly assigned to the groups prior to the start of each experiment.

Objects: The objects to be rated by each experiment subject are process models from different areas. The investigated visualization concepts are applied to these process models as different factors.

Factor and Factor Levels: The factor is *process visualization* with the factor levels *Bubble*, *BPMN3D*, and *control concept*.

Dependent Variables: The following dependent variables are considered [MRC07, Men08, Nor88]: *syntactic comprehensibility*, *semantic comprehensibility*, *quick comprehensibility*, *simplicity*, *clearness*, *stimulation*, *interest*, *pleasantness*, *clarity*, and *quick overview*. All response variables are assigned to one of three topics: *understandability*, *aesthetic appearance*, and *clarity* (of process models). Additionally, execution times are logged during the experiment in order to investigate how fast tasks are accomplished [HFL12]. Also, we consider the number of errors made during the experiment as measurement for participants' performance [ZPR$^+$12].

Instrumentation: Data was collected with an online tool[3] providing a stop watch as well as error detection functions for experiment tasks. The tool allowed for the collection of qualitative feedback on the usability, aesthetic appearance, and clarity of process models based on a structured questionnaire.

Data Analysis Procedure: For data analysis, well-established statistical methods and standard metrics are applied, including *t-test*, *Kolmogorov-Smirnov test*, and *Shapiro-Wilk test* (cf. Section 11.6).

11.5 Hypothesis Formulation

Based on the defined research question, three hypotheses are derived. These are related to the *comprehensibility* (H1), the *aesthetic appearance* (H2), and *clarity* of process models (H3).

As already described in the context of the first experiment (cf. Chapter 10), we consider the *understandability* of process models [HFL12]. *Aesthetic appearance*, in turn, plays a role in the areas of user interface design and information visualization [Bir33]. As the case studies [HMR11b] revealed the need for providing a better overview, finally, we consider the *clarity* of process models as well.

H1: Understandability of Process Models We investigate whether the different visualization concepts influence the understandability of process models:

- **0-Hypothesis** $H_{0,1}$**:** There is no significant difference regarding the understandability of process models visualized by Bubble, BPMN3D and the control concept.

[3] http://onlineumfrage.com

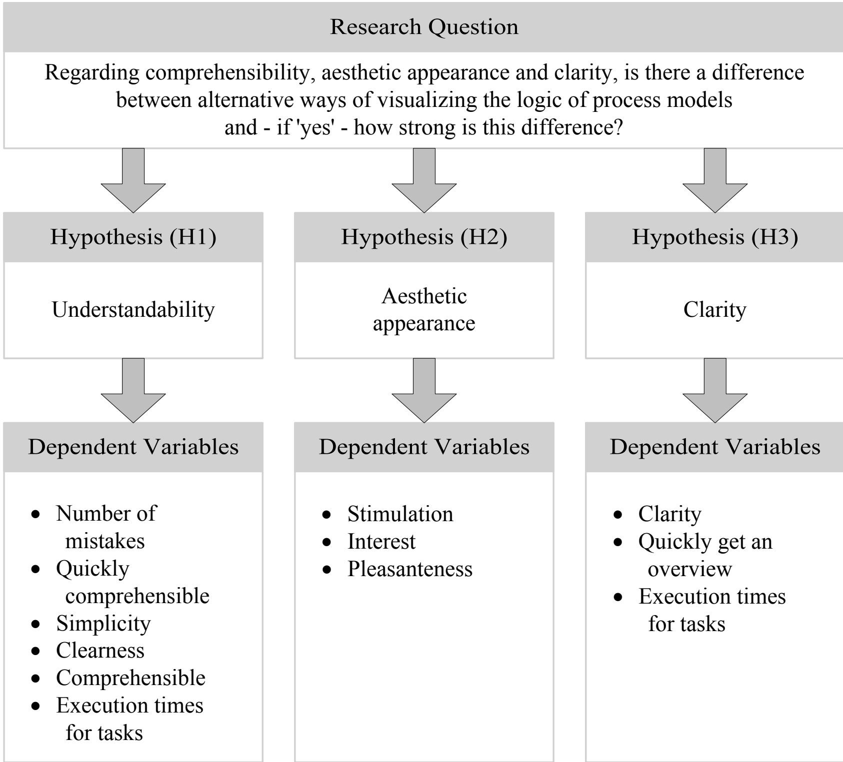


Figure 11.7: Deriving the Response Variables.

- **Alternative Hypothesis $H_{1,1,1}$:** Process models visualized by BPMN3D are significantly better understandable compared to process models visualized by Bubble.
- **Alternative Hypothesis $H_{1,1,2}$:** Process models visualized by BPMN3D are significantly better understandable compared to process models visualized by the control concept.
- **Alternative Hypothesis $H_{1,1,3}$:** Process models visualized by Bubble are significantly better understandable compared to process models visualized by the control concept.

H2: Aesthetic Appearance of Process Models We investigate the effects of the visualization concepts regarding the aesthetic appearance of process models:

- **0-Hypothesis $H_{0,2}$:** There is no significant difference regarding the aesthetic appearance of process models visualized by Bubble/BPMN3D compared to process models visualized by the control concept.
- **Alternative Hypothesis $H_{1,2,1}$:** Process models are perceived as more aesthetic when visualized by BPMN3D compared to process models visualized by Bubble.
- **Alternative Hypothesis $H_{1,2,2}$:** Process models are perceived as more aesthetic when visualized by BPMN3D compared to process models visualized by the control concept.
- **Alternative Hypothesis $H_{1,2,3}$:** Process models are perceived as more aesthetic when visualized by Bubble compared to process models visualized by the control concept.

H3: Clarity of Process Models We further investigate whether there are differences regarding the perceived clarity of process models when applying the different visualization concepts:

- **0-Hypothesis $H_{0,3}$:** There is no significant difference in respect to the clarity of process models visualized by Bubble, BPMN3D, and the control concept.
- **Alternative Hypothesis $H_{1,3,1}$:** Process models visualized by BPMN3D provide a significantly better clarity compared to process models visualized by Bubble.
- **Alternative Hypothesis $H_{1,3,2}$:** Process models visualized by BPMN3D provide a significantly better clarity compared to process models visualized by the control concept.
- **Alternative Hypothesis $H_{1,3,3}$:** Process models visualized by Bubble provide a significantly better clarity compared to process models visualized by the control concept.

11.6 Experiment Execution

The experiment was performed in three sessions. The first session took place in May 2013 at Ulm University. It involved 9 subjects. The second and third sessions took place at the University of Applied Sciences Ravensburg-Weingarten (RW) in June 2013 with 40 subjects in total. The remaining 17 subjects performed the experiment on their own, i.e., using an online tool and getting introduced to the concepts via Skype (cf. Tab. 11.2).

	Ulm	RW (1)	RW (2)	Online
Bubble	3	8	6	5
BPMN3D	3	8	5	6
Control Concept	3	7	6	6
Total	9	23	17	17

Table 11.2: Experiment subjects.

The execution of the experiment is illustrated in Figure 11.8. Approximately, it took 40 minutes per subject. Prior to the start of the experiment, an introductory lecture is given, motivating the topic and the importance of process visualization. Furthermore, subjects are informed about the goals and rules of the experiment (part 1). However, specific visualization concepts are not presented yet. In an individual training based on a PowerPoint presentation, each subject is made familiar with the specific visualization concept. Basic elements are introduced and functionality of the concepts is described. Group A investigates Bubble, group B BPMN3D, and group C investigates the control concept.

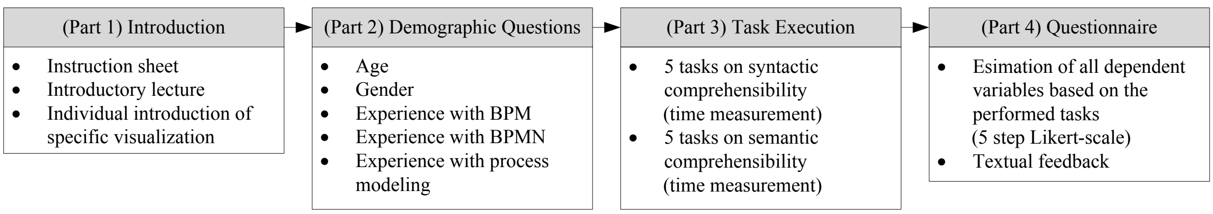


Figure 11.8: Execution of the single factor experiment.

After collecting some demographic data (part 2), subjects are asked to perform various experiment tasks. Thereby, the *syntactic* and *semantic* comprehensibility of the visualization concepts is measured (part 3). For this purpose, subjects must answer questions such as *"May task B be executed before task C?"* to investigate syntactical comprehensibility. Additionally, subjects must compare process models with a textual process documentation to investigate semantic comprehensibility. When performing these experiment tasks, the process model to be investigated is only visible for 30 seconds and disappears before the subjects can answer the questions. Both, time to answer and number of detected errors are monitored. Finally, a questionnaire that evaluates comprehensibility, aesthetic appearance and clarity of the concepts must be filled out by all subjects[4] in part 4 (cf. Figure A.3.1).

11.7 Data Preparation

Before presenting experiment results, experiment data is analyzed and prepared in several steps.

11.7.1 Data Validation and Analysis

A single online tool is used to automatically collect experiment data. Collected data include the numbers of errors made by the subjects when executing experiment tasks, the time to perform the tasks, and the subjects' evaluation of the response variables, i.e., questionnaire results.

Plausibility of data is analyzed using *box-wisker-plot diagrams* (cf. Figure 10.5 in Section 10.5). Such diagrams visualize the distribution of a sample and show outliers. A low number of outliers indicate plausible data [Coo09]. The experiment data is plausible since only very few (negligible) outliers can be observed.

11.7.2 Developing Scales

In this section, a scale is developed for each hypothesis. Thereby, a scale combines a group of response variables (items) into a single, more aggregated variable [Mic90]. To do so, a prerequisite is that all items show high reliability [Kli99], i.e., all items measure the same general topic. Therefore, *Cronbach's* α[5] is calculated. Table 11.3 shows the scales used in the experiment.

Hypothesis	**Scale**	**Items**	**Cronbach's α**
H1	*Understandability*	Quickly comprehensible Simplicity Clearness Comprehensible	.88
H2	*Aesthetoc Appearance*	Stimulation Interest Pleasanteness	.74
H3	*Speed of navigation*	Clarity Quickly get an overview	.73

Table 11.3: Scales used in experiment 2.

Besides these scales we use measured execution times and errors made during experiment task execution as variables to investigate the hypotheses, as well.

[4]Subjective estimations of variables have been evaluated the same Likert-scale as applied in the pretest.
[5]According to [Kli99], α>0.6 indicates acceptable and 0.7< α<0.9 indicates good reliability.

11.7.3 Control Variables and Correlations

First of all, we investigate, whether the control variables reveal significant differences between the three groups. The applied *t-tests* between all combinations do only reveal one significant difference (cf. Table 11.4). The independent variable #3 significantly differs between experiment groups A and C and is therefore considered as covariant in the according significance tests.

#	Control Variable	Group	N	M	SD	t-test
1	How old are you?	A	22	23.95	1.68	A/B: $p_2 = .23$
		B	22	24.91	3.27	B/C: $p_2 = .45$
		C	22	24.23	2.56	A/C: $p_2 = .68$
2	Are you experienced in the area of BPMN? (1=yes, 2=no)	A	22	1.23	0.43	A/B: $p_2 = .74$
		B	22	1.27	0.46	B/C: $p_2 = .48$
		C	22	1.18	0.40	A/C: $p_2 = .72$
3	I feel competent in the area of BPM? (5=highly competent, 1=no competence)	A	6	3.71	0.47	A/B: $p_2 = .12$
		B	7	4.06	0.77	B/C: $p_2 = .49$
		C	8	4.22	0.55	A/C: $p_2 = .01^*$
4	Are you experienced with process modeling? (1=yes, 2=no)	A	4	1.05	0.21	A/B: $p_2 = .31$
		B	7	1.14	0.35	B/C: $p_2 = .67$
		C	8	1.18	0.40	A/C: $p_2 = .16$

Table 11.4: Differences between groups.

Second, we investigate, whether the dependent variables correlate with the control variables. In this case, the according dependent variables have to be considered as *covariant* in the significance tests. As can be seen in Table 11.5, non of the dependent variables shows a significant correlation to one of the control variables. Therefore, the significant tests can be performed without considering a covariant.

#	Control Variable		Scale H1	Scale H2	Scale H3
1	How old are you?	Sig. (2-tailed)	A/B: $p_2 = .99$	A/B: $p_2 = .88$	A/B: $p_2 = .88$
			B/C: $p_2 = .57$	B/C: $p_2 = .74$	B/C: $p_2 = .62$
			A/C: $p_2 = .69$	A/C: $p_2 = .37$	A/C: $p_2 = .96$
2	Are you experienced in the area of BPMN?	Sig. (2-tailed)	A/B: $p_2 = .50$	A/B: $p_2 = .32$	A/B: $p_2 = .97$
			B/C: $p_2 = .42$	B/C: $p_2 = .10$	B/C: $p_2 = .56$
			A/C: $p_2 = .40$	A/C: $p_2 = .89$	A/C: $p_2 = .72$
3	I feel competent in the area of BPM?	Sig. (2-tailed)	A/B: $p_2 = .67$	A/B: $p_2 = .61$	A/B: $p_2 = .97$
			B/C: $p_2 = .57$	B/C: $p_2 = .84$	B/C: $p_2 = .86$
			A/C: $p_2 = .74$	A/C: $p_2 = .53$	A/C: $p_2 = .64$
4	Are you experienced with process modeling?	Sig. (2-tailed)	A/B: $p_2 = .59$	A/B: $p_2 = .30$	A/B: $p_2 = .26$
			B/C: $p_2 = .89$	B/C: $p_2 = .14$	B/C: $p_2 = .27$
			A/C: $p_2 = .58$	A/C: $p_2 = .16$	A/C: $p_2 = .95$

Table 11.5: Correlations between control variables and dependent variables.

11.7.4 Data Analysis

Main goal of the experiment is to investigate whether or not there is a difference between the experiment results of the three groups. More specifically, we analyze the three presented hypotheses. Initially, the respective 0-hypotheses are considered as correct. By applying significance tests (e.g., *t-test* or an additional sign test if the t-test fails) we are able to assess whether the means of two samples statistically differ from each other [Coo09]. A successful test rejects the 0-hypothesis. Specifically, the tests are executed based on a 5% significance level (α=0.05). All used tests are explained in detail in the following.

Explorative Data Analysis: A *Kolmogorov-Smirnov test* and a *Shapiro-Wilk test* are used to analyse whether a data set is well-modeled by a normal distribution (*test of normality*). Not all data sets in our experiment show normal distribution. The used significance tests are described in the following.

Significance Tests for Data Sets with Normal Distribution: Data samples from normally distributed data are analyzed using a *t-test*. With this test, the statistical difference between different data samples is measured.

Significance Tests for Data Sets not showing Normal Distribution: We used the *Mann-Whitney U test* and the *Kruskal-Wallis test* to analyze significances in non-normally distributed data sets.

Statistical Measures: For all significance tests, we provide descriptive statistics of the samples (number n, the *mean*, the *median*, the biggest (*max*) and smallest (*min*) value, and the *standard deviation* σ). For reporting results from significance tests we provide the p-values[6] and additionally all necessary values according to the APA style [Fie13].[7]

11.8 Results

11.8.1 Understandability

Results of the developed scale (cf. Figure 11.9) show that presented visualization concepts are very *understandable* (*mean* group A: $M = 4.06$, *standard deviation*: $SD = 1.03$, group B: $M = 4.64, SD = 0.49$, and group C: $M = 4.30, SD = 0.84$). However, none of the experimental groups (A or B) shows significantly higher means compared to the control group. Only a slight tendency comparing group B and C is noticeable ($U = 179.00, z = -1.54, p_1 = .06, r = -0.23$)[8]. Comparing the two experimental groups, group B, working with BPMN3D, shows a significantly higher result compared to group A ($U = 155.00, z = -2.12, p_1 = .02^*, r = -0.32$).

Figure 11.10 presents results of other considered dependent variables (i.e., execution times and number of errors) collected during the experiment. Figures 11.10a+b show the number of errors made by subjects when performing tasks that deal with the syntactic and semantic understandability of process models. Both experimental concepts do not reveal significant differences compared to the control concept. However, there is a tendency that subjects from group B make less mistakes during tasks regarding syntactic understandability ($U = 194.00, z = -1.33, p_1 = .09, r = -0.31$) and semantic understandability ($U = 187.00, z = -1.31, p_1 = .09, r = -0.31$) compared to the control group (cf. Figure 11.10a+b).

[6] p_2 represents the p-value for 2-tailed tests, and p_1 for 1-tailed tests.
[7] APA Style: http://www.apastyle.org/
[8] Using directed hypotheses, we can use 1-tailed significance tests. Therefore, p might be halved [Kli99].

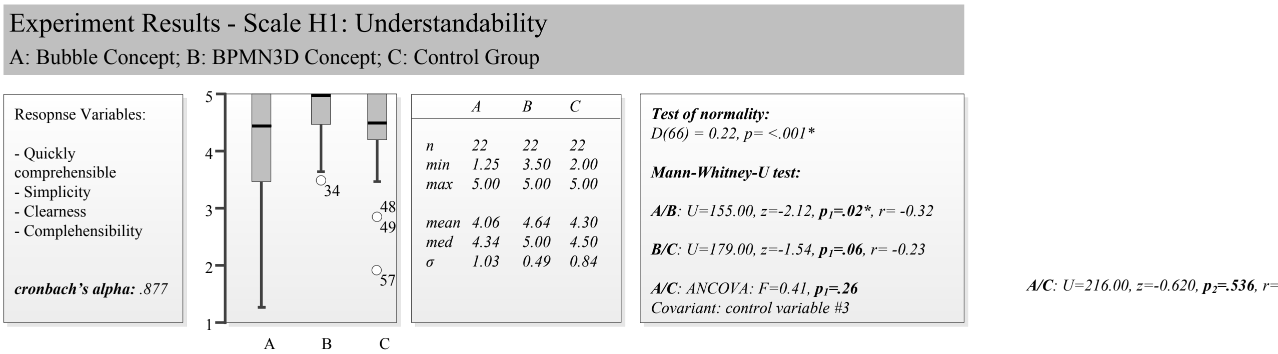


Figure 11.9: Scale for hypothesis H1.

Subjects working with Bubble make significantly more mistakes than subjects working with BPMN3D ($U = 152.00, z = -2.38, p_1 = .01^*, r = -0.56$).

We additionally measure the understandability by measuring the time subjects needed to perform a given task on syntactic (cf. Figure 11.10c) and semantic understandability (cf. Figure 11.10d). Specifically, Figure 11.10c shows that subjects working with one of the experimental concepts completed the respective tasks significantly faster than subjects using the control concept (A/C: $F = 0.01, t(40) = -4.19, p_1 =< .001^*, r = 0.55$ and B/C $F = 0.22, t(40) = -2.19, p_1 = .02^*, r = 0.33$).

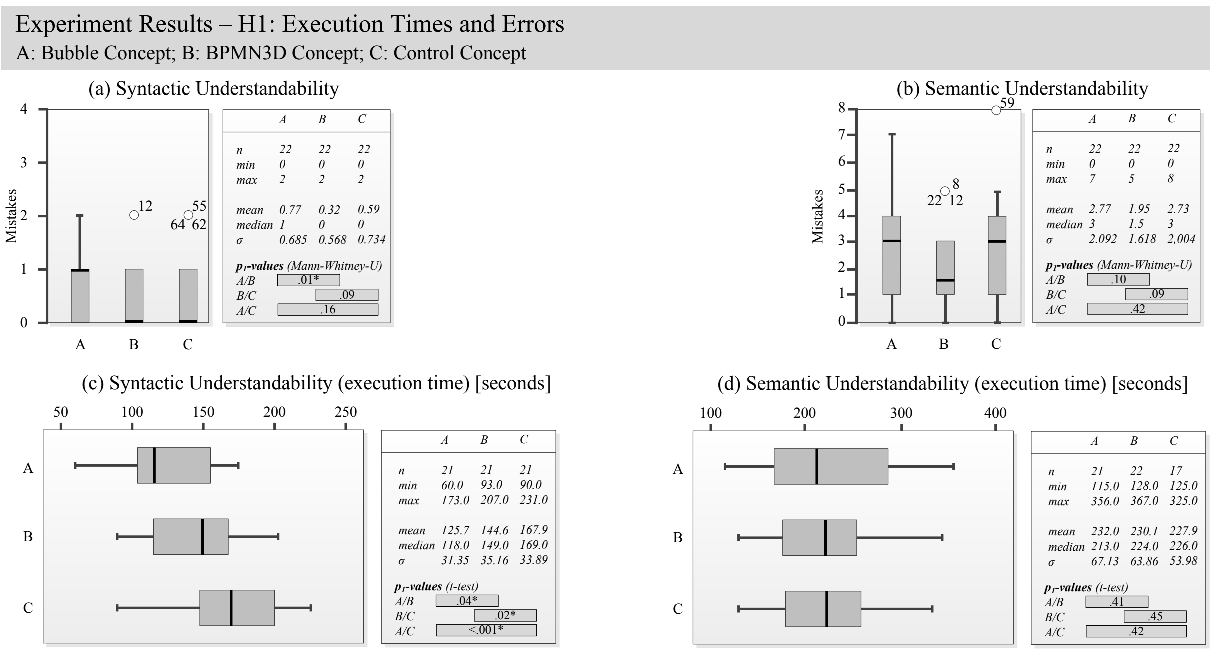


Figure 11.10: Additional results for hypothesis H1.

Based on the results we can reject 0-hypothesis $H_{0,1}$. In turn, we accept alternative hypothesis $H_{1,1,1}$ based on the significant scale result. Despite some clear tendencies, we cannot accept alternative hypothesis $H_{1,1,2}$ with certainty, as only execution times for tasks addressing syntactical understandability show a significant difference.

$H_{1,1,1}$: Process models visualized by BPMN3D are significantly better understandable compared to process models visualized by Bubble.✓
$H_{1,1,2}$: Process models visualized by BPMN3D are significantly better understandable compared to process models visualized by the control concept.✗
$H_{1,1,3}$: Process models visualized by Bubble are significantly better understandable compared to process models visualized by the control concept.✗

11.8.2 Aesthetic Appearance

Results concerning the hypothesis H3 (cf. Figure 11.11) show that the main difference in *aesthetic appearance* of process models can be identified between BPMN3D and the control concept ($U = 156.00, z = -2.03, p_1 = .02^*, r = -0.31$). Between the two experimental concepts we can identify at least a tendency towards BPMN3D compared to Bubble ($U = 175.5, z = -1.58, p_1 = .06, r = -0.31$).

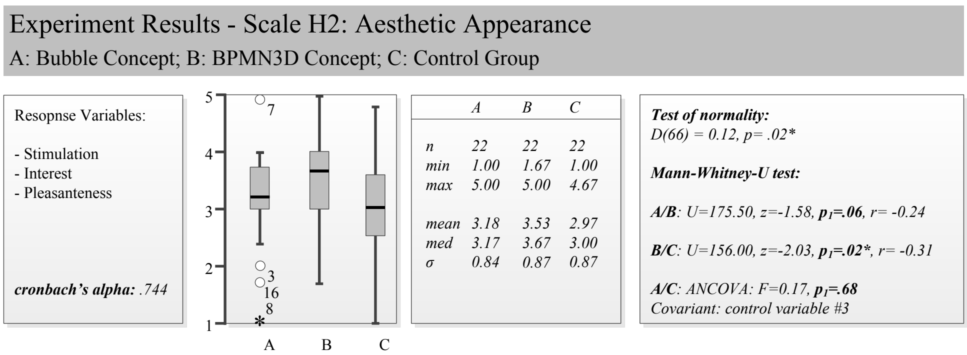


Figure 11.11: Scale for hypothesis H2.

Based on these results, we reject 0-hypothesis $H_{0,2}$. However, only alternative hypothesis $H_{1,2,2}$ can be accepted.

$H_{1,2,1}$: Process models are perceived as more aesthetic when visualized by BPMN3D compared to process models visualized by Bubble.✗
$H_{1,2,2}$: Process models are perceived as more aesthetic when visualized by BPMN3D compared to process models visualized by the control concept.✓
$H_{1,2,3}$: Process models are perceived as more aesthetic when visualized by Bubble compared to process models visualized by the control concept.✗

11.8.3 Clarity

As can be seen in Figure 11.12, significant differences in *clarity* of process models can only be identified between the two experimental groups A and B ($U = 147.00, z = -2.31, p_1 = .01^*, r = -0.35$). The comparison of the experimental systems with the control system, however, does not reveal any significant differences.

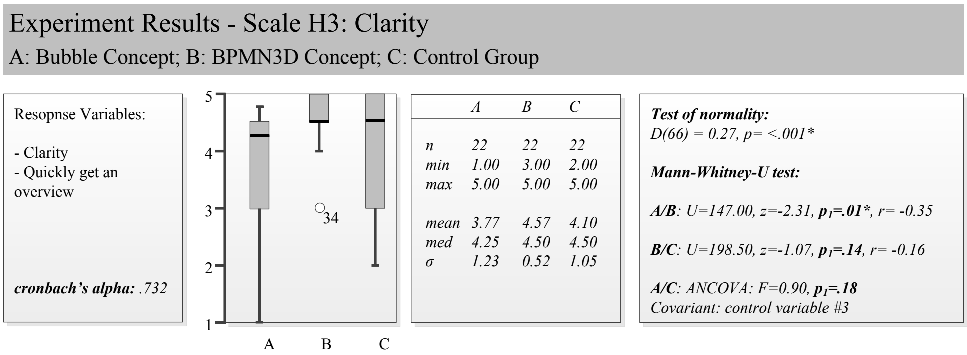


Figure 11.12: Scale for hypothesis H3.

Besides the developed scale, we also take into account execution times for experiment tasks regarding the clarity of process models (cf. Figure 11.13). However, no significant differences can be identified. Surprisingly, the control concept shows the lowest mean. As time measurements for this specific task seem to not correlate with the subjective impressions of subjects regarding the clarity of process models, we do not take them into account for our conclusion.

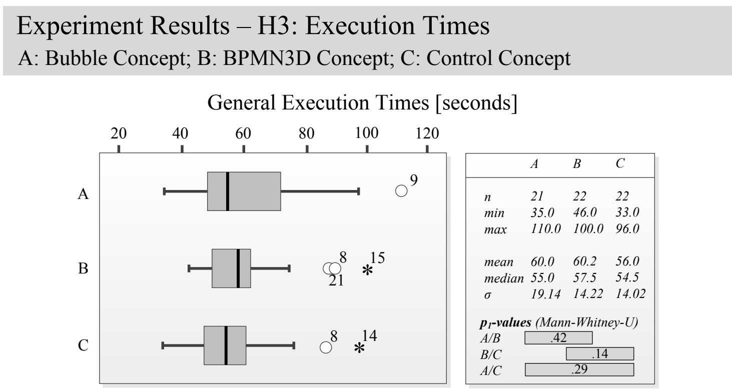


Figure 11.13: Additional results for hypothesis H3.

Based on the results regarding clarity of process models, we can reject 0-hypothesis $H_{0,3}$, as alternative hypotheses $H_{1,3,1}$ can be accepted, due to the significant scale results.

$H_{1,3,1}$: Process models visualized by BPMN3D provide a significantly better clarity compared to process models visualized by Bubble.✓
$H_{1,3,1}$: Process models visualized by BPMN3D provide a significantly better clarity compared to process models visualized by the control concept.✗
$H_{1,3,1}$: Process models visualized by Bubble provide a significantly better clarity compared to process models visualized by the control concept.✗

11.9 Threats to Validity

When performing experimental research, several risks have to be taken into account. In particular, factors threatening the experiment's *internal validity* and *external validity* must be considered.

Threats to internal validity are as follows:

- **Subjects:** The experience of subjects has been identified as a factor threatening the internal validity of controlled experiments. By randomly assigning the 66 subjects to the three experimental groups, we controlled this factor. We could verify a uniform distribution of experience among the three groups (cf. Section 11.7).
- **Training:** Subjects received the same introduction on the visualization concepts to guarantee for similar levels of knowledge.
- **Objects:** The provided process models have been presented to subjects in exactly the same size and structure. Moreover, exactly the same process models have been applied to each visualization concept. We further used same font sizes to avoid an imbalance in readability. A simple context has been chosen for the presented process models, e.g., cooking and shopping. Thus, subjects were able to focus on the visualization concept solely.

Threats to external validity are as follows:

- **Size of Process Models:** To guarantee for the generalization of experimental results, the used process models consisted of 8 to 18 process tasks, which can be considered as an average number of process tasks in practice [WRMR11].
- **Students instead of Professionals:** The experiment has been conducted with 66 participants. Most of them were students. However, it has been shown that results of student experiments are transferable and can provide valuable insights into an analyzed problem domain as well [HRW00].

11.10 Discussion

Based on the experiment results, none of the alternative hypotheses assuming Bubble would perform better compared to the control concept could be accepted. None of the results show better results for Bubble compared to the other two concepts. However, subjects were at least able to perform tasks on the syntactical comprehensibility of process models significantly faster than subjects dealing with the control concept. All other results do not significantly differ from the control concept.

Subjects further evaluated the three concepts along a 10 point rating scale based on their overall impression. This aggregated overall rating is presented in Figure 11.14. Even if BPMN3D is rated significantly better than Bubble ($U = 140.50, z = -2.41, p_1 = .01^*, r = -0.36$), none of the experimental concepts differs significantly from the results of the control concept. Bubble even shows a lower mean compared to the other two concepts.

In turn, two of three 0-hypotheses could be rejected in favor of BPMN3D. The latter obtains higher means compared to the control concept in all nine response variables addressed by the questionnaire (2 significant results). Additionally, subjects that evaluate BPMN3D make less mistakes when performing the experiment tasks related to the semantic and syntactic comprehensibility of process models. Specifically, these subjects performed tasks dealing with syntactical comprehensibility 20 seconds faster per average than subjects evaluating the control concept. Overall, the best overall rating indicates that BPMN3D is the most suitable visualization concept among the presented ones.

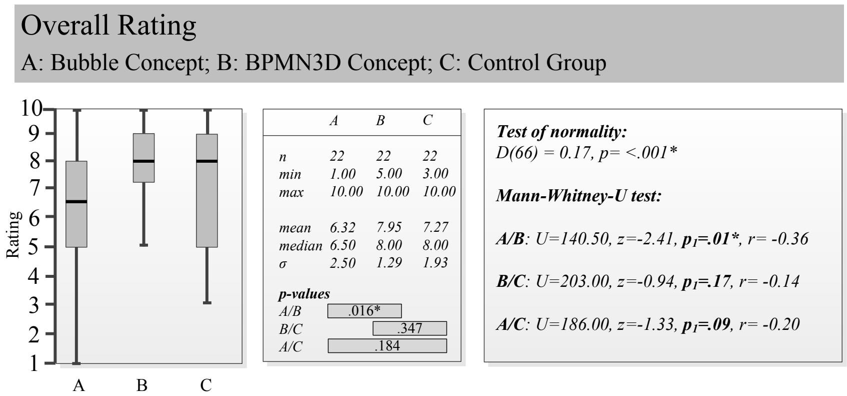


Figure 11.14: Experiment Results - overall rating.

Answering the research question, we can conclude that there is a significant difference regarding understandability, aesthetic appearance and clarity between the BPMN3D and the other two concepts (excluding results regarding *clarity*). However, no other significant difference can be identified.

Since BPMN3D is based on BPMN, it may be assumed that the expertise of participants biases their feedback. Note that Mendling et al. have confirmed that factors such as the amount of theoretical modeling knowledge may play a role when conducting experiments on the comprehensibility of process models [MRC07].

The measurement of execution times turned out to be not very meaningful as gathered data did not correlate with the measured numbers of mistakes. Therefore, we considered execution times, but focus more on the subjective perceptions when estimating the hypotheses. Nevertheless, BPMN3D proves that only small changes in visualization are necessary to improve the understandability, aesthetic appearance and clarity of process models.

11.11 Summary

This section presented results of a user experiment investigating different concepts for the logic-based visualization of process models. We compared two conceptual visualization concepts—Bubble and BPMN3D—and a control concept in a *between-subjects* experiment among 66 participants. In particular, we investigated three basic hypotheses regarding *understandability*, *aesthetic appearance*, and *clarity*. Two of the defined 0-hypotheses could be rejected based on the results of the experiment (cf. Table 11.6).

0-hypothesis	rejected
H1: There is no significant difference regarding the understandability of process models visualized by Bubble, BPMN3D and the control concept.	✓
H2: There is no significant difference regarding the aesthetic appearance of process models visualized by Bubble/BPMN3D compared to process models visualized by the control concept.	✓
H3: There is no significant difference in respect to the clarity of process models visualized by Bubble, BPMN3D, and the control concept.	✗

Table 11.6: Overview on the investigated hypotheses.

For the measurement we used several dependent variables combined to three different scales. Both the number of errors made during task execution and execution times were considered as well. Additionally, subjects gave a subjective estimation on different variables regarding *understandability*, *aesthetic appearance* and *clarity*. These variables were considered to calculate various scales. Further, subjects were asked to provide an overall rating of the presented visualization concepts (cf. Figure 11.14). Again, the result designates BPMN3D as the highest rated concept, with a significant difference compared to Bubble ($U = 140.50, z = -2.412, p_1 = .01^*, r = -0.36$).

According to the presented results, BPMN3D provides better understandability and better aesthetic appearance compared to the control concept. However, in respect to clarity no significant statements can be made, i.e., the presented research question can only be partially answered.

12 Applying ProNaVis to Process-oriented Information Logistics

This chapter[1] illustrates how the ProNaVis framework can be applied to process-oriented information logistics (POIL)–a semantic framework integrating process models and instances with related process information, not explicitly captured in the models [Mic15]. In particular, combining ProNaVis and POIL allows enriching the navigation space with process information. The approach was implemented in *iCare*, a prototype demonstrating how patient treatment processes may be supported.

The remainder of the chapter is organized as follows. Section 12.1 introduces POIL concepts. Section 12.2 then discusses the combination of ProNaVis and POIL. Section 12.3 presents iCare and Section 12.4 concludes the chapter.

12.1 Process-oriented Information Logistics

Providing knowledge workers and decision makers with needed information is often neglected in the field of process-aware information systems (PAIS) [MMR12b]. This is surprising as it is particularly important for complex, knowledge-intensive processes such as product engineering, customer support, or patient treatment. Examples of needed information include emails, office files, forms, checklists, guidelines, best practices, and other kind of information from data sources not explicitly documented in the process model (cf. Figure 12.1).

Particularly challenging in this context is the alignment of external process information with business processes and their tasks. In practice, process information is usually managed separately from process models, i.e., process information is not captured in the process models in terms of data objects. Instead, shared drives, databases, enterprise portals, and enterprise information systems are used to store and manage process information [Sut96, Pet05]. In turn, business process models are managed using process management technology [MRB08].

Michelberger et al. [MMR11b, MMR12a, MMR12b] close this gap by introducing POIL as emerging paradigm for combining process models and process information. In particular, POIL allows for the process-oriented delivery of process information to process participants.

[1] The chapter is based on the following referred papers [HMMR13, MRM$^+$13]:

1. Markus Hipp, Bernd Michelberger, Bela Mutschler, and Manfred Reichert. *A Framework for the Intelligent Delivery and User-adequate Visualization of Process Information.* in: Proc 28th Symposium on Applied Computing (SAC'13), pp. 1383–1390, ACM, 2013
2. Bernd Michelberger, Armin Reisch, Bela Mutschler, Jörg Wurzer, Markus Hipp, and Manfred Reichert. *iCare: Intelligent Medical Information Logistics.* in: Proc 15th Int'l Conf on Information Integration and Web-based Applications & Services (iiWAS'13), pp. 396–399, ACM, 2013
3. B. Michelberger, M. Hipp, and B. Mutschler. *Process-oriented Information Logistics: Requirements, Techniques, Application.* in: Advances in Intelligent Process-Aware Information Systems, 2015. Accepted for Publication

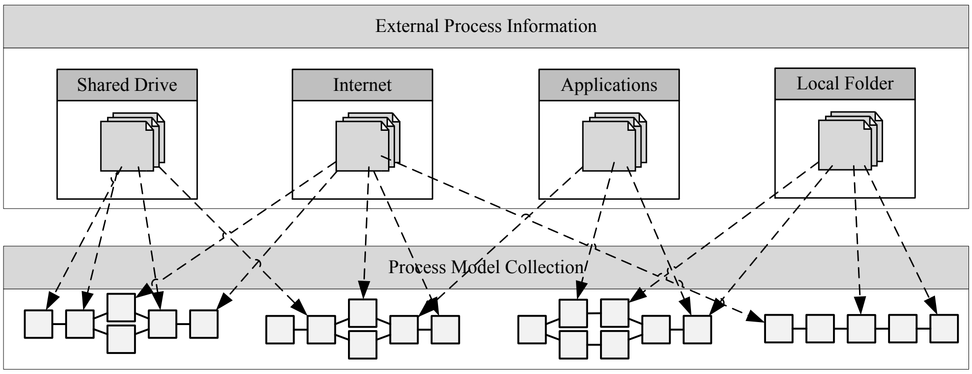


Figure 12.1: Aligning external process information to process models.

12.2 Combining POIL and ProNaVis

Business processes increasingly become knowledge-intense, i.e., consolidated knowledge is required to deal with single process tasks. More specifically, each process task is associated with a multitude of process information, such as engineering documents, development guidelines, contact information, or tool instructions [EM00]. Note that providing this process information is far from being trivial [MPVW04, MMR12b]. Usually, conventional information management concepts and information retrieval approaches are used for this task [Pet05, Sut96]. Office documents, for example, are provided on shared drives. Appointments are managed using personal information management tools and emails are analyzed using full text search engines. Finally, business data is provided by enterprise information systems.

POIL allows integrating and analyzing this information in a semantic structure called *semantic information network (SIN)*. The SIN is the core component of POIL that comprises homogeneous information objects (external process information), process elements (e.g., tasks, events, roles), and relationships between them. In particular, a SIN allows discovering objects linked with each other in different ways, e.g., objects addressing the same topic or object needed when performing a particular process task [MMR12b, MMHR13, MUG+14]. As will be shown in this chapter, to provide navigation and visualization support, ProNaVis can be applied to POIL.

Figure 12.2 indicates how POIL and ProNaVis can be combined. While POIL refers to the *integration* (A) and *analysis* (B) layers, ProNaVis deals with the *navigation* (C) and *visualization* (D) layers (cf. layers A-D on the left of Figure 12.2).

12.2.1 Layer A: Integration

The *integration* layer integrates data from different data sources (cf. Figure 12.2a) realizing a uniform view on the data. We distinguish between data sources comprising *process objects* (i.e., business processes), *information objects* (i.e., external process information), and *context objects* (i.e., context information) (cf. Figs. 12.2b-d).

Process objects correspond to process elements such as tasks, gateways, events, and sequence flows (according to the Business Process Management and Notation (BPMN) terminology). Note that business processes are considered at both the process schema and process instance levels. Thereby, a process schema constitutes a reusable business process template (e.g., describing patient examination processes

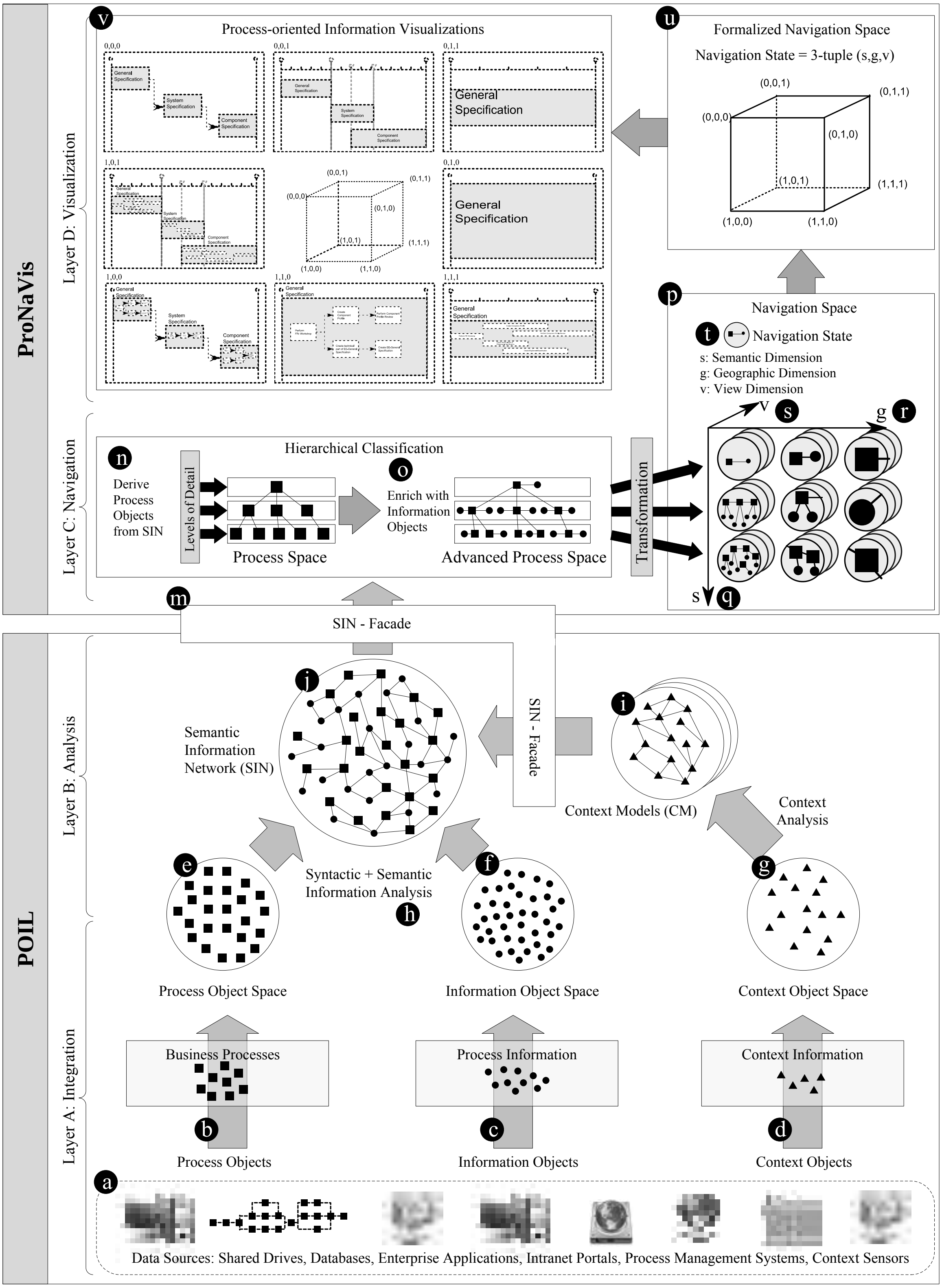


Figure 12.2: The big picture.

in general) that comprises, for example, tasks and sequence flows. In turn, a process instance (e.g., an examination of a certain patient) corresponds to a case concurrently executed with other instances of the same or other process schemas by a process management system [RRD04]. Note that data objects are considered as process objects as well in this context. Process objects are represented in the SIN as squares. In turn, *information objects* refer to external process information needed when working on business processes. In the SIN, they are represented as circles. Examples include emails, office files, informal process descriptions, or best practices. Finally, *context objects* represent information characterizing the work context of a process participant such as user name, roles, experiences, current tasks, used devices, locations, and time [MMR12a] (represented as triangles in the SIN).

For each data source, at least one *interface* must be implemented. Interfaces transform proprietary data objects into generic process, information or context objects. All generic objects follow the same structure and comprise attributes such as id, url, author, file format, or raw content (e.g., the entire text of an email, the coordinates of a user's position). Note that the uniform object structure constitutes a prerequisite to accomplish the syntactical and semantical analyses for identifying the associations between objects. Specific results of the integration are three independent object spaces: the *process object space*, *information object space*, and *context object space* (cf. Figs. 12.2e-g). In turn, an object space (OS) can be defined as a set of generic process, information and context objects (o): $OS = \{o_1, o_2, ..., o_n\}$.

12.2.2 Layer B: Analysis

The mentioned object spaces constitute the foundation of the *analysis* layer. The main purpose of this layer is to create a SIN (cf. Figure 12.2j) based on the available information and process object space. The SIN is constructed and maintained in six consecutive phases (cf. Figure 12.2h): (1) integration of process objects, (2) integration of information objects, (3) identification of process object relationships, (4) identification of information object relationships, (5) identification of cross-object relationships, and (6) maintenance. In [MMR12b], these phases are described in detail.

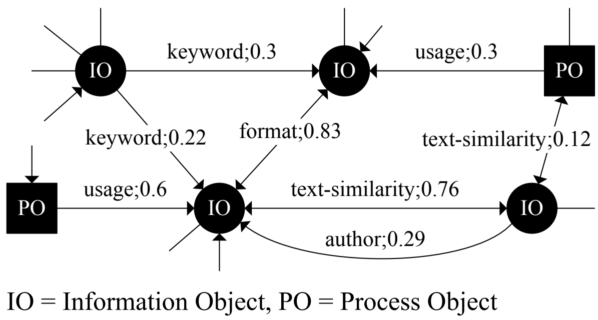


Figure 12.3: Simplified part of a SIN.

Figure 12.3 shows a simplified part of a SIN. As can be seen, the SIN not only comprises information (i.e., circles) and process objects (i.e., squares), but also relations (i.e., black arrows) between these objects. Relations may exist between process objects (e.g., an event triggering a task), between information and process objects (e.g., a file required for the execution of a process step), and between information objects (e.g., a file similar to another one). Relations are labeled with the reason of the relation and are weighted with its relevance (cf. Figure 12.4). A weight is expressed in terms of a number ranging from 0 to 1 (with 1 indicating the strongest possible relationship) [Wur08]. This allows determining why objects are related and how strong their relation is. For identifying the relations between objects, a combination of syntactical and semantical analyzes[2] are applied (cf. Figure 12.2h).

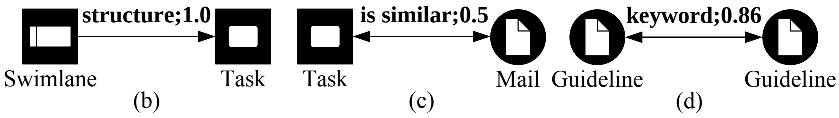


Figure 12.4: SIN relation types.

A SIN can be defined as a labeled and weighted digraph $SIN = (V, E, L, W, f_l, f_w)$, where V is the set of objects (vertices), E the set of relations (edges), L the set of labels, W the set of weights, f_l the labeling function, and f_w the weighting function. Labeling function $f_l : E \to L$ assigns to each relation $e \in E(SIN)$ a label $f_l(e)$. In turn, weighting function $f_w : E \to W$ assigns to each relation $e \in E(SIN)$ a weight $f_w(e) = [0, 1]$.

In addition to the SIN, a context model (CM) (cf. Figure 12.2i) is constructed based on available context objects [MMR12a]. The CM corresponds to an ontology-based model relying on predefined context factors such as user, location or time [MMR12a]. The CM allows characterizing the work context of a process participant, which can then be used to filter the SIN. Based on this, the identification and delivery of currently needed process information becomes more accurate and user-centric (as the delivery of process information can be adapted to the used device or to the experience level of the respective user). The CM is completely independent from the SIN, i.e., context objects are only stored in the CM, but not in the SIN. Hence, there exists one central SIN for all users, but a specific CM for each user. Like the SIN, the CM is a labeled and weighted digraph $CM = (V, E, L, W, f_l, f_w)$, where V is the set of objects (vertices), E the set of relations (edges), L the set of labels, W the set of weights, f_l the labeling function, and f_w the weighting function. Labeling function $f_l : E \to L$ assigns to each relation $e \in E(CM)$ a label $f_l(e)$. In turn, to each relation $e \in E(CM)$ the weighting function $f_w : E \to W$ assigns a weight $f_w(e) = [0, 1]$.

The CM is applied to the SIN by the *SIN facade* (cf. Figure 12.2m). The latter constitutes an interface to retrieve both process information (e.g., office files, working instructions, forms) and process objects (e.g., tasks, gateways) taking the working context of the user into account. Thereby, we distinguish between an *explicit* and an *implicit* information demand. Examples of an explicit information demand include *full-text retrieval* (e.g., delivery of medical reports of a patient using the search query "John Doe report"), *concept-based retrieval* (e.g., delivery of files dealing with a certain concept like the disease "diabetes"), or *graph-based retrieval* (e.g., delivery of related process information to a certain process schema). An example of an implicit information demand is *context-based retrieval*; e.g., a patient record may be delivered taking the doctor's location into account, i.e., the work context of the user is considered to retrieve information and process objects.

12.2.3 Layer C: Navigation

According to the ProNaVis approach (cf. Chapter 4), a navigation space can be created based on a process space, i.e., a hierarchical representation of a process model collection. POIL, however, can

[2] These analyzes are provided by and realized with a semantic middleware [WM09]. More precisely, algorithms from the fields of data mining, text mining (e.g., text preprocessing, linguistic preprocessing, vector space model, clustering, classification, information extraction) [HNP05], pattern-matching, and machine learning (e.g., supervised learning, unsupervised learning, reinforcement learning, transduction) are applied. Specific algorithms are (inverse) term frequency algorithms, link popularity algorithms, and utilization context algorithms [MMHR13, Wur08].

provide this navigation space in terms of the SIN (through the *SIN facade*). Therefore, a specific relation type (*structure relation*; cf. Figure 12.4a) between process objects is established when integrating process models (layer B in Figure 12.2).

Structure relations refer to child relations between two process objects. Based on these relations, the process space, as introduced in Section 4.3, can be derived from the SIN (cf. Figure 12.2n). In order to align process information with process models, the process space may be enriched with SIN information objects (cf. Figure 12.2o) to an *advanced process space*. Therefore, relations established between process and information objects can be used (e.g., *usage, author, is similar*; cf. Figure 12.4b). Identified information objects are assigned to the same level of detail as the process objects they are related to[3]. For example, Figure 12.5 presents a part of an exemplary process space and shows how an advanced process space may look like.

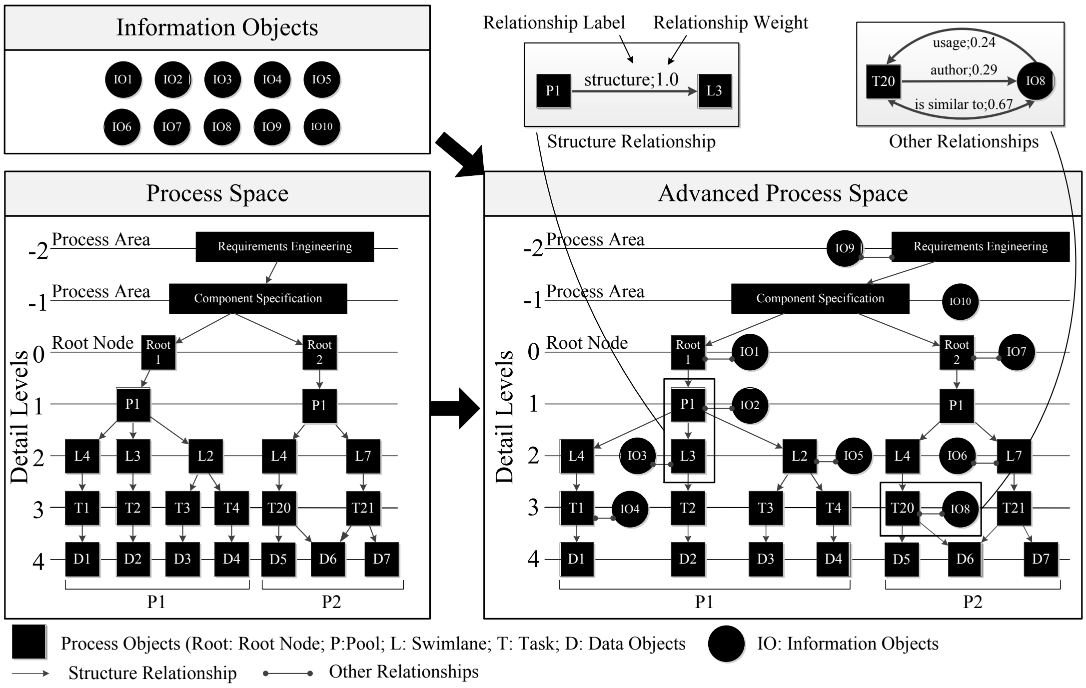


Figure 12.5: Deriving the advanced process space.

The illustrated part of the process space comprises two process object models (POMs) $P1$ and $P2$, i.e., hierarchical representations of two process models. Additionally, 10 information objects ($IO1$-$IO10$) from the SIN are considered when constructing the advanced process space. Based on the identified relations to process objects, information objects are assigned to the same levels of detail.

Using this advanced process space, the navigation space can be derived as described in Section 4.4 (cf. Figure 12.2p). Within the navigation space, a *navigation state* may comprise process as well as information objects. The presented ProNaVis formalizations for navigating in such a space (cf. Chapter 5) can still be applied (cf. Fig 12.2u). Hence, a navigation state corresponds to a point in a Cartesian coordinate system. Unit vectors represent state transitions triggered by user interactions (adjusting levels of the navigation dimensions).

[3]more details regarding the applied algorithms can be found in [MMHR13, MUG+14]

12.2.4 Layer D: Visualization

The visualization dimension, as described in Chapter 6, can then be applied to the advanced process space. Figure 12.2v shows an example of visualizing a simple navigation space with eight navigation states. Two semantic levels, two geographic levels, and two visualization types (logic-based view: 0, time-based view: 1) are considered. Navigation state $(0, 0, 0)$ shows three processes on an abstract geographic and semantic level (with both levels being 0). The view is logic-based, i.e., logic predecessor/successor relations are presented as arrows. Moving to navigation state $(0, 0, 1)$ results in a time-based view, i.e., a timeline is now shown where the length of the process boxes corresponds to process duration. In order to obtain more detailed information, the user may navigate to navigation state $(1, 0, 1)$, providing single process steps. Finally, by adjusting the geographic dimension, the user may zoom into one process to visualize corresponding process steps (this corresponds to navigation state $(1, 1, 1)$). For example, a requirements engineer may benefit from this detailed navigation state, since process information is provided on the level of detail needed. In turn, a manager may be free to navigate to any other (e.g., more abstract) navigation state within the SIN.

12.3 Applying the Approach in the Healthcare Domain

This section presents iCare[4], a web-based semantic Java application based on the semantic middleware iQser GIN platform 1.6 [WM09], the web framework Wicket 1.5.6, the JavaScript library jQuery 1.72, HTML5, and CSS3. It implements the presented four layers (A-D) of the combined POIL and ProNaVis approaches. We introduce a real-world application scenario to illustrate its functionality. This scenario is based on results of a case study we performed at a large German university hospital [HMR11b, MMR11a]. It deals with the treatment of patients in the healthcare domain. The underlying process (cf. Figure 12.6) is knowledge-intensive, i.e., it comprises complex tasks (e.g., examinations and diagnosis), complex data objects (e.g., patient records, laboratory reports, notes) and unstructured process information (e.g., emails, information from the Internet, and personal notes).

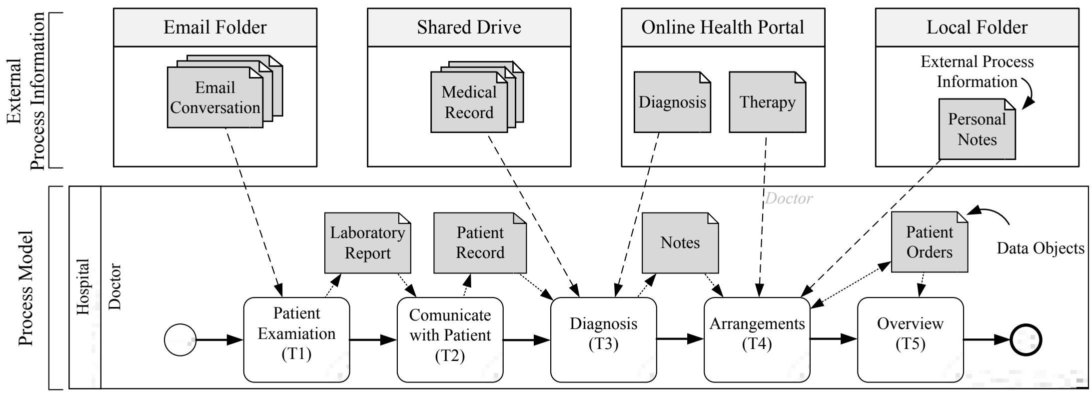


Figure 12.6: A patient treatment process.

In particular, the scenario deals with patient examination requiring that the doctor needs access to patient records, medical notes and laboratory reports. Note that for illustration purposes, a simplified process model for patient examination is used (for a more detailed description of this process, we refer

[4]A screencast presenting the iCare application is available at http://nipro.hs-weingarten.de/screencast.

to[PMLR15]). First of all, the doctor examines the patient in the context of Task $T1$. Prior to the examination, process information, such as emails from the patient or from other doctors, might provide useful insights into the patient's medical history. Then, the doctor communicates with the patient and makes notes during the regular ward round ($T2$). Based on this information, he diagnosis the patient's illness ($T3$). Therefore, medical records from other patients suffering from similar sicknesses might be helpful. Furthermore, the doctor might consider differential diagnoses from an online health portal during decision-making. In Task $T4$, the doctor sets up a medical arrangement. Thereby, he might be supported by an online health portal or by personal notes. Finally, the doctor gets an overview on the patient's state of health (e.g., diagnosis and therapy) in Task $T5$. In current practice, however, process information is usually hard to find and therefore important information might be ignored during treatment. In the following, we describe how this scenario can be supported by iCare.

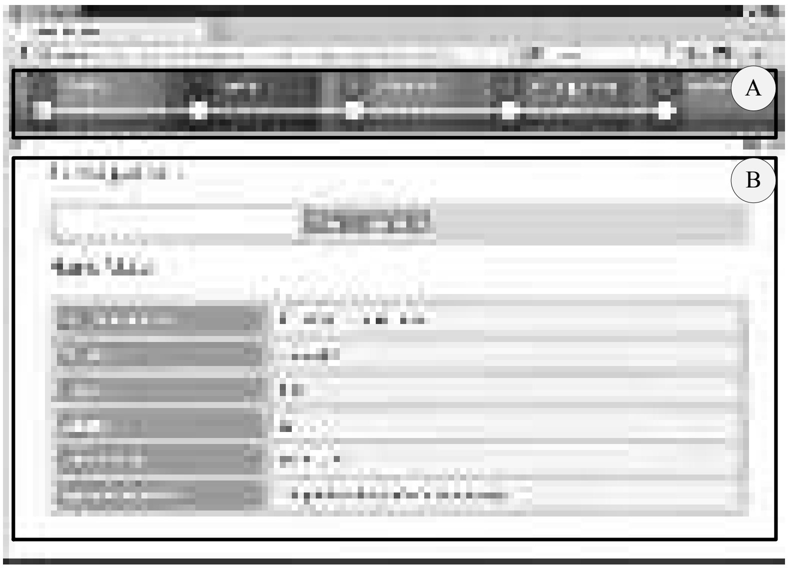


Figure 12.7: Process task 1.

The main features of iCare are:

- iCare enables the integration of structured, semi-structured and unstructured process information from different data sources.
- iCare enables the automatic syntactic and semantic analysis of information to determine relationships from which new knowledge can be derived and generated.
- iCare enables the personalized delivery of needed information to process participants and represents a central access point.
- iCare allows navigating along different navigation dimensions, e.g., different detail levels.
- iCare provides different visualizations on process models and process information.

iCare comprises two basic display areas. The *process overview* (cf. Figure 12.7A) and a detailed *information view* on single tasks (cf. Figure 12.7B). The former illustrates the currently executed process (or task), whereas the latter shows the corresponding process information in different visualizations.

As the application scenario only comprises one single process, the respective navigation space is limited (cf. Figure 12.8). In fact, only two levels of detail along the semantic dimension need to be supported–one for the root nodes (detail level 0) and one for process tasks (detail level 1). Aditionally, two visualizations are provided: a logic- and a list-based visualization.

In iCare, semantic level 0 is used for displaying the process model in the *process overview* area. The entire process model is presented using the logic-based visualization (geographic level 0). Semantic level 1 is applied for displaying detailed information about single process tasks in the *information view*. In this case, a list-based visualization is used to display the information (geographic level 1).

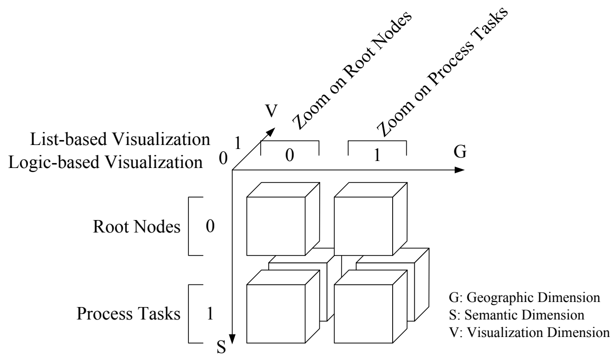


Figure 12.8: The used navigation space.

To support Task $T1$ in the scenario, a search box is offered to select single patients. After having selected a patient, iCare provides available information such as name, pre-existing diseases, allergies, gender, weight, and date of birth from the respective patient record in a list-based visualization (cf. Figure 12.7). When performing Task $T2$, existing medical notes (documented in the patient record) for the selected patient are shown, i.e., information about the patient's health status (cf. Figure 12.9). Upon need, the doctor may add, update or delete medical notes. A simple list-based visualization is used to display different notes in a chronological order.

Figure 12.9: Process task 2.

Based on an analysis of available medical information, considered as process information within the SIN, suggestion for potential diseases and treatment options are then automatically determined when performing task $T3$. For example, the analysis takes the patient record, medical notes, and medical information from Onmeda[5] into account and can automatically conclude that sore throat, croakiness, rheumatic pains and absence of appetite are potentially caused by disease "flu" (cf. Figure 12.10). More specifically, each entry in the presented list represents one SIN information object, automatically related to process task $T3$. The relation is based on relationships between the given notes from the patient record and the disease description from Onmeda (e.g., *Text similarity:0.13*). Note that there exist multiple other relationships. For the sake of simplicity, however, only the most relevant one is shown to the user.

Figure 12.10: Process task 3.

As an additional result of the syntactic and semantic analysis, the doctor is informed about treatment options (cf. Figure 12.11) in terms of process information. If a treatment option is selected, a more detailed treatment description and respective instructions can be displayed. In the context of Task $T4$ the doctor can then add or update medical orders. Finally, the patient record, medical notes, and medical orders are summed up and can be finally updated in task $T5$.

In summary, iCare supports the doctor during patient treatment by reducing the time for searching and handling process information. iCare automatically delivers needed process information dependent on the current work context.

12.4 Summary

The alignment of process information with business processes is a challenging task, especially since the two perspectives are usually addressed separately. While process information is stored and managed using databases, information systems and shared drives, process management technology is used to coordinate business processes. To close this gap, the chapter showed how ProNaVis can be combined with the POIL framework. In particular, semantic technology enables the seamless and automated analysis and alignment of process information with business processes.

[5]Since we have no access to medical libraries we use the health portal Onmeda (http://www.onmeda.de) instead.

Figure 12.11: Process task 4.

To illustrate the benefits of this combined approach, we presented iCare—a semantic prototype enabling the process-oriented integration, analysis, and delivery of process information. Its major goal is to deliver medical information (e.g., patient records) to doctors in an intelligent way during patient treatment. We showed how doctors might be supported with additional process information from external data sources along the patient treatment process. Note that iCare constitutes only one example of a combined POIL and ProNaVis application. Other applications can be easily implemented in other domains as well [MMB+14].

Part IV

Discussion & Summary

13 Discussion

The increasing size and complexity of process model collections (PMC) [WRMR11] forces enterprises to provide more effective support for process owners as well as process participants. For this purpose, process-aware information systems (PAIS) were introduced to create [Hav05], execute [WRWRM09] and monitor [Men08] the models of a PMC. However, supporting end users in navigating within PMC and complex process models has been neglected so far [BRB05, HMR11b]. Tackling this challenge, this thesis introduced ProNaVis, a generic navigation and visualization approach for PMC. In particular, ProNaVis provides a navigation space enabling process participants to navigate along three dimensions, i.e., the *semantic*, the *geographic* and the *visualization* dimension. This chapter discusses contributions and limitations of the presented approach.

ProNaVis was developed in the niPRO project.[1] In particular, niPRO applies semantic technology (e.g., semantic networks, semantic search and semantic analysis) to realize intelligent and user-adequate process information portals. The overall project goal was to support knowledge workers and decision makers with personalized process information depending on their current work context. The niPRO framework itself is based on two main pillars (cf. Fig. 13.1): POIL and ProNaVis.

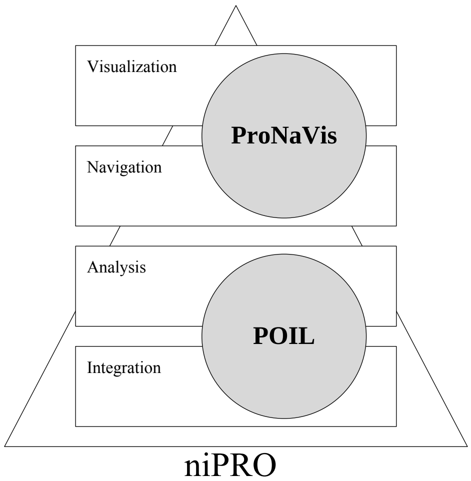


Figure 13.1: The niPRO project.

POIL targets at the provision of the right process information, in the right format and quality, at the right place, at the right point in time, and to the right actors. Actors need not search for required process information anymore, but are automatically linked with relevant process information. The latter is ensured even if the work context of an actor is dynamically changing. The major POIL concept is the semantic network, which comprises both process and information objects as well as the relations

[1] The user-adequate process information portals (niPRO) project was funded by the German Federal Ministry of Education and Research (BMBF) under grant number 17102X10.

between them (cf. Chapter 12). ProNaVis, in turn, aims to support a flexible navigation within and across complex business processes.

In the following, we reflect the contributions provided by the ProNaVis framework along the research questions introduced in Chapter 1. We address each research question and discuss the strengths and weaknesses of the related results.

> *RQ #1: What are existing problems and requirements regarding the navigation within process model collections as well as the visualization of the latter from the perspective of the end user?*

Answering RQ #1 required comprehensive case study research in different domains. In detail, we performed two case studies, one in the automotive domain and another one in the healthcare domain. Additionally, we conducted an online survey with more than 200 participants from various domains in order to confirm case study results. Based on this initial research, we derived 6 fundamental requirements regarding the navigation in PMC (*NavReq*) and 5 requirements regarding PMC visualization (*VisReq*) (cf. Table 13.1).

Req #	**Requirement**
NavReq #1	Depending on a process participant's experience, the level of detail regarding a process task should be adjustable.
NavReq #2	Process participants should be able to adjust the level of detail regarding process model collection in order to obtain a quick overview on a specific task that is currently executed.
NavReq #3	Users should be enabled to access process tasks in other process areas.
NavReq #4	Relevant process information must be accessible at the level of single process models from the process model collection.
NavReq #5	Roles must be globally defined in a detailed manner.
NavReq #6	Process participants must be able to access process models on different levels of detail.
VisReq #1	Task descriptions must be documented in a well understandable manner.
VisReq #2	Temporal and logical dependencies must be considered when visualizing processes.
VisReq #3	Complex process information must be visualized in a comprehensible manner.
VisReq #4	Information about roles must be intuitively identifiable.
VisReq #5	The amount of visualized information should not overload process participants.

Table 13.1: Overview on main requirements.

> *RQ #2: How should a navigation concept for process model collections be approached?*

To answer RQ #2, three major challenges to approach a navigation concept for PMC were identified:

- Navigating on different levels of detail.
- Navigating by zooming.
- Navigating between different visualizations.

Inspired by zoomable user interface (ZUI) concepts (e.g., [RB09, ZJR11, BH94]) and well-known concepts from geographic information systems (GIS), we introduced ProNaVis, a generic process navigation and visualization framework. In particular, ProNaVis has been inspired by navigation concepts known from Google Maps. However, the most significant contribution of ProNaVis is to split the *zoomig dimension* into a *geographic* dimension on one hand and a *semantic* one on the other. This enables us to display detailed information on an abstract zooming level. Note that this has not yet been possible with Google

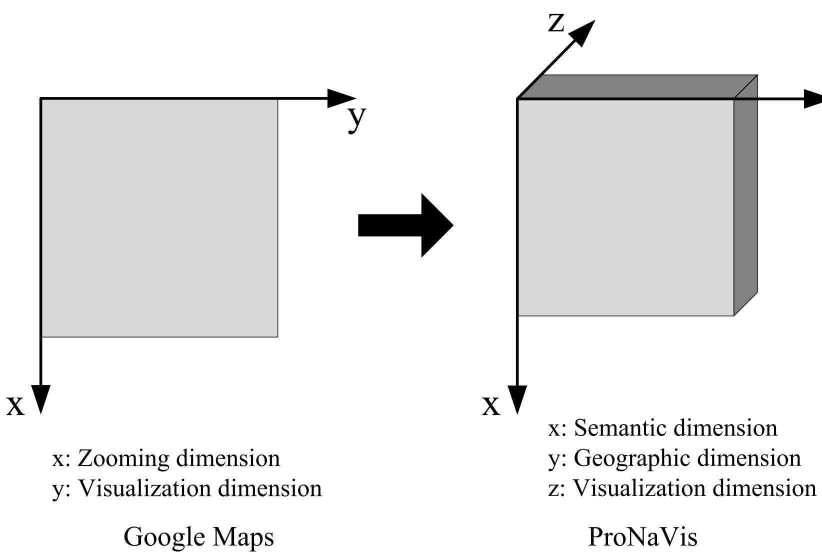


Figure 13.2: The navigation space.

Maps due to “hard-wired” semantic and geographic dimensions (i.e., the zooming dimension). Note that this idea also distinguishes ProNaVis from other navigation concepts.

The *navigation space* constitutes the main component of ProNaVis. It consists of three independent navigation dimensions addressing the aforementioned challenges: the *semantic*, the *geographic*, and the *visualization* dimension. In particular, ProNaVis extends navigation concepts from Google Maps by one additional dimension (cf. Figure 13.2).

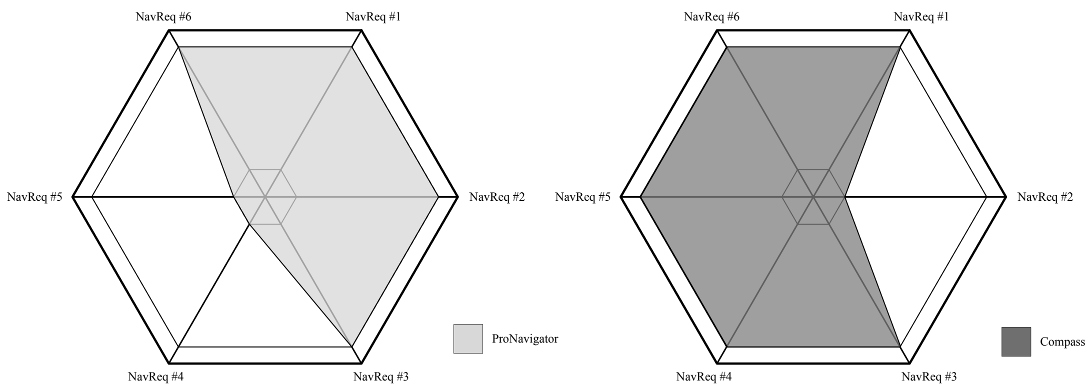


Figure 13.3: Requirements met by the ProNaVis prototypes.

To be able to validate ProNaVis concepts as well as to discuss them with end users, two prototypes were developed. First of all, *ProNavigator* was created to illustrate ProNaVis functions, i.e., the 3-dimensional navigation space. In turn, *Compass* was developed for industrial use. Compass constitutes a process navigation tool supporting process participants during the development of E/E car components. Figure 13.3 illustrates in how far the prototypes meet the navigation requirements.

New challenges emerged when using the prototype, which must be taken into account in future work. First, the refinement of the geographic and the visualization dimensions, together with the integration of existing approaches addressing these dimensions, need to be further investigated. Second, the practical use of ProNaVis should be further explored. As presented in Chapter 4, for example, the navigation space builds upon a collection of BPMN process models. In practice, however, process models are often distributed across heterogeneous data sources. Consequently, the following questions need to be addressed:

- How can process models be extracted from heterogeneous data sources?
- How can process models be transferred into a homogeneous, machine-readable representation?
- How can semantically related process models from different sources be combined?
- What alternative concepts exist to transfer process models into an integrated hierarchical structure?

> *RQ #3: How may process model collections be visualized in a comprehensible manner?*

To tackle RQ #3, we addressed two specific areas of visualization. On one hand, we presented different visualization types, i.e., basic visualizations of which each serves a specific purpose [BBR06]. In this context, we considered the visualization requirements discussed in Chapter 2. In particular, we presented a *time-based*, a *logic-based*, a *text-based*, and a *list-based* visualization type. On the other hand, we addressed the *logic-based* visualization in more detail, as it constitutes the most common notation for process models (e.g., the BPMN standard). In this context, we presented four different logic-based visualization concepts in order to improve BPMN-like visualizations.

Figure 13.4 indicates how the visualization requirements are met by the developed visualization types. As can be seen, none of them meets all five requirements. However, each requirements is satisfied by at least one of the visualization types. Therefore, we may conclude that visualizing process models in a way that fits all user requirements cannot be achieved with a single visualization. Instead, various visualization types should be provided. The four examples can therefore be considered as an initial set of basic visualization types serving the majority of the visualization requirements.

Understandability of process models depends structural aspects [MRC07]. However, the visualization itself constitutes a key factor for understandability as well [BRB05]. In order to improve the understandability of process models from a user's point of view, we developed four conceptual visualization concepts serving as alternatives for the common BPMN models. Initially, we identified a set of requirements specifically investigating the effects of logic-based model visualizations on users. When deriving these requirements, we considered aspects such as *understandability of process models* [MRC07], *aesthetic measures* [Bir33], and *usability engineering* [Wri03]. Figure 13.5 shows how the presented visualization concepts meet these requirements.

Each visualization concept was evaluated in a user experiment (cf. Chapter 11). Results indicate that visualization concepts similar to the ones known from BPMN perform better in respect to the requirements presented. This might be explained with the fact that people tend to favour familiar things [Men08]. Effects of this bias, therefore, need to be taken into account in future empirical research.

> *RQ #4: How can the benefit of a user-driven navigation concept be measured?*

In order to measure the benefit of ProNaVis compared to existing process navigation concepts, we performed a controlled user experiment (cf. Chapter 10). in the context of this experiment, we applied

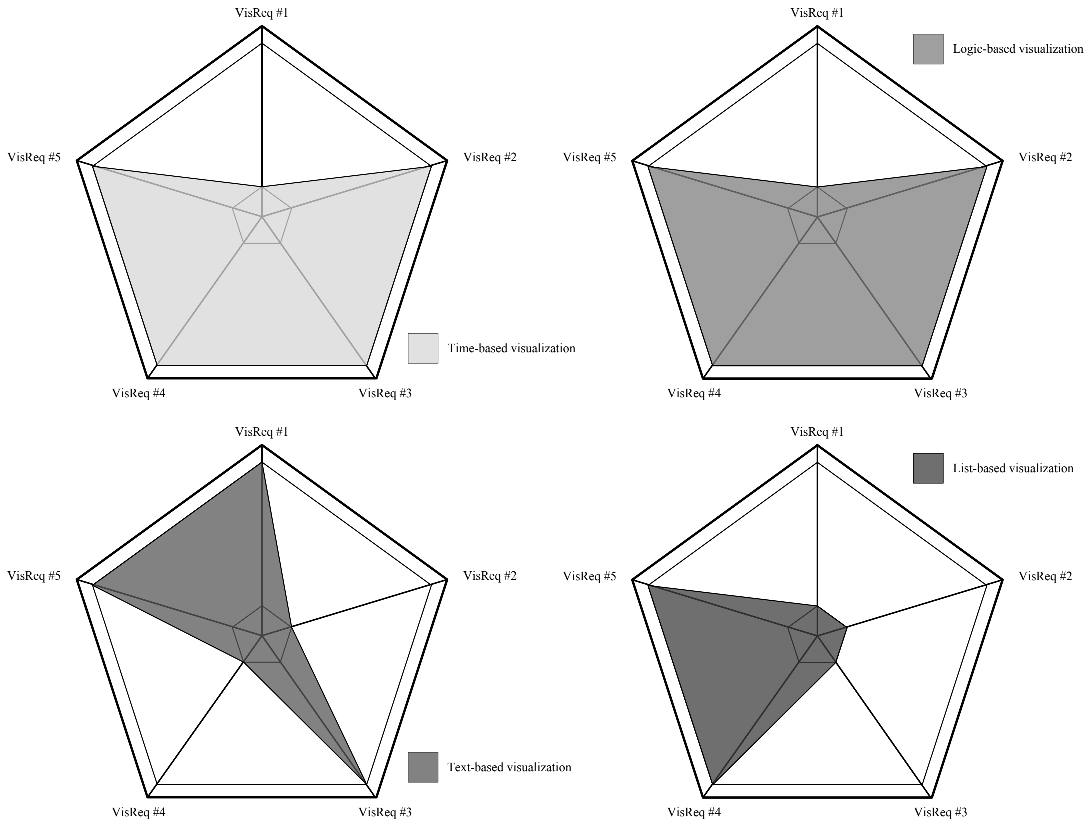


Figure 13.4: Requirements met by the visualization types.

ProNavigator (cf. Chapter 9). To be more precise, we used two different versions of *ProNavigator*. The first one implemented the entire 3-dimensional ProNaVis functions, whereas the second one solely provided a 1-dimensional navigation concept based on existing process portals. The latter has been used as control system by the control group during the experiment. Note that the provision of two systems with identical user interfaces ensured optimal conditions for the experiment, as only the different navigation concepts constitute factor levels [WRH+12].

Experiment results confirm that the 3-dimensional ProNaVis navigation concept is better suitable for navigating in complex process model collections compared to the 1-dimensional one. Though the experiment did not always reveal significant differences, it clearly indicates higher means for almost all response variables in favor of a 3-dimensional navigation. Despite its increased complexity, the navigation concept does not negatively bias user performance. In particular, subjects performed tasks significantly faster.

However, the experiment revealed several limitations that need to be discussed.

First, the experiment showed that 3-dimensional ProNaVis concepts were not as intuitively comprehensible by subjects as the 1-dimensional concepts that was used by the control group. Therefore, we introduced a third experimental group, which received a more intensive introduction into the ProNaVis concepts to ensure that all participants understand the given functions. Results have shown that these participants performed significantly better. In future work, this effect must be investigated in a more detailed manner. Specifically, practical practical applications, the acceptance of ProNaVis will correlate

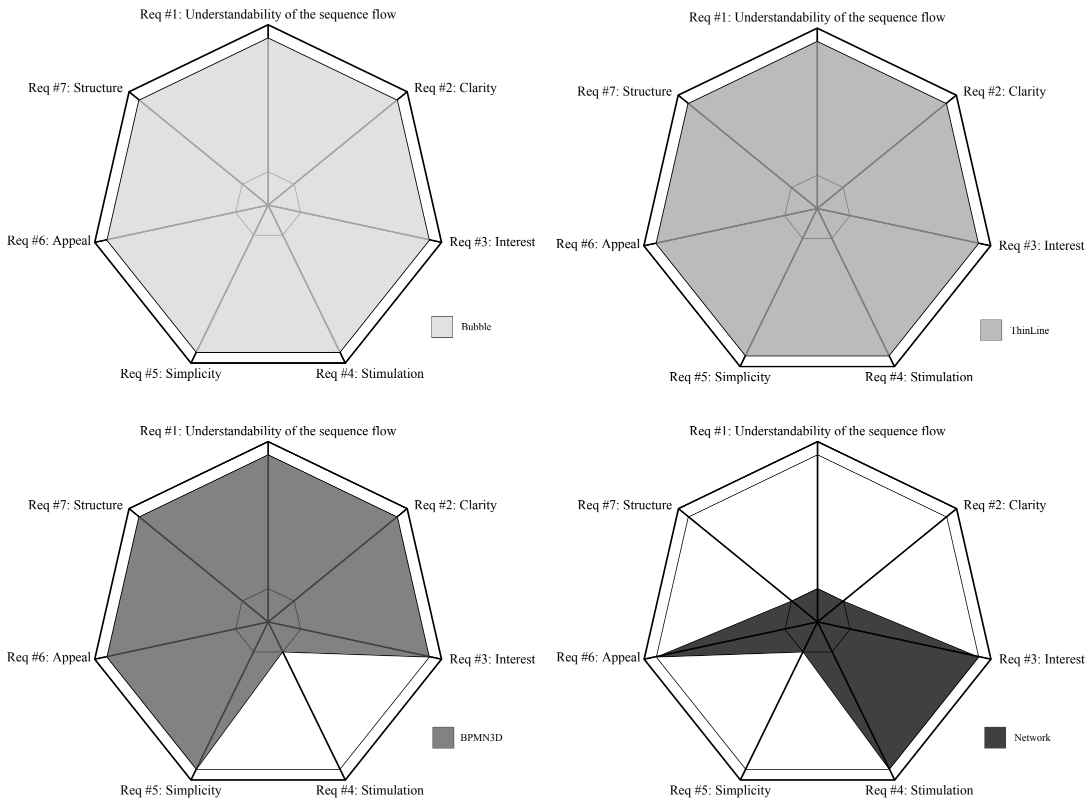


Figure 13.5: Requirements met by developed visualizations.

with its understandability. Therefore, the way ProNaVis is introduced to users will constitute a key factor.

Second, the time period the participants worked with the ProNaVis concepts during the experiment was limited to 30-45 minutes per participant. In this context, we could not conclude that the results can be transferred to real-world scenarios, in which ProNaVis concepts will have been used over longer periods (e.g., multiple years). Therefore, future research must include field studies in a real-world environment to confirm our experiment results over longer time periods as well.

Third, the complexity level of the used PMC was chosen very low in order to avoid difficulties in understanding process contents. Furthermore, we chose subjects having similar prior knowledge about BPM to ensure that only the different navigation concepts constitute factors to be investigated. However, in practice, the complexity of process models differs within companies. Furthermore, employees have different levels of knowledge (e.g., experiences staff compared to new employees). We assume that these factors affect the understandability of process navigation concepts as well. Therefore, future work should also include *multi-factor* experiments [JM01] taking into account the following factors: *navigation concept*, *complexity of process model collections*, and *level of knowledge of participants*.

RQ #5: How can comprehensibility and aesthetic appearance of process visualizations be measured?

Comprehensibility [MRC07], *aestetic appearance* [Bir33], and *clarity* [RM11] have been identified as key characteristics for visualizing process models. To answer RQ #5, we performed a second user experiment investigating these factors (cf. Chapter 11). Performed as a *single factor* experiment, it only considered the *visualization* of a process model. Therefore, different visualizations were applied to the same process models. Results indicate that there have been significant differences between the different visualization concepts.

However, some limitations need to be discussed and picked up in future work.

First, based on the experiment results, we noticed that presenting data objects apart from other process elements could potentially increase the understandability of process models (cf. *ThinLine* and *BPMN3D* concepts). Thereby, subjects are enabled to faster differentiate between data objects and other process elements as they are presented in different areas on the screen. In future work, this topic must be taken into account.

Second, the experiment neglected a few important factors, which potentially have affected the results. Examples include the *complexity of process models* and the *knowledge level of participants*. The impact of these factors should be investigated in a *multi-factor* experiment.

RQ #6: How does the navigation concept support process participants in their daily work?

To answer RQ #6, ProNaVis provides navigation concepts in terms of three navigation dimensions. These dimensions allow process participants to navigate within a PMC, i.e., they support process participants to navigate to the needed information in the right level of detail. Further, more different visualization types can be chosen. These concepts as well as their limitations are summarized in the following.

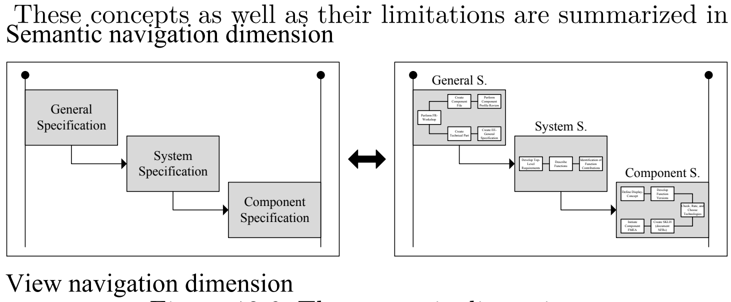


Figure 13.6: The semantic dimension.

In the *semantic dimension*, PMC may be displayed in different levels of detail. On a high semantic level, for example, only the names of the process areas shall be shown (cf. Figure 13.6). If the semantic level of the respective process area shall be increased, additional details (e.g., duration, responsible roles, and contact persons) may be shown as well. The semantic dimension is created based on the given PMC, i.e., on the hierarchical structure of the given process space. It allows deriving of a semantic dimension for any given PMC. In future work, however, alternative concepts should be applied as well (e.g., from the area of ZUI [RB09]).

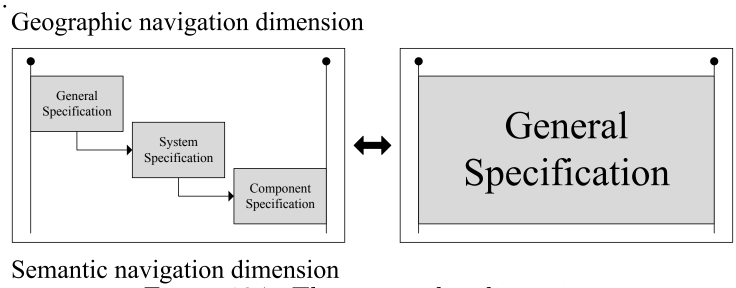


Figure 13.7: The geographic dimension.

The *geographic dimension* allows for a visual zooming without changing the level of detail (cf. Figure 13.7). Think of a magnifier while reading a newspaper. To set different geographic levels, we refer to a specific reference object (cf. Section 4.4.2). Thereby, the scale of the geographic dimension always refers to this object. However, geographic zooming on a free scale has not been considered by ProNaVis. In the area of user interface design, Wijk et al. [vWN03] have already introduced such techniques. In particular, they provide animation techniques to support users in keeping the overview of the environment when navigating along the geographic dimension.

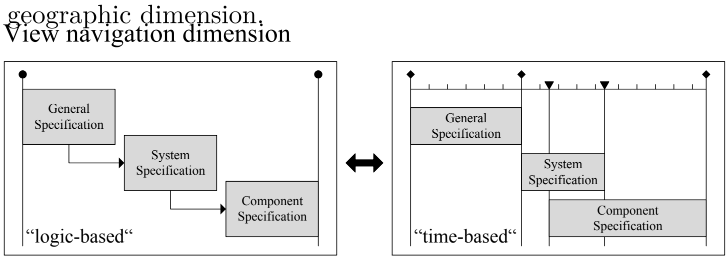


Figure 13.8: The visualization dimension.

The *visualization dimension* allows users to focus different process information such as time, documents, contact persons, or logical relationships with other information (cf Figure 13.8). As opposed to the semantic dimension, the information displayed remains on a constant level of detail, i.e., only the point of view is changed. Specifically, we introduced four different visualization types. The *time-based* visualization type emphasizes the time perspective. The *logic-based* visualization type accentuates logic relations between process steps. Finally, the *text-based* visualization represents task descriptions. Finally, the *list-based* visualization provides a list with all process elements. However, other existing visualization types should be considered in future work as well (e.g., [KKR12, BRB07, LKR13, KFKF12]).

Department	Employees	Process Models	Documents	Area
Business Unit A	257	50	290	Bus
Business Unit B	47	15	60	Truck
Business Unit C	37	23	30	Car
Business Unit D	23	4	10	Car

Table 13.2: Details on the use of Compass.

Altogether, ProNaVis provides a generic navigation and visualization framework for complex process model collections. The developed prototypes (cf. Chapter 9) and their evaluation provide evidence that a 3-dimensional navigation approach supports process participants in their daily work. Specifically, Compass was successfully implemented in a real-world environment. Table 13 illustrates how Compass has been used by an automotive OEM.

Compass implements the generic ProNaVis functions, but has still been customized for the automotive domain. In turn, ProNavigator provides a generic, non-domain specific approach, but still lacks functionality. For future field studies, it would be interesting, for example, to apply Compass to other domains as well (e.g., the logistics or the financial sector). s In summary, with ProNaVis, this thesis made a significant contribution in the area of business process management (BPM), specifically concerning the user-adequate navigation and visualization of PMC. The presented research questions introduced in Chapter 1 have been addressed throughout the entire thesis. Figure 13.9 illustrates which chapters have answered them in detail.

Research Questions / Chapters	I		II					III					IV	
	Chapter 1	Chapter 2	Chapter 3	Chapter 4	Chapter 5	Chapter 6	Chapter 7	Chapter 8	Chapter 9	Chapter 10	Chapter 11	Chapter 12	Chapter 13*	Chapter 14*
Research Question 1	●	●						●						
Research Question 2			●	●	●		●		●	●	●	●		
Research Question 3						●			●	●	●	●		
Research Question 4					●		●		●	●	●	●		
Research Question 5						●			●	●	●	●		
Research Question 6									●	●	●	●		

* not relevant with respect to the defined research questions

Figure 13.9: Answering the research questions.

14 Summary and Outlook

Enterprises and organizations struggle with the increasing size and complexity of process model collections (PMC). A particular challenge constitutes the handling of PMC by domain experts or subject matter experts. Typically, existing PMC are presented to these user groups as well as to the process participants themselves in a rather static manner, e.g., as images not allowing for any context-specific user interaction. However, as the various user groups have different roles and needs, such rigid approaches are by far not sufficient to assist them in their daily work.

To tackle this challenge, the thesis introduced the *Process Navigation and Visualization (ProNaVis)* framework. In particular, ProNaVis provides navigation concepts for complex PMC. In detail, navigation is based on a 3-dimensional navigation space, which comprises three independent navigation dimensions allowing for a flexible navigation within a PMC.

Starting with two case studies and an online survey we were able to gather insights into practical issues and challenges related to PMC navigation and visualization. Based on real-world use cases and two case studies, we then derived fundamental requirements for designing the ProNaVis framework. Picking up ideas from Google Maps, we developed a PMC *navigation space*, consisting of three navigation dimension; i.e., the *semantic*, the *geographic*, and the *visualization* dimension. Thereby, the semantic and geographic dimensions are independent from each other, which distinguishes ProNaVis from related approaches. Furthermore, to provide a sound basis we formalized the developed navigation concepts. Moreover, we presented different PMC visualization types as well as specific visualization concepts for the logic-based visualization of process models.

We validated the ProNaVis framework and practically applied it in cooperation with an industrial partner. In the latter context, selected ProNaVis concepts were implemented in *Compass*–a tool that allows navigating in complex PMC in the area of E/E engineering processes. With *ProNavigator* and *iCare*, we further realized two additional prototypes implementing ProNaVis concepts in other domains. Moreover, in a controlled experiment we were able to demonstrate practical benefits of the 3-dimensional ProNaVis navigation space compared to a 1-dimensional navigation space. In another experiment, we showed that ProNaVis visualization concepts are more comprehensible to users compared to standard process model notations. Finally, we combined ProNaVis with the process-oriented information logistics (POIL) approach to illustrate the generic applicability of the ProNaVis navigation concepts.

In summary, the main contributions of this thesis are as follows:

- The case study research allowed identifying fundamental real-world use cases for process navigation.
- Generic requirements on the navigation and visualization of PMC were elicited.
- A *process space* for PMC consisting of three independent navigation dimensions was designed and formalized.
- Four novel PMC *visualization concepts* were introduced.

- The practical applicability of the ProNaVis approach was demonstrated.
- In two experiments, we provided evidence that the navigation and visualization concepts perform better than existing navigation approaches.

The development of the ProNaVis concepts has not come to an end yet. Future work will become necessary, for example, to further evaluate the use of ProNaVis in practice. In particular, process participants should be provided with the developed navigation concepts over a longer time period in order to get familiar with them. Moreover, it will be crucial to take performance issues into account as well, i.e., to ensure that the developed framework is scalable and will be applicable even when facing repositories with thousands of process models or dealing with very large process models. Furthermore, the applicability of ProNaVis should be validated in other domains as well.

Future work on ProNaVis must also address various disciplines in the BPM area that emerged during the last years. For example, a PMC might comprise process families [ATR+12, ATW+13]; i.e., collections of related process model variants that share common parts, but may also exhibit variant-specific parts depending on the context model variants are used [HBR10, Hal10]. The challenge will be to adopt the described navigation concept to be also applicable when facing process families. Another challenge will be to provide navigation support for approaches targeting at a tighter integration of processes, data and users from the very beginning [KR11, Kün13]. Finally, cross-organizational processes must be also considered when further developing ProNaVis. Thereby, the challenge will be to cope with collaborative processes and adopt the presented concepts to navigate within process choreographies as well [FIRMR15].

Finally, it needs to be investigated how ProNaVis concepts can be integrated with existing process modeling tools. We showed that visualizing process models in the same way as modeled by process designers is far from being appropriate for process participants in their different roles and domains. This thesis, however, indicates that business process management should prioritize end user needs over functional complexity of modeling tools. Therefore, a much stronger consideration of *user interface design* and *usability engineering* will be required in future.

Bibliography

[AB05] Ishtiaq Ahmed and James Blustein. *Navigation in information space.* in: Proc 2nd Int'l Conf on Web Based Communities (WBC'05), pp. 281–286, IADIS, 2005.

[ARI07] *IDS Scheer AG: ARIS Web Publisher 7.0.2.* Fact Sheet, 2007.

[ATR+12] Clara Ayora, Victoria Torres, Manfred Reichert, Barbara Weber, and Vicente Pelechano. *Towards Run-time Flexibility for Process Families: Open Issues and Research Challenges.* in: Proc 2nd Int'l Workshop on Process Model Collections (PMC'12), LNBIP 132, pp. 477–488, Springer, 2012.

[ATW+13] Clara Ayora, Victoria Torres, Barbara Weber, Manfred Reichert, and Vincente Pelechano. *Enhancing Modeling and Change Support for Process Families through Change Patterns.* in: Proc 14th Int'l Working Conference on Business Process Modeling, Development, and Support (BPMDS'13), LNBIP 147, pp. 246–260, Springer, 2013.

[Bas05] Sarita Bassil. *Workflow technology for complex socio-technical systems.* PhD Thesis, Universite de Montreal, 2005.

[BBR06] Ralph Bobrik, Thomas Bauer, and Manfred Reichert. *Proviado – Personalized and Configurable Visualizations of Business Processes.* in: Proc 7th Int'l Conf on Electronic Commerce and Web Technologies (EC-WEB'06), LNCS 4082, pp. 61-71, Springer, 2006.

[BEH+08] Stefanie Betz, Daniel Eichhorn, Susan Hickl, Stefan Klink, Agnes Koschmider, Yu Li, Andreas Oberweis, and Ralf Trunko. *3D Representation of Business Process Models.* Modellierung betrieblicher Informationssysteme - Modellierung zwischen SOA und Compliance Management (MobIS'08), LNI 141, pp. 73–87, Springer, 2008.

[BEL+07] Sabine Buckl, Alexander M. Ernst, Josef Lankes, Florian Matthes, Christian M. Schweda, and André Wittenburg. *Generating Visualizations of Enterprise Architectures using Model Transformations.* in: Proc Enterprise Modelling and Information Systems Architectures (EMISA'07), LNI P-119, pp. 33–46, Springer, 2007.

[Ben01] David Benyon. *The new HCI navigation of information space.* in: J Knowl.-Based Syst, 14(8), pp. 425–430, 2001.

[BGM04] Benjamin B. Bederson, Jesse Grosjean, and Jon Meyer. *Toolkit Design for Interactive Structured Graphics.* in: J IEEE Trans. Software Eng., 30(8), pp.535-546, 2004.

[BH94] Benjamin B. Bederson and James D. Hollan. *Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics.* in: Proc 7th ACM Symposium on User Interface Software and Technology (UIST'94), pp. 17–26, ACM, 1994.

[BHP+96] Benjamin B. Bederson, James D. Hollan, Ken Perlin, Jonathan Meyer, David Bacon, and George W. Furnas. *Pad++: A Zoomable Graphical Sketchpad For Exploring Alternate Interface Physics.* in: J Vis. Lang. Comput., 7(1), 27, pp. 3–32, ACM, 1996.

[Bir33] George D. Birkhoff. *Aesthetic Measure*. Cambridge Massachusetts University Press, 1933.

[BK08] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. The MIT Press, 2008.

[BKK04] Sarita Bassil, Rudolf K. Keller, and Peter G. Kropf. *A Workflow-Oriented System Architecture for the Management of Container Transportation*. in: Proc 2nd Int'l Conf on Business Process Management (BPM'04), LNCS 3080, pp. 116–131, Springer, 2004.

[BLW+12] Irene Barba, Andreas Lanz, Barbara Weber, Manfred Reichert, and Carmelo Del Valle. *Optimized Time Management for Declarative Workflows*. in: Proc 13th BPMDS'12 Working Conference (BPMDS'12), LNBIP 113, pp. 195–210, Springer, 2012.

[BM98] Ben Bederson and Jon Meyer. *Abstract Implementing a Zooming User Interface: Experience Building Pad++*. in: J on Software: Practice and Experience, 28(10), pp. 1101–1135, 1998.

[BMG00] Benjamin B. Bederson, Jon Meyer, and Lance Good. *Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java*. in: Proc 13th ACM Symposium on User Interface Software and Technology (UIST'00), pp. 171–180, ACM, 2000.

[Bob08] Ralph Bobrik. *Konfigurierbare Visualisierung komplexer Prozessmodelle*. PhD Thesis, University of Ulm, 2008.

[BR09] Ross A. Brown and Jan C. Recker. *Improving the Traversal of Large Hierarchical Process Repositories*. in: Proc 20th Australasian Conference on Information Systems (ACIS'09), pp. 144–153, 2009.

[BRB05] Ralph Bobrik, Manfred Reichert, and Thomas Bauer. *Requirements for the Visualization of System-Spanning Business Processes*. in: Proc 1st Int'l Workshop on Business Process Monitoring and Performance Management (BPMPM'05), pp. 948–954, IEEE, 2005.

[BRB07] Ralph Bobrik, Manfred Reichert, and Thomas Bauer. *View-Based Process Visualization*. in: 5th Int'l Conf on Business Process Management (BPM'07), LNCS 4714, pp. 88–95, Springer, 2007.

[BRW11] Ross Brown, Jan Recker, and Stephen West. *Using virtual worlds for collaborative business process modeling*. in: Business Proc Manag Journal 17(3), pp. 546–564, 2011.

[BRWSH86] Victor R. Basili, Jr. Richard W. Selby, and David H. Hutchens. *Experimentation in Software Engineering*. in: J IEEE Trans. Software Engineering, pp. 733–743, 1986.

[BWJ02] Claudio Bettini, Xi-Shi Wang, and Suchil Jajodia. *Temporal reasoning in workflow systems*. in: J on Distributed and Parallel Databases, 11(3), pp. 269-306, 2002.

[BWS93] Benjamin. B. Bederson, Richard. S. Wallace, and Eric. L. Schwartz. *Control and Design of the Spherical Pointing Motor*. in: Proc Int'l Conf on Robotics and Automation: Volume 2, pp.630–636, IEEE, 1993.

[CDT00] Chun Wei Choo, Brian Detlor, and Don Turnbull. *Information Seeking on the Web: An Integrated Model of Browsing and Searching*. in: J First Monday, 5(2), 2000.

[CGJ+07] Carlo Combi, Matteo Gozzi, Jose M. Juarez, Barbara Oliboni, and Giuseppe Pozzi. *Conceptual Modeling of Temporal Clinical Workflows*. in: Proc 20th Int'l Symp on Temporal Representation and Reasoning, pp. 70–81, IEEE, 2007.

[CGPP12] Carlo Combi, Matteo Gozzi, Roberto Posenato, and Giuseppe Pozzi. *Conceptual Modeling of Flexible Temporal Workflows.* in: J on Trans. Auton. Adapt. Syst., 7(2), pp. 19:1–19:29, ACM, 2012.

[Cha93] Matthew Chalmers. *Using a Landscape Metaphor to Represent a Corpus of Documents.* in: Proc Conf on Spatial Information Theory (COSIT'93), LNCS 716, pp. 377–390, Springer, 1993.

[Cla22] Wallace Clark. *The Gantt Chart.* The Ronald Press Co., 1922.

[CMS99] Stuart K. Card, Jock D. Mackinlay, and Ben Schneiderman. *Readings in Information Visualization: Using Vision to Think (Interactive Technologies).* Morgan Kaufmann, 1999.

[Coo09] Hugh Coolican. *Research Methods and Statistics in Psychology.* Routledge, 2009.

[CRM91] Stuart K. Card, George G. Robertson, and Jock D. Mackinlay. *The Information Visualizer, an Information Workspace.* in: Proc Conf Human Factors in Computing Systems (CHI'91), pp. 181–188, ACM, 1991.

[CS91] Paul Chandler and John Sweller. *Cognitive Load Theory and the Format of Instruction.* in: J Cognition and Instruction 8(4). pp. 293–332, 1991.

[DDF+90] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. *Indexing by Latent Semantic Analysis.* in: J on the American Society for Information Science, 41(6), pp. 391–407, 1990.

[DF98] Andreas Dieberger and Andrew U. Frank. *A City Metaphor to Support Navigation in Complex Information Spaces.* in: J Vis. Lang. Comput., 9(6), pp. 597–622, 1998.

[Dij76] Edsger W. Dijkstra. *A Discipline of Programming.* Prentice-Hall, 1976.

[DO01] Rene Descartes and Paul J. Olscamp. *Discourse on Method, Optics, Geometry, and Meteorology.* Hackett Publishing Co., 2001.

[Eff12] Philip Effinger. *A 3D-Navigator for Business Process Models.* in: Proc 1st Int'l Workshop on Theory and Applications of Process Visualization (TAProViz'12), LNBIP 132, pp. 737–743, Springer, 2012.

[EM00] Angela Edmunds and Anne Morris. *The Problem of Information Overload in Business Organisations: A Review of the Literature.* in: Int'l J on Information Management, 20(1), pp. 17–28, 2000.

[EPPR99] Johann Eder, Euthimios Panagos, Heinz Pozewaunig, and Michael Rabinovich. *Time Management in Workflow Systems.* in: Proc 3rd Int'l Conf on Business Information Systems (BIS'99), LNBIP 21, pp. 265–280, Springer, 1999.

[EPR99] Johann Eder, Euthimios Panagos, and Michael Rabinovich. *Time Constraints in Workflow Systems.* in: Proc 11th Int'l Conf on Advanced Information Systems Engineering (CAiSE'99), LNCS 1626, pp. 286–300, Springer, 1999.

[ES10] Philip Effinger and Johannes Spielmann. *Lifting business process diagrams to 2.5 dimensions.* in: Proc Visualization and Data Analysis (VDA'10), 7530, p. 75300, SPIE, 2010.

[Fie13] Andy Field. *Discovering Statistics using IBM SPSS Statistics.* SAGE Publications Ltd, 2013.

[FIRMR15] Walid Fdhila, Conrad Indiono, Stefanie Rinderle-Ma, and Manfred Reichert. *Dealing with change in process choreographies: Design and implementation of propagation algorithms.* in: J on Information Systems, 49, pp. 1–24, Elsevier, 2015.

[Fow04] Martin Fowler. *UML Distilled: A Brief Guide to the Standard Object Modeling Language.* Addison-Wesley (Addison-Wesley object technology series), 2004.

[FR01] Michael T. French and John S. Robotham. *Time inheritance scene graph for representation of media content.* Google Patents (US6266053 B1), 2001.

[FRS+10] Marie-Christine Fauvet, Marcello La Rosa, Mehrad Sadegh, Abdurrahman Alshareef, Remco M. Dijkman, Luciano García-Bañuelos, Hajo A. Reijers, Wil M. P. van der Aalst, Marlon Dumas, and Jan Mendling. *Managing Process Model Collections with AProMoRe.* in: Proc 8th Int'l Conf on Service Oriented Computing (ICSOC'10), LNCS 6470, pp. 699–701, Springer, 2010.

[FT94] Andrew U. Frank and Sabine Timpf. *Multiple representations for cartographic objects in a multi-scale tree - An intelligent graphical zoom.* in: J Computer & Graphics, 18(6), pp. 823–829, 1994.

[Ger05] German Association of the Automotive Industry (VDA). *Engineering Change Management. Part 1: Engineering Change Request (ECR).* V 1.1, Doc. No. 4965, 2005.

[GMR07] Fredrik Gundelsweiler, Thomas Memmel, and Harald Reiterer. *ZEUS - Zoomable Explorative User Interface for Searching and Object Presentation.* Symposium on Human Interface (HCI'07), LNCS 4557, pp. 288–297, Springer, 2007.

[GMS+13] Gregor Grambow, Nicolas Mundbrod, Vivian Steller, Manfred Reichert, Andreas Schiffleitner, Thomas Bley, and Christian Feick. *State-of-the-Art and Requirements for Collecting and Managing Sustainability Data Along Today's Supply Chains.* in: 6th Int'l Conf on Life Cycle Management (LCM'13), pp. 548–552, 2013.

[GOR12] Gregor Grambow, Roy Oberhauser, and Manfred Reichert. *Enabling Automatic Process-aware Collaboration Support in Software Engineering Projects.* in: Selected Papers of the ICSOFT'11 Conference, CCIS 303, pp. 73–89, Springer, 2012.

[GOR13] Gregor Grambow, Roy Oberhauser, and Manfred Reichert. *Automated Software Engineering Process Assessment: Supporting Diverse Models using an Ontology.* in: Int'l J on Advances in Software, 6 (1 & 2), pp. 213–224, 2013.

[GRF08] Tera Marie Green, William Ribarsky, and Brian D. Fisher. *Visual analytics for complex concepts using a human cognition model.* in: Proc Int'l IEEE Symposium on Visual Analytics Science and Technology (VAST'08), pp. 91–98, IEEE, 2008.

[GRMR+08] Christian W. Guenther, Stefanie Rinderle-Ma, Manfred Reichert, Wil M. P. van der Aalst, and Jan Recker. *Using Process Mining to Learn from Process Changes in Evolutionary Systems.* in: Int'l J on Business Process Integration and Management, Special Issue on Business Process Flexibility, 3(1), pp.61–78, 2008.

[GRRv06] Christian W. Guenther, Stefanie Rinderle, Manfred Reichert, and Wil M. P. van der Aalst. *Change Mining in Adaptive Process Management Systems.* in: Proc 14th Int'l Conf on Cooperative Information Systems (Coopls'06), LNCS 4275, pp. 309–326, Springer, 2006.

[Hal10] Alena Hallerbach. *Management von Prozessvarianten.* PhD Thesis, Ulm University, 2010.

[Hav05] Michael Havey. *Essential Business Process Modeling.* O'Reilly Media, 2005.

[HBR10] Alena Hallerbach, Thomas Bauer, and Manfred Reichert. *Capturing Variability in Business Process Models: The Provop Approach.* in: J on Software Maintenance and Evolution: Research and Practice, 22(6-7), pp. 519–546, Wiley InterScience, 2010.

[HCC12] Ian Heywood, Sarah Cornelius, and Steve Carver. *An Introduction to Geographical Information Systems (4th Edition).* Prentice Hall, 2012.

[HFL12] Constantin Houy, Peter Fettke, and Peter Loos. *Understanding Understandability of Conceptual Models - What Are We Actually Talking about?* in: Proc 31st Int'l Conference on Conceptual Modeling (ER'12), LNCS 7532, pp. 64–77, Springer, 2012.

[HH93] Deborah Hix and H. Rex Hartson. *Developing User Interfaces: Ensuring Usability Through Product & Process.* John Wiley & Sons, Inc., 1993.

[Hil94] Wiliam C. Hill. *Recommending and evaluating items on the basis of communal history of use.* in: Bellcore, Technical Report #TM-ARH-023560, NJ 07960, 1994.

[HMM00] Ivan Herman, Guy Melançon, and M. Scott Marshall. *Graph Visualization and Navigation in Information Visualization: A Survey.* in: J IEEE Transactions on Visualization and Computer Graphics, 6(1), pp. 24–43, 2000.

[HMMR13] Markus Hipp, Bernd Michelberger, Bela Mutschler, and Manfred Reichert. *A Framework for the Intelligent Delivery and User-adequate Visualization of Process Information.* in: Proc 28th Symposium on Applied Computing (SAC'13), pp. 1383–1390, ACM, 2013.

[HMMR14] Markus Hipp, Bernd Michelberger, Bela Mutschler, and Manfred Reichert. *Navigating in Process Model Repositories and Enterprise Process Information.* in: Proc 8th Int'l Conf on Research Challenges in Information Science (RCIS'14), IEEE, pp. 1–12, 2014.

[HMPR04] Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. *Design Science in Information Systems Research.* in: J MIS Quarterly, 28(1), pp. 75–105, 2004.

[HMR11a] Markus Hipp, Bela Mutschler, and Manfred Reichert. *Navigating in Process Model Collections: A new Approach Inspired by Google Earth.* in: Proc 1st Int'l Workshop on Process Model Collections (PMC'11), LNBIP 100, pp. 87–98, Springer, 2011.

[HMR11b] Markus Hipp, Bela Mutschler, and Manfred Reichert. *On the Context-aware, Personalized Delivery of Process Information: Viewpoints, Problems, and Requirements.* in: Proc 6th Int'l Conf on Availability, Reliability and Security (ARES'11), LNCS 6908, pp. 390–397, Springer, 2011.

[HMR12] Markus Hipp, Bela Mutschler, and Manfred Reichert. *Navigating in Complex Business Processes.* in: Proc 23rd Int'l Conf on Database and Expert Systems Applications (DEXA'12), LNCS 7447, pp. 466–480, Springer, 2012.

[HNP05] Andreas Hotho, Andreas Nürnberger, and Gerhard Paa ss. *A Brief Survey of Text Mining.* in: J for Computational Linguistics and Language Technology, 20(1), pp. 19–62, 2005.

[Hon05] Seok-Hee Hong. *MultiPlane: A New Framework for Drawing Graphs in Three Dimensions.* in: Graph Drawing, 13th Int'l Symp, LNCS 3843, pp. 514–515, Springer, 2005.

[HRW00] Martin Höst, Björn Regnell, and Claes Wohlin. *Using Students as Subjects-A Comparative Study of Students and Professionals in Lead-Time Impact Assessment.* in: J Empirical Software Engineering, 3(5), pp. 201–214, 2000.

[HSM+14] Markus Hipp, Achim Strauss, Bernd Michelberger, Bela Mutschler, and Manfred Reichert. *Enabling a User-Friendly Visualization of Business Process Models.* in: Proc 3rd Int'l Workshop on Theory and Applications of Process Visualization (TaProViz'14), pp. 395-407, 2014.

[IBM06] *Best Practices for Using WebSphere Business Modeler and Monitor.* IBM Redbook Paper, 2006.

[ISO95] *Ergonomic requirements for office work with visual display terminals (VDTs) - Part 10: Dialogue principles.* European Standard EN ISO 9241-10, 1995.

[ISO98] *Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11 : Guidance on usability.* European Standard EN ISO 9241-11, 1998.

[JF98] Susanne Jul and George W. Furnas. *Critical Zones in Desert Fog: Aids to Multiscale Navigation.* in: ACM Symposium on User Interface Software and Technology, pp. 97–106, 1998.

[JG07] Stefan Jablonski and Manuel Goetz. *Perspective Oriented Business Process Visualization.* in: Proc Business Process Management Workshops (BPM'07), LNCS 4928, pp. 144–155, Springer, 2007.

[JGH+08] Robert J. K. Jacob, Audrey Girouard, Leanne M. Hirshfield, Michael S. Horn, Orit Shaer, Erin Treacy Solovey, and Jamie Zigelbaum. *Reality-based interaction: a framework for post-WIMP interfaces.* in: Proc Int'l Conf on Human Factors in Computing Systems (CHI'08), pp. 201–210, ACM, 2008.

[JKGR08] Hans-Christian Jetter, Werner A. König, Jens Gerken, and Harald Reiterer. *ZOIL - A Cross-Platform User Interface Paradigm for Personal Information Management.* in: Proc CHI 2008 Workshop - The Disappearing Desktop: Personal Information Management (CHI'08), pp. 1–9, ACM, 2008.

[JM01] Natalia Juristo Juzgado and Ana María Moreno. *Basics of software engineering experimentation.* Kluwer, 2001.

[KFKF12] Sonja Kabicher-Fuchs, Simone Kriglstein, and Kathrin Figl. *Timeline Visualization for Documenting Process Model Change.* in: Proc 5th Int'l Workshop on Enterprise Modelling and Information Systems Architectures (EMISA'12), LNI 206, pp. 95–108, Springer, 2012.

[KKR12] Jens Kolb, Klaus Kammerer, and Manfred Reichert. *Updatable Process Views for User-centered Adaption of Large Process Models.* in: Proc 10th Int'l Conf on Service Oriented Computing (ICSOC'12), LNCS 7636, pp. 484–498, Springer, 2012.

[Kli99] Paul Kline. *Handbook of Psychological Testing.* Routledge, 1999.

[KLMR13] Jens Kolb, Henrik Leopold, Jan Mendling, and Manfred Reichert. *Creating and Updating Personalized and Verbalized Business Process Descriptions.* in: Proc 6th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling (PoEM'13), LNBIP 165, pp. 191–205, Springer, 2013.

[Kol15] Jens Kolb. *Abstraction, Visualization, and Evolution of Process Models.* PhD Thesis, Ulm University, 2015.

[KR11] Vera Künzle and Manfred Reichert. *PHILharmonicFlows: towards a framework for object-aware process management.* in: J on Software Maintenance and Evolution: Research and Practice, 23(4), pp. 205–244, Wiley, 2011.

[KR13a] Jens Kolb and Manfred Reichert. *Data Flow Abstractions and Adaptations through Updatable Process Views.* in: Proc 28th Symposium on Applied Computing (SAC'13), pp. 1447-1453, ACM, 2013.

[KR13b] Jens Kolb and Manfred Reichert. *A Flexible Approach for Abstracting and Personalizing Large Business Process Models.* in: J on Applied Computing Review, 13(1), pp. 6–17, ACM SIGAPP, 2013.

[KR13c] Jens Kolb and Manfred Reichert. *Supporting Business and IT through Updatable Process Views: The proView Demonstrator.* Demo Track of the 10th Int'l Conference on Service Oriented Computing (ICSOC'12), LNCS 7759, pp. 460–464, Springer, 2013.

[KRM12] Simone Kriglstein and Stefanie Rinderle-Ma. *A Visualization Concept for High-Level Comparison of Process Model Versions.* in: Proc Business Process Management Workshops (BPM'12), LNBIP 132, pp. 465-476, Springer, 2012.

[KRR09] Werner A. König, Roman Rädle, and Harald Reiterer. *Squidy: a zoomable design environment for natural user interfaces.* in: Proc 27th Int'l Conf on Human Factors in Computing Systems (CHI'09), pp. 4561–4566, ACM, 2009.

[KRW12] Jens Kolb, Manfred Reichert, and Barbara Weber. *Using Concurrent Task Trees for Stakeholder-centered Modeling and Visualization of Business Processes.* in: Proc 4th Int'l Conf on Subject-Oriented Business Process Management (S-BPM ONE'12), CCIS 284, pp. 237–251, Springer, 2012.

[Kün13] Vera Künzle. *Object-Aware Process Management.* PhD Thesis, Ulm University, 2013.

[KWRM13] Simone Kriglstein, Günter Wallner, and Stefanie Rinderle-Ma. *A Visualization Approach for Difference Analysis of Process Models and Instance Traffic.* in: Proc 11th Int'l Conf on Business Process Management (BPM'13), LNCS 8094, pp. 219–226, Springer, 2013.

[LJ08] George Lakoff and Mark Johnson. *Metaphors We Live By (2nd).* University of Chicago Press, 2008.

[LKR13] Andreas Lanz, Jens Kolb, and Manfred Reichert. *Enabling Personalized Process Schedules with Time-aware Process Views.* in: Proc CAiSE 2013 Workshops, 2nd Int'l Workshop on Human-Centric Information Systems (HCIS'13), LNBIP 148, pp. 205–216, Springer, 2013.

[LPCR13] Andreas Lanz, Roberto Posenato, Carlo Combi, and Manfred Reichert. *Controllability of Time-Aware Processes at Run Time.* in: Proc 21st Int'l Conf on Cooperative Information Systems (CoopIS 2013), LNCS 8185, pp. 39–56, Springer, 2013.

[LPR12] Richard Lenz, Mor Peleg, and Manfred Reichert. *Healthcare Process Support: Achievements, Challenges, Current Research.* in J on Knowledge-Based Organizations (IJKBO), 2(4), 2012.

[LR07] Richard Lenz and Manfred Reichert. *IT Support for Healthcare Processes - Premises, Challenges, Perspectives.* in: Data and Knowledge Engineering, 61(1), pp. 39–58, 2007.

[LR14] Andreas Lanz and Manfred Reichert. *Dealing with Changes of Time-Aware Processes.* in: Proc 12th Int'l Conf on Business Process Management (BPM 2014), LNCS 8659, pp. 217–233, Springer, 2014.

[LWR14] Andreas Lanz, Barbara Weber, and Manfred Reichert. *Time patterns for process-aware information systems.* in: Requirements Engineering, 19(2), pp.113–141, Springer, 2014.

[MAGM13] Bernd Michelberger, Ralph-Josef Andris, Hasan Girit, and Bela Mutschler. *A Literature Survey on Information Logistics.* in: 16th Int'l Conf on Business Information Systems (BIS'13), LNBIP 157, pp. 138-150, Springer, 2013.

[May99] Deborah J. Mayhew. *The Usability Engineering Lifecycle: A Practitioner's Guide to User Interface Design.* Morgan Kaufmann Publishers, 1999.

[May01] Harvey Maylor. *Beyond the gantt chart: Project management moving on.* in: European Management Journal, 19(1), pp. 92–100, 2001.

[MBN04] Jan Mendling, Alberto Brabenetz, and Gustaf Neumann. *EPML2SVG - Generating Websites from EPML Processes.* in: Proc 3rd GI Workshop on Event-Driven Process Chains (EPK'04), pp. 55–64, 2004.

[Men08] Jan Mendling. *Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness.* Springer, 2008.

[MHHR06] Dominic Müller, Joachim Herbst, Markus Hammori, and Manfred Reichert. *IT Support for Release Management Processes in the Automotive Industry.* in: Proc 4th Int'l Conf on Business Process Management (BPM'06), LNCS 4102, pp. 368–377, Springer, 2006.

[MHM15] B. Michelberger, M. Hipp, and B. Mutschler. *Process-oriented Information Logistics: Requirements, Techniques, Application.* in: Advances in Intelligent Process-Aware Information Systems, 2015. Accepted for Publication.

[Mic90] Joel Michell. *An Introduction To the Logic of Psychological Measurement.* Psychology Press, 1990.

[Mic15] Bernd Michelberger. *Process-Oriented Information Logistics: Aligning Process Information with Business Processes.* PhD Thesis, Ulm University, 2015.

[MKR12] Nicolas Mundbrod, Jens Kolb, and Manfred Reichert. *Towards a System Support of Collaborative Knowledge Work.* in: Proc 1st Int'l Workshop on Adaptive Case Management (ACM'12), LNBIP 132, pp. 31–42, Springer, 2012.

[MMB+14] Bernd Michelberger, Bela Mutschler, Daniel Binder, Jan Meurer, and Markus Hipp. *iGraph: Intelligent Enterprise Information Logistics.* in: Proc 10th Int'l Conf on Semantic Systems (SEMANTiCS'14), Posters & Demonstrations Track, 1224, pp. 27–30, 2014.

[MMHR13] Bernd Michelberger, Bela Mutschler, Markus Hipp, and Manfred Reichert. *Determining the Link and Rate Popularity of Enterprise Process Information.* in: Proc 21st Int'l Conf on Cooperative Information Systems (CoopIS'13), LNCS 8185, pp. 112–129, Springer, 2013.

[MMN+06] Jan Mendling, Malte Moser, Gerhard Neumann, H. M. W. Verbeek, and Boudewijn F. Van Dongen. *Faulty EPCs in the SAP Reference Model.* in: Proc 4th Int'l Conf on Business Process Management (BPM'06), LNCS 4102, pp. 451–457, Springer, 2006.

[MMR11a] Bernd Michelberger, Bela Mutschler, and Manfred Reichert. *On Handling Process Information: Results from Case Studies and a Survey.* in: Proc 2nd Int'l Workshop on Empirical Research in Business Process Management (ER-BPM'11), LNBIP 99, pp. 333–344, Springer, 2011.

[MMR11b] Bernd Michelberger, Bela Mutschler, and Manfred Reichert. *Towards Process-oriented Information Logistics: Why Quality Dimensions of Process Information Matter.* in: Proc 4th Int'l Workshop on Enterprise Modelling and Information Systems Architectures (EMISA'11), LNI 190, pp. 107–120, Springer, 2011.

[MMR12a] Bernd Michelberger, Bela Mutschler, and Manfred Reichert. *A Context Framework for Process-oriented Information Logistics.* in: Proc 15th Int'l Conf on Business Information Systems (BIS'12), LNBIP 117, pp. 260–271, Springer, 2012.

[MMR12b] Bernd Michelberger, Bela Mutschler, and Manfred Reichert. *Process-oriented Information Logistics: Aligning Enterprise Information with Business Processes.* in: 16th Int'l Enterprise Distributed Object Computing Conference (EDOC'12), pp. 21–30, IEEE, 2012.

[MMRS09] Joachim Melcher, Jan Mendling, Hajo A. Reijers, and Detlef Seese. *On Measuring the Understandability of Process Models.* in: Proc Business Process Management Workshops (BPM'09), LNBIP 43, pp. 465–476, Springer, 2009.

[MPVW04] Ulrich Meissen, Stefan Pfennigschmidt, Agnès Voisard, and Tjark Wahnfried. *Context- and Situation-Awareness in Information Logistics.* in: Proc Current Trends in Database Technology Workshops (EDBT'04), pp. 335–344, Springer, 2004.

[MRB08] Bela Mutschler, Manfred Reichert, and Johannes Bumiller. *Unleashing the Effectiveness of Process-oriented Information Systems: Problem Analysis, Critical Success Factors and Implications.* in: J Transactions on Systems, Man, and Cybernetics (SMC) - Part C: Applications & Reviews, 38(3), pp. 280-291, IEEE, 2008.

[MRC07] Jan Mendling, Hajo A. Reijers, and Jorge Cardoso. *What Makes Process Models Understandable?* in: Proc 5th Int'l Confon Business Process Management (BPM'07), LNCS 4714, pp. 48–63, Springer, 2007.

[MRM+13] Bernd Michelberger, Armin Reisch, Bela Mutschler, Jörg Wurzer, Markus Hipp, and Manfred Reichert. *iCare: Intelligent Medical Information Logistics.* in: Proc 15th Int'l Conf on Information Integration and Web-based Applications & Services (iiWAS'13), pp. 396–399, ACM, 2013.

[MS08] Jan Mendling and Mark Strembeck. *Influence Factors of Understanding Business Process Models.* in: Proc 11th Int'l Conference on Business Information Systems (BIS'08), LNBIP 7, pp. 142–153, Springer, 2008.

[MUG+14] Bernd Michelberger, Klaus Ulmschneider, Birte Glimm, Bela Mutschler, and Manfred Reichert. *Maintaining Semantic Networks: Challenges and Algorithms.* in: Proc 16th Int'l Conf on Information Integration and Web-based Applications & Services (iiWAS 2014), pp. 365–374, ACM, 2014.

[Mun97] Tamara Munzner. *H3: laying out large directed graphs in 3D hyperbolic space.* in: Symposium on Information Visualization (InfoVis'97), pp. 2–10, IEEE, 1997.

[MW08] Derek Miers and Stephen A. White. *BPMN Modeling and Reference Guide - Understanding and Using BPMN.* Future Strategies Inc., 2008.

[MY12] Kazuo Misue and Seiya Yazaki. *Panoramic View for Visual Analysis of Large-Scale Activity Data.* in: Proc 1st Int'l Workshop on Theory and Applications of Process Visualization (TAProViz'12), LNBIP 132, pp. 756–767, Springer, 2012.

[ND86] Donald A. Norman and Stephen W. Draper, editors. *User Centered System Design: New Perspectives on Human-Computer Interaction.* Erlbaum, 1986.

[New96] Gregory B. Newby. *Metric Multidimensional Information Space.* in: 5th Text Retrieval Conf (TREC-5'96), 1996.

[NM02] Klaus Neumann and Martin Morlock. *Operations Research.* Hanser Fachbuch, 2002.

[Nor88] Donald A. Norman. *The Design of Everyday Things.* The MIT Press, 1988.

[OS08] Annika Oehgren and Kurt Sandkuhl. *Information Overload in Industrial Enterprises - Results of an Empirical Investigation.* in: Proc 2nd European Conf on Information Management and Evaluation (ECIME'08), pp. 343-350, Academic Publishing, 2008.

[Ped13] Volnei A. Pedroni. *Finite State Machines in Hardware - Theory and Design (with VHDL and SystemVerilog).* MIT Press, 2013.

[Pet05] Tom Petrocelli. *Data Protection and Information Lifecycle Management.* Prentice Hall, 2005.

[PF93] Ken Perlin and David Fox. *Pad: An Alternative Approach to the Computer Interface.* in: J on Computer Graphics (SIGGRAPH '93), ACM, 1993.

[PMLR15] Rüdiger Pryss, Nicolas Mundbrod, David Langer, and Manfred Reichert. *Supporting medical ward rounds through mobile task and process management.* in: J on Information Systems and e-Business Management, 13(1), pp. 107–146, Springer, 2015.

[PRJB13] Erik Poppe, Jan Recker, Daniel Johnson, and Ross Brown. *Using Natural User Interfaces for Collaborative Process Modelling in Virtual Environments.* in: Proc Workshop on Models and their Role in Collaboration (MoRoCo'13), 1037, pp. 65–67, CEUR, 2013.

[Ras00] Raskin, Jef. *Humane Interface: New Directions for Designing Interactive Systems.* Addison-Wesley Publishing, 2000.

[Ray10] Roger Rayle. *Google Earth: A Platform for Open Data.* in: J Solstice : Electronic Journal of Geography and Mathematics, Institute of Mathematical Geography, 2010.

[RB09] Harald Reiterer and Thorsten Büring. *Zooming Techniques.* Ency. of Database Systems, pp. 3684–3689, 2009.

[RC01] Mary B. Rosson and John M. Carroll. *Usability Engineering: Scenario-Based Development of Human-Computer Interaction (Interactive Technologies).* Morgan Kaufmann, 2001.

[Rec10] Jan C. Recker. *Opportunities and constraints : the current struggle with BPMN.* in: J on Business Process Management, 16(1), pp. 181–201, 2010.

[Rei11] Manfred Reichert. *What BPM Technology Can Do for Healthcare Process Support.* in: Proc 3th Conf on Artificial Intelligence in Medicine (AIME'11), LNAI 6747, pp.2–13, Springer, 2011.

[Rei12] Manfred Reichert. *Visualizing Large Business Process Models: Challenges, Techniques, Applications.* 1st Int'l Workshop on Theory and Applications of Process Visualization (TAProViz'12), BPM'12 Workshops (Invited Keynote), LNBIP 132, pp. 725–736, Springer, 2012.

[Rei13] Wolfgang Reisig. *Understanding Petri Nets - Modeling Techniques, Analysis Methods, Case Studies.* Springer, 2013.

[RFME11] Hajo A. Reijers, Thomas Freytag, Jan Mendling, and Andreas Eckleder. *Syntax highlighting in business process models.* in: J on Decision Support Systems, 51(3), pp. 339-349, 2011.

[RKBB12] Manfred Reichert, Jens Kolb, Ralf Bobrik, and Thomas Bauer. *Enabling Personalized Visualization of Large Business Processes through Parameterizable Views.* in: 27th ACM Symp. On Applied Computing (SAC'12), pp. 1653-1660, ACM, 2012.

[RM11] Hajo A. Reijers and Jan Mendling. *A Study Into the Factors That Influence the Understandability of Business Process Models.* in: J on Transactions on Systems, Man, and Cybernetics, 41(3), Part A, IEEE, 2011.

[RMD11] Hajo A. Reijers, Jan Mendling, and Remco M. Dijkman. *Human and automatic modularizations of process models to enhance their comprehension.* in: J on Information Systems, 36(5), pp. 881–897, 2011.

[Rot13] Edgar Rot. *Kontrollierte Experimente zur Verständlichkeit, Ästhetik und Übersichtlichkeit von Prozessmodellen.* Master Thesis, University of Applied Science, Ravensburg-Weingarten, Germany, 2013.

[RRD04] Stefanie Rinderle, Manfred Reichert, and Peter Dadam. *On Dealing with Structural Conflicts between Process Type and Instance Changes.* in: Proc 2nd. Int'l Conf Business Process Management (BPM'04), LNCS 3080, pp. 274–289, Springer, 2004.

[RRMD09] Manfred Reichert, Stefanie Rinderle-Ma, and Peter Dadam. *Flexibility in Process-aware Information Systems.* LNCS Transactions on Petri Nets and Other Models of Concurrency (ToPNoC), Special Issue on Concurrency in Process-aware Information Systems. Vol. 2, LNCS 5460, pp. 115–135, Springer, 2009.

[RRv+11] Marcello La Rosa, Hajo A. Reijers, Wil M. P. van der Aalst, Remco M. Dijkman, Jan Mendling, Marlon Dumas, and Luciano García-Bañuelos. *APROMORE: An advanced process model repository.* in: J Expert Syst. Appl., 38(6), pp. 7029–7040, 2011.

[RW12] Manfred Reichert and Barbara Weber. *Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies.* Springer, 2012.

[SAtDL04] Maarten W.A. Steen, David H. Akehurst, H. t. Doest, and Marc M. Lankhorst. *Supporting Viewpoint-Oriented Enterprise Architecture.* in: Proc 8th Int'l Enterprise Distributed Object Computing Conference (EDOC'04), pp. 201-211, IEEE, 2004.

[SB06] Tina Seufert and Roland Bruenken. *Cognitive load and the format of instructional aids for coherence formation.* in: J Applied Cognitive Psychology, 20(3), pp. 321–331, 2006.

[SBH+05] Hartmut Schulze, Henning Brau, Siegmar Haasis, Michael Weyrich, and Tobias Rhatje. *Human-Centered design of engineering applications: Success factors from a case study in the automotive industry.* in: J Human Factors in Ergonomics and Manufacturing, 15(4), pp. 421–443, 2005.

[Seu03a] Tina Seufert. *Supporting coherence formation in learning from multiple representations.* in: J Learning and Instruction 13(2), pp. 227–237, 2003.

[Seu03b] Tina Seufert. *Wissenserwerb mit multiplen Repräsentationen: Wirksamkeit von Kohärenzbildungshilfen.* PhD Thesis: Logos Verlag Berlin, 2003.

[SGL12] Jonas Söderland, Joana Geraldi, and Thomas Lechter. *Gantt charts revisited.* in: Int'l J on Managing Projects in Business, 5(4), pp. 578–594, 2012.

[Shn91] Ben Shneiderman. *Tree visualization with Tree-maps: A 2-d space-filling approach.* in: J ACM Transactions on Graphics, 11, pp. 92–99, 1991.

[Shn96] Ben Shneiderman. *The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations.* in: Proc Symposium on Visual Languages, pp. 336–343, IEEE, 1996.

[SJB07] Tina Seufert, Inge Jänen, and Roland Bruenken. *The impact of intrinsic cognitive load on the effectiveness of graphical help for coherence formation.* in: J Computers in Human Behavior, 23(3), pp. 1055–1071, 2007.

[SKGM12] Andreas Seyfang, Katharina Kaiser, Theresia Gschwandtner, and Silvia Miksch. *Visualizing Complex Process Hierarchies during the Modeling Process.* in: Proc 1st Int'l Workshop on Theory and Applications of Process Visualization (TAProViz'12), LNBIP 132, pp.768–779, Springer, 2012.

[Som12] Ian Sommerville. *Software Engineering.* Pearson Studium, 2012.

[SPB05] Alexander T. Streit, Binh L. Pham, and Ross A. Brown. *Visualisation Support for Managing Large Business Process Specifications.* in: Proc Int'l Conf on Business Process Management (BPM'05), LNCS 3649, pp. 205–219, Springer, 2005.

[Spe00] Robert Spence. *Information Visualization.* Addison Wesley, ACM, 2000.

[SRW11] Sergey Smirnov, Hajo A. Reijers, and Mathias Weske. *A Semantic Approach for Business Process Model Abstraction.* in: Proc 23rd Int'l Conf on Advanced Information Systems Engineering (CAiSE'11), LNCS 6741, pp. 497–511, Springer, 2011.

[Str12] Achim Strauss. *Information Visualisation in Process-Oriented Semantic Information Networks.* Diploma Thesis, Ulm University, 2012.

[Sut96] Michael J. D. Sutton. *Document Management for the Enterprise: Principles, Techniques, and Applications.* Wiley, 1996.

[SvBE00] Bastiaan Schönhage, Alex van Ballegooij, and Anton Eliëns. *3D Gadgets for Business Process Visualization.* in: Proc 5th Symposium on the Virtual Reality Modeling Language (WEB3D-VRML-00), pp. 131–138, ACM, 2000.

[SZ10] Jörg Schäuffele and Thomas Zurawka. *Automotive Software Engineering.* Springer DE, 2010.

[TC09] Alfredo Raúl Teyseyre and Marcelo R. Campo. *An Overview of 3D Software Visualization.* in: J IEEE Trans. Vis. Comput. Graph, 1(15), pp. 87–105, 2009.

[TP66] F. Töpfer and Wolfgang Pillewizer. *The Principles of Selection, A Means of Cartographic Generalization.* in: Cartographic J, 3(1), pp. 10–16, 1966.

[vPD08] Jens von Pilgrim and Kristian Duske. *Gef3D: a framework for two-, two-and-a-half-, and three-dimensional graphical editors.* in: Proc Symposium on Software Visualization (SOFTVIS'08), pp. 95–104, ACM, 2008.

[vWN03] Jarke J. van Wijk and Wim. A. A. Nuij. *Smooth and Efficient Zooming and Panning.* in: Proc 9th annual IEEE Conf on Information Visualization (InfoVis'03), pp. 15-23, IEEE, 2003.

[vWN04] Jarke J. van Wijk and Wim A. A. Nuij. *A Model for Smooth Viewing and Navigation of Large 2D Information Spaces.* in: J Transactions on Visualization and Computer Graphics, 10(4), pp. 447–458, IEEE, 2004.

[WBR10] Stephen West, Ross A. Brown, and Jan C. Recker. *Collaborative business process modeling using 3D virtual environments.* in: Proc 16th Americas Conference on Information Systems : Sustainable IT Collaboration around the Globe, Association for Information Systems (AMCIS'10), p. 249, Association for Information Systems, 2010.

[WH06] Roel J. Wieringa and Hans J.M.G. Heerkens. *The methodological soundness of requirements engineering papers: a conceptual framework and two case studies.* in: J Requirements Engineering, 11(4), pp. 295–307, Springer, 2006.

[WLS98] Allison Woodruff, James A. Landay, and Michael Stonebraker. *Constant information density in zoomable interfaces.* in: Proc Working Conference on Advanced Visual Interfaces (AVI'98), pp. 57–65, ACM, 1998.

[WM09] Jörg Wurzer and Bela Mutschler. *Bringing innovative Semantic Technology to Practice: The iQser Approach and its Use Cases.* in: Proc 4th Int'l Workshop on Applications of Semantic Technologies (AST'09), LNI 154, pp. 3026–3040, Springer, 2009.

[WMMR05] Roel J. Wieringa, Neil Maiden, Nancy Mead, and Colette Rolland. *Requirements engineering paper classification and evaluation criteria: a proposal and a discussion.* Requirements Engineering 11(1), 102-107. (2006), Springer, 2005.

[WRH+12] Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in Software Engineering.* Springer Berlin Heidelberg, 2012.

[Wri03] Peter C. Wright. *Funology - From Usability to Enjoyment.* in: J Human-Computer Interaction, 3, Springer, 2003.

[WRMR11] Barbara Weber, Manfred Reichert, Jan Mendling, and Hajo A. Reijers. *Refactoring large process model repositories.* Computers in Industry, 62(5), pp. 467–486, Elsevier, 2011.

[WRRM08] Barbara Weber, Manfred Reichert, and Stefanie Rinderle-Ma. *Change Patterns and Change Support Features - Enhancing Flexibility in Process-Aware Information Systems.* in: J Data & Knowledge Engineering Archive, 66(3), pp. 438–466, 2008.

[WRWRM09] Barbara Weber, Manfred Reichert, Werner Wild, and Stefanie Rinderle-Ma. *Providing Integrated Life Cycle Support in Process-Aware Information Systems.* in: Int'l Journal of Cooperative Information Systems (IJCIS'09) , 18(1), pp. 115-165, 2009.

[WSWW06] Ferdinand Wagner, Ruedi Schmuki, Peter Wolstenholme, and Thomas Wagner. *Modeling Software with Finite State Machines - A Practical Approach.* Taylor & Francis, 2006.

[Wur08] Jörg Wurzer. *New approach for semantic web by automatic semantics.* in: Proc European Conf on Semantic Technology (ESTC'08), 2008.

[WW96] Kent Wittenburg and Louis Weitzman. *Qualitative Visualization of Processes: Attributed Graph Layout and Focusing Techniques.* in: Proc Symposium on Graph Drawing (GD'96), LNCS 1190, pp. 401-408, Springer, 1996.

[ZJR11] Michael Zöllner, Hans-Christian Jetter, and Harald Reiterer. *ZOIL: A Design Paradigm and Software Framework for Post-WIMP Distributed User Interfaces.* in: J on Distributed User Interfaces, pp. 87–94, HCI Series, 2011.

[ZPR+12] Stefan Zugal, Jakob Pinggera, Hajo Reijers, Manfred Reichert, and Barbara Weber. *Making the Case for Measuring Mental Effort.* in: Proc 2nd Workshop on Experiences and Empirical Studies in Software Modelling (EESSMod'12), pp. 37–42, ACM, 2012.

Acronyms

ABS	anti-lock breaking system
ANOVA	analysis of variance
BM	basis model
BPM	business process management
BPMN	Business Process Management and Notation
CM	context model
CMP	critical path method
CSS	cascading style sheets
DOM	document object model
dr	density ratio
E/E	electric/electronic
EPC	event-driven process chain
FSM	Finit State Machines
GIS	geographic information systems
HCI	Human-computer interaction
LA	Linear Algebra
NavSeq	navigation sequence
niPRO	user-adequate and intelligent process information portals
niPRO	user-adequate process information portals
NM	navigation model
NS	navigation state
NUI	natural user interface
OEM	Original Equipment Manufacturer
PAIS	process-aware information systems
PL	Predicate Logic
PMC	process model collection
PN	Petri Nets

PN	process navigation
POIL	process-oriented information logistics
POIL	process-oriented information logistics
POM	process object model
ProNaVis	Process Navigation and Visualization
SESE	single entry single exit
SIN	semantic information network
STS	State Transition Systems
WIMP	windows, icons, menus, and pointers
WS-BPEL	WS-business process execution language
XML	extensible markup language
ZUI	zoomable user interface

A Appendix

A.1 Requirements Engineering Process

```
<definitions id="sid-d94a69e5-2b9a-40bd-ba41-b82e37b7da26" ...>
  <collaboration id="sid-beaaecbb-01d1-4b6f-ae96-b842aff0702c">
      <participant id="sid-56DC4517-4DF8-42E5-A0A5-6122438FFC31"
      name="(R1) E/E Development"
      processRef="sid-77525E02-6689-43EB-8BDC-B4458A5E4B16">
      </participant>
  </collaboration>
  <process id="sid-77525E02-6689-43EB-8BDC-B4458A5E4B16"
    isClosed="false"
    isExecutable="false"
    name="(R1) E/E Development"
    processType="None">
    <extensionElements/>
    <dataObject id="sid-73d0d882..." name="(D1) Req Eng Handbook Ch. 5.1- 5.2"/>
    <dataObject id="sid-8c1224ca..." name="(D7) Tec Part of gereral spec"/>
    <dataObject id="sid-9d18b497..." name="(D9) Safety Measures"/>
    <dataObject id="sid-b48224fa..." name="(D6) Req Eng Handbook Ch. 5.5"/>
    <dataObject id="sid-e09413b7..." name="(D12) Req Eng Handbook Ch. 5.6"/>
    <dataObject id="sid-6e9d7f89..." name="(D3) Feature list"/>
    <dataObject id="sid-e7ebaf84..." name="(D8) EE General Specification"/>
    <dataObject id="sid-5ee9fef0..." name="(D11) Decision maker template"/>
    <dataObject id="sid-937f7960..." name="(D10) Change Requests"/>
    <dataObject id="sid-a1c870f2..." name="(D2) Worksop Documents"/>
    <laneSet id="sid-699b3783-8593-4e9b-a309-e22d8a23c1d2">
      <lane id="sid-183A8882..." name="(R3) Experts">...</lane>
      <lane id="sid-7DC993BB..." name="(R2) Component responsible">...</lane>
      <lane id="sid-4DD5E8FA..." name="(R4) Project responsible">...</lane>
      <lane id="sid-62D2E63A..." name="(R5) Decision Maker">
        <extensionElements>
          <signavio:signavioMetaData metaKey="bgcolor" metaValue=""/>
        </extensionElements>
        <flowNodeRef>sid-8A473318-181F-4ACD-9FFC-7061BA8AE1D9</flowNodeRef>
        <flowNodeRef>sid-7F8D1CBF-24C2-4FA0-A2A1-030A5D078B47</flowNodeRef>
        <flowNodeRef>sid-B36052F0-FEC1-43E2-B946-5DA7338D2EA6</flowNodeRef>
        <flowNodeRef>sid-CE7E26C2-CA35-4B61-97D2-57E72EDFB5FD</flowNodeRef>
        <flowNodeRef>sid-D437A677-7E07-43DB-B0BA-3375B584FD09</flowNodeRef>
      </lane>
    <startEvent id="sid-CCC98825-F192-4D31-AE7D-0CBD107A98EA"
      name="SE">...</startEvent>
    <task completionQuantity="1"
      id="sid-BC3C4DDC-F6B6-4EAC-AF6A-8C5DA76F3338"
      isForCompensation="false" name="(T1) Plan RE Workshop"
      startQuantity="1">...</task>
    <subProcess completionQuantity="1" id="sid-2FC8916D..."
```

```
    isForCompensation="false" name="(T2) Perform RE Wokshop"
    startQuantity="1" triggeredByEvent="false">...</subProcess>
    <task completionQuantity="1" id="sid-A8E3EB1C-590E-40C1-8541-37F53E5F8A2A"
    isForCompensation="false" name="(T3) Write technical part of general spec. "
    startQuantity="1">
      <extensionElements>
        <signavio:signavioMetaData metaKey="bgcolor" metaValue="#ffffcc"/>
      </extensionElements>
      <incoming>sid-1C4B2088-2BEF-45F0-AEB7-AF82190A1972</incoming>
      <outgoing>sid-8287E761-1B11-4A4B-873D-69D648917643</outgoing>
      <ioSpecification id="sid-6ce29bfb-7dd3-4088-8ede-01bab8259bcd">
        <dataInput id="sid-ff15cfac-51f5-4cbf-a5e4-3864126adae6"/>
        <dataInput id="sid-5b26bd6e-330c-4b2a-bb5f-09afe905c598"/>
        <dataOutput id="sid-3bba1abe-359f-4bba-b080-3d2a9f3e5401"/>
        <inputSet id="sid-1ab95166..." name="DefaultInputSet">
          <dataInputRefs>sid-ff15cfac-51f5-4cbf-a5e4-3864126adae6</dataInputRefs>
          <dataInputRefs>sid-5b26bd6e-330c-4b2a-bb5f-09afe905c598</dataInputRefs>
          <outputSetRefs>sid-f932f3c6-ffb7-4394-80af-61da39ecc94b</outputSetRefs>
        </inputSet>
        <outputSet id="sid-f932f3c6..." name="DefaultOutputSet">
          <dataOutputRefs>sid-3bba1abe...</dataOutputRefs>
          <inputSetRefs>sid-1ab95166-4e04-4fe6-93da-0269ea2e164d</inputSetRefs>
        </outputSet>
      </ioSpecification>
      <dataInputAssociation id="sid-4F7D6235-B2D8-4A6E-B49B-A7E6D296E8BF">
        <sourceRef>sid-6A749F2A-BF2B-4978-A18C-072A91CE813C</sourceRef>
        <targetRef>sid-ff15cfac-51f5-4cbf-a5e4-3864126adae6</targetRef>
      </dataInputAssociation>
      <dataInputAssociation id="sid-4B488B0F-E3E3-48E3-908D-7A9F5948E3FD">
        <sourceRef>sid-B9AF53EB-1A35-4234-91F3-2AA31E439307</sourceRef>
        <targetRef>sid-5b26bd6e-330c-4b2a-bb5f-09afe905c598</targetRef>
      </dataInputAssociation>
      <dataOutputAssociation id="sid-C04666E6-56C7-4424-9AFB-EFEBBC1A886C">
        <sourceRef>sid-3bba1abe-359f-4bba-b080-3d2a9f3e5401</sourceRef>
        <targetRef>sid-B3E7CCC6-02C5-4F00-AA36-9FC0364B8C78</targetRef>
      </dataOutputAssociation>
    </task>
    <task completionQuantity="1" id="sid-B92C5D58-982E-40BF-BF12-B3FAB04728BE"
    sForCompensation="false" name="(A4) Write general specification."
    startQuantity="1">...</task>
    <startEvent id="sid-CCC98825-F192-4D31-AE7D-0CBD107A98EA"
    name="SE">...</startEvent>
    <task completionQuantity="1" id="sid-1D11883F-21B9-44FF-9EA3-3190C301C73E"
    isForCompensation="false" name="(A5) Integrate component specification to
    general specfication." startQuantity="1">...</task>
    <task completionQuantity="1" id="sid-CAF40F33-571A-471B-A867-3163163E8075"
    isForCompensation="false" name="(A8) Perform FMEA analysis"
    startQuantity="1">...</task>
    <exclusiveGateway gatewayDirection="Converging"
    id="sid-8E3544A9-0688-4742-B05B-FC4157A05165" name="(G1)">
    ...</exclusiveGateway>
    <task completionQuantity="1" id="sid-8A473318-181F-4ACD-9FFC-7061BA8AE1D9"
    isForCompensation="false" name="(A6) Evaluate and give strategic direction."
    startQuantity="1">...</task>
    <exclusiveGateway gatewayDirection="Diverging"
    id="sid-7F8D1CBF-24C2-4FA0-A2A1-030A5D078B47"
    name="(G2) Change Request available?">...</exclusiveGateway>
```

```
    <task completionQuantity="1" id="sid-B36052F0-FEC1-43E2-B946-5DA7338D2EA6"
    isForCompensation="false" name="(A7) Incorporate change requests."
    startQuantity="1">...</task>
    <task completionQuantity="1" id="sid-CE7E26C2-CA35-4B61-97D2-57E72EDFB5FD"
    isForCompensation="false" name="(A9) Release general specification."
    startQuantity="1">...</task>
    <endEvent id="sid-D437A677-7E07-43DB-B0BA-3375B584FD09" name="">
    ...</endEvent>
    <sequenceFlow id="sid-5B489B98..." name="" sourceRef="sid-CCC98825..."
    targetRef="sid-BC3C4DDC..."/>
    <sequenceFlow id="sid-8287E761..." name="" sourceRef="sid-A8E3EB1C..."
    targetRef="sid-B92C5D58..."/>
    <sequenceFlow id="sid-71EA84A1..." name="" sourceRef="sid-B92C5D58..."
    targetRef="sid-8E3544A9..."/>
    <sequenceFlow id="sid-BA42AF6D..." name="" sourceRef="sid-8A473318..."
    targetRef="sid-7F8D1CBF..."/>
    ...
    <dataObjectReference dataObjectRef="sid-73d0d882..." id="sid-E8059C91..."
    name="(D1) Requirements Engineering Handbook Chapter 5.1- 5.2">
      <extensionElements>
        <signavio:signavioMetaData metaKey="bgcolor" metaValue="#ffffff"/>
      </extensionElements>
    </dataObjectReference>
    <dataObjectReference dataObjectRef="sid-8c1224ca..." id="sid-B3E7CCC6..."
    name="(D7) Technical Part of gereral specification">...</dataObjectReference>
    <dataObjectReference dataObjectRef="sid-9d18b497..." id="sid-F51A38B0..."
    name="(D9) Safety Measures">...</dataObjectReference>
    <dataObjectReference dataObjectRef="sid-b48224fa..." id="sid-B9AF53EB..."
    name="(D6) Requirements Engineering Handbook Chapter 5.5">
    ...</dataObjectReference>
    <dataObjectReference dataObjectRef="sid-e09413b7..." id="sid-BF0D50F3..."
    name="(D12) Requirements Engineering Handbook Chapter 5.6">
    ...</dataObjectReference>
  </process>
  <bpmndi:BPMNDiagram id="sid-eaa5bccf-47ca-44c7-9742-8c7b9e8abff1">
    ...
  </bpmndi:BPMNDiagram>
</definitions>
```

Listing A.1: XML file of the requirements engineering process model.

A.2 Experiment 1

A.2.1 Introduction 1-dimensional Navigation Concept

Begriffe

Prozess: Prozess ist die allgemeine Bezeichnung einer Abfolge von Prozessschritten. Diese können elementar sein oder weitere Prozessschritte enthalten (Subprozess).

Prozessschritt: Ein Prozessschritt ist eine Arbeitseinheit eines Prozesses. Er kann elementar sein oder einen Subprozess (Plus Symbol) enthalten.

Subprozess: Ein Subprozess ist der Prozess in einem durch ein Plus Symbol markierten Prozessschritt. Er enthählt weitere Prozessschritte.

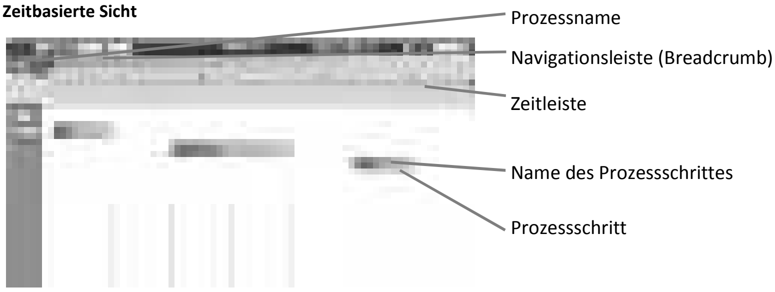
Intro

Max und seine Kollegen möchten einen Urlaub machen. Die einzelnen Prozessschritte mit Subprozessen von der Planung, über den Urlaub selbst bis zur Nachbereitung sind in einem Prozessinformationsportal dokumentiert. Um Aufgaben zu verteilen wurde zudem ein Organigramm mit zugeordneten Farben zu Rollen und Personen erstellt. Die Prozessschritte sind dann mit den entsprechenden Farben markiert.

Im Prozessinformationsportal kann in einem Prozess interaktiv navigiert werden, zudem können Prozessinformationen zu Prozessschritten dargestellt werden. Die Anwendung verwendet dabei vier unterschiedliche Sichten auf einen Prozess.

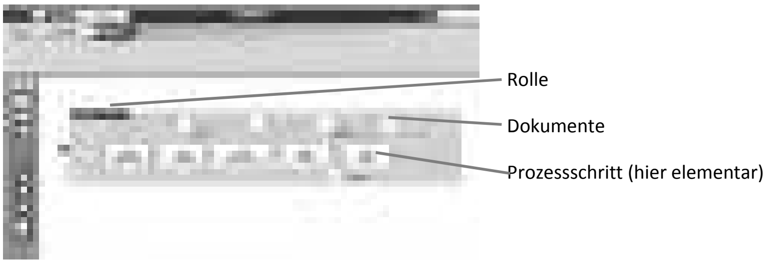
Sichten

Zeitbasierte Sicht



In dieser Sicht sind Prozessschritte als Boxen mit unterschiedlicher Länge dargestellt. Die Länge einer Box entspricht dabei der Dauer des Prozessschrittes. Je nach Einheit der Zeitleiste ist diese in Monaten oder Tagen angegeben. Die Farbe der Prozesse gibt Aufschluss über die zugeordnete Rolle. Ist ein Plus Symbol auf einem Prozessschritt zu sehen, ist ein Subprozess enthalten.

Logische Sicht

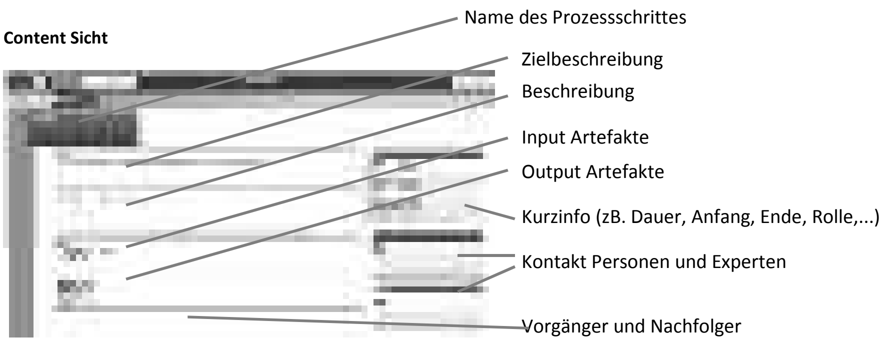


In der Logischen Sicht wird die logische Reihenfolge von Prozessschritten ähnlich wie in BPMN dargestellt (Kontrollfluss). Wenn eine Entscheidung getroffen werden muss, oder zwei Prozessschritte parallel ausgeführt werden können, kann sich der Kontrollfluss auch verzweigen.

Turtle Sicht

Die Turtle Sicht enthält eine detaillierte Beschreibung eines Prozessschrittes, sowie Informationen zu Input- und Output-Dokumenten, Vorbedingungen, Hilfsmitteln und Rollen.

Content Sicht



Die Content Sicht ist aufgebaut wie eine Art Wiki-Seite. Sie enthält alle zusätzlichen Informationen zu einem Prozessschritt (siehe Beschriftungen) und ist eher textlastig.

Sidebar:

Alle Prozessschritte der Zeitbasierten Sicht haben eine Sidebar mit Attributen zum Prozessschritt. Die Sidebar enthält zudem einen Button, um in die Content Sicht eines Prozessschrittes aufzurufen.

Button zum Öffnen der Content View

Aufruf der Content Sicht:

Klick auf den Button in der Sidebar.

Absteigen in einen Subprozess bzw. in die Turtle Sicht:

Zwei Klicks auf einen Prozessschritt. Erster Klick markiert den Prozessschritt, mit einem weiteren Klick steigt man ab. In der Logischen Sicht wird dadurch in die Turtle Sicht abgestiegen.

Ebene nach oben:

Verwendung der Navigationsleiste (Breadcrumb).

Anmerkung: Es sind nicht alle Zustände im Prototypen vorhanden. Dadurch sind beispielsweise nicht alle Turtle Sichten vorhanden.

Einstiegsaufgaben:
Ausgangszustand für jede Aufgabe ist der Prozess „Holiday“.

E1: Markiere den Prozessschritt „Planung“ mit einem Klick darauf. Verwende nun den Button „Content View“ in der Sidebar, um in die Content Sicht des Prozessschrittes zu gelangen. Gehe wieder zurück zur ersten Ebene, indem du in der Breadcrumb auf „Holiday“ klickst.

E2: Klicke zweimal auf den Prozessschritt „Planung“, um ihn zu markieren und in den Subprozess abzusteigen. Steige nun in den Prozessschritt „Terminfindung“ ab. Wechsle dann mit Hilfe der Breadcrumb auf den Prozess „Planung“ und anschließend auf den Ausgangsprozess „Holiday“.

E3: Steige wieder in den Prozessschritt „Planung“ ab. Danach in den Prozessschritt „Recherche“. Dieser Prozess wurde in der Logischen Sicht modelliert. Mache nun zwei Klicks auf den Prozessschritt „Termine zusammenstellen“, um die Turtle des Prozessschrittes aufzurufen. Verwende anschließend die Breadcrumb, um zum Ausgangsprozess „Holiday“ zu gelangen.

A.2.2 Introduction 3-dimensional Navigation Concept

Begriffe

Prozess: Prozess ist die allgemeine Bezeichnung einer Abfolge von Prozessschritten. Diese können elementar sein oder weitere Prozessschritte enthalten (Subprozess).

Prozessschritt: Ein Prozessschritt ist eine Arbeitseinheit eines Prozesses. Er kann elementar sein oder einen Subprozess (Plus Symbol) enthalten.

Subprozess: Ein Subprozess ist der Prozess in einem durch ein Plus Symbol markierten Prozessschritt. Er enthählt weitere Prozessschritte.

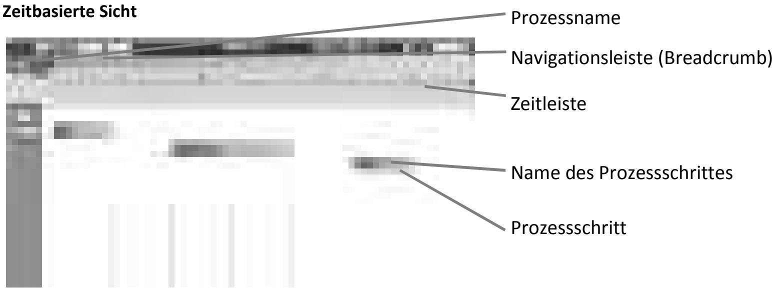
Intro

Max und seine Kollegen möchten einen Urlaub machen. Die einzelnen Prozessschritte mit Subprozessen von der Planung, über den Urlaub selbst bis zur Nachbereitung sind in einem Prozessinformationsportal dokumentiert. Um Aufgaben zu verteilen wurde zudem ein Organigramm mit zugeordneten Farben zu Rollen und Personen erstellt. Die Prozessschritte sind dann mit den entsprechenden Farben markiert.

Im Prozessinformationsportal kann in einem Prozess interaktiv navigiert werden, zudem können Prozessinformationen zu Prozessschritten dargestellt werden. Die Anwendung verwendet dabei vier unterschiedliche Sichten auf einen Prozess.

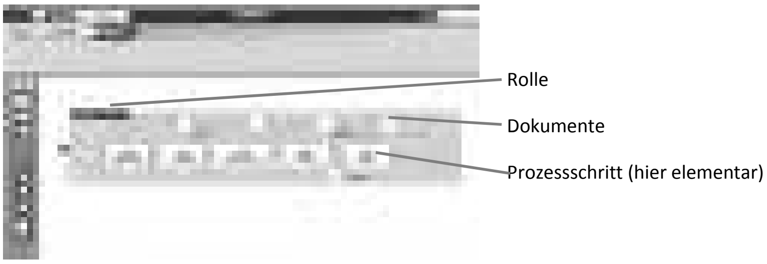
Sichten

Zeitbasierte Sicht



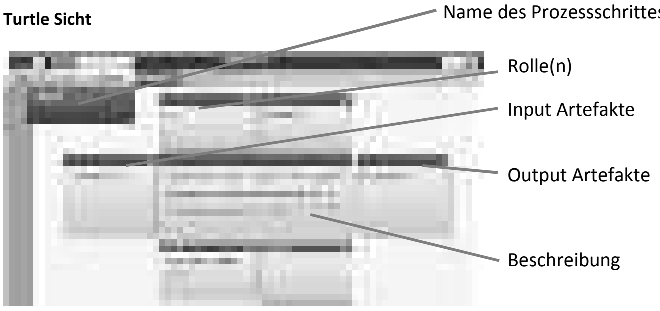
In dieser Sicht sind Prozessschritte als Boxen mit unterschiedlicher Länge dargestellt. Die Länge einer Box entspricht dabei der Dauer des Prozessschrittes. Je nach Einheit der Zeitleiste ist diese in Monaten oder Tagen angegeben. Die Farbe der Prozesse gibt Aufschluss über die zugeordnete Rolle. Ist ein Plus Symbol auf einem Prozessschritt zu sehen, ist ein Subprozess enthalten.

Logische Sicht



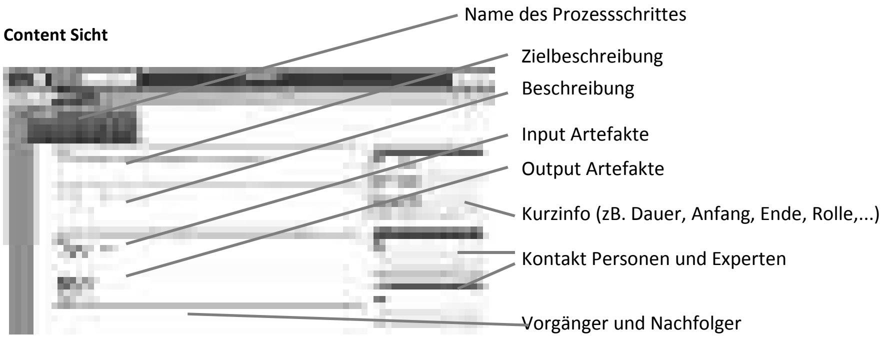
In der Logischen Sicht wird die logische Reihenfolge von Prozessschritten ähnlich wie in BPMN dargestellt (Kontrollfluss). Wenn eine Entscheidung getroffen werden muss, oder zwei Prozessschritte parallel ausgeführt werden können, kann sich der Kontrollfluss auch verzweigen.

Turtle Sicht



Die Turtle Sicht enthält eine detaillierte Beschreibung eines Prozessschrittes, sowie Informationen zu Input- und Output-Dokumenten, Vorbedingungen, Hilfsmitteln und Rollen.

Content Sicht



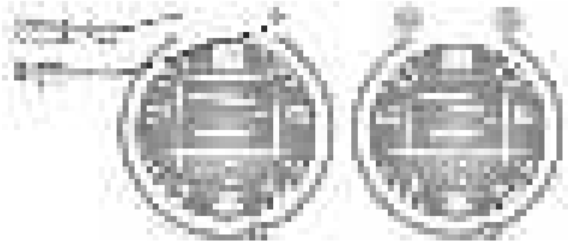
Die Content Sicht ist aufgebaut wie eine Art Wiki-Seite. Sie enthält alle zusätzlichen Informationen zu einem Prozessschritt (siehe Beschriftungen) und ist eher textlastig.

Navigationselement und Navigationsdimensionen:

Geografische Dimension:

Erlaubt das hinein- und herauszoomen. (selektiv möglich)

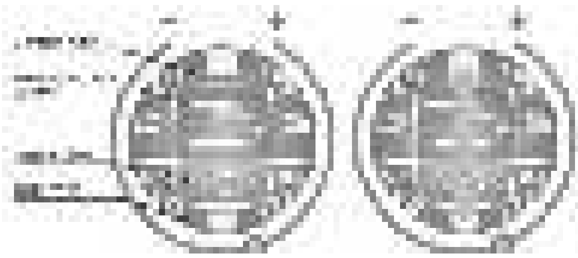
➔ Im Prototyp nur mit Klick auf Plus oder Minus Icon möglich. (Später Zoom mit Mausrad auf die Mausposition)



View Dimension:

Ermöglicht das Ändern der Sicht zu Zeitbasierter-, Logischer-, Turtle- oder Content Sicht eines Prozesses bzw. Prozessschrittes. (selektiv möglich für Turtle- und Content View)

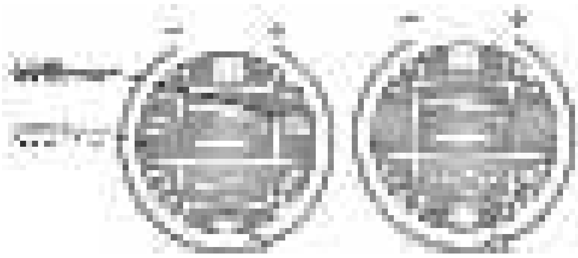
➔ Verwendung der vertikalen Icons in der Mitte des Navigationselements. Im Prototyp per Klick auf das entsprechende Icon. Die Kugel dreht sich dann um die horizontale Achse.



Semantische Dimension:

Ermöglicht das Zu- und Abschalten von Subprozessen in der aktuellen Sicht.

➔ Vertikale Icons rechts und links neben den Sichten-Icons. Im Prototyp nur rechts und links der aktuellen Sicht.



Absteigen in einen Subprozess:
(Kombination von Semantischer und Geografischer Navigationsdimension)

1.Möglk.:
Selektives Heranzoomen an einen Prozessschritt und Zuschalten der Subprozesse. Ein Prozessschritt wird mit einem Klick markiert, dann wird mit dem Plus Icon herangezoomt und anschließend werden Subprozesse zugeschaltet.

2.Möglk.:
Zwei Klicks auf einen Prozessschritt. Erster Klick markiert den Prozessschritt, mit einem weiteren Klick steigt man ab.

Ebene nach oben:

Herauszoomen oder Verwendung der Navigationsleiste (Breadcrumb).

Anmerkung: Es sind nicht alle Zustände im Prototypen vorhanden und damit sind beispielsweise nicht immer alle Sichten aufrufbar.

Einstiegsaufgaben:
Ausgangszustand für jede Aufgabe ist der Prozess „Holiday“ in der zeitbasierten Sicht.

E1: Verwende mehrmals das Plus-Icon am Navigationselement, um auf den Prozessschritt „Planung“ zu zoomen. (Simuliert das Zoomen auf die Position des Mauszeigers mit Strg + Mausrad. Dabei zeigt der graue Rahmen jeweils zuerst an, wo im nächsten Schritt hingezoomt wird.)
Zoome anschließend mit dem Minus-Icon wieder vollständig heraus.

E2: Verwende das Icon rechts neben dem Icon der aktuellen Sicht, um Subprozesse zu allen sichtbaren Prozessschritten zuzuschalten. Schalte die Subprozesse anschließend wieder ab.

E3: Mache zwei Klicks auf den Prozessschritt „Planung“, um in den Subprozess abzusteigen. Verwende anschließend mehrmals das Minus-Icon oder die Breadcrumb um wieder den übergeordneten Prozess „Holiday“ vollständig darzustellen. Schalte die Subprozesse ab.

E4: Ändere nun mit Hilfe des Navigationselements die Sicht auf den Prozess. Klicke dazu beispielsweise auf das Icon der Logischen Sicht. Wechsle dann auf die Content Sicht und verwende das Plus Icon zum Zoomen. Zoome anschließend wieder heraus und wechsle auf die Zeitbasierte Sicht.

E5: Selektiere den Prozessschritt „Planung“ und wechsle auf die Turtle Sicht. Durch das selektive Wechseln der Sicht wird automatisch an die Turtle herangezoomt. Zoome heraus und wechsle wieder zur Zeitbasierten Sicht.

E6: Schalte Subprozesse zu und markiere den Prozessschritt „Planung“. Mache noch einen Klick auf den Prozessschritt, um in den Subprozess abzusteigen. Schalte dann noch einmal Subprozesse zu. Verwende nun die Breadcrumb, um zum Ausgangszustand im Prozess „Holiday“ in der Zeitbasierten Sicht zu gelangen.

E7: Teste selbst.

A.2.3 Questionnaire

Gruppe C

Im Folgenden werden zunächst Fragen zu Ihrer Person gestellt. Danach folgen Aufgaben, die parallel im Klickprototyp bearbeitet werden sollen. Anschließend gibt es einen kleinen Feedback-Fragebogen.

Autor

Markus Hipp, Janine Barner

Code

Geben Sie bitte zunächst den vorgelegten Code an. Dieser wird dazu verwendet der Umfrage die Aufnahme zuzuordnen.

Fragen zur Person

Sie sind...

- männlich
- weiblich

Fragen zur Person

Wie alt sind Sie?

Fragen zur Person

Haben Sie bereits Erfahrung mit Prozessmodellierung?

- ja
- nein

Fragen zur Person

Wie schätzen Sie Ihre Erfahrung mit Prozessmodellen und Prozessmodellierung ein?

sehr gut ○ ○ ○ ○ ○ sehr schlecht

Fragen zur Person

Wie gut kennen Sie sich mit der Notation BPMN aus?

sehr gut ○	○	○	○	○ sehr schlecht	○ Notation ist mir nicht bekannt.

Aufgabe 1

Mila ist eine der beiden Teamleiter und möchte wissen, in welchen der Prozesse „Planung", „Urlaub" und „Nachbereitung" sich Prozessschritte der Teamleiter befinden. Diese Prozessschritte sind in der Zeitbasierten Sicht orange eingefärbt.

1.a)

In welchen der folgenden Prozesse liegen Prozessschritte der Teamleitung? (Die Rollenfarbe der Teamleiter ist orange) Finde einen möglichst effizienten Weg zur Lösung der Aufgabe.

Mehrfachantwort möglich

- [ ] "Planung"
- [ ] "Urlaub"
- [ ] "Nachbereitung"

Aufgabe 2

Auch Max ist Teamleiter und übernimmt die „Recherche“ in der „Planung“ des Urlaubs. Der Prozessschritt „Recherche“ ist aufgeteilt in einzelne Teilaufgaben, von denen Max bereits die ersten drei Aufgaben erledigt hat. Diese waren das „Online recherchieren“, das „Preise verlgeichen“ und das „Termine zusammenstellen“. Er muss nun die Aufgabe „Doodle erstellen“ erledigen.

2.a)

Navigieren Sie zum Prozessschritt "Doodle erstellen". (In Planung -> in Recherche)

2.b)

Wo kann er die genaue Beschreibung der Aufgabe „Doodle erstellen“ nachlesen?

2.c)

Welche Input Dokumente benötigt er für die Aufgabe "Doodle erstellen"?

2.d)

Welche Outputs müssen nach der Aufgabe vorliegen?

2.e)

Aus welchem Prozessschritt stammen die Input Dokumente? Welche Alternativen gibt es, um dies zu prüfen?

2.f)

Welche Outputs von "Doodle erstellen" werden weiter verwendet und in welchem Prozessschritt werden diese weiter verwendet?

Aufgabe 3

Moritz ist eines der Teammitglieder und muss im Prozess „Planung“ einen Teil der „Terminfindung“ übernehmen. Vor dem Beginn der Aufgabe möchte er sich einen groben Überblick über die zeitliche Einteilung und Überschneidungen seiner Aufgaben mit anderen verschaffen.

3.a)

Steigen Sie in den Subprozess von „Planung“ ab.

3.b)

Welche Prozessschritte überschneiden sich im Prozess „Planung“ und um wieviele Tage überschneiden sie sich?

3.c)

Wie kann er vergleichen, welche in Terminfindung und Buchen enthaltenen Subprozesse sich überschneiden? Beschreibe dein Vorgehen. Gibt es verschiedene Möglichkeiten?

3.d)

Welche Subprozesse überschneiden sich?

3.e)

Um wieviele Tage überschneiden sich die Subprozessschritte?

Zusatzaufgabe 1

Um sich einen Überblick über Prozessinformationen zum Prozess "Planung" zu verschaffen kann die Content View von "Planung" betrachtet werden.

Z1 a)

Markieren Sie den Prozessschritt "Planung" und wechseln Sie in dessen Content Sicht.

Z1 b)

Welche Dokumente sind Output des Prozesses "Planung"?

Zusatzaufgabe 2

Der Dokumentenfluss kann mit Hilfe der Pfeile in den Input und Output Bereichen der Turtle direkt verfolgt werden. Ein ausgegrauter Pfeil bedeutet dabei, dass das Dokument nicht weiter verwendet wird.

Z2 a)

Navigieren Sie in die Turtle von "Termine zusammenstellen" im Prozess "Recherche".

Z2 b)

Verwenden Sie die Pfeile, um das Output-Dokument zum nächsten Prozessschritt zu verfolgen. Was ist die Prämisse(Premises) in diesem Schritt?

Fragebogen

Es folgt ein kleiner Feedback-Fragebogen. Die Aufnahme kann nun beendet werden.

Fragobogen

Wie sehr stimmen Sie diesen Aussagen zu?

	Trifft zu	Trifft eher zu	Weder noch	Trifft eher nicht zu	Trifft nicht zu
Ich konnte die Aufgaben mit Hilfe des Navigationskonzepts **schnell** lösen.					
Die Navigation hat **Spaß** gemacht.					
Ich konnte mir während der Navigation immer eine **Übersicht** über die relevanten Prozessschritte/Subprozesse verschaffen.					
Die **Breadcrumb** ist für die Orientierung im Prozess **wichtig**.					
Die **Verfolgung von Dokumenten** in der Turtle Sicht ist **hilfreich**.					

Navigation

Die Navigation zu Subprozessen und Prozessinformationen ist...

	Trifft zu	Trifft eher zu	Weder noch	Trifft eher nicht zu	Trifft nicht zu
interessant					
anregend					
nachvollziehbar					
leicht erlernbar					
verständlich					
einfach					
intuitiv					

Geografisches Zoomen

Ich finde geografisches Zoomen in der Prozesswelt... (Gehen Sie davon aus, dass geografisches Zoomen auch mit Mausrad möglich ist)

	Trifft zu	Trifft eher zu	Weder noch	Trifft eher nicht zu	Trifft nicht zu
hilfreich					
wichtig					
einfach					
intuitiv					
einfach zu erlernen					

Semantisches Zoomen

Ich finde semantisches Zoomen (Zuschalten von Subprozessen) in der Prozesswelt...

	Trifft zu	Trifft eher zu	Weder noch	Trifft eher nicht zu	Trifft nicht zu
hilfreich					
wichtig					
einfach					
intuitiv					
einfach zu erlernen					

Sichten

Die verschiedenen Sichten auf einen Prozess sind ...

	Trifft zu	Trifft eher zu	Weder noch	Trifft eher nicht zu	Trifft nicht zu
hilfreich					
wichtig					
intuitiv verwendbar					
einfach zu erlernen					

Navigationselement

Das Navigationselement ist ...

	Trifft zu	Trifft eher zu	Weder noch	Trifft eher nicht zu	Trifft nicht zu
intuitiv bedienbar					
einfach erlernbar					
ästhetisch					
interessant					
anregend					

Fragebogen

Möchten Sie uns noch etwas mitteilen?

Gruppe C

Autor
Markus Hipp, Janine Barner

A.2.4 Experiment Results

Note that the used statistic tool SPSS[1] inverts the Likert scale. In turn to Chapter 10, tables presented in the appendix use 1 for *I totally agree* and 5 for *I totally disagree*. This does not affect the results of the experiment.

[1]IBM SPSS: http://www-01.ibm.com/software/analytics/spss/

Descriptives

		N	Mean	Std Dev	Std Err	Mean Bound	Mean Upper Bound	Min	Max
The geographic dimension is helpful.	Group A	9	1,22	,441	,147	,88	1,56	1	2
	Group B	9	1,11	,333	,111	,85	1,37	1	2
	Total	18	1,17	,383	,090	,98	1,36	1	2
The geographic dimension is important.	Group A	9	1,56	,726	,242	1,00	2,11	1	3
	Group B	9	1,67	,707	,236	1,12	2,21	1	3
	Total	18	1,61	,698	,164	1,26	1,96	1	3
The geographic dimension is easy.	Group A	9	1,67	,500	,167	1,28	2,05	1	2
	Group B	9	1,22	,441	,147	,88	1,56	1	2
	Total	18	1,44	,511	,121	1,19	1,70	1	2
The geographic dimension is intuitive.	Group A	9	1,67	,500	,167	1,28	2,05	1	2
	Group B	9	1,22	,667	,222	,71	1,73	1	3
	Total	18	1,44	,616	,145	1,14	1,75	1	3
The geographic dimension is easy to learn.	Group A	9	1,67	,500	,167	1,28	2,05	1	2
	Group B	9	1,33	,500	,167	,95	1,72	1	2
	Total	18	1,50	,514	,121	1,24	1,76	1	2
The semantic dimension is helpful.	Group A	9	1,44	,527	,176	1,04	1,85	1	2
	Group B	9	1,11	,333	,111	,85	1,37	1	2
	Total	18	1,28	,461	,109	1,05	1,51	1	2
The semantic dimension is important.	Group A	9	1,56	,527	,176	1,15	1,96	1	2
	Group B	9	1,44	,527	,176	1,04	1,85	1	2
	Total	18	1,50	,514	,121	1,24	1,76	1	2
The semantic dimension is easy.	Group A	9	2,11	,928	,309	1,40	2,82	1	4
	Group B	9	1,56	,726	,242	1,00	2,11	1	3
	Total	18	1,83	,857	,202	1,41	2,26	1	4
The semantic dimension is intuitive.	Group A	9	2,22	,833	,278	1,58	2,86	1	3
	Group B	9	1,56	,527	,176	1,15	1,96	1	2
	Total	18	1,89	,758	,179	1,51	2,27	1	3
The semantic dimension is easy to learn.	Group A	9	1,78	,972	,324	1,03	2,52	1	4
	Group B	9	1,78	,833	,278	1,14	2,42	1	3
	Total	18	1,78	,878	,207	1,34	2,21	1	4
The visualization dimension is helpful.	Group A	9	1,44	,527	,176	1,04	1,85	1	2
	Group B	9	1,44	,726	,242	,89	2,00	1	3
	Total	18	1,44	,616	,145	1,14	1,75	1	3
The visualization dimension is important.	Group A	9	1,44	,527	,176	1,04	1,85	1	2
	Group B	9	1,56	,726	,242	1,00	2,11	1	3
	Total	18	1,50	,618	,146	1,19	1,81	1	3
The visualization dimension is intuitive.	Group A	9	2,22	,972	,324	1,48	2,97	1	4
	Group B	9	1,67	,500	,167	1,28	2,05	1	2
	Total	18	1,94	,802	,189	1,55	2,34	1	4
The visualization dimension is easy to learn.	Group A	9	2,11	,601	,200	1,65	2,57	1	3
	Group B	9	2,00	,866	,289	1,33	2,67	1	4
	Total	18	2,06	,725	,171	1,69	2,42	1	4

Figure A.1: Descriptives of variables concerning the navigation dimensions.

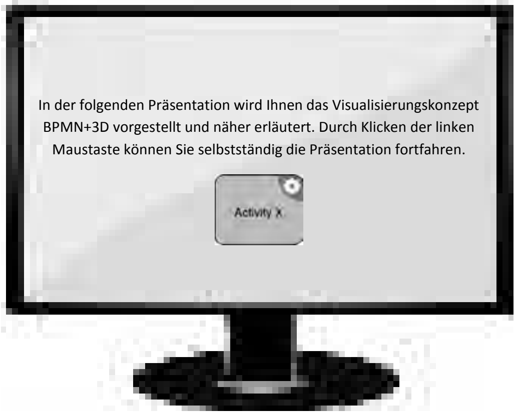
A.3 Experiment 2

Note that the following questionnaire exemplarily shows the questionnaire dealing with the BPMN3D concept. Questionnaires dealing with the Bubble and the control concept have the same structure and contents and are thereby not explicitly presented in this appendix.

A.3.1 Questionnaire

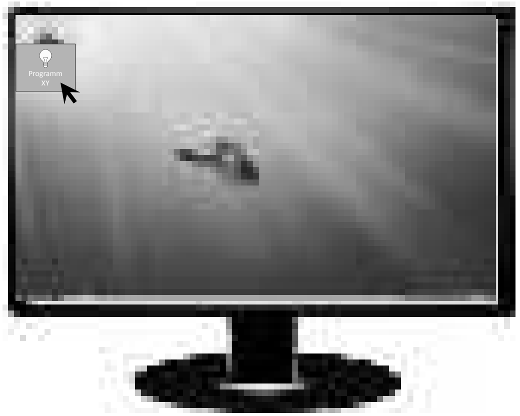
Der Experimentablauf

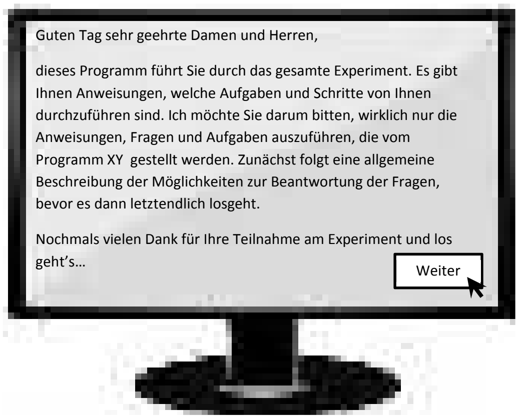
Einführung



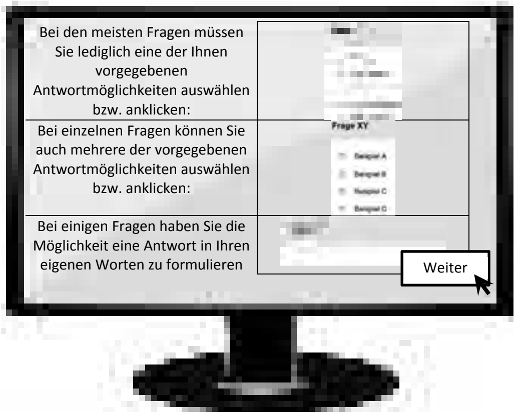
1

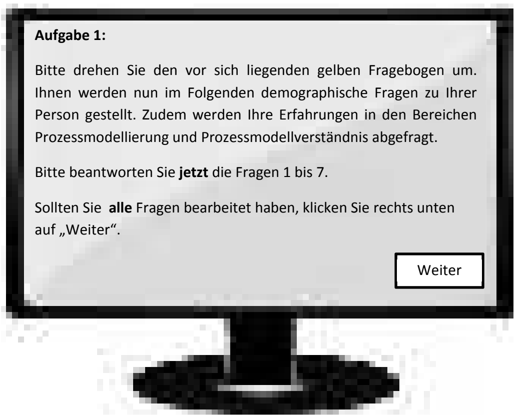
Themenblock I



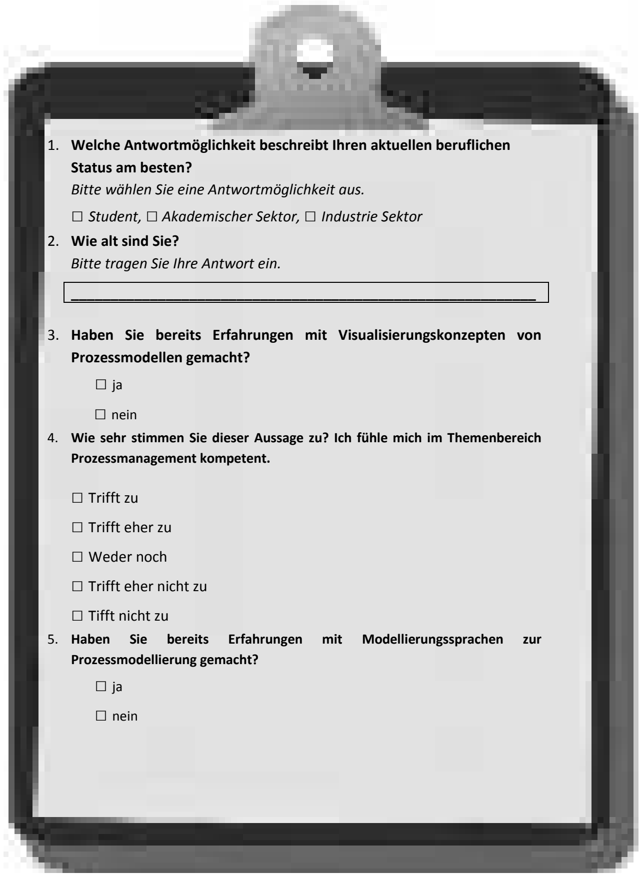


2



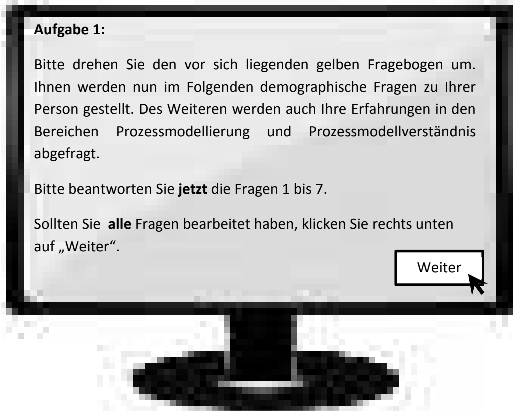


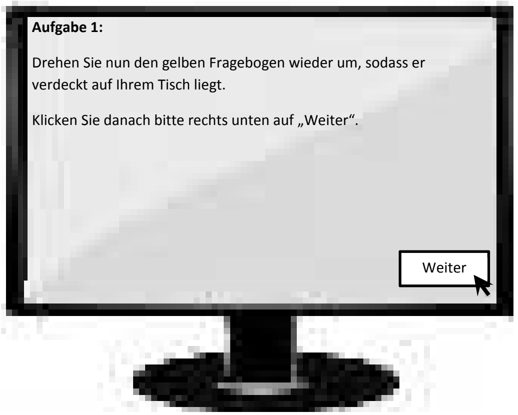
3



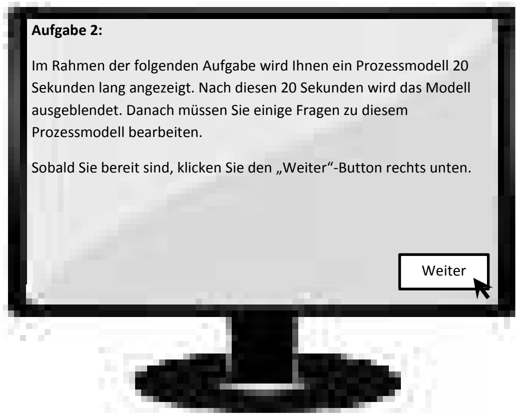
1. **Welche Antwortmöglichkeit beschreibt Ihren aktuellen beruflichen Status am besten?**
   Bitte wählen Sie eine Antwortmöglichkeit aus.
   ☐ *Student,* ☐ *Akademischer Sektor,* ☐ *Industrie Sektor*
2. **Wie alt sind Sie?**
   Bitte tragen Sie Ihre Antwort ein.
   ____
3. **Haben Sie bereits Erfahrungen mit Visualisierungskonzepten von Prozessmodellen gemacht?**
   ☐ ja
   ☐ nein
4. **Wie sehr stimmen Sie dieser Aussage zu? Ich fühle mich im Themenbereich Prozessmanagement kompetent.**
   ☐ Trifft zu
   ☐ Trifft eher zu
   ☐ Weder noch
   ☐ Trifft eher nicht zu
   ☐ Tifft nicht zu
5. **Haben Sie bereits Erfahrungen mit Modellierungssprachen zur Prozessmodellierung gemacht?**
   ☐ ja
   ☐ nein

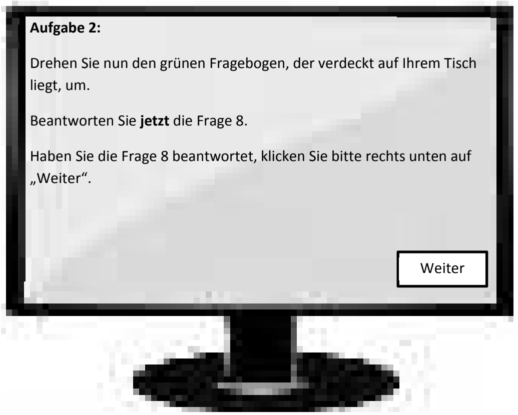
4

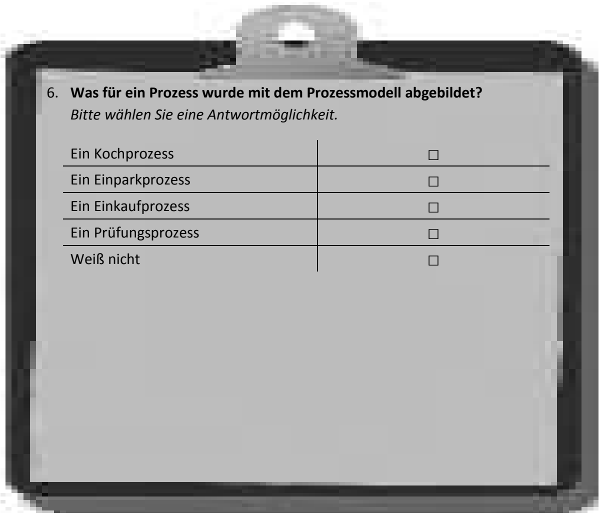




Themenblock II



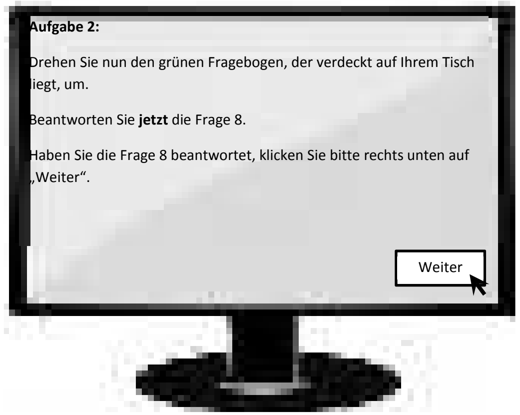


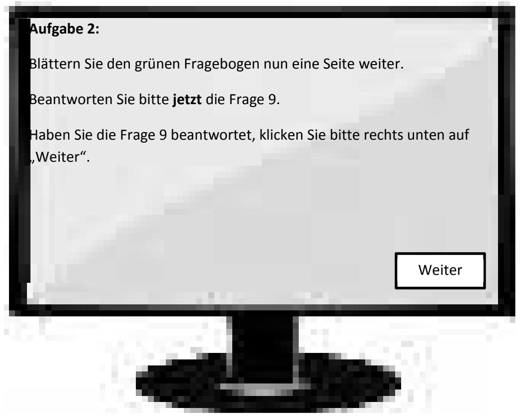


6. **Was für ein Prozess wurde mit dem Prozessmodell abgebildet?**
Bitte wählen Sie eine Antwortmöglichkeit.

Ein Kochprozess	☐
Ein Einparkprozess	☐
Ein Einkaufprozess	☐
Ein Prüfungsprozess	☐
Weiß nicht	☐

7





8

7. **Wie viele Aktivitäten waren im Prozessmodell abgebildet?**
Bitte wählen Sie eine Antwortmöglichkeit.

1-3	☐
4-7	☐
8-11	☐
12-15	☐
Weiß nicht	☐

Aufgabe 2:

Blättern Sie den grünen Fragebogen nun eine Seite weiter.

Beantworten Sie bitte **jetzt** die Frage 9.

Haben Sie die Frage 9 beantwortet, klicken Sie bitte rechts unten auf „Weiter“.

Weiter

Aufgabe 2:

Blättern Sie den grünen Fragebogen nun eine Seite weiter.

Beantworten Sie bitte **jetzt** die Frage 10.

Haben Sie die Frage 10 beantwortet, klicken Sie bitte rechts unten auf „Weiter“.

Weiter

8. **Wie viele Ereignisse konnten den Prozess starten?**
Bitte wählen Sie eine Antwortmöglichkeit.

1	☐
2	☐
3	☐
Weiß nicht	☐

Aufgabe 2:

Blättern Sie den grünen Fragebogen nun eine Seite weiter.

Beantworten Sie bitte **jetzt** die Frage 10.

Haben Sie die Frage 10 beantwortet, klicken Sie bitte rechts unten auf „Weiter".

Weiter

Aufgabe 2:

Blättern Sie den grünen Fragebogen nun eine Seite weiter.

Beantworten Sie bitte **jetzt** die Frage 11.

Haben Sie die Frage 11 beantwortet, klicken Sie bitte rechts unten auf „Weiter".

Weiter

11

9. **Wie viele Ereignisse beendeten den Prozess?**
Bitte wählen Sie eine Antwortmöglichkeit.

1	☐
2	☐
3	☐
Es gab kein Endereignis	☐
Weiß nicht	☐

Aufgabe 2:

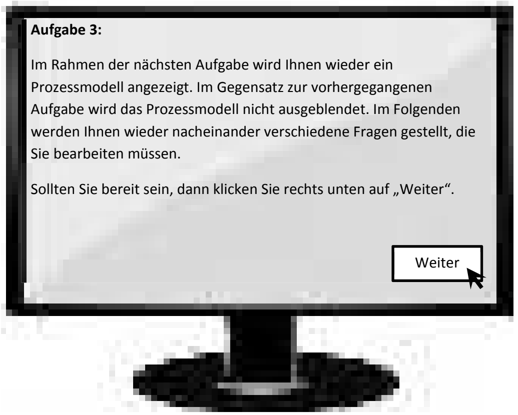
Blättern Sie den grünen Fragebogen nun eine Seite weiter.

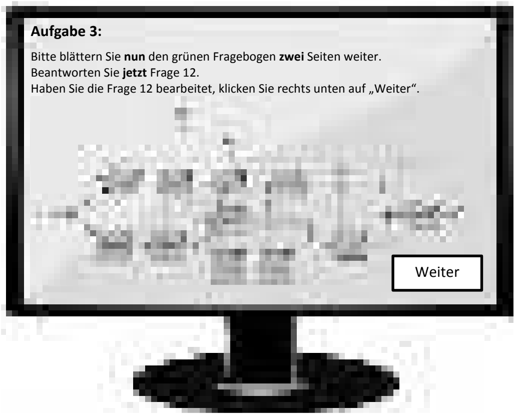
Beantworten Sie bitte **jetzt** die Frage 11.

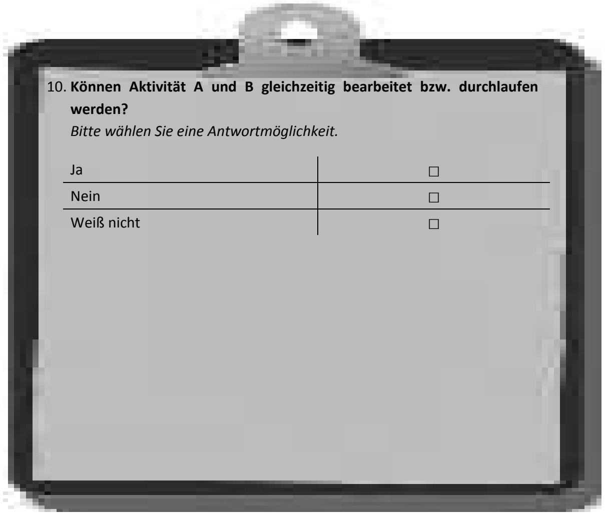
Haben Sie die Frage 11 beantwortet, klicken Sie bitte rechts unten auf „Weiter".

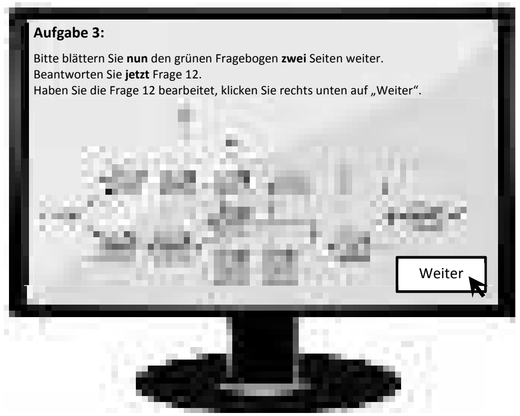
Weiter

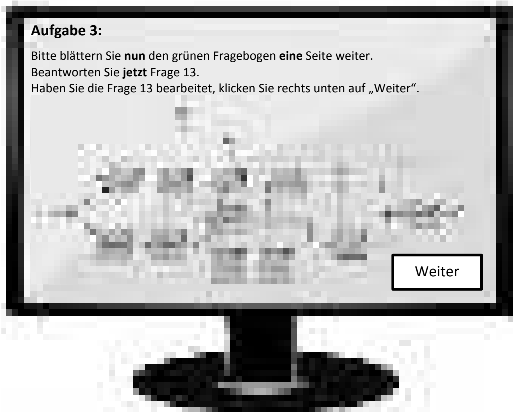
12

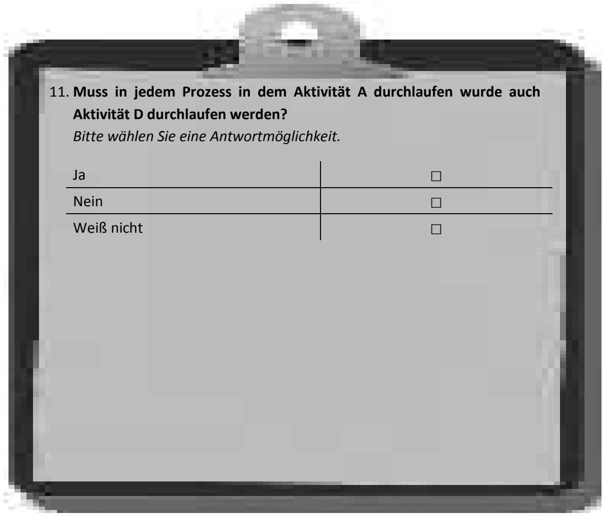

Aufgabe 3:
Im Rahmen der nächsten Aufgabe wird Ihnen wieder ein Prozessmodell angezeigt. Im Gegensatz zur vorhergegangenen Aufgabe wird das Prozessmodell nicht ausgeblendet. Im Folgenden werden Ihnen wieder nacheinander verschiedene Fragen gestellt, die Sie bearbeiten müssen.
Sollten Sie bereit sein, dann klicken Sie rechts unten auf „Weiter".
Weiter


Aufgabe 3:
Bitte blättern Sie **nun** den grünen Fragebogen **zwei** Seiten weiter.
Beantworten Sie **jetzt** Frage 12.
Haben Sie die Frage 12 bearbeitet, klicken Sie rechts unten auf „Weiter".
Weiter

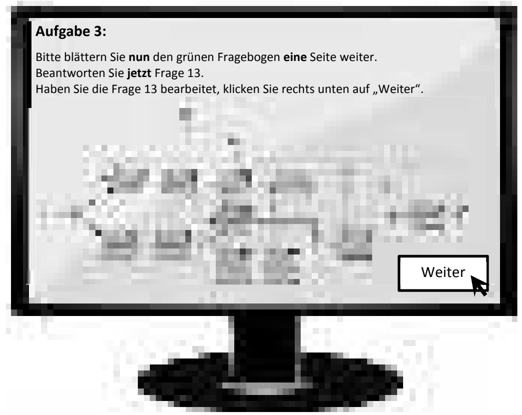

10. **Können Aktivität A und B gleichzeitig bearbeitet bzw. durchlaufen werden?**
Bitte wählen Sie eine Antwortmöglichkeit.
Ja ☐
Nein ☐
Weiß nicht ☐


Aufgabe 3:
Bitte blättern Sie **nun** den grünen Fragebogen **zwei** Seiten weiter.
Beantworten Sie **jetzt** Frage 12.
Haben Sie die Frage 12 bearbeitet, klicken Sie rechts unten auf „Weiter".
Weiter


Aufgabe 3:
Bitte blättern Sie **nun** den grünen Fragebogen **eine** Seite weiter.
Beantworten Sie **jetzt** Frage 13.
Haben Sie die Frage 13 bearbeitet, klicken Sie rechts unten auf „Weiter".
Weiter


11. **Muss in jedem Prozess in dem Aktivität A durchlaufen wurde auch Aktivität D durchlaufen werden?**
Bitte wählen Sie eine Antwortmöglichkeit.
Ja ☐
Nein ☐
Weiß nicht ☐

15

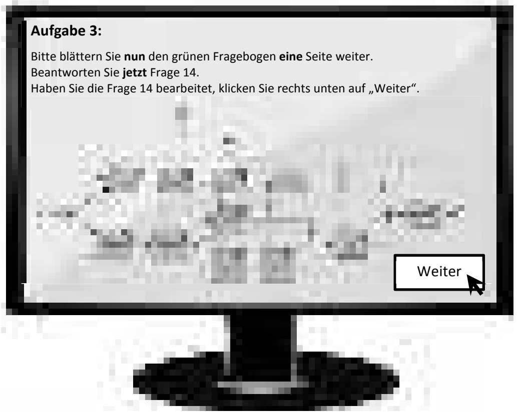

Aufgabe 3:
Bitte blättern Sie **nun** den grünen Fragebogen **eine** Seite weiter.
Beantworten Sie **jetzt** Frage 13.
Haben Sie die Frage 13 bearbeitet, klicken Sie rechts unten auf „Weiter".
Weiter

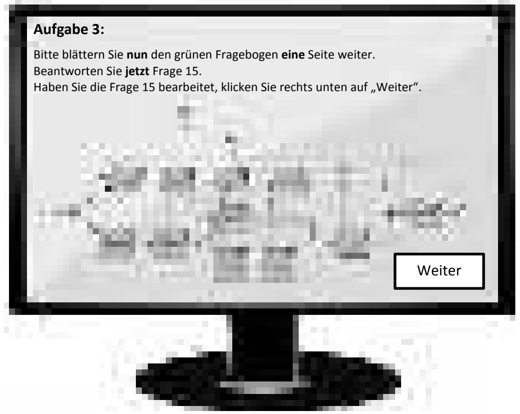

Aufgabe 3:
Bitte blättern Sie **nun** den grünen Fragebogen **eine** Seite weiter.
Beantworten Sie **jetzt** Frage 14.
Haben Sie die Frage 14 bearbeitet, klicken Sie rechts unten auf „Weiter".
Weiter

16

12. **Müssen Aktivität A, B und E in einem Prozess durchlaufen werden?**
Bitte wählen Sie eine Antwortmöglichkeit.

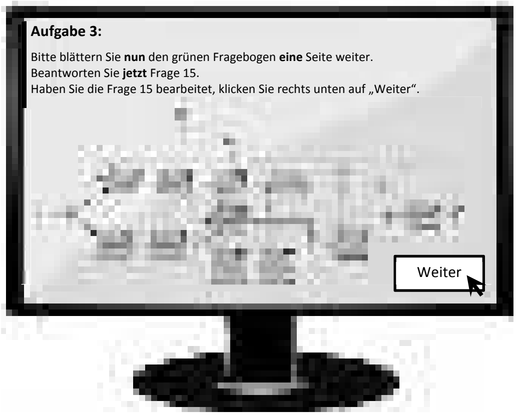
Ja	☐
Nein	☐
Weiß nicht	☐





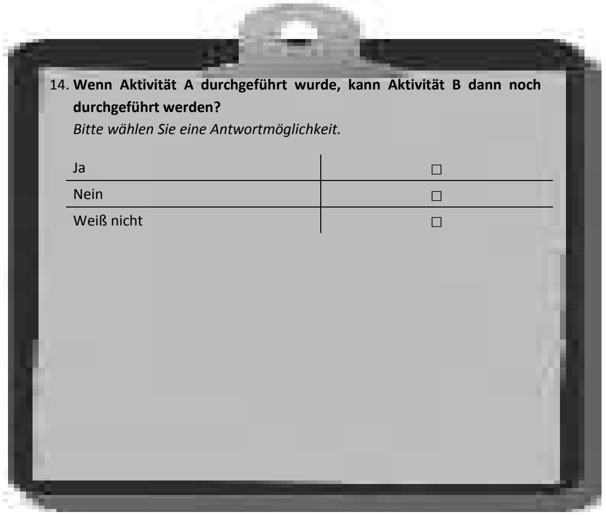
13. **Welche Aktivität muss nach Aktivität A durchgeführt werden?**
Bitte wählen Sie eine Antwortmöglichkeit.

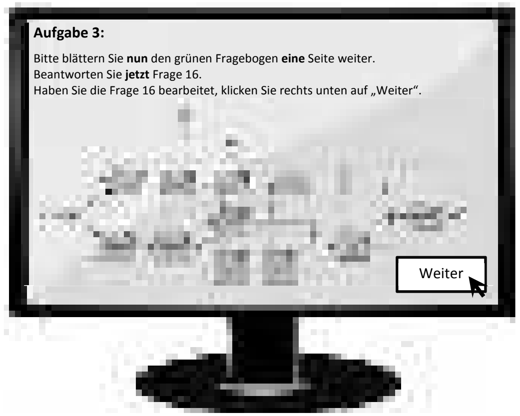
Aktivität B	☐
Aktivität D	☐
Aktivität F	☐
Aktivität G	☐
Der Prozess ist beendet	☐
Weiß nicht	☐


Aufgabe 3:
Bitte blättern Sie **nun** den grünen Fragebogen **eine** Seite weiter.
Beantworten Sie **jetzt** Frage 15.
Haben Sie die Frage 15 bearbeitet, klicken Sie rechts unten auf „Weiter".
Weiter

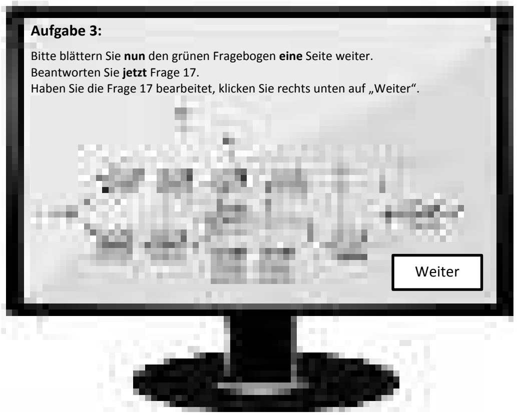

Aufgabe 3:
Bitte blättern Sie **nun** den grünen Fragebogen **eine** Seite weiter.
Beantworten Sie **jetzt** Frage 16.
Haben Sie die Frage 16 bearbeitet, klicken Sie rechts unten auf „Weiter".
Weiter

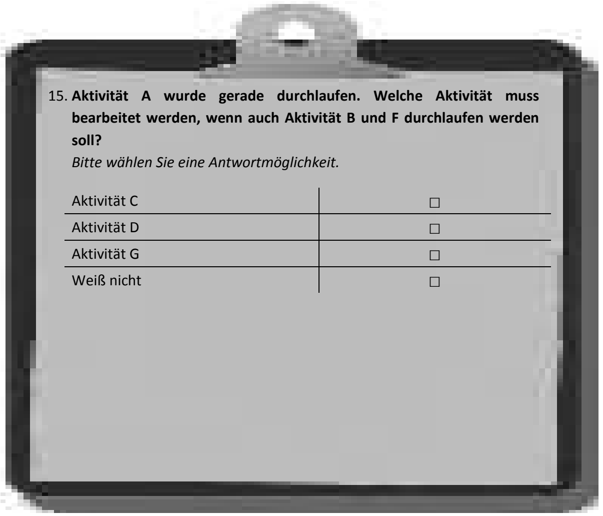
19


14. **Wenn Aktivität A durchgeführt wurde, kann Aktivität B dann noch durchgeführt werden?**
Bitte wählen Sie eine Antwortmöglichkeit.
Ja ☐
Nein ☐
Weiß nicht ☐


Aufgabe 3:
Bitte blättern Sie **nun** den grünen Fragebogen **eine** Seite weiter.
Beantworten Sie **jetzt** Frage 16.
Haben Sie die Frage 16 bearbeitet, klicken Sie rechts unten auf „Weiter".
Weiter

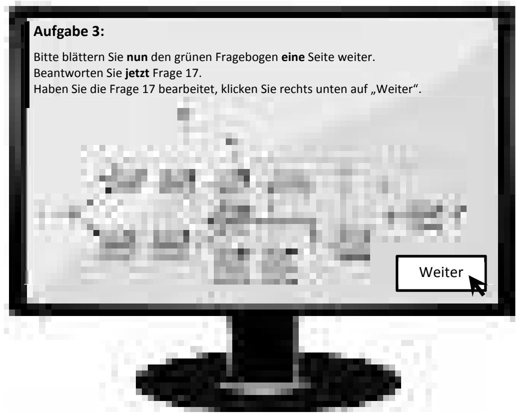
20

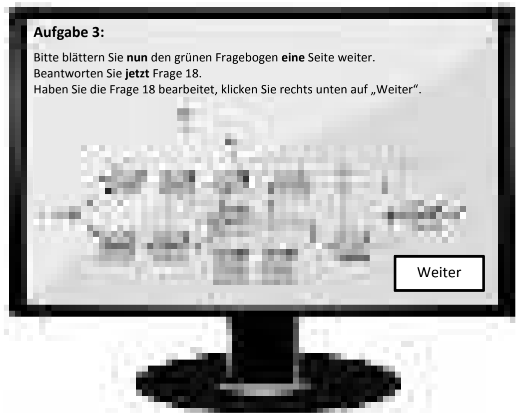




15. **Aktivität A wurde gerade durchlaufen. Welche Aktivität muss bearbeitet werden, wenn auch Aktivität B und F durchlaufen werden soll?**
Bitte wählen Sie eine Antwortmöglichkeit.

Aktivität C	☐
Aktivität D	☐
Aktivität G	☐
Weiß nicht	☐





16. **Wenn Aktivität A durchlaufen wurde, musste Aktivität B Vorher durchlaufen werden?**
Bitte wählen Sie eine Antwortmöglichkeit.

Ja	☐
Nein	☐
Weiß nicht	☐

Aufgabe 3:

Bitte blättern Sie **nun** den grünen Fragebogen **eine** Seite weiter.
Beantworten Sie **jetzt** Frage 18.
Haben Sie die Frage 18 bearbeitet, klicken Sie rechts unten auf „Weiter".

Weiter

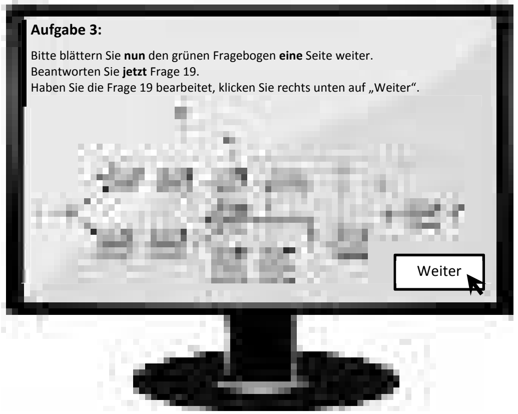
Aufgabe 3:

Bitte blättern Sie **nun** den grünen Fragebogen **eine** Seite weiter.
Beantworten Sie **jetzt** Frage 19.
Haben Sie die Frage 19 bearbeitet, klicken Sie rechts unten auf „Weiter".

Weiter

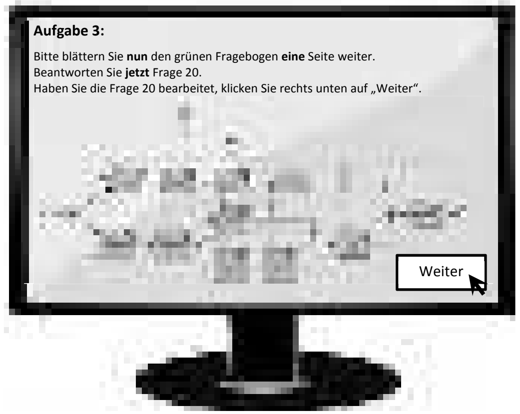
17. **Wenn Aktivität A durchlaufen wird, ist Aktivität B schon beendet?**
Bitte wählen Sie eine Antwortmöglichkeit.

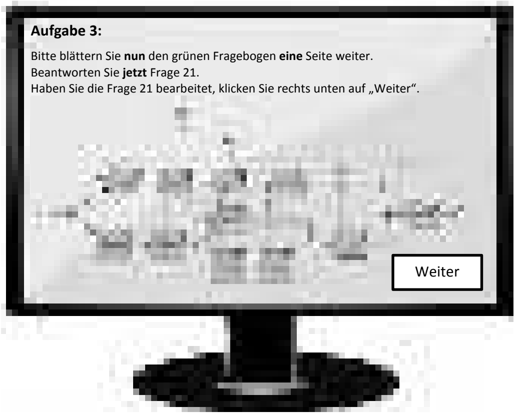
Ja	☐
Nein	☐
Weiß nicht	☐



18. **Kann Aktivität C ausgeführt werden, nachdem Aktivität D schon durchlaufen wurde?**
Bitte wählen Sie eine Antwortmöglichkeit.

Ja	☐
Nein	☐
Weiß nicht	☐

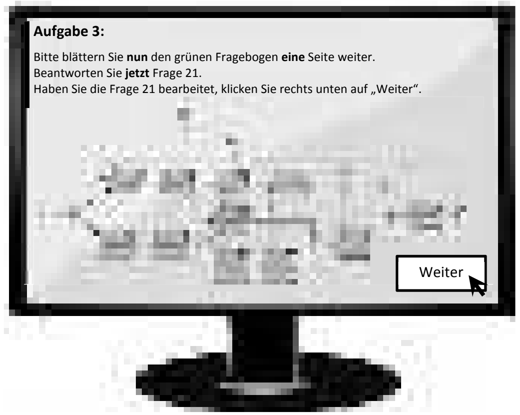


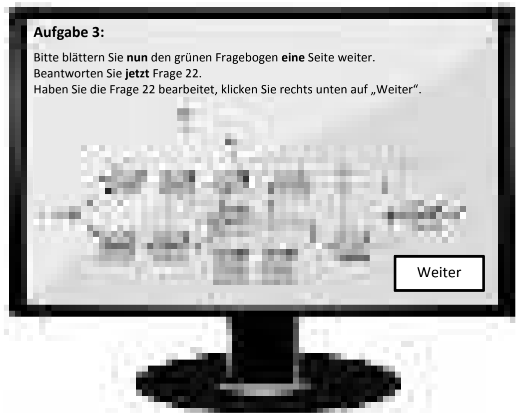


19. **Nach durchlaufen von Aktivität C steht eine Entscheidung an. Mit welchen Aktivitäten kann der Prozess weitergeführt werden?**
Bitte wählen Sie eine Antwortmöglichkeit.

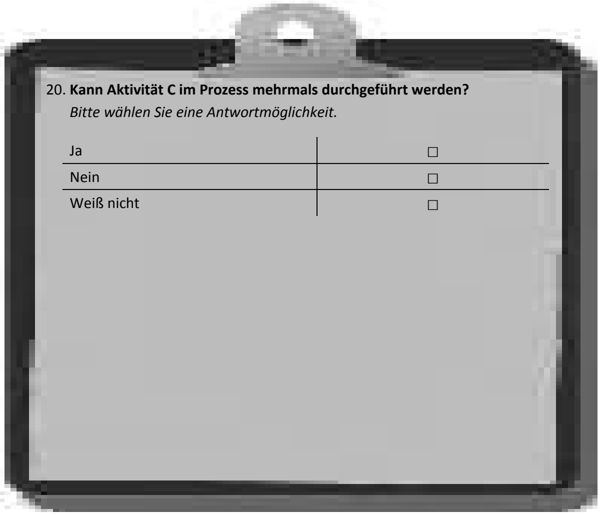
Aktivität A oder Aktivität B	☐
Aktivität A und Aktivität B	☐
Aktivität B und Aktivität D	☐
Aktivität B oder Aktivität D	☐
Weiß nicht	☐

27



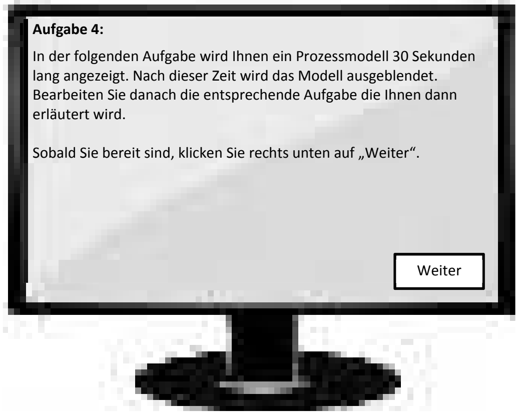


28

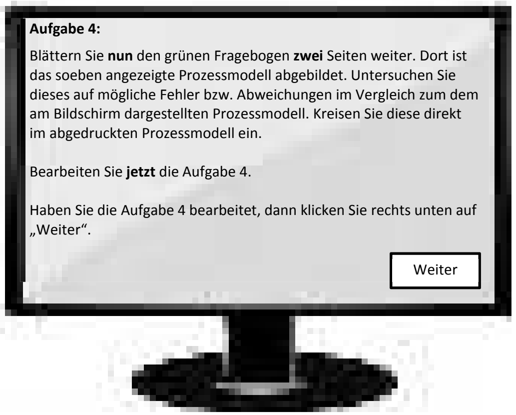

20. **Kann Aktivität C im Prozess mehrmals durchgeführt werden?**
Bitte wählen Sie eine Antwortmöglichkeit.

Ja	☐
Nein	☐
Weiß nicht	☐

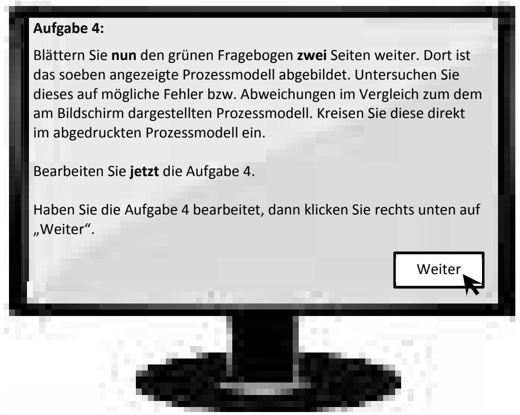
29

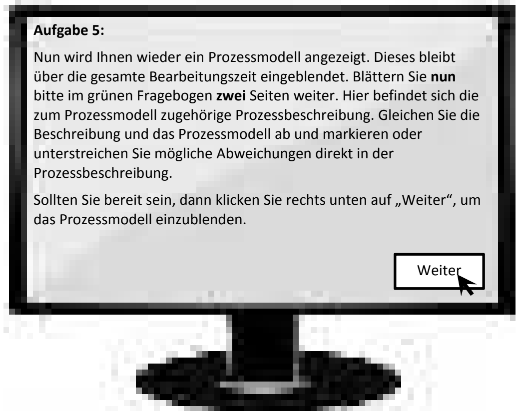


30

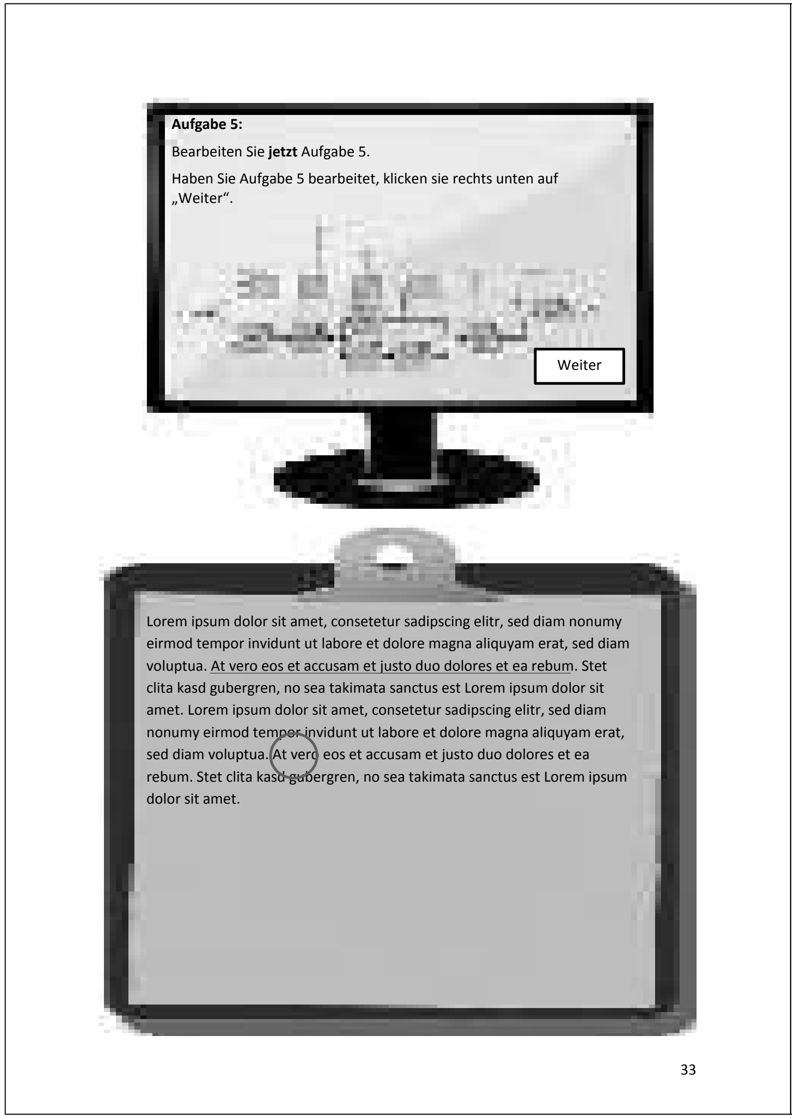

Aufgabe 4:
Blättern Sie **nun** den grünen Fragebogen **zwei** Seiten weiter. Dort ist das soeben angezeigte Prozessmodell abgebildet. Untersuchen Sie dieses auf mögliche Fehler bzw. Abweichungen im Vergleich zum dem am Bildschirm dargestellten Prozessmodell. Kreisen Sie diese direkt im abgedruckten Prozessmodell ein.
Bearbeiten Sie **jetzt** die Aufgabe 4.
Haben Sie die Aufgabe 4 bearbeitet, dann klicken Sie rechts unten auf „Weiter“.
Weiter

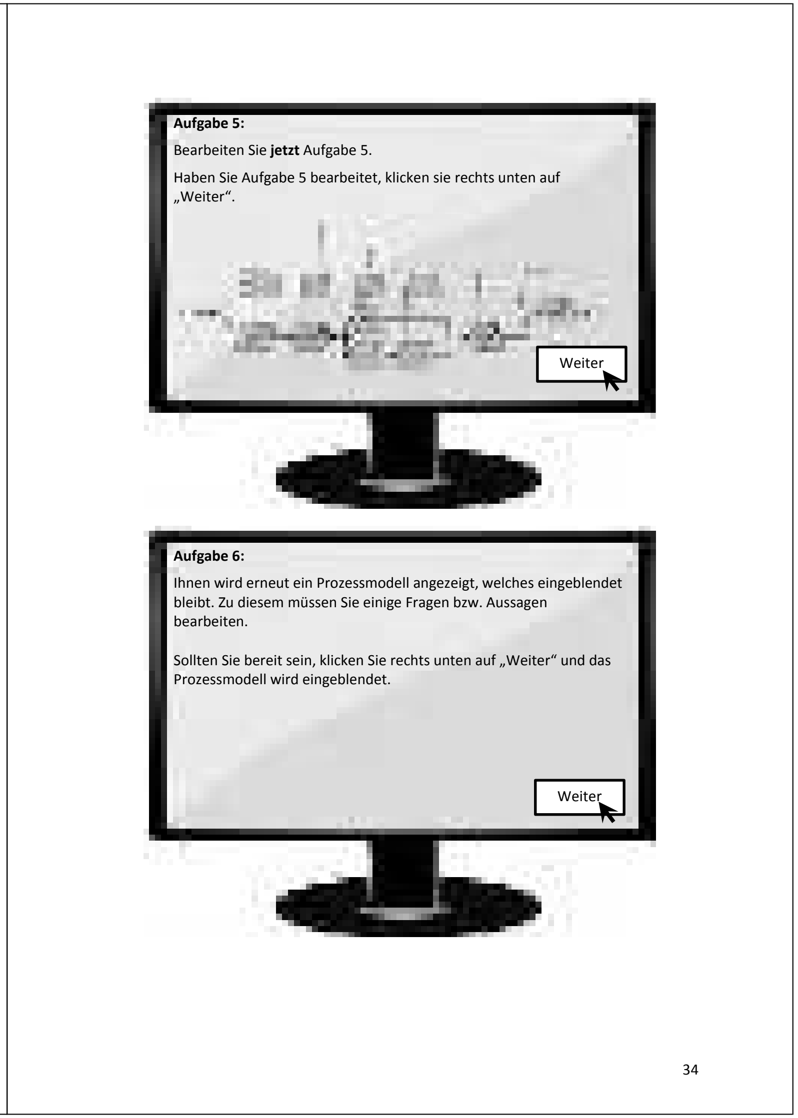
31

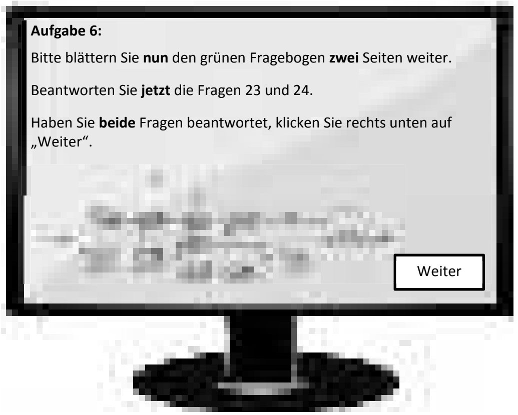

Aufgabe 4:
Blättern Sie **nun** den grünen Fragebogen **zwei** Seiten weiter. Dort ist das soeben angezeigte Prozessmodell abgebildet. Untersuchen Sie dieses auf mögliche Fehler bzw. Abweichungen im Vergleich zum dem am Bildschirm dargestellten Prozessmodell. Kreisen Sie diese direkt im abgedruckten Prozessmodell ein.
Bearbeiten Sie **jetzt** die Aufgabe 4.
Haben Sie die Aufgabe 4 bearbeitet, dann klicken Sie rechts unten auf „Weiter“.
Weiter

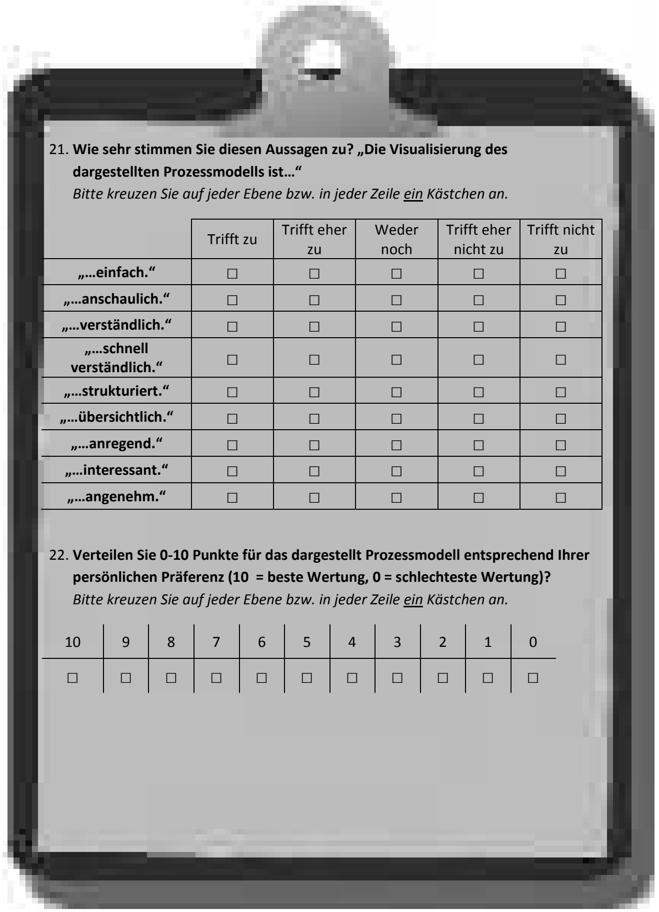

Aufgabe 5:
Nun wird Ihnen wieder ein Prozessmodell angezeigt. Dieses bleibt über die gesamte Bearbeitungszeit eingeblendet. Blättern Sie **nun** bitte im grünen Fragebogen **zwei** Seiten weiter. Hier befindet sich die zum Prozessmodell zugehörige Prozessbeschreibung. Gleichen Sie die Beschreibung und das Prozessmodell ab und markieren oder unterstreichen Sie mögliche Abweichungen direkt in der Prozessbeschreibung.
Sollten Sie bereit sein, dann klicken Sie rechts unten auf „Weiter“, um das Prozessmodell einzublenden.
Weiter

32


Aufgabe 5:
Bearbeiten Sie **jetzt** Aufgabe 5.
Haben Sie Aufgabe 5 bearbeitet, klicken sie rechts unten auf „Weiter“.
Weiter
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.
33


Aufgabe 5:
Bearbeiten Sie **jetzt** Aufgabe 5.
Haben Sie Aufgabe 5 bearbeitet, klicken sie rechts unten auf „Weiter“.
Weiter
Aufgabe 6:
Ihnen wird erneut ein Prozessmodell angezeigt, welches eingeblendet bleibt. Zu diesem müssen Sie einige Fragen bzw. Aussagen bearbeiten.
Sollten Sie bereit sein, klicken Sie rechts unten auf „Weiter“ und das Prozessmodell wird eingeblendet.
Weiter
34



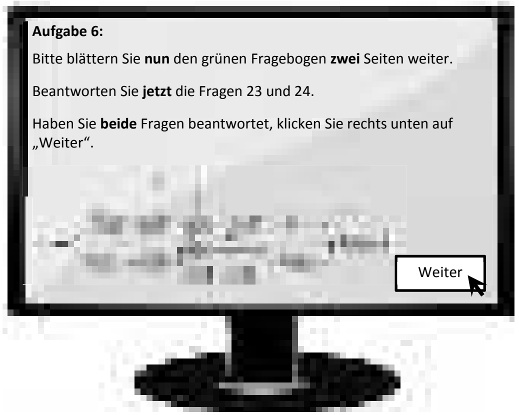


21. **Wie sehr stimmen Sie diesen Aussagen zu? „Die Visualisierung des dargestellten Prozessmodells ist…“**
Bitte kreuzen Sie auf jeder Ebene bzw. in jeder Zeile ein Kästchen an.

	Trifft zu	Trifft eher zu	Weder noch	Trifft eher nicht zu	Trifft nicht zu
„…einfach.“	☐	☐	☐	☐	☐
„…anschaulich.“	☐	☐	☐	☐	☐
„…verständlich.“	☐	☐	☐	☐	☐
„…schnell verständlich.“	☐	☐	☐	☐	☐
„…strukturiert.“	☐	☐	☐	☐	☐
„…übersichtlich.“	☐	☐	☐	☐	☐
„…anregend.“	☐	☐	☐	☐	☐
„…interessant.“	☐	☐	☐	☐	☐
„…angenehm.“	☐	☐	☐	☐	☐

22. **Verteilen Sie 0-10 Punkte für das dargestellt Prozessmodell entsprechend Ihrer persönlichen Präferenz (10 = beste Wertung, 0 = schlechteste Wertung)?**
Bitte kreuzen Sie auf jeder Ebene bzw. in jeder Zeile ein Kästchen an.

10	9	8	7	6	5	4	3	2	1	0
☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐


Aufgabe 6:
Bitte blättern Sie **nun** den grünen Fragebogen **zwei** Seiten weiter.
Beantworten Sie **jetzt** die Fragen 23 und 24.
Haben Sie **beide** Fragen beantwortet, klicken Sie rechts unten auf „Weiter“.
Weiter


Aufgabe 6:
Bitte legen Sie **jetzt** den grünen Fragebogen wieder verdeckt auf Ihren Tisch.
Haben Sie dies getan, klicken Sie rechts unten auf „Weiter“.
Weiter

37