



Fakultät II – Informatik, Wirtschafts- und Rechtswissenschaften  
Department für Informatik

# **KnoVA: A Reference Architecture for Knowledge-based Visual Analytics**

Dissertation zur Erlangung des Grades eines  
Doktors der Ingenieurwissenschaften

vorgelegt von

**Dipl.-Inform. Stefan Flöring**

Gutachter:

**Prof. Dr. Dr. h.c. H.-Jürgen Appelrath**

**Prof. Dr. Tobias Isenberg**

Tag der Disputation: 20. Juli 2012



---

## Zusammenfassung

Steigende Speicherkapazitäten und der Fortschritt im Bereich der Informationsverarbeitung führen zu einem rapiden Wachstum der gespeicherten Daten. In wachsenden Datenbeständen gestaltet sich die Gewinnung von Informationen und Wissen aus den Datenbeständen zunehmend schwierig. Bestehende automatische Analyseverfahren sind nicht ausreichend um die Datenbestände zu analysieren. Das Problem, dass die Menge der gespeicherten Daten schneller wächst, als die verfügbare Rechenleistung zur Auswertung der Daten, wird *Information Overload Phänomen* genannt. Die visuelle Analyse ist ein Ansatz diesem Problem zu begegnen. Sie vereint die Vorteile maschineller Auswertung zur schnellen Auswertung wiederkehrender Muster und zur Verarbeitung großer Datenmengen mit menschlichen Stärken wie Flexibilität, Intuition und Hintergrundwissen.

Im Prozess der visuellen Analyse wird Wissen durch Expertenbenutzer angewandt, um die Analyse voranzutreiben. In vielen Fällen wird von den Experten gleichartiges Wissen fortlaufend in mehreren Iterationen oder über verschiedene Analysefragestellungen hinweg wieder angewandt. Dieser Ansatz ist zeitraubend, kostenintensiv und frustrierend für die Experten. Daher kann ein Bedarf für Konzepte und Methoden zur Verhinderung von sich wiederholenden Analyseschritten identifiziert werden.

In dieser Arbeit wird eine Referenzarchitektur für wissensbasierte visuelle Analyse, die KnoVA RA, vorgestellt. Diese bietet Konzepte und Methoden um Wissen in Anwendungen zur visuellen Analyse zu repräsentieren, zu extrahieren und wiederanzuwenden. Die Idee hinter der Referenzarchitektur ist es Wissen, das im Verlauf der visuellen Analyse angewandt wurde, zu extrahieren um daraus automatische Verfahren abzuleiten, oder diese zu verbessern. Ziel dieses Ansatzes ist es, die Experten von Routineaufgaben zu entlasten und die Nachvollziehbarkeit und Reproduzierbarkeit der Ergebnisse zu verbessern. Die KnoVA RA besteht aus vier Teilen: einem Modell des Analyseprozesses, dem KnoVA process model, einem Meta-Datenmodell für Anwendungen zur wissensbasierten visuellen Analyse, dem KnoVA meta model sowie Konzepten und Algorithmen zur Wissensextraktion und -Wiederverwendung. Mit diesen Konzepten bildet die Referenzarchitektur eine Blaupause für Anwendungen zur wissensbasierten visuellen Analyse.

Zur Erstellung der Referenzarchitektur werden in dieser Arbeit zwei Anwendungsszenarien aus unterschiedlichen Domänen (Automobilbau und Gesundheitswesen) vorgestellt. In diesen Szenarien werden Anforderungen ermittelt, die als Grundlage für Design-Richtlinien für Anwendungen zur wissensbasierten visuellen Analyse dienen.

Am Beispiel der eingeführten Anwendungsszenarien wird die KnoVA RA in zwei Systemen zur visuellen Analyse implementiert. In TOAD, einem System zur Analyse von Busnachrichtenprotokollen in automobilen Bus-Systemen und in CARELIS, einer Anwendung zur interaktiven, visuellen Vereinigung von medizinischen Datensätzen. Diese Systeme veranschaulichen den Einsatz der KnoVA RA über verschiedene analytische Anforderungen und Problemklassen hinweg.



---

## Abstract

The increase in storage capacity and the progress in information technology today lead to a rapid growth in the amount of stored data. In increasing amounts of data, gaining insight becomes rapidly more difficult. Existing automatic analysis approaches are not sufficient for the analysis of the data. The problem that the amount of stored data increases faster than the computing power to analyse the data is called *information overload phenomenon*. Visual analytics is an approach to overcome this problem. It combines the strengths of computers to quickly identify re-occurring patterns and to process large amounts of data with human strengths such as flexibility, intuition, and contextual knowledge.

In the process of visual analytics knowledge is applied by expert users to conduct the analysis. In many settings the expert users will apply the similar knowledge continuously in several iterations or across various comparable analytical tasks. This approach is time consuming, costly and possibly frustrating for the expert users. Therefore a demand for concepts and methods to prevent repetitive analysis steps can be identified.

This thesis presents a reference architecture for knowledge-based visual analytics systems, the KnoVA RA, that provides concepts and methods to represent, extract and re-apply knowledge in visual analytic systems. The basic idea of the reference architecture is to extract knowledge that was applied in the analysis process in order to enhance or to derive automated analysis steps. The objective is to reduce the work-load of the experts and to enhance the traceability and reproducibility of results. The KnoVA RA consist of four parts: a model of the analysis process, the KnoVA process model, a meta data model for knowledge-based visual analytics systems, the KnoVA meta model, concepts and algorithms for the extraction of knowledge and concepts and algorithms for the re-application of knowledge. With these concepts, the reference architecture servers as a blueprint for knowledge-based visual analytics systems.

To create the reference architecture, in this thesis, two real-world scenarios from different application domains (automotive and healthcare) are introduced. These scenarios provide requirements that lead to implications for the design of the reference architecture.

On the example of the motivating scenarios the KnoVA RA is implemented in two visual analytics applications: TOAD, for the analysis of message traces of in-car bus communication networks and CARELIS, for the aggregation of medical records on an interactive visual interface. These systems illustrate the applicability of the KnoVA RA across different analytical challenges and problem classes.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Problem Statement . . . . .	6
1.3	Approach and Contribution . . . . .	7
1.4	Thesis Overview . . . . .	7
<b>2</b>	<b>Visual Data Exploration</b>	<b>11</b>
2.1	Information Visualisation and Visual Analytics . . . . .	11
2.2	Visualisation Models . . . . .	15
2.3	Classification of Visual Data Exploration Techniques . . . . .	18
2.4	Comparative Evaluation . . . . .	20
2.5	Summary . . . . .	22
<b>3</b>	<b>Requirements</b>	<b>25</b>
3.1	VA of In-Car Bus Communication Networks . . . . .	25
3.2	Visual Support in Manual Data Aggregation Tasks . . . . .	31
3.3	Implications for the Design of knowledge-based VA Systems . . . . .	41
3.4	Summary . . . . .	45
<b>4</b>	<b>The KnoVA Taxonomy</b>	<b>47</b>
4.1	Methodology and Outline . . . . .	47
4.2	Properties of VA Systems . . . . .	48
4.3	The KnoVA Taxonomy . . . . .	62
4.4	Summary . . . . .	74
<b>5</b>	<b>The KnoVA Reference Architecture</b>	<b>79</b>
5.1	Methodology and Outline . . . . .	79
5.2	Fundamental Structural Architecture of VA Systems . . . . .	80
5.3	Requirements for the KnoVA RA . . . . .	82
5.4	The Analysis Process . . . . .	83
5.5	Analysis Traceability . . . . .	88
5.6	Knowledge Extraction . . . . .	102
5.7	Knowledge Application . . . . .	108
5.8	Validation of the KnoVA Reference Architecture . . . . .	112
5.9	Summary . . . . .	115
<b>6</b>	<b>Implementation</b>	<b>117</b>
6.1	TOAD: A Tool for the VA of ICNs . . . . .	117
6.2	CARELIS: Visual Support in Manual Data-Aggregation Tasks . . . . .	128
6.3	Summary . . . . .	136

---

<b>7</b>	<b>Evaluation</b>	<b>137</b>
7.1	Methodological Foundation . . . . .	137
7.2	Evaluation Concept . . . . .	141
7.3	Accomplishment and Results . . . . .	144
7.4	Critical Reflection of the Results . . . . .	164
7.5	Summary . . . . .	169
<b>8</b>	<b>Summary and Outlook</b>	<b>171</b>
8.1	Summary . . . . .	171
8.2	Outlook . . . . .	175
<b>A</b>	<b>Example Data Sets</b>	<b>179</b>
<b>B</b>	<b>Evaluation Questionnaires</b>	<b>181</b>
B.1	System Usability Scale (SUS) . . . . .	181
B.2	NASA Task Load Index (TLX) . . . . .	182
<b>C</b>	<b>Evaluation Results</b>	<b>183</b>
	<b>Glossary</b>	<b>187</b>
	<b>Abbreviations</b>	<b>193</b>
	<b>Symbols</b>	<b>195</b>
	<b>Figures</b>	<b>199</b>
	<b>Tables</b>	<b>201</b>
	<b>References</b>	<b>203</b>



# 1 Introduction

The technological progress in the field of information and storage technology leads to a rapid increase in the amount of collected data [Car03]. It becomes particularly more difficult to derive knowledge from the data in growing amounts of data [Kei02b]. A reason for this is that the increase in the rate at which data is collected exceeds the progress in computing power and the development of novel algorithms for faster data analysis. As a result, the abilities to analyse data fall behind the abilities to collect data. This is being referred to as *information overload phenomenon* [KMS<sup>+</sup>08]. It leads to the circumstance that a large amount of the collected data is not used in a structured analytical process [Car03]. Such a process is necessary in order to gain insight in and to eventually derive knowledge from the data [AN95]. Manual data processing, on the other hand, is limited by the cognitive abilities of humans, which makes it impractical for large amounts of data. It poses the risk of erroneous output [GGS97]. Visual analytics is recognised as an approach to overcome the information overload phenomenon [KMS<sup>+</sup>08].

**Term (Visual Analytics).** *Visual Analytics (VA) is defined as the science of analytical reasoning facilitated by interactive visual interfaces [TC05]. It aims to embrace human strengths such as flexibility, creativity, intuition and, contextual knowledge into the analysis process [Sch07] by making use of the humans visual capabilities [GEGC98]. Advantages of VA are, that it is suitable for the analysis of heterogeneous data and no knowledge about complex statistical or mathematical algorithms is necessary in order to directly make qualitative statements about data phenomena [Sch07].*

According to [KMSZ09], VA is an iterative process with three distinctive steps: data selection and preprocessing, visualisation and model building. The iteration evidently leads to insight and, therefore, to the generation of knowledge. This knowledge then can be applied to the previous steps in the process, in a feedback loop, until the process of analytical reasoning is finished. This shows, that the application of expert knowledge is essential in the analysis process.

Hence the development of knowledge representations to capture, store and reuse knowledge applied and generated throughout the analysis process was recently identified as a major challenge by the VA research community [TC05].

## 1.1 Motivation

The demand for a knowledge based VA approach is affirmed by a variety of analytical challenges across different application domains. This chapter introduces two application scenarios. In each of the scenarios three key factors are identified: the application of expert knowledge by user interaction, the extraction of expert knowledge and the re-application of extracted knowledge in different analysis contexts.

The first scenario is the analysis of in-car bus communication networks. The development of these networks, today, is the single largest cost factor in automotive engineering. In this task it is vital for the automotive engineers to share their knowledge with their

colleagues in collaborative analysis sessions to track down the causes for possible errors. The second scenario is concerning visual support in manual data quality management tasks for medical records. When automatic quality management steps fail, records are integrated with the help of interactive visual interfaces. In this scenario expert knowledge about the course of diseases and about common errors and disambiguations is applied in the VA process.

### 1.1.1 Scenario 1: VA of In-Car Bus Communication Networks

In modern cars, a growing number of electronic control units (ECU) supported by a large amount of sensors and actuators are implemented to provide advanced functionality of the car. Examples are the airbag control, the engine control or the diverse multimedia facilities. All these components are connected with each other by specialised automotive bus systems.

**Term** (In-Car Bus Communication Network). *The components connected to the bus systems form an **In-Car Bus Communication Network** (ICN). An ICN is a network of electronic control units (ECU), sensors and actuators connected by automotive bus systems (such as CAN, MOST, FlexRay) [BG07]. It is used to implement advanced functionality in modern cars.*

These ICNs are used to control most of the functions in the car. Apart from the engine control, examples range from simple functions such as indicators or lights to highly complex integrated multimedia and navigation systems, and up to safety-critical systems such as airbag control and break assistants. The amount of hosts connected to the bus systems has grown rapidly over the last years up to over 100 ECUs and 300 sensors and actuators per car which are all interconnected to each other by 13 different bus types [SIB<sup>+</sup>11]. As a result over 15.000 messages per second are distributed across the ICNs. The proper operation of the ICNs is critical to the operation of the car. Errors in the intercommunication of ECUs and other hosts on the bus lead to failures in the car functionality and, eventually, can threaten the safety of the driver. It is, therefore, necessary to intensively test the ICNs in order to ensure vehicle and passenger safety. To achieve this goal, test engineers conduct comprehensive tests in simulated laboratory situations as well as during test drives with car prototypes. The objective of these test drives is to simulate real-world situations that can occur in order to provoke possible errors.

#### Analytical challenge

During the test drives, all bus messages are recorded. Test drives last from several minutes up to many hours. With 15.000 messages per second the amount of recorded data is very large. The collected message traces are stored in flat text files in which the messages appear in temporal order. A file that covers the events of an one-hour test drive typically consists of several GB of data [SIB<sup>+</sup>11]. Within this data a broad range of errors occurs, ranging from corrupted data due to interspersed noise by environmental influences, over to false outputs from ECUs, and even to concurrency issues, where the

large amount of messages on the bus results in dropped messages. As a result debugging ICNs is a huge practical data analysis challenge [Bro06, Hei05]. The objective of the debugging process is to identify the errors in the recorded message trace to reason about possible causes and influence factors. In this analytical process, the engineers entirely rely on the content of the recorded traces.

### Collaboration and Knowledge-Sharing

Due to the large number of different control units, typically there are experts for certain kinds of ECUs or even for certain car functions implemented in an ECU. If errors occur it is essential to pinpoint the ECU which caused the error. Often more than one ECU is participating in the message trace that lead to an error. In this case a single specialised analyst can not conduct the analysis alone. The analyst has to involve colleagues who are familiar with the other participating ECUs. Hence, the analysis becomes a collaborative task in which several engineers work on the same analysis question. Therefore, it is necessary for an analysis system to support this kind of collaboration.

Once an error is tracked down, the next step is to identify the cause of this error. If errors are found that might occur in similar situations elsewhere, for instance with other cars that have a comparable setup, other message traces are examined to identify such situations. In this way knowledge that an engineer has gathered in the analysis is applied back to the process. Only the expert who gained the insight has this knowledge though. This is unfortunate in situations where similar errors occur in message traces of other engineers. Hence a mechanism for knowledge sharing would be beneficial, that enables the engineers to share their knowledge with colleagues and to apply it across different analysis tasks. The following two challenges can be identified to create a VA system for the analysis of ICNs:

**Challenge 1:** How can collaborative analysis be supported by VA systems? For example how can experts collaboratively work at the same analysis questions and how can a system support knowledge sharing in the collaboration?

**Challenge 2:** How can implicit expert knowledge be extracted, to transfer results to other analysis tasks? For instance, how can findings be transferred from one trace file to another to simplify the analysis of the other trace file?

#### 1.1.2 Scenario 2: Visual Support in manual Data Aggregation Tasks

In the epidemiological cancer registry Lower Saxony (EKN) , data records for all cancer occurrences of cancer that are diagnoses in the federal state of Lower Saxony in Germany are collected. The objective of the EKN to collect and maintain a large data base about cancer diseases in Lower Saxony, as a foundation for further medical research. The EKN as organisation is separated into two divisions. The trusted party (VST), that stores all personal information of a cancer patient, is located in Hannover. The registry party (RST), that stores all medical information of a cancer patient, is located in Oldenburg.

There is a rigorous separation of concerns between the two parties, enforced by data protection regulation.

The records for a patient can originate from different sources. The two most common examples are clinical records and pathological records. Clinical records are created by the physician in a clinic. Pathological records are created by a pathologist who observes tissue samples collected in a surgery or in a biopsy. As a result often many records for the same patient arrive at the registry.

### Analytical challenge

To maintain high data quality and to keep track of the course of diseases it is inevitable for the registry to identify which records belong to the same patient or identify the same disease. Two different cases occur: Multiple records for the same patient and multiple records for the same tumour.

The first step to import a new medical record into the database is, to identify whether the record refers to a new or to an existing patient. To comply with data protection regulation the RST does not store any personal information. Therefore it is not possible to directly identify associated records. The analytical challenge here is to identify matching records in compliance with the data protection guidelines. To overcome this problem a number of fields for the personal information are replaced by cryptographic pseudonymous control numbers [TAS94] by the VST. The RST stores these ciphered control numbers.

The procedure follows the recommendations of [AMST96] for the technical realisation of the procedures defined by the cancer registry legislation (KRG) [Bun94]. The control numbers are used to compare the records, to identify matching records compliant to data protection regulation. Due to two technical limitations the control number based comparison does not yield in distinct result. Firstly none of the control numbers is a unique identifier and secondly the encryption is vulnerable to data quality issues. This is due to the fact that a small variation in the collected data, such as a misspelled name due to phonetic ambiguity – Jon vs. John – results in a mismatch for the generated control number.

To overcome these limitations the aggregation process in the RST is separated into two steps. Firstly records are automatically aggregated. This automatic aggregation is based on a set of business rules that decide whether records have to be aggregated or not. When the automatic aggregation step fails a manual aggregation is performed. Medically trained employees examine the records in a VA system called CARELIS to aggregate them facilitated by an interactive visual interface.

CARELIS presents the records in a tabular visualisation. Figure 1.1 illustrates a simplified example for the manual aggregation. On the left side of the figure an unassigned record and two possibly matching patients with their records are visualised. Each patient has a number of personal records. Both patients also have a tumour assigned to them. To each tumour at least one medical record is assigned. There are also unassigned records shown in the left part of the figure.

On the right side of the figure the same set of records is shown, after the analyst has as-

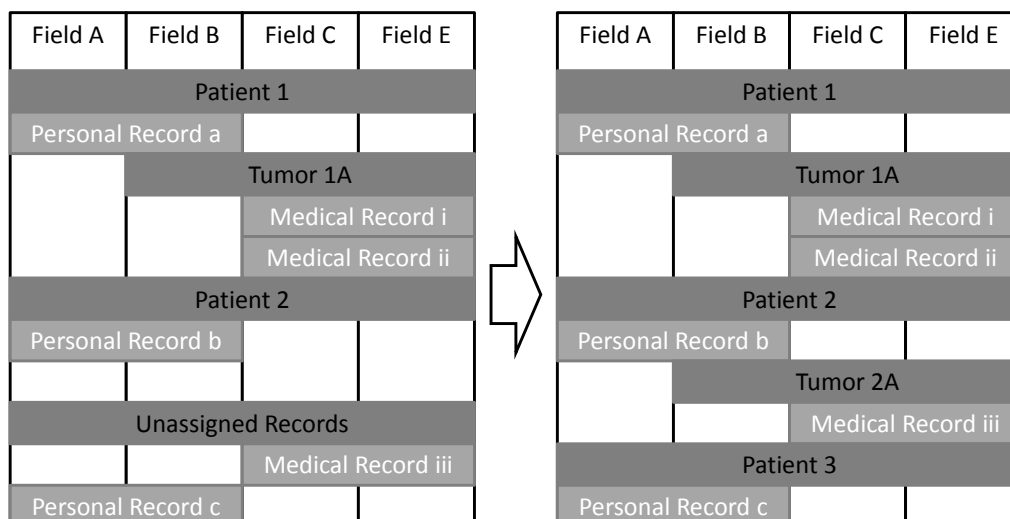


Figure 1.1: Schematic Overview over the Visual Interface of CARELIS.

sociated the unassigned medical record to the second patient. When this happens a new tumour is created for the patient to which the medical record is assigned to with information that can be derived from the record. The remaining personal record is discarded, when no other tumours match.

There are two possible manual aggregations in CARELIS: the described aggregation of new medical or new personal records and the aggregation of patients where two or more patients will be selected and aggregated. This results in one new patient that inherits all records that were associated with the previous patients.

### Knowledge management

Both, the manual as well as the automatic aggregation depend on the application of knowledge. The knowledge for the automatic aggregation is explicitly expressed in the form of business rules, which are automatically evaluated. When the automatic aggregation fails, trained experts perform a manual aggregation supported by an interactive visual interface.

Ideally all records are aggregated automatically, making manual aggregation unnecessary. When the system was first introduced in 1999 the largest part of records was aggregated automatically. Since then the amount of records that are aggregated automatically steadily decreases. Two influence factors contribute to this problem and ultimately lead to the demand for a novel approach to the aggregation process.

**Outdated Rules:** The knowledge for the automatic aggregation expressed in the business rules ages due to several reasons. This happens, for instance, when changes in encodings for certain values occur. An example for this is the encoding of geographic coordinates, which changed over the time from gauss-krüger coordinates to the UTM

projection. As a result rules relying on gauss-krüger coordinates do not apply any longer or - even worse - will result in false aggregations due to the missing values.

**Growing Database:** As the amount of records in the database increases, the ambiguity in the matching process increases. For records with common values (e.g. very common last names) possibly a large number of matching records exists in the database. For large sets the automatic aggregation fails as the available rules are not specific enough to resolve the ambiguity.

The growing amount of records that are not aggregated automatically leads to a larger amount of records in the manual aggregation task. In this task the analysts at the EKN apply their background knowledge to identify true matches. If this implicitly applied knowledge was extracted, it can be used to create more specific rules. In this way the rule base grows during the task of manual aggregation and thus provides the potential to steadily improve the results of the automatic aggregation. To support this approach the following challenges can be identified in this scenario:

**Challenge 3:** How can a VA system support the representation of expert knowledge that is implicitly applied by the expert in the analysis process?

**Challenge 4:** How can this knowledge be extracted into a knowledge base, in order to be re-applied the analysis process? For instance, how can this knowledge be used to support automatic aggregation steps?

## 1.2 Problem Statement

In both scenarios, the application of expert knowledge can be identified as important aspect in the analysis process. In the first scenario, expert knowledge is applied to pinpoint the sources of erroneous bus communication. In this scenario a demand for knowledge sharing between experts and across analysis tasks was identified. In the second scenario, knowledge is implicitly applied in the process of manual aggregation and a demand for the extraction of this knowledge to improve automatic analysis steps was identified. The challenges identified above can be grouped into domain specific challenges and challenges that are independent of the application scenario. The domain specific challenges concern the requirements for the interactive visual interfaces in the specific task. Independent from the domain, in each scenarios leads to challenges to represent and extract expert knowledge to make it reusable in other analysis scenarios. This is not supported by existing VA solutions and leads to the research question to be addressed in this thesis is:

*With which concepts and methods can expert knowledge, that was applied during the process of VA, be represented and extracted, to make it reusable?*

### 1.3 Approach and Contribution

To answer the research question, this thesis introduces the knowledge based VA (KnoVA) reference architecture (RA). The KnoVA RA aims to provide the architectural and algorithmic foundations to create VA applications that support the extraction and reuse of knowledge. It consist out of four elements:

**KnoVA Process Model:** As part of the KnoVA RA a process model for knowledge based VA applications is introduced. With this process model the steps of the analysis can be modelled. In VA applications it provides the foundation for a traceability of the analysis process.

**KnoVA Meta Model:** Complementary to the process model, a meta data model for VA systems is introduced. This meta data model is based upon a declarative classification of the properties of VA systems and builds the foundation for the representation of knowledge that is applied in the analysis.

**Knowledge Extraction:** Concepts to extract knowledge based upon the process model and the meta data model are introduced. It is also shown, how extracted knowledge can be generalised into a more abstract representation.

**Knowledge Application:** To apply extracted knowledge in another context it is necessary to identify, which knowledge can be applied. For this a matching algorithm is introduced, which identifies applicable knowledge. In addition to this algorithm it is shown how the identified knowledge can be de-generalised in order to be applied in other analysis situations.

The reference architecture serves as a foundation for the design of knowledge based VA systems. The process model is created based upon the examination of existing process models for VA and by taking typical elements of VA systems into account. The meta data model is derived in a bottom up process, based upon an examination of selected VA applications. It describes the properties of these systems in a hierarchical structure. The meta data model makes it possible to compare different analysis situations. The concepts for knowledge extraction and application are defined on the basis of this meta data model and are facilitated by an algorithms that generalise and de-generalise extracted knowledge.

### 1.4 Thesis Overview

This document is structured into eight chapters: Introduction, Visual Data Exploration, Requirements, The KnoVA Taxonomy, The KnoVA Reference Architecture, Implementation, Evaluation, and Summary and Outlook. Figure 1.2 shows an overview over the structure of this thesis. Illustrated are the chapters from top to bottom. On the right, the figure shows the most important results artefacts each chapter surrounded by a white circle. The figure also shows the core contributions of this thesis. These are surrounded by a black bordered rectangle.

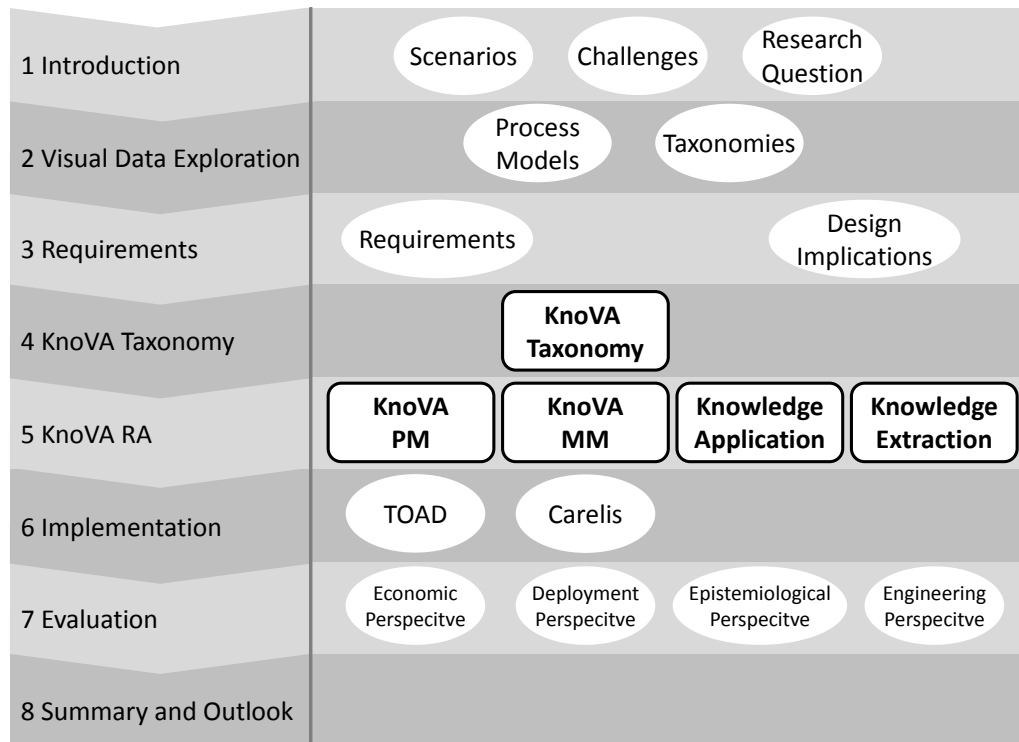


Figure 1.2: Overview of the Structure of the Thesis.

The thesis is outlined along the two scenarios introduced above. Firstly the introduction describes the two real world scenarios that lead to challenges and ultimately to the definition of the research question.

The following second chapter examines relevant fundamentals of visual data exploration within the scope of the scenarios. The most important aspects are the analysis of existing process models for VA and the analysis of existing approaches for taxonomies for visual data exploration. These fundamentals provide the foundation to approach the research question.

After this examination, the third chapter identifies the specific requirements for both scenarios. From these requirements general design implications for knowledge based VA systems are derived.

The fourth chapter explores the design space of knowledge based VA systems by examining a selection of existing VA tools. Based upon this examination the chapter then introduces the KnoVA Taxonomy as a core contribution.

This taxonomy for knowledge based VA that extends the related work in the field of taxonomies for visual data exploration. It is not directly derived from the scenarios. Instead, it is based upon the examination of related work, own previous work and VA systems with a general application domain. This bottom-up approach to derive the tax-



onomy ensures that the taxonomy is more universally applicable. Based upon the design implications and the taxonomy, then, the KnoVA RA is developed as the most important contribution of this thesis. As described above the KnoVA RA consists out of four parts: The KnoVA Process Model, the KnoVA Meta Model, concepts for knowledge extraction and concepts for knowledge application.

The sixth chapter shows how the KnoVA RA is used as the foundation for the implementation of two VA tools: TOAD and CARELIS. Each of these systems individually addresses the requirements in on of the motivating scenarios. Subsequently the chapter examines how the implementations based upon the KnoVA RA are facing the challenges identified above.

After this, chapter seven presents the evaluation of the KnoVA RA. The chapter firstly introduces the methodology of the evaluation. Then the implemented VA tools and the development process that lead to their implementation are examined in the evaluation. Chapter eight concludes the thesis with a summary and outlook.



---

## 2 Visual Data Exploration

This chapter sums up previous work within the scope of this thesis. For this, the domain of VA and the closely related domain of information visualisation (IV) are introduced with a focus on the process of exploration. Subsequently previous approaches to model the exploration process are set out and evaluated. After this, existing approaches for the classification of visual data exploring techniques are discussed for the better understanding of the properties of systems for visualisation exploration. Summed up these fundamentals lead to a description of the key aspects of knowledge application in the process of visualisation exploration and characterise the reference architecture discussed in the following chapter.

### 2.1 Information Visualisation and Visual Analytics

Historically the term visualisation applies to the task of forming a mental image to conceive real world or theoretical phenomena [Spe07]. IV is the science of creating visual mappings for abstract numeric data to gain insight. IV does not compulsory include information technology. Prominent examples of early information visualisation tasks are John Snows map of the cholera outbreak in the London SoHo district from 1854 or Charles Joseph Minards chart of the losses in Napoleons march to moscow from 1969, both generated using simple tools such as pen and paper [Spe07].

According to [KW02] the importance and possibilities of IV are increasing. This lead to a shift in the meaning of the term visualisation from the forming of a mental image to the creation of a visual mapping for abstract, numeric or textual data. Information technology that allows fast processing of large amounts of data is an important factor for this shift. In the last two decades, a large number of computer implemented visualisation techniques have been developed to support the exploration of large datasets. They are the data analyst's greatest single resource [Tuk65], as Tukey predicted. One reason for this is that the visual system is the human sense which provides more bandwidth and processing power than any other human sensory modality [SB03], which Shneiderman illustratively sums up with *the eyes have it* [Shn96].

#### 2.1.1 The Information Visualisation Cycle

IV is a process with several steps. On an abstract, level the process is defined by Shneiderman's information seeking mantra: *Overview first, zoom and filter, then details-on-demand*. [Shn96]. In a more detailed view it becomes clear that data exploration by IV is an iterative task, in which data sources of interest are any sort of information that can be visually structured but does not have a pre-determined physical structure. In this process, the analyst interacts with the visualisation system to explore the data according to the information seeking mantra. This interaction evidentially leads to insight, which influences the understanding and the mental image the analyst has of the data. It is, therefore applied back to the process iteratively. Card et al. describe this with their IV

cycle [CMS99]. This iterative process with four distinctive steps is illustrated figure 2.1.

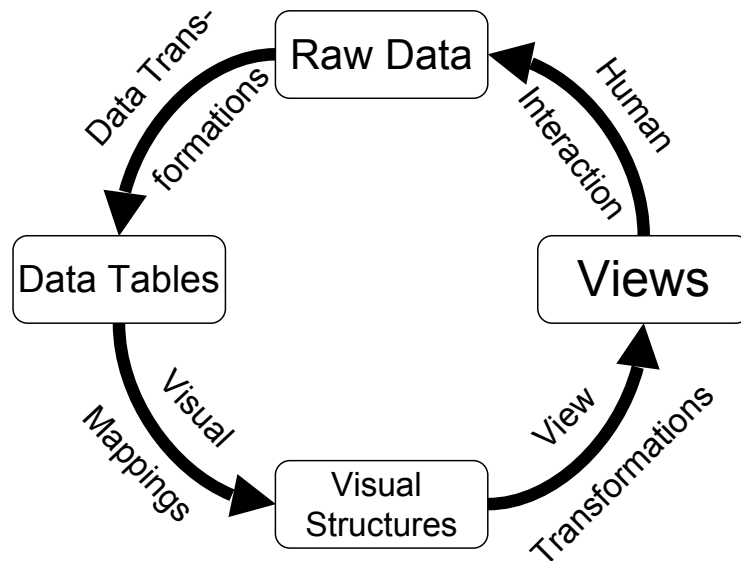


Figure 2.1: The Information Visualisation Cycle [CMS99].

The key feature of this process is that user interaction indicates the transitions between the iterative steps. In Card's model, the user specifies what parameters of the raw data are presented in a data table, e.g. by querying a subset of elements of a table in a database. Then the user defines a visual mapping, for instance by mapping nominal values to colours and then defines a view transformation, for instance changing the size or the orientation of the view. Finally, the insight is applied back to the process by human interaction with the views. The IV cycle provides an overview for the visualisation exploration process and does not provide a fine-grained model of the process. Even though user interaction can be identified as the key feature in the process, in IV research, the creation of meaningful visual mappings and the definition of transitions between the steps in the IV cycle has received most attention [JK03].

### 2.1.2 Visual Analytics

Recently out of the broad fields of scientific visualisation and IV, VA evolved as an independent research area. It is hard to draw a clean line between VA and IV, as both fields largely overlap. However, in general, the research focus in VA is in the fields of user interaction, sense making and reasoning. In their VA research agenda, Thomas and Cook define VA as the science of analytical reasoning facilitated by interactive visual interfaces [TC05]. Visual analytics aims to combine the strengths of machines with those of humans by integrating knowledge discovery in databases (KDD), statistics and mathematics for automatic analysis with the human capabilities to perceive, relate and conclude based upon knowledge and intuition.

Accordingly human factors such as interaction, cognition, perception, collaboration, presentation and dissemination play a key role in the process of VA [KMS<sup>+</sup>08]. The process of VA combines visual exploration with automated data analysis steps tightly coupled and coordinated by human interaction in order to gain knowledge from data [KKM<sup>+</sup>10]. To match the extended complexity and the slightly varying focus of typical VA task the information seeking mantra was refined to the VA mantra in [Sch07]: *Analyse first, show the important, zoom, filter and analyse further, details on demand*. The VA mantra takes respect of the fact that visual analytics aims to be a combination of automated processing (analyse first), user interaction and sense making. Accordingly, research in the field of VA focuses on the integration of automated and interactive steps and sense making. Often research not only incorporates visualisation and the generation of meaningful visual mappings but also specialised hardware, such as the PowerWall in Findex [KSS<sup>+</sup>06], a large display with a very high pixel density aimed to take maximum benefit of the users visual input bandwidth. Other systems such as [FHTA09], [HFS09], [FH10], [IC07] and [TIC09] focus on novel input facilities and human factors such as collaboration.

### 2.1.3 The Visual Analytics Process

Comparable to the IV cycle, the VA process is defined in [KMS<sup>+</sup>08]. Figure 2.2 gives an abstract overview of this process. It has four steps, each covering a different artefact of the exploration process. In between these steps a set of transitional relations is defined. Keim also introduces a formal model for this process in [KMS<sup>+</sup>08]. The following description is based upon this source, supplemented with information from the description of the VA process in [KKM<sup>+</sup>10]. In the formal model the four steps are defined as data sources  $S$ , visualisation  $V$ , hypothesis  $H$  and insight  $I$ .

In [KMS<sup>+</sup>08] the VA process is formally defined as follows:

The VA process is a transformation  $F : S \rightarrow I$ , where  $F$  is a concatenation of functions  $f \in \{D_W, V_X, H_Y, U_Z\}$ .

$D_W$  describes the basic data pre-processing functionality with  $W \in \{T, C, SL, I\}$  including data transformations  $D_T$ , data cleaning  $D_C$ , data selection  $D_{SL}$  and data integration  $D_I$  that are needed to make analysis functions applicable to the data.

$V_X, X \in \{S, H\}$  symbolises the visualisation functions, which are either functions visualising data  $V_S$  or functions visualising hypothesis  $V_H$ .

$H_Y, Y \in \{S, V\}$  represents the hypothesis generation process which can either be hypothesis from data  $H_S$  or hypothesis from visualisations  $H_V$ .

User interactions  $U_Z, Z \in \{V, H, CV, CH\}$  can either effect only visualisations  $U_V$  or only hypothesis  $U_H$ . Furthermore, insight can be concluded from visualisations  $U_{CV}$  or from hypothesis  $U_{CH}$ .

The VA process is an iterative process. Figure 2.2 indicates this with a feedback loop between  $I$  and the input.

An example of an analysis task modelled by this process is provided in [KMS<sup>+</sup>08]. The VA process is more detailed than the IV cycle. Specifically there are more iterative steps

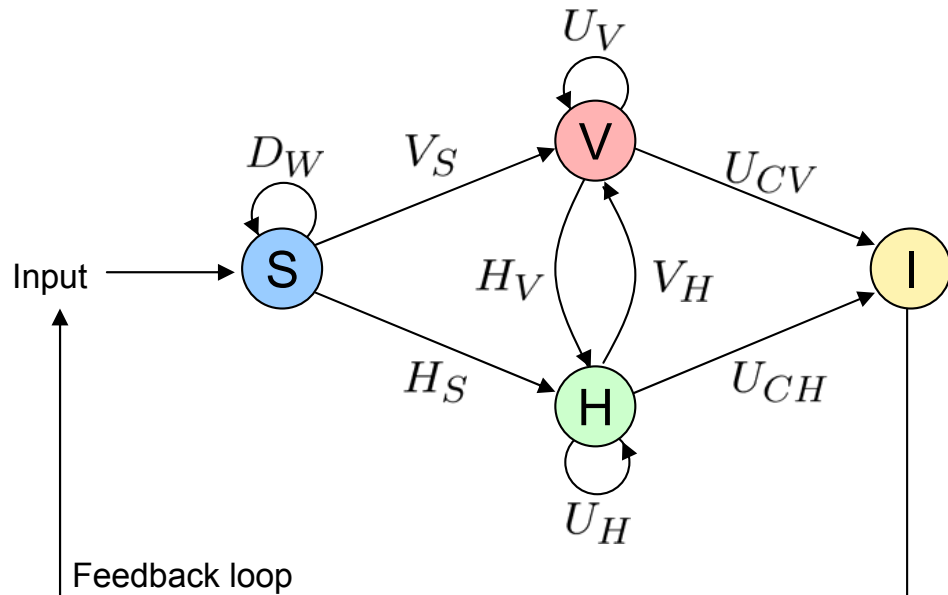


Figure 2.2: The Visual Analytics Process [KMS<sup>+</sup>08, KKM<sup>+</sup>10].

in the process. In addition to that Keim provides a comprehensive formalisation of the process [KMS<sup>+</sup>08]. However they do not propose the usage of a formal model of the process in VA applications. The process therefore is a descriptive model and hence on a comparable level of abstraction as the IV cycle.

#### 2.1.4 Relevance for this Thesis

In this section, the scientific fundamentals and basic principles of IV and VA were introduced as well as two existing concepts to model the analysis process.

IV is a mature research field with a history that reaches back beyond the computer age. During the past years, the scope in this research field has switched to highly interactive and explorative visual interfaces, which lead to the evolving of VA as research area. In this thesis the term VA is used according to the definition 1 by Thomas and Cook, which was explained in more detail in paragraph 2.1.2.

According to Card's IV cycle and Keim's VA process, analytical reasoning with the help of interactive visual interfaces is an iterative task, facilitating multiple user controlled steps, which eventually lead to insight and the generation of knowledge. The VA process is a more detailed model of this process than the IV cycle and is specifically defined to describe VA tasks. It not only provides a high-level abstraction of the steps in the analysis but also introduces a formalised definition.

Both process models encompass comparable elements. Especially relevant in the scope of this thesis are the explorative elements, described in the VA process. Therefore, a reference architecture for knowledge-based VA will have to reflect the steps of this process.

## 2.2 Visualisation Models

The IV cycle and the VA process provide models to describe visualisation exploration on a procedural level. They both exclude a detailed description of the creation of the visual mapping. The task of mapping input data (values) to the graphical display (the view) is referred to as visualisation transformation [JK03]. A visualisation transformation is necessary whenever a graphical representation for data is desired.

In scientific visualisation (SciVis) the visualisation transformation is typically shaped by the spatial appearance of the data. Data from medical imaging technologies such as MRI or CRT, for instance, represents the examined tissue and its structure. In many cases several data reduction and aggregation steps can be made to derive more expressive data. Several digital images of the human brain can be processed in order to illustrate the tracks of blood vessels, brain fibres etc. (compare [SEI10]). Its spatial origin enables a visualisation of the data that allows a direct correlation with the data source.

In contrast IV and VA focus on abstract numerical or textual data, abstracted from a direct physical representation and hence the definition of an arbitrary visualisation transformation is inevitable. The visualisation transformation is a mathematical function that maps input parameters e.g. from a database to output parameters of a visualisation. As an example, numerical values could be mapped to a colour palette according to a pre-defined mapping. Several attempts to model the process of visualisation exist. The following three models represent widely spread and recent approaches.

### 2.2.1 Data-Flow Model

In the data-flow model ([HM90, HLC91]) the process of visualisation is considered as a pipeline of subsequent visualisation transformations. The transformations connected build a pipeline of visualisation stages  $V_T$  connected by data-flow edges  $E$ . The data-flow model is advantageous for exploratory interfaces where the desired visualisation is not known at the beginning of the analysis [JK03]. The stages and edges of the visualisation transformation network form a directed graph  $VTN = (V_T, E)$ . Each  $v_i \in V_T = (IN_i, OUT_i, g_i)$  consists of a set of inputs  $IN_i$ , outputs  $OUT_i$  and a function to map inputs to outputs  $g_i : IN_i \rightarrow OUT_i$ . The whole visualisation transformation from input data to the graphical display is a path in the  $VTN$ . The data-flow model is most widely used in scientific visualisation applications [JK03].

### 2.2.2 Data-State Model

The data-state model introduced by Chi and Riedl [CR98] aims to be a conceptual model for all possible visualisation operations [CR98], whereas with operation they refer to user interactions with the visualisation system. To create the data-state model properties of operators are observed. Two classes of operators are identified: functionally similar operators and operationally similar operators. The first ones are operators which are semantically similar across applications, whilst the underlying implementation differs for different datasets. Examples are database operations such as joins or projections.

These operations are functionally equal on the data level. The interaction with the visualisation system that lead to an operation may vary. The latter ones are operators where the underlying implementation is similar across different implementations. Examples are rotations, scaling, translation, camera position manipulations etc. which - given the same rendered view based upon graphic primitives such as lines and polygons - can be implemented identically. Furthermore, they identify view and value as dimension to classify operators.

View operators are operators that only affect the visual mapping of data while value operators change the data itself. However, they note that no sharp line can be drawn between view and value operators. In addition to that, the sets of view and value operators intersect with operationally and functionally similar operators. Upon this assumption they create the data-state model based upon Card's visualisation pipeline [CM97, CMS99]. This is due to the observation, that the visualisation pipeline starts with data (values) and ends in a graphical representation of the data (views) [CR98]. The data-state model focuses on the transformation of data-states in the visualisation pipeline. In [JK03] a formal definition of the data-state model is given. The data-state visualisation pipeline is a directed graph  $DVP = (VD, ED)$  of data states  $VD$  which are connected by operations  $ED$ . Each state  $vd_i \in VD$  represents a piece of data. Each  $ed_i = (vd_{from}, vd_{to}, h_i) \in E$  is a visualisation operation  $h_i : vd_{from} \rightarrow vd_{to}$  which transforms the source state  $vd_{from}$  (its input data) into the destination state  $vd_{to}$ . The data-state model is the visualisation model most widely used in IV [JK03, Chi00]. Chi showed, that under the assumption that only data-states with single-input and single-output are defined the expressiveness of the data-state model and the data-flow model are equal [Chi02].

### 2.2.3 P-Set Model

The P-Set Model introduced by Jankun-Kelly [JK03, JKMG07] provides a model and framework for visualisation exploration. It expands the data-state model and gives a formal definition for the parameters of the input and output data of each state. It aims for a representation of visualisation paths as sequences of steps in the process of visualisation exploration. This graphical representation of steps over the course of the analysis and allows the user to identify repetitive steps and reoccurring patterns during the exploration. The P-Set model consists out of three elements: a visualisation transformation model, a parameter derivation calculus, and a visualisation session model.

The visualisation transformation model and the parameter derivation calculus are used to represent the applied visualisation transformation. The parameter derivation calculus also describes what types of parameters are being used in this process. Formally the visualisation transformation model and the parameter derivation calculus are described by a visualisation transform function  $t : D_1 \times \dots \times D_n \times P_1 \times \dots \times P_m \rightarrow R$  to describe the mapping of dataset types  $D_1$  through  $D_n$  and visualisation transformation parameters types  $P_1$  through  $P_m$  to the visualisation transform result type  $R$ . Visualisation transform parameter types are informally defined as any set of parameters (P-Set) that is part of



the domain of visualisation transformation functions [JK03]. The visualisation transformation result type is informally defined as a set that is in the range of a visualisation transformation function. Members of this set are directly representable in graphical form such as a raster image or shaded geometry [JK03].

The P-Set model incorporates a session model, which defines three state changing relations. These form the parameter derivation calculus [JK03]. In this calculus a P-Set is defined as  $p_j = \{p_j(1), \dots, p_j(o)\}$  with each  $p_j(l) \in P_k$  being a different parameter value for the same parameter type  $P_k$ . The tree relations of the parameter derivation calculus are:

1. Parameter application: A certain parameter value within a P-Set is replaced by another parameter value in order to derive a new P-Set.
2. Parameter Range: A continuous range of parameter values is generated between two discrete parameter values and applied to a P-Set.
3. Function Application: New parameter values are calculated by some function and then applied to a P-Set.

The complete formal definition of these relations can be found in [JK03]. The calculus can be used to record traces of visualisation session results  $s = (p, r, ts, d)$ . These results are tuples of a P-Set  $p$ , the visualisation transformation result derived from the P-Set  $r$ , a timestamp  $t$  and a parameter derivation calculus instance  $d$  which describes how this result was derived with  $d = \emptyset$  for the initial visualisation session result. All session results form a visualisation session  $VIS = (T, P, R, SR)$  with the sets of visualisation transformations  $T$ , P-Sets  $P$ , visualisation results  $R$  and the corresponding session results  $SR$  accordingly [JK03].

#### 2.2.4 Relevance for this Thesis

A challenge to be addressed in this thesis is to enable a knowledge transfer throughout different analysis sessions and knowledge sharing across different users. For this is it necessary to identify the knowledge that was applied in the course of the analysis. Above a description of the course of the analysis on a process level, for which the process models introduced in section 2.1 can serve as a foundation, it might be advantageous to incorporate a description of the state of the VA system in the reference architecture.

In this section pervious approaches to model the visualisation transformation, in which values are mapped to the view. In the iterative process of analytical reasoning on interactive visual interfaces this mapping is adjusted in each step. Previous approaches are either flow based, as the data-flow model or state based as the data-state model and the P-Set model, which is based on the data-state model. The P-Set model provides a simple definition for input parameter sets. However it is argued that in order to fully formalise the process of visualisation exploration the types of the data sets must be known but lacks to provide a general data model, which is not present in the P-Set model. A general data model will allow a comparison of states in the course of the analysis on a

higher abstraction level. Therefore in this thesis the data-state model will be used as a foundation for a process model and will be extended by a general data model. In the next section necessary foundations to approach the design of a general data model for VA systems will be addressed.

## 2.3 Classification of Visual Data Exploration Techniques

All of the approaches to model the visualisation exploration process focus on the procedural parts of the process and perform an abstraction of the actual properties of visualisation techniques and their means of interaction. For a better understanding of the domain of visualisation exploration, it is necessary to examine the properties of the large number of visualisation techniques that have been developed over the last two decades [KW02]. Several publications aim to provide a comprehensive overview of visualisation systems with varying focus.

With focus on systems used in IV research [SB03, Spe07, KW02] or with focus on the visual representation of data [Tuf83, KBT<sup>+</sup>08, McC09] or providing an example-based overview of the research field of interactive visualisation [ZSAL08]. While these publications provide an overview over different techniques, systems and the research field, they do not provide approaches or criteria for classification of visualisation techniques and their means of interaction.

There have been approaches to classify visualisation systems based upon the visualisation exploration process. For example in [Chi00] a taxonomy of visualisation techniques using the data-state model was introduced. Here visualisation systems are examined according to the steps of the data-state model and nine different classes of visualisation systems were identified. The taxonomy lacks a description of the identified classes and of criteria which allow to comprehend the ranking.

Another approach for classification of visualisation systems with two distinctive classes (task and data type) was introduced by [Shn96]. The task by data taxonomy identifies seven data types (one-dimensional, two-dimensional, and three-dimensional data, temporal and multi-dimensional data, and tree and network data) and seven tasks (overview, zoom, filter, details-on-demand, relate, history, and extract) as classifying properties. A detailed description for each of these properties is given and some examples for visualisation systems which can be described by the attributes are listed. It is aimed to be only a starting point and further research and increased detail is encouraged. However the methodology that lead to the selection of the classes and properties remains unclear and hence the work can only provide a vague overview. Other approaches for the classification of IV are data oriented [RM90], based upon the steps of the analysis [PHP03] or problem or task oriented [WL90, VPF06].

### 2.3.1 Visual Data Exploration Taxonomy

Recently Keim has introduced a taxonomy for visual data exploration (in the following abbreviated to VDET), with the focus on current research in the field of VA [Kei01,

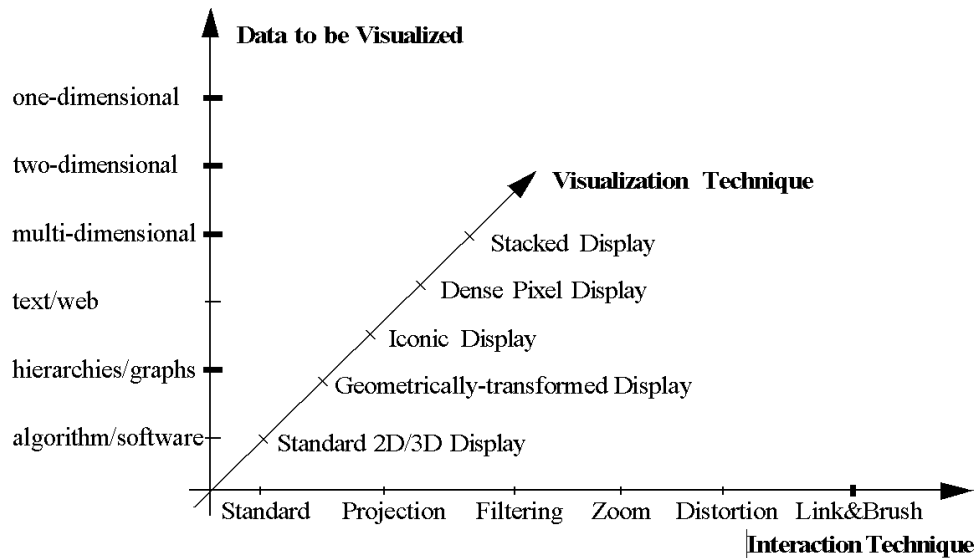


Figure 2.3: The Visual Data Exploration Taxonomy [Kei01].

KW02]. Here, based upon to the work of Shneiderman [Shn96], three orthogonal classes are defined: data to be visualised, visualisation technique and interaction technique. Within these classes several classifying properties are introduced, to allocate a visualisation system in a virtual three-dimensional space of visual data exploration by identifying the classifying properties matching to the properties of the system. Figure 2.3 gives an overview of the VDET. A point in this virtual space represents a VA specific VA system. The classification starts with the identification of the data to be visualised [Kei01], following the task by data taxonomy [Shn96]. The class describes one-dimensional data (for instance temporal data), two-dimensional data (for instance geographical maps), multi-dimensional data (for instance relational tables), text and hypertext (for instance web documents), hierarchies and graphs and algorithms, and software. Next comes the classification of the visualisation technique into five display categories: standard 2D/3D, geometrically-transformed, iconic, dense pixel and stacked (compare figure 2.3). The third dimension of the classification contains interaction and distortion techniques. Interaction techniques allow users to proceed the visualisation exploration by interaction with the visualisations [KW02]. The VDET defines six different categories: standard, projection, filtering, zoom, distortion and link & brush (compare figure 2.3).

Keim proclaims orthogonality of the three classes [Kei01], which means that any property of one class can be combined with any properties of the other classes and systems may incorporate several properties of the same class. For instance a specific visualisation system may support one-dimensional data as well as two-dimensional data [KW02].

### 2.3.2 Relevance for this Thesis

In paragraph 2.2.4 it was argued, that in order to derive knowledge from states in the process of VA it is necessary to provide a detailed description of the respective state. This can be achieved by a general data model for VA which allows to make comparisons of states on the abstraction level of the data model. As a foundation to approach a general data model in this section existing work in the field of classification of VA systems was introduced. A detailed classification scheme can serve as starting point for the definition of a data model, because such a classification will encompass the relevant elements of VA systems and thus subsume everything the potential of VA systems.

The VDET is a recent approach to classify VA systems in a structured way by identified descriptive classifying properties. Because the classifying properties here are descriptive and distinctive, this classification will be enhanced in this thesis to serve as the foundation for a general data model for VA systems based.

## 2.4 Comparative Evaluation

In this section important elements of the introduced process models and visualisation models are identified and summarised in order to allow for a qualitative comparison of these models with each other. This description will later be used to compare the approach for knowledge-based VA which is introduced in this thesis with the related work. In this chapter two approaches to describe the process of visualisation exploration have been shown and three approaches to model the visualisation transformation task, with the P-Set model being a hybrid approach.

Process/Model	Detail	Abstraction	Formalism	Knowledge
Information Visualisation Cycle	Low	Exploratory / Procedural	<input type="checkbox"/>	<input type="checkbox"/>
Visual Analytics Process	Fine-Grained	Exploratory / Procedural	■	( <input type="checkbox"/> )
Data-Flow Model	–	Visualisation Transformation	■	<input type="checkbox"/>
Data-State Model	–	Visualisation Transformation	(■)	<input type="checkbox"/>
P-Set Model	–	Hybrid	■	( <input type="checkbox"/> )

– criterion not applicable, ■ criterion is supported, (■) criterion is supported with limitations, () rudimentary support,  criterion is not supported,

*Table 2.1: Qualitative Comparison of existing Process Models for Visualisation Exploration.*

Even though the process models for IV and VA and the visualisation models describe

visualisation exploration on different levels of abstraction, they share common properties that can be used for a comparison. Table 2.1 shows four criteria: level of detail, level of abstraction, formalism and the support of knowledge extraction and sharing. In the following, these four criteria are used for a comparative discussion of the five models.

**Level of Detail:** This criterion describes the level of detail which is provided by the techniques. This criterion is only applicable to the two process models as the visualisation models are on a different level of abstraction. The IV cycle by [CMS99] roughly describes the process of IV as circular iterative process of four steps where user interaction leads to the transitions between the steps. It does not provide a fine-grained model of the visualisation exploration process. Specifically it lacks a description of user interaction in the process. Keims VA process [KMS<sup>+</sup>08] provides a higher level of detail in terms of user interaction. In addition to that, it takes into account that from visualisation always model building is proceeded and it defines that evidentially knowledge is being generated as insight is gained either from visualisation or from models.

The state for data sources  $S$  of the VA process in combination with the transition for data transformation  $D_W$  subsume the states raw data and data tables with the according transitions from the IV cycle. The stage visualisation  $V$  with the transition user interaction  $U_Z$  from the VA process subsume the stages visual structures and views from the IV cycle. The stages hypothesis  $H$  and insight  $I$  as well as the feedback loop and the integration of automated analysis are unique to the VA process and have no direct counterpart in the IV process.

**Level of Abstraction:** This criterion describes on which level of abstraction the respective technique can be ranked. Two distinct categories are identified. Exploratory or procedural models mainly aim to provide a high-level description of the process of visualisation exploration, not suitable for or at least not intended to serve as foundation of a software implementation of the process. Aim of these techniques is communication and understanding of the domain and a description of the process of visualisation exploration. The other category is visualisation transformation. Models with this level of abstraction provide a description of the task of creating a visual mapping from input data. All three visualisation models fall into this category. The P-Set model is an exception, as it provides a session model (compare 2.1) which is comparable to a process model.

**Formalism:** This criterion describes whether a formal description of the technique is available. With the exception of the IV cycle, all techniques offer some sort of formalism. The formalism for the data-state model is - in comparison to the others - not as strong, because it is ambiguous [CR98] and does not provide strong criteria to determine between view and value operators. The VA process offers the most detailed formalism for the process of visualisation exploration whilst the P-Set model offers the most detailed formalism for visualisation transformation.

**Knowledge Extraction and Sharing:** The scope of this thesis is to provide concepts and techniques for knowledge-based VA. Here the aim is to enable domain experts for an extraction of knowledge applied in the process, in order to share it with other experts and across analysis tasks. None of the techniques evaluated here (or rather the systems

based upon these techniques) offers mechanisms for explicit knowledge extraction. The VA process has a step knowledge, which refers to insight generated from visualisation or models within the process. It proposes a feedback loop in which knowledge gained is applied back to the process. However, it does not provide any specific definition of how the extraction of knowledge can be made possible. In particular it does not provide a data model for the datasets  $D_i \in (1, \dots, n)$  and the  $A_{i_1}, \dots, A_{i_k}$  and hence an explicit description of user interaction or knowledge is not possible.

The aim of the P-Set model is to visualise visualisation [JK03, JKMG07]. A data-state model based visualisation transformation model enriched by a session model achieves this. Thus, it allows to record a session and visualise the outcomes of this session in a graph-based visualisation to the user. The user can then detect reoccurring patterns in the exploration. In addition to that, it allows to record visualisation sessions in order to recap them.

The representation in these sessions is based upon the outcome of the visualisation transform and visual properties such as colour or rendered visualisation results. Therefore, an extraction of knowledge is not possible. The description of parameters is based upon pre-defined mime types, but no sophisticated data model is provided [JK03, JKMG07].

### 2.4.1 Conclusion

The examined techniques for visualisation exploration provide methodology for the description of the analysis process and for the visualisation transformation. The VA process provides the most detailed formal description of the analysis process and the P-Set model provides the most detailed formal visualisation model. In [JK03, JKMG07] it is argued that in order to fully formalise the process of visualisation exploration the types of the data sets must be known but lacks to provide a general data model. A general data model for the steps in the process of visualisation exploration used to describe the current visualised parameters, will allow a comparison of different visualisation steps with each other. In this chapter, the VDET was introduced. This classification provides a fundamental taxonomy that can be used to classify visualisation systems. Therefore such a taxonomy already fulfils basic requirements for a data model and can suit as a starting point for the creation of a general data model for visualisation exploration.

## 2.5 Summary

The objective of this chapter was to introduce important fundamentals in the field of VA and IV. For this at first the domain of IV was introduced in section 2.1, where at the term IV was defined, the historical development of IV was summarised and then the IV cycle as process model was presented. Then a detailed description of VA followed and the VA process was introduced. Lastly, the concepts introduced in this section were evaluated upon their relevance for this thesis.

After these two concepts were introduced, the terms visualisation transformation and visualisation model were defined in section 2.2. Here three important visualisation models

from previous work in this area were examined. Techniques and existing approaches for the classification of visual data exploration techniques were introduced in section 2.3 with a detailed view on the VDET.

Finally, in section 2.4 the existing approaches were compared with each other. The comparison results in a matrix of qualitative attributes for the classification and comparison of the approaches, which will serve in ongoing chapters as the foundation for a delimitation of the existing approaches and related work in comparison to the concepts and methods introduced in this thesis.





---

## 3 Requirements

To create a better understanding of the application context of knowledge-based VA, this chapter has two main objectives. First, it analyses the requirements in the two motivating scenarios. These requirements will later serve as the foundation for reference implementations in both scenarios. The second objective of this chapter is to identify general implications for the design of knowledge-based VA systems.

Section 3.1 analyses the requirements in the first scenario and section 3.2 respectively the requirements of the second scenario. These two sections have a similar structure. First the methodology of the requirements analysis is described and then the requirements are analysed. Subsequently the outcome of this analysis is summarised and discussed.

In section 3.3 the requirements for both scenarios are investigated for similarities across the diverse analytical challenges and the different domains. This leads to the identification of four design implications, which will serve as the foundation for the reference architecture for knowledge-based VA that is introduced in this thesis. Finally, section 3.4 summarises the chapter.

### 3.1 VA of In-Car Bus Communication Networks

The main challenges identified for the scenario of visual analysis of in-car bus communication networks in section 1.1 are partly domain specific and partly domain independent. In the requirements analysis, it is difficult to distinguish between domain specific and domain independent requirements.

Therefore, an approach is chosen where initially no differentiation between those categories was made. Instead the requirements are categorised in section 3.3.

#### 3.1.1 Methodology of the Requirement Analysis

To collect the requirements two methods were applied: semi structured telephone interviews preceding the development process and iterative interviews with guided demos via web conferences. This was accompanied by the iterative development process with reoccurring evaluation sessions in a user centred design approach. Participants in both processes were ten students of a student project group, two senior software engineers who mentored the students, the group manager of the test engineers and a varying number of test engineers as end users.

#### Target Users

Before the two methods to acquire the requirements are described in detail, the target users, which took part in the analysis process are introduced. Two different groups of target users took part in the requirements analysis:

**Group Manager:** The group manager works on an executive level and is supervising the engineering teams. He does not work with the analysis tools in his daily work but he is responsible for the acquisition of the analysis tools used, as well as for the de-

definition and the controlling of the overall process. The initial contact was established with the group manager, who was the most important partner in the initial interview phase. Even though the group manager would not work with the analysis system on a daily basis, he still is an end user, as he uses the analysis tools for instance in weekly/monthly review sessions.

**Test Engineer:** The test engineers work on the actual data. The data analysis is an important part of their work, which, over the analysis itself, encompasses the design of test setups and the collection of data from simulators or prototype vehicles. As the test engineers work with the analysis environment on their daily basis they were the most important partner during the agile development process, where they continuously gave feedback to the developed prototypes of the software.

### Semi-structured Telephone Interviews

In total three telephone interviews and video conferences were held preceding the development process. The focus of these initial interviews was to define the scope of the VA system to be developed as well as to get an overview over the target domain of the VA system and to form an essential understanding of the users. According to [Mun09] the domain problem characterisation is the first step in the process of creating IV systems. In preparation of the interviews, questionnaires were sent to the participants of the interviews. The first two of the interviews were held with a group manager. These two interviews mainly focussed on a higher level overview of the domain. In a third interview an end user was incorporated in the interview in addition to his supervisor. This was done in order to gain insight into his daily work and in order to get a more detailed view of understanding of the analytical challenges.

After each interview, a protocol was written and sent to the participants of the interview alongside additional questions that arose while summarising the interview. The participants were encouraged to annotate supplementary notes and to correct possible errors or misunderstandings. Additionally in this initial interview-phase the engineers were asked to either explain their daily work or show it in hands-on examples on their existing systems. In these sessions, the engineers were asked to employ a think aloud protocol and were specifically encouraged to express of what they think is helpful to them and at which stages in the existing process they need improvement. The outcome of this initial interview-phase was then used as the foundation of the agile software development process.

### Agile Development Process

During the initial interview phase, it became clear, that the requirements could not sufficiently be described in the telephone interviews alone. Hence, an agile, iterative software development process was chosen which allows adapting to complementary requirements that possibly arise during the ongoing implementation.

A development team of ten master students supervised by two assistant researchers was responsible for the development process over the course of six month. First, a set of

initial use cases was derived from the outcome of the telephone interviews. The use cases were then refined in additional telephone conferences. Participants in these telephone conferences were the members of the development team, their supervisors and a group of up to three users (the manager mentioned above and up to two end users).

After an agreement on an initial set of use cases was achieved, the implementation process started. The agile development process was structured into six iterations, each of which lasted four weeks. At the end of each iteration, an evaluation session was held with a group of target users. In this evaluation sessions intermediate results were presented to the end users via video demonstrations or life demonstrations guided by a developer. In these demonstrations either lo-fi prototypes (such as paper prototypes, artwork or PowerPoint slides) or working software prototypes were presented. Within the evaluation sessions, the users were encouraged to interject with comments. After each demonstration session, a discussion session about the process and possible adoptions to the requirements was held. During both demonstration, and discussion sessions voice recordings were conducted in order to retain the results. Protocols of the voice recordings were written after each session and sent back to the users who took part in the interview. These protocols contained newly discovered requirements and changes to original requirements to reconcile the protocol with the intention of the users. The edited requirements were then used as the foundation of the agile development process in the next iteration.

### 3.1.2 Requirements

As one outcome of the requirements analysis, we gained a deeper insight into the application domain, the existing analysis process and the analytical challenges. These aspects are already described in section 1.1.

Based upon the initial telephone interviews and the subsequent refinement we identified the following requirements, concerning the analysis process, collaboration, knowledge sharing as well as requirements concerning the design of interactions and visualisations.

#### Heterogeneous Visualisation Methods

During the interview-phase it became clear, that even though the work flow described follows a data analysis work flow from data integration over data cleansing and aggregation to data visualisation, a multitude of different tools was used. The analysis teams are split into specialised groups, each dealing with certain aspects of the car development. It was noted that different teams used specialised tool sets for their workflows. However even within the same team it occurred that different employees used different tool sets for the same basic workflows. The tool sets used differed in their appearance, in their functionality, their analytical scope and the level of detail they covered and even regarding the data formats, they supported.

It was observed that the heterogeneity of the tool sets is a major problem in collaborative tasks, where different users have to share their knowledge on the same problem. This happens for instance, when users from different specialised groups have to pinpoint the

source of a problem that influences the car functionality in both of their distinct fields of competence. Switching between different tool sets resulted in a loss of productivity. Whenever data needs to be reformatted or reintegrated in order to be used with a specific toolset this problem occurs. Additionally switching between tools with different analytical scope and level of detail implies a loss due to the users unfamiliarity with the tools of the respective other team, e.g. tools with different kinds of visualisations for the same or for comparable data. From the workflow description the following requirements can be derived:

**R1 – Integration of heterogeneous visualisation tools:** To allow for a smooth transition between a heterogeneous collection visualisation tools it is necessary to integrate all visualisations needed in the course of the analysis.

**R2 – Fast switching between heterogeneous views:** For the support of collaborative tasks, where several analysts share their knowledge about a certain problem, it is necessary to allow for an ideally instant switching between different views which allows the different experts to work on their preferred view.

**R3 – Direct mapping between heterogeneous views:** As a way to further support the communication between analysts it was required that several different views of the same aspect of the data can be shown at the same time. Hence, a mapping between heterogeneous views is needed, which allows a direct connection (brushing and linking) between the views to identify common aspects across the linked views.

**R4 – Transition between individual and group work:** The engineers demanded for a functionality to directly share their findings with their colleagues. Following the workflow description it became clear, that the engineers frequently switch between individual and group work and that there is a demand for a seamless transition between these two in various scenarios. For instance, analysts present some findings to share them with their colleagues. Then they work individually from that point. Hence, the analysis system must support various combinations of individual work and group interaction. Therefore, a function to distribute the current state of the VA system with another expert is necessary. Here again a mapping between different views was demanded, so that every analyst can examine the findings in his preferred view.

### Varying Levels of Abstraction

In-car bus communication networks consist out of up to 100 ECUs connected to each other by a number of different bus-systems and a number of interconnection between the bus systems. This results in large number of messages, today up to 15.000 messages per second [BG07]. The messages themselves have a high-dimensionality. A typical message has a number of fixed fields such as timestamps, bus type etc. and in addition to that a number of optional fields, depending on the message type. This structure further increases the complexity of the analysis task.

To cope with the sheer mass of information, for the analysis the bus-traces are aggregated. An analysis approach that tries to handle the data without pre-processing is time consuming and cost intensive [SIB<sup>+</sup>11]. Therefore, the data is pre-processed by mapping the timely course of the messages to predefined state machines. The interviews showed that, even though the aggregated traces can be used to quickly determine errors that occurred, they are not suitable to analyse the cause of an error. For this, the raw-data trace files have to be examined. A difficulty while handling the raw data files is to identify the root cause that lead to an erroneous state, as this may occur back in time and with 15.000 messages per second a large amount of messages may lay between the cause of an error and its actual appearance. In the interview phase, it became clear, that the message traces are being analysed on varying levels of abstraction. The following requirements concerning this issue could be identified:

**R5 – Overview visualisation of aggregated bus traces:** An adequate visualisation of the aggregated data according to the predefined state machines was demanded, that allows for a quick overview of the state machines defined in a bus trace. To achieve a quick overview, a high level of abstraction is necessary. Respectively the overview should mask details of the state machines.

**R6 – Detail visualisation of aggregated bus traces:** In addition to the overview visualisation a detailed visualisation of the timely behaviour of the state machines was pointed out as very important. Specifically the engineers asked for a functionality that would allow them to smoothly change the observed timespan, according to the fact that errors can occur long after their initial causes.

**R7 – Visualisation of raw data:** The engineers described that in order to pinpoint the actual cause of an error it is necessary to pierce through to the raw data, which can be associated with the state machines that are currently being analysed. Hence, a visualisation of the raw data is essential.

**R8 – Switch between abstraction levels:** The described analysis process allows to conclude, that a seamless switch between the varying levels of abstraction is invaluable for the engineers. Therefore, the visualisation system has to allow switching between the abstraction-levels at any given point.

**R9 – Sorting and filtering:** Due to the huge amount of messages the engineers stated the condition that is needed to apply sorting and filtering on the data in order to be able to approach and unveil the information hidden in the message traces. This was especially demanded for the visualisation of raw data because of its high-dimensionality which makes a naive approach to access the data virtually impossible.

**R10 – On-view selection and interaction:** In the interviews it became clear, that the course of the analysis is assigned by the findings within the traces. For instance, only those state machines which have initially been identified as interesting will be selected for further investigation. The engineers asked for direct interaction with the

visualisation that allows them to implement a smart pipelining and filtering function analogue to what is known in command line pipelining or video editing.

### Collaboration and Knowledge Sharing

When errors occur within a certain ECU, like for instance the ECU of an integrated headlight and only affect the behaviour of this ECU, the responsibility to trace the cause of the error can be determined to the analyst or the analysis team with the competence for lighting integration. During the interviews, however the engineers pointed out that it is more likely for errors to occur in the communication between different ECUs and that often errors that initially occur in the scope of one ECU will change the state or the behaviour of other ECUs. If this is the case, then no clear judgement can be made concerning the responsibility for this error team-wise. Instead, the engineers reported that in such cases both a close communication and collaboration between individuals or teams with varying competence is needed in order to share their specific knowledge. It was said that in a typical use case one engineer would mark his finding in an analysis and then hand it over to another analyst. This analyst than can, at a later point in time, work on the problem with his expertise. The following requirements were identified concerning these aspects of and knowledge sharing:

**R11 – Traceability and visual history:** In R4 the support for a smooth transition between individual and group work was demanded. For this, the engineers wished for a functionality that would allow them to resemble the course of the analysis that lead to the finding. Hence not only the finding itself is relevant but also the analysis trace which lead to this finding. This expresses the demand for features, which allow the analyst to reason about the course of the analysis, e.g. which views he used and which operations on the data he defined. The group manager specifically said that a visualisation of the course of the analysis preferably on different abstraction levels would be helpful both in individual analysis sessions and in communication with other analysts. This was referred to as visual history of the analysis session.

**R12 – Knowledge transfer:** The engineers asked for means to transfer their findings to their colleagues. As stated above for this not only the transfer of the actual state but also a transfer of the whole analysis trace is necessary for a meaningful transfer of knowledge. In addition to that, they asked for methods to directly share knowledge e.g. methods to present certain findings, methods to work collaboratively on the same data etc.

**R13 – Traceability and reproducibility:** The analysis often is a complex task and tracing down the causes of errors can span across longer time-spans sometimes even across more than one workday. From this arose the requirement to save the complete course of an analysis, so it can be re-opened at a later point in time in order to preserve insight gained in a session. They specifically asked for a overview function that allows them to identify where in the data they are and which analysis steps they already took and for a trace function that allows them to see the steps that lead to the

current state in the smart pipelining and filtering (see R10).

**R14 – Externalisation of findings:** The engineers reported that often they have to deal with reoccurring patterns. They asked for a method to mark phenomena found in an analysis in order to find similar occurrences in other traces. They described a variety of properties which can define patterns, ranging from the identifier of states, state names to patterns in communication. Specifically they wanted to be able to express findings in a way that they can use these findings to apply them to other traces (or even within the same trace) in order to save time in the analysis.

## 3.2 Visual Support in Manual Data Aggregation Tasks

As motivated in section 1.1, there are two major problems in the existing application for data aggregation that lead to the demand for a new approach: the fact that the existing rule set is outdated as a result of changing encodings in standards and other reasons and the fact, that the ambiguity in the matching process increases with a growing database thus reducing the effectiveness of the automatic aggregation.

The main challenges identified in section 1.1 are to identify the requirements for an interactive visual interface that facilitates the extraction of expert knowledge and then to identify how the expert knowledge can be extracted in order to be used in an automotive aggregation process. Facing the same difficulties as in the previous example to determine between domain specific requirements and the domain independent requirements, a similar integrated approach to analyse the requirements was chosen, resulting in a comprehensive set of requirements for the VA system. Likewise section 3.3 categorises the requirements across both scenarios.

### 3.2.1 Methodology of the Requirement Analysis

The requirements analysis here was structured into four phases. At first in an initial, unstructured interview with the medical director of the EKN was held. In addition to that in this phase the documentation of the existing knowledge-base was examined (for details see section 6.2). Based upon the impressions gained in this phase, the work process was analysed in a field study where selected users were examined performing their daily work. After this field study all people, currently working with the manual data aggregation tool took part in semi-structured interviews. The outcome of these initial phases then was used as the foundation of an iterative, user centred design process, in which the requirements were refined further. In the following subsections, the approach in the four phases is explained in detail.

#### Target Users

Three distinct roles of users working with the visual interface for data aggregation in the EKN were named: The medical director of the EKN, a number of medical record technicians and two data analysts. All of these users took part in the requirement analysis. Each of the roles has different duties and responsibilities and hence provides a different

perspective in the requirements analysis.

**Medical Director:** The medical director (MD) works on the executive level and is supervising all other employees of the EKN. He is working with the tool for visual aggregation regularly, but not necessarily on a daily basis. He keeps track of long-term development of the registry, decides how to react on changing requirements and defines the data management processes. Additionally he defines quality standards regarding the data management, is responsible for data quality management (DQM), and shares some duties with the data analyst.

**Medical Records Technician:** At the time of the requirements analysis there were six medical records technicians (MRT) working at the EKN, each of which was participating in the requirements analysis at some point. The MRTs are responsible for the whole process of data integration and aggregation, supported by diverse tools for import, error correction, record linkage and visual aggregation according to the guidelines set by the MD. Each MRT has several years of experience with the existing tool for visual aggregation. The MRTs are on an expert training level concerning medical documentation, specifically for the documentation of cancerous diseases. The level of expertise concerning the knowledge-base in the background, which performs automated aggregation steps before the manual aggregation task, varies largely.

**Data Analyst:** There are two data analysts in the EKN, one is holding a master of public health degree. The analysts primarily work on the analysis of the aggregated and quality assured data integrated into a data warehouse (DWH). They work with specific analysis tools for report generation, which allow for a graphical representation of the data on thematic maps. They manually enrich the reports with medical background knowledge and conclusions. They use the visual interface for data aggregation, where all details of the data can be seen if they need to check specific values. While doing this they sometimes perform DQM tasks, when they stumble upon errors.

During requirement analysis, it became clear, that there is a shared responsibility for DQM and thus actually an implicitly defined role for quality management exists. It has been reported, that all six MRT are performing DQM alongside their daily work as errors occur, while examining records that are supposed to be aggregated. Mainly though the responsibility for DQM was upon the medical director and upon one of the two data analysts. In addition to that, another user, who could not be assigned to any of the above roles, was reported to occasionally use the visual interface for data aggregation for DQM purposes. As nearly all users therefore somehow performed DQM tasks therefore the requirements concerning DQM were collected alongside the requirements to the interactive visual interface for data aggregation.

From the three user roles, the MRTs are the most significant user group for the visual data aggregation, as they use the software on a daily basis and they perform the actual aggregation task. They spent a large portion of their daily work in the visual interface for data aggregation. In fact, at least selective MRTs spend most of their daily work



on the visual interface. Other users use the visual interface primarily to view data, for instance to check for specific details and less frequently for DQM as errors may occur by coincidence while they are viewing the data.

### Initial interviews

To start of the process of the requirements analysis several unstructured as well as a semi-structured interview were held. Participants were the MD and between one and three of the MRTs. These interviews were following several objectives. The first objective was a domain problem characterisation [Mun09], similar to the previous example. The second objective was to determine organisational aspects such as how to analyse the requirements in co-operation with the users and the last objective was to identify possible technical limitations such as the number and the characterisation of existing workstations etc.

The unstructured part of the interviews was held alongside with existing monthly meetings concerning the overall software setup and development. Each meeting had a time frame of 60 minutes. Between three and four developers took part in these meetings, each developer was responsible for specific tools with two developers being responsible for the tool for visual data aggregation. During a number of these meetings, the main focus was specifically set to the visual data aggregation. The domain problem characterisation itself was brief, as the developers were familiar with the domain itself. The focus of the meetings therefore was mainly on changing and novel requirements, which could not be fulfilled with the existing solution and on changes in the data and the domain which lead to the demand for a novel approach (compare section 6.2).

Questions that occurred in these meetings were addressed directly in the meetings, which in some cases lead to an extension of the time-frame. After each meeting, a protocol was written and spread amongst the participants for review. In the interviews, the MRTs had the opportunity to express their view on the software. They were especially encouraged to express problems in the existing setting. For this they typically brought a list of errors and design flaws with them, that occurred to them in their daily work.

In addition to that, before the further requirements were analysed, a master student and a developer held a semi structured kick-off interview with the MD. The objective of this interview was to determine organisational aspects concerning the further process of the requirements analysis and to identify technical limitations. This interview was structured by a questionnaire with eight questions. An audio protocol of this interview was recorded and the requirements were analysed in a follow up meeting of the developers.

The kick-off interview lead to the decision for two complementary approaches for the further requirements analysis: semi structured interviews with the MRTs followed by a field study to observe the daily work of the MRTs

### Semi structured Interviews

After the initial interviews were done a semi-structured interview with each of the six MRTs was held. These interviews were scheduled to 90 minute time frame. The objective of these interviews was to gain insight into the processes and approaches to the

tasks and to the individual work methods of every MRT and to motivate the users for additional steps in the development process by clarifying the objectives and goals of the new development. Another objective was to prepare the users for the development process and to insure that the user is comfortable with his role in this process.

A main focus was to gain a detailed illustration of every step in the task of visual data aggregation. For this the MRTs showed their work approach in the existing setting with a set of sample records on which they explained the work process and the task from their perspective. The developer view of the problem domain and possible solutions to the requirements were held back in this phase in order to avoid a biasing in the requirements analysis.

Preliminary to the interview, a questionnaire with nine questions and an explanatory introduction was given to the MRTs, which then was used to structure the interview itself. The outcome of the interviews was collected in a written protocol. After each interview, the protocol was reflected with the participant in the interview until the participant agreed upon the selected outcome.

### Field Study

The field study was chosen as complementary element to the semi-structured interviews with the MRTs. The outcome of the previous two phases was essentially the user's view on the target domain and the users view on design flaws. The field study followed three main objectives. The first objective was to gain a detailed insight into the tasks and the work approach of the users and to create a thorough understanding of the process by the developer. In addition to that, an objective of the field study was to collect requirements concerning future challenges from the user. The third objective lastly did not concern requirements directly but to govern the domain from the developers' perspective. This concerned especially the structure of the data and the structure and characterisation of the knowledge applied by the user. The outcome of this last objective is described in section 6.2.

In each field study, session a developer examined the work of one MRT in a real-world setting. Each of these sessions was determined for a 60 minutes time frame. The sessions were structured by a short questionnaire, which was handed over to the MRT in beforehand to prepare the examination (see appendix B). During the sessions, the MRTs were performing their daily work on the existing interface for visual data aggregation. Alongside this examination a protocol was written down which was reflected after the session.

The examination was later repeated on a prototype of a new visual interface with selective MRTs. In this setting, no automated aggregation steps were performed before the manual aggregation. The objective of this was to gain greater insight into the knowledge applied, which was easier when no automated rules for knowledge application are active.

### 3.2.2 Requirements

Based upon the initial interviews, the subsequent refinement, and the field study the following requirements, concerning the visual interface, knowledge extraction and management concerning traceability and knowledge sharing were identified.

#### Visual Interface

As mentioned above, an integrated approach for the requirements analysis was chosen. In this approach not only the requirements concerning the knowledge extraction and management and the knowledge-based analysis process were collected, but also general system requirements concerning the visual interface and the interaction with the system. As these requirements are concerning the general behaviour and appearance of the analysis system, these are described here first.

**R15 – Information reduction and filtering:** In order to deal with record sets containing large collections of records the MRTs demanded for interaction methods that are advanced compared to their existing system. In the field study, it became clear, that although large amounts of records may occur within one transitive closure, typically the MRTs process these in a pipelined order, comparing only a few records at a time. Hereby they focus on records, which are likely to be aggregated, in order to reduce the amount of stand-alone records as quickly as possible. A large amount of records that are not likely to be aggregated with each other therefore might clutter the display and distract from the aggregation task. Hence, meaningful methods for record sorting and filtering are needed. In addition to that, the field study has shown that from all available fields of the records only a subset is important for the majority of the aggregation tasks. Only when no judgment can be made upon this (user defined) subset additional fields will be observed. Therefore, a need for information reduction techniques, which allow isolating and sorting of these important fields on the visual interface, in order to customise the interface to the specific preference of a user can be identified.

**R16 – Context sensitive information reduction:** After tests with a prototype of the new visual interface, which already provided methods for information reduction according to R15, the MRTs noted, that the important fields might vary according to specific values. This lead to the requirement for a context sensitive information reduction, which will automatically adjust the shape of the visual interface (e.g. by modifying the set of the visualised parameters) based upon the value of certain fields.

**R17 – Advanced highlighting of important information:** In the existing visual interface for data aggregation values, which differ across different records, are highlighted by red coloration. When transitive closures contain large numbers of records, the probability that values differ quickly increases and therefore the coloration quickly turns useless, as even if in a set of 20 records only one has a deviant value in a certain field, the respective fields of all records will be marked red. Therefore, the MRTs

demanding that they can specifically choose which records should be examined for the comparison and the highlighting.

**R18 –Adjustable probabilistic distance:** In the existing visual interface transitive closures of records are displayed. These transitive closures are representing the outcome of the automatic matching process. They contain, according to the definition of a transitive closure, all those records, which are associated to each other based upon their weight including transitive associations. In some circumstances, especially when the amount of transitive associations is very high, this can lead to closures, which contain a large amount of presumably matching records, of which only a few match directly. In the visual interface these will be displayed as associated even though the probabilistic match weight between any two records (a, b) of the closure in average is particularly low. The fact that further records in the existing database can match with a record in a transitive closure. Hence, this existing record will be included in the closure, even though it might be unrelated to all the other records in the closure.

This further increases the number of records that are visualised at the same time and hence the visual clutter. This problem leads to the request, that in the new visual interface only those records will be visualised coevally that exceed a certain match weight in the direct matching, alongside possibly records from the existing database which match with a certain weight to one of those records. Thus, the amount of coevally displayed records will decrease. For this approach, it is necessary for the MRTs to be able to adjust the threshold of the match weight, which defines which records are displayed, so that more records from the transitive closure can be shown.

### Knowledge extraction and management

Examining the MRTs in the field study showed that, even though there is only one tool for the aggregation task, there are different kinds of knowledge that the users apply to the system during the aggregation tasks. There are three distinctive tasks, which are performed by the MRTs: aggregation of tumour records consisting mostly of medical data and aggregation of patient records consisting mostly of epidemiological and social-demographic data. For both of the aggregation tasks the MRTs will typically select a subset of the entities displayed, associating those entities to each other based upon knowledge about the composition of fields. The data is hierarchically structured, with one patient having one or more subsidiary tumour records. Thus, if two patients are associated and hence marked to be the same entity, their tumours are automatically allocated to the newly created merged patient record.

Within one transitive closure that is displayed this step may occur several times until the MRT is satisfied with the outcome of this process. After each step or after a series of steps the MRTs define best-of values. These are values, which represent the best available value out of the original value to build one virtual record spanning across a number of aggregated physical records. Ideally, all records, which are aggregated, should share the same values for all their attributes. In reality, values might be missing or some records might contain values that are more specific (e.g. a more specific diagnosis) or more generic than the others, or some records might be incomplete etc.

Some of the variables of two associated records may vary in many cases. For instance, specific records may contain a more specific diagnosis, depending on the equipment of the diagnostic site that allocated the record. In these cases, the MRTs have to make a decision which of the ambiguous values best represents the aggregated entity. This process is defined as best-of definition. The following requirements concerning the application and management of knowledge could be identified concerning upon the observations in the field study and the interviews:

**R19 – Direct and continuous knowledge extraction:** The MRTs pointed out, that due to the large variety of different analysis situations it is hard for them to define a comprehensive set of the knowledge they apply. Therefore, they asked for means to directly define the knowledge they applied in the process of the application. The MD, who mentioned that the data evolves over time, also favoured this, because he said that the knowledge has to be adapted to the existing data continuously. In addition to that, the MRTs requested the process of knowledge extraction to be as less intrusive to their normal work patterns as possible. Some even demanded that an extraction of knowledge has to be made optional, so it can be deactivated. However, others considered this unfortunate.

**R20 – Intuitive knowledge extraction:** The MRTs have deep knowledge about medical, epidemiological and social-demographic interrelations, which are important for the aggregation. However, they have little or no knowledge about a possible technical realisation, predicate logic or the definition of business rules. Therefore, they favoured a way of knowledge extraction, which blends coherently into their work environment. Still they demanded that the process of knowledge extraction has to be traceable and reproducible and that the effect of a certain rule is displayed.

**R21 – Abstraction of discrete ranges of value:** In the field study it became clear, that different kinds of knowledge were applied. In some cases the MRTs commented that their decision in a specific situation is based upon an ascertain characteristic of the visualised data (e.g. because a variable in two records had a distinct named value). In other cases they expressed more complex rules, for instance if a set of variables in both records is equal, regardless of the distinct values. This lead to the conclusion that there is knowledge which abstracts from the actual values and that the system needs to support to extract knowledge not only based upon actual values but also based upon abstractions, for instance based upon existing encoding tables which group value ranges.

**R22 – Extensibility of the knowledge domain:** In the interview phase it was pointed out, that, although the fields relevant for the analysis task can be named for the current data model, the variables that are relevant in the aggregation task may change in future when new fields are added to the records. Thus, the methods for knowledge extraction must adapt to changes in the data model.

**R23 – Complex interrelationships and functional dependencies:** In the field study it was observed, that the aggregation of records is not only done by the pair-wise comparison of attributes. Instead, complex interrelationships of values and functional dependencies that lead to an aggregation could be identified. As an example there are different record types (records from pathologists, death certificate only records and others), which influence the possibilities of aggregation. For instance, a death certificate message cannot be aggregated with a message of a living patient that was collected later than the death certificate. Therefore, it can be reasoned that specific knowledge influences the priority and validity of other the applicable knowledge. Different scenarios could be identified, e.g. where for two messages a certain field has to be equal/unequal to allow the aggregation based upon the comparison of another field and many others.

**R24 – Definition of best-of values:** When records are aggregated the best-of values are being calculated. There can be complex interrelationships, which enable the MRTs to define higher quality values than those present in the records. For instance best-of values might be derived based upon pre-defined value ranges (e.g. official guidelines for medical documentation). It was mentioned, that depending on the value it might be necessary to either allow the users to express such a value manually or to choose a value from the existing values present in a set of records.

**R25 – Context sensitive knowledge:** It was pointed out by the MRTs, that there is context sensitive knowledge, which can only be applied in certain situations. For example, certain knowledge might only be applicable for specific message types (comparable to the death certificate only example in R5). Another example, where the validity of knowledge changes according to the context are rules of thumb for data quality depending on previous experience with certain data suppliers. Here it was mentioned, that data from some suppliers has to undergo a more thorough check, whereas an aggregation may be performed automatically for other suppliers. This means, that the validity of knowledge changes according to the value of certain fields in a present set of records.

**R26 – Conflict and priority management:** The MRTs mentioned during the interviews and alongside the study, that conflicting knowledge might exist. Therefore, means to resolve conflicts have to be implemented. An idea of the MD was to integrate a priority mechanism, which will allow the experts to mark a priority for knowledge application. Ideas for a possible solution were either an a priori definition of priorities of what the experts called chunks of knowledge or a manual conflict resolving mechanism.

**R27 – Knowledge management and traceability:** As mentioned above the users knowledge is subject to an ageing process. There are two reasons for this: changes in encoding guidelines and guidelines for medical documentation and changes in the data format. With a method for automatic knowledge derivation, the knowledge-base

should in theory adapt to these changes automatically. It can be possible though, that existing knowledge might conflict with newly derived knowledge. Therefore it is necessary that the users can manually manage the knowledge-base e.g. to mark outdated knowledge. Here it was noted that neither the MD nor the MRTs who define the knowledge are experts in IT technology and therefore the methods for knowledge management have to be easy accessible and intuitive. It was explicitly said that a selection of knowledge by example data was favoured, where all knowledge that has an effect on this exemplary data can be accessed. Above it was demanded, to identify data that is affected by possible changes in the knowledge. This results in the requirement for traceability of the knowledge application. Hence, a mechanism to retrieve a trace of knowledge application has to be integrated.

**R28 – Knowledge quality management:** During the initial interviews the MD mentioned that a fully automatic knowledge derivation might be unfortunate because it cannot be guaranteed that knowledge might be derived falsely. A reason for this could be simply that knowledge derived on a subset of the data might be correct for this specific set of records but may have unforeseen influence on other records or the existing data base. Therefore, a method for a downstream knowledge quality management was requested, to project the changes that newly derived knowledge will cause. This should happen either on the existing database or on a specific set of comprehensive test database. In this, a user should be able to view the projected changes and then validate or invalidate the derived knowledge.

### Traceability and knowledge sharing

Whenever new records arrive they are matched against the existing database. Therefore newly arrived records can change the analysis situation even of previous aggregation, which makes it necessary to review previous aggregations. As an example, if a newly arrived record matches with some existing records that have previously been aggregated with some uncertainty, then it is possible that this previous aggregation has to be split and new aggregations have to be forged based upon the advanced information that is available. Use cases like this are not supported in the existing software. In the interviews with the MRTs, during the field study and in the interview with the MD a number of requirements to support such use cases were identified.

**R29 – Annotation of knowledge:** Evolutionary changes in the database will lead to shifts in the relevance of certain parts of the derived knowledge and hence to shifts in the validity of the knowledge. This leads to the problem, that the reason why specific parts of the knowledge have been extracted might become ambiguous. Therefore the MD requested a feature to semantically annotate the extracted knowledge, which allows the MRTs who defined the knowledge to express the thoughts that lead to the extraction of certain parts of the knowledge, in order to be able to evaluate the knowledge-base in future. In addition to that it was requested that the data sets that lead to the extraction of certain knowledge are stored alongside the knowledge in

conjunction with the user who extracted the knowledge, in order to allow further traceability of the process of knowledge derivation.

**R30 – Traceability of aggregation steps:** Due to the fact that the aggregation is performed in several steps over a possibly longer time frame it is necessary to ensure a traceability of these aggregation steps. In the existing software only the most current snapshot of the aggregation process can be seen and no historical information about the evolution of a dataset is stored. The MD noted that access to the trace of the evolution of the datasets is advantageous. Specifically it was demanded to trace which records were aggregated (automatically or manually) at which time and – in the case of an automatic aggregation – which knowledge lead to the decision for this aggregation.

**R31 – Knowledge sharing:** Although the MRTs work individually it could be observed that there are certain use cases in which they collaborate. If the MRTs are uncertain about whether records should be aggregated or which best of values to derive, they will ask their colleagues for advise. In the current setup to deal with this the MRTs will write down the IDs of the transitive closures or the IDs of specific records on a sheet of paper alongside with annotations about their problems and then hand these notes to their colleagues so they can manually select the closure from the data base and process it. The MRTs wished for a functionality to support these use cases, so they can annotate transitive closures and hand them over to their colleagues. Even, if necessary, to a specific colleague who specialised in certain problems. Furthermore it was demanded that others are able to reproduce decisions regarding the aggregation or the best of derivation in terms of quality management. This is comparable to the traceability of aggregation steps in R16 enhanced by the traceability of best of derivation and the traceability of incomplete processed aggregations.

**R32 – Ad-hoc interactive modification of aggregations:** As mentioned in R16 the current software only supports to visualise the latest state of the aggregation, even an evolutionary process modifies the datasets over time. It can occur that a record, which was previously aggregated with some other records based upon the data that was available at the time of the aggregation should rather be aggregated with another newly arrived record at a later point in time. In the current system for this the whole aggregation has to be split up and all aggregation steps are made withdrawn. The MD and the MRTs requested a feature that allows them to only withdraw single steps while keeping the rest of the closure intact. For this single patients or tumors have to be split out of an aggregated set in order to be aggregated with other messages. If changes like this occur, the derived and saved best of values have to be made invalid, if necessary, as the formally aggregated records might have added to these best of values.



### 3.3 Implications for the Design of knowledge-based VA Systems

The requirements analysis in both scenarios has shown that VA systems have to support a broad selection of features. Similarities in the requirements can be identified across both scenarios. In a closer investigation, these requirements can be separated into two groups requirements that solely concerning the specific analysis application or the application domain and requirements with a broader validity. Requirements of the first group can be further grouped into three categories: the visual interface of the system (R16, R17, R18), the visual representation of the data (R1, R5, R6, R7) and the interaction with the system (R10, R15, R18). Some parts of the requirements that fall into one of these categories can be carried out as general requirements in VA systems.

The categories in the second group are subsuming requirements concerning the analysis process, knowledge extraction and management, and knowledge application and sharing. In these categories several requirements can be identified, which contain elements that are valid for VA systems independent from the application domain. Many of which can be identified in both scenarios or where at least similarities to requirements across the scenarios can be identified.

Requirements		Design Implication
Scenario 1	Scenario 2	
Visual interface		
–	R16 – R18	–
Visual representation of the data		
R1, R5 – R7	–	–
Interaction		
R10	R15, R18	–
Analysis process		
R1, R2, R5 – R8 R11, R13	R15, R17, R18 R31, R33	DI 1 – Analysis Process DI 2 – Analysis Traceability
Knowledge Extraction and Management		
R13, R14	R19–R24, R26–R28, R30	DI 3 – Knowledge Extraction
Knowledge Application and Sharing		
R2–R4, R9, R11–R14	R25, R29–R31	DI 4 – Knowledge Application

*Table 3.1: Categorisation of Requirements and Design Implications.*

Table 3.3 provides an overview over this categorisation. The six categories are listed from top to bottom. For the first three categories, no design implications are identified, as they are out of the scope of the reference architecture.

For the next three categories four design implications (DI1 – DI4) are identified. The table also shows the relationship between the requirements in the specific scenario and the design implication that is based upon the requirements. In the following sections, the relationship between the requirements and the design implications in the last three

categories will be discussed in detail. For this discussion, the requirements are examined for similarities and general objectives, which are then translated into implications for the design of knowledge-based VA systems.

### 3.3.1 Analysis Process

Visual data exploration tasks follow an iterative process in which expert users pursue their analysis objective (compare chapter 2). The analysis process hereby consists out of several, typically reoccurring steps, which eventually lead to insight. This is reflected by requirements such as fast switching between heterogeneous views (R2) and switch between abstraction levels (R8), where each view/abstraction level reflects a step in the analysis process. Some requirements even contained the precise demand for a structured process, such as (R10) from the first scenario, where a smart pipelining and filtering function as analogy to what is known in command line pipe lining or video editing was requested. Similar requests were made in the second scenario, where in (R15) it was noted that there is a pipelined order in which the records are processed. This is also reflected in the general description of the second scenario, where the analysts will aggregate records in several steps and by the fact that within the same analysis session the MRTs pursue different tasks (aggregation of patients, aggregation of tumours and best of derivation).

Within the analysis, process data needs to be examined in different levels of detail and abstraction, encompassing rich and meaningful visualisations. The requirement for the integration of heterogeneous visualisation tools (R1), the demand for different visualisation methods (R5, R6, R7) in the first scenario and the demand for advanced highlighting of important information (R17), for information reduction and filtering (R15) for adjustable probabilistic distance (R18), which effects the level of detail of the visualised data reflect this across both scenarios. Summarised this leads to the following design implication:

**Design Implication 1 – Analysis Process:** VA systems have to support the analysis process. For this, next to meaningful visual abstractions and representations, VA systems need to provide a tangible and structured support of the analysis process, in which the analyst is aware and in control of multiple reoccurring and varying analysis steps.

The iterative nature of visual data exploration tasks leads to possibly lengthy and complex analysis sessions. Several requirements show that the overview respectively the loss of overview is an important concern for the analysts. In (R11) for instance specific support for traceability of the course of the analysis in form of a visual history was demanded. This also relates to the requirement for traceability and reproducibility (R13) where an overview function was demanded, that allows to reason about the state of the analysis and the steps taken thus far. Similar functionality was requested in the second scenario. In this scenario, the visualised data is modified in various aggregation steps. Specifically it was demanded to trace which records were aggregated at which time by (R31) and functionality that allows an ad-hoc interactive modification of aggre-

gations. Thus, a functionality that lets the experts work directly on the visualised course of the analysis to change specific parts e.g. when existing circumstances have to be re-evaluated based upon superior information or newly arrived data (R33). This demand for traceability and direct manipulation of the analysis trace leads to the following design implication:

**Design Implication 2 – Analysis Traceability:** VA systems have to support a traceability of the analysis process, allowing experts to reason which steps they have taken. Hereby it is not only necessary to provide a trace of the course of the analysis in an accounting way. Rather a support for controlling mechanisms, which allow modifications of the analysis process (e.g. modifications of former analysis steps), is necessary.

### 3.3.2 Knowledge Extraction and Management

Following the VA process, (see section 2.1.2) the steps of the analysis process, whether they are automated or manual, eventually lead to insight and the creation knowledge, which is then re-applied in order to refine the analysis. During the requirements analysis the users described several scenarios of knowledge extraction across both scenarios.

In the first scenario, the engineers asked for methods to save the course of the analysis in order to preserve the insight gained in a session (R13). Moreover, they specifically asked for a function to extract findings by specifying properties of reoccurring patterns (R14). Hence a function to specify and extract the knowledge they applied during the analysis.

One major challenge for the second scenario is such an extraction of knowledge in order to optimise automatic analysis steps. Therefore very specific requirements even according to the embodiment of the knowledge extraction appeared during the interviews and alongside the field study. Most interestingly, the MRTs required the knowledge extraction to take place continuously alongside the routine analysis process directly on the visual interface (R19) in an intuitive way that will let them express their knowledge without a deeper understanding of the technical structure of the underlying knowledge-base (R20). The field study revealed that certain knowledge can be abstracted from the actual displayed values (R21) and that complex interrelationships exist within the knowledge (R23) and hence not only simple value by value comparisons contribute to the analysis.

Furthermore it was requested, that the knowledge domain can be extended (R22) if the composition of the records changes in future. Lastly, it became clear, that, even though the primary objective of the analysis is the aggregation of matching entities, there is another task, the derivation of best of values (R24) which is performed alongside the aggregation, yet encompasses entirely different knowledge, which needs to be extracted. These examples illustrate, that within an analysis task there is a multitude of circumstances in which an extraction of knowledge is supposed to be invaluable by the users. This includes procedural knowledge as well as contextual knowledge.

The demand for knowledge extraction automatically leads to the need for methods of

knowledge management. Once extracted it is necessary, that the knowledge is accessible for the experts. According to the requirements, this can be in order to re-use it (R13, R14) or to manage it. In the first scenario, the engineers asked for a method to extract their findings in a way that allows them to use these findings in other analysis situations (R14). This means they asked for means to generalise and modify the knowledge, they extracted. In addition to this in both scenarios, the experts asked for means to trace the origin of knowledge (R13, R27). A traceability of the origin was also requested with regard to evolutionary changes of the data-format or the knowledge (R29) and the evolution of the database itself (R30).

The MRTs in the second scenario noted, that they need an overview functions that allows them to judge how data is affected by application of extracted knowledge (R27). In the second scenario a major concern of the experts was that there might be conflicting knowledge, as several experts derive the knowledge concurrently and therefore methods for conflict and priority management are needed (R26) as well as methods for quality management (R28). Again, overview functions were demanded, comparable to (R27).

In summary it can be reasoned, that VA systems that support the extraction of expert knowledge also have to provide methods to manage the extracted knowledge. However, in the scenarios given the requirements for knowledge management were merely consequences of domain specific problems. Hence, apart from claiming that the integrated knowledge model may not make the knowledge management impossible. Hence, the following design implication can be formulated:

**Design Implication 3 – Knowledge Extraction:** VA systems have to represent the knowledge that is applied by the analyst in order to be able to describe the applied knowledge. Thereby it is inevitable that a rich set of methods is provided to allow the experts to mark and extract valuable knowledge into a knowledge-base, to use it in deviant analysis situations and that the knowledge in the knowledge-base is generalisable and therefore applicable in other analysis situations.

### 3.3.3 Knowledge Application and Sharing

According to [KMS<sup>+</sup>08] collaboration, presentation and dissemination play a key role in VA systems. In the requirement analysis this was expressed by the demand for the support of collaboration and knowledge sharing (R11-R14) in the first scenario and similar requirements for knowledge sharing by annotation of aggregation steps in the second scenario (R31). Above this several other requirements could be identified that deal with the task of re-application of knowledge and sharing of knowledge with others. In the requirements for transition between individual and group work (R4) and the requirement for a direct mapping between heterogeneous views (R3) and the fast switching between heterogeneous views (R2) is described how knowledge that is applied to the system in a specific state needs to be transferred either to other views or to other experts either by direct interaction or by asynchronous transfer, where one expert saves findings which then can be viewed by other experts. Similar requirements could be identified in the second scenario, where the users demanded for a traceability of the knowledge applied

by others (R30) and methods to annotate knowledge and analysis steps (R29, R31) in order to enable a knowledge transfer.

Matching the main objective in the second task: to extract knowledge in order to use it to enhance automatic analysis steps in future analysis situations, requirements for the knowledge extraction were found in the first scenario. In R9 and R14 the engineers requested mechanisms for sorting and filtering, which they customise based upon patterns in the data, so that they can apply the filters they defined in one scenario to other scenarios. This means that they want to change the appearance of the analysis system based upon the data that is displayed. This is comparable to the requirement for context sensitive knowledge applications (R25) where the MRTs pointed out that based upon the data that is displayed the validity of the knowledge varies.

It becomes clear that there are many ways in which previously extracted or applied knowledge is shared or re-applied in VA systems. Knowledge application and sharing can occur either directly, e.g. in collaborative work with colleagues, or indirectly by automated analysis steps or by manual selection of knowledge to be supplied. All of these ways have in common, that the current state of the analysis system defines whether the application of knowledge is feasible at in a given analysis situation. Based upon the similarities identified in the two scenarios the following design implication can be derived.

**Design Implication 4 – Knowledge Application:** VA systems have to provide methods to re-apply and share knowledge that was previously extracted or that was applied in a previous analysis step. For this, it is vital to identify whether the knowledge actually is applicable in a given situation.

### 3.4 Summary

In this chapter general requirements to knowledge-based VA systems have been examined. For this initially in section 3.1 requirements for the first scenario were presented. Two different types of target users contributed to the requirement analysis, a group manager and test engineers. The requirement analysis itself was divided into two phases. The first phase encompassed telephone interviews and a web conference preceding the development process, in which an initial set of requirements was collected. These initial requirements were then refined in an agile development process and regular review sessions with the group manager and the test engineers. Finally the identified requirements were summarised.

Subsequently in section 3.2 the requirements for the second scenario were examined. Three different user roles took part in the process: the medical director, a number of medical records technicians and data analysts, although the first two contributed the most to the requirements analysis. The process of the analysis collection also varied slightly compared to the previous scenario. The requirements analysis started with initial interviews. The results of these interviews were edited and evaluated and then the set of collected requirements was presented to all of the medical records technicians. Finally, a field study, where the medical records technicians were observed while they were performing their daily analysis tasks completed the requirements collection.

In section 3.3 the requirements collected in both scenarios were analysed according to similarities. For this, the requirements were grouped into three categories: analysis process, knowledge extraction and management and knowledge application and sharing. The similarities were identified and based upon this four general design implications for knowledge-based VA systems were derived. These serve as the basis for the design of a reference architecture for knowledge-based VA in the following chapter.

---

## 4 The KnoVA Taxonomy

The objective of this chapter is to provide an overview of the design space of knowledge-based VA, leading to a taxonomy for knowledge-based VA systems, the knowledge-based visual analytics taxonomy or KnoVA taxonomy.

In section 4.1 firstly the methodology to derive the taxonomy is introduced. After this outline in section 4.2 descriptive properties of VA systems that are relevant in the scope of this thesis are examined. Based upon the identified properties the KnoVA Taxonomy is created and introduced in section 4.3. Finally section 4.4 summarises this chapter.

### 4.1 Methodology and Outline

Different approaches to explore and understand the design space of VA systems exist. As such, the VDET stands out as a recent approach where the exploratory character of the VA process is taken into account.

In an active field of research, such as VA, a closure in the examination of the design space cannot be achieved. Examples of classifying properties that are not present in the VDET can be found. In addition, the classifying properties of the VDET aim to be generic rather than specific. It only identifies six different properties for data to be visualised and does not dive into the description of complex data structures.

CARELIS, for instance, as introduced in the motivation, visualises multi-dimensional data of various types with complex functional dependencies within the data. In the VDET, this would simply be classified as multi-dimensional. Respectively the VDET is not detailed enough to describe all aspects of this VA system. Therefore, in this chapter, the VDET is extended to the KnoVA taxonomy with the objective to describe the design space of VA systems in enough detail, to serve as the foundation of a meta data model for VA systems.

There are two different approaches to derive classifying properties: a top-down approach and a bottom-up approach. The top-down approach is to collect the requirements for a specific analysis application to be implemented and then derive the classifying properties from these requirements. This approach is problematic, as the resulting model will only cover the requirements for the specific analysis application. This results in the risk that the meta data model will be limited to analysis applications which share the same set of requirements.

In the bottom-up approach a number of existing analysis applications and visualisation methods are examined in order to derive the classifying properties. This approach has the advantage, that a more generic set of classifying properties can be identified. Therefore, to create the KnoVA meta model the bottom-up approach is applied.

For this approach it is firstly necessary to identify, examine and evaluate the properties of existing VA systems, to then extract the classifying properties and classes that form a taxonomy. The next two sections are structured based upon this premise.

## 4.2 Properties of VA Systems

In this section, existing VA systems are analysed in order to identify properties that can be used to classify VA systems. At first, seven existing VA systems that are relevant in the scope of knowledge-based VA are examined. For each system it is explained why it is included in the examination before the system is described in detail.

In the subsections 4.2.1 – 4.2.3 three systems from own previous work in the field of VA are examined: The VAT System, the TaP System and the 3D-Cube system.

This is followed by an examination of the Multi-dimensional Statistical Data Analysis Engine (Mustang) in section 4.2.4. After this examination in subsections 4.2.5 – 4.2.7 selected related work is examined to complete the analysis.

The balance point in the examination lies on VA systems that either support a structured VA process, systems suitable for the analysis of multi-dimensional data, systems with a broad application spectrum or systems providing an integrated knowledge support. Also VA systems that share the application domain with one the motivating scenarios are included.

### 4.2.1 TaP

TaP<sup>1</sup> is a VA system for the analysis of multi-dimensional data on a large scale multi-touch surface computer [FH10, FH09, FHTA09].

TaP was initially developed in cooperation with the EKN. Thus it shares the application domain with the second motivating scenario that was introduced in section 1.1. The examination of TaP therefore promises to provide properties relevant for that application domain. Another reason why TaP is included in the examination is because despite of its initial purpose, TaP can be seen as a system for general analysis as it can be used in a wide range of application scenarios because the underlying database supports the multi-dimensional data model.

**Term** (multi-dimensional data model). *The **Multi-Dimensional Data Model**, also known as OLAP (online analytical processing) model, is a specialised data model that is used in data warehousing to enable ad-hoc execution of complex analytical queries [CCS93]. The core element of the OLAP model are n-dimensional OLAP cubes, which are a representation of the data that specifically supports ad-hoc queries. In many databases a number of dimensions are identifying a single entity. For instance, a cancer record in the EKN might be identified by time of the diagnosis, the treatment or the time of death or residency of the patient.*

*OLAP cubes can be thought of as a multi-dimensional array of values from different data dimensions. Each axis of the cube is assigned with a data dimension. Each cell of the cube represents a numeric fact, called measure that can be categorised by the dimensions of the cube. The OLAP model provides a number of different operators for the exploration of the data. The drill-down operator, for instance, allows users to view the data on a finer level of detail alongside hierarchically structured dimensions*

<sup>1</sup> A video demonstration of TaP can be found here [http://www.youtube.com/watch?v=9glwdkNto\\_I](http://www.youtube.com/watch?v=9glwdkNto_I)



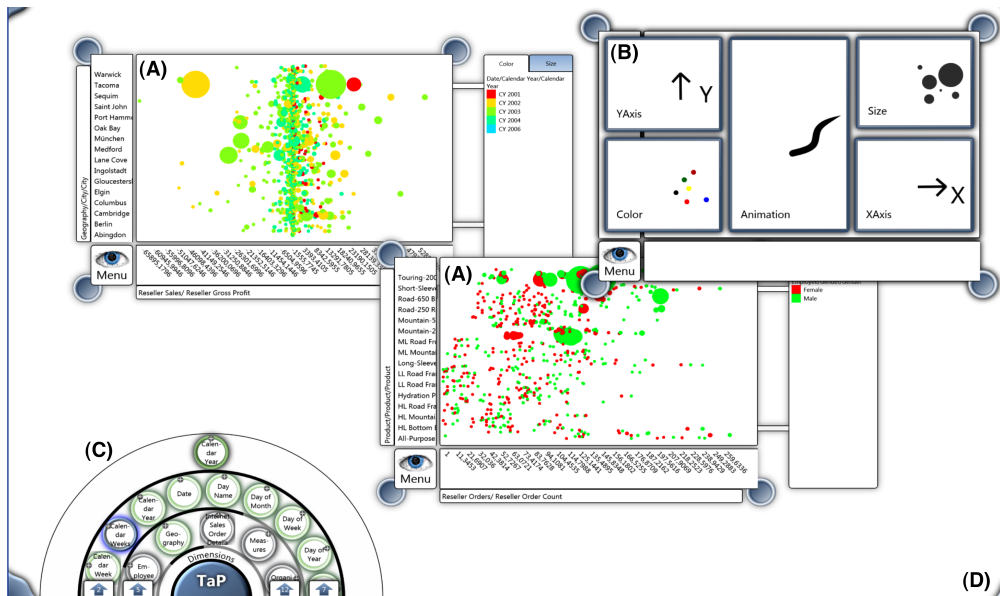


Figure 4.1: Screenshot of the TaP System.

[Tho97]. As an example, the time dimension often can be drilled-down from years to month. Such a drill-down will result in a replacement of every yearly cell with twelve new cells, one for each month in a year, each representing the measure for a month instead of a year.

A screenshot of the TaP system is shown in figure 4.1. The interface is presented to the user on a surface computer that has an interactive screen of 1300x900mm with a resolution of 1280x800 pixels. A picture of the surface computer with two analysts is shown in figure 6.2. A major design goal was to optimise the analysis system and all elements of the user interface for touch gesture based interaction. The elements of the user interface are described below:

**(A) Bubble Chart:** The bubble chart is an extension of a classical scatter plot (see figure 4.1-A). In the TaP system, the bubble chart has five axis of visualisation: x-axis and y-axis, colour axis, size axis, and animation. The x-axis and y-axis in combination display data in a cartesian coordinate system, similar to a classical scatter plot. The colour axis displays a third measure in the same coordinate system, for values that share a dimension with data that is already displayed on the x-axis or y-axis and, hence, can be correlated. The values are translated into a colour map and then the corresponding dots in the chart are tinted in the colour that matches to their value in the additional dimension. The size axis works likewise, only that the values of yet another additional dimension are translated into a diameter for each of the dots. Thus, the size of the dots varies according to the value of the underlying data. This can be seen in both of the exemplary charts in figure 4.1.

The animation axis in the chart is a speciality. On this axis, only time-variant data can be assigned that shares one dimension with the data that is already visualised in the chart. Then, an animation will be created which shows the changes of the data in the chart over time. At runtime, there can be multiple (unlimited) charts at a time. The charts can be freely positioned across the table, much like windows on a desktop computer.

**(C) Stacked half-pie menu:** The stacked half-pie menu is a specialised hierarchical menu, optimised for the usage on surface computers [HFS09]. It is used to explore deep hierarchical structures while consuming a minimal screen space. To explore hierarchical levels, concentric rings are opened around the centre of the menu. A new ring is opened for each level in the hierarchy. By clicking on an element with children a new ring will pop out on the outer border of the menu. In order to save screen space, and keep moving distances small, the inner rings will collapse as soon as more than four levels are to be displayed. Only the three most recent hierarchical levels are shown in the menu. The elements within the ring have a fixed size, large enough to be easily operated by touch interaction. If the number of elements to be displayed in a hierarchical level exceeds the space in a ring, additional elements are hidden. The hidden elements can be accessed by a tap and drag gesture. The interaction is comparable to the operation used to dial on old-fashioned telephones. In the TaP system, the user can drag each of the elements out of a ring and drop it onto one of the drop zones of the chart, thus assigning this element to the correspondent visualisation axis.

The stacked half-pie menu offers access to the hierarchical data in the data cubes which are modelled in the Analysis Server database. OLAP databases offer operations to explore the data. These operations are called OLAP operations. In the TaP system, the OLAP operations are triggered by gestures on the chart. Thus, the user is directly manipulating the data, following Shneiderman's direct manipulation model [Shn83]. There are four different gestures to trigger OLAP operations. If two fingers in the chart are touched down and moved away from each other (spread gesture), a drill-down operation is triggered, if possible. If the fingers are moved towards each other (pinch gesture), a roll-up operation is triggered. If the spread gesture is made on an element of the scale (the textual identifiers on the side of the chart) is performed, the displayed data will be limited to the element under which the gesture was made. This interaction represents a slice operation in the OLAP nomenclature. If the pinch gesture is performed on the scale, the last slice operation will be reversed.

**(B) Drop Zones:** The axis in the chart can be assigned either with measures or dimensions. To perform this assignment, the user can choose either a dimension or a measure in the stacked half-pie menu (C) and drag it onto the five drop zones, shown in figure 4.1-B. The drop zones are a special feature to allow a touch only interaction with the analysis system. They correspond to the five axis of visualisation which are offered by the bubble chart. The assignment of values is done by dragging elements from the menu and drop these elements over the drop zone which corresponds to the axis that this elements should get assigned to. Build in rules verify that only valid assignments can be made.

**(D) Active Corners:** On each corner of the surface computer a blue touchable zone provides access to an input layer. This input layer gives access to further functionality

of the system. The user interacts with this layer by drawing path-based gestures.

## Conclusion

Although limited to a single visualisation method the TaP system already includes a range of different properties that are relevant for the KnoVA taxonomy. TaP is suitable for the analysis of data with different characteristics, ranging from *one-dimensional* in the simplest form over *two-dimensional* to *multi-dimensional* data with many attributes. The structure of the data that TaP can visualise is always *hierarchical* due to the underlying OLAP database. The integrated OLAP database provides data of various orders. TaP can visualise *qualitative*, *ordinal*, *quantitative*, *timely* and *categorical* data.

TaP only provides a single visualisation method, the multi-dimensional bubble chart. The visualisation technique of this bubble chart is a *standard projection* of the data to the two-dimensional screen. The stack half-pie menu in TaP is a hierarchical menu to access the data. Yet it visualises the structure of the data in a *stacked display* and therefore uses a visualisation technique.

There are many way to interact with the TaP system. Some interactions with the visual interface lead to changes in the visual interface but do not modify the underlying data. *Panning and scrolling* in the bubble chart as well as and *optical zoom* in this visualisation are examples for such interaction techniques. Other interactions lead to a change in the section of the visualised data. TaP maps gesture exploration techniques to operations of the underlying OLAP database. An *explorative zoom* on one of the axis, initiated by a spread or pinch gesture, performs the OLAP operations roll-up and drill-down. TaP also provides techniques to perform *selections*. Gesture based interaction on single values of the axis will perform slice operations and of course the stacked half-pie menu directly *selects* a specific element of the data source. Another feature of the TaP system that can be used to compare it with other VA systems is the support for *discrete* animations. This integration of a dynamic representation differentiates TaP from other VA systems.

### 4.2.2 VAT System

The Visual Analytics Transformation (VAT) system ([Uph10, FA11]) is a system for the interactive VA of multi-dimensional data. It was developed in co-operation with the EKN and shares the application domain with the second motivating scenario. It is based upon the TaP system and has the intention to integrate multiple visualisation methods and various data sources in one streamlined analysis process. In addition to this intention, it includes a model to abstract between different elements in the system.

Figure 4.2 shows a screenshot of the TAP system. In the screenshot, an analysis path can be seen. In the VAT system, the user defines such an analysis path by drag-and-drop operations. A menu gives access to the different system elements which are available in the VAT system (data sources, visualisation methods, etc.).

The interaction with the VAT system in [Uph10] is described as follows: Initially, the user starts with an empty workspace. Then he chooses the system elements he needs in the analysis and places them on the workspace via drag-and-drop interaction. In a first

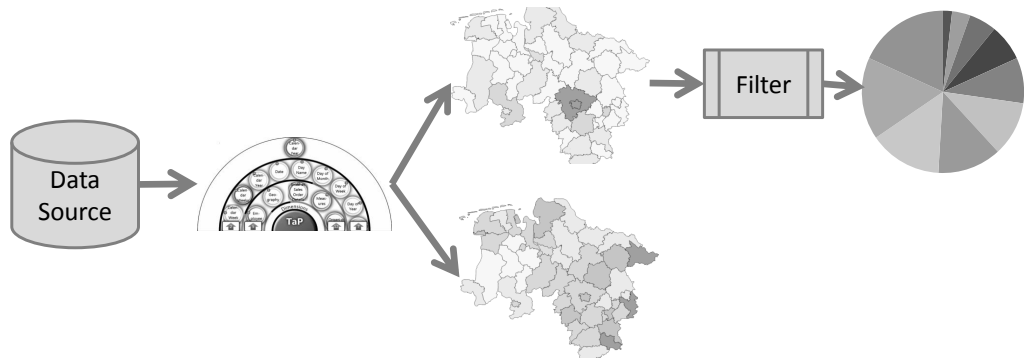


Figure 4.2: Exemplary Analysis Path to illustrate the VA Process.

step, he might select a data source. The next step in the exemplary path is the stacked half-pie menu as selection menu to explore the data source. The menu can be dragged into the workspace. In the VAT system, it is then necessary to connect the data source to the selection menu by drawing a path between the two elements.

The analysis paths are a speciality of the VAT system. The user can change the composition of the paths at any given point. He could, for instance, add another selection menu to the data source and thus create a branch in the path. Another example where branches like this occur is, when several visualisation methods are connected to the same predecessor. This is shown in figure 4.2 with the two different thematic maps that both follow the stacked half-pie menu.

The concept of analysis paths adds a great amount of flexibility to the VAT system. This flexibility is achieved by an integrated abstraction layer that uses model-driven software development (MDS) techniques to integrate new system elements into the system, as mentioned in [TF08]. The layer provides a standardised description for visualisation methods, interaction methods, and data sources. When new system elements are integrated into the system, a mapping between this domain specific model and the corresponding properties of the new system element is defined. The code to integrate the new component is automatically generated by the VMTS [LLMC05, MMC09] MDS framework. At runtime, an instance of the abstract model is used to create a mapping between the different system models. Therefore, it is possible to create analysis paths between system elements, whenever their models match at least partly.

The intention of the system according to [Uph10] is to quickly explore large epidemiological data sets. The system, therefore, provides a number of pre-configured system elements: an OLAP database (Microsoft Analysis Server), the stacked half-pie menu as selection menu to explore the database, a quantitative filter where upper and lower bounds or ranges for values can be defined, a two-dimensional x-y scatterplot, a bubble chart, a thematic map, and several standard charts such as bar-charts and pie-charts. The bubble chart in the VAT system is comparable to the one shown in figure 4.1. It offers four axis of visualisation: the x-axis, the y-axis, a colour axis, and a size axis. In contrast

to the TaP system, the bubble chart in the VAT system has no support for the animation axis. The thematic map is based upon a geographic information system (GIS) and allows displaying geo-spatial values that are important in epidemiological analysis.

The VAT system was developed in co-operation with the EKN to analyse the data collected in the registry. The integrated domain specific model (DSM) not only allows the integration of various system elements but also the abstraction from user interaction. In the DSM, operations are defined which have a certain effect on the VA system, as for instance, the OLAP operations.

## Conclusion

VAT shares many properties with the TaP system on which it is based. For instance, it shares the support for data with *one-dimensional*, *two-dimensional*, and *multi-dimensional* characteristics as well as the support for data with a *hierarchical* structure. Due to the abstraction layer VAT supports the integration of various data sources and of components. With integrated statistical components VAT provides additional support for *algebraic* and *complex* data structures.

Also inherited from TaP is the support of data with *qualitative*, *ordinal*, *quantitative*, *timely*, and *categorical* order.

VAT provides a number of different visualisation techniques. The bubble chart as *standard two-dimensional projection*, other standard charts and a *geo-spatial* thematic map. The user can explore the data by interaction with the visual system. In addition to the gesture based *selection* and the *explorative zoom* of the bubble chart the user can also define *filters* to modify the data.

### 4.2.3 3D-Cube

The 3D-Cube system [Kra10] is, like the previous two presented tools, a system for the interactive visual exploration of data in the OLAP data model. A screenshot of the 3D-Cube system can be seen in figure 4.3. It shares the application domain with the second motivating scenario and therefore is included in this examination.

The 3D-Cube system was developed based upon the idea to directly visualise the cells of OLAP cubes. The basic idea of the 3D-Cube system is to visualise each cell in an OLAP cube as a three-dimensional cube on an interactive user interface. The result can be seen figure 4.3. There the measures are mapped to an adjustable colour map and each cell is colourised according to this mapping.

The visualisation gives a direct qualitative feedback of the underlying data without the creation of more complex visual mappings. It addresses the problem that, in order to approach large, unexplored databases, the analysts have to get an overview first to find ranges of the data which are worth a closer look. The goal of the 3D-Cube system is to generate queries which allow to reduce the amount of data to be analysed further. The 3D-Cube system can be seen as a visually enhanced selection menu for data in the OLAP data model. In contrast to classical selection menus such as tree views or the stacked half-pie menu presented above, it provides a direct qualitative feedback of the

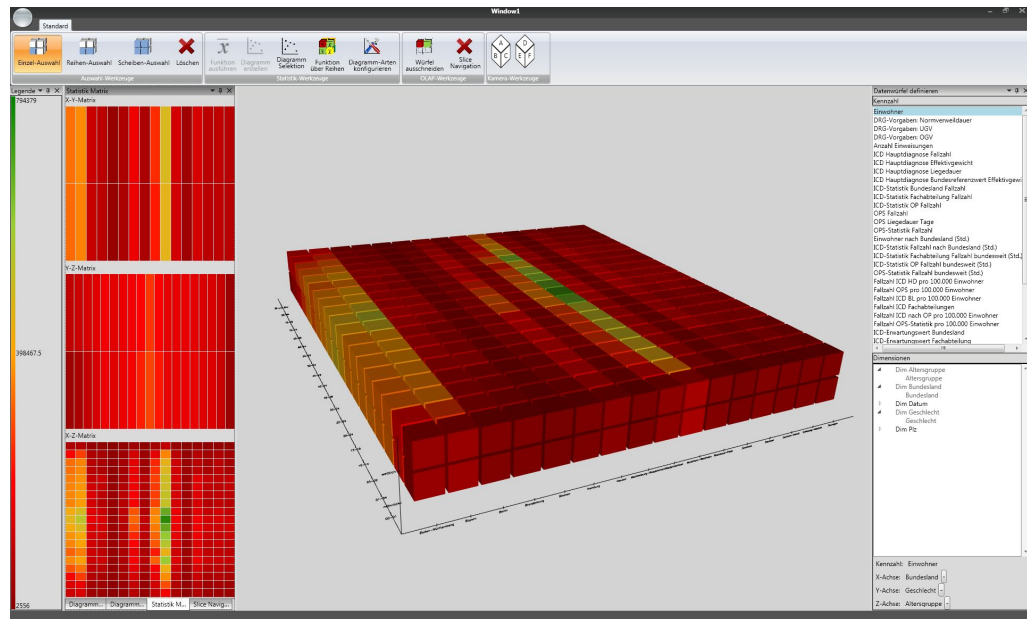


Figure 4.3: Screenshot of the 3D-Cube System.

underlying data.

In figure 4.3 three basic elements can be seen. The most prominent in the middle is the visualised 3D-Cube itself. On the right side, a list view is displayed, providing access to the measures and dimensions of the underlying data. From the list view the user can select dimensions and measures to be displayed in the 3D-Cube visualisation by a drag-and-drop interaction. Dimensions and measures can be quickly replaced in the view to enable the user to quickly browse through large amounts of data.

On the left side, additional two-dimensional views can be seen. Due to the direct 3D representation values in the 3D-Cube are easily occluded by other values in the viewing direction. A number of interaction methods are integrated in the 3D-Cube to reason about occluded values. These interaction methods allow the selection of a single cell or a number of cells (e.g., a complete row or complete slice of the cube). The items selected this way, can then be displayed in the additional views on the left side. Therefore, these can be seen as detailed projections of a selected areas of the cube. Also a sub-cube (representing the OLAP dice operation) can be selected directly on the visual interface. These additional representations can be defined in a rule-based system and can be enriched with additional operations. For instance a projection could define a statistical measure which is calculated for a certain selection of cells. Hence, the elements displayed in the additional views not only represent cells from the 3D-Cube. Instead a single cell in one of the additional views can represent the aggregated values of a number of cells from the cube. Therefore, a bi-directional brushing link between the additional views and the cube is established. Highlighting of a cell or a number of cells on the ad-

ditional view will result in a highlighting of the original cells in the 3D-Cube which are represented by the additional view. The integrated statistical functionality is provided by a statistical software tool (R-Project).

In addition, the 3D-Cube allows a direct exploration in the three-dimensional space, the analyst can rotate and optically zoom in to the view. Specialised user interface elements are integrated for a quick navigation in the three-dimensional space.

When the user has finished the exploration of the database with the 3D-Cube system, queries can be generated which represent the course of the iterative selection and refinement process. These queries can then be used in other applications for a detailed analysis of the selected data. The visual exploration in the 3D-Cube software is limited to three data dimensions at once due to the direct spatial mapping between the data dimensions and the visualised 3D cube. However, by user interaction and the definition of statistical or mathematical functions which map two dimensions to a new artificial dimension it is possible to explorer measures with more than three data dimensions.

## Conclusion

3D-Cube is based upon an OLAP database and therefore like TaP and VAT features *one-dimensional* and *multi-dimensional* data of *hierachical* structure. 3D-Cube also integrates the statistical software R-Project and thus supports *algebraic* data structures with *qualitative*, *ordinal*, *quantitative*, *timely* and *categorical order*.

The visualisation of that 3D-Cube provides is supposed to provide a quick overview over large data sets. It is separated into two parts. A standard *three-dimensional projection* of coloured cubes and additional projections that provide further information about selected cubes in *pixel based stacked display* visualisation techniques. The user can tailor these techniques by the integration of additional statistical functions. In 3D-Cube the user can interact with the visualisation to explore the database. With special tools he can *select* data, *create projections* to the stacked displays, *define filters* based upon statistical measures and browse along the hierarchies of the data using OLAP operations in an *explorative zoom*.

### 4.2.4 Mustang

The Multidimensional Statistical Data Analysis Engine (Mustang) is a framework for the implementation of statistically enriched VA systems [TRM10] developed at the OFFIS Institute for Information Technology.

Mustang was initially developed for the analysis of high-dimensional epidemiological data collected in the context of the EKN. It has since then evolved into a platform to implement a broad range of VA system across different application domains. Due to it's broad application spectrum Mustang provides a large set of properties to classify VA systems, hence it's inclusion in the examination.

In figure 4.4 a screenshot of the analysis tool is shown. Mustang is not only an analysis tool but can rather be seen as an integrated platform for an analytical information system. It incorporates an OLAP based data warehouse (DWH).

To support geo-spatial analysis, that is invaluable in the epidemiological domain, a geographic information system (GIS) is integrated into Mustang. Additionally advanced statistical measures can be calculated by an integrated statistics tool (R-Project). This enables the analyst to not only explore and visualise the available data but also to generate advanced knowledge by statistical values which can automatically be calculated.

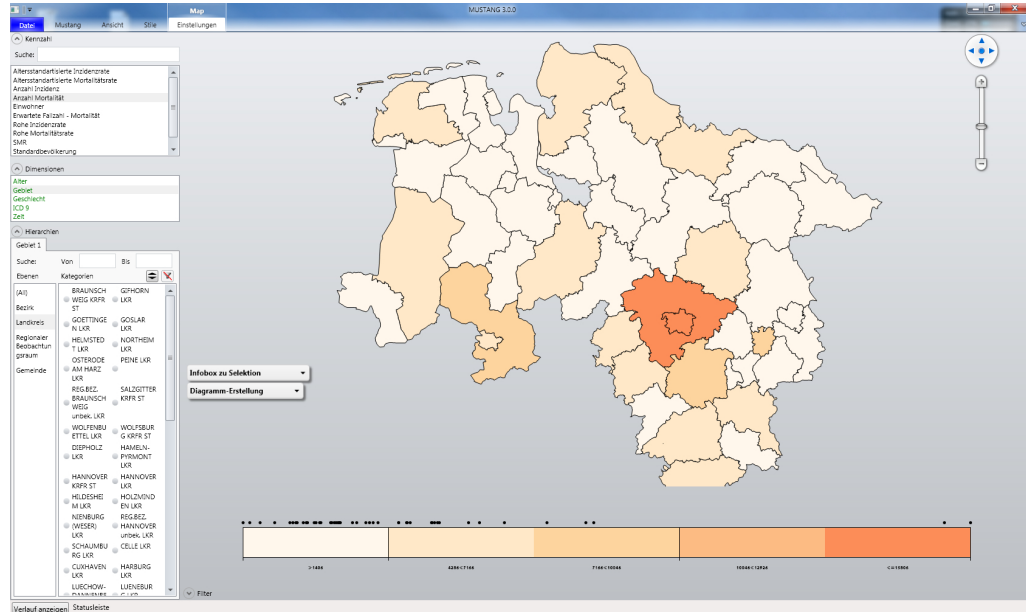


Figure 4.4: Screenshot of Mustang.

Mustang features various visualisation methods. In the screenshot in figure 4.4 an exemplary visualisation of a thematic map is illustrated. In this map, smaller regions are shown, each representing the boundaries of a certain administrative district. On the left side of the tool, user interface elements to explore the OLAP database are shown. These are (from top to bottom) lists for available measures, dimensions and hierarchies. From these lists the analyst can choose measures and dimensions to display. When the data is enriched with spatial information (thus features a geographic dimension) it can be mapped meaningfully to a thematic map, as shown in the screenshot.

Above the thematic map various other visualisation methods are integrated, ranging from static standard charts to explorative pivot tables that visualise the data in numerical form and allow to directly explore it along the hierarchies of the OLAP database.

## Conclusion

Mustang is a VA tool with general applicability and as such features a number of properties that are relevant for the KnoVA taxonomy. It features numerous visualisation techniques (*standard two-dimensional* charts as well as *geo-spatial* thematic maps) and it can integrate various different OLAP databases. Thus it provides access to a rich set of



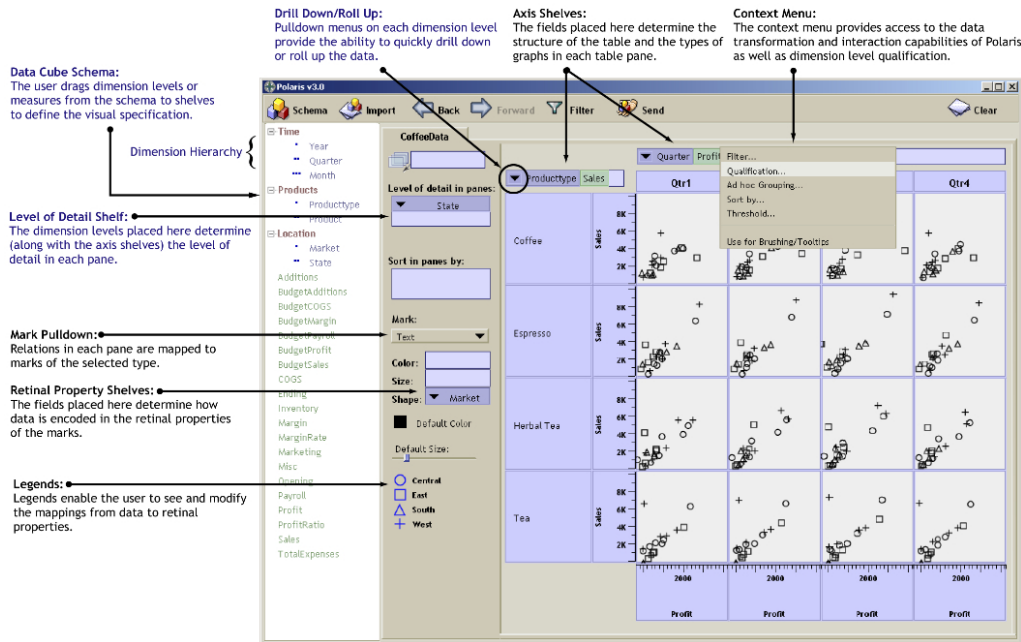


Figure 4.5: Screenshot of the POLARIS UI [STH02].

data sources and supports to analyse *hierarchically* structured multi *dimensional* data. Like 3D-Cube it also integrates the R-Project and thus supports the analysis of data with *algebraic* and *complex* dependencies.

#### 4.2.5 POLARIS and Tableau

POLARIS is a system for the analysis and visualisation of multi-dimensional relational databases. It features an interface to visually query large databases rapidly, supporting the analysis cycle of hypothesis and experimentation [SH02]. POLARIS was developed in a research project at Stanford University<sup>2</sup> and is one of the most widely cited and published examples for a generic integrated VA system in InfoVis and VA communities. The results of this research project were later transformed into a commercial product called Tableau<sup>3</sup> that – according to the vendor – is one of the most widespread commercial VA solutions. Tableau features the same basic structure as POLARIS, enriched by the integration of additional data sources and visualisation methods. POLARIS is included in this examination because it has a broad application spectrum and because of its importance in research and as a commercial VA solution. A screenshot of the visual interface of POLARIS can be seen in figure 4.5.

The interface of POLARIS can be structured horizontally into three sections: a list for

<sup>2</sup> <http://graphics.stanford.edu/projects/polaris/>

<sup>3</sup> <http://www.tableausoftware.com/>

data selection, a panel for refinement of the visual specification of the visualisation, and a panel for the visualisation. On the left side a tree-like structured list for data selection can be seen that is used to explore the data source. POLARIS integrates a number of relational database systems, OLAP databases, and CSV files. In the tree-like list the user can choose dimensions, measures and hierarchical levels to be visualised. The section of the data to be visualised is assigned to the visualisation by drag-and-drop interaction. The next panel is for the specification of the visualisation. Several controls here allow a refinement of the visual specification. For instance, the level of detail can be manipulated and the visual encoding can be defined. In addition to that, a legend for the current mapping of data to visual properties is displayed here. On the right side of the screenshot, the actual visualisation is shown. POLARIS originally extends the concept of the pivot table by adding visual encodings for numerical values to a table-based structured visualisation [SH02]. The table-based visualisation consist of a number of rows and columns. Each axis of the table may contain multiple nested dimensions or measures. By default dimensions of databases are interpreted as independent variables and measures as functional dependent variables. Each cell in the table contains a visualisation of the data. The visual encoding is defined by the user and can either be a visual representation of a single value (for instance a direct visual mapping of a numeric value to a visual property such as diameter of a circle) or a multi-dimensional visualisation such as the scatterplots shown in figure 4.5 [SH02].

POLARIS features three different types of visualisations, concerning the composition of the axis: ordinal-ordinal, ordinal-quantitative and quantitative-quantitative. These different compositions of values can be visualised. Nested visualisations allow mixed types of these base types. Examples for this are scatterplots matrices or thematic maps [SH02, STH02].

POLARIS features several methods for the analyst to interact with the visualised data. These features are referred to as visual queries. Visual queries are defined by the user by his interaction with the visual interface and translated into a special algebra, called VisQL (visual query language) [Han06]. The queries are mapped to the algebras of the underlying database. Hence visual queries can be seen as a method of abstraction to integrate multiple data source in a consistent user interface.

## Conclusion

POLARIS is a widely used VA system with a broad applicability. As such, it features a large variety of visualisation techniques: *standard two-dimensional* charts, *geometric transformations* such as parallel coordinates, *pixel based* visualisations, *stacked displays* such as tree maps, *geo-spatial* thematic maps and most prominently icon and glyph techniques that were first used to demonstrate the integrated VisQL.

Another perspective that provides classifying properties present in POLARIS is the perspective of the data to be analysed. POLARIS is suitable for the analysis of *multi-dimensional*, *hierarchical* data. Data in POLARIS can be explored by interactions to define *selections* and to refine the visualised section of the data in *explorative zooms*.

### 4.2.6 Cambiera

Cambiera<sup>4</sup> is a tabletop system designed for co-located collaborative VA of text document collections [IFM<sup>+</sup>10, IF11]. It is included in this examination because it encompasses mechanisms for collaboration and knowledge sharing.

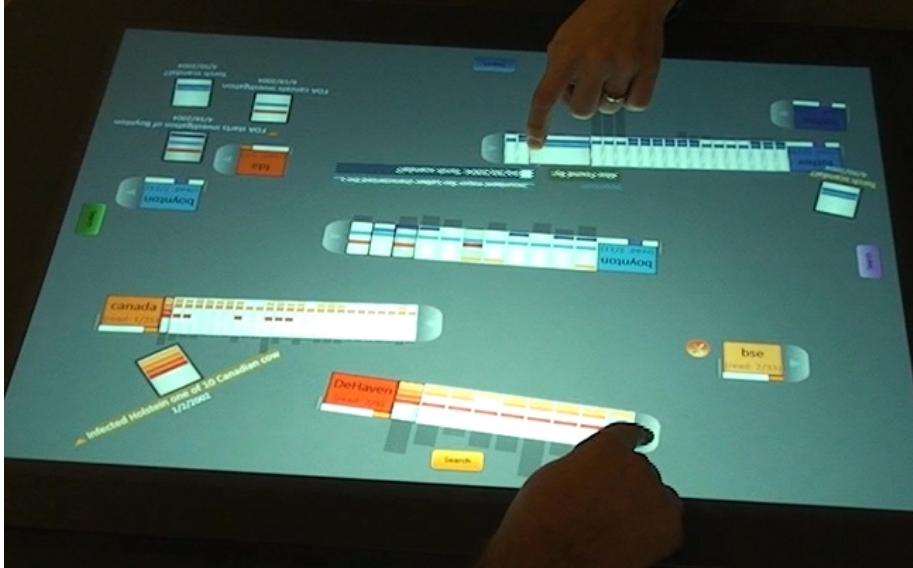


Figure 4.6: Two users working with Cambiera [IF09].

Figure 4.6 shows a screenshot of Cambiera. The screenshot shows the fingers of two analysts working collaboratively. The system is operated by manual gestures on an interactive tabletop. It can be used in an exploratory, cyclic process of foraging, evidence gathering, and hypotheses generation [IF09].

The user can browse through document collections with Cambiera by expressing search queries. Search results are presented in a visualisation where the user can directly access to the document for detailed analysis. The search is started by a coloured search button. When the user touches the search button on the interactive surface, an on-screen keyboard appears in which the user enters the search query.

For each search result then a search box is displayed. A number of these search boxes can be seen in figure 4.6, where currently both users are interacting with a search box. The search boxes show the search term, the number of documents found for this search term and an iconographic visualisation of each documents with colour coded stripes which indicate where in the document the term occurred.

As a key aspect, Cambiera features several collaborative techniques [IF09] for knowledge sharing between analysts. The search boxes are colour-coded, according to the user who initiated the search. For this each user is represented by a distinct base colour, which may appear in different hues across the analysis surface to allow a quick assertion

<sup>4</sup> A video demonstration of Cambiera can be found here <http://www.youtube.com/watch?v=E9izFMJ5yms>

of search boxes to the respective user. This is done in order to allow the users to maintain the identity.

If a user searches for more than one search term, each of the search boxes will receive a certain hue of the same colour. For instance a search box for one term may be dark orange whilst the search box for another term might be light orange, with orange representing a specific user. However, as soon as multiple search boxes are active, the results are also displayed in the other search boxes with a bar in their specific hue, thus indicating if more than one of the current search terms occurs in a document.

When another user performs a search, his search boxes will receive different hues of another colour, for instance dark blue and light blue. In this way user can instantly see the if search terms of other users appear in their current search results.

Another feature to support co-located collaboration is a colour-coded mark in each open document that indicates which other user is reading the same document. This is implemented by the display of small coloured glyphs in a corner of an open document, one for each person who is concurrently reading the same document.

## Conclusion

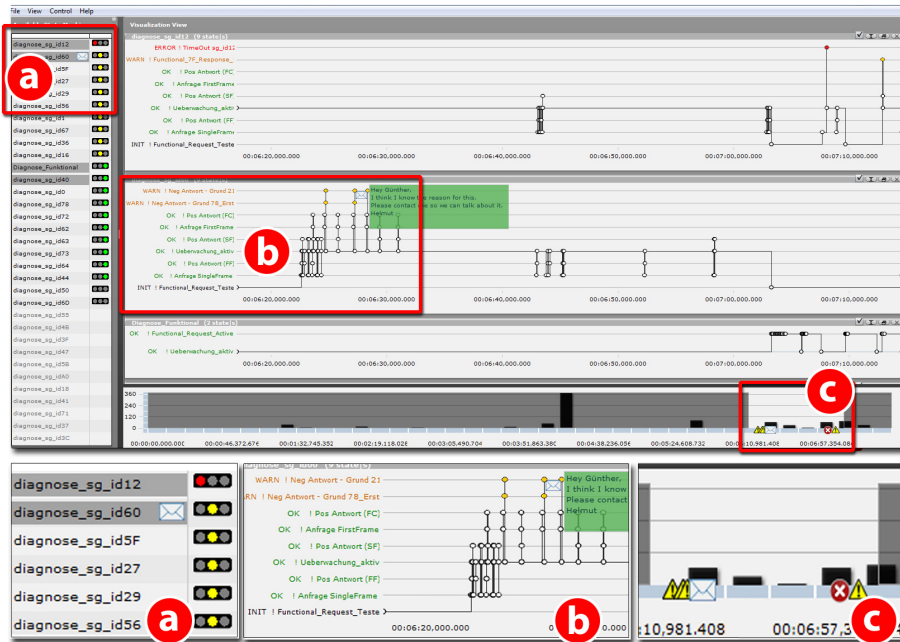
Unlike the previous examples, Cambiera is not aimed at the analysis of multi-dimensional data. Therefore, properties that were not present in the previous examples can be examined in Cambiera. The objective of Cambiera is the analysis of large collections of *text documents* with *icon and glyph* visualisations. In addition, Cambiera encompasses mechanisms for knowledge sharing, especially it allows analysts to trace the course of the analysis of their colleagues allowing them to identify when they work on the same documents and, thus, enabling further collaboration. With the support for knowledge sharing Cambiera is an example of a VA system where integrated mechanisms for knowledge representation and sharing are important. The knowledge sharing in Cambiera is implemented by a visualisation of search terms across users. This is an example how knowledge is shared directly to support collaborative work.

### 4.2.7 Cardiogram

Cardiogram is a VA system that supports automotive engineers in debugging ICNs through the analysis of message traces [SIB<sup>+</sup>11]. The Cardiogram software was developed in a multi-year field study in cooperation between a large car manufacturer and the University of Munich, Germany. It is included in this examination because the analytical challenge that lead to Cardiogram closely resembles the first motivating scenario. Cardiogram is a single-user system. A screenshot of Cardiogram can be seen in figure 4.7.

Instead of analysing the raw bus traces, Cardiogram features an abstraction of the data. An automated pre-processed step maps the timely course of the messages to predefined finite state machines (FSM), that express the idealised behaviour of the ECUs on the ICN.

The FSMs describe the state of the ICU on an abstract level. The trace is replaced by

Figure 4.7: Screenshot of Cardiogram [SIB<sup>+</sup>11].

the timely behaviour of the FSMs, in which all state changes over time are present. New messages on the trace can result in a state changes. States can be of the types *error*, *warning*, or *okay*, and *init*. The init state is used to encode the initialisation that occurs in every FSM before a defined state is reached for the first time.

Cardiogram provides three different views (compare figure 4.7): (A) state machine view, (B) visualisation view, and (C) range slider / overview bar. A complete view of the Cardiogram software is shown in the top half of figure 4.7. Magnified details of the screenshot are visible on the bottom-half. The views resemble an analyse first, details on demand process. The three different views are described in detail next:

**(A) State Machine View:** In this view, all FSMs that are present in a test run are visualised, ordered by their relevance. FSMs that at some point in the test run have been in an error state are listed on top, followed by FSMs with warnings and okay states. FSMs where no state changes occurred are visualised on the bottom of the list. In the magnified view (A) it can be seen that each entry in the list consists out of a textual label (the name of the FSM). A horizontal visualisation that resembles a traffic light is visible on the left. It indicates the most relevant state type reached in this particular FSM colour-coded, with red = defect, yellow = warning and green = okay. In addition to that a small mail envelope can be seen. This envelope indicates that another analyst has already worked on this specific trace file and has annotated this FSM with a textual note. In this way a rudimentary mechanism for knowledge sharing is integrated into the Cardiogram tool that allows analysts to post written notes to their colleagues.

**(B) Visualisation View:** In this view the timely behaviour of a FSM can be displayed.

For this, every state that is present in a certain FSM is represented by a horizontal line, it's life line that represents the time from the initial start to the end of the test run. On the right side, as visible in the magnified view (B), a label with the name of every state is visualised. The colour coding for these labels is the same as above. It can be seen that a FSM can encompass a number of different error, warning, and okay states. In this list the states are ordered in a similar manner as the FSMs in the FSM view.

The most important states (error) are shown on top, followed by the warning states in the middle and the green okay states on the bottom. The state at the bottom is the init state. During the test run several state changes can occur. These are indicated by dots on the life line of a state. Whenever a state is entered or a state is left, a dot on the life line will be shown. Vertical lines between two corresponding states visualises the transitions between the states.

In figure 4.7 a text field can be seen next to an envelope icon close to the life line of one state. These icon represents an annotation that has been made by one analyst in order to be read by others. The annotation can be expanded directly in the visualisation view, as shown in the screenshot.

**(C) Range Slider / Overview:** This view serves as an overview to quickly navigate within the trace file to time stamps which are most interesting. For the navigation, the aggregated sum of all transitions within a certain adjustable time interval is shown as a vertical bar chart. This can be seen in the magnified view (C) where five different bar charts for six different intervals are shown, indicating the number of transitions in each respective interval. In this example, no transitions happened in the first interval. In addition to that, icons indicate when an error state has been reached (red circle with white cross) or a warning state has been reach (yellow triangle with black question mark). In the same manner annotations are visualises by the envelope icon.

## Conclusion

Cardiogram is an example for a VA system for the analysis of ICNs. Therefore, the properties of Cardiogram are relevant in the scope of the first motivating scenario. Cardiogram features several visualisations on different levels of abstractions for the analysis of bus traces. Standard *two-dimensional* charts extended by *textual* annotations, *textual* lists enhanced by *icon an glyph* techniques, and complex customised visualisations. The interaction with Cardiogram follows the direct manipulation paradigm. *Selections* are made directly on the visual interface. The visualisation techniques in Cardiogram visualise *qualitative* and *ordinal* information in the trace files, *textual* annotations, and the *timely* behaviour of the bus messages with their *complex* structure. Cardiogram is an example for a VA system that includes visualisation techniques that are domain specific adoptions of standard visualisations.

## 4.3 The KnoVA Taxonomy

In this section a taxonomy for the classification of knowledge-based VA systems, the KnoVA RA, will be introduced. The properties and classes of the taxonomy are derived

from the description of the VA systems in the previous section. Starting point for the design of this taxonomy is the VDET introduced by Keim [Kei01]. In the VDET a virtual three-dimensional space of visualisation is being defined by three orthogonal classes: *data to be visualised*, *visualisation technique*, and *interaction technique*. Keim identifies 17 classifying properties across the three orthogonal classes.

The taxonomy presented here consists out of five distinct classes and identifies 38 descriptive classifying properties. In addition to the VDET two additional classes are identified: *exploration technique* and *dynamic representation*.

The VDET was chosen as the foundation of the KnoVA RA because it is well known and wide spread in the scientific communities for IV and VA. The VDET is a descriptive taxonomy that can easily be extended by new properties, without breaking existing classifications. When new properties are added, the existing classifications remain. A mapping between any existing classification in the VDET can be made to classify the same system in the KnoVA RA.

There are two reasons why the VDET is insufficient in the scope of this thesis. Firstly the intention of the VDET is the classification of visual data exploration techniques. Therefore, it is off the scope of knowledge-based VA. In the following derivation of the classifying properties it becomes clear, that not all aspects of knowledge-based VA systems can be classified by the VDET.

Secondly the VDET intends to provide a classification of VA systems. The intention of the KnoVA RA is to provide the fundamentals for a meta data model for VA systems. Therefore the requirements to the KnoVA taxonomy are different.

Next to additional classifying properties there are also new classes introduced in the KnoVA RA. The class *exploration technique* is introduced a specialisation of the existing class *interaction technique*. An exploration technique in this scope defines a state changing operation. This discrimination is important to identify between those user interactions, which solely change the visual appearance of the VA system and those interactions which change the actual state of the underlying data. In chapter 5 it will be shown that the latter ones are important for the identification of applied knowledge.

The class *dynamic representation* is introduced because certain features of the examined VA systems cannot be described by the VDET and the classifying properties do not fit in any of the existing classes. Therefore this new class is introduced.

Some of the classifying properties that are present in the KnoVA RA are not derived from the examination of properties of VA systems in the previous section. Instead they are directly derived from the VDET and are included here for completeness. Such properties are especially mentioned in the following description.

#### 4.3.1 Iconographic Language for the Design Space of VA

To visualise the derived properties, an iconographic language based upon the work of Aigner et al. [ABM<sup>+</sup>07] is used. Aigner et al. originally introduced the iconographic language to describe the design space of a conceptual VA framework for the analysis of time and time-oriented data. Here, the iconographic language is used to describe the

design space of knowledge-based VA. Where both design spaces overlap, the icons introduced in [ABM<sup>+</sup>07] are being used. Where no matching icons are available, new ones are introduced. The iconographic language is used because for a quick discrimination of the classifying properties to extend the comprehensibility of the KnoVA taxonomy. The importance of the comprehensibility is underlined by Frank's suggestion to evaluate this aspect in order to measure the quality of a reference model [IWR<sup>+</sup>10].

The following subsections are structured according to the classes of the KnoVA taxonomy. Each subsection describes the classifying properties of its respective class.

The iconographic language was enhanced by a colour coding to identify the classes. Every class is represented by a specific colour. The icons of the properties in a class share the same colour as their background colour.

### 4.3.2 Data To Be Visualised

The starting point for the taxonomy is the class data to be visualised as this is the central and defining element of the analysis. In VA systems, the number of data sets to be analysed is typically large, as the assets of VA lie in the analysis of large amounts of data sets where automatic or manual evaluation fails.

**Term (Data Set).** A *Data Set* is a collection of data which represents a distinctive observation of a real or virtual phenomenon with a distinctive number of interrelated attributes. Each attribute in a data set has a specific type. As an example, in science a data set could describe all measured parameters of an experiment, e.g. temperature and time. In an relational database, a data set typically represents a real world entity (such as employee) with all attributes of this entity (such as name, date of birth and salary) [EN09].

Various different types of data to be visualised can be identified. To further structure the domain of this class, and therefore to extend the VDET, three subclasses are introduced at this point: *data characteristic* to describe classifying attributes concerning the specific occurrence of the data, *data structure* to describe inner structures, hierarchies and interrelations to classify data, and *data order* to describe order relations that exist within the data.

The subclasses are introduced to create a more fine-grained taxonomy that can serve as the foundation of a meta data model for VA systems. The elements of the subclasses are realised in the meta data model in different ways. The elements in the subclass *data order* will serve as the foundation for functional dependencies. The subclass *data structure* serves as the foundation to model visualisation axes and their values. The class *data characteristic* subsumes elements that can be used to describe the dimensionality of axes and their values.

All classifying properties in the class data to be visualised can be allocated into one of these subclasses. The subclasses and the allocated classifying properties can be seen in figure 4.9. The following examination is limited to structured or semi-structured abstract data (such as numerical values) which is *descriptive* than *representative*.



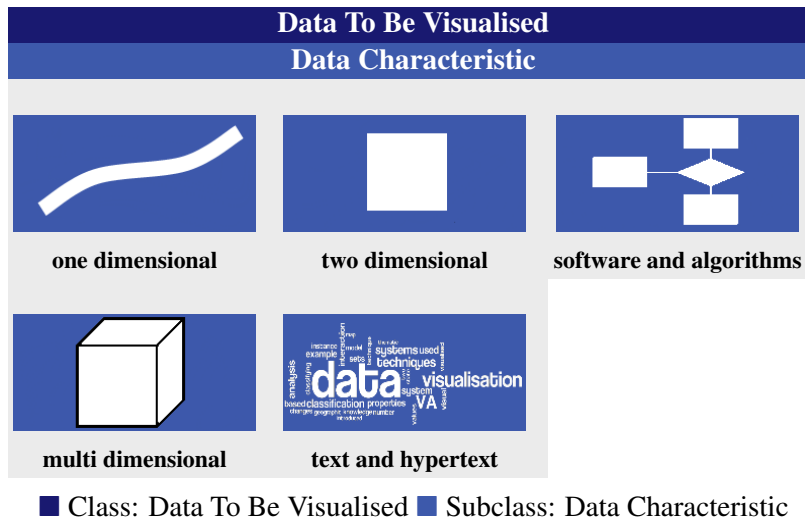


Table 4.1: KnoVA Taxonomy, Class: Data To Be Visualised, Subclass: Data Characteristic.

VA focuses on descriptive data (compare section 2.1) and is therefore separated from scientific visualisation, which is focussed on representative data that typically has a direct spatial mapping, like for instance digital x-ray images which directly represent real world objects.

**Data Characteristic:** Data occurs with different characteristics. Even when the analysis is limited to structured or semi-structured data a number of distinct occurrences can be identified. The selection presented here to describe the design space of knowledge-based VA is based on the VDET, where six classifying attributes for the class data to be visualised were identified. These attributes are subsumed in this taxonomy into the subclass data characteristic.

The number of attributes of a data set defines its dimensionality. Data sets with a single attribute are called *one-dimensional*. One-dimensional data sets often represent one continuous attribute that allows ordering the data. Typical examples for one-dimensional data sets are series of numerical measures with no inherent order. In the domain of knowledge-based VA one-dimensional data is a rare special case. Nevertheless many VA systems support the analysis of one-dimensional data at least indirectly. In TaP and Mustang for instance certain visualisation methods can be used to display a measure with only one dimension. In these systems this typically is incorporated by restricting data sets with more than one dimension. Most VA data sets encompass a large number of attributes [Kei02a]. This is reflected by the evaluated VA systems that all support the analysis of *multi-dimensional* data sets with many attributes. Examples for multi-dimensional data sets are tuples from relational databases (Polaris and Tableau), cell sets from OLAP databases (TaP, VAT, 3D-Cube and Mustang) and the XML based records of bus messages analysed in Cardiogram. In the VDET *two-dimensional* data is

specifically outlined for geo-spatial data, for which a direct two-dimensional mapping is defined. An example for this are measures which are distributed over geographic regions. Although VA, like InfoVis in general, deals with abstract data without a spatial mapping, geo-spatial data is a possible exception. However two-dimensional data in the scope of knowledge-based VA is a special case of multi-dimensional data. Often geo-spatial data analysed in VA systems will have a high-dimensionality. The epidemiological records that can be analysed in TaP, Mustang and VAT are a good example of this. Actually here a single patient as entity can have multiple geographic coordinates such as work places, living places etc.


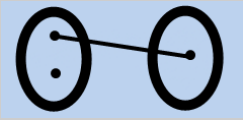

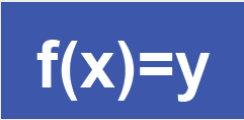
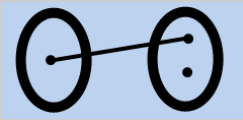
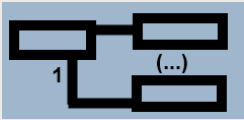

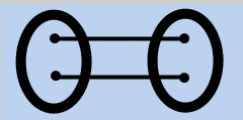
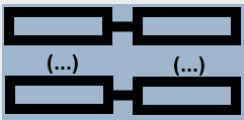
Not all data sets can be specified by a fixed dimensionality. In VA research the analysis of *text and hypertext* is an important factor [TC05, IWR<sup>+</sup>10]. Due to the growing amount of documents on the web and the growing amount of email communication the relevance of text and hypertext is increasing [Kei02a]. This is reflected in Cambiera, which is specifically aimed at the analysis of large amounts of documents.

Related to text and hypertext, a third distinguishable class of data to be visualised is source code. Therefore Keim identifies *software and algorithms* as another classifying property, which therefore is listed for completeness. Source code is the textual representation of executable software and algorithms. The analysis of software and algorithms is important in VA as a research area but cannot be identified as important in the scope of this thesis.

**Data Structure:** In addition to the specific occurrence of the data, there are other attributes of data that can be used to classify the data in more detail. All properties introduced thus far are directly derived from Keim's VDET. Another attribute of data that can be used for classification is its inner structure.

Data which contains interrelations and functional-dependencies cannot be described by the classifying properties introduced thus far. In OLAP data cubes, for instance, dimensions often have a *hierarchical* structure. An example for a hierarchical dimension is time. It has several hierarchical levels (seconds, minutes, hours, days, month, years). TaP, VAT, 3D-Cube, and Mustang feature such hierarchical data cubes to explore the data throughout the hierarchical levels. When a measure is available for a lower hierarchical level, as for instance sales per day, measures for higher hierarchical levels can often be calculated based upon pre-defined rules. In this example the yearly sales can be calculated by adding up all daily sales measures. Other types of hierarchical data are network topologies and file systems.

Another attribute that can affect the presentation of data in VA systems is an existing *algebraic dependency* in the data [RM90]. An example for an algebraic dependency is the total production cost for a specific element of a product. The measure can be algebraically derived as the sum of actual manufacturing cost and research cost. Algebraic dependencies are comparable to hierarchical dependencies. Other than through hierarchical dependencies, it is not necessarily possible to explore through the algebraic hierarchies in both directions. For instance, a drill-down path from total production cost to the both addends exists but a roll-up path from one of the addends to the sum cannot be drawn. Therefore, it is useful to distinguish algebraic dependencies from hierarchical

Data To Be Visualised		
Data Structure		
	Coverage	Cardinality
 hierarchical	 surjective	 one to one
 algebraic	 injective	 one to many
 complex	 bijective	 many to many

■ Class: Data To Be Visualised ■ Subclass: Data Structure ■ Cardinality ■ Coverage

Table 4.2: KnoVA Taxonomy, Class: Data To Be Visualised, Subclass: Data Structure.

dependencies, as they might affect the presentation and interaction of the data in the VA system. Of the examined VA systems, Mustang allows to analyse data across algebraic dimensions that are calculated ad-hoc by integrated statistical tools.

Many algebraic mappings will have results in the same value type. Some algebraic dependencies, though, can yield to other value types. These *complex* dependencies describe a functional dependency between two attributes of one data set. An example is the duration of a medical infection, indicated by two fields: day of the diagnosis and day of the recovery. The curation period can then be defined as an interval between those two time stamps. Therefore, two time values are mapped to a timespan. Complex dependencies will typically need to be visualised on specialised visualisation methods. For a timespan, for instance, a bar chart with start and end point can be used or a specialised visualisation such as UML sequence charts which are used for this purpose in Cardiogram.

Functional dependencies between data sets can also be characterised based upon their *cardinality*. The cardinality expresses the number of elements of a set to which a relation can map from an element of another set [RM90]. The cardinality of a relation can influence the representation of the data in the VA system and the interaction with the data. VA systems featuring relational data (Polaris) or OLAP systems (TaP, VAT, 3D-

Cube, Mustang) incorporate functional dependent data structures of different cardinality. Here, the cardinality refers to relations between entities. The cardinality of a relation can either be *one-to-one*, *one-to-many*, or *many-to-many*.

In addition to the cardinality functional dependencies can also be described by their *coverage* as *surjective*, *injective*, and *bijective* based upon the mathematical definition for relations between two sets [Bou06]. The coverage describes whether VA systems are suitable to only analyse data with specific functional dependencies. These properties limit the suitability of a certain visualisation.

**Data Order:** Another class that can be used to describe data sets in VA systems it's scale or partial order according to order theory, which allows to compare elements of a set with each other.

**Term (Partial Order).** A *Partial Order* is a binary relation over a set  $PO$  which is reflexive, antisymmetric, and transitive for all elements  $a, b \in PO$  with definitions of reflexive, antisymmetric and transitive according to [Kna01].

Fields in databases can be characterised as *nominal*, *ordinal*, *quantitative*, and *categorical* (or interval) [SH02, Ber83, Ste46]. Nominal scales are descriptive attributes with no intrinsic way to arrange such as male and female or Asia, Europe, America, Africa, Australia. An example for ordinal values are the system states in Cardiogram that contains a pre-defined order indicating the relevance of reached states. Quantitative values are all kinds of numerical values that can be compared by mathematical operators such as larger or equal. Categorical values define a mapping between an ordinal scale (such as low, medium, high), quantitative values or ranges of quantitative values. Some ordinal dimensions also contain hierarchical structures or functional dependencies as, for instance, *time* and *geographic* scales. The data scale influences the way data sets can be mapped to a visual representation.

### 4.3.3 Visualisation Technique

With the exception of geo-spatial data, VA systems focus on data sets lacking inherent spatial semantics and, therefore, also lacking a standard mapping of abstract data onto the physical space [Kei01]. A large number of visualisation techniques has been developed over the past decade to visualise larger and more complex or multi-dimensional data sets [CMS99, Kei01].

**Term (Visualisation Technique).** A *Visualisation Technique* or *visualisation method* is a visual representation for data. An example for a visualisation technique is a pie-chart.

Keim identifies five different classifying properties in the class visualisation technique: *Standard 2D / 3D* visualisation techniques like X-Y plots, bar charts, line charts, pie-charts, etc. that are available in most VA tools. For instance VAT, Polaris, and Mustang feature a large number of these standard visualisation techniques. *2.5D* visualisation techniques like 2D projected 3D images can also be categorised with this property. These standard visualisation techniques typically do not perform a visual transformation (refer

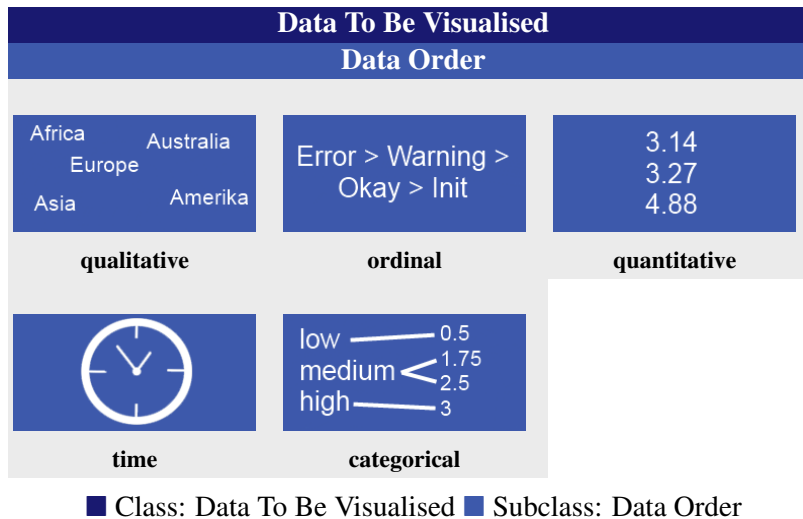


Table 4.3: KnoVA Taxonomy, Class: Data To Be Visualised, Subclass: Data Order.

to section 2.2). Instead, typically an attribute of the data is mapped directly to an attribute of the visualisation. As an example, the total amount of sales can be mapped directly to the length of a bar chart.

Due to the direct mapping between data attributes and visual attributes, standard 2D / 3D techniques are limited to a relatively small number of dimensions. To overcome this problem, geometrically transformed displays are using *geometric transformations and projections* to produce useful visualisations. Important techniques for visualising data sets with a large number of numerical attributes are parallel coordinates [Ins09] and various techniques for graph visualisation [Kei01].

*Icon and glyph* techniques display each data set as an icon and target the human capability to quickly discriminate and recognise variable shape and sizes of objects. Attributes of the data are mapped to a specific visual feature of the icon or glyph. Many icon and glyph techniques are very specialised and, therefore have limited applications. Combined with standard 2D and 3D techniques, however, icon and glyph techniques appear in many VA systems. Examples are the email icon and traffic light visualisation in Cardiomgram, and in the scatter plot matrix of POLARIS and Tableau.

The biggest disadvantage of icon and glyph techniques is that the number of displayed elements on the screen is proportional to the number of data sets. Thus, with a large number of data sets it becomes increasingly difficult to identify differences due to the limits of human cognition [Spe07]. An approach to increase the amount of visualisable data are *pixel-based* displays in which every pixel gets assigned a value. Pixel-based displays, therefore, offer the highest possible information density. However, their application is limited to data sets that are relatively equal to each other across the value range and in which changes are expected in larger associated regions of the data instead of in

single data points. A diverse distribution of values will result in noise images. Changes in single values will result in changes of single pixels and, therefore, might be overseen. In order to display hierarchical multi-dimensional data and, at the same time, provide a meaningful visualisation of the hierarchy, *stacked displays* are used. The stacked half-pie menu in TaP or pivot tables as provided by Mustang, Polaris, and Tableau are examples of such hierarchically stacked displays.

As mentioned above, most visualisation techniques in VA system focus on abstract data sets. An exception is is geo-spatial data or geographically distributed data such as regional sales figures or epidemiological data sets. Data containing geographic information can easily be displayed on *geo-spatial* thematic maps representing geographic regions. The VDET does not define a special classifying property for geo-spatial visualisation methods. However, geo-spatial visualisation techniques are integrated into TaP, Mustang, VAT, Polaris, and Tableau. Therefore, this classifying property can be identified as an important in the scope of this thesis. Geographic regions can occur either as coordinates (e.g. Gauss-Krueger coordinates) or as categoric values (Africa, Asia, Europe).

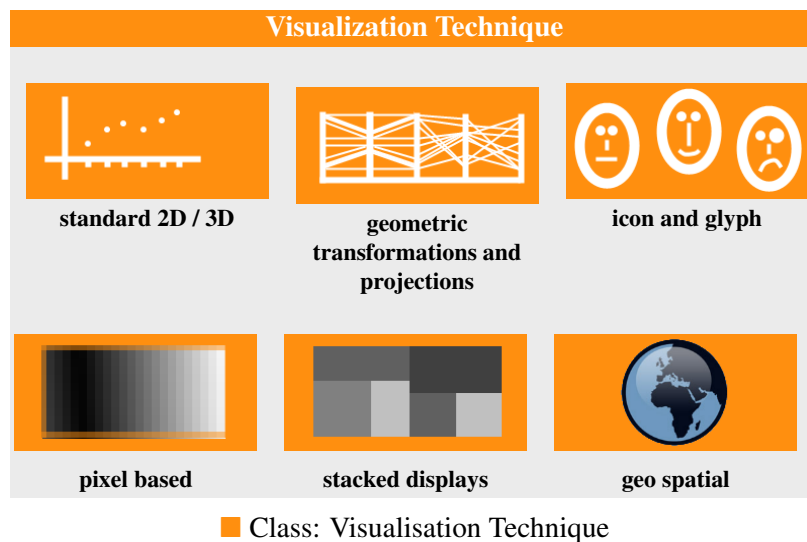


Table 4.4: KnoVA Taxonomy, Class: Visualisation Technique.

#### 4.3.4 Interaction Technique

The third class of the VDET relates to interaction and distortion technique. According to Keim, interaction techniques allow users to interact directly with a visualisation and are used to make dynamic changes according to the exploration objectives, whilst interactive distortion techniques support the data exploration process by preserving an overview of the data during drill-down operations [Kei01]. Originally six different classifying prop-

erties are identified, each representing a specific interaction or distortion technique. The properties presented in the VDET describe different classes of interaction techniques. When looking closely at VA systems, it can be distinguished between interaction techniques which only affect the visual appearance of a VA system and do not result in a change of the state of the analysis system and exploration techniques that do.

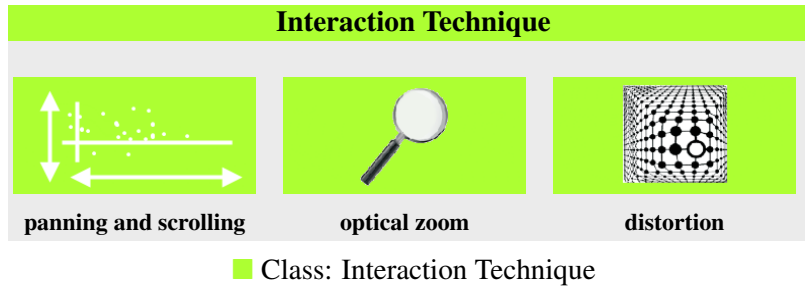


Table 4.5: KnoVA Taxonomy, Class: Interaction Technique.

**Term** (Interaction Technique). An **Interaction Technique** describes a technique to modify the appearance of a visualisation technique. Interaction techniques can be seen as transient user interactions with the visualisation system. Changes resulting from interaction techniques have no effect on the underlying data. Instead they only effect the visual representation of the data. An example for an interaction technique is a stateless zoom by which a certain area of the visualised data is magnified but the visualised data itself remains unchanged.

State changes are the defining element of the analysis process in a knowledge-based VA system. Therefore, to match the scope of this thesis, in the taxonomy presented here two classes are introduced to replace the original class interaction and distortion technique: the class interaction technique, which is described in this subsection, and the class exploration technique, which is described in the next section.

Even when visualisation techniques with a high data density are used, often the amount of data to be visualised exceeds the available space. This results in the demand for techniques to navigate and specifically highlight certain aspects of the data [SBM08].

The simplest stateless interaction techniques are *panning and scrolling*. With the help of panning and scrolling the visualisation space can be virtually extended, allowing to display a larger amount of data in hidden regions of the visualisation display. Panning and scrolling is integrated in most graphical user interfaces. All of the VA systems presented above support panning and scrolling.

Related to the panning and scrolling interaction is the *optical zoom* interaction. This interaction also modifies the visible visualisation space. In contrast to panning and scrolling, an optical zoom will increase or decrease the magnification and, therefore, increase or decrease the amount of visualised data and, hence, the amount of detail that can be identified by the user.

A third type of stateless interaction techniques are *interactive distortion* techniques. These techniques modify the shape of the virtual visualisation space. An example for this is an adaptive magnification such as a fish eye lens that magnifies the elements on the visualisation space according to their relevance, for instance according to their distance from the mouse cursor. This results in an image where more important data elements are visualised in higher detail.

#### 4.3.5 Exploration Technique

As described above, the defining element of the analysis process in knowledge-based VA systems are state changes. These state changes occur during the exploration of the data by the user. In VA systems interactive visualisations provide techniques that allow to modify the visualised data.

**Term** (Exploration Technique). *An **Exploration Technique** describes a technique to modify the underlying data or the visual transformation of a visualisation system. Exploration techniques can be seen as persistent user interactions with the visualisation system. Changes resulting from exploration techniques affect the state of the analysis system. Exploration techniques result in operations on the data such as OLAP operations.*

An example for a state-changing modification is a change in the level of abstraction or *explorative zoom* into the data. Unlike the optical zoom that does not affect the state of the analysis system, an explorative zoom does not merely change the optical representation but triggers an operation on the data. In the TaP system, for instance, gestures can be used to zoom across the hierarchical levels of the underlying OLAP data source. The gestures are mapped to OLAP operations for drill-down or roll-up.

Another important type of stateless interaction techniques is *linking and brushing*. Linking describes the generation of a connection between two or more visualisations. When two visualisations are linked to each other, they display the same area of the data, sometimes in different levels of abstraction. Hence, a mapping function needs to be defined between two linked visualisations that defines the parameters that are linked and how the hierarchical mapping is being calculated. The link between the two visualisations can be used for an exploration of the data. In the 3D-Cube system, for instance, a linkage between the cube visualisation and the projections on the side is calculated. The selected part of the cube is visualised in the projections on the side. If the analyst hovers the mouse pointer over an element in the projections, the corresponding elements in the cube and in the other projections are highlighted. This interaction is called brushing. Brushing is especially useful in projections that are calculated based upon statistical aggregations. In this case, one cell in a projection may represent a set of cells from the cube. By highlighting this circumstance with the brushing operation, the analyst gets valuable information about such complex dependencies.

Brushing itself is a stateless interaction. However, as a link function has to be defined before a brushing operation can be performed these two interactions are inseparable.



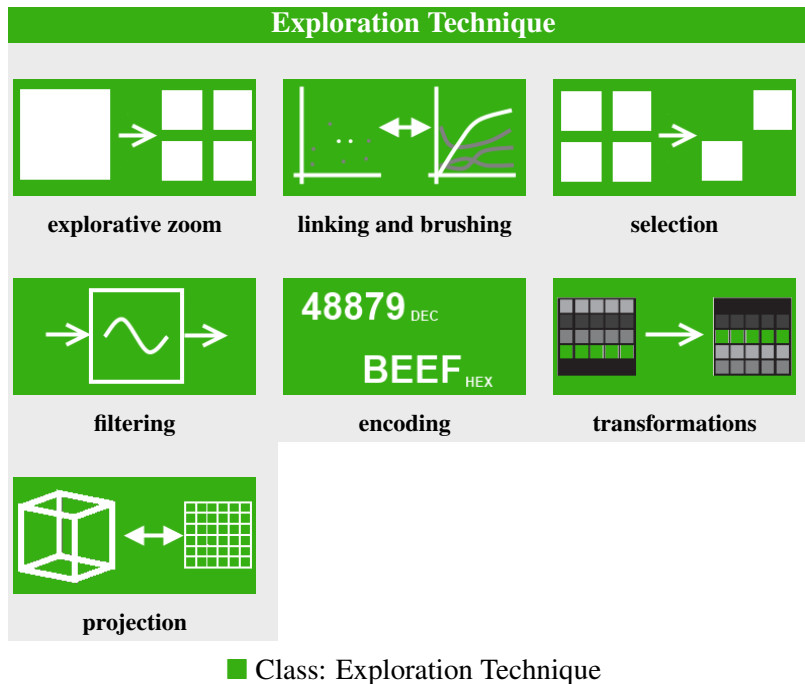


Table 4.6: KnoVA Taxonomy, Class: Exploration Technique.

Besides brushing, linked visualisations can also be used to perform a *selection*. If, for instance, in one visualisation categorical values are displayed and in another linked visualisation numerical values which can be grouped into the categories of the first visualisation, then the categorical view can be used to perform a selection. An example for such an operation can be found in the Mustang system where a tree view with categories (e.g., continents, states, administrative districts) are visualised and in the thematic map numerical values which can be assigned into these categories (e.g., curation / district) are visualised. If, then a certain district is selected in the tree view the thematic map could be altered so that only this district is visualised afterwards.

Selections can also be defined directly on one visualisation, for instance by selecting distinct items and, therefore, are a distinct exploration technique. Closely related to selections are *projection* operations. These can be seen as selections which limit the number of visualised attributes, comparable to projections in relational algebra. Another related operation is the definition of *filters* that is supported in many visualisations. In the 3D-Cube system, for instance, complex filters can be defined based upon statistical calculations with the integrated statistical engine.

Filters are typically defined for a larger number of data sets based upon specific attribute and can be calculated automatically. Another type of state changing operations are *encoding* operations. As an example, some values might be best represented as percentages. Lastly, some visualisation methods allow *transformations* of the visualised data

as, for instance, the definition of an artificial order, e.g. by manual re-ordering of data sets.

#### 4.3.6 Dynamic Representation

Another class that can be identified in VA systems is the dynamic representation of the visualisation. In the VDET this class is not included. Many VA systems only provide a *static* representation of the data that will only change the appearance after user interaction. Even the brushing techniques introduced in the previous subsection depend on user interaction. There are several reasons why animated visualisation techniques, so far, are not popular. Historically, VA evolved out of the research field information visualisation where animated visualisation techniques are less common. In addition in the past the computing power was not sufficient to calculate ad-hoc animated visualisations of large amounts of data. In [Jai99] is an increasing importance of animated visualisations predicted.

**Term** (Dynamic Representation). *The **Dynamic Representation** of a describes whether the visual display is, apart from changes that occur upon user interaction or other external triggers, dynamic (animated) or static.*

There are good reasons to use animated visualisation techniques, especially the fact that certain parts of human cognition and visual perception are specialised to rapidly detect the changes and moving objects [STT06].

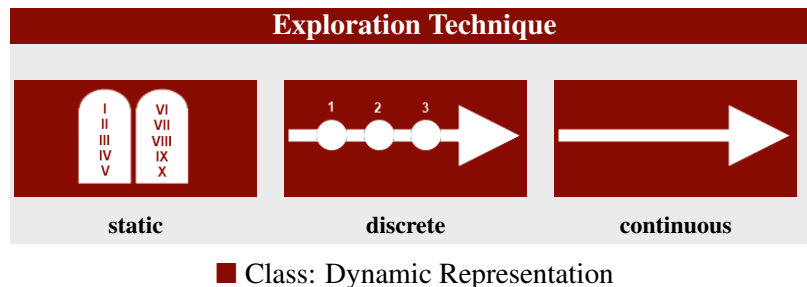


Table 4.7: KnoVA Taxonomy, Class: Dynamic Representation.

The lack of dynamic representation as a class in the VDET leads to the problem that VA systems cannot be categorised based upon this feature. Nevertheless differences between VA systems according to their dynamic representation can be examined. Animated visualisation techniques can be used to discover trends in data [RFF<sup>+</sup>08].

The TaP system supports the animation of timely data by providing a *discrete* slide show of predefined images on the multi-dimensional bubble chart. As an example, the average life span could be assigned to the y-axis, the infant mortality rate could be assigned to the x-axis, different geographic regions could be colour-coded, and the population density in the geographic regions could be assigned to the size of the bubbles. The time

dimension can be used to visualise the variation of these variables over time and, hence, allow the user to analyse the changes and trends these four variables run through over the visualised timespan. Characteristic for *continuous* animations is a smooth transition between frames.

#### 4.4 Summary

A summarising visual overview of the KnoVA Taxonomy can be seen in table 4.9. In the table all properties and classes identified in this chapter are included. In addition to that table 4.4 summarises the attributes in a textual overview, structured horizontally to also represent the classes and subclasses of the taxonomy.

The KnoVA taxonomy consists of 38 descriptive classifying properties that are arranged in five distinct classes, derived in a bottom-up approach by the examination of seven relevant existing VA systems for classifying properties in section 4.2.

At first the TaP system is introduced. It is relevant because it represents a VA system for the analysis of cancer records and, therefore, has a comparable application domain as the second motivating scenario. It is followed by the VAT System that shares the same application domain. In addition it incorporates a basic DSM based model to represent the state of the analysis systems.

The 3D-Cube system and the Mustang system also share the same application domain. Although, regardless of its origin in the EKN domain, Mustang is designed as a universally applicable system that can be used across a variety of application domains, which was another reason to include it in the examination. The same reasons also apply for the investigation of POLARIS and Tableau.

These related systems share the broad applicability with Mustang. In addition POLARIS is one of the most wide spread systems examined in research. Its commercial offspring Tableau claims to be the market leader for general VA systems.

Cambiera was included in the examination because it supports the application and sharing of knowledge that matches the intention of this thesis. Cardiogram, finally, was examined because it represents an existing approach for VA of ICNs which is the objective of the first application scenario.

Following the examination of existing VA systems, the properties and classes of KnoVA taxonomy were derived in section 4.3. The objective of the KnoVA taxonomy is to explore the design space of knowledge-based VA to provide the foundations for a meta data model for VA systems.

The derived properties were visualised using a novel iconographic language that was introduced in subsection 4.3.1.

Eventually in subsection 4.4, a summarising overview over all classes, properties and the VA systems where they have been identified was given.

Data To Be Visualised		
Data Characteristic	one dimensional two dimensional multi dimensional text and hypertext software and algorithms	VAT, Mustang, TaP, 3D-Cube, Polaris <sup>1</sup> VAT, Mustang, TaP, 3D-Cube, Polaris VAT, Mustang, TaP, 3D-Cube, Polaris Cambiera, (Cardiogram) -, (*)
Data Structure	hierarchical algebraic complex	VAT, Mustang, TaP, 3D-Cube, Polaris VAT, Mustang, 3D-Cube VAT, Cardiogram
	Cardinality	one-to-one one-to-many many-to-many
	Coverage	surejective injective bijective
Data Order	qualitative ordinal quantitative time categorical	Mustang, TaP, VAT, 3D-Cube, Polaris, Cardiogram Mustang, TaP, VAT, 3D-Cube, Polaris, Cardiogram Mustang, TaP, VAT, 3D-Cube, Polaris Mustang, TaP, VAT, 3D-Cube, Polaris, Cardiogram Mustang, TaP, VAT, 3D-Cube, Polaris
Visualization Technique		
Standard 2D/3D geometric transformations and projections pixel based stacked displays geo spatial icon and glyph		Mustang, TaP, 3D-Cube, Polaris Polaris 3D-Cube, Polaris 3D-Cube, Polaris, TaP VAT, Mustang, Polaris Cardiogram, Cambiera, Polaris
Interaction Technique		
panning and scrolling optical zoom distortion		- - (*)
Exploration Technique		
linking and brushing selection projection filtering explorative zoom encoding transformations		- VAT, TaP, Mustang, 3D-Cube, Polaris, (Cardiogram) 3D-Cube VAT, 3D-Cube VAT, TaP, Mustang, 3D-Cube, Polaris Polaris Mustang
Dynamic Representation		
static discrete continuous		- TaP -

(System) indicates rudimentary support or support remains unclear

(\*) feature is derived from Keims original classification only or listed here for completeness

- general validity or not applicable

<sup>1</sup> Polaris stands for Polaris and its offspring Tableau

Table 4.8: Overview of Properties of the KnoVA Taxonomy.

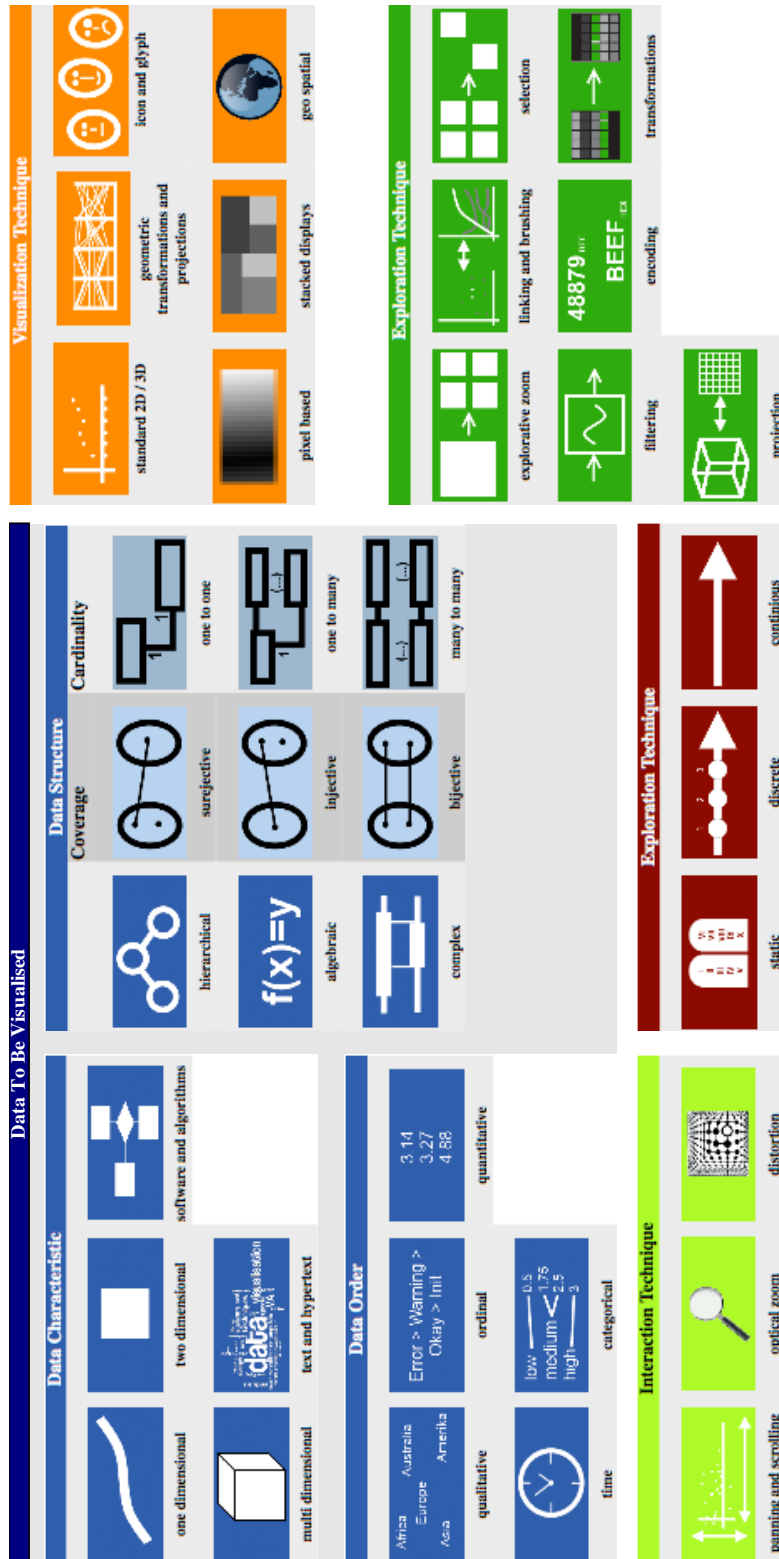


Table 4.9: The Knowledge based Visual Analytics Taxonomy.



## 5 The KnoVA Reference Architecture

This chapter introduces the KnoVA Reference Architecture. The KnoVA RA intends to provide the necessary foundation to create a broad variety of VA systems in which the representation, extraction, management and sharing of knowledge is important. Based upon the requirements, implications for design and the taxonomy introduced in the previous chapters, it can serve as a blueprint for knowledge-based VA systems. It provides concepts, methods and techniques for the design of such systems.

Section 5.1 outlines the applied methodology and the structure of this chapter. Section 5.3 identifies general requirements for a RA based upon the design implications that were identified in section 3.3.

Subsequently section 5.4 introduces a model for the analysis process: the KnoVA Process Model (KnoVA PM). The following section 5.5 defines the KnoVA meta model (KnoVA MM), a meta data model for VA systems. The KnoVA MM is based upon the KnoVA taxonomy.

Section 5.6 shows how the process model and the data model in combination can be used to extract knowledge in the course of the analysis by introducing a conceptual design for knowledge extraction. Section 5.7 uses this design as the foundation for knowledge application. An exemplary algorithm to re-apply knowledge is introduced to the KnoVA RA.

Section 5.8 concludes this chapter. Here the KnoVA RA is reviewed in an evaluative comparison and it is discussed how the KnoVA RA faces the challenges and answers the research question introduced in section 1.1.

### 5.1 Methodology and Outline

The design of the KnoVA RA is structured into three main parts. The first part is devoted to preliminary examinations. The second part is the conceptual part where the concepts and methods of the KnoVA RA are designed. The third part finally is a conclusive part where the introduced concepts are validated.

The first part is formed by the identification of general requirements and the introduction of a fundamental structural architecture in the sections 5.3 and 5.2. Combined these two sections form the foundation for the RA. The methodological approach in section 5.3 is to review the four design implications for general requirements towards a RA for knowledge-based VA systems. Six general requirements [GR1] – [GR6] are specified. After this in subsection 5.2 a fundamental structural architecture for VA systems is introduced, which is based upon the 3-Tier architecture pattern.

The second part is dedicated to the concepts of the KnoVA RA. It is formed by the four sections 5.4 – 5.7 where the concepts of the KnoVA RA are introduced. The concepts for the RA can be structured into four main parts, each of which responds to one of the four design implications and the respective requirements as identified in 5.3. Each of these parts is reflected with a dedicated concept section. An overview of the relationship between design implications, general requirements and the concepts of the KnoVA RA

can be seen in 5.1. In this table the relationship between the design implication (DI1 – DI4), the general requirements ([GR1] – [GR6]) and the corresponding concepts of the KnoVA RA is shown.

Design Implication	General Requirement	Section / Concepts
DI 1: Analysis Process	GR1: Various Visualisations	Section 5.3: KnoVA Process Model
	GR2: Structured Process	
DI 2: Analysis Traceability	GR3: Contextual Model	Section 5.4: KnoVA Meta Model
DI 3: Knowledge Extraction	GR4: Knowledge Model	Section 5.5: Knowledge Items Algorithms 1 & 2
	GR5: Knowledge Abstraction	
DI 4: Knowledge Application	GR6: Knowledge Application	Section 5.6: Algorithms 3 & 4

Table 5.1: Relationship between Design Implications and Concepts.

The four concepts sections are each structured into three subsections: a subsection for problem definition, for the respective concepts and a subsection on how the concepts fit into the RA. Thus, the fundamental structural architecture is extended stepwise to form the KnoVA RA.

In the third part, a conclusion is drawn and the introduced concepts are evaluated. In section 5.8 the research challenges and the research question that were introduced in 1.1 are revisited and it is discussed how the KnoVA RA faces these challenges answers the research question. After this the KnoVA RA is compared to the existing approaches based upon the criteria for comparison that were identified in section 2.4.

## 5.2 Fundamental Structural Architecture of VA Systems

A common architectural style in software architecture is the layered architecture [Pre09]. Applications with user interfaces are commonly structured into three layers in this architectural style. This is also reflected by various architectural pattern for interactive applications such as the model-view-viewmodel pattern or the model-view-presenter pattern [Gar11]. As VA systems are applications with user interfaces a general, architecture for VA systems also incorporates these three layers. However, the above-mentioned patterns define specific details of the implementation such as the communication between the layers. To be applicable in a broad variety of scenarios independent from specific implementation details a fundamental architecture for VA systems needs to abstract from these. Therefore, the fundamental architecture here is derived from a high level examination of the layers of VA systems.

Visual Analytics is defined by Keim as the task of *analytical reasoning facilitated by*



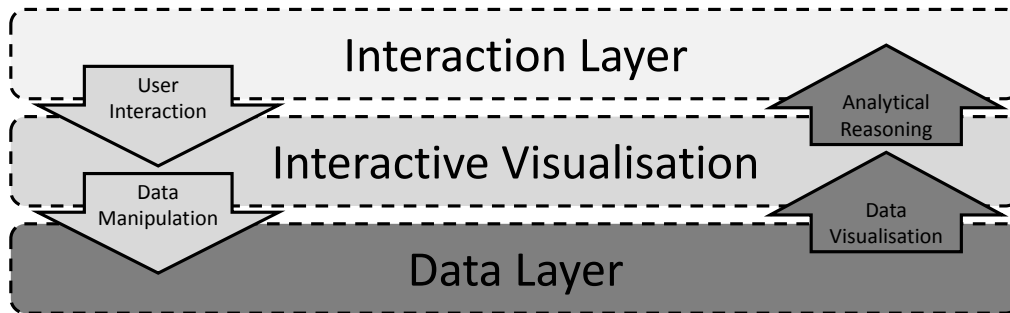


Figure 5.1: Fundamental Structural Architecture of VA Systems.

*interactive visual interfaces* [KMSZ09]. This definition already leads to the key objective of VA systems: analytical reasoning. In philosophy analytical reasoning refers to the analysis of sentences or statements based upon the information in the statement alone [Pal87], by applying the background knowledge of the analysing person about the interpretation of the words. This can be transferred into the domain of VA, where data is analysed based upon the information in the data alone with the background knowledge of the analyst about the interpretation of the data. The first fundamental element of VA systems is the data that needs to be analysed. Thus, translated to a structural architecture of VA systems a data layer can be identified as important element.

According to the definition above in VA systems, the analytical reasoning is facilitated by interactive visual interfaces. Thus visual interfaces to represent the data are utmost important in VA systems. For the fundamental architecture, therefore a layer for interactive visual interfaces can be identified above the data layer.

In the definition of VA the interactiveness of the visual interfaces is emphasised. In Keims VA process four steps are defined: data, visualisation, models and knowledge. In short it can be summarised that according to this process data is visualised which is then used to build models and to derive knowledge. The first steps in this process are already reflected with architectural elements.

The latter two steps refer to user who works with the VA system. VA aims to take advantage of human strengths such as flexibility, creativity, intuition and contextual knowledge in the analysis [Sch07]. For this, the user has to interact with the visual interfaces, where the representation of the data changes according to the interaction. The visual representation of the data eventually leads to insight. Translated to structural architecture it is therefore necessary to represent the user interaction as layer on top of the visual interfaces.

In figure 5.1 an overview of this fundamental architecture for VA systems can be seen. The elements of this architecture are inevitably present in any VA system. This architecture will be gradually extended in the following four sections and lead to the design of the KnoVA RA for knowledge-based VA systems. This RA consists out of four elements, which correspond to the four design implications. It fulfils the general requirements that

were identified following each of the four design implications. For this in each section at first the problems to be addressed are explained. This problem definition is then followed by detailed solution statements, which in summary built the main contribution of this thesis.

### 5.3 Requirements for the KnoVA RA

In chapter 3 detailed requirements for the two application scenarios introduced in chapter 1 were collected. From these requirements generalised implications for the design of knowledge based VA systems have been developed in section 3.3. Below these implications for design are revisited and general requirements for a RA for knowledge-based VA are identified.

**Design Implication 1 – Analysis Process:** VA systems have to support the analysis process. For this, next to meaningful visual abstractions and representations, VA systems needs to provide a tangible and structured support of the analysis process, in which the analyst is aware and in control of multiple reoccurring and varying analysis steps.

⇒ Two general requirements towards a RA for knowledge-based VA an be identified here:

**GR1 – Various Visualisations:** The RA needs to be suitable to create VA systems, which allow the integration of meaningful visual abstractions and representations. Even though the actual visualisation methods are often domain specific special cases, the general requirement here is, that the RA needs to support the integration of various different visualisation methods.

**GR2 – Structured Process:** The second requirement that can be drawn from this is the support of a structured analysis process. The RA needs to provide a procedural model to describe the steps of the analysis process in a way that these steps can be used in VA systems that support direct manipulation of analysis steps.

**Design Implication 2 – Analysis Traceability:** VA systems have to support a traceability of the analysis process, allowing experts to reason which steps they have taken. Hereby it is not only necessary to provide a trace of the course of the analysis in an accounting way. Rather a support for controlling mechanisms, which allow modifications of the analysis process (e.g. modifications of former analysis steps) is necessary.

⇒ The general requirement that can be drawn from this design implication is, that a RA needs to support a traceability of the analysis steps.

**GR3: Contextual Model:** It is not only sufficient to provide a procedural model of analysis steps. Also a contextual model of the state of the analysis system is needed. This contextual model needs to express all relevant elements of the analysis system, in order to provide a comprehensive trace of all information that is relevant for the analyst to control and modify specific parts of the trace.

**Design Implication 3 – Knowledge Extraction:** VA systems have to represent the knowledge that is applied by the analyst in order to be able to describe the applied knowledge. Thereby it is inevitable that a rich set of methods is provided to allow the experts to mark and extract valuable knowledge into a knowledge-base, in order to use it in deviant analysis situations and that the knowledge in the knowledge-base is generalisable and therefore applicable in other analysis situations.

⇒ This design implication leads to the general requirement for an integrated knowledge model for the RA.

**GR4: Knowledge Model:** The RA needs to support the creation of applications which allow to mark and extract valuable knowledge into a knowledge-base. Therefore the knowledge model needs to enable the representation of the knowledge in a structured way, so it can be extracted into a knowledge-base.

**GR5: Knowledge Abstraction:** Another general requirement that can be drawn from this design implication is, that the embodiment of the knowledge model has to account for an abstraction of knowledge, which allows to algorithmically reason if knowledge is generalisable.

**Design Implication 4 – Knowledge application:** VA systems have to provide methods to re-apply and share knowledge that was previously extracted or that was applied in a previous analysis step. For this functionality it is vital to identify whether the knowledge actually is applicable in a given situation.

⇒ This leads to the general requirement for knowledge application.

**GR6: Knowledge Application:** The RA has to provide concepts to re-apply knowledge that was extracted into a knowledge-base as proposed following design implication 3. Based upon this concept the RA needs to support the creation of applications for automatic or semi-automatic knowledge application, based on an algorithmic evaluation of the current analysis situation.

The general requirements identified above need to be addressed in the design of the KnoVA RA. The design is developed in the following four sections 5.4 – 5.7. These four sections are structured and named according to the design implications.

## 5.4 The Analysis Process

Visual Analytics is an iterative process in which visualised data is manipulated by user interaction until insight is generated, which can then be re-applied to the process. The user adapts the course of the analysis according to the analysis question at hand, which defines the analysis process dynamically. Varying visualisations have to be integrated into the analysis system and have to be linked to each other. This process can encompass varying data sources, for instance various trace files, as explained in the first motivating scenario. Moreover, the VA systems have to adapt to various workflows, depending on

the individuality of the users. This dynamic characteristic of the VA process needs to be reflected by the RA. However, a number of problems arise, when the dynamic nature of the analysis process is expressed in a generic model.

#### 5.4.1 Problem definition

VA systems have to provide a profound flexibility. Components integrated to the system and their interaction with each other needs to be dynamically reconfigurable so that the analyst working on the system can retrofit the system to the specific needs of a certain analysis task.

The approach of the KnoVA RA is to create a model of the analysis process that reflects this characteristic and, therefore, can be used as the foundation of a broad variety of VA systems. To cover this approach, two sub problems have to be solved:

1. *Categorisation of the Elements of the Analysis Process*: In order to create a meta data model for the analysis process, a definition of the elements which occur in the analysis process in VA systems is needed. To approach this, typical elements of the VA process need to be identified.
2. *Description of the Analysis Process*: Once the elements of the process are defined, it is possible to create a description of the analysis process based upon these elements.

To address these problems, the following section identifies typical elements of the analysis process and provides a description of the process model.

#### 5.4.2 The KnoVA Process Model

In figure 4.2 an exemplary analysis path can be seen. In the following, this exemplary analysis path is investigated and general properties of the analysis path of VA systems are identified.

The example path can be read from left to right, arrows indicate the data flow. The first element in this path is a data source.

The next example in this path is a selection menu, in this case a stacked half-pie menu. The stacked half-pie menu is followed by two further steps of visualisation: thematic maps of Lower Saxony. This corresponds to the layer for interactive visual interfaces in the structural architecture.

One of the thematic map visualisations is followed by a filtering step. The filter in this example was defined by the user and could, for instance, narrow down the data range - i.e., perform a selection. In the example then this selection is displayed in the last step a pie chart visualisation. In this example process, three different classes of steps can be identified:

- source states, which represent data sources,
- transformation states such as the filter, which modify data and

- visualisations states, in which data is displayed.

Figure 5.2-A shows an abstract representation of the exemplary analysis path shown in figure 4.2. In this figure, the elements of the analysis path are replaced by abstract elements indicating the type of the respective element: data source, visualisation or transformation. As illustrated elements can have more than one type. These abstract elements are typical for VA systems and are defined as follows:

**Term (Source State).** A **Source State** provides access to the data to be analysed. Data sources typically have a data model (e.g., the relational model). In the analysis process, a data source is typically integrated at the beginning of the visualisation process.

**Term (Visualisation State).** In a **Visualisation State**, the input data is rendered to a meaningful visual representation. Here, the actual visual mapping (see chapter 2) is being performed.

In the exemplary analysis path shown in figure 4.2, the scatterplot, the thematic map and the pie chart are visualisation states. A special kind of visualisation state is the stacked half-pie menu. In this menu the structure of the data (the hierarchy of dimensions and their elements) are being displayed as well as the selection path through the hierarchy.

**Term (Transformation State).** In a third class of states in the analysis process, the data is transformed. These **Transformation States** define functions on the data. Examples are filters, selection menus.

In the exemplary analysis path the filter and the stacked half-pie menu are transformation states.

In VA exploration, the user defines the visualisation path by his interactions with the system. The result is a number of analysis states, where the transition between two states is triggered by user interaction. This is illustrated in figure 5.2-B, where analogue to the exemplary analysis path the underlying graph of typeless states is shown. In the figure, each state is combined with a data model. This is because each state represents a specific section of the data to be analysed. In the exemplary path in figure 4.2, for instance, the first state, the data source subsumes all data to be analysed. In the second state, a selection on this data is defined by the stacked half-pie menu, which is then visualised in the third and fourth state and so forth. Therefore, an analysis state can be defined based upon the analysis parameters.

**Definition 1 (Analysis Parameters).** Let  $A$  be the set of attributes of a data source  $DS$ . The set of **Analysis Parameters**  $AP$  consists out of tuples  $ap_i = (a_1, \dots, a_n)$  of data with values  $a_1, \dots, a_n \in A$  and  $n \in \mathbb{N}^*$ .

**Definition 2 (Analysis State).** Let  $AS$  be the set of **Analysis States**. An analysis state  $as := \{p_1, \dots, p_m\} \in AS$  is defined a set of analysis parameters  $\{p_1, \dots, p_m\} \in AP$  with  $m \in \mathbb{N}^*$  which are valid in the analysis state  $as$ .

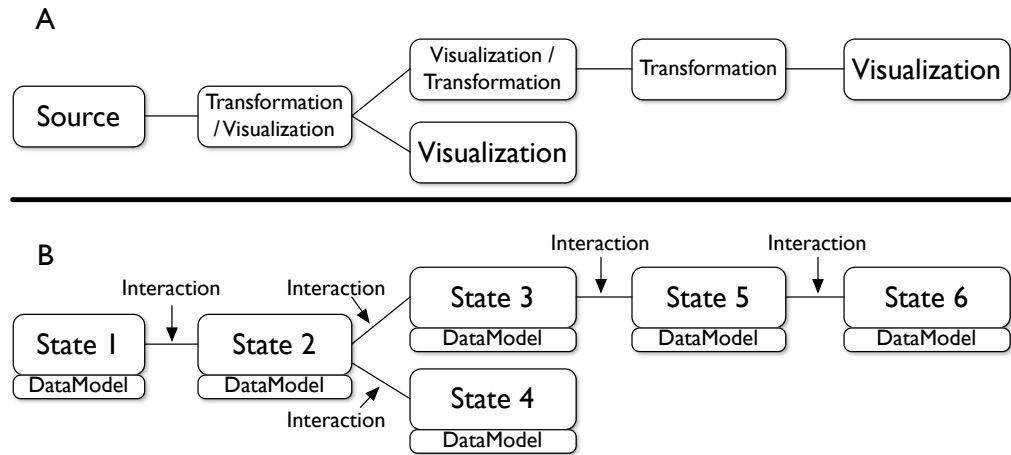


Figure 5.2: Abstract Representation the Analysis Path shown in Figure 4.2: A) Classes of Elements in the Path. B) State Changes by User Interaction.

This definition of analysis states is comparable to the definition of data-states in Chi's and Riedl's data-state model [CR98]. The data-state model aims to model the process of visualisation transformation. Therefore, mapping of values and visualisation operations is created which describes how the data is transformed in order to be visualised. The data-state model alone, therefore, is not sufficient to model the VA process.

As illustrated in figure 5.2, every state in course of the analysis (with the exception of the final state) has one or more successors. Likewise, every state (with the exception of the first one) has exactly one predecessor. As mentioned above, the transition between to states is initiated by the user who interacts with the analysis system and applies his knowledge. This results in a sequence of state changes which together represent the analysis process. Accordingly, a model for the analysis process has to include a definition for these analysis steps.

**Definition 3** (Analysis Step). An *Analysis Step*  $\sigma$  is a tuple  $\sigma := (as_k, \tau)$  of an analysis state  $as_k \in AS$  and a set of successors  $\tau := \{as_{k+1}, \dots, as_{k+l}\} \in AS^k$  with  $k, l \in \mathbb{N}$ . All analysis steps  $\sigma$  form the set of analysis steps  $\Sigma$ .

There are two possibilities to model analysis steps: state-based and transition-based. In the state-based model all states are saved. In the transition-based model, in contrast, only the initial analysis state with its section of the data is saved, accompanied with a set of state changing operations that lead to the final state. This transition-based definition has the advantage, that only the initial data state has to be saved and, therefore, it is efficient concerning memory consumption. There are two problematic issues with this model though: first it is difficult to predict all possible state changing operations and, therefore, difficult to define a comprehensive model of operations and second it is difficult to make comparisons between two states. To compare two states all operations in the chain that

lead to the states have to be evaluated in order to re-create the states.

Because of these issues, therefore, a state based model of analysis steps is preferred in this thesis. In definition 3 accordingly a step consists out of a specific analysis state and its immediate successors. Based upon this definition, the complete analysis process can be defined as the ordered set of all analysis steps.

**Definition 4** (KnoVA Process Model). *The KnoVA Process Model KPM is an ordered set of analysis steps  $KPM := \{\sigma_h \in \Sigma \mid \sigma_h < \sigma_{h+1} \forall h \in \mathbb{N}\}$*

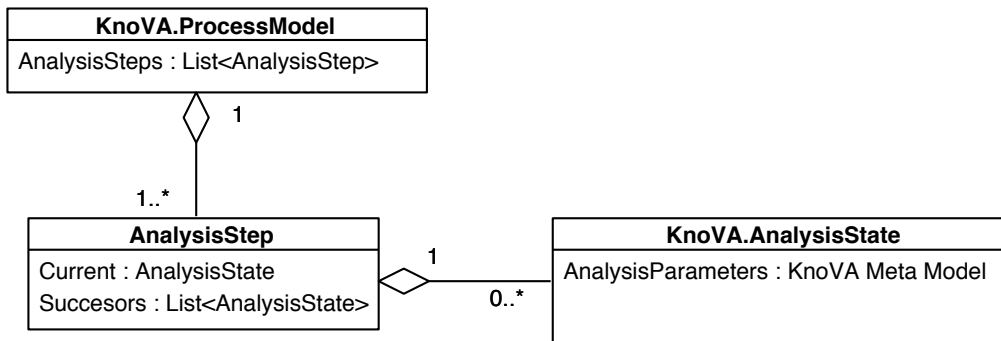


Figure 5.3: UML class diagram of the KnoVA Process Model.

Figure 5.3 illustrates how this process can be represented as UML class diagram. The figure shows three classes: the class KNOVA.PROCESSMODEL, the class KNOVA.ANALYSISSTEP and the class KNOVA.ANALYSISSTATE.

The class KNOVA.PROCESSMODEL holds a reference to a list of one or more KNOVA.ANALYSISSTEP objects in the property ANALYSISSTEPS and thus represents the analysis process according to definition 4. In this definition it is proposed that the analysis process is an ordered set.

Each KNOVA.ANALYSISSTEP has two properties, CURRENT and SUCCESSORS. The property CURRENT gives access to a KNOVA.ANALYSISSTATE which represents the current analysis step. The property SUCCESSORS gives access to a possibly empty list of succeeding analysis states.

The class KNOVA.ANALYSISSTATE represents an analysis state as defined in definition 2. It defines a property ANALYSISPARAMETERS as an reference to an instance of the class KNOVA.METAMODEL. This represents the analysis parameters following definition 1 and thus the data model of the analysis process. The KnoVA MM will be introduced in section 5.5.

### 5.4.3 Structural Architecture including the KnoVA Process Model

Figure 5.4 shows how the KnoVA PM extends the structural architecture of VA systems that was introduced in section 5.2. In the structural architecture data from the data layer

is visualised in the interactive visualisation layer. User interaction on the interactive visualisation results in data manipulation on the data layer.

The KnoVA PM builds an additional layer between the data layer and the interactive visualisation layer. In this layer, the sequence of analysis states is shown. The analysis parameter sets from the data layer are mapped to a system state. The figure illustrates this as data binding between the data layer and the analysis state. After this, the analysis states are visualised in the interactive visualisation layer.

Interaction leads to a manipulation of the state which is then propagated to the data layer in figure 5.4 this is referred to as data manipulation. This propagation of the manipulation eventually leads to a new analysis state in the KnoVA PM layer.

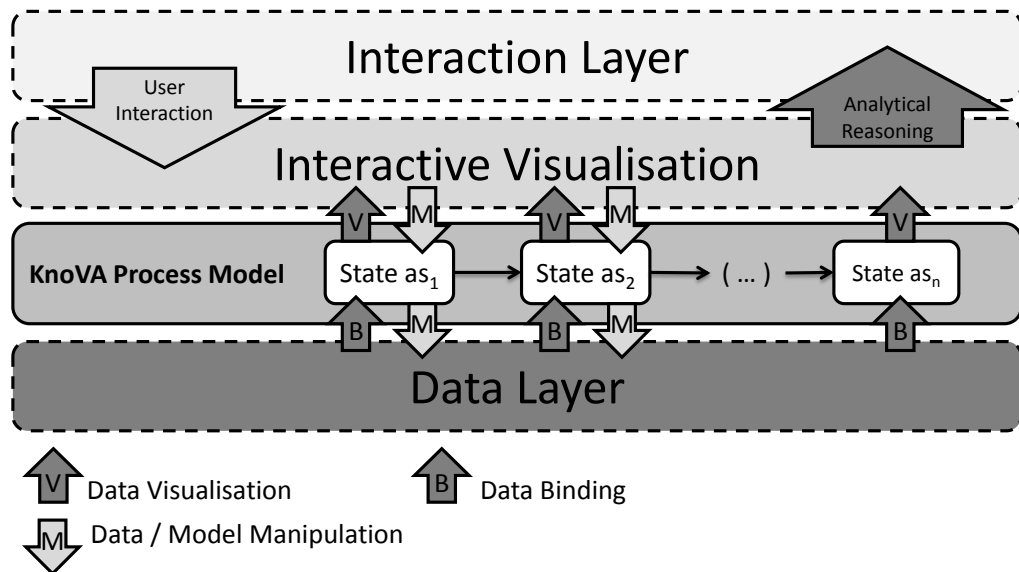


Figure 5.4: KnoVA Reference Architecture including the KnoVA Process Model.

## 5.5 Analysis Traceability

In this section the KnoVA meta model is introduced to extend the structural architecture with a meta data model that can be used to create a contextual model.

Interaction with the analysis system can trigger state changing operations based upon the knowledge applied by the user. In this way the VA process can be seen as a sequence of analysis steps, where between two steps knowledge is applied by user interaction with the visual interface. This is illustrated in figure 5.2.

The second design implication was, that VA systems need to support a traceability of the course of the analysis. A generic approach for this traceability therefore has to be integrated in the RA.



### 5.5.1 Problem definition

With the KnoVA process model it is possible to describe the course of the analysis on a procedural level. It is therefore suitable to describe *which* steps occur in the analysis. In figure 5.2 it is illustrated, that every state has its own data model, representing the currently visualised data. This data model describes the actual state of the analysis step. The KnoVA process model does not include such a data model and hence the RA so far does not provide a contextual model, which is necessary to respond to the general requirement [GR3]. That contextual model allows for the traceability of the analysis process by defining *what* the actual state of the analysis system is. To approach this, the following sub problem needs to be addressed:

1. *Development of a Meta Data Model:* The KnoVA taxonomy describes the design space and the properties of VA systems. Therefore the taxonomy can serve as the foundation of a meta data model of VA systems, in order to fulfil the general requirement [GR3] for a contextual model.

In the next section the KnoVA MM is developed. For this it is discussed which elements of VA systems are necessary to describe the state of an analysis system in order to derive knowledge and how the elements of the KnoVA taxonomy can be translated into a meta data model.

### 5.5.2 The KnoVA Meta Model

To explain the intended purpose of the KnoVA meta model a short example for the application of expert knowledge in the Mustang platform is introduced. In the example it is shown how knowledge that is applied in a VA system leads to a change of the actual state between to analysis steps.

After the example the KnoVA MM is developed based upon the KnoVA taxonomy. The KnoVA MM is presented in an UML based notation. It features all elements of the KnoVA taxonomy and therefore abstracts from the explanatory example. To complete the subsection about the KnoVA MM it is shown how the model can be applied.

#### Example for the Application of Expert Knowledge

Figure 5.5 shows an exemplary analysis path in the Mustang VA system. In this example, epidemiological datasets are displayed. The data sets consist of four variables, a geographic region, and associated to each region as a functional dependent measure, the the absolute number of incidences in the respective region (absolute value), the incidence per 100.000 inhabitants (raw incidence), and the incidence normalised to the age structure (age distributed incidence).

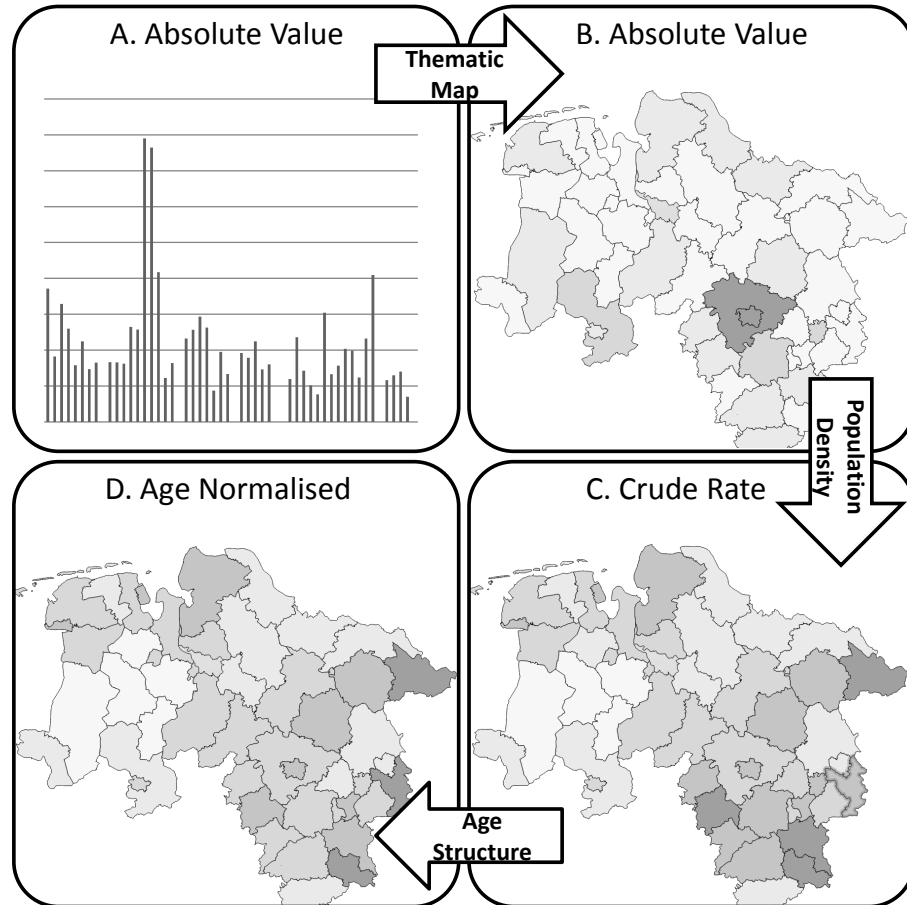


Figure 5.5: Example of Knowledge Application in Mustang.

In the exemplary path shown in figure 5.5 four analysis steps are named: A. absolute value of the measure in a bar chart visualisation, B. absolute value in a thematic map visualisation, C. raw incidence in a thematic map visualisation and D. statistically enriched incidence in a thematic map visualisation. The complete underlying data set can be seen in appendix A.1.

The exemplary path illustrates a number of different kinds of knowledge applications. At first, the data set is visualised in a bar chart visualisation (figure 5.5-A). The values are mapped to a cartesian coordinate system. In this example, the geographic region is mapped to the abscissa (x-coordinate) as a nominal value and the functional dependent measure is as absolute value is mapped to the ordinate (or y-coordinate). With the background knowledge that geographically distributed values are probably better displayed in a thematic map, the user then changes the the visualisation to a thematic map of Lower Saxony where the absolute value of the measure is mapped to a colour.

The geographic region of the data has to be mapped to the regions in the map and the numerical measure has to be mapped to a range of colours. The result can be seen in figure

5.5-B. The colour coding here can either be user-adjusted or auto-adjusted, depending on the implementation of the thematic map. In this step, therefore, the knowledge that was applied was to exchange the bar chart with a thematic map because the data is geographically distributed.

In the example, high values are tinted in darker hues and lower values in brighter hues. In 5.5-B, the region in and around the federal state capital Hannover stands out with the darkest colour. An analyst could interpret that the incidence in this region is extraordinary high. However, the thematic map here gives a false impression. A higher incidence should be expected in regions with a higher population count.

To reflect this circumstance, the analyst modifies the thematic map. In figure 5.5-C, the absolute value is replaced by the raw incidence. The raw incidence is the number of cases normalised to the population density. The knowledge that was implicitly applied here is that the population density varies with the geographic region and, hence, the measure has to be modified to reflect this. Finally, in figure 5.5-D another example of implicitly applied expert knowledge can be seen. Here, the age distribution of the population is reflected. Age, like population density, varies with the geographic regions. The background knowledge that was applied here is that the incidence varies over age and people are not equally distributed regionally according to their age.

These are examples of how implicit expert knowledge is applied in the analysis process. By the application of knowledge in between two analysis steps, the analyst changes the actual state of the analysis system and, hence, the internal data model of the analysis system. The next sections shows how the KnoVA taxonomy can be used to create a meta data model for VA systems. After that it will be shown how the MM can be used to represent the state changes that occur on the application of knowledge.

### Introduction of the KnoVA Meta Model

The direct manipulation paradigm [Shn83] describes how the interaction with an analysis system is based upon direct interaction of the user with the visual interface. This results in changes to the visual interface. Carried over to VA, a direct manipulation results in a change of the visualisation. Although, as described above, the change has to be reflected in the underlying data model of the VA system, only the change of the visualisation is observable by the user. Therefore, it can be judged that any change that occurs in a VA system can be expressed by changes in the properties of the visual interface.

The basic idea that leads to the development of the KnoVA MM is that a data model to reflect the changes in the actual states of VA systems only needs to reflect the properties of the visual interface. Thus, the data model can be built on a comprehensive set of properties that describe the design space of VA systems. The classification of VA systems presented in section 4.3, therefore, provides the necessary foundation for such a data model. In this section it is shown how the classification can be used to derive the KnoVA MM.

All of the properties identified in section 4.3 can be used to describe the composition of a VA system or of a visualisation method. This is illustrated on the example of the thematic map in figure 5.5-B. The thematic maps can display two dimensions: the geographic re-

gions and a functional dependent measure. Depending on the specific implementation of the map, a geographic region could be either a nominal value or geographic coordinates. In a data source, the geographic regions may also be encoded in different ways. Hence, a mapping between the geographic regions of the thematic map and the geographic coordinates has to be established to create the visualisation.

In the following sections the meta model will be developed based and it will be shown how the structural architecture for VA systems is extended by the meta model. An example illustrates how the meta model can be used to represent state changes in VA systems. The complete KnoVA MM can be seen in an UML class diagram notation in figure 5.6. Starting point for the creation of the KnoVA MM are the properties of VA systems. Above their grouping into classes the classifying properties identified in section 4.3 can be divided into different groups based upon their relevance for the meta model. Three groups can be identified: Properties concerning the actual state of the VA system, properties concerning state changing operations and descriptive properties.

In figure 5.6 the elements of the meta model, which represent the properties that describe the state are shaded in dark grey. Elements in the diagram with no background shade relate to state changing operations and elements with a light grey shading represent descriptive properties. The properties concerning the current state of the VA system are all those properties, which can be used to describe which data is visualised and how this data is visualised. Properties concerning state changing operations are the properties that describe operations that lead to changes in the current state of the analysis system. Descriptive properties are those properties which can be used to describe a transient or general aspect of the analysis system which is independent from the current state. The differentiation into these three groups is important in the following paragraphs because according to their group the transformation of these properties into the meta model varies.

The top most class in the diagram 5.6 is KNOVA META MODEL. The complete class hierarchy is described in the paragraphs below.

### Data To Be Visualised

Visualisation methods in VA systems represent either a single or multiple analysis parameters. These parameters are mapped to different visual elements. This mapping is specific for every visualisation method. The meta model, therefore, has to abstract from this specific mapping in order to be applicable across different VA systems and visualisation methods.

In definition 1, analysis parameters were defined as tuples of attributes of the data. This definition is sufficient for the definition of the analysis process. However, the analysis of existing VA systems in has shown that data occurs with complex data types and functional dependencies between values or attributes. The definition of analysis parameters does not sufficiently reflect these complex data structures and relationships. The attributes in the class data to be visualised of the KnoVA taxonomy can be used to describe these complex data structures and relationships.

In order to achieve an abstraction from the specific mapping between analysis para-

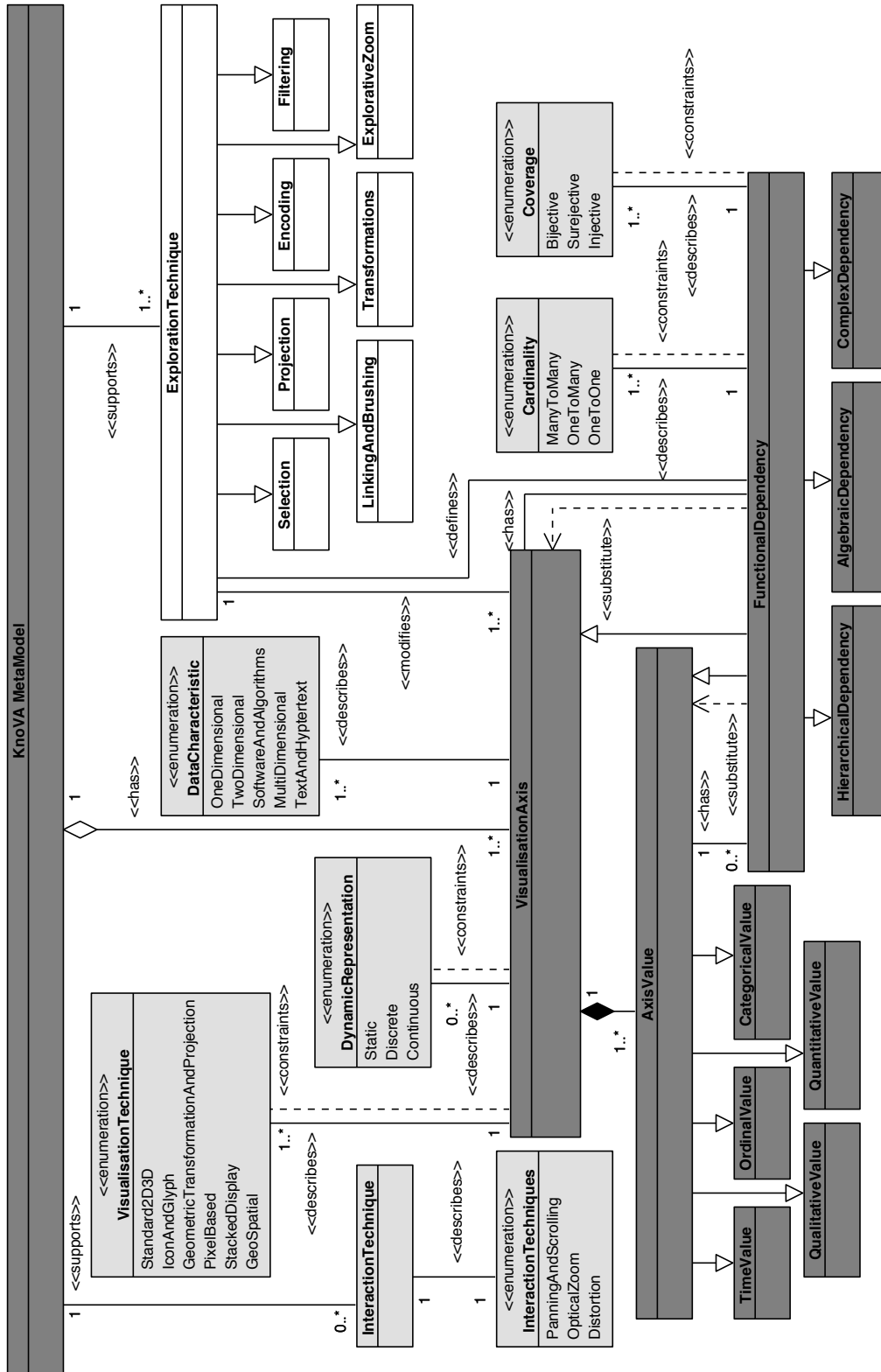


Figure 5.6: UML class diagram of the KnoVA Meta Model.

meters and the visual elements of the visualisation methods the concept of visualisation axes is introduced in the KnoVA MM. In figure 5.6, a class VISUALISATIONAXIS can be seen. Visualisation methods can have multiple axes of visualisation, this circumstance is reflected by the one-to-many aggregation between KNOVA META MODEL and VISUALISATIONAXIS. Each visualisation method has to provide at least one axis of visualisation.

As an example, the bubble-chart in the TaP system provides five axes of visualisation: the geometric x-axis and y-axis, the colour axis, the size axis and the animation axis. Each visualisation axis can visualise a set of values from the underlying data source. To reflect this, the meta model incorporates the class AXISVALUE. In section 4.3 it was examined that data to be visualised occurs in different data orders: quantitative, qualitative, ordinal, categorical, and timely.

To reflect these orders, AXISVALUE is modelled as a generalisation of five distinct classes which are named accordingly. It is important to notice that this results in a polymorphism of AXISVALUE. Thus it is insured that one axis can be used to represent multiple types. In the TaP example, for instance, the x-axis and y-axis are suitable to visualise data in any of the five orders. The size axis can be used to visualise quantitative and ordinal values only. The colour axis can be used to visualise qualitative values, ordinal values, categorical values, and the animation axis finally can be used to visualise timely values only. The class AXISVALUE and its super-classes combined provide a model for the subclass data order of the KnoVA taxonomy.

As another subclass of the class data to be visualised in the KnoVA taxonomy the subclass data structure is defined. The subclass includes properties to describe functional dependencies in the data. To reflect these functional dependencies, the class FUNCTIONALDEPENDENCY is introduced in the meta model. This class is polymorph to three superclasses HIERARCHICALDEPENDENCY, ALGEBRAICDEPENDENCY and COMPLEXDEPENDENCY, comparable to the polymorphism of AXISVALUE and its superclasses.

Functional dependencies in visualisation methods can be defined upon values or upon visualisation axes. To reflect this in the model associations between FUNCTIONALDEPENDENCY and VISUALISATIONAXIS are included. Examples for functional dependencies on values are statistical or mathematical methods which manipulate the data. Quantitative values, for instance, could be grouped to categorical values. In this case, the functional dependent values substitute the original values, which corresponds to the substitute constraint between FUNCTIONALDEPENDENCY and AXISVALUE included in the meta model.

Comparable functional dependencies are defined on VISUALISATIONAXIS, which will then be valid for all visualised values. These values also substitute the original values and, therefore, the substitute constraint is modelled between FUNCTIONALDEPENDENCY and VISUALISATIONAXIS. Functional dependencies can be defined on the values alone or include additional data sources, depending on a specific implementation. In addition to the functional dependencies on VISUALISATIONAXIS can also define hierarchical dependencies between elements, thus defining whether the data can be spe-

cialised or generalised along a visualisation axis. In the TaP system, this kind of functional dependency is defined on the x-axis and the y-axis where gestures can trigger and OLAP operations to drill-down or roll-up along the data hierarchies.

Another type of functional dependencies on VISUALISATIONAXIS constraints the binding between the underlying database and the axis. For instance, a functional dependency could define the constraint that one VISUALISATIONAXIS is dependent on another VISUALISATIONAXIS. In the TaP example the functional dependencies between the visualisation axes define that the presence of at exactly one dimension (or ordinal value) is mandatory in every bubble-chart. In addition it is necessary to provide at least one measure (or quantitative value) in order to generate a visualisation.

With FUNCTIONALDEPENDENCY, AXISVALUE, VISUALISATIONAXIS and the associated classes, the properties concerning the current state of the analysis system are outlined. In figure 5.6 these classes are shaded in dark grey. However, not all aspects of the class data to be visualised from the KnoVA taxonomy are represented with these classes in the KnoVA MM. FUNCTIONALDEPENDENCY inherits from VISUALISATIONAXIS and AXISVALUE in order to be polymorphic to each of these classes. This enables FUNCTIONALDEPENDENCY to substitute either of these classes and also enables AXISVALUE to substitute VISUALISATIONAXIS by a functional dependency, which is necessary for instance to realise hierarchical dependencies.

The subclass data characteristic and the categories cardinality and coverage from the subclass data structure define descriptive properties. In the KnoVA MM, these are indicated as light grey shaded enumeration classes. The reason for this design decision is that these properties do not describe the state of VA system itself. They rather describe aspects of the VA system on another semantic level. CARDINALITY and COVERAGE describe the embodiment of functional dependencies. In a specific implementation of a VA system with these enumerations the embodiment of a functional dependency can be described either as a constraint or as a feature at runtime. A constraint will limit a certain functional dependency for specific situations (e.g., only applicable in one-to-many situations). A feature allows judgement about the suitability (e.g., can be used for surjective mappings).

The enumeration class DATACHARACTERISTIC is modelled with a describe association only. It is intended to be used to describe the suitability of a certain VISUALISATIONAXIS for a specific DATACHARACTERISTIC. Each VISUALISATIONAXIS can be suitable for one or more characteristics. With the introduction of the enumeration classes for DATACHARACTERISTIC, CARDINALITY and COVERAGE the class data to be visualised from the KnoVA taxonomy is completely represented in the KnoVA MM.

### Visualisation Technique

The class visualisation technique from the KnoVA taxonomy describes six classifying properties. These properties describe the kind of visualisation technique used, respectively the kind of visual mapping that is created. There are some visualisation techniques which are suitable for very specific kinds of data. For instance, geo-spatial visualisation techniques are restricted to geographically distributed data. Visualising

non-geographically distributed data on a thematic map will most likely result in a visualisation that may easily be misinterpreted. Still, the same geographically distributed data can be used in a visualisation that is not specialised for geo-spatial data. In figure 5.5-A, an example for this can be seen. Here, the geographic dimension is mapped to the ordinal y-axis of the bar chart.

This example illustrates that the visualisation technique used can provide relevant information and can in some cases constraint the application spectrum of a visualisation method or of specific axes of a visualisation method. Therefore, comparable to CARDINALITY and COVERAGE an enumeration class VISUALISATIONTECHNIQUE is introduced in the KnoVA MM. A describe association between VISUALISATIONAXIS and VISUALISATIONTECHNIQUE expresses this relationship. In addition to this a constraint is included in the model to express that the application spectrum of some VISUALISATIONAXIS is limited.

### Exploration Technique

In the KnoVA taxonomy the new class exploration technique was introduced in addition to the class interaction technique, which is already present in the VDET. Exploration techniques differ from interaction techniques as exploration techniques lead to changes in the state of the VA systems and therefore are the defining element of the VA process. Exploration techniques, therefore, trigger the application of expert knowledge and lead to state changes in the KnoVA PM.

To reflect this characteristic, the class EXPLORATIONTECHNIQUE is introduced to the KnoVA MM. Every KNOVA META MODEL will support at least one EXPLORATIONTECHNIQUE as otherwise interaction with the VA system is impossible. It can however provide multiple techniques. This relationship is reflected by the one-to-many association between these two classes.

The class EXPLORATIONTECHNIQUE is polymorphic against seven super classes SELECTION, LINKINGANDBRUSHING, PROJECTION, TRANSFORMATION, ENCODING, EXPLORATIVEZOOM and FILTERING. This is introduced into the KnoVA MM for the same reasons as the polymorphism of VISUALISATIONAXIS and FUNCTIONALDEPENDENCY in order to allow specific implementations to create combined types.

EXPLORATIONTECHNIQUES results in a modification of a VISUALISATIONAXIS. In the TaP example, gesture based interaction on the x-axis or y-axis leads to an OLAP drill-down or roll-up operation. In the terminology of the KnoVA MM the interaction is an EXPLORATIVEZOOM. Each VISUALISATIONAXIS is modified by one or more EXPLORATIONTECHNIQUES. This relationship is represented by the modifies association between EXPLORATIONTECHNIQUE and VISUALISATIONAXIS. In addition to the modification of VISUALISATIONAXIS an EXPLORATIONTECHNIQUE can also lead to the definition of a functional dependency. FILTERING, TRANSFORMATIONS, LINKINGANDBRUSHING, or ENCODING for instance define a functional dependency on one or more VISUALISATIONAXIS. This relationship is reflected by the defines association between EXPLORATIONTECHNIQUE and FUNCTIONALDEPENDENCY.

The introduction of the class EXPLORATIONTECHNIQUE to represent the state changing



properties of the KnoVA taxonomy completes the KnoVA MM. The respective classes are shaded white in figure 5.6. In the next section the example for the application of expert knowledge that was introduced earlier in this section will be revisited to show how the KnoVA MM can be used in VA systems to reflect the knowledge application.

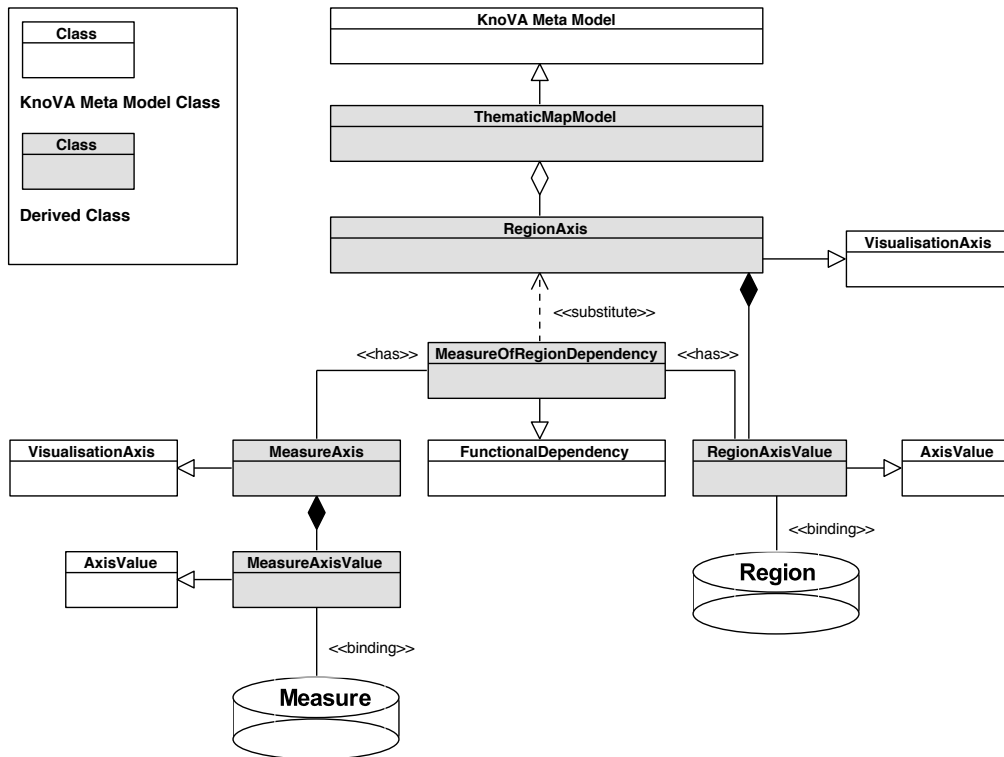


Figure 5.7: Thematic Map Meta Model as Application of the KnoVA Meta Model.

### Dynamic Representation

An enumeration class **DYNAMICREPRESENTATION** is introduced to the KnoVA MM to represent the three classifying properties from the class dynamic representation introduced in the KnoVA taxonomy. Every **VISUALISATIONAXIS** supports one or more of these attributes. Some visualisation methods provide axes which can only be used for animated data. The visualisation axis in the TaP system is an example for this relationship. **DYNAMICREPRESENTATION** also defines a constraint on **VISUALISATIONAXIS**.

### Interaction Technique

The class interaction technique from the KnoVA taxonomy differs from the previous two classes, as the visualisation technique describes a property of the VA systems and not a property of the **VISUALISATIONAXIS**. In addition to this difference, the interaction technique in an implementation is executable code. To reflect this nature a class **INTER-**

ACTIONTECHNIQUE is introduced to the KnoVA MM. Each KNOVA META MODEL may support any number of INTERACTIONTECHNIQUE. Above the class INTERACTIONTECHNIQUE an enumeration class INTERACTIONTECHNIQUES is introduced to describe the type of INTERACTIONTECHNIQUE.

A KNOVA META MODEL can provide interaction techniques but it not necessarily has to. Therefore, the meta model also does not include the possibility to define constraints based on the class INTERACTIONTECHNIQUE or the enumeration class INTERACTIONTECHNIQUES.

With the introduction of the enumeration classes for VISUALISATIONTECHNIQUE and DYNAMICREPRESENTATION and the class INTERACTIONTECHNIQUE with the associated enumeration class INTERACTIONTECHNIQUES all properties of the KnoVA taxonomy which are solely descriptive are represented in the KnoVA MM. All these elements are shaded in light grey in image 5.6. The last group that needs to be examined are properties concerning state changing operations.

### Application of the Meta Model

The UML class diagram shown in figure 5.7 illustrates the application of the KnoVA MM in the Mustang example, that was introduced earlier in subsection 5.5.2. In the figure a specialised thematic map meta model is introduced which is based upon the abstract KnoVA MM.

In the figure the model is limited to the properties concerning the actual state of the analysis system. For clarity purposes the descriptive properties and the state changing properties are not shown in this example. In the figure classes which are part of the KnoVA MM have a white background. Classes which are derived from the KnoVA MM to form the model of the visualisation are shaded in light grey.

The UML diagram in this figure shows how a specialised meta model for the visualisation shown in figure 5.5-B can be implemented based upon the KnoVA MM. In the diagram a class THEMATICMAPMODEL is introduced, which is derived from the class KNOVA META MODEL. The class THEMATICMAPMODEL subsumes all elements of the thematic map meta model.

The thematic map visualisation has two axes of visualisation. The class REGIONAXIS in the UML diagram represents the regional mapping in the visualisation. The thematic map is clustered into a number of geographic regions. Each region is represented by a REGIONAXISVALUE in the thematic map meta model. REGIONAXIS is derived from the class VISUALISATIONAXIS in the meta model. In the diagram REGIONAXIS is composed out of REGIONAXISVALUE objects. A binding association between the class REGIONAXISVALUE and a data source REGION indicates that each REGIONAXISVALUE is bound to a value from the data source.

The other axis of visualisation that is offered by the thematic map visualisation is an axis to visualise a numeric measure. In the thematic map the numeric measures are mapped to colour values and the geographic regions are then shaded in the colour that matches to the value of the measure that is functional dependent to the specific region. In the diagram this axis is represented by the class MEASUREAXIS, which is derived from

VISUALISATIONAXIS and composed from MEASUREAXISVALUE objects, which are derived from AXISVALUE. A binding association between MEASUREAXISVALUE and a data source MEASURE indicates that each MEASUREAXISVALUE is bound to a value from the data source.

The measures are functional dependent on the region. To represent this a class MEASUREOFREGIONDEPENDENCY, which is derived from FUNCTIONALDEPENDENCY from the KnoVA MM is introduced to the thematic map meta model. This class inherits the relationships that are modelled for FUNCTIONALDEPENDENCY, AXISVALUE and VISUALISATIONAXIS in the KnoVA MM. Therefore MEASUREOFREGIONDEPENDENCY can substitute MEASUREAXISVALUE and REGIONAXISVALUE as compositional elements for REGIONAXIS. In this case the functional dependency will map each REGIONAXISVALUE to the matching functional dependent MEASUREAXISVALUE from the MEASUREAXIS. In this case REGIONAXIS will receive matching value pairs as input data to be visualised.

The thematic map meta model can be used to represent the visualisation state shown in figure 5.5-B. The meta model excludes descriptive properties and exploration methods for simplicity. The descriptive properties can simply be modelled as references to the respective enumeration classes, which are instantiated with the valid values in the thematic map meta model. The functional dependency in this case can be characterised with the CARDINALITY One-To-One, as each region is mapped to exactly one measure and with a bijective COVERAGE as there is exactly one measure for every region and exactly one region for every measure.

In the KnoVA PM therefore the thematic map meta model can be used to represent the analysis step shown in figure 5.5-B. It is not sufficient yet to represent the state changes that occur in the introducing example illustrated in figure 5.5. In figure 5.8 the thematic maps shown in 5.5-B (crude rate) and 5.5-C (age normalised) are revisited. Between these two steps knowledge was applied by the user. In this case the knowledge that the age distribution in the geographic regions has to be considered as a factor when examining the incidence.

The figure is structured into two levels, separated by a horizontal bar. On top level the resulting visualisation is shown. On the bottom level the corresponding thematic map meta model is visualised. In 5.8-A for crude rate a simplified version of the thematic map meta model shown in figure 5.7 is shown. In this figure only the class THEMATICMAPMETAMODEL, the two axes, the functional dependency and the data sources are shown. Super classes from the KnoVA MM and AXISVALUE and its derivatives are left out for simplicity.

Upon user interaction the thematic map on the visualisation level changes. The user applied the knowledge that the age distribution over the regions is important. This results in the definition of a new functional dependency AGESTRUCTURE between the MEASUREAXIS and the existing functional dependency MEASUREOFREGIONDEPENDENCY including a matching data source.

The user interaction that leads to the definition of the functional dependency is specific for an actual implementation. In the thematic map meta model a class derived from EX-

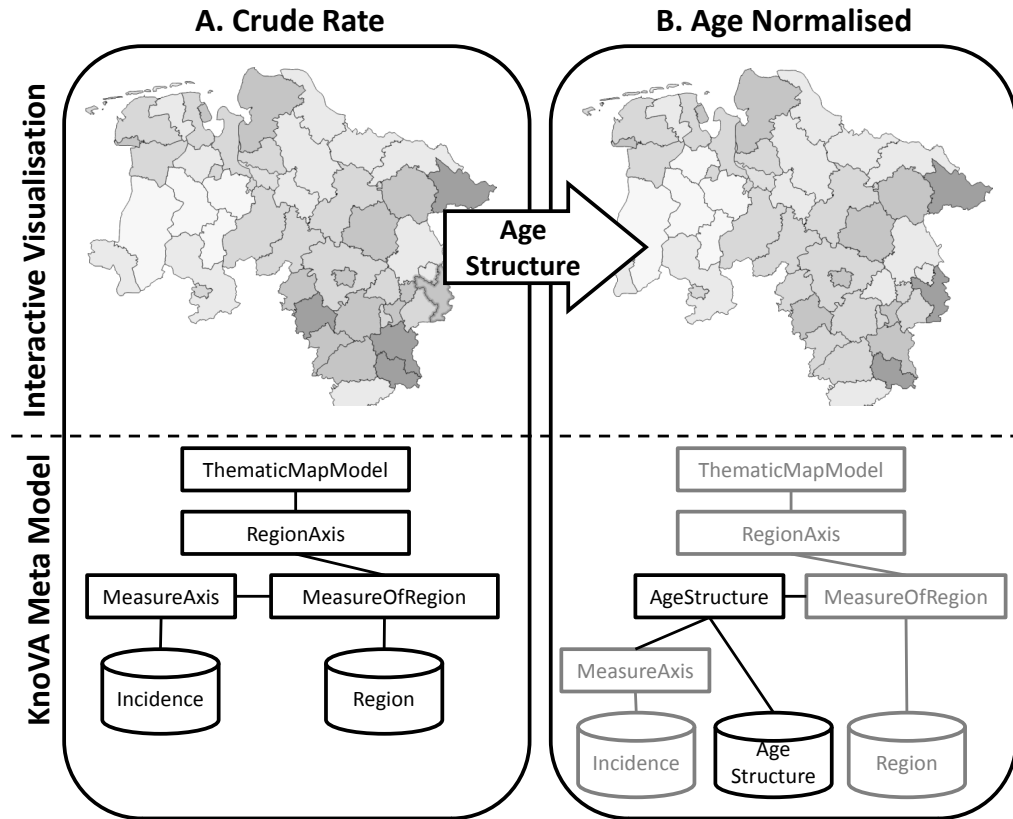


Figure 5.8: Changes in the KnoVA Meta Model upon Knowledge Application.

PLORATIONTECHNIQUE from the KnoVA MM can be introduced to represent the state changing operation that defined the functional dependency. As functional dependencies are polymorphic to VISUALISATIONAXIS the new functional dependency can replace the existing dependency between MEASUREAXIS and REGIONAXIS.

All classes in the thematic map meta model derive from classes defined in the KnoVA MM. This makes it possible to map the structural changes in the thematic map meta model to changes in an abstracted equivalent KnoVA MM.

Thus knowledge application that results in a change of the thematic map meta model can be represented by a change in the KnoVA MM. After this example for the application of the KnoVA MM, it will be shown how the KnoVA MM enhances the structural architecture in the following subsection.

### 5.5.3 Structural Architecture including the KnoVA Meta Model

Figure 5.9 illustrates how the structural architecture introduced in section 5.2 and enhanced by the KnoVA PM in section 5.4.2 is enhanced by the KnoVA MM.

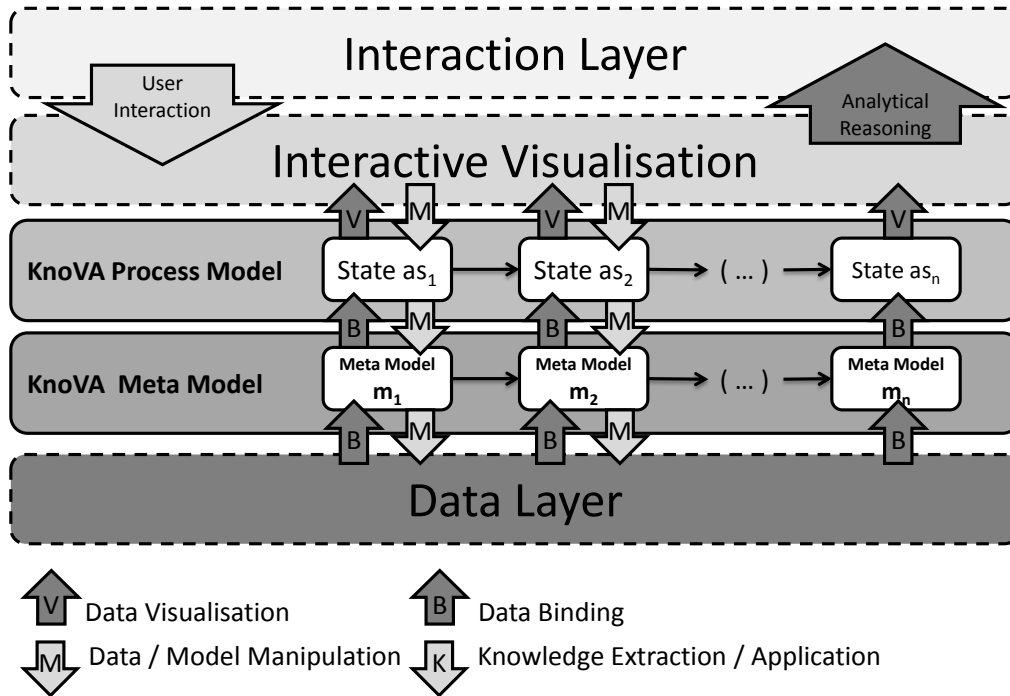


Figure 5.9: KnoVA Reference Architecture including the KnoVA Meta Model.

The KnoVA MM forms an additional layer between the KnoVA PM and the data layer. In this layer for each state  $as_1, \dots, as_n \in AS$  in the process model a corresponding instance of the respective meta model  $m_1, \dots, m_n$  exists.

**Definition 5** (Meta Model Instance). A *Meta Model Instance*  $m_{as_i}$  is a specific instance of the KnoVA MM valid in the analysis state  $as_i$  with  $i \in \mathbb{N}^*$ . All meta model instances form the set  $M$ .

Based upon the KnoVA MM the definition of analysis states 2 can thus be refined to a definition of reference analysis states:

**Definition 6** (Reference Analysis State). A *Reference Analysis State* is a tuple  $\rho := \{as_i, m_{as_i}\}$  consisting out of an analysis state  $as_i \in AS$  a corresponding KnoVA MM instance  $m_{as_i}$ . All reference analysis states  $\rho_i$  form the set  $\mathcal{RS}$ .

In contrast to figure 5.4 now the analysis parameters from the data layer are mapped to an instance of the meta model. In this figure this is illustrated as a data binding between the data layer and the meta model instance. Accordingly the parameters of the meta model are mapped to a system state on the process model layer, likewise illustrated as data binding. After this the analysis states on the process model layer are visualised in the interactive visualisation layer.

The KnoVA MM layer creates an abstraction between the data layer and the process model layer and thus allows the states in the process model to be abstracted from the actual underlying data. Knowledge that is applied to the analysis system, leading to state changes, can be described based upon the KnoVA MM. The next section shows how applied knowledge can be represented in the KnoVA MM to be extracted into a knowledge-base.

## 5.6 Knowledge Extraction

In this section a structural architecture is extended by knowledge items, a concept for knowledge representation and by algorithms for the extraction and generalisation of knowledge. These concepts and algorithms respond to the design implication DI3 for knowledge extraction. Thus they face the general requirements [GR4] for a knowledge model and [GR5] for knowledge abstraction. While the KnoVA PM and the KnoVA MM in combination already allow for the representation of knowledge, concepts for the knowledge extraction have yet to be integrated into the RA.

### 5.6.1 Problem definition

To extract the applied expert knowledge into a knowledge-base, the following two sub problems have to be addressed:

1. *Extraction of Knowledge into a Knowledge-Base:* A definition for knowledge has to be provided in order to be able to extract the implicitly applied knowledge into a knowledge-base. For this the knowledge needs to be represented in a structured way.
2. *Generalisation of Knowledge:* A concept for a generalisation of knowledge has to be provided which allows to abstract knowledge applied in a specific state. This is necessary to enable a comparison across states with a comparable representation on this generalised.

In the following two subsections concepts and methods for knowledge extraction based upon the KnoVA PM and the KnoVA MM are introduced and it is shown how this enhances the KnoVA RA. At first in subsection 5.6.2 a formal definition for knowledge items based upon the definition of analysis steps and reference analysis states is given. Further in the same subsection an algorithm for the generalisation of the extracted knowledge items is introduced. In subsection 5.6.3 it is then explained how the concepts for knowledge extraction and are integrated into the KnoVA RA.

### 5.6.2 Concepts for Knowledge Extraction

In order to derive knowledge from the analysis process at first a definition for knowledge in the context has to be made. In section 5.5.2 it was exemplarily shown that knowledge, which is applied implicitly by the user can be represented with the KnoVA MM. Specifically if knowledge is applied in a reference analysis state a state change will occur based

upon the applied knowledge. This results in two reference analysis states with corresponding meta model instances. A comparison of the two meta model instances allows to identify the implicitly applied knowledge. Based upon the definition 6 for reference analysis states a knowledge item can be defined as tuple of two succeeding reference analysis states.

**Definition 7** (Knowledge Item). *Let  $\rho_1 := \{as_1, m_{as_1}\}$  and  $\rho_2 := \{as_2, m_{as_2}\}$  be two succeeding reference analysis states with  $\rho_1 < \rho_2$ . A **Knowledge Item** is defined as a tuple  $\vartheta_{12} := \{\rho_1, \rho_2\}$  of the analysis steps. All knowledge items form the set  $\Theta$ .*

As a knowledge item contains the preceding analysis state with its corresponding meta model instance and the succeeding analysis state with its corresponding instance it implicitly also contains the applied knowledge that lead to the change between the two states.

Figure 5.10 shows an UML class diagram which introduces the architectural elements necessary to include a knowledge extraction functionality in the KnoVA RA. In the figure these elements are shaded grey. In total the figure contains five classes. The classes KNOVA META MODEL, KNOVA.ANALYSISSTATE and EXPLORATIONTECHNIQUE shaded in white are already known from sections 5.4 or 5.5.

The class KNOVA.ANALYSISSTATE holds a reference to an instance of KNOVA META MODEL and represents the reference analysis state as introduced in definition 6. The class EXPLORATIONTECHNIQUE here provides a method TRIGGERSTATECHANGE() to indicate that a state change will occur upon interaction with the analysis system. With the call of TRIGGERSTATECHANGE() the process of knowledge extraction is initiated. To support the knowledge extraction process two classes are introduced in figure 5.10: the class KNOWLEDGEITEM and the class KNOWLEDGEENGINE. The class KNOWLEDGEITEM represents the knowledge item as introduced by definition 7. For this it holds a reference to an INITIALREFERENCESTATE and a SUCCEEDINGREFERENCESTATE, representing the two reference analysis states  $\rho_1$  as initial state and  $\rho_2$  as succeeding state following definition 7.

The class KNOWLEDGEENGINE represents the foundation for the algorithmic backend of the knowledge extraction process. It features the method EXTRACTKNOWLEDGEITEM(). This method is called upon a state change following the call of the TRIGGERSTATECHANGE() method. EXTRACTKNOWLEDGEITEM expects two instances of ANALYSISSTATE as parameter. The initial state and the succeeding state following the notation introduced above. From these states the method will create a KNOWLEDGEITEM and put it into the knowledge-base. This is indicated by the extract knowledge association between KNOWLEDGEENGINE and the knowledge-base illustrated as a data source below the class in figure 5.10.

The whole process of knowledge extraction is summarised in algorithm 1 in pseudo code. In the first line it is evaluated whether a reference analysis state change is triggered. This corresponds to a call of the method TRIGGERSTATECHANGE() of EXPLORATIONTECHNIQUE as shown in figure 5.10. If such a state change is triggered then firstly the initial reference analysis state  $\rho_1 := \{as_1, m_{as_1}\} \in \mathcal{RS}$  is read in line two.

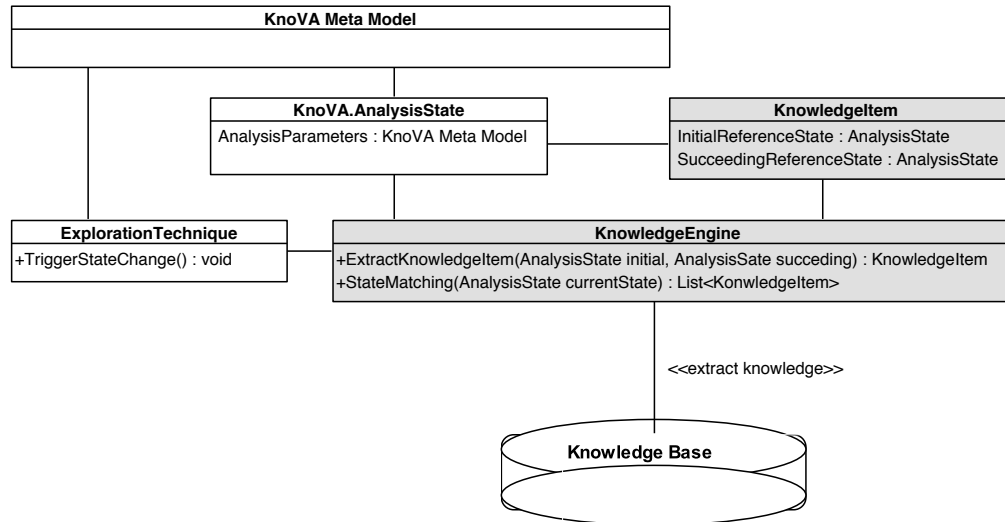


Figure 5.10: UML class diagram of the classes for Knowledge Extraction and Application.

After this the succeeding reference analysis state  $\rho_2 := \{as_2, m_{as_2}\} \in \mathcal{RS}$  is created in line three. With these two reference analysis states the knowledge item  $\vartheta_{12} := \{\rho_1, \rho_2\}$  can be created as tuple of the two reference analysis states in line four.

In order to generalise knowledge the algorithm 1 reads generalisation rules from a list of generalisation  $\Delta$  in the algorithm in line five. For each  $\delta_i \in \Delta$  the algorithm then creates a generalised knowledge item  $\vartheta_{\delta_i}(\vartheta_{12})$  in line six and saves  $\vartheta_{\delta_i}$  to a knowledge-base in line seven. With this the algorithm terminates.

The list of generalisation rules  $\Delta$  in the algorithm is representative for rules which define whether knowledge should be extracted or not. There are two basic ideas behind the generalisation rules. Firstly these rules can be used to discriminate whether it is valuable to derive a specific knowledge item. It may not be in the interest of an analyst to derive every specific knowledge item in to the knowledge-base. The evaluation which knowledge items are valuable is specific for every VA system and may even rely on the judgement of the analyst, as some of the requirements in section 3.2 have shown. Therefore the evaluation whether knowledge should be extracted or not is not specified further in the KnoVA RA.

The second idea is to use the generalisation rules to derive more general knowledge items based on the extracted  $\vartheta_{\delta_i}$ . Although some universally valid rules might be identified, the rule set is typically also specific for every VA system and will vary depending on the users requirements to the knowledge extraction.

For instance in the Mustang example, that was introduced in subsection 5.5.2, the reference analysis model states  $\rho_A$  for raw incidence and  $\rho_B$  for age distributed incidence can be assumed for the two state A and B illustrated in the figure. The user applies



**Extraction of Knowledge**

```

1: if reference analysis state change is triggered then
2:   read initial reference analysis state  $\rho_1 := \{as_1, m_{as_1}\} \in \mathcal{RS}$ 
3:   create succeeding reference analysis state  $\rho_2 := \{as_2, m_{as_2}\} \in \mathcal{RS}$ 
4:   create knowledge item  $\vartheta_{12} := \{\rho_1, \rho_2\}$ 
5:   for all generalisation rules  $\delta_i \in \Delta$  do
6:     create generalised knowledge item  $\vartheta_{\delta_i}(\vartheta_{12})$ 
7:     save  $\vartheta_{\delta_i}$  to knowledge-base
8:   end for
9: end if

```

**Algorithm 1:** Algorithm for the extraction of knowledge.

the knowledge to define the additional functional dependency as illustrated in figure 5.8. The knowledge applied so far expresses that in the specific present analysis situation the reference analysis step  $\rho_B$  follows the step  $\rho_A$ .

By mapping the concrete instances of the thematic map reference model to the KnoVA MM a universal generalisation can be made. As an example a derived knowledge item could be created, where the concrete allocations for the axis are replaced by their representation in the reference model.

**Generalisation of Knowledge**

```

1: read knowledge item  $\vartheta_{12}$ 
2: for all reference analysis states  $\rho_i \in \vartheta_{12}$  do
3:   replace analysis states  $\rho_i := \{as_i, m_{as_i}\} \xrightarrow{as_i, *} \rho_{i*} := \{*, m_{as_1}\}$ 
4:   for all model instances  $m_i \in \rho_i$  do
5:     for all meta model elements  $mme_k \in m_i$  do
6:       read meta model type  $t_k(mme_k)$ 
7:       replace  $m_i \xrightarrow{mme_k, mme_k} m_{i*}$ 
8:     end for
9:   end for
10: end for
11: create generalised knowledge item  $\vartheta_{1*2*} := \{\rho_{1*}, \rho_{2*}\}$ 

```

**Algorithm 2:** Exemplary algorithm for the generalisation of knowledge.

Algorithm 2 shows this generalisation. In line one the knowledge item  $\vartheta_{12}$  created in algorithm 1 is read. In lines two and three then for all reference analysis states  $\rho_i \in \vartheta_{12}$  then the analysis states  $as_i$  are replaced by a wild card, resulting in  $\rho_{i*} := \{*, m_{as_1}\}$ . This indicates that the state is deleted and hence the knowledge item is now generalised from the analysis state.

So far the model instances in the knowledge items are untouched. In order to create a generalised knowledge item the model instances have to be generalised as well. This is done by two nested for all loops in lines four to nine. At first in line four all model

instances  $m_i$  are read. Then in line five all meta model elements  $mme_k$  that are present in the model instance are read. Meta model elements represent all elements defined in the KnoVA MM (VISUALISATIONAXIS, AXISVALUE, FUNCTIONALDEPENDENCIES) and derived types.

After this in line six the meta model type  $mmt_k$  from the set of all meta model type  $MT$  for each of meta model elements is read. This meta model type then replaces the actual axis type in line seven to create a generalised model instance  $m_{i*}$ . For the measure axis in the Mustang example for instance this meta model type is defined by the type of the AXISVALUE. According to the KnoVA MM the type of AXISVALUE can actually be polymorphic. For reasons of simplicity in algorithm 2 an explicit type for the axis is assumed. Finally in line eleven a generalised knowledge item  $\vartheta_{1*2*} := \{\rho_{1*}, \rho_{2*}\}$  is created with the newly created generalised reference states.

It depends on the specific implementation of the KnoVA RA, the generalisation rules  $\Delta$  or even on user input, whether all elements in a meta model instance are generalised or whether just specific parts are generalised (e.g. just an abstraction from specific values is made or only specific types are replaced). Therefore the KnoVA RA does not constraint the generalisation.

### Multi Step Derivation

The concept for knowledge derivation so far assumes that the important knowledge is applied between two directly succeeding reference analysis steps. The knowledge items are built upon these succeeding reference analysis steps. It may be possible though that actually the important knowledge application does not occur between two directly succeeding steps but in a series of steps  $\Sigma_\rho := \{\rho_1, \dots, \rho_n\}$ .

As the knowledge between either two analysis steps can be derived with the introduced concepts simply knowledge items  $\vartheta_{i,i+1} := \{\rho_i, \rho_{i+1}\}$  for all pairs of succeeding reference analysis steps  $\{\rho_i, \rho_{i+1}\} \in \Sigma_\rho$ . Then all knowledge applied in the series of steps  $\Sigma_\rho$  is presented in the knowledge-base.

A second possibility to reflect this is to rely on user interaction to define a series of steps that should explicitly be extracted. The KnoVA PM contains the definition of analysis steps  $\sigma := \{as_i, \tau\}$ , where all analysis steps  $\sigma$  define the analysis process  $\Sigma$ . The class KNOVA.PROCESSMODEL reflects this by holding a reference to a list of all ANALYSISSTEP instances. Therefore all possible sequences are available in the KnoVA PM. Both possibilities to derive sequences of knowledge provide equal results. The second possibility requires user interaction to define the sequence while the first possibility potentially results in a larger number of knowledge items in the knowledge-base. The actual implementation of the derivation of sequences therefore largely depends on the specific implementation of a VA system.

### 5.6.3 Structural Architecture including Knowledge Extraction

Figure 5.11 shows how the concepts for knowledge extraction and generalisation integrates into the KnoVA RA. The knowledge extraction spans across three of the existing

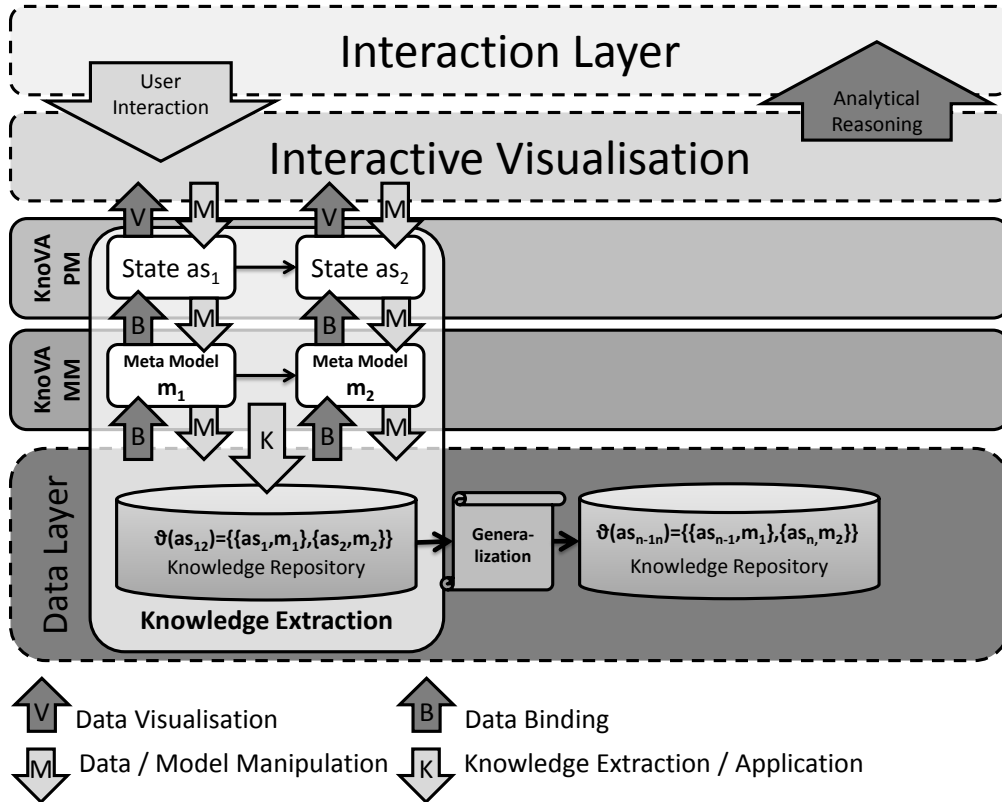


Figure 5.11: KnoVA Reference Architecture including Concepts for Knowledge Extraction.

five layers: the KnoVA PM layer, the KnoVA MM layer and the data layer. All items involved in the knowledge extraction are marked by a translucent bounding box. It can be seen that a knowledge repository is added to the data layer. This knowledge repository is independent from the data to be analysed and only contains derived knowledge.

Two analysis states  $as_1$  and  $as_2 \in AS$  with their corresponding meta model instances  $m_1(as_1)$  and  $m_2(as_2) \in M$  are shown in the figure. It is also indicated that a knowledge item  $\vartheta_{12} := \{\{as_1, m_1\}, \{as_2, m_2\}\}$  is extracted as rule with the initial state  $as_1$  the succeeding state  $as_2$  into the knowledge-base. The actual states including the currently visualised data section are included in the extraction. Thus the architecture is extended by a knowledge extraction process for domain or even analysis specific knowledge following algorithm 1. As last element the architecture is extended by a generalisation algorithm. For this another copy of the knowledge repository is illustrated in the figure. The algorithm generalises the knowledge as shown in algorithm 2. In the generalised knowledge-base it is exemplary shown how the knowledge item  $\vartheta_{12}$  is transformed in to a generalised knowledge item  $\vartheta_{n-1n} := \{\{as_{n-1}, m_1\}, \{as_n, m_2\}\}$ .

## 5.7 Knowledge Application

The final step in Keims VA process [KMS<sup>+</sup>08, KKM<sup>+</sup>10] is to re-apply extracted knowledge to the VA system to refine the analysis process. In Keims process the generation of insight and the re-application of knowledge happens implicitly. Knowledge is created within the analysts mental model and is re-applied by interaction of the analyst with the VA system.

With the introduction of an algorithmic knowledge extraction for the KnoVA RA in section 5.6 knowledge can be extracted into a knowledge-base and hence is accessible across analysis sessions and available for knowledge sharing. The fourth design implication, that was identified, is, that VA systems have to provide methods to re-apply and share extracted knowledge. By providing methods to extract knowledge into a knowledge-base and by generalising the knowledge the KnoVA RA already provides the necessary foundation for knowledge sharing. Knowledge that is present in the knowledge-base is no longer just present in the mental model of the analyst who generated the knowledge but rather available to anyone with access to the knowledge-base. In this section therefore the missing concepts for the automatic or semi-automatic knowledge application of extracted knowledge are introduced in order to account for the second part of the design implication.

### 5.7.1 Problem definition

In order to derive concepts to re-apply knowledge represented in a knowledge-base based upon the KnoVA MM, either automatically or semi-automatically, the following two sub-problems have to be solved:

1. *Identify Applicable Knowledge*: To be able to apply knowledge that is present in the knowledge-base it is necessary to identify whether knowledge is applicable in a specific analysis situation. For this a concept to extract the knowledge from the knowledge-base and to evaluate its applicability is needed.
2. *Apply Knowledge*: Once applicable knowledge is identified it has to be applied to the analysis process. For this a concept to map the generalised knowledge in a specific analysis has to be designed.

In the following subsections concepts and methods to approach these sub-problems are introduced based upon the elements of the KnoVA RA which have been introduced so far. Finally it is shown how the KnoVA RA is completed by these concepts and methods in subsection 5.7.3.

### 5.7.2 Concepts for Knowledge Application

In the process of knowledge application the first step is to identify applicable knowledge. In the UML class diagram in figure 5.10 the class KNOWLEDGEENGINE provides

a method STATEMATCHING(). To determine whether knowledge is applicable in a certain analysis situation is possible whenever a new step in the analysis process is done. Mapped to the KnoVA RA architecture this happens whenever a new reference model state is reached. Translated to the architectural elements for knowledge extraction and application introduced in subsection 5.6.2 this occurs whenever the method TRIGGER-STATECHANGE() from the class EXPLORATIONTECHNIQUE is called.

STATEMATCHING() expects a reference to an instance of ANALYSISSTATE representing the just reached current analysis state. The method will then retrieve knowledge items from the knowledge-base and evaluate whether they match to the current reference analysis state and hence can be applied in the current state. The method then returns a list of all matching knowledge items. Algorithm 3 illustrates how this matching can be done.

### Knowledge Retrieval

```

1: if reference analysis state change is triggered then
2:   read current reference analysis state  $\rho_1 := \{as_1, m_{s_1}\}$ 
3:   create generalised reference analysis state  $\rho_{1*} := \{*, m_{1*}\}$ 
4:   for all knowledge items  $\vartheta_k := \{\{*, m_k\}, \{*, m_l\}\}$  in the knowledge-base do
5:     read generalised model instance  $m_k$ 
6:     set match := true
7:     for all meta model elements  $mme_k \in m_k$  and  $mme_{1*} \in m_{1*}$  do
8:       if  $mme_k \neq mme_{1*}$  then
9:         set match := false
10:      end if
11:    end for
12:    if match = true then
13:      return  $\vartheta_k$ 
14:    end if
15:  end for
16: end if

```

**Algorithm 3:** Algorithm for the extraction of knowledge.

In the algorithm at first the current reference analysis state  $\rho_1$  is read in line two. From this analysis state a generalised reference analysis state is created by removing the analysis state  $as_1$  and generalising the meta model instance  $m_1$  to  $m_{1*}$  in line three. This is analogue to the generalisation of knowledge introduced in algorithm 2 by replacing specific derived types in the meta model with their super types from the KnoVA MM. When the meta model instance has been generalised, all knowledge items  $\vartheta_k := \{\{*, m_k\}, \{*, m_l\}\}$  are retrieved in line four. In line five from these knowledge items their generalised model instance  $m_k$  for the initial reference analysis state of the knowledge item is read in line five.

In line six the variable *match* is set to true. This is a helper variable to determine the matching knowledge items. In the for loop starting in line seven all elements of the generalised meta model instances  $mme_k$  and  $mme_{1*}$  are compared to each other. When

a non matching element is found, then the variable *match* is set to false. Finally in line 14 an if statement is used to test if *match* is still set to true. This means that all elements of  $m_k$  and  $m_{1*}$  are matching. In this way all knowledge items are retrieved where the initial reference analysis state is matching to the current reference analysis state according to their respective generalisation in the KnoVA MM.

#### Knowledge Application

- 1: read knowledge item  $\vartheta_k := \{\{*, m_k\}, \{*, m_l\}\}$  to be applied
- 2: read generalised model instance  $m_l$  from  $\vartheta_k$
- 3: read current reference analysis state  $\rho_{n-1} := \{as_{n-1}, m_{as_{n-1}}\}$
- 4: read model instance  $m_{as_{n-1}}$  from  $\rho_{n-1}$
- 5: **for all** meta model elements  $mme_l \in m_l$  and  $mme_1 \in m_1$  **do**
- 6:     create de-generalised meta model element  $mme_l(mme_{n-1})$
- 7: **end for**
- 8: create de-generalised meta model instance  $m_l(m_{n-1})$
- 9: create succeeding reference analysis state  $\vartheta_n := \{as_n, m_l(m_{n-1})\}$

**Algorithm 4:** Algorithm for the extraction of knowledge.

After the matching knowledge items have been retrieved this way, it is possible to apply the knowledge to the system. A knowledge item consists out of two reference analysis states, the initial state and the succeeding state. It expresses that when the initial state is reached the succeeding state is one possible next state. Therefore, if the initial state of a knowledge item matches to the current reference analysis state of the VA system, it can be judged, that the succeeding state of the knowledge item is a possible next state for the VA system in the current analysis.

Depending on the requirements these conclusions can be made either automatically, if there is no conflicting knowledge in the knowledge-base, or with user interaction by visualising the applicable knowledge and then request the analyst to make the most reasonable choice.

The requirements for this vary depending on the VA system or even current situations (compare chapter 3 for requirements concerning knowledge application). Therefore no general solution for this problem can be given. Hence the KnoVA RA is not constraint to concepts for automatic or semi-automatic knowledge application.

However, once if a knowledge item is chosen for application (either automatic or semi-automatic) it needs to be transformed in order to be integrated into the analysis process. Algorithm 4 shows how a knowledge item can be transformed into the context of the current analysis process and then be applied to create the next reference analysis model step in the current process.

In algorithm 4 for this at first the knowledge item  $\vartheta_k := \{\{*, m_k\}, \{*, m_l\}\}$  that was chosen to be applied is read in line one. The initial generalised reference analysis state  $\rho_k := \{*, m_k\}$  from this knowledge item matches with the model instance  $m_{as_{n-1}}$  from the current reference analysis state  $\rho_{n-1}$ . Therefore the knowledge is applicable, as determined by algorithm 3.

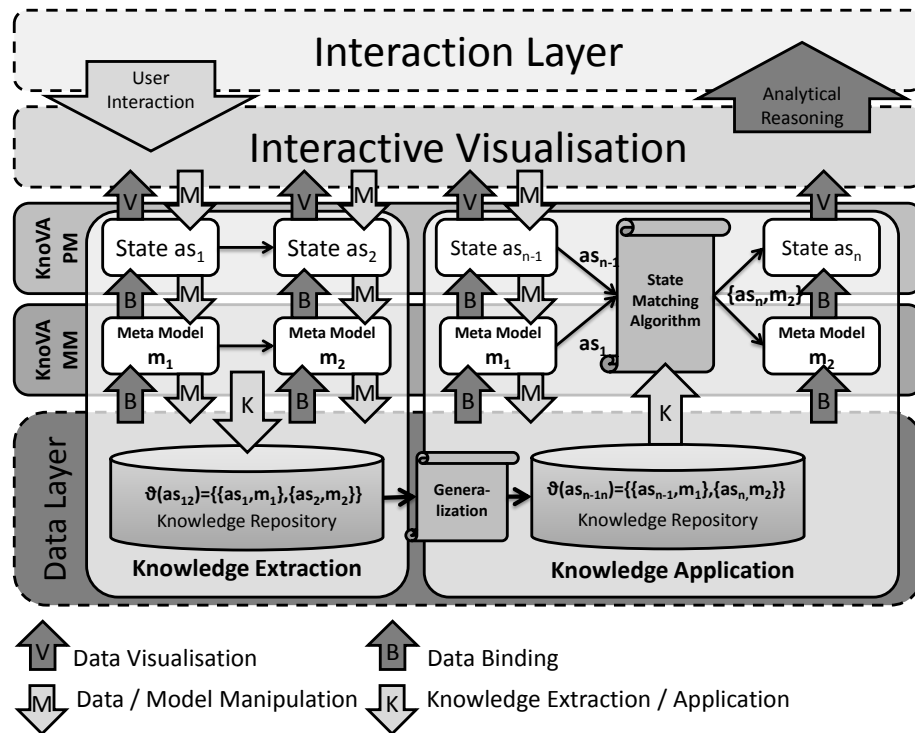


Figure 5.12: The KnoVA Reference Architecture.

The succeeding generalised reference analysis state  $\rho_l := \{*, m_l\}$  is to be applied in the current analysis situation. For this the generalised model instance  $m_l$  is read from  $\rho_l$  in line two. In line three the current reference analysis state  $\rho_{n-1}$  is read to determine the current model instance  $m_{as_{n-1}}$  in line four. This model instance contains all information concerning the data that is currently visualised. To apply the knowledge item, all this information has to be mapped to the succeeding generalised model instance  $m_l$  from  $\vartheta_k$  that was read before.

For this all meta model elements from  $mme_l$  from  $m_l$  and  $mme_{n-1}$  from  $m_{n-1}$  are iterated in the for all loop in line five and a de-generalised model element  $mme_l(mme_{n-1})$  is created in line six. Based upon these de-generalised model elements a de-generalised meta model instance  $m_l(m_{n-1})$  is created in line eight and with this the new succeeding reference analysis state  $\rho_n := \{as_n, m_l(m_{n-1})\}$  is created in line nine.

The whole process of de-generalisation can be seen as inversion of the generalisation with the speciality that the model instance of the current reference analysis step is used to create a new model instance for a generalised knowledge item. After this the new succeeding state  $\vartheta_n$  can be introduced into the analysis process and the process of knowledge application is complete.

### 5.7.3 Structural Architecture including Knowledge Application

In figure 5.12 the KnoVA RA is illustrated with all four elements. The KnoVA PM introduced in 5.4, the KnoVA MM introduced in section 5.5, the concepts for knowledge extraction introduced in 5.6 and the concepts for knowledge application introduced in this section.

The concepts for knowledge application are illustrated as translucent bounding box that spans across the KnoVA PM layer, the KnoVA MM layer and the data layer, comparable to the concepts for knowledge extraction introduced in 5.11.

The figure shows a current analysis state  $as_{n-1}$  with its corresponding meta model instance  $m_{as_{n-1}}$ . Next to this state the state matching algorithm is illustrated. It receives the current analysis state and the meta model instance as input. Below this the knowledge-base with the generalised knowledge items is illustrated. These are also an input to the matching algorithm, reflecting the input of knowledge items as described in algorithm 3. The algorithm produces as output a succeeding state  $as_n$  and the corresponding meta model instance  $m_n := m_l(m_{n-1})$ .

With the introduction of the concepts and methods for knowledge application the KnoVA RA is completed. In the following section the KnoVA RA is validated whether it can be used to answer the research question and to face the research challenges identified in 1.1 and it is compared to the related work examined in 2.

## 5.8 Validation of the KnoVA Reference Architecture

In section 1.1 four research challenges were identified in in the two real world scenarios. These challenges then lead to the research question:

*With which concepts and methods can expert knowledge, that was applied during the process of VA, be represented and extracted, to make it reusable?*

In this section the four research challenges identified in 1.1 are revisited and it is argued, how the KnoVA RA approach responds to the research challenges and thus is suitable to answer the research question.

**Challenge 1:** How can collaborative analysis be supported by VA systems? For example how can experts work at the same analysis questions and how can a system support knowledge sharing in the collaboration?

⇒ There are multiple elements in the KnoVA RA to face this challenge. The KnoVA PM allows VA systems to support the analysis process by providing a formal description of this process. This results in a tangible and structured support of the analysis process, in which the analyst is aware and in control of multiple reoccurring and varying analysis steps. This is the first element to provide support for knowledge sharing.

Above this the KnoVA PM provides means to represent all steps in the analysis process. This can be used as the foundation for features such as history functions which provide an overview over the course of the analysis and hence can be used to communicate this



course of the analysis in collaborative scenarios. By providing a tangible and structured model of the analysis process this process is explicitly recognisable for people working in collaborative settings. Without a tangible and structured model of the analysis process the course of the analysis would only implicitly be known only by the person who defined it.

Knowledge sharing is made possible with the KnoVA MM. This model allows to extract applied knowledge into a knowledge-base, where it is accessible for multiple experts. Hence multiple experts can apply their shared knowledge to the same analysis questions. Also the KnoVA MM in conjunction with the KnoVA PM allows knowledge sharing by sharing of reference analysis model states. In the same manner as knowledge can be extracted and saved to a knowledge-base it can also be extracted and re-loaded into VA systems of other experts for a direct knowledge sharing.

**Challenge 2:** How can implicit expert knowledge be extracted, in order to transfer results to other analysis tasks? For instance, how can findings be transferred from one trace-file to another to simplify the analysis of the other trace-file?

⇒ All four elements of the KnoVA RA combined allow to respond to this challenge. The KnoVA PM and the KnoVA MM allow to represent the applied knowledge. Additionally in algorithm 1 it was shown how applied knowledge can be extracted. In algorithm 2 it was shown how this knowledge then can be generalised in order to be used in a different context (for instance for different trace-files). In the algorithms 3 and 4 it was shown how knowledge can be matched to and applied in other analysis situations, allowing to automatically or semi-automatically create results based upon earlier findings. This allows to simplify the analysis of other trace-files.

**Challenge 3:** How can a VA system support the representation of expert knowledge that is implicitly applied by the expert in the analysis process?

⇒ The KnoVA PM allows to express the course of the analysis. For this in this subsection 5.4.2 a formal definition of the analysis process was introduced. Based upon this formal definition then the architectural elements necessary to represent the KnoVA PM were illustrated in figure 5.3. Finally it was shown how the structural architecture of VA systems is enhanced by the KnoVA PM in subsection 5.4.3.

The KnoVA MM, which was introduced in subsection 5.5.2 serves as a meta data model to represent the actual state of the analysis system in the KnoVA PM. With definition of reference analysis states in definition 6 the KnoVA MM and the KnoVA PM can be used to represent the knowledge that was applied in the analysis process and in this way supports the representation of expert knowledge that was applied in the analysis process.

**Challenge 4:** How can this knowledge be extracted into a knowledge-base, in order to be re-applied the analysis process? For instance, how can this knowledge be used to support automatic aggregation steps?

⇒ This challenge is comparable to challenge 2. All four elements of the KnoVA RA combined allow to respond to this challenge. Automatic aggregation steps can be sup-

Process/Model	Detail	Abstraction	Formalism	Knowledge
Information Visualisation Cycle	Low	Exploratory / Procedural	□	□
Visual Analytics Process	Fine-Grained	Exploratory / Procedural	■	(□)
Data-Flow Model	–	Visualisation Transformation	■	□
Data-State Model	–	Visualisation Transformation	(■)	□
P-Set Model	–	Hybrid	■	(□)
KnoVA Reference Architecture	Fine-Grained	Hybrid	■	■

– criterion not applicable, ■ criterion is supported, (■) criterion is supported with limitations, (□) rudimentary support, □ criterion is not supported,

Table 5.2: Qualitative State-of-the-Art Comparison of the KnoVA RA.

ported by extraction of applied knowledge from previously processed manual aggregation steps. These manual steps will result in a change of the reference analysis states in the analysis process and therefore the concepts and methods for knowledge application and sharing provided in the KnoVA RA can be used to extract this knowledge. It can then automatically or semi-automatically be applied in the process by the introduced algorithms for knowledge application.

In the following subsection the KnoVA RA is compared to the related work introduced in chapter 2 in it is argued how the KnoVA RA enhances the state of the art.

### 5.8.1 Comparative Evaluation

In sections 2.1 to 2.2 existing approaches to model the analysis process and existing approaches for visualisation models were introduced. These were examined in a comparative evaluation in section 2.4 resulting in table 2.1 that provides a summarising overview over common features of the existing approaches.

The KnoVA RA includes a model of the analysis process. Therefore it can be compared to existing visualisation models on this level. In the following subsections the features of the comparison are examined and it is discussed how the KnoVA RA compares to the existing approaches.

#### Level of Detail

This criterion describes the level of detail which is provided by the techniques. The KnoVA RA resembles all steps in Keims VA Process and refines significant aspects. For instance the KnoVA PM defines the analysis process not only formal but also provides

concepts for architectural elements. Also the KnoVA MM provides a fine-grained definition meta data model for the states of the analysis process, which is missing in the VA process. Therefore in summary it can be argued that the KnoVA RA provides a fine-grained level of detail.

### Level of Abstraction

This criterion describes on which level of abstraction the respective technique can be ranked. Two different categories are distinguishable. Exploratory or procedural models mainly aim to provide a high level description of the process of visualisation exploration which is not suitable for or at least not intended to serve as foundation of a software implementation of the process. These techniques are rather aimed at communication and understanding of the domain and to provide a description of the process of visualisation exploration. The other categories is visualisation transformation. Models with this level of abstraction provide a description of the task of creating a visual mapping from input data. All three visualisation models fall into this category.

Although the KnoVA RA is clearly focussed to serve as foundation for a software implementation it also incorporates an exploratory or procedural model. Therefore, like the P-Set Model the KnoVA RA is a hybrid model, which can serve for both purposes.

### Formalism

This criterion describes whether a formal description of the technique is available. The KnoVA PM is formally defined. The KnoVA MM is defined based upon a UML representation which can also be interpreted as formal description. Therefore, unlike the other systems the KnoVA RA provides a thorough formal description of the process and of the meta data model.

### Knowledge Extraction and Sharing

This criterion describes whether the techniques provide concepts and methods for knowledge extraction. From the existing systems only the P-Set Model and Keims VA process examine this factor. They do not provide concepts for this though. In particular they do not provide a data model or even an algorithmic support for knowledge extraction and sharing.

As a unique feature the KnoVA RA provides a rich support for knowledge-based features, a detailed meta data model as well as algorithms for knowledge extraction, generalisation and application and therefore significantly extends the state of the art in this field.

## 5.9 Summary

In this chapter the KnoVA reference architecture for VA systems was designed based upon the challenges defined in section 1.2. The KnoVA RA provides concepts and methods to represent and extract expert knowledge, that was applied during the process of VA in order to make it reusable.

In section 5.3 general requirements for a reference architecture were identified by revisiting the design implications that were introduced earlier. After this a structural architecture for VA systems was introduced in section 5.2.

In the following four sections this structural architecture was successively extended to create the KnoVA RA. Each of these four sections addresses one of the identified design implications and the corresponding general requirements. After the problems to be solved are identified in each section novel concepts to address the problems are introduced to the structural architecture.

In section 5.4 a formal model of the analysis process, the KnoVA PM was introduced. For this at first the analysis process is examined in order to identify the elements of the process. After this a formal model to express these elements is introduced. The formal model then serves as the foundation for architectural elements representing the analysis process.

The definition of the analysis process is followed by section 5.5 in which a meta data model for VA systems is introduced, the KnoVA MM, which is based upon the KnoVA taxonomy. Combined with the KnoVA PM the KnoVA MM allows to represent the actual state of the analysis system and by representing state changes they also allow to represent the applied expert knowledge.

Based upon this in section 5.6 concepts and the algorithmic foundation for the extraction of knowledge into a knowledge-base and the generalisation of knowledge are introduced. These concepts and algorithms then lead to the next section 5.7 in which concepts and algorithms for the application of extracted knowledge are introduced. With this the definition KnoVA RA is completed.

In section 5.8 the KnoVA RA is examined in a comparative validation. For this the challenges identified in section 1.2 are revisited and it is argued how the KnoVA RA can be used to face these challenges and it is shown how the KnoVA RA compares to the state of the art.

## 6 Implementation

This chapter shows how the KnoVA RA is used to implement systems for knowledge-based VA. The objective of this chapter is to showcase the applicability of the KnoVA RA in the real world scenarios.

In section 6.1 the system TOAD for VA of ICNs is described. Here the RA was used to implement knowledge sharing between experts within and across analysis sessions.

In section 6.2 the RA is used to implement the tool CARELIS for visual aggregation of medical records, according to the second motivating scenario. In this implementation experts can create rules to extract the knowledge they applied during the analysis process.

Each of those two sections first introduces the implemented VA system and then describes how the KnoVA RA serves as its foundation. After this it is discussed how the VA systems are facing the challenges identified in the introduction. Section 6.3 closes this chapter with a summary.

### 6.1 TOAD: A Tool for the VA of ICNs

TOAD is used to visualise pre-recorded traces of messages of ICNs. The first step in the analysis process is an automated step. Here finite state machines (FSM) are extracted from the bus message traces. The states of these and the permitted transitions between these states are pre-defined by the analysts. The FSMs are extracted based upon an algorithm that is described in [SIB<sup>+</sup>11]. The same abstraction is also used in Cardiogram as described in 4.2.7, and well known to the engineers. Briefly summarised the algorithm analyses the message trace and derives the timely behaviour of the FSM based upon a specific message trace.

Like in Cardiogram the state machines describe the state of the ICU on an abstract level. The abstracted trace file contains the behaviour of multiple FSMs, where a typical FSM will contain between 2 – 15 states and can contain hundreds of state changes [SIB<sup>+</sup>11]. For the analysis the trace is replaced by a file with a description of a timely behaviour of the FSM. States can be of the types *error*, *warning*, *okay* or *info* and *init*. The *init* state is used to encode the initialisation that occurs in every trace before a defined state is reached for the first time.

State changes that are triggered by the bus messages can lead from one state type to another (*okay* → *okay*, *okay* → *error*, *error* → *warning* etc.). The only exception is the *init* state, from which only can occur once as the first state type in for every FSM. The file with the FSM also contains the raw bus messages, annotated to the states based upon their timely order. Therefore it is possible to find the messages that were transported over the ICN in correlation of a state change based upon the time stamp.

In figure 6.1 a screenshot of the TOAD system can be seen. These elements are labelled from a – f in ascending order: (a) state machine overview selection view, (b) message-sequence chart view, (c) state-transition list view, (d) raw data view, (e) visual history and (f) context menu. These elements and the additional functionality will be described

below.

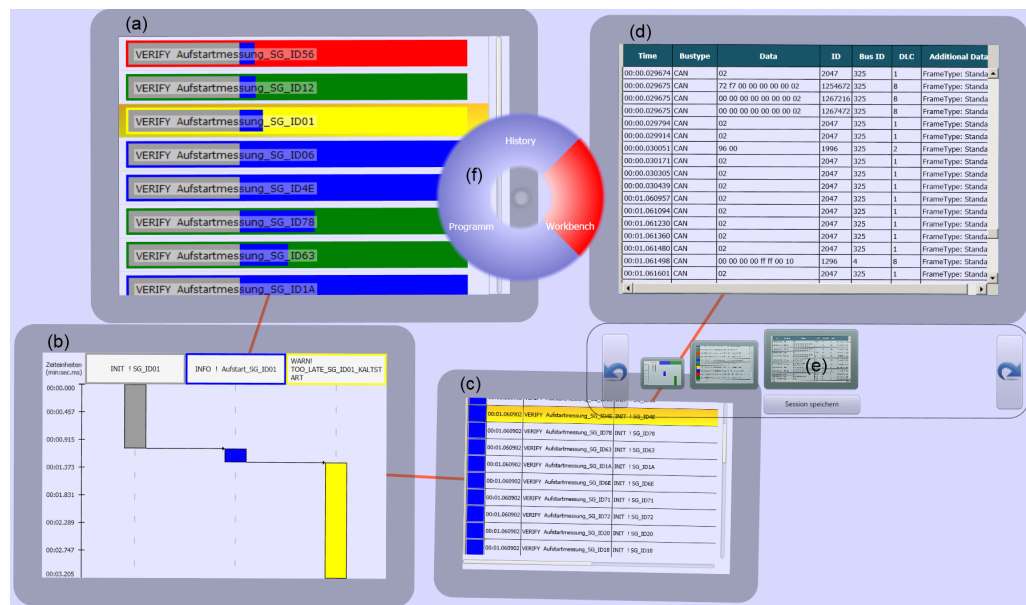


Figure 6.1: Screenshot of the TOAD System for the VA of ICNs.

In order to start the analysis, the user starts with an empty window, which is called workbench in the TOAD system (compare the interaction of the left user in figure 6.2). The empty windows shows two folder icons. By touching one of these the user can either open a new trace file to start a new analysis session or open saved analysis sessions. The interaction with the analysis system relies on manual gestures on an interactive surface computer.

**Surface Computer:** Like the TAP system, that was described in subsection 4.2.1, the TOAD was implemented on a multitouch surface computer. This design decision was made upon requirement [R4] where a transition between individual and group work was requested. In previous research in the field of Computer Supported Cooperative Work (CSCW) it was shown that horizontal devices are typically better suited for collaborative work, as they provide better opportunities for face to face communication [RL04, MTWS08].

The engineers described that in situations of collaborate work in their current practise they gather around the desktop computer or laptop of a colleague. In such a setting however only one person at a time can interact with the analysis system. In [R4] the engineers requested to switch seamlessly between individual and group work. Such a working scenario is not easily possible on a desktop computer. Previous research in the field of CSCW shows that these limitations can be overcome by additional hardware such as interactive surface computers [BBB<sup>+</sup>08].

Another requirement that supported the decision to use a surface computer as the foun-

dition of the TOAD system was [R10] where the engineers requested on-view selection and interaction. Surface computers integrate the input device (the touch sensitive surface) and the output device (the large display) into one interactive surface. This allows a direct interaction with with the visual user interface without the usage of other input devices. The interactive surface also allows the concurrent interaction of multiple analyst by its design, as shown in figure 6.2 where two users are collaborating in the same VA session.

The interactive surface computer used for the TOAD system was the same one that was used in the TaP example (compare subsection 4.2.1). The interactive display has a size of 1300x900mm and a resolution of 1280x800 pixels. It is based upon an optical tracking and can accept a large number of parallel touches, limited by computing power and by the area of the surface. It was developed at the OFFIS Institute for IT.



Figure 6.2: Two Users in a VA session at the Multitouch Surface Computer.

**Views:** According to the requirements the VA systems needs to integrate heterogeneous visualisation tools [R1]. The TOAD system therefore features four different kinds of views. These can be seen in figure 6.1. The views in the figure are (a) the state machine overview selection view (SOL view), (b) the message-sequence chart view (MSC view), (c) the state-transition list view (STL view), and (d) the raw data view (raw view).

The SOL view, which can be seen in figure 6.1 (a), is the first view that is shown when a new trace file is opened. The view shows a list of all FSMs that are present in a trace file. Therefore it provides an overview visualisation of aggregated bus traces [R5]. In the view every state machine is represented by a vertical bar. This vertical bar shows the timely behaviour of the FSM it is representing, scaled to the length of the vertical bar. Starting from left to right it is shown in which state the FSM resided at a given point in time. The colour coding is as follows: grey for init states, blue for info states, green for okay states, yellow for warning states and red for error states. The colour coding was chosen based upon the colour coding that the analyst use in existing tools

such as MS Excel. In addition to the timely behaviour a text label shows the name of the FSM. Around each vertical bar is a border in the colour of the most important state that was reached in this FSM. The importance is defined based upon the FSM type. The importance in descending order is error > warning > okay > info > init. The user can select a certain FSM to be analysed in more detail by touching the respective state. After this a new work bench opens with a more detailed view of the selected FSM in the MSC view [R8].

The MSC view, like the SOL view, visualises the timely behaviour of the FSM. Other than the SOL view though the MSC view also displays the transitions between states. Thus it visualises a more detailed view of the aggregated bus traces [R6] than the SOL view. An exemplary MSC view can be seen in figure 6.1 (b). The visualisation of the MSC view uses the notation of UML sequence diagrams to represent the FSMs. The life line of the sequence diagram represents the whole period that the recorded trace encompasses. This can be between minutes and several hours. There is one life line for each state that was reached within the trace file. When a message on the bus triggers a state change, a message on the sequence chart is visualised according to the UML notation. The activation boxes on the life lines represent the time the FSM resides in a certain state. In figure 6.1 (b) an example with three states can be seen: the init state, an info state and a warning state. Each state has a life line. On the top of the work bench labels show the name of the state. The FSM resided for about 950ms in the init state, switches to the info state for a short period of time to then switch to a warning state where it resides until the trace is closed. The arrows that indicate the transition in this view can be selected by the user. After this a new work bench opens with a view showing detailed information about this transition in the STL view [R8].

The STL view is shown in figure 6.1 (c). Here all transitions are shown in a table with four columns. The transitions are listed here in timely order, newer transitions at the bottom. The four columns show the type of the state to which the transition is leading, the exact time stamp of the transition, the name of FSM and the name of the state that the transition points to. In the figure the transition between the init state and the info state in figure 6.1 (b) is highlighted. The STL view is synchronised with the MSC view. It will always show the part of the transition list with the transition that was selected by interaction with the MSC view. It provides a detailed visualisation of the aggregated bus traces [R6]. When a transition in the STL view is selected the raw view will be opened [R8].

The raw view displays all fields present in the raw bus messages according to [R7] in a timely order. This can be seen in figure 6.1 (d). When opened by interaction with the STL view the raw view will be synchronised with the STL view and display the message that matches with the transition that was selected in the STL view. The number of fields visualised in this view varies depending on the message type and bus type. The elements in the raw view can be sorted based upon different orders (e.g. timely order based upon the time stamp) [R9].

**Analysis Paths and History:** Each view is opened in a separate workbench. This allows the analyst to quickly switch between abstraction levels [R2] from the MSC view with



the highest level of abstraction to the raw view with the highest level of detail. Indeed several views with different abstraction levels and even for different analysis traces can be opened at the same time and the user can freely switch between these levels [R8].

To evaluate the user interface and the interaction concept of the views and workbenches several smaller user studies with up to 10 participants were performed. These studies were performed with students and volunteers in order to save domain experts time, following the recommendations of [SIBB10]. The participants in these studies were invited to work collaboratively on typical analysis problems in groups of two users.

During these studies, it became clear that an issue in the design was that users tended to open many workbenches at the same time, most of the time overlapping each other. This resulted in the problem that in collaborative analysis sessions, users lost track of who opened which workbench, and forgot which analysis steps they had taken.

To overcome this problem the concept of visual analysis paths was introduced. The analysis paths are visualised as a red line connecting two succeeding views. Each time a novel view is opened from another view a line i.e., an analysis path, is added between these views. Analysis can therefore lead to linear or branched connections between views. In collaborative scenarios involving multiple analysts and multiple opened views this concept clearly facilitated analysts in keeping track of the linkage between views. This ensures the traceability of the analysis [R11, R13].

While explaining the analysis process, experts repeatedly asked for the ability to retain knowledge about the course of the analysis process. To support this requirement, visual history function was created (see figure 6.1 (e) [R13]). The visual history shows the steps taken in the analysis.

The history can be opened through the context menu for every work bench. A work bench with attached visual history can be seen in figure 6.1 (f). It shows the history in the current analysis path, starting from the root node. The current node is displayed in the middle of the visual history. Previous steps are shown left to the current step, succeeding steps are shown to the right. The steps are scaled down based upon their distance to the current step. This ensures the scalability in real-world scenarios with many recorded steps. A touch on a certain element of the history will result in a switch to the recorded system state. In addition to this, the visual history allows users to navigate in the history, i.e. , undo and redo recorded steps.

**Session Persistence:** As identified in the requirements analysis [R12, R14], the system needs to support interruptions of analysis sessions and needs to support to externalise findings. Thus, a method to save a complete analysis session was integrated into the system, including all analysis steps taken, as well as the configuration of the workspace. Using this mechanism, users can continue the session at any time and at any compatible analysis platform throughout the company (including possible remote sites).

To re-apply the derived knowledge, the user can choose filter application from the marking menu in the finite state machine view. This will result the view to change into a selection list where all derived rules are displayed. In this list the analyst can perform a multiple selection of rules. After he applies the rules, the finite state machine view will display only those state machines, which fit to any of the applied filters.

**Workbench Synchronisation:** Another feature of the TOAD system that was inspired by ideas gained in user studies is the support for workbench synchronisation. As mentioned above each view in the TOAD system is visualised in a separate workbench. When prototypes of the software were presented to the experts in the demonstrations during the agile development process the engineers expressed their demand to share the current view with other colleagues in group work scenarios [R4] with a direct mapping between the views [R3].

These requirements lead to the development of the workbench synchronisation feature. With this feature the current state of the view in a certain work bench and hence the knowledge that was applied by one expert to this point can be shared with other analyst, allowing multiple experts to work on the same data concurrently without the need to work on the same work bench [R12].

When the state of a work bench is transferred to another workbench this new work bench is a exact copy of the first one, visualising the same aspect of the data. From there the analyst can work either synchronously or asynchronously. In the synchronous setting all changes that are made are visible in both synchronised views. In the asynchronous setting the experts can work individually from the point they synchronised the views.

**Smart Filtering:** Another requirement that was demanded was a method to mark certain points in the visualised data and a functionality to express the interesting support knowledge transfer to other analysts [R14] and across analysis sessions. In particular the analysts asked for methods to extract analysis results and findings to present them to other analysts and to use them in future analysis scenarios [R12]. These requirements were addressed by a smart filtering function. With this function filters can be derived by the user in the MSC view. These filters can then be applied in future analysis scenarios to find matching FSMs in the SOL view. In figure 6.3 the process of filter derivation and filter application is shown.

In the figure four screenshots of work benches from the TOAD system can be seen. The work bench in figure 6.3 (a) shows how knowledge is extracted. This view is opened when the user selects to extract knowledge by the context menu. In this view knowledge that can be derived in the current analysis step is shown in a list.

To extract knowledge the user simply has to select an item from the list. It will then be stored in the knowledge-base as a smart filter. The definition of a filter is based on pre defined rules. Extracted can be a sequence of states by ID or a sequence of states abstracted from their IDs of a specific FSM (e.g.  $\text{init} \rightarrow \text{info} \rightarrow \text{okay}$ ), a FSM name, a single state name or a set of state names a specific transition between two states or a sequence of transitions. This pre defined rule set for knowledge extraction is not comprehensive.

In the prototypic implementation of TOAD these four cases have been identified by the engineers as most important. Once a specific knowledge item is extracted, it can then be applied in the SOL view, as shown in figure 6.3 (b). When the user opens the knowledge-base from the SOL view, the existing knowledge items are shown in a list, this can be seen in figure 6.3 (c). From this list the user can choose a specific knowledge item.

The SOL view will then be filtered to only represent the FSMs that apply to this filter,

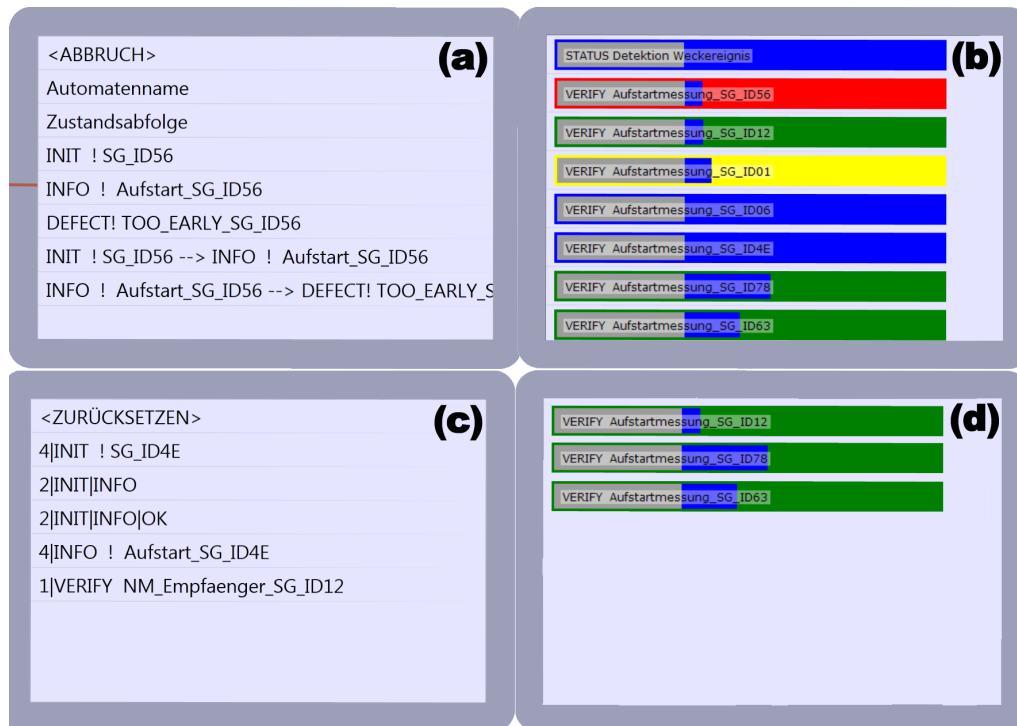


Figure 6.3: Smart Filtering in the TOAD System.

as shown in figure 6.3 (d). If for instance the filter labelled 2—*Init—Info—OK* is chosen the SOL view shown in figure 6.3 (b) will change to the view shown in figure 6.3 (d). The number in front of the filters is an ID representing the analysis session in which the filter was derived.

The filtering reduces the number of visualised FSMs and thus allows to more quickly determine points of interest across and within trace files. Filter creation and application allows to transfer findings to other analysts who can simply select the filter and apply these findings to their own problem.

### 6.1.1 Realisation based upon the KnoVA RA

TOAD is the result of a one year student project, which was accomplished in co-operation with a team of test engineers of an industry leading car manufacturer. In the implementation of the tool TOAD the KnoVA RA was used as a foundation. In this subsection the features that are implemented based upon the KnoVA RA are reviewed.

Firstly a brief overview over the integration of the KnoVA RA into the TOAD system is shown in figure 6.4. On the topmost layer in the figure illustrated are the four views of the TOAD system. Below the interactive visualisation layer another layer for the KnoVA PM is shown. Here The classes of the process model according to figure 5.3

are present. Derived from the class `KNOVA.ANALYSISSTATE` the class `WORKBENCH` is shown. The four views implement this generic workbench class. Each view in the TOAD system resides in a separate workbench. Therefore the workbench can be seen as a container for the views.

In the TOAD system each state changing operation with the VA system will result in the creation of a new workbench. An association between the workbench and the TOAD visualisation indicates this relationship.

Below the KnoVA PM layer the layer for the KnoVA MM is illustrated. The actual state of the analysis system in TOAD is represented by the visualisation, as intended by the KnoVA RA. To realise this aspect therefore the views in TOAD hold a reference to a meta model instance. For this four classes are introduced `SEQUENCECHARTVIEWMODEL`, `RAWDATAListViewMODEL`, `STATEMACHINESELECTIONVIEWMODEL` and `TRANSITIONLISTVIEWMODEL`.

TOAD is implemented based upon the Windows Presentation Foundation (WPF) technology [Mac10], which is part of the the Microsoft .NET Framework. Applications implemented in this technology use a specialised architectural pattern, the Model – View – ViewModel pattern [Mac10].

**Term** (Model – View – ViewModel). *The **Model - View - ViewModel (MVVM)** pattern is an architectural pattern for applications featuring interactive visual user interfaces. The pattern defines three architectural layers. The view layer, which contains all visual elements of the application. The model layer, which contains all entities and underlying data structures. Between those two layers the viewmodel layer contains any business logic to retrieve the data to be visualised from the model. Hence the pattern creates an abstraction between the model and the view. Thus MVVM ensures a separation of concerns.*

*In this pattern the viewmodel contains any business logic to retrieve and enrich data from the model. The viewmodel is consumed by the view and includes all necessary attributes for the view to be rendered. The view itself does not contain any logic. All interaction with the system is interpreted by the business logic in the viewmodel. Typically the viewmodel is unknown to the view at compile time and the dependency between view and viewmodel is only evaluated at run time.*

*There are several ways to implement the viewmodel. Either a passive viewmodel that can be consumed by the view is created by specialised creational classes (e.g. the abstract factory [Fow02]) and does not contain business logic or an active viewmodel, which is actively providing the fields to be consumed by the view.*

Due to its structure the KnoVA MM fits into this architectural pattern as a base class for the viewmodel. Therefore the viewmodels for all of the four views in TOAD are derived from the class `KNOVA META MODEL` as introduced in figure 5.6. With the integration of the KnoVA PM and the KnoVA MM the defining architectural parts of the KnoVA RA are integrated into the TOAD system. The features below are realised based upon this architecture.

The viewmodel in this implementation is a passive viewmodel, which contains all busi-

ness logic that is necessary to translate between the work benches and the actual XML trace files underneath. The viewmodel in this implementation therefore also contains the business logic to trace the changes in the system states according to the KnoVA PM.

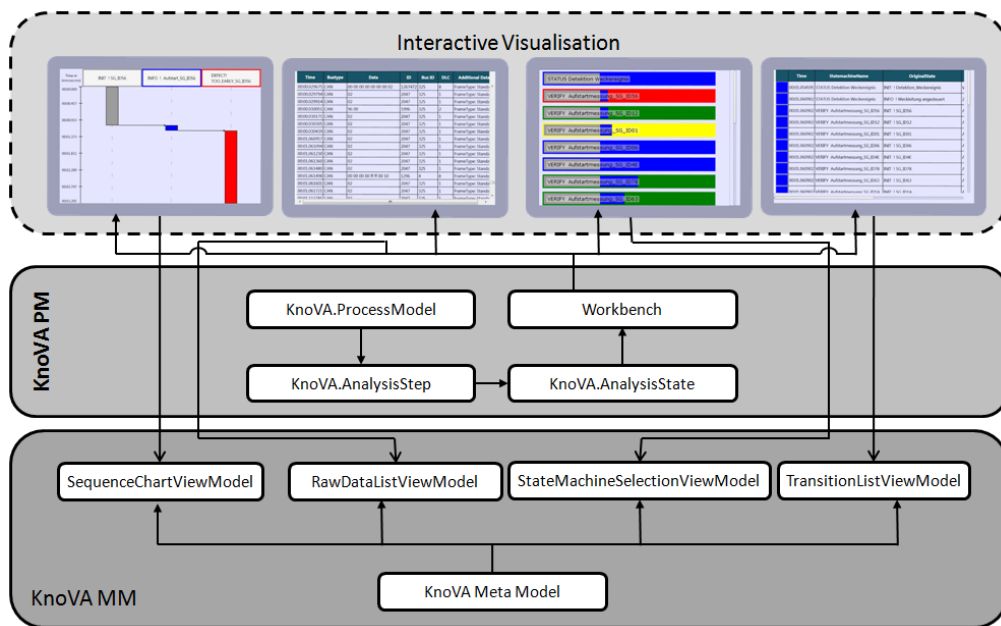


Figure 6.4: Integration of the KnoVA RA into the TOAD System.

**Analysis Paths and History:** The history function of TOAD is realised by a combination of the KnoVA PM and the KnoVA MM. The class `KNOVA.PROCESSMODEL` holds a reference to a list of `ANALYSISSTEP`. Each view in the TOAD system has a reference to its viewmodel, which is derived from the KnoVA MM. Every workbench, which is derived from `ANALYSISSTATE` holds a reference to a view. Every step in the analysis therefore is already represented in this class. Therefore the course of the analysis can be drawn from this list. To create a history function only the visualisation for the history (see figure 6.1 (e)) had to be integrated into the TOAD system.

The visualisation paths in TOAD likewise can be implemented based upon the KnoVA PM and the KnoVA MM, as the visualisation paths simply represent the analysis path which is modelled in the KnoVA PM.

**Session Persistence:** The requirement for session persistence in the TOAD system is realised based upon the knowledge items, defined in the KnoVA RA. In order to save a complete session it is sufficient to save all knowledge items, as these represent the state changing operations.

Therefore a method for knowledge extraction based upon the concepts introduced in 5.6. When all knowledge items of an analysis path are extracted, then the complete analysis path can be reconstructed. In order to be able to identify the course of a specific analysis

session therefore in TOAD the knowledge-base was implemented based on an XML file format, with one XML file containing all knowledge items of a specific session.

When a certain trace file is loaded and the respective XML file from the knowledge-base is selected a complete analysis session can be re-loaded. Above this, based upon algorithm 2 for the generalisation of knowledge, the knowledge items in the XML can be abstracted from the actual trace file, thus allowing to open the course of a previous analysis with another trace file. During the evaluation it became clear though, that this feature is only valuable in a limited number of situations. If the extracted session is not generalised, then it can only be applied to traces which include exactly the same data. As the traces contain time-stamps, this will rarely be the case. However if the knowledge is generalised to much, than the extracted sessions are very arbitrary. The value of the session persistence therefore lies mainly in sharing the session and the specific trace file.

**Workbench Synchronisation:** The workbench synchronisation in TOAD was realised based upon the KnoVA MM. As all viewmodels in the TOAD system derive from KNOVA META MODEL it is possible to determine at run time, which parts of the currently presented views represent the same aspect of the underlying data. For this a specialised matching algorithm was implemented, which gets two viewmodels as input, casts this to the representation in the KnoVA MM and then compares or maps the elements of one viewmodel to the other. This allows to create the workbench synchronisation features described above.

**Smart Filtering:** The smart filtering function was realised based upon the KnoVA RA. The extraction of knowledge is available from the MSC view. As described above each view in the TOAD system is represented in the KnoVA PM by a reference model analysis state based upon the KnoVA MM. When the user chooses to extract knowledge by the context menu, the currently visualised reference model analysis state is extracted, following the algorithm 1 for knowledge extraction.

As described in section 6.1 the TOAD system defines a set of pre defined rules for the extraction and generalisation of applied knowledge. When the user chooses to extract knowledge, all knowledge items that can possibly be derived are visualised in the current workbench (see figure 6.3 (a)). The generalisation of the knowledge items according to algorithm 2 was respectively completed before this process.

The application of knowledge is triggered manually by the user. All available extracted knowledge items are visualised in a list (compare figure 6.3 (c)). Therefore a matching of which knowledge items are applicable in the current state according to algorithm 3 is not performed. In this list the user chooses the knowledge item to be applied.

The knowledge item chosen by the user is then used to change the reference analysis model state of the current SOL view. For this the knowledge item is de-generalised to be matched to the current reference analysis model state, following algorithm 4. After the knowledge application only those FSMs which match to the criteria described in the knowledge item are shown. When additional knowledge items are chosen, other FSMs from the current trace are added. Thus the more knowledge items are selected as filter, the more messages apply. This conjunctive implementation of the filters was chosen based upon the requirements of the engineers. It is also the reason why a matching

of which knowledge is applicable is not necessary because all knowledge items can be applied. If a knowledge item does not match with any of the FSMs an empty list will be visualised, indicating the user that he has to broaden the selection.

### 6.1.2 Conclusion

As shown above the TOAD system addresses the requirements [R1 – R14] that were identified in section 3.1. To implement the requirements in TOAD the KnoVA RA is used as a foundation. In the subsection 1.1.1 two challenges were identified in the first application scenario. These challenges are reviewed here to see whether the TOAD system addresses these challenges and therefore can suit as an exemplary implementation to be examined in the evaluation of the KnoVA RA.

**Challenge 1:** How can collaborative analysis be supported by VA systems? For example how can experts work at the same analysis questions and how can a system support knowledge sharing in the collaboration?

⇒ There are three questions to be answered in this challenge. The first two questions are concerning the collaborative work of experts. The third question is related to knowledge sharing. The TOAD system provides several features to support collaborative work. The design decision for a surface computer is a fundamental element of this, as it helps to overcome the limitations of classical desktop workstations [BBB<sup>+</sup>08].

The features of the TOAD system to support collaborative work are optimized for the usage of the surface computer. The implementation of these features is based upon the KnoVA RA. The first feature to support collaborative work are analysis paths, which allow for the traceability of the analysis process. This enables the experts to keep track of multiple concurrent analysis and therefore supports the collaboration. This is further supported by the visual history.

Another feature to support the collaboration is the workbench synchronisation. With this feature both the collaboration and the knowledge sharing between is supported. Analyst can directly share their knowledge with other analyst who are working on the same analysis question.

**Challenge 2:** How can implicit expert knowledge be extracted, in order to transfer results to other analysis tasks? For instance, how can findings be transferred from one trace-file to another to simplify the analysis of the other trace-file?

⇒ There are two important features in the TOAD system to support the extraction and the transfer of knowledge: session persistence and smart filtering. The session persistence allows to extract the complete course of the analysis in order to apply it in the same or in a different context. The sessions can be abstracted by the algorithms for generalisation of knowledge items. The smart filtering function is another feature that is implemented based upon the KnoVA RA to face this challenge. By defining the filter the experts can extract the knowledge the implicitly applied into the knowledge-base. Due to the generalisation of knowledge items these are then

applicable in different analysis situations and can also be shared with other experts who have access to the same knowledge-base.

In summary the TOAD system faces both challenges with the features that were implemented based upon of the KnoVA RA. Therefore TOAD is a valuable artefact to be examined in the evaluation of the KnoVA RA in the next chapter. Above this the TOAD system shows the practical applicability of the KnoVA RA, especially because all elements of the KnoVA RA are represented in the architecture or the implementation of TOAD.

## 6.2 CARELIS: Visual Support in Manual Data-Aggregation Tasks

The requirements for CARELIS can be separated into requirements which solely the visual representation and interaction with the system and requirements dealing with knowledge extraction and management, analysis traceability and knowledge sharing (compare 3.2.2).

In this subsection at first the newly developed CARELIS system and changes to its visual interface are introduced. After this it is shown how the requirements concerning knowledge extraction and management as well as analysis traceability and knowledge sharing can be implemented based upon the KnoVA RA.

In figure 6.5 the elements of the visual interface of the newly developed CARELIS system can be seen. In the figure eight main elements (a) – (h) can be seen, which support the task of interactive visual data aggregation: (a) patient record visualisation, (b) medical record visualisation, (c) personal record visualisation, (d) relative navigation, (e) direct navigation, (f) save button, (g) interaction highlighting, (h) highlighting of important information.

There are three interface elements supporting the general interaction with the system. A navigation element (d) to stepwise browse through the records. This element also allows to jump to the first or last record. A field to directly select a certain record by its ID (e) and a button to save the modified state (f).

**Visual Interface:** Several general enhancements to the visual interface are implemented in the newly developed CARELIS system. Comparable to the old CARELIS version records for which a certain probabilistic match weight is calculated are visualised on the interactive user interface. In contrast to the old version the user interface was changed so it can be adjusted, e.g. certain values can be excluded for information reduction [R15]. In addition to this the values can be re-arranged. In this way only the properties which are important for the analysis task are visualised. For instance the position of the column PatientId could be interchanged with the position of the column G-PA. In this way the most important features can be assorted to be visualised next to each other.

In addition to this the highlighting of the important information has been changed (h). Instead of a global scope for the highlighting, where all values in all currently visualised records are evaluated, a local scope is defined. By hovering the mouse over a certain record the user selects this record as the basis for the information highlighting. Only values that vary according to this record are then highlighted [R17]. The user can also



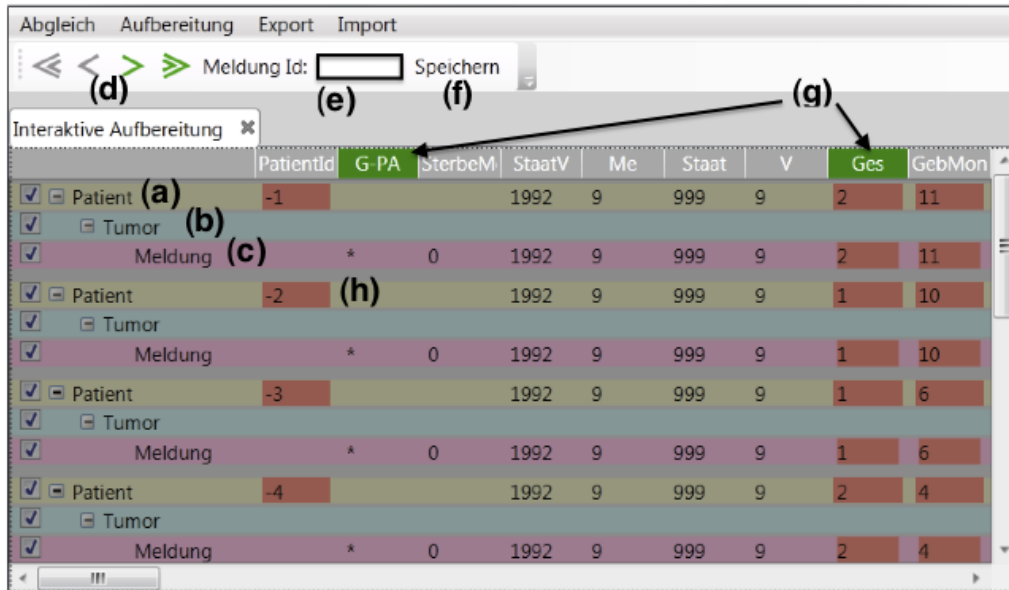


Figure 6.5: Elements of the Visual Interface of CARELIS.

exclude certain records from this comparison by selecting or de-selecting them by setting a combo box that is visualised right to the record. In figure 6.5 all records are selected. The largest part of the application is consumed by the elements to visualise the data (a) – (c). These interface elements represent the hierarchical data structure as present in the underlying database.

In the figure a patient record (a) is visualised with a yellow background. A patient represents a virtual entity subsuming a number of medical records and personal records (compare subsection 6.2). Medical records (b) are visualised with a green background. They subsume all available medical information such as the stadium of the tumour and the localisation of the tumour in the body. The personal records (c) are visualised with a pink background. They subsume all available personal information such as the date of birth or the encrypted values for firstname and lastname. As explained in subsection 6.2 a patient record aggregates a number of medical and personal records.

Other requirements towards the visual interface have not been implemented in the prototype. Especially the context sensitive information reduction [R16], the adjustable probabilistic distance [R18], the definition of best-of values [R24], the annotation of knowledge [R29] and annotations for knowledge sharing [R31] have been postponed to a future release.

**Aggregation:** The aggregation in the new CARELIS has undergone the largest changes. In the previous version transitive closures of records were visualised (compare 6.2). The concept of transitive closures was modified in the new CARELIS. Instead of presenting full transitive closures, only records which directly match to each other and

records that directly match with one of these are visualised in one closure, following [R15].

The aggregation itself is more flexibly than in the previous version of CARELIS. In the previous version there were two possible aggregations: a number of patients could be aggregated into one and a number of medical records could be aggregated into one. Also a patient could be split, which would result into a single new patient for all medical and personal records that was previously aggregated. Hence the whole aggregation was reverted.

These interactions are still present in the newly developed CARELIS. However in addition to this new medical or personal records can be created for a patient and new patients can be created from a context menu or via keyboard shortcuts. Single patients can be deleted, single medical or personal records can be deleted either by a context menu or by a keyboard short-cut. Single medical or personal records can be shifted from one patient to another either via drag-and-drop or by keyboard short-cuts for cut and paste. This allows for an ad-hoc interactive modification of aggregations [R32].

**Knowledge Derivation:** Several features for knowledge derivation were added to the newly developed CARELIS. In figure 6.6 the interface elements (a) – (b) to support the knowledge derivation can be seen. These elements for knowledge derivation are integrated directly into the interface for visual aggregation. They can be accessed whenever an aggregation was made by to create a direct and continuous knowledge extraction [R19].

The knowledge derivation results in the definition of rules for the aggregation, that can be used in an automatic aggregation processes. The rules express complex interrelationships and functional dependencies between the fields of the records [R23] and express value combinations and functional dependencies that lead to an aggregation of records. In the interviews it became clear, that from the large number of fields that are present in the visual interface usually only a small subset is relevant for a specific aggregation to be made. If all fields are automatically evaluated as important the number of possible rules to be derived can be very large. Depending on the strategy that is applied to generate the rules in the extreme case all possible combination of fields (the power set of fields) with all possible values from their respective value range has to be considered as a rule. This of course is impractical and will also result in a large number of rules which are very specific.

Therefore to create an intuitive knowledge derivation [R20], derive rules which abstract of discrete value ranges [R21] and feature the extensibility of the knowledge domain [R22] it was decided that the analyst manually define which fields were relevant in the current aggregation. The first step in the knowledge derivation therefore is the selection of the relevant fields. This is done by a click on the header of a specific field and thus integrated directly in the visual interact for aggregation [R19, R20]. After this the header will be shaded with a green background colour. This is visualised in figure 6.5 (g).

There are two possible ways to start the knowledge derivation. It can either be initiated on user interaction with a context menu, or it can be initiated when the user chooses to

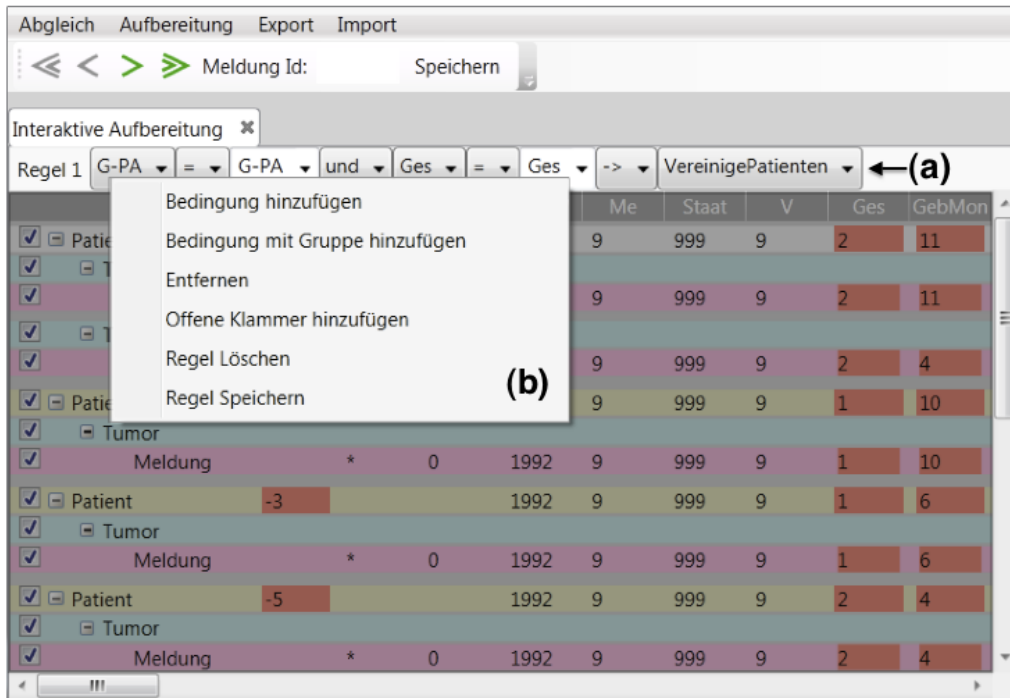


Figure 6.6: Rule Derivation in the CARELIS.

safe a set of aggregations. In both situations a list of possible rules that can be derived is shown on top of the visual interface for the aggregation. An example of this is shown in figure 6.6 (a), where one possible rule is visualised. The list of possible rules that can be derived is created automatically based upon the selection of fields that the user has made as shown in figure 6.5 (g).

The list of rules that was automatically generated can be edited by a context menu, that is shown in figure 6.6 (b). From this context menu the user can choose from bottom to top: to save a rule (Regel Speichern), to delete a rule (Regel Löschen) or to modify a rule by adding or deleting elements in the rule. This can be either adding a group of brackets (Offene Klammer hinzufügen), deleting items (Entfernen), adding a condition concerning fields (Bedingung mit Gruppe hinzufügen) or on of the logical operators AND, OR (Bedingung hinzufügen). The rules in CARELIS always result in either the aggregation of two patients to one patient, the aggregation of two medical records to one record, the allocation of a personal record to a patient or the allocation of a medical record to a patient.

The number of possible elements for the comparison is not limited. A rule can span over any number of fields, which can be connected either by the logical AND or the logical OR operator. In addition to that elements can be grouped by brackets to determine the sequence in which the rules is evaluated. In this way complex interrelationships and

functional dependencies can be expressed. The rule shown in figure 6.6 (a) aggregates two patients (conclusion `VereinigePatienten`) and spans over two variables: `G-PA` and `Ges`, which are connected by the logical operator `AND` (`und`). It expresses that when for two patients the fields `G-PA` and `Ges` are identical, then the patients can be aggregated. In addition to the equality operator `=` there are the operators `<`, `>`, `>=`, `<=` and `IN` available. The availability of the operators depends on the value range. The operators `<`, `>`, `>=`, `<=` are only available for fields which support ordinal comparison. The `IN` operator also works on nominal fields. The value range of the fields is encoded in the database. When the `IN` operator is chosen, a range of values can be specified by choosing the first and the last valid value in a range from the database. Additional ranges can of course be added for the same or other fields by a disjunction.

### 6.2.1 Realisation based upon the KnoVA RA

CARELIS is a commercial software project and is in production use at the EKN. For the implementation of the newly developed CARELIS the KnoVA RA was used as a foundation. In this subsection the features that are implemented based upon the KnoVA RA are reviewed. Illustrated in this figure 6.7 is shown how the architecture of the newly developed CARELIS is defined by the KnoVA RA. For architectural layers are shown in the figure, from top to bottom: the layer for the interactive visual interface, the layer for the KnoVA PM, the layer for the KnoVA MM and the data layer. For the sake of simplicity the interaction layer is left out. Also not illustrated in this figure are the layers for knowledge extraction and knowledge application, that are shown in figure 5.12 where the complete reference architecture is visualised. The integration of the elements of these lateral layers are referred to in the following description. Shown in the figure are the most important elements of the CARELIS system, comparable to the illustration of the elements of the TOAD system in figure 6.5 and the relationships between these elements. Some elements shown can correspond to classes that are present in the CARELIS source code. For instance `CARELIS.RULEENGINE` is such an element. Other elements shown in the architecture do not directly correspond to classes. The `CARELIS.MODEL` for instance corresponds to a collection of classes.

In the interactive visualisation layer a screenshot of the visual interface for manual aggregation of the newly developed CARELIS is shown. Comparable to the TOAD system CARELIS is also implemented on the Model – View – ViewModel pattern [Mac10]. The visual interfaces consumes the viewmodel, as indicated with a respectively named relationship between `CARELIS.VIEWMODEL` and the screenshot of the CARELIS system. Comparable to the implementation of TOAD the `KNOVA META MODEL` is the base class for the class `CARELIS.VIEWMODEL`. The viewmodel in the figure has a relationship to the `CARELIS.MODEL`, which represents the classes of the CARELIS system which provide access to the underlying database in the data layer.

Unlike the TOAD system the viewmodel of CARELIS is a passive viewmodel, which is only consumed by the view but provides no business logic. Instead all business logic is carried out to the two classes `CARELIS.VIEWMODEL BUILDER` and `CARELIS.RULEEN-`

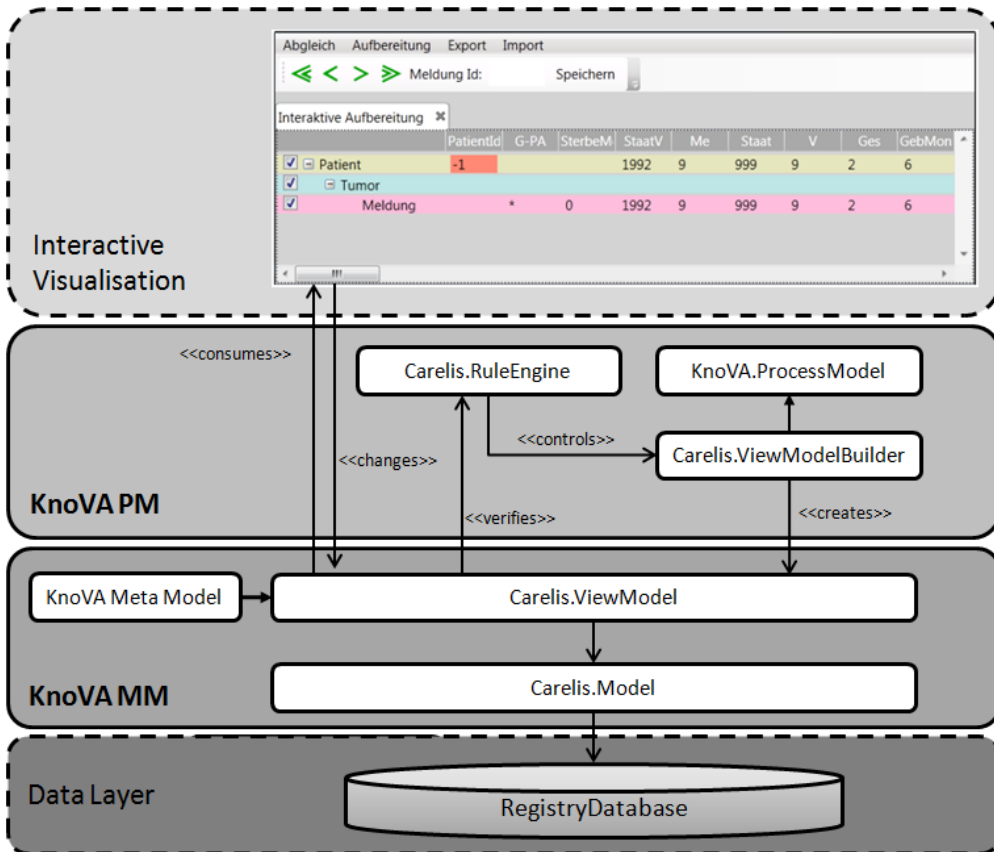


Figure 6.7: Integration of the KnoVA RA into the CARELIS System.

GINE.

When the viewmodel changes upon user interaction at first the changes in the viewmodel are propagated to the CARELIS.RULEENGINE. This is indicated by an annotated association between CARELIS.VIEWMODEL and CARELIS.RULEENGINE in the figure. The CARELIS.RULEENGINE then verifies the changes based upon business rules. This is also indicated with an annotated. If the changes are valid, the CARELIS.RULEENGINE propagates the changes in the viewmodel to the CARELIS.VIEWMODELBUILDER. This class implements a creational pattern. It creates all viewmodels which are actually consumed by the view. Therefore it creates a new instance of the viewmodel which includes the changes.

In addition to this the CARELIS.VIEWMODELBUILDER implements the KNOVA.PROCESSMODEL. It therefore stores all analysis states, which have been made in the current analysis session. In CARELIS an analysis session is defined as the time span between the first visualisation of a set of matched records and the time the user chooses to save the possibly made aggregations. When a new set of records is loaded, a new analysis

session starts. The steps recorded by the CARELIS.VIEWMODELBUILDER up to that point are mapped to the classes of the CARELIS.MODEL and made persistent.

**Visual Interface:** Some of the changes to the visual interface were implemented based upon the KnoVA RA. The features for customizing the visual interface make use of the KnoVA MM. When the user changes the visual interface, of course these changes are reflected in the viewmodel of the visual interface, which is based on the KnoVA MM. Due to the architecture of the CARELIS system these changes are then evaluated by the CARELIS.RULEENGINE. Here special rules exist to make the changes to the visual interface persistent. These rules are later evaluated by the CARELIS.VIEWMODELBUILDER which then creates the viewmodel depending on these rules (e.g. changing the order of fields or the selection of fields visualised).

Although postponed to a future release the integration of the CARELIS.RULEENGINE in this process allows for an easy integration of context sensitive information reduction according to [R16] in future. The same is true for the adjustable probabilistic distance according to [R18]. Both features effect the way the viewmodel is generated by the CARELIS.VIEWMODELBUILDER. Therefore special rules derived from the changes in the viewmodel can be used to modify the creation of the viewmodel. The rule set used by the CARELIS.VIEWMODELBUILDER will have to be extended so that the current state of the analysis system can be evaluated in the viewmodels creation process.

**Aggregation:** In the previous version of CARELIS sets of aggregated records could only be de-aggregated completely, resulting in a large number of records which then often at least partially needed to be aggregated manually again. The extended flexibility of stepwise aggregation and ad-hoc interactive modifications [R32] was implemented on the foundation of the KnoVA RA. As explained above changes to the viewmodel are evaluated by the KNOVA.RULEENGINE. Valid changes are then stored as a step of the KNOVA.PROCESSMODEL by the CARELIS.VIEWMODELBUILDER. Thus when the user chooses to save the aggregation the whole trace can be made persistent. Therefore all steps in this trace can be reverted step wise. The extended flexibility of the aggregation therefore is a result of the flexibility provided by the KnoVA PM.

**Knowledge Derivation:** The features for knowledge derivation in the new CARELIS are also implemented based upon the KnoVA RA. All changes to the records are traced by the CARELIS.VIEWMODELBUILDER as shown above. The viewmodel derives from KNOVA META MODEL. The instances of the viewmodel therefore represents the model instance of a reference analysis state. To create a reference analysis state the CARELIS.VIEWMODELBUILDER adds the analysis state that is created internally.

When the user triggers the rule derivation, as explained in subsection 6.2, the current viewmodel is compared with the previous viewmodels. The changes in the viewmodel are evaluated. In this evaluation only the fields selected by the user for knowledge derivation are examined. The power set of all possible rules that can be derived based upon these changes (combination of all variables, all possible operators etc.) is generated. In this process of rule generation the rules are also generalised by replacing the meta model instances with their generic representation in the KnoVA MM.

From this power set of rules the user then chooses the rules which he wants to extract.

In addition to that the user can define additional constraints and make changes to the rules. The definition of value ranges for instance is something that is excluded from the automatic rule generation. The list of all possible value ranges that might be applicable is often too exhaustive and may result in many thousand different rules. In early prototypes this list was generated and presented to the user. It quickly became clear that the MRTs are not willing to select the applicable rules from the list. Also the generation process in some cases took longer than a manual definition of the respective rule.

### 6.2.2 Conclusion

In the implementation of the new CARELIS system the focus was on the implementation of features that benefit from the integration of the KnoVA RA. Above was shown that the new CARELIS responds to the requirements [R15 – R23], [R29 – 32]. The requirement for priority management [R26] was addressed in the implementation. Although not specifically mentioned above the database in which the rules are stored contains a field for the priority of the rule definition. This was not used in the prototype though, as to define a priority a tool to compare and manage all available rules is necessary. This is independent from CARELIS and the KnoVA RA and therefore was not implemented in the prototype. The same reasons also lead to the decision to postpone the requirement for knowledge management [R25] and [R27 – R28].

As in subsection 6.1.2 in this subsection now the challenges identified in subsection 1.1.2 in the second motivating scenario are reviewed to see whether CARELIS addresses these challenges to investigate whether CARELIS is a result that can be used in the evaluation of the KnoVA RA.

**Challenge 3:** How can a VA system support the representation of expert knowledge that is implicitly applied by the expert in the analysis process?

⇒ In the CARELIS system the knowledge is applied on the basis of the KnoVA RA. The viewmodel of the visual interface for aggregation implements the KnoVA META MODEL.

**Challenge 4:** How can this knowledge be extracted into a knowledge-base, in order to be re-applied the analysis process? For instance, how can this knowledge be used to support automatic aggregation steps?

⇒ CARELIS supports the extraction of knowledge in several ways. In the visual interface knowledge is extracted by the modification of the interface itself. The configuration of the interface is translated into business rules, which later are evaluated by the CARELIS.VIEWMODEL BUILDER to modify the process of viewmodel generation.

Above this CARELIS features the derivation of aggregation rules, which can later be used in the automatic matching process. This process precedes the manual aggregation and reduces the number of record sets that have to be aggregated manually. Therefore it reduces the workload of the MRTs. An advantage of the integrated knowledge derivation is, that in this way changes in the process can be reflected.

For instance when the decision of whether two records should be aggregated or not changes in future (e.g. if other fields are important for this decision), then the rules can be derived as described above.

In summary CARELIS faces both challenges based upon the KnoVA RA as a foundation to implement these features. Therefore CARELIS is a valuable artefact to be examined in the evaluation of the KnoVA RA in the next chapter. Above this CARELIS shows the transferability of the KnoVA RA to another implementation. This is also valuable for the evaluation, especially because TOAD and CARELIS both address different analytical problem classes (the trace of errors vs. the aggregating of records) across different application domains (automotive vs. health care).

### 6.3 Summary

In this chapter two exemplary implementations of VA systems were presented, which respond to the challenges identified in the two motivating scenarios.

At first in section 6.1 the TOAD system for the VA of ICNs was introduced, facing the challenges identified in this scenario. TOAD is a VA system that makes use of an interactive surface computer and its main objective is collaborative VA. To support this five different features for knowledge sharing and extraction were introduced: the concept of analysis paths, a visual history function, session persistence, workbench synchronisation and a smart filtering function. The TOAD system is implemented based upon WPF and the Microsoft .NET Framework. Its architecture, shown in figure 6.4, is based upon the KnoVA RA and key features, such as the analysis paths, are implemented based upon elements of the RA.

In section 6.2 the CARELIS system for visual support in manual data-aggregation tasks was introduced. The new implementation of CARELIS features the semi-automatic derivation of aggregation rules, to derive knowledge from manual aggregation tasks. It faces the challenges of the second motivating scenario, that were also identified in section 1.1.

CARELIS is implemented in WPF and based upon the Microsoft .NET Framework. Comparable to TOAD, the architecture of CARELIS is based upon the KnoVA RA. In figure 6.7 it was shown how the KnoVA RA builds the foundation of CARELIS, where state changes that occur upon user interaction are used to derive knowledge that can be re-applied in other analysis situations.



## 7 Evaluation

Objective of this chapter is the evaluation of the KnoVA RA. The evaluation includes a reflection on the exemplary implementations TOAD and CARELIS described in the previous chapter. It follows the recommendations of Frank [Fra07], who provides guidelines for the evaluation of reference models, such as the KnoVA RA. In this chapter the recommendations of Frank are supplemented by the recommendations of Isenberg [IWR<sup>+</sup>10] on the evaluation of VA systems.

The objective of the evaluation is to determine whether the KnoVA RA is suitable to fulfil the requirements for knowledge-based VA that were analysed in chapter 3 and whether it addresses the challenges identified in section 1.1. In summary, the objective of the evaluation is to show that the approach presented in this thesis responds to the research question for concepts and methods to represent and extract expert knowledge applied in the process of VA, to make it reusable.

This chapter consists of five sections. Section 7.1 introduces the methodological foundation for the evaluation based upon the work of Frank and Isenberg.

Based upon these foundations section 7.2 introduces a concept for the evaluation of the KnoVA RA. In this concept, the recommendations of the previous two sections are examined and used to identify criteria and methods for the evaluation. These are then used to structure the evaluation in section 7.3. Section 7.4 critically reflects the results of the evaluation before section 7.5 summarises the chapter.

### 7.1 Methodological Foundation

In this section the methodological fundamentals for the evaluation are introduced. At first in subsection 7.1.1 the fundamentals to structure the evaluation and to provide an epistemological foundation are introduced. For this an approach for the evaluation of reference models is presented, which provides a number of criteria for the evaluation of reference models.

After this in subsection 7.1.2 techniques for the evaluation of VA systems are introduced.

#### 7.1.1 Evaluation of Reference Models

With the KnoVA meta model and the KnoVA classification as its foundation the KnoVA RA provides a detailed and formal description of the domain of knowledge-based VA. In addition to that, by definition of the KnoVA process model, the introduction of an UML based representation and exemplary algorithms for knowledge extraction and application it aims to serve as a blueprint for knowledge-based VA systems.

Thus it fulfils Franks description of a reference model [Fra07]. In this work, Frank examines different approaches for the evaluation of reference models and process models and defines a framework of four perspectives to structure the evaluation of reference models. The framework encompasses the economic perspective, the deployment perspective, the engineering perspective and the epistemological perspective. Each perspective defines

a number of evaluation criteria. In the following subsections these perspectives are discussed and the criteria that Frank defines are examined as described in [Fra07] in order to provide a methodological foundation for the evaluation of the KnoVA RA.

### The Economic Perspective

The economic perspective subsumes criteria to analyse cost and benefits. Frank introduces three areas in which the usage of a reference model can produce cost.

**Costs for introduction:** This criteria subsumes initial costs such as costs for the acquisition or the licensing of a reference model or for tools to support the usage of the reference model as well as initial effort for professional training. The first ones are largely specific for a certain reference model, whilst the latter ones largely depend on the complexity of the reference model.

**Costs for integration:** This subsumes costs for the examination of existing structures and the integration or adaption of the reference model to a specific task or to existing structures. Hence, in contrast to costs for introduction the costs for integration concern the retro fitting of the reference model to existing structures.

**Costs for maintenance:** This criteria subsumes costs for the maintenance of tools, costs for adaption of the model to new requirements and costs that arise from the need to sustain knowledge and skills concerning the reference model.

Opposed to costs Frank advocates the comparison with the financial benefits of the reference model. Objective for the usage of reference models is to approach new challenges and to increase efficiency in the approach of existing challenges. To measure this Frank introduces three criteria and gives examples for each of the criteria.

**Efficiency:** To examine concerning efficiency is the efficiency of software development and maintenance for reference models targeting software development or the increase in efficiency for reference models targeting business and management processes.

**Flexibility:** To measure the flexibility of a reference model Frank suggests to examine the dependency from IT vendors and the openness, where the less dependent a reference model is from a specific IT vendor, the greater the flexibility, the expressive power of the reference model and the relationship to other IT artefacts.

**Protection of Investment:** Here Frank suggests to examine the spreading of a reference model. The more wide spread a reference model is used, the better the investment is protected as a wide usages on the one hand promises a longevity of the reference model and on the other hand evidences a profitableness. Another aspect to examine concerning the protection of investment is the robustness of the reference model against technological change.

### The Deployment Perspective

The success of a reference model heavily depends on its users [Fra07]. Frank describes that the ability and the willingness of the user to deal with the model is an important factor for the success of the model. To reflect this factor, the framework suggests examining the following three aspects.

**Understandability:** To measure the understandability Frank suggests to examine the

comprehensiveness of the documentation, application scenarios and examples as well as the familiarity and publicity of the modelling language and terminology and the intuitiveness of the model for different stake holders.

**Appropriateness:** To measure the appropriateness of a reference model Frank suggests to examine the amount of support for purposes that are relevant for the user (e.g. the quality of the solutions that can be implemented with the reference model) and to examine whether the model supports technologies that can be expected in near future.

**Attitude:** With attitude Frank encourages to examine whether the reference model bear conflict potential. Especially whether resistance of the users against the usage of the reference model or solutions based upon the reference model is to be expected.

### The Engineering Perspective

Frank defines that a reference model is a design artefact to be regarded as a specification of possible solutions to a range of problems. He describes two pivotal questions concerning the engineering perspective: *Fulfils the model the requirements that have to be taken into account?* and *Is the specification suited for supporting the intended purposes of the model?*. To analyse these questions he introduces four aspects.

**Definition:** To evaluate this aspect Frank suggests to examine the criteria whether the reference model provides a comprehensive description of the intended application domains and purposes.

**Explanation:** This aspects encompasses the criterion to assign model elements to the requirements, to justify design decisions, to discuss design compromises and resulting drawbacks and to discuss alternative approaches.

**Language Features:** Here he encourages to examine criterion such as the level of formalisation, the extensibility, the supported conceptual views, the integration of views, tool support, support for concepts and the adaption of models as well as to foster model integrity.

**Model Features:** Concerning this aspect he suggests to examine the formal correctness/consistency of the model, the model architecture, the uses of classes, generalisation, specialisation and the use of modularisation and flexibility.

### The Epistemological Perspective

The fourth perspective that Frank introduces is the epistemological perspective that he motivates with the similarity between reference models and scientific theories. Like theories reference models are supposed to provide representations for classes of problems or instances. Also like scientific theories, reference models are contributions to the body of knowledge in their respective domain. Therefore, he reasons that it is sense full to apply criteria for the evaluation of scientific theories to the evaluation of reference models. He points out that as a major difference a theory aims at describing the world as it is; hence, they are descriptive, whereas reference models are also prescriptive. In summary, he introduces four interrelated aspects to examine concerning the epistemological perspective.

**Evaluation of theories:** To review this aspect Frank recommends a precise descrip-

tion of core concepts with respect to corresponding real world concepts facilitated by a precise description of underlying assumptions.

**General principle of scientific research:** Here Frank provides three general principles to examine: Abstraction, Originality and Judgement.

**Critical reflection of human judgement:** Concerning the critical reflection Frank recommends to take the subjective nature of underlying decisions into concern as well as a bias through possible familiarity with modelling languages and the fact that certain criteria such as a high degree of spreading may mistakenly imply quality.

**Reconstruction of scientific progress:** To reconstruct the scientific progress Frank suggests to discuss the long term goals of the research and to elaborate the documentation with respect to these principles as well as the discussion of alternatives.

With his work, Frank provides a catalogue of criteria to examine for the evaluation of reference models. He encourages the evaluation of the model against multiple perspectives, preferably all. To support this evaluation he provides a number of aspects and a rich set of criteria to create evaluation scenarios.

Frank also mentions that the evaluation of reference models is complex due to their prescriptive nature. It is hard to identify criteria to evaluate the quality, effectiveness or efficiency of a reference model. Therefore, a mixed approach is recommended that discusses aspects of the reference model itself regarding a tailored set of criteria. Additionally he also encourages examining artefacts designed by the use of a reference model in order to make assumptions about the quality of the model. The basic idea behind this is that high quality artefacts will at least evidence a suitability of the reference model for the domain or the approached problem.

While Frank provides perspective and criteria to evaluate reference models he does not provide specific methods or approaches for evaluation. Before section 7.2 introduces a concept for the evaluation of the KnoVA RA, methods for the evaluation of VA systems are being examined to provide a methodological background for the evaluation of the KnoVA RA in the two application scenarios.

### 7.1.2 Evaluation in Visual Analytics

In [IWR<sup>+</sup>10] Isenberg et al. examine the state-of-the-art of evaluation of VA systems. The following section gives a summary of this article. The article declares, that in VA there is a lack of solid findings, models and theories and that as a result, VA still has a long way to go to be considered a mature technology. They further point out that evaluation will play a crucial role in the process of VA to become a mature technology. They therefore examine particular problems in the evaluation of VA applications and present a set of recommendations for these tasks.

The KnoVA RA aims to model the process of information visualisation. It intends to serve as a foundational architecture for VA applications. As described by [IWR<sup>+</sup>10] for the evaluation of reference models, it is useful to examine whether the applicability of the reference models to create useful applications in a qualitative approach in which the RA is examined according to a set of criteria. According to [IWR<sup>+</sup>10] the key aspects for

quality in VA systems are effectiveness, efficiency and user satisfaction and therefore perspectives for a comprehensive evaluation should involve the examination of users, tasks, artefacts and data.

The level of expertise of the users of VA systems can range from experts to naive users. Depending on the objective of an evaluation, it can be sufficient to replace experts with a naive user group. The general usability of a system (e.g. readability of textual labels or appropriateness of interaction metaphors) is an example where this might be sufficient. However if a VA systems targets expert users, for a profound judgement of the suitability of the system it is necessary to thoroughly understand the needs of the experts and to integrate expert users in the evaluation. Tasks in VA can be complex and respectively require long periods to complete. With such lengthy tasks, it has to be considered whether the findings of lab-experiment can scale with the complexity of such tasks. Artefacts refers to all elements of VA systems, such as graphical representations on a detailed level, software tools on a higher level and even the suitability of technologies in general. Finally, the data to be analysed likewise influences the evaluation, as VA often deals with combinations of heterogeneous or highly-dimensional data, which further increases the complexity of the evaluation especially in lab-experiments. Complexity reduction concerning tasks and data is an approach for the evaluation in lab-experiments. However, it has to be shown that the evaluation results scale with more complex problems.

[IWR<sup>+</sup>10] identify two main antithetical approaches for the evaluation of VA applications:

**Empirical Evaluation:** Methods for empirical evaluation include field studies, field experiments, laboratory experiments, experimental simulations, judgement studies, sample surveys, formal theories and computer simulation. This list is based upon the work of [Car08]. Another classification of an empirical approach can be found in [Mun09] where the process of creating visualisation is structured into four nested levels, domain problem characterisation, data and operation abstraction design, encoding and interaction technique design and the design of algorithms. They argue that in each of these levels, distinct evaluation methods are suitable and hence in the evaluation of VA systems it has to be determined which level needs to evaluate.

**Competition and Contests:** Competition and Contests are widely used in the scientific community, where researchers face certain challenges, such as the VAST challenge<sup>1</sup>. Other research communities, like the graph drawing community, the information visualisation community and the software visualisation community, provide other challenges in the scope of VA research.

## 7.2 Evaluation Concept

This section introduces the concept for the evaluation of the KnoVA RA. Following Frank's recommendations [Fra07] the evaluation will examine four perspectives: the

---

<sup>1</sup> The VAST challenge is an annual contest alongside the IEEE Conference VA Science and Technology (VAST), where researchers can evaluate their VA approaches against a given dataset under pre-defined analysis questions.

economic perspective, the deployment perspective, the epistemological perspective and the engineering perspective.

In the scope of this thesis, the KnoVA RA was applied in two real-world engineering scenarios from different application domains. These scenarios built the foundation of the evaluation concept. Table 7.2 shows a matrix of these scenarios and Frank's four perspectives. The table represents a matrix between the four perspectives and the two application scenarios. Each cell in this matrix represented a certain aspect of the evaluation. In each cell, the criteria that are the objective of the respective aspect are summarised. In addition to that, the numbers shown in the cells are a reference to the subsections of this chapter that refer to the respective part of the evaluation.

	<b>Scenario 1: Visual Analysis of In-Car Bus Communication Networks</b>	<b>Scenario 2: Visual Support in manual Data Aggregation Tasks</b>
<b>Economic</b>		Objective: Efficiency, Flexibility Method: Field Study (7.3.4)
<b>Deployment</b>	Objective: Understandability, Appropriateness, Method: User Study (7.3.1)	Objective: Understandability, Appropriateness, Method: User Study (7.3.3)
<b>Epistemology</b>	Objective: Applicability Method: Prototype, Critical reflection (7.4)	Objective: Transferability Method: Prototype, Critical reflection (7.4)
<b>Engineering</b>	Objective: Definition, Explanation Method: Developer Survey, Critical reflection (7.4)	

*Table 7.1: Overview of the Evaluation of the KnoVA RA.*

Reference models are theoretical concepts, which only gain meaning by their implementation. It is not possible to evaluate a reference model self contained [Fra07]. Frank therefore suggests the evaluation of artefacts with a design that is using the reference model. The evaluation concept for the KnoVA RA follows this recommendation. The evaluation examines the VA systems designed in the two application scenarios. This is complemented by a critical reflection of the engineering process according to the influence of the KnoVA RA to this process and by an economic reflection of the outcome.

The figure shows that for each application scenario three of Franks perspectives are evaluated. Following the principle of methods diversity [Fra06] the evaluation concept covers the four perspectives multiply, with the exception of the economic perspective and the engineering perspective. For each perspective and each scenario an objective of the evaluation is shown in figure 7.2 as well as the applied methods to pursue the objective. The following subsections outline the concept of the evaluation in the four perspectives.

### 7.2.1 Concept for Evaluation of the Economic Perspective

From the two scenarios introduced, only the second scenario can be considered for the economic perspective. External constraints on the TOAD project prohibited an examination of the economic perspective. Nevertheless, some testimony on the economic perspective could be obtained in the user study for the deployment perspective (compare 7.2.2) of this scenario.

In the second scenario, the economic perspective is analysed towards the efficiency and the flexibility of the tools created in this scenario in a comparative field study. The field study examines the existing tools for the manual data aggregation towards these criteria. To accomplish this examination, CARELIS is introduced to the working environment of the MRTs. It is then examined with an exemplary data set.

The evaluation of the other criteria for the economic perspective was not feasible in the context of the real world scenarios. The costs for introduction of the KnoVA RA cannot objectively be measured. In order to achieve this, it would be necessary to estimate the additional or saved development effort for the use of the KnoVA RA. The usage of the RA directly relates to required features in the application scenarios. There are two approaches to estimate the cost for a software system. Either directly, by measuring the cost of the implementation or by estimates.

Another implementation for comparison is not possible because of the additional costs that arise by such a project. In addition to that, this approach is also not efficient because two software projects are hardly comparable. Estimates then are the only way to compare the costs for a software system that can be made without additional cost. Estimate for the cost of a software development project always contain a large amount of uncertainty. This makes the comparison of the results disputable.

These arguments are likewise transferable to costs for integration and maintenance. Both are lacking the foundation for a comparison. Therefore, only the efficiency and the flexibility can be evaluated in this context. In addition, the protection of investment will be briefly discussed in the evaluation.

### 7.2.2 Concept for Evaluation of the Deployment Perspective

The prototypes that were developed allow for an evaluation of the deployment perspective for both scenarios by a deployment of the prototypes and a succeeding evaluation of the prototypes. Frank recommends examining the criteria understandability and appropriateness in the deployment perspective. To evaluate such qualitative criteria [IWR<sup>+</sup>10] recommend user studies as method.

Therefore, both prototypes are deployed in the evaluation and then bespoke user studies for the prototypes are performed in order to evaluate them according to their understandability and appropriateness.

### 7.2.3 Concept for Evaluation of the Epistemological Perspective

Objective of the evaluation in the epistemological perspective is the applicability of the KnoVA RA to real world problems. This will be done in section 7.4 by a critical reflection of the prototypic implementations and the discussion of the contribution of the KnoVA RA to the body of knowledge by a reflection of the applicability of the RA to different problem classes.

Above this the scientific approach and the general principles abstraction and originality will be reflected in this section based upon the prototypical implementations introduced in chapter 6 and by a discussion of the modelling approach and the methodology that was applied in the design of the RA in chapter 5.

### 7.2.4 Concept for Evaluation of the Engineering Perspective

To examine the engineering perspective Frank suggests considering whether the specification suited for supporting the intended purposes of the model. In the first application scenario, a team of ten student developers worked on the implementation of the prototype. For the evaluation of the quality of the KnoVA RA from an engineering perspective the criteria introduced by Frank concerning this perspective are translated into a survey for these developers in subsection 7.3.2. This result of this survey is then critically reflected in section 7.4. A qualitative evaluation of the KnoVA RA in the second application scenario is prohibited on the outset of the subjectivity of the involved engineers. Therefore, the survey was also held with a control group of software engineers who have previously not implemented a system based on the KnoVA RA. The results of both surveys are then summed up in 7.3.2.

## 7.3 Accomplishment and Results

The qualitative evaluation of the KnoVA RA results in three artefacts. In subsection 7.3.1 an user study of the TOAD system, where the automotive test engineers as the target users of the TOAD system are asked to value the prototypic implementation towards its usefulness and towards their requirements. This evaluation focuses on the deployment perspective but also incorporates some results that can be used to discuss the economic perspective.

After this the engineering perspective is evaluated in subsection 7.3.2 by a survey amongst the engineers who implemented the TOAD system. They were asked to evaluate the KnoVA RA concerning the criteria for the engineering perspective that have been introduced by Frank.

Lastly in subsection 7.3.3 a comprehensive survey based usability study of the user interface for knowledge extraction in the second application scenario, that were introduced in section 6.2, is presented.



### 7.3.1 TOAD User Study

To evaluate the deployment perspective, following Frank's recommendations in [Fra07] the prototype of the TOAD system was used in an evaluation with the target users. For this a user study was performed following the recommendations in [IWR<sup>+</sup>10], where user studies are introduced as appropriate method.

In addition to feedback gathered along the UCD process, which lead to the requirements in chapter 3, TOAD was evaluated in a focus group with the test engineers at the car manufacturer. During this focus group, the TOAD prototype was used in the notion of a technology probe [HMW<sup>+</sup>03]. The engineers were invited to bring along their own data and to test TOAD with it.

The objective of this focus group was to assess the potential value of the system for the VA workflow within the company. A focus was on the aspect of collaboration and knowledge sharing mechanisms in the prototype that are based on the KnoVA RA. The intention of this qualitative study was to evaluate whether the features implemented based on the KnoVA RA are considered useful by the target users.

This is following the recommendations of [IWR<sup>+</sup>10], where end user evaluation is encouraged as a method to evaluate VA systems. Based upon the recommendations by [Fra07] such a qualitative study can provide evidence of the usefulness of a reference model by showing that the reference model can be used to implement reasonable real world system designs.

#### Methodology

Nine participants with a background in in-car communication analysis took part in the focus group. The aim was to gather feedback from parties with different perspectives. Roles of the participants in the company were (a) engineers responsible for the analysis of in-car communication traces, (b) sub-contractors collaboratively working with engineers of the company, (c) IT professionals responsible for the deployment of analysis tools within the company, and (d) managers responsible for analysis workflows and processes within the company. Due to practical reasons, the evaluation was conducted in a room on the company campus. Overall, the evaluation session took 2 hours.

After a brief video introduction, the participants were invited to join a hands-on session of the actual TOAD prototype. For this purpose, the prototype was deployed on a multi-touch capable stand-alone PC. The display of the system was 24 inches in size, and was positioned horizontally, so that it mimicked a smaller version of the bigger interactive tabletop. As the aim was to gather qualitative feedback only, the usage of a smaller-scale multitouch device seemed feasible.

Feedback by the participants whether this approach would be sufficient for them to get an adequate impression about TOAD was positive. During the hands-on session, the participants were encouraged to load their own data into the system and to collaboratively explore the data. After the hands-on session, feedback gathered regarding the usability of the system, as the participants were asked to discuss questions regarding the analysis workflow, the appropriateness of used visualisations, the interaction concept,

and the suitability of a surface computer for co-located collaboration. This discussion was recorded as an audio file and evaluated in addition of the focus group. Due to legal restrictions, the recording of video data during the hands-on was not possible. However, notes were taken from the observation of the participants during the hands-on session.

## Results

The overall feedback about usability and potential utility was very positive. In fact, the engineers consider using the prototype in their daily work. They found that the system is dynamic and flexible enough to support collaboration within the company. They also stated that TOAD exactly reflected the analysis workflow of typical analysis sessions. A positive result in this area was expectable, as the TOAD prototype was designed in close collaboration with the target users.

In the following, therefore the focus lays more on the general and critical discussion that triggered by the prototypic systems. In line with the requirement to integrate heterogeneous visualisations, the participants argued for having a broader scope and extending the system. For instance, one participant mentioned that the system could also be used for the definition of the FSM themselves. At the other end of the analysis workflow, it could be used to debug problems on the functional level within individual ECUs. They highlighted that the information density within the system is insufficient at certain points. This particular targeted the wish for higher resolution displays that contain more information, and the wish to display details regarding specific aspects on demand.

Nevertheless, consensus between the participants was that, due to the increasing complexity ICNs, collaborative workflows will strongly increase in the future. One participant explicitly stated, "there is no way around collaborative [VA] processes in the future".

Regarding *costs*, there was also a consensus that the financial effort for introducing a collaborative technology, including hardware and software, was an insignificant aspect if an increase in effectiveness or efficiency could be achieved. In such large industrial settings, staff costs outweigh technology costs by far. Thus, an initial investment in a collaborative technology would be quickly compensated. Yet, technical reliability and robustness of hard- and software is a crucial prerequisite. Even though this is a purely qualitative feedback, based upon a rough intuitively estimation of the experts (including the management of the analysis group), this result reflects some aspect of cost, which [Fra07] sees as a key aspect for the evaluation of reference models.

Further questions targeted methods for *collaboration and knowledge sharing* enabled by the system. They found that TOAD was particularly beneficial regarding the intra-disciplinary forming and verification of hypotheses about the cause of errors in in-car communication. Within a group of engineers with similar domain knowledge, the system could facilitate getting ideas about the cause of an error faster than before, as it enables individual and group work at the same time, at the same place and in the same workspace. They also assessed that the system was valuable for inter-disciplinary work involving experts with different domain knowledge. Benefits were seen in the integration of viewpoints of other experts in the analysis and in the identification of an expert who is respon-

sible for fixing an error in the system design. A statement of a participant subsumes this aspect quite strikingly: *“Analysts often play analysis ping-pong because it is not clear who is responsible for fixing a particular problem. In these cases, the system would provide clear benefits by bringing these parties together at the same table”*.

Further, the system was helpful for handing work over to experts of other fields. They said that the system could be used at neuralgic intersection points between different groups or departments.

The history function integrated in the system based upon the KnoVA RA was assessed as invaluable to reproduce interactions of analysts that led to the identification of an error, which is often necessary to finally identify causes of errors in the system design and on the levels of the electronic control units in the car.

Following up on this aspect, it was further inquired about the importance of incorporating support for *knowledge sharing by asynchronous, collaborative settings* into the system. Participants considered asynchronous collaboration important when including sub-contractors into the analysis process, which are often not located in the same building than the company, and for collaboration of analysts distributed in offices among the company campus.

Also, asynchronous VA was found necessary when using multiple collaborative devices, which are used by different teams. As a further extension, it was mentioned not only to include support for asynchronous collaboration, but also for distributed collaboration, possibly including a mixture of different devices, such as interactive tablespots, interactive whiteboards, laptops, and projectors. In addition to the filter functionality, which can be used to extract knowledge and to highlight points of interest in the system, it was also suggested to integrate written or drawn annotations in the history and throughout the system for the support of asynchronous collaboration settings. By the combination of annotations, the interaction history, and the visualisations included in the system, the system was considered to work in asynchronous settings, too.

### 7.3.2 TOAD Engineer Survey

To evaluate the engineering perspective Frank suggests to examine whether the model fulfils the requirements and whether the specification is suited for supporting the intended purposes of the model. To approach these questions for the KnoVA RA a survey amongst the engineers of the TOAD system was performed.

Ten developers developed the TOAD system within a student projects over the course of one year. Features for knowledge extraction and sharing are integrated based upon the KnoVA RA (see section 6.1). For this, the concepts of the KnoVA RA were introduced to the developers after the requirements analysis in this scenario. The KnoVA RA then was used as the basic architecture for the TOAD system.

Objective of the survey was to gather qualitative feedback from the engineers on the RA and its application in the engineering process.

## Methodology

Frank names a number of criteria are to evaluate the engineering perspective. These are summarised in 7.1.1. To evaluate the KnoVA RA against these criteria in a survey it is necessary to translate the criteria into a questionnaire.

Frank introduces four criteria to be examined to evaluate the engineering perspective: definition, explanation, language features and model features. These four criteria are used as the foundation of the questionnaire. For each of these criteria a set of questions are included into the questionnaire. In the following, these criteria are revisited and the questions for the questionnaire are introduced alongside the criteria they relate to. The criteria are defined as in 7.1.1. In front of each of the questions, a keyword to identify the question is introduced. The keywords are included here as a short reference to the question that is used in the subsequent text and in the respective tables and figures.

**Criterion Definition:** For this criterion, three questions are included in the questionnaire. Possible answers for these questions are yes, no and other. The participants are encouraged to make additional comments when choosing other. When participants answer yes to one of the questions this indicates that the RA fulfils the criterion. Accordingly, no, as answer will give evidence that the participants would consider that the RA does not fulfil the criterion. A textual answer needs further examination.

1. **Application Domain:** *Does the RA describe its application domain sufficiently accurate?* The objective of this question is to evaluate whether the description of the RA describes the application domain of the RA in sufficient detail. This question therefore especially targets the KnoVA classification, which describes the domain of knowledge-based VA.
2. **Purpose:** *Does the RA describe its application purpose?* The objective of this question is to evaluate whether the description of the RA describes the objective of the RA and its intended usage.
3. **Suitability:** *Is the RA suitable for its intended application purpose?* The objective of this question is to gather qualitative feedback from the engineers whether the RA is suitable to fulfil its intended application purpose.

**Criterion Explanation:** For this criterion, two questions are included in the questionnaire. Possible answers for these questions are yes, no and other. The participants are encouraged to make additional comments when choosing other. When participants answer yes to one of the questions this indicates that the RA fulfils the criterion. Accordingly, no, as answer will give evidence that the participants would consider that the RA does not fulfil the criterion. A textual answer needs further examination.

4. **Requirements:** *Can the elements of the RA be mapped to the requirements?* The objective of this question is to evaluate whether the elements of the RA can be mapped to the requirements it intends to fulfil.

5. **Design:** *Is the design of the RA comprehensible?* The objective of this question is to evaluate whether the design the RA describes is comprehensible for the participants in the survey with their existing experience and knowledge in software architectures.

**Criterion Language Features:** For this criterion, one question is introduced in the questionnaire. Possible answers for these questions are yes, no and other. The participants are encouraged to make additional comments when choosing other. When participants answer yes to one of the questions this indicates that the RA fulfils the criterion. Accordingly, no, as answer will give evidence that the participants would consider that the RA does not fulfil the criterion. A textual answer needs further examination.

6. **Formalism:** *Is the RA described and formalised on an appropriate level?* The objective of this question is to gather the participants view whether the RA is sufficiently described and formalised to enable them to understand and use it.

**Criterion Model Features:** For this criterion, two questions are introduced in the questionnaire. Possible answers for these questions are rather good, rather bad and other. The participants are encouraged to make additional comments when choosing other. When participants answer yes to one of the questions this indicates that the RA fulfils the criterion. Accordingly, no, as answer will give evidence that the participants would consider that the RA does not fulfil the criterion. A textual answer needs further examination.

7. **Extensibility:** *How would you estimate extensibility of the RA?* The objective of this question is to gather the participants view on the extensibility of the RA.

8. **Flexibility:** *How would you evaluate the RA concerning simplicity, flexibility and the level of abstraction?* The objective of this question is to gather the participants view whether the RA is flexible enough to be used in different scenarios and whether the level of abstraction is appropriate to support this according to their expectations regarding software architectures.

**Summary:** In addition to these seven questions for the four engineering perspectives two summarising question are included. Possible answers for the first one are yes, no and other. The participants are encouraged to make additional comments when choosing other. The second question can be answered by free text. When participants answer yes to one of the questions this indicates that the RA fulfils the criterion. Accordingly, no, as answer will give evidence that the participants would consider that the RA does not fulfil the criterion. A textual answer needs further examination.

9. **Overall:** *Would you agree that the RA can support you in the development of applications for knowledge-based VA?* The objective of this question is to gain an overall summarised view of the participants on the suitability of the RA.

10. **Closure:** *Do you have any additional comments on the KnoVA RA or are there any elements missing in the architecture?* This open question aims to create a closure on the completeness of the RA. A larger number of answers for this question will possibly hint to insufficiencies of the RA.

Some of the questions above rely on the previous experience in the field of VA and in the fields of software development and software architecture of the participants. Therefore, a number of demographic questions are included to gain an overview of the experience of the participants in these fields. All demographic questions were marked as optional.

1. *What is your highest academic degree?* This is a multiple choice question with the following four options: B.Sc./B.A., Diplom / M. Sc., Doctoral Degree, other. When other is chosen, the participants are asked to specify the degree in a free text field, including none if they have no academic degree.
2. *In which subject did you obtain this degree?* This question provides a free text field for the answer.
3. *How many years of development experience do you have?* This question is answered by a field accepting a numerical value.
4. *Which software architecture patterns are you familiar with?* This question is answered by a free text field in which the participants could enter all architectures they know.
5. *Which design patterns are you familiar with?* This question is answered by a free text field in which the participants could enter all design patterns they know.
6. *How familiar are you with the following five concepts: UML modelling, Pseudo code, Formalisation, Design Patterns and Software Architecture?* For each of the five concepts the participants can enter numerical values from 1–10 where 1 stands for unfamiliar and 10 stands for expert level.

The questionnaire was given to two groups of engineers. The first group consists out of the engineers of the TOAD system. The second group consists out of software engineers who have no previous experience with the KnoVA RA. The survey was performed with these two groups for two reasons. The participation in the survey was voluntary; this was explicitly pointed out in the invitation to take part in the survey.

The first reason is to obtain a richer set of results due to the different background knowledge and experiences of the members in each group. This will possibly allow a comparison of the groups and allow a judgement about the transferability of the results. The second reason is to extend the number of participants which will possibly enhance the validity of the results.

The qualitative questions refer to the description of the RA. In preparation of the survey the participants therefore are given a document containing a detailed description of the KnoVA RA. This includes a description of the objective of the KnoVA RA as well as description of all four elements of the KnoVA as described in chapter 5 as well as a description of requirements that the RA intends to fulfil. These are based upon the research question and the challenges identified in this thesis.

For the TOAD engineers, who are already familiar with the KnoVA RA, this description served as a reference. For the control group of engineers this description served as the foundation to take part in the survey.

## Results

In total 16 participants took part in the survey, nine TOAD engineers and seven engineers in the second group. All participants answered the questions regarding the KnoVA RA. In the first group only eight participants answered the demographic questions, in the second group all seven participants answered the demographic questions.

Of the eight TOAD engineers who answered the demographic questions, three answered that B.Sc. / B.A. is their highest education and five answered that Diplom / M.Sc. is their highest education. All participants had studied either information systems or computer science. In the second group, all participants were educated at least to Diplom / M.Sc. level. One participant held a doctoral degree. All participants obtained their degree in computer science.

All participants in the survey had significant development experience. The participants in the TOAD study answered with experience ranging between 4–14 years with an average of seven years, median eight years. This was exceeded by the experience in the second group, which was ranging from 2–18 years with an average of over nine years and a median of ten years.

In summary, therefore it can be said that all participants are educated in a relevant subject and are experts in the field of software development. This assumes that they are capable to evaluate the KnoVA RA towards the engineering perspective and their judgement is profound and valuable.

As the KnoVA RA is a reference architecture for a specialised category of software systems, above the general engineering expertise the questionnaire included questions about the software architectures and software design pattern the engineers are familiar with. In total 14 different software architectures were named and 21 different design patterns. The most frequently named design pattern was the model-view-controller (MVC) pattern. It was named as design pattern for seven times in the first group and for four times in the second group. Three participants in the first group and one participant in the second group also named it as software architecture. It was closely followed by the similar model-view-viewmodel (MVVM) pattern that was familiar as design pattern to all eight participants in the first group, who answered the demographic questions and to two participants in the second group.

This is relevant as MVC and MVVM both represent comparable layered structural design patterns. In this they resemble the KnoVA RA that also recommends a layered structure. In fact an enriched version of MVVM can be used as a foundation to implement the KnoVA RA, as shown in chapter 6 for both application scenarios. It has to be noted though that a familiarity of the first group with the MVVM pattern is expected, as the TOAD system makes use of this pattern in conjunction with the KnoVA RA.

The most frequently named software architecture was the client-server architecture (four in the first group and eight in the second group), followed by n-Tier (three in the first

group and six in the second group). This is also relevant, as n-Tier as architectural style can be seen as an abstract version of structural patterns such as MVVM and MVC. From the large number of different design patterns and architectural styles the engineers are familiar with and based upon the fact that especially the comparable layered structural patterns MVVM and MVC and the related n-Tier style were named it can be judged that both groups are familiar with relevant architectural styles and design patterns. Hence, it can be assumed that they are able to comprehend and evaluate the KnoVA RA. Lastly, the participants were asked to rate their experience with UML modelling, pseudo code and formalisation that are techniques used to describe the KnoVA RA. In addition, the participants were asked to rate their experience with design patterns and software architecture. The objective of this question was to gain a concluding impression about the expertise of the participants towards these concepts.

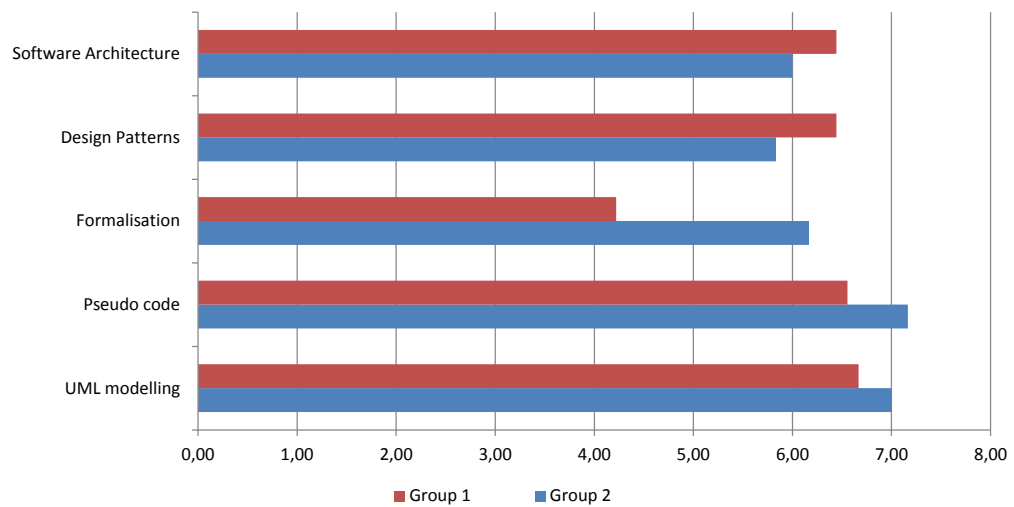


Figure 7.1: Expertise of the Participants in the Engineering Survey.

The main objective of the five demographic questions was to retrieve the experience of the participants in order to judge about their expertise. In figure 7.1 the average self-rating value is shown for the five techniques that were rated for the first group and for the second group. The average values range from 4.22 for formalisation (group one) to 7.17 for pseudo code (group two), with an average value over all questions and all groups of 6.25. Only formalisation in group one was rated significantly lower than average. In summary, it can be assumed that the participants exhibit the necessary expertise and experience in software development to be able to comprehend the KnoVA RA and to evaluate it in the qualitative survey.

In figure 7.2 the average results of the qualitative survey are shown. The figure is structured as follows. The answers are grouped into three categories: Criterion supported, criterion not supported and other.



A criterion in this account is one of the four criteria that Frank introduced to evaluate the engineering perspective (definition, explanation, language features and model features). Criterion supported in the figure refers to positive answers in respect of the evaluation of the KnoVA RA. The definition of positive answer here is defined above alongside the questions and depends on the possible answer to the question. The definition of criterion not supported accordingly relates to a negative answer, other was given as an option for participants who were not able to make a clear judgement.

On the vertical axis in this chart, the questions can be seen. On the horizontal axis, the average value for the three answer categories over all engineers in both groups is visualised. This numerical value is visualised by the length of the vertical bar, which indicates the percentage of participants who answered the question with either supported, not supported or other. In this chart, a positive answer (criterion supported) gives evidence that the participants judge this specific criterion to be supported by the KnoVA RA. Likewise, a negative answer gives evidence for the opposite. The answer categories are shaded according to the legendary below the chart.

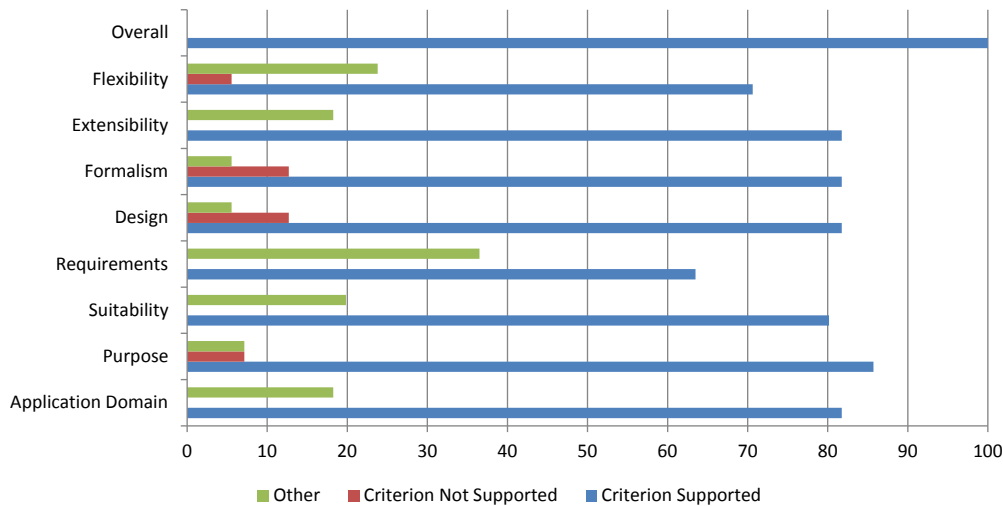


Figure 7.2: Average Results of the the Engineering Survey.

It can be seen that most question were given a positive answer. I fact all questions average at 80% positive answers or above. The only two exceptions are the requirements question, which averages at 63.5% and the flexibility question, which averages at 70.6%. In contrast to this, negative answers peaked with 12.7% for the questions formalism and design. Instead 36.5% answered with the category other for requirements and 23.8% for flexibility. To investigate this aspect figure 7.3 shows the results in more detail, separated by group.

In the figure, the shade of the vertical bars identifies the question. The mapping between the shade and the name of the question can be seen in the legendary below the chart.

In this figure, the vertical axis displays the answer category, as indicated by the textual label on the axis. For each answer category, a group of answers can be seen, related to the survey groups, also indicated by a textual label.

Here it can be seen that also in this detailed chart most questions were given a positive answer. Noteworthy is the response to the overall question: Would you agree that the RA can support you in the development of applications for knowledge-based VA? All participants gave a positive answer to this question. Negative answers are in general below 20%.

With the exceptions of the questions for flexibility and requirements in the first group all questions got more than 66% positive answers. It can also be seen that the question for requirements got 44.4% other answers in the first group and 28.6% in the second group. The question for flexibility here got 33.3% of answers in the category other. It can also be seen that the question for suitability got 28.6% of answers for the category other in the second group. Therefore, the questions for requirements, flexibility and suitability will be examined in more detail.

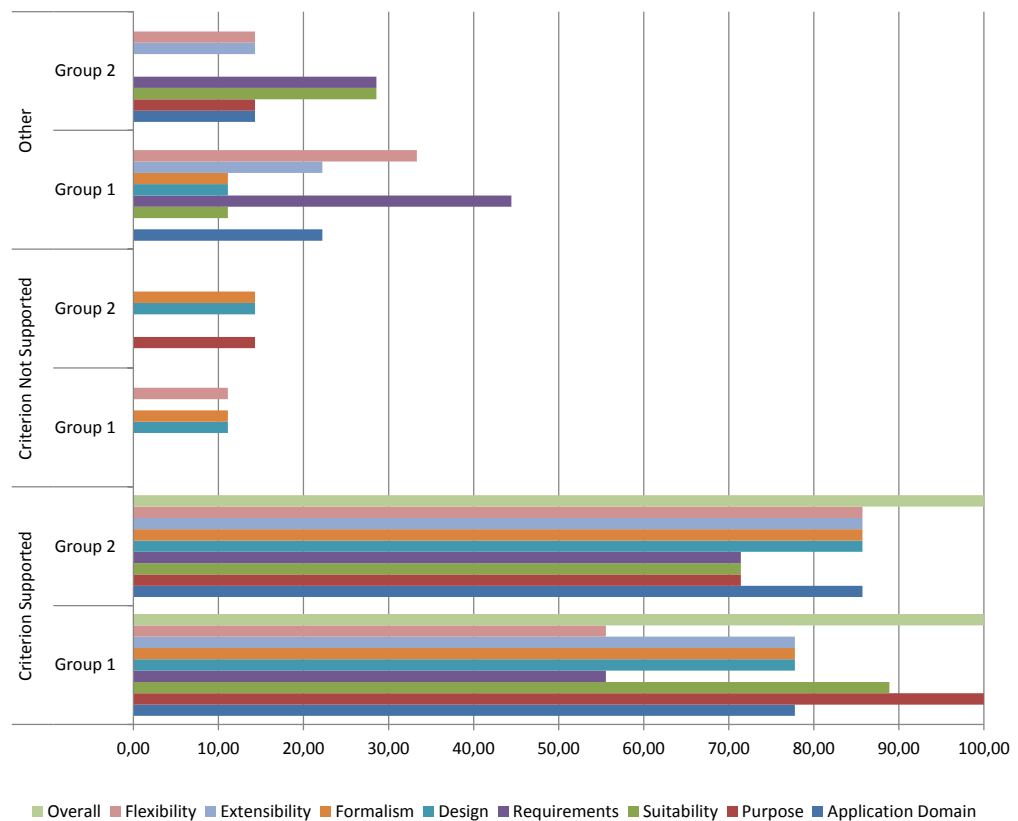


Figure 7.3: Results of the the Engineering Survey ordered by Group and Answer.

Concerning the requirements question, this reflects a total number of six engineers who

answered in the category other (four in the first group and two in the second group). From these six engineers five gave a textual feedback to the question (three in the first group and two in the second group). In the first group, one engineer asked for a tabular overview of the requirements in the short description. In the second group, both engineers who answered said that the requirements were not completely clear in the short description. This leads to the assumption that the short description is insufficient especially for engineers who are unfamiliar with the RA (the second group).

Other feedback in the first group was that concerning the requirement for collaborative VA. One engineer here noted that this requirement is not dealt with sufficiently, with the exception of knowledge transfer. Apart from this, the engineer said he would rate positive on the requirement side. Another engineer said that he could not identify the element of the RA that is suitable to be used to create VA systems supporting collaborative work. This statement is especially surprising, as the engineers of the first group have indeed used the RA to implement such a feature.

In summary, it is shown that the majority of engineers in both groups were able to comprehend the KnoVA RA and could map between the elements of the RA and the requirements.

For the flexibility question a total number of four engineers answered with other (three in the first group and two in the second group). All four provided detailed feedback concerning this question. In the first group, one engineer commented that he had to read the description of the concepts and algorithms for knowledge extraction, generalisation and application several times to comprehend them.

Another comment was that the usage of the term data binding, which the engineer considered to be biased towards C# as programming language. The term data binding is very commonly used for the synchronisation of data between different objects. However, there are many other fields in which data binding is also used with the same meaning, for instance for Java UI elements and Java objects. The third engineer commented that the simplicity is limited but apart from this, the RA fulfils the criteria. Lastly, the engineer from the second group who answered with other commented that to him it remained unclear how knowledge is represented and extracted. Therefore, this rating should be counted as negative.

In summary, the majority of engineers considered this criterion to be fulfilled, especially in the second group of engineers. The limitations and comments from the engineers showed that the flexibility of the KnoVA RA might result in a demand for training or at least result in an increased learning effort to comprehend the RA.

Concerning the suitability a total number of three engineers answered with other (one in the first group and two in the second group). The engineer in the first group asked in his comment to which application purpose the question refers and said if the application purpose is knowledge extraction and application then his answer is positive. The two engineers in the second group commented that they are lacking the expertise to answer the question.

The complaints that were made for some questions mostly rely on misinterpretation of either the description that was made for the survey or the questionnaire itself. Overall,

the results of the qualitative engineering survey are positive. All questions were answered with at least 63.5% positive answers. The demographic questions have shown that the engineers asked exhibit the necessary expertise and experience in software development to be able to comprehend the KnoVA RA and to evaluate it in the qualitative survey.

### 7.3.3 CARELIS Usability Study

This subsection presents the results of the evaluation of the user interface of CARELIS. To test the usability of CARELIS two industry standard surveys were pursued, the system usability scale (SUS) [Bro96] and the NASA Task Load Index (TLX) [HSS88]. The SUS aims to derive a measure for usability whilst the TLX focuses on stress level and cognitive and physical workload. The methodology and the results of these tests are summarised below.

#### System Usability Scale

The SUS aims to provide a numerical measure for the usability of a user interface. The SUS is based upon a questionnaire with ten questions. The questionnaire can be seen in appendix B.1. For the use in the cancer registry, the SUS questionnaire was translated to the German language because this was the native language for all participants.

In [Bro96] it is suggested to perform the same experiment on a number of participants in order to gain a number of scores. Each score is subjective depending on the participating individual. A number of scores allows reasoning about a general validity of the results.

The SUS questionnaire is structured in a way that the participants have to think about each question before they answer. Therefore, the ten questions of the questionnaire can be structured into two categories. Questions where a higher score represents a positive answer and questions where a higher score represents a negative answer. Details on the construction of the SUS can be read in [Bro96].

In the evaluation, five participants took part. They were asked independently and advised to not discuss the test result with their colleagues before the evaluation was completed. All participants are professional MRTs from the cancer registry who are trained to expert level in their domain specific task. Each participant filled out the questionnaire after using the system for a day working on their routine aggregation tasks with the additional work assignment to use the interface for rule definition and knowledge extraction.

The SUS consists out of ten questions that are answered on scales with five distinct values ranging from one to five. A value of one translates to the answer *strongly disagree* (with the question) and five translates to *strongly agree*. To calculate the SUS score the questions are separated into two groups. The group of positive influence factors (1, 3, 5 and 7) and the group of negative influence factors (2, 4, 6 and 8).

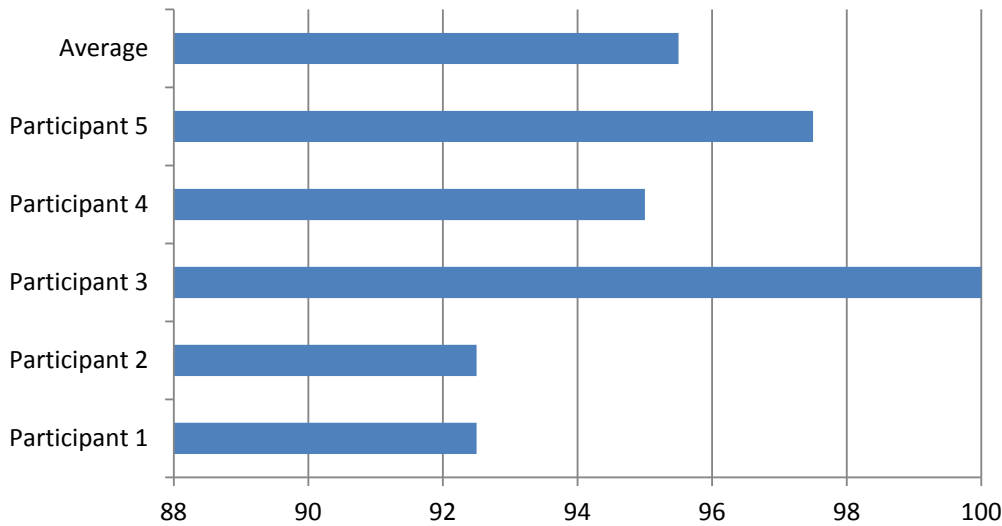


Figure 7.4: Individual Results of the SUS Questionnaire in the CARELIS Evaluation.

Then for all positive influence factors the score, determined on the questionnaire is subtracted by one. For instance, a score of five for the first question will result in a positive factor of four for this question. After this, the score for the negative influence factors is calculated by subtracting the score determined on the questionnaire from number five. For instance, a score of five on the questionnaire will result in an influence factor of  $5 - 5 = 0$ .

Finally, the sum of all ten influence-factors is built and multiplied by 2.5. The SUS therefore can range between 0 – 100. A higher SUS score indicates a high satisfaction of the users with the system. Hence, a high SUS score is a subjective indicator for a high usability of the system.

In figure 7.5 the overall SUS is presented for each user and as average value over all users. The individual results in figure 7.4 are the influence factors that are calculated according to the algorithm explained above. It can be seen that the answers to each question were very positive, resulting in values between between 3.0 and 4.0 for every single question, with an average between 3.4 for question nine and 4.0 for questions 2, 6, 7, 8 and 10. In the SUS four is the highest possible value for an influence factor.

This extremely positive values in the individual answers accordingly results in an exceptionally good overall SUS value (see figure 7.5). Outstanding is participant 3 who answered perfect for any question, resulting in an SUS score of 100 for this participant. The SUS for the other participants ranges between 92.5 and 97.5.

Although this exceptional SUS result at first assumes a positive usability, it has to be questioned why the results are this good. One possibility is that the users, who were trained to expert level with the tool for manual data aggregation were not exclusively valuated the newly integrated functionality but also valuated the familiar UI elements.

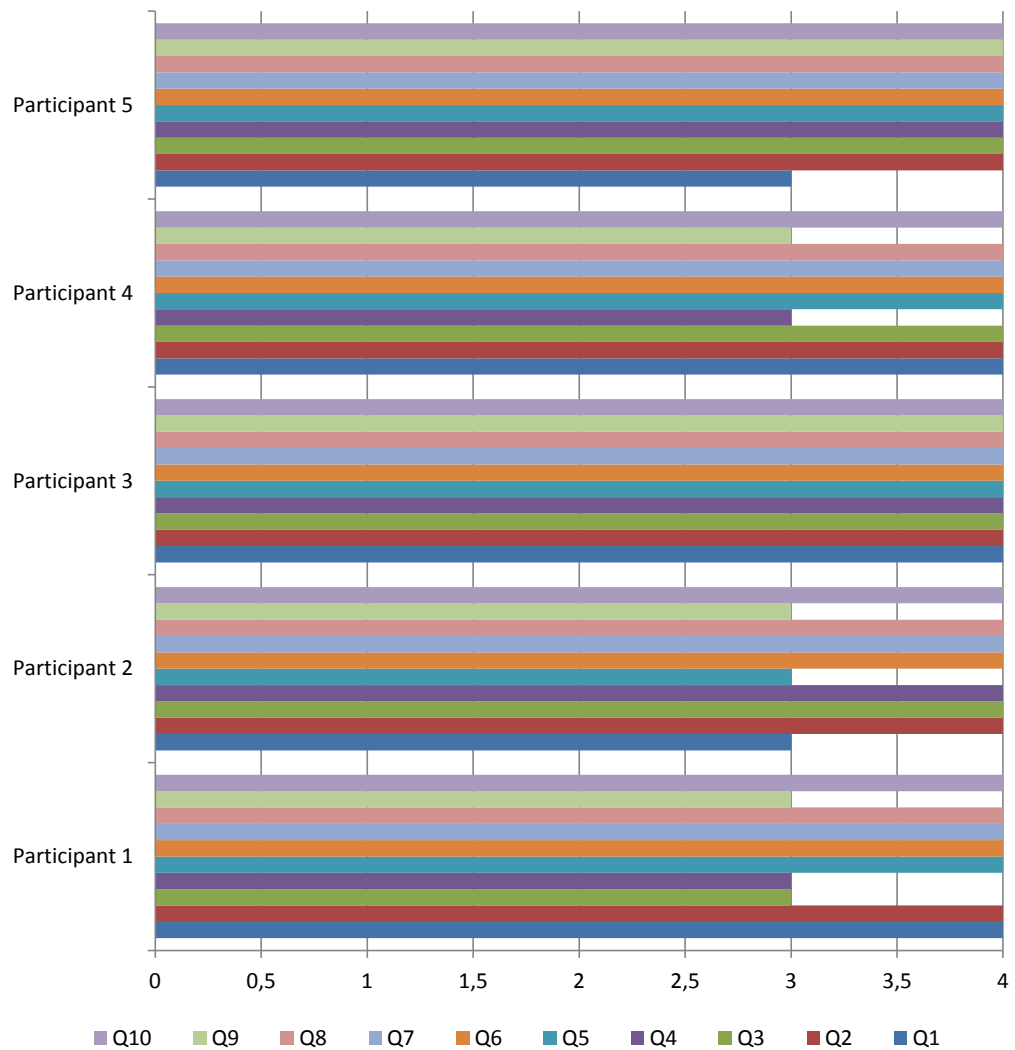


Figure 7.5: SUS Score in the CARELIS Usability Evaluation.

This however would not reflect the answer to question seven: *I would imagine that most people would learn to use this system very quickly*, because if the training level of the users had influenced the results then it should be expected that the users would answer to question seven in a less positive manner. Still the resulting factor for question seven was four for all users. Also surprising under the assumption that the preliminary training in the existing UI elements influenced the rating is the fact, that question nine: *I felt very confident using the system* received the lowest overall rating with an average factor of 3.4.

Combined it can be judged, that the positive SUS score can at least give a hint that the usability of the UI elements for knowledge derivation integrated into CARELIS is rated

positive by the users. The SUS score has to be read as a qualitative measure indicating a trend rather than an absolute measure describing a fact.

### NASA Task Load Index

The TLX aims to provide a numerical measure for the task load. It was developed by NASA in 1988 [HSS88].

Ten main factors for task load construct the TLX questionnaire: overall task load, complexity of the task, time pressure, performance, mental demand, physical demand, frustration level, stress level, fatigue and intuitiveness.

These factors have proven to produce a low variability in the results in test scenarios and thus producing representable and reproducible results. Details about the creation of the TLX can be found in [HSS88]. From the ten main factors that were identified in the design of the TLX six were forged to a questionnaire. The other factors were excluded because either they were too general (like the question for overall task load) or not general enough (stress level, fatigue and intuitiveness). The resulting questionnaire can be seen in figure B.2 in appendix B. The TLX questionnaire consists out six questions:

**Mental demand:** How mentally demanding was the task?

**Physical demand:** How physically demanding was the task?

**Temporal demand:** How hurried or rushed was the pace of the task?

**Performance:** How successful were you in accomplishing what you were asked to do?

**Effort:** How hard did you have to work to accomplish your level of performance?

**Frustration:** How insecure, discouraged, irritated, stressed, and annoyed were you?

Each question is answered with twenty distinct values ranging from very low to very high. In addition to this the TLX allows to rate the factors, thus resulting in an importance ranking of the factors against each other (see bottom half of figure B.2). In this ranking, the participants have to mark for all pairs of factors which of the two elements of a pair compared to each other were more important. In figure 7.6 the average result of the TLX questionnaire over all participants can be seen, grouped by the six questions. In figure 7.7 the results are presented for every participant individually.

In the TLX users can answer the questionnaire by crossing a field in the scale for each question. There are twenty fields on each scale, therefore the resulting numerical values range from 1–20. For most questions, a higher score represents a higher workload. This is illustrated in figure B.2 where for mental demand, physical demand, temporal demand, effort and frustration one as the lowest value is labelled *very low* and 20 as the highest value is labelled *very high*. Only for performance the lowest value indicates a *perfect* result and the highest value indicates a *failure*. This rating has to be considered when interpreting the results. The average mental demand was 7.0 and hence in the lower half of the scale. Participant one rated with ten and participant four rated with 15 indicating a higher mental demand. In the interviews, the participants said that understanding the newly introduced concepts was mentally demanding although they expect the mental demand will decrease after a trainings period.

A similar result occurred for temporal demand, where again participant one and participant four stand out with a slightly higher work load slightly higher than average. The

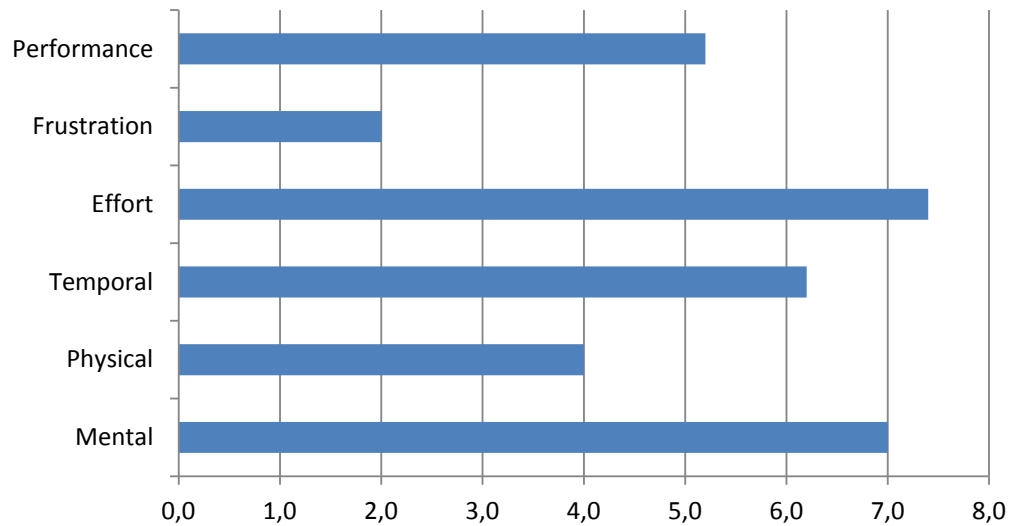


Figure 7.6: Average Results of the TLX in the CARELIS Usability Evaluation.

same pattern can also be seen for Effort, where these two participants declare a significantly higher effort than the other participants, which indicates that individual characteristics of the participants have a large influence on the results in this evaluation. A similar pattern also occurs in performance. Here a larger value indicates a worse performance and hence can be translated to a higher workload. Also not as present as for mental demand, physical demand, temporal demand and effort still the same pattern can be found in this chart.

The only exception from this pattern is frustration. The frustration level was two on average and even the highest frustration of four is well beyond the centre of the scale on the positive side. The low frustration with the UI elements backs up the good result in the SUS. In summary, in consideration of the fact that the participants have never worked with the new knowledge-based features before and that the underlying concepts were completely new and deviant from the well-trained work process, the results of the TLX can be rated positive. Regardless of some outlining negative ratings overall the average score for each answer is well below the centre of the scale, with values between 2.0 for frustration and 7.4 for effort.

#### 7.3.4 CARELIS Field Study

To evaluate the economic perspective a field study was performed in which the KnoVA RA based CARELIS prototype, that was introduced in section 6.2. Frank suggests six different criteria to choose from when evaluating the economic perspective: costs for introduction, costs for integration, and costs for maintenance, efficiency, flexibility and protection of investment. The objective of the field study was to gain insight on efficiency



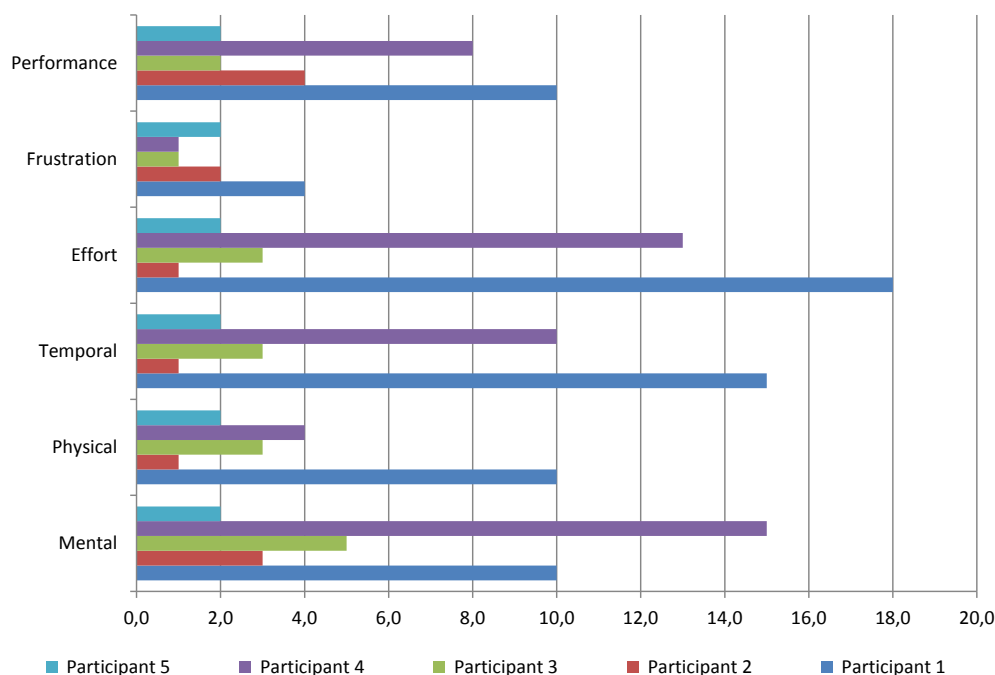


Figure 7.7: Detailed Results of the TLX in the CARELIS Usability Evaluation.

and flexibility of the provided solution by examining the performance of the prototype in a real world scenario.

## Methodology

Participants of the field study were five participants (all female). All participants were professional MRTs from the cancer registry who are trained to expert level in their domain specific task. For the field study, the CARELIS prototype was deployed into the existing working environment for the MRTs. For this, a dedicated workstation was set-up in the cancer registry. The reason why a dedicated workstation was used is that it would have been too risky and too costly to install the prototype on all five personal workstations of the participants.

The field study was succeeding the CARELIS usability study. Therefore, the MRTs already were trained in using the newly developed functionality and had an understanding of its objective. For the field study a separate instance of the registries database was used to insure that the production system is unaffected by the study. The example database was an exact copy of the registries database. To prepare the manual data aggregation task a random package of new records was chosen. This package contained 182 records to be aggregated.

The field study was performed with one user at a time, as only one workstation was

available. This results in a total number of five evaluation cycles, one for each MRT who took part. To be able to compare the results of different users the database was reset after each cycle. The information which participant produced which result set was discarded. However, the results were stored individually to be able to determine between the cycles. The objection of the field study was to evaluate if the systems is efficient. Efficient in this scenario means that it was to be shown that the relevant knowledge could be derived by the MRTs.

Therefore, deviant to the normal data management process as introduced in section 1.1.2, no automatic aggregation was performed. Instead, the newly imported records were directly used in the manual aggregation. This approach was chosen to prevent that the existing rules perform automatic aggregations, as these then would have been unavailable for knowledge extraction.

This is also compliant to the planned deployment of software. All existing rules will be abandoned, by request of the MD, to create a fresh set of new rules. The most important reason for this are changes to the database encodings and changes to the requirements towards the aggregation, which prevent the existing rules from being applicable.

## Results

For the field study, the MRTs were asked to perform the manual data aggregation on the example data set. The different steps in the aggregation process were recorded based upon the implementation of the KnoVA RA in CARELIS, as described in section 6.2. The MRTs were then asked to use the implemented features for knowledge extraction to define the rules that they applied in the aggregation task. The rules they extracted were then externalised into the database. The participants could abort the study at any time.

In each cycle, the MRTs were asked to perform the manual data aggregation until they finished aggregating all 182 records in the example data set. The duration and the outcome of this process varied according two factors. The different MRTs had slightly different criteria to aggregate the records. Therefore the rules derived in the five cycles differ. In addition, the time needed to complete the aggregation task varied based upon skills and work approach of the different MRTs. The longest time that the MRTs needed for the completion of the task was 90 minutes (the task was aborted at that point by the participant), the shortest time was 45 minutes.

In summary a number a total number of 76 rules were derived in the field study. A rule could be derived multiple times in separate cycles. The total number of distinct rules that have been derived was 45. In fact, because the MRTs were asked to express the knowledge for every aggregation they made, the same rule could be derived multiple times in a specific cycle.

The number of rules varied between the participants. The second participant (five rules) derived the lowest number of rules. The third participant (40 rules) derived the largest number of rules.

In table C a matrix between the 45 distinct rules and the participants is shown. The actual rules are not included in this table due to their complexity. The rules in total span over 22 different fields and can contain multiple nested conditions. The table does not show

null values. Due to the complexity of the rules, the table only shows numerical IDs for the rules. The numbers in the fields of this matrix indicate how often a rule was derived by one participant in the respective cycle, a – indicates that this rule was not derived by this participant. In the table the rules are numbered from R1 – R45.

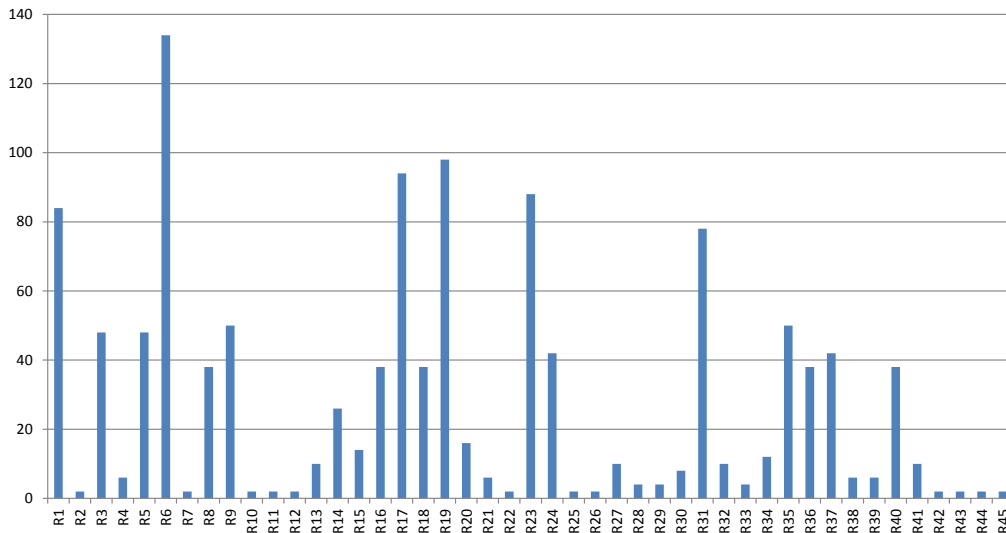


Figure 7.8: Number of Aggregations per Rule.

To evaluate the relevance of the rules the number of aggregations that a specific rule made on the test data set counted. Figure 7.8 illustrates the results. For each rule is shown how many of the 182 records in the test set the rule will aggregate. This was evaluated for each rule on a clean data set, where no previous aggregations have been made. The average number of records one of the rules created for the test data set would aggregate is 27.15. The maximum number of aggregations in the test data set is 134. In total twelve rules only aggregated two of the 182 records, which was the lowest number of aggregations per rule.

It can be seen that the derived rules are not independent from each other. When R6 would be executed before any other rule, then only 48 records would be left for aggregation. As there are rules that aggregating more than 48 records there must be an overlapping between the sets of rules that a particular rule aggregates. When looking at the rules in more details this becomes clear. The rule R6 is defined as follows:

*If for two records the fields Lastname AND DDRCode AND BWControlnumber AND PhoneticalName are equal, then aggregate the records.*

This is a relatively small rule with only four variables. In the derived rule set, there were several similar rules, which would extend R6 by additional attributes. For instance, R1 is defined as follows:

*If for two records the fields Lastname AND DDRCode AND BWControlnumber AND PhoneticalName AND FirstName are equal, then aggregate the records.*

Here it is shown that R6 actually aggregates a superset of R1. Therefore, when R6 executes before R1, no more aggregations can be made by R1. Comparable patterns occurred several times in the derived rule set. Some of the participants created rules that were very specific. For instance, R13, which was derived by the second participant, contained 14 different variables. In a personal discussion with this participant after the field study, it became clear that this participant not just created rules subsuming the variables that were discriminating in a specific situation but also for all other variables that were equal for a specific record. This approach will result in very specific rules, which may only affect a very limited number of records. A training of the MRTs towards which knowledge is sense fully be extracted therefore seems to be necessary.

## 7.4 Critical Reflection of the Results

This section critically reflects the results of the evaluation. The concept for the evaluation of the KonVA RA is based upon the four perspectives to evaluate a reference model that were introduced by Frank [Fra07]. In the concept for the evaluation criteria for the evaluation of the KnoVA RA were identified for each of the four perspectives (compare table 7.2). This section has four subsections, which correspond to these four perspectives. Additionally for each of these perspectives the criteria identified in 7.1.2 are revisited and the results are discussed.

### 7.4.1 The Economic Perspective

To evaluate the economic perspective Frank suggest choosing applicable criteria out of a set of six criteria: costs for introduction, costs for integration, and costs for maintenance, efficiency, flexibility and protection of investment. As outlined in the concept, the evaluation of the KnoVA focussed on the criteria efficiency and flexibility. After that, the protection of investment is examined.

#### Efficiency

Frank suggests examining the efficiency concerning the software development process or efficiency concerning implemented business processes. The efficiency towards the software development process cannot be measured directly, for the same arguments that were discussed in 7.2.1.

The efficiency towards the implemented business processes though can be evaluated based upon performance of the result artefacts. As indicated in 7.2.1 external constraints prohibited such an examination in the first application scenario. In the second application scenario, however it was possible to deploy the prototype in order to evaluate the economic perspective in a field study.

The results of this field study in 7.3.4 show that requirements towards knowledge extraction can efficiently be implemented based upon the KnoVA RA. In the field study it was measured how quickly the users can derive rules with the implemented features. All users were able to comprehend the implemented features for knowledge extraction.

Most participants were able to derive rules for all aggregations they made. Only one participant aborted the evaluation.

The derived rules therefore are used to make aggregations for the complete test data set. Although the test data set is of course not representative for all possible data sets, it provides the estimate that quickly a knowledge base will be derived that can significantly reduce the amount of records that have to be aggregated manually. This is backed up by a discussion with the MD who said that apart from changing requirements that cannot be foreseen, he is certain that they will be able to quickly define a set of rules that covers the majority of records and thus will significantly reduce the workload of the MRTs. Another indicator for the efficiency of the rule derivation are the exceptionally good results of the CARELIS user study, which was presented in 7.3.3 which indicates that the techniques are quickly comprehend able without much training effort.

The derivation of single rules that cover large parts of the test data set (such as R6) is another evidence for this. It was noted though that the rule derivation is time consuming and therefore will not be used continuously alongside the aggregation tasks. The MD rather said that either a specific MRT will be named to use the features or that the rule derivation will be used periodically when changes to the requirements are at foreseeable.

### Flexibility

To measure the flexibility Frank suggests reviewing the dependency from IT vendors and the openness of the reference model. The KnoVA RA represents a software architecture that is independent from its implementation in a software system. The fact that this architecture is intended to be the foundation of a broad range of VA systems and is derived based on a systematic scientific approach underline the flexibility of the RA. The implementation of the RA in two application scenarios that span over independent and differing application and problem domains further underlines this argument.

These arguments are also useful to examine the protection of investment. In case of the second application scenario the protection of investment can also be discussed in comparison to an implementation based upon other approaches. The main reason for the new development of the CARELIS system was a lack of flexibility of the business rule system and problems that arose out of the lack of vendor support for the existing business rule engine. With the implementation based upon the KnoVA RA the cancer registry has the complete control over the software and is therefore able to protect the investment.

#### 7.4.2 The Deployment Perspective

For the deployment, perspective Frank suggests to evaluate the ability and willingness of the users to deal with the model or respectively of the tools implemented based upon the model. The evaluation here is done by a discussion the criteria suggested by Frank in [Fra07]: understandability and appropriateness.

## Understandability

The understandability here is discussed based upon three approaches. Firstly, the understandability of the RA itself. This will be discussed in a critical reflection on this thesis and remarks on the engineering survey.

To comprehend the objective of the KnoVA RA in this thesis at first in chapter 1 a number of challenges are identified and motivated in two application scenarios. The KnoVA RA is designed an approach to these challenges. For this in chapter 3 requirements for the application scenarios are collected and implications for the design of the RA are derived. In chapter 5 these implications are then used as the foundation of the RA. The chapter develops the RA stepwise, structured along the design implications. For this the objective of the elements of the RA are thoroughly described and illustrated by examples. After this the application of the RA in the two scenarios is described in chapter 6. The reader therefore is strictly guided in the process to understand the RA and its application. The understandability of the RA was also evaluated in the engineering survey. The overall positive results of this survey were presented in subsection 7.3.2 indicate that the understandability of the RA is good, as even the engineers who were not previously involved in engineering projects that built upon the RA were able to comprehend and evaluate the RA.

The evaluation also showed some issues in the understandability. A small number of participants made comments specifically concerning the understandability. For instance, one engineer was unsure which requirements were meant. This however was probably an issue of the survey rather than an issue of the RA. However to gain a resilient result on the source for these problems further investigation is necessary.

## Appropriateness

To evaluate the appropriateness of the systems implemented based upon the RA user studies were performed, following the recommendations in [IWR<sup>+</sup>10]. In the toad user study, presented in subsection 7.3.1 the overall feedback was very positive. A number of positive remarks from the users, as, for instance, the remark that the system would provide clear benefits by bringing several parties together underlined this.

Along the lines of this another user study was made in the second application scenario. Here not only qualitative feedback was gathered but also quantitative measurements could be made towards the usability of the system, in the CARELIS usability study in 7.3.3 and towards the quality of the system in the CARELIS field study in 7.3.4.

The results of the usability study were exceptionally good. Above this the results of the field study showed, that the implemented system is appropriate to fulfil the requirements. In summary, therefore this can be seen as evidence that the KnoVA RA is suitable as the foundation of knowledge-based VA systems. However, as the KnoVA RA was derived based upon the two application scenarios, it is necessary to further investigate the appropriateness to show that the KnoVA RA can also be used in other application scenarios.

### 7.4.3 The Epistemology Perspective

[Fra07] suggests to examine four criteria for the evaluation of reference models in the epistemological perspective: the evaluation of theories, general principle of scientific research, critical reflection of human judgement and the reconstruction of the scientific process.

#### Evaluation of Theories

For each of these four criteria Frank suggests how to approach the evaluation. For the evaluation of theories, he suggests an examination of whether the elements of the reference model correspond to real world concepts. In section 1.1 real world scenarios were shown and based upon these scenarios challenges were derived. In chapter 3 requirements were gathered which then lead to the definition of design implications.

In chapter 5 concepts and methods were introduced following these design implications to face the challenges. This culminates in the definition of the KnoVA RA. The RA then was reviewed in a comparative evaluation against the state-of-the-art in section 5.8.

The closure between the introduced concepts and methods and the applicability in real world scenarios was then achieved by the realisation of two VA systems based upon the KnoVA RA, as described in chapter 6. By the implementation of two prototypes, the concepts and methods introduced in the KnoVA are mapped to real world artefacts. These artefacts are evaluated in this chapter. In summary, this shows the successful application of introduced concepts and methods across different application domains and for different problem domains.

#### General Principles of Scientific Research

The KnoVA RA is a generic architecture for various application scenarios. The KnoVA RA describes in an abstract way concepts and methods to implement knowledge-based VA systems. The level of abstraction of the KnoVA RA is high due to the formally described elements of the RA that are independent from a specific application domain. The elements of the RA were designed by general considerations and the examination of a number of VA systems in chapter 5 which are independent from the realisation in the two application scenarios that was presented in 6. The KnoVA RA therefore provides self-contained approaches to the problem definition and challenges introduced in chapter 1, to which no previous approaches were known. As far as existing approaches provide partial solutions to the challenges these were identified and evaluated and their limitations were shown as well as the approach how the KnoVA RA overcomes these limitations.

A transparent evaluation of the KnoVA RA can be made upon the perspectives examined in this chapter. The evaluation of the perspectives is underlined by a number of qualitative and quantitative evaluation approaches, following the principle of methods diversity, which are also described in this chapter.

## Critical Reflection of Human Judgement

Here Frank suggests examining whether an evaluation of the reference models can withstand a critical reflection based upon human judgement. The evaluation of the KnoVA RA based upon the criteria introduced in 7.1.1 for the evaluation of reference models according to [Fra07] allow for such a critical reflection.

The reflection is further supported by the introduction of the methodology for the evaluation in section 7.1.2 and by the examination of the results from various perspectives and across different applications scenarios. The introduced concepts and methods on the conceptual level are largely independent from the application domain as well as they are independent from specific technologies.

## Scientific Process

The scientific approach and the contribution of the introduced approach is pointed out in section 1.3. Here the goals of this research were named: The KnoVA RA aims to answer the research question that concepts and methods allow to represent and extract expert knowledge that was applied during the process of VA, to make it reusable.

This research question was derived from challenges that describe two real-world problems in different application scenarios. In section 5.8 it was shown how the KnoVA RA can be used to face these challenges and therefore provides an answer to the research question.

The evaluation of the introduced concepts and methods in the real world scenarios then has shown that the KnoVA RA is a domain independent conceptual approach to the challenges.

### 7.4.4 The Engineering Perspective

To evaluate the engineering perspective the KnoVA RA has been examined according to the four criteria introduced by Frank: definition, explanation, language features and model features in the engineering survey that was presented in subsection 7.3.2. These four criteria are addressed by the engineering survey. There the questions that were asked are grouped according to these four criteria. In addition to that, questions for an overall judgement and for a closure were included.

The overall positive result in the engineering survey in both groups that were asked indicates that the KnoVA RA fulfils the four criteria. The results of the engineering survey are discussed in detail in section 7.3.2. Apart from the problems identified here, it can be reasoned that definition, explanation, language features and model features of the KnoVA RA are sufficiently covered. Thus the questions introduced by Frank [Fra07], whether the model fulfils the requirements, and whether the specification is suited for supporting the intended purposes of the model, can answered positively.



## 7.5 Summary

In this chapter the evaluation of the concepts and methods introduced in chapter 5 was introduced.

For this intention at first in section 7.1 the methodological foundation for the evaluation was introduced. In subsection 7.1.1 four perspectives to be examined for the evaluation of reference models according to [Fra07] were named: the economic perspective, the deployment perspective, the epistemological perspective and the engineering perspective. In section 7.1.2 guidelines for the evaluation of VA systems according to [IWR<sup>+</sup>10] were discussed.

Based upon these foundations in section 7.2 a concept for the evaluation of the KnoVA RA was introduced which is based upon the principle of methods diversity as introduced in [Fra06]. After this in section 7.3 the evaluation is carried out by investigating the two VA systems implemented in the application scenarios complemented by an engineering survey where the KnoVA RA was evaluated by two groups of engineers.

In section 7.4 finally the results of the evaluation are examined according to the four perspectives, forming a closure to the methodological foundation of the evaluation.

In summary the evaluation showed that the KnoVA RA provides a suitable approach to the research question that was formulated in section 1.2.



## 8 Summary and Outlook

This chapter summarises the previous chapters and concludes with an outlook of remaining questions. Section 8.1 gives a summary of the objective of this thesis and of the gathered fundamentals, requirements and challenges. This section also takes upon the introduced approach to face the challenges and summarises the concepts and methods that this approach provides. After this this section summarises the the evaluation of the introduced RA.

Section 8.2 discusses remaining questions and gives an outlook on possible on further research.

### 8.1 Summary

In the introduction the demand for knowledge-based VA was motivated. For this initially the field of VA was introduced followed by two real world scenarios for analytical tasks from different application domains. The scenarios were chosen because they deal with divergent classes of analytical problems: collaborative analysis and knowledge sharing between analysts in the first scenario and enhancement of the analysis process and derivation of automated analysis steps in the second scenario. In each example three key factors can be identified identified: the application of expert knowledge by user interaction, the extraction of expert knowledge and the re-application of extracted knowledge. These key factors lead to the identification of four challenges:

**Challenge 1:** How can collaborative analysis be supported by VA systems? For example how can experts work at the same analysis questions and how can a system support knowledge sharing in the collaboration?

**Challenge 2:** How can implicit expert knowledge be extracted, in order to transfer results to other analysis tasks? For instance, how can findings be transferred from one trace-file to another to simplify the analysis of the other trace-file?

**Challenge 3:** How can a VA system support the representation of expert knowledge that is implicitly applied by the expert in the analysis process?

**Challenge 4:** How can this knowledge be extracted into a knowledge-base, in order to be re-applied the analysis process? For instance, how can this knowledge be used to support automatic aggregation steps?

In section 1.2 these challenges are the foundation for the definition of the research question which concepts and methods allow to represent and extract expert knowledge, that was applied during the process of VA, to make it reusable. In section 1.3 it was outlined how this research question was approached and how this approach contributes in to the field of VA. Finally in section 1.4 the subsequent chapters were introduced.

The objective of the second chapter was to sum up the necessary fundamentals within the scope of this work and evaluate their relevance. For this the domain of VA and the closely related domain of IV were introduced with a focus on the process of exploration.

Then three existing approaches to model the exploration process were introduced and compared. The data-flow model, where the process of visualisation is considered to be a pipeline of subsequent visualisation transformations, the data state model, which aims to be a conceptual model for all possible visualisation operations and the P-Set model which is based on the data state model and provided as model and framework for the visualisation of the exploration task. After this in section 2.3 Keims visual data exploration taxonomy, an existing approach to classify the domain of VA was introduced. Eventually in section 2.4 the introduced existing approaches were examined for similarities in and comparative evaluation. Here the features of the three visualisation models were compared and in addition to that similarities between the introduced process models and the visualisation models were identified and lined out in a qualitative comparison leading to a set of qualitative attributes which can be used to compare concepts and methods in the scope of the thesis.

In the third chapter the requirements for the two motivating application scenarios were analysed. The approach in this requirement analysis was to examine the requirements in the two application scenarios to then derive domain independent design implications for knowledge-based VA systems from these scenarios.

In section 3.3 the requirements from both scenarios were revisited and examined for similarities across the different application domains and across the diverted analytical challenges. This examination leads to four implications for the design of knowledge-based VA systems. These design implications were sorted into three groups: concerning the analysis process two design implications were identified, the implication for a support of the analysis process and the implication for analysis traceability. With respect to knowledge extraction and management the implication for the extraction of expert knowledge was identified. Lastly concerning knowledge application and management the design implication for knowledge application was identified.

Objective of the fourth chapter was the introduction of the KnoVA Taxonomy, based upon an examination of a set of existing VA systems. The taxonomy provides a classification for the design space of VA systems. It is based upon Keims VDET and largely extends this classification. It encompasses 38 classifying properties in the five classes data to be visualised, visualisation technique, interaction technique, exploration technique and dynamic representation. The KnoVA taxonomy is illustrated by using a novel iconographic language based upon the work of Aigner [ABM<sup>+</sup>07]. In this language every class is represented by a specific colour and every classifying property is represented by a specific icon.

The fifth chapter introduced the KnoVA RA, a software architecture for knowledge-based VA systems, which responds to the research question and is based upon the implications for design that were identified in the third chapter.

The structure of the fifth chapter, therefore, is aligned to the four implications for design. Firstly in section 5.3 the design implications were revisited and translated into requirements for a reference architecture for knowledge-based VA. This was followed by the introduction of a fundamental structural architecture for VA systems in section 5.2. This fundamental architecture was based upon the three-tier architectural pattern

translated into the domain of VA systems and provides a high level architectural view on VA systems.

After this in the four sections 5.4 – 5.7 the concepts and methods provided by the KnoVA RA were introduced. They correspond to the requirements for a reference architecture for knowledge-base VA identified in 5.3. The fundamental architecture was extended by these concepts and methods in each section to form the KnoVA RA.

At first in section 5.4 the KnoVA process model, a model for the analysis process was introduced. For this at first the problems that have to be solved when designing a generic model for the analysis process were described, based upon the general requirements identified for the corresponding design implication. Then the KnoVA process model was introduced. For this the elements of the analysis process were identified and a formal definition for these elements was introduced. This includes the definition of analysis parameters, analysis states, analysis steps and eventually the analysis process. Finally this definition was shown how the analysis process can be represented with UML classes. The section closes with a description how the fundamental architecture is enhanced by the KnoVA process model.

In the next section the KnoVA taxonomy was used as the foundation of the KnoVA meta model: a meta data model that can be used to describe the actual state of VA systems. Starting point for the creation of the KnoVA meta model was an example for the application of expert knowledge in VA systems. The KnoVA meta model was introduced in UML class diagram notation. On the introduced example the elements of the KnoVA meta model were described and it was shown how these elements can be used to describe the actual state of VA systems.

Finally the formal definition of analysis states was extended to the definition of reference model analysis states, which encompass the newly introduced meta model. The section concludes with an illustration how these concepts extend the the general architecture by introducing a new layer for the KnoVA meta model.

In section 5.6 the third design implications was approached. From this design implication the requirement to integrate a knowledge model that supports the extraction of expert knowledge was derived. To approach this requirements to subproblems were identified: a definition for knowledge to be extracted into a knowledge-base was required and concepts for the generalisation of knowledge were needed to abstract knowledge from the actual system state.

To approach these two problems at first a formal definition of a knowledge item was introduced, based upon the formal definition of reference model analysis states. This formal definition was then used to design structural elements for the KnoVA RA that can be used for knowledge extraction. These elements were complemented by an algorithm to extract knowledge into the knowledge-base, which uses the definition of knowledge items. It is triggered by state changing operations. After a state changing operation the two reference model analysis states, which represent the former state and the newly reached state in the analysis process were saved into a knowledge item. With this concept for knowledge representation and extraction a concept to approach the first subproblem was introduced.

The approach for the second subproblem, the generalisation of knowledge, was another algorithm which is based upon the definition of knowledge items. This algorithm reads all extracted knowledge items and generalises them from their meta model instances depending on implementation specific derivation rules. With this concept for generalisation of knowledge items a concept to approach the second subproblem in this section was introduced.

These concepts were completed by a discussion how multi step derivations can be handled, where the relevant knowledge was not applied between two succeeding steps but over a series of steps. Finally it was shown how the structural architecture is extended by the concepts for knowledge extraction and generalisation.

The last design implication that was identified in 3.3 involves knowledge application. In this design implication the requirement for concepts to re-apply extracted knowledge was derived. To approach this requirement was the objective of section 5.7. In this section at first two subproblems that arise when from this requirement were identified. First the knowledge that is applicable in a certain analysis state needs to be retrieved from the knowledge-base and second the knowledge needs to be applied in the analysis situation.

To address these subproblems the KnoVA RA is extended by two algorithms, one for knowledge retrieval and one for knowledge application. The algorithm for retrieval compares the state of the analysis system with the knowledge items in the knowledge-base and retrieves matching items.

The algorithm for application de-generalises retrieved knowledge items generates new reference model analysis states from this de-generalised knowledge item. Lastly it was shown how these concepts were integrated to complete the KnoVA RA.

In section 5.8 the KnoVA RA was validated. For this at first the research question and the challenges that lead to the research question in section 1.2 were revisited and it was pointed out how the concepts and methods introduced with the KnoVA RA can be used to face the challenges. After this discussion the KnoVA RA was examined in a comparison based upon the attributes for comparison identified in the comparative evaluation of the fundamentals in section 2.4. Here it was outlined how the KnoVA RA differs from existing approaches and extends the state of the art. Finally the last section 5.9 in this chapter summarises the chapter.

In chapter 6 two prototypical analysis tools were introduced which were implemented based on the KnoVA RA in .NET technologies. At first in section 6.1 the tool TOAD was introduced. TOAD is a tool for the analysis of in-car bus communication data and implements the requirements from the first application scenario. After a description of TOAD it was shown how the KnoVA RA was used to implement features concerning knowledge-based VA. In section 6.2 the tool CARELIS for the visual support in manual data aggregation tasks was described. The visual interface and the functionality of CARELIS was explained. CARELIS implements the requirements gathered for the second application. After this it was shown how novel features for knowledge extraction were realised based upon the KnoVA RA and how the RA integrates into the CARELIS implementation. After this the chapter was summarised in section 6.3.

In chapter 7 the KnoVA RA was evaluated. For this a methodological foundation for the evaluation was presented in section 7.1. The evaluation was based on the work of Frank towards the evaluation of reference models [IWR<sup>+</sup>10] and the work of Isenberg about evaluation in VA [KKM<sup>+</sup>10].

With the foundations in section 7.2 a concept for the evaluation was developed. Following the approach of methods diversity [Fra06] the concept encompasses the evaluation of the KnoVA RA in two scenarios. In these scenarios a number of different artefacts were evaluated in four perspectives.

The accomplishment of the evaluation and the results were then presented in section 7.3 and then critically reflected in section 7.4. In summary the evaluation showed that the KnoVA RA is suitable for its intended purpose and therefore provides an answer to the research question and the challenges identified in section 1.1.

## 8.2 Outlook

The KnoVA RA is a software architecture that provides concepts and methods that can be used to implement knowledge-based VA systems. The knowledge is applied in between two reference analysis model steps, which can be modelled following the KnoVA process model and the KnoVA meta model. The algorithm to extract this knowledge so far extracts knowledge items without weighting their importance. Certain steps in the analysis might be more important than other steps and hence it might not be efficient to extract all possible steps.

In the exemplary implementations this problem was solved by explicit incorporation of user interaction for knowledge extraction. Such an explicit user interaction may be unfortunate in some scenarios. Therefore further research towards automatic techniques for the evaluation of relevant knowledge is interesting. This research will have to provide an answer to the questions which metrics can be used to evaluate the importance of applied knowledge.

Another field of interesting future research is the knowledge generalisation. The algorithm for knowledge generalisation assumes that a set of application specific generalisation rules exists. It might also be possible to create metrics to evaluate which meta model elements in a specific reference analysis state can or should be generalised.

Another starting point for further research is the application of knowledge. In the TOAD scenario a user specifically selects which knowledge to apply. In the CARELIS system all applicable knowledge is applied when possible. There might be scenarios where other means for knowledge application are more feasible. For instance an automatic recommendation function could provide possible next steps in an integration interactive approach. Research in this area will have to find metrics to evaluate which solutions are valuable next steps from a list of all possible next steps.

In addition, a long term study towards the acceptance of the KnoVA RA would be interesting. For this it is necessary that the RA is used in a variety of VA systems and further qualitative surveys with the developers of the RA are needed.

Concerning the evaluation also a long term study of the economic perspective is of in-

terested. Such a study will have to find answer to the question how cost effective the KnoVA RA is. An initial idea for this was to gather information about the increase in quality or the decrease in the number of records that are manually aggregated in the CARELIS scenario. However due to structural changes in the composition of the records with the integration of the newly developed CARELIS such a comparative study was not possible. A very long term study in this context could show though how the initially derived rule set evolves over time. Certain measures, which have to be defined, could indicate the quality and the cost for the integration of the knowledge derivation based upon the KnoVA RA.

Another interesting field for future research would be to investigate the applicability of the KnoVA RA in other application domains. Here especially domains which deal with data that changes rapidly, such as trading data or streaming data from sensor networks are interesting. In these domains it will be interesting to investigate the behaviour of VA systems based upon the KnoVA RA towards measures such as scalability and performance.



# Appendix



## A Example Data Sets

Region	Absolute Incidence	Crude Rate	Age Normalised
BRAUNSCHWEIG KRFR ST	7427	292,60	240,91
GIFHORN LKR	3651	233,90	216,56
GOETTINGEN LKR	6573	248,80	220,24
GOSLAR LKR	5195	322,70	257,27
HELMSTEDT LKR	3160	312,50	260,57
NORTHEIM LKR	4489	294,10	241,26
OSTERODE AM HARZ LKR	2941	331,90	264,12
PEINE LKR	3303	263,30	224
SALZGITTER KRFR ST	3336	287,80	244,62
WOLFENBUETTEL LKR	3314	273,30	232,41
WOLFSBURG KRFR ST	3248	257,40	221,78
DIEPHOLZ LKR	5292	264,70	228,73
HAMELN-PYRMONT LKR	5148	317,40	254,86
HANNOVER KRFR ST	15806	304,10	251,18
HANNOVER LKR	15289	263,10	229,37
HILDESHEIM LKR	8350	287,70	240
HOLZMINDEN LKR	2456	297,30	240,21
NIENBURG (WESER) LKR	3278	269,50	230,36
SCHAUMBURG LKR	4653	290	237,88
CELLE LKR	5133	291,20	246,35
CUXHAVEN LKR	5870	297,70	248,89
HARBURG LKR	5253	247,20	218,52
LUECHOW-DANNENBERG LKR	1742	342,20	274,04
LUENEBURG LKR	3906	258,30	224,02
OSTERHOLZ LKR	2677	259,60	230,96
ROTENBURG (WUEMME) LKR	3841	256,30	225,55
SOLTAU-FALLINGBOSTEL LKR	3576	269,70	231,17
STADE LKR	4487	249,40	223,16
UELZEN LKR	2926	306,80	249
VERDEN LKR	3207	255,90	226,68
AMMERLAND LKR	2395	234	208,87
AURICH LKR	4716	265,10	236,41
CLOPPENBURG LKR	2849	211,60	204,88
DELMENHORST KRFR ST	2035	264,10	234,10
EMDEN KRFR ST	1542	299,60	256,50
EMSLAND LKR	6091	214,50	206,78
FRIESLAND LKR	2653	273,10	233,17
GRAFSCHAFT BENTHEIM LKR	3140	252,90	225,11
LEER LKR	4071	267,70	238,65
OLDENBURG (OLDENBURG) KRFR ST	3982	266,80	227,82
OLDENBURG (OLDENBURG) LKR	2477	223,50	202,35
OSNABRUECK KRFR ST	4643	279,30	236,54
OSNABRUECK LKR	8192	246,20	222,90
VECHTA LKR	2329	202,90	196,35
WESERMARSCH LKR	2593	279,60	240,66
WILHELMSHAVEN KRFR ST	2809	311,60	254,48
WITTMUND LKR	1406	256,90	225,59

Table A.1: Mustang Example Data Set.



## B Evaluation Questionnaires

### B.1 System Usability Scale (SUS)

#### *System Usability Scale*

© Digital Equipment Corporation, 1986.

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	1	2	3	4	5
2. I found the system unnecessarily complex	1	2	3	4	5
3. I thought the system was easy to use	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	1	2	3	4	5
5. I found the various functions in this system were well integrated	1	2	3	4	5
6. I thought there was too much inconsistency in this system	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	1	2	3	4	5
8. I found the system very cumbersome to use	1	2	3	4	5
9. I felt very confident using the system	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	1	2	3	4	5

Figure B.1: The System Usability Scale Questionnaire as presented in [Bro96].



## C Evaluation Results

	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5	Average
Question 1	4.0	3.0	4.0	4.0	3.0	3.6
Question 2	4.0	4.0	4.0	4.0	4.0	4.0
Question 3	3.0	4.0	4.0	4.0	4.0	3.8
Question 4	3.0	4.0	4.0	3.0	4.0	3.6
Question 5	4.0	3.0	4.0	4.0	4.0	3.8
Question 6	4.0	4.0	4.0	4.0	4.0	4.0
Question 7	4.0	4.0	4.0	4.0	4.0	4.0
Question 8	4.0	4.0	4.0	4.0	4.0	4.0
Question 9	3.0	3.0	4.0	3.0	4.0	3.4
Question 10	4.0	4.0	4.0	4.0	4.0	4.0
SUS	92.5	92.5	100.0	95.0	97.5	95.5

Enumeration of the questions is according to the questionnaire shown in figure B.1

*Table C.1: Results of the SUS Questionnaire in the CARELIS Scenario.*

	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5	Average
Mental	10.0	3.0	5.0	15.0	2.0	7.0
Physical	10.0	1.0	3.0	4.0	2.0	4.0
Temporal	15.0	1.0	3.0	10.0	2.0	6.2
Effort	18.0	1.0	3.0	13.0	2.0	7.4
Frustration	4.0	2.0	1.0	1.0	2.0	2.0
Performance	10.0	4.0	2.0	8.0	2.0	5.2

*Table C.2: Results of the NASA TLX Questionnaire in the CARELIS Scenario.*

Criterion	Question	Criterion Supported			Criterion Not Supported			Other		
		Group 1	Group 2	All	Group 1	Group 2	All	Group 1	Group 2	All
Definition	Application Domain Purpose Suitability	77.78	85.71	81.75	0.00	0.00	0.00	22.22	14.29	18.25
		100.00	71.43	85.71	0.00	14.29	7.14	0.00	14.29	7.14
		88.89	71.43	80.16	0.00	0.00	0.00	11.11	28.57	19.84
Explanation	Requirements	55.56	71.43	63.49	0.00	0.00	0.00	44.44	28.57	36.51
	Design	77.78	85.71	81.75	11.11	14.29	12.70	11.11	0.00	5.56
Language Features	Formalism	77.78	85.71	81.75	11.11	14.29	12.70	11.11	0.00	5.56
Model Features	Extensibility	77.78	85.71	81.75	0.00	0.00	0.00	22.22	14.29	18.25
	Flexibility	55.56	85.71	70.63	11.11	0.00	5.56	33.33	14.29	23.81
Overall		100.00	100.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00

Table C.4: Results of the Engineering Survey.



Rule	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5
R1	3		1		
R2	1		1		
R3	6		1		
R4	1				
R5	1				
R6	3			1	
R7	1				
R8	1				
R9	1				
R10	1				
R11	1				
R12		2			
R13		1			
R14		2	1		
R15			1		
R16			1		
R17			1	4	
R18			1		
R19			1		
R20			3		
R21			1		
R22			1		
R23			2		
R24			2		
R25			4	1	
R26			1		
R27			1		
R28			1		
R29			1		
R30			1		
R31			3		
R32			1		
R33			1		
R34			2		
R35			1		
R36			1		
R37			2		
R38			1		
R39			1		
R40				1	
R41				1	
R42					1
R43					1
R44					1
R45					1

Table C.3: Matrix of Rules (ID) and their Fire Frequency in the CARELIS Field Study.



## Glossary

The following glossary summarizes in alphabetical order the essential definitions of this thesis. The symbol  $\sim$  is used in the glossary as place holder for the term that is explained in the respective entry. A vertical  $\uparrow$  in front of an term references another glossary entry.

**Application Domain:** The  $\sim$  of a software system describes its area of usage. Exemplary application domains are the automotive sector and the health care sector. The  $\sim$  influences the requirements for the software system.

**Brushing:**  $\sim$  describes the  $\uparrow$  information visualisation technique to highlight selected values on  $\uparrow$  user interaction.

**Business Intelligence:**  $\sim$  is a category of applications and techniques for the collection, persistence and analysis of data with to support decisions in business processes.

**CAN:**  $\sim$  is a widespread bus-system in the automotive  $\uparrow$  domain. It is used for communication between  $\uparrow$  electronic control units in vehicles.

**CANalyzer:**  $\sim$  is a tool to derive  $\uparrow$  aggregated data from can bus protocols.

**Cancer Registry Legislation:** The  $\sim$  (KRG) is a German law to provide the fundamentals for cancer registries that was introduced in 1994 [Bun94].

**Choropletic Map:**  $\sim$  is a visualisation technique for  $\uparrow$  data with a geographic mapping where spatial or artificial regions on the map are colourised according to the characteristics of a measure. They are used to show the geographic distribution of values.

**Classification:**  $\sim$  describes the process of deriving distinctive classes from a larger set of unordered values.

**Cluster:**  $\sim$  describes a set of elements which share a common property. Clusters of values in a Cartesian coordinate plot for example can be described as set of values within a geometric threshold.

**Clustering:**  $\sim$  is the task of identification of  $\uparrow$  clusters in a data-set.

**Collaboration:**  $\sim$  describes a problem solving technique where multiple persons precede a common strategy.

**Continuous Animation:** A  $\sim$  describes an animation with smooth transitions between frames in opposition to a  $\uparrow$  discrete animation.

**Cube:** See  $\uparrow$  data-cube.

**Data-Cube:** A  $\sim$  is a data structure for multi-dimensional data. The  $\uparrow$  OLAP data model is based on data-cubes.

- Data order:**  $\sim$  is a subclass of the class  $\uparrow$  data to be visualised of the  $\uparrow$  KnoVA classification. It describes order relations of data (for instance ordinal or nominal).
- Data structure:**  $\sim$  is a subclass of the class  $\uparrow$  data to be visualised of the KnoVA classification. It describes the inner structure of the data. Exemplary data structures are hierarchically structured or functional dependent.
- Data to be visualised:**  $\sim$  is a class of the  $\uparrow$  KnoVA classification. It subsumes qualitative descriptive properties of data.
- Data characteristic:**  $\sim$  is a subclass of the class  $\uparrow$  data to be visualised of the  $\uparrow$  KnoVA classification. It subsumes the descriptive properties describing the representation of the data. Exemplary data characteristics are one-dimensional, multi-dimensional or text and hypertext.
- Data-Flow Model:**  $\sim$  is a  $\uparrow$  visualisation transform model introduced by [HM90] where the  $\uparrow$  visualisation transform is seen as a pipeline where every step in the pipeline defines a transformation.
- Data-State Model:** The  $\sim$  introduced by Chi and Riedl [CR98] aims to be a conceptual model for all possible visualisation operations. It is a  $\uparrow$  visualisation transform model where the  $\uparrow$  visualisation transform is seen as a sequence of steps, where state changing operations trigger the transition to another state.
- Dense-Pixel:**  $\sim$  describes the visualisation techniques where every pixel of a display represents a data point, thus creating the densest possible representation. Often in  $\sim$  the data points can be grouped and sorted to create visual patterns.
- Details On Demand:**  $\sim$  is a step in the  $\uparrow$  information seeking mantra. Details on demand describes the approach to hide details in early analysis steps and only show details after user interaction.
- Diffused Illumination:**  $\sim$  (DI) is a technique for surface computers with optical tracking. In a  $\sim$  setup, the interactive surface and a small area above the surface is illuminated by diffused (infra-red) light, which allows a detection of blobs and proximity detection for the interactive surface.
- Domain:** See  $\uparrow$  Application Domain.
- Electronic Control Unit:** An  $\sim$  (ECU) is an embedded system used in the automotive domain with defined functionality. ECUs control functions of a car. They have inputs of sensors and in-car user interface elements such as the handles of electric window lifts and evaluate the action that should be performed whenever an interface element is operated. ECUs communicate with each other via  $\uparrow$  in-car-bus communication.
- Epidemiological Cancer Registry Lower Saxony:** The  $\sim$  (EKN) collects records of cancer diseases from various sources such as hospitals, cancer centres, physicians and pathologies.

**Exploration:** The task of accessing unknown terrain with the purpose to discover new circumstances or information. In the context of this thesis exploration is used in connection with data analysis and describes the task of discovering new information in large data sets.

**Explorative Analysis:**  $\sim$  is the task of data analysis by means of interactive visual interfaces with the goal of hypothesis generation.

**Fact:** See  $\uparrow$  measure.

**Frustrated Total Internal Reflection:** The  $\sim$  (FTIR) is a technique for surface computers with optical tracking. In this technique (infra red) light is fed into a surface which consists out of a material with the property of  $\uparrow$  total internal reflection (e.g. a sheet of acrylic). The  $\sim$  uses the physical phenomenon that  $\uparrow$  total internal reflection can be distracted by changes in the optical density of a material. For acrylic sheets for example the total internal reflection gets distracted on areas where a finger (or human skin) touches the acrylic. These areas will appear as bright spots in the optical tracking of the  $\uparrow$  surface computer.

**Geo-Spatial Visualisation:** A  $\sim$  is a  $\uparrow$  visualisation dedicated for spatial geographic mappings. An example are  $\uparrow$  thematic maps.

**Gesture-based Interface:** A  $\sim$  is a  $\uparrow$  human machine interface which provides input facilities triggered by user gestures.

**Human Machine Interface:** The  $\sim$  of a technical system subsumes the system elements which allow human bi-directional interaction with the system. For example in a car the steering wheel is part of the human machine interface.  $\sim$  can be split into two groups: interfaces where the input devices for the machine are separated from the output (for example on desktop computers the screen is the visual output devices while the input devices are keyboard and mouse) and interfaces with a combined input and output devices, such as touch operated smart phones.

**Iconographic Language:** In the context of this thesis, and iconographic language is a language where icons serve as the syntax of the language, each icon representing a word in the language.

**In-Car Bus Communication:**  $\sim$  describes communication between  $\uparrow$  electronic control units in the auto

**Information Overload:**  $\sim$  describes the problem that data analysis cannot keep up with the growth in data. The  $\sim$  is caused by the growth in data storage capabilities, which exceed the growth in computing power.

**Information Seeking Mantra:** Overview first, zoom and filter, then details on demand. Mantra introduced by Shneiderman [Shn96] to describe the process of information visualisation and exploration.

**Information Visualisation:** Is the scientific discipline of visual representation of abstract data such as numerical values, lines of code in software systems, social networks with the intention to explore the data and to find unknown phenomena and relationships.

**Iterative Development:** Describes the approach of designing a system in small steps with repetitive user feedback. This approach is often used in engineering tasks where the requirements are not clearly identifiable at the beginning of the development.

**KnoVA classification:** Descriptive collection of classifying properties describing the design space of ↑ visual analytics applications.

**Krebsregistergesetz:** See ↑ Cancer Registry Legislation

**Linking:** ~ is a term from the domains of ↑ information visualisation and ↑ visual analytics. It refers to the technique of connecting two views which represent the same data or comparable data areas. Due to ~ a change in one view will result in a change in the other view. For instance a first view could present list of countries whilst a second view contains some figures aggregated according to country. Selecting or deselecting a country in the first view in this case could cause the respective values to appear or to disappear in the second view.

**Measure:** A ~ is a numerical value, representing a real world relationship, on which aggregates (sum, count, average, minimum, maximum) can be computed.

**Multi-dimensional Data Model:** See ↑ OLAP data model.

**Model-View-Viewmodel:** ~ (MVVM) is a ↑ design pattern used in software engineering. It is used to ensure an architectural division between the model, where data objects are held and the view which is loosely bound against an independent ↑ viewmodel. Special focus lies on a separation of the UI development from the application development. The ↑ viewmodel is intended to provide all display and business logic, whilst the view should be a mere passive consumer of the ↑ viewmodel functionality.

**Multitouch:** ~ -Devices are touch based input devices for computer systems which allow for input of more than one touch event at a time.

**OLAP Data Model:** The ~ is a specialised data model that is used in data warehousing to enable ad-hoc execution of complex analytical queries [CCS93].

**Rear Diffused Illumination:** ~ (Rear-DI) is a technique for ↑ surface computers with optical tracking. In a ~ setup the surface to be tracked gets enlighten by a light source from the back of the surface. Reflections of the light source are tracked by the camera. Changes in the reflection are evaluated as interaction.

**Reference Model:** A ~ is according to [Fra07] a descriptive and prescriptive model of an application or problem domain that is suitable to describe phenomena of the real world and to serve as a blue print to create tools in its domain.

---

**Relational Model:** The  $\sim$  is a widespread data model used in relational database systems. Data is organised in functional depended relations, each containing a set of numerical, textual or binary attributes.

**Surface Computer:** A  $\sim$  is a computer system which is operated by a large touch screen surface. Examples for surface computers are interactive walls and interactive tables. is a prominent example of a surface computer. They typically provide  $\uparrow$  multitouch. They represent  $\uparrow$  human computer interfaces where input and output devices are combined allowing for  $\uparrow$  direct manipulation.

**System-Usability-Scale:** The  $\sim$  (SUS) is a simple questionnaire used to evaluate the  $\uparrow$  usability of  $\uparrow$  human machine interfaces with special focus on computer based interfaces. The  $\sim$  was developed in 1986 by John Brooke at the Digital Equipment Corporation

**Thematic Map:** A  $\sim$  is a map enriched with additional information such as epidemiological figures spread over geographic areas.

**Think aloud protocol:** The  $\sim$  is a technique from cognitive science, often used in  $\uparrow$  usability engineering. The think aloud protocol demands a subject in a scientific study to express his thoughts and emotions verbally while preceding a given task. An observer will create a protocol of the expressed thoughts and emotions.

**Total Internal Reflection:**  $\sim$  is the physical phenomenon that electromagnetic radiation fed into certain material gets reflected by the inner bonds of the material and thus does not diffuse out of the material. In fibre optics for example light fed into the fibre will not diffuse out of the sides of the fibre.

**Usability engineering:**  $\sim$  is a scientific approach to maximize the usability of  $\uparrow$  human computer interfaces in a  $\uparrow$  user centred design process.

**Usability:**  $\sim$  is a measure for the complexity of use and learn ability of the interaction with real-worlds objects. In the scope of this thesis usability refers to the ease of use of information visualisation applications.

**User centered design process:**  $\sim$  (UCD) is an approach to (industrial) design where the needs and limitations of humans shape the outcome of the item to be designed.

**User Study:** A  $\sim$  is a scientific approach to falsify a hypothesis by involving a group of users. In  $\uparrow$  iterative development tasks it is often used to analyze the requirements for a technical system or to test properties of a system or prototype.

**ViewModel:** A  $\sim$  is a part of the  $\uparrow$  model-view-viewmodel design pattern. The  $\sim$  implements the business logic and provides an interface to the underlying model, to be consumed by the view.

**Visual Analytics Mantra:** Analyse first, show the important, zoom, filter and analyse further, details on demand. Mantra defined in [Sch07] to describe the ↑ visual analytics process, introduced by [KMS<sup>+</sup>08] according to the ↑ information seeking mantra.

**Visual Analytics:** ~ is a scientific discipline derived out of ↑ information visualisation. In contrast to this visual analytics focuses on analytical reasoning facilitated by interactive visual interfaces. Hence in ~ the visual representation constantly changes according to user interaction.

**Visualisation Technique:** A ~ is a specialised visual representation which displays the underlying data in a certain way. An example for a visualisation technique is a pie chart, where underlying numerical data is displayed relative to all other values in distinguishable areas, each representing a certain aspect of the data.

**Visualisation Transform:** A ~ is a function to transform data from one representation into another in order to be visualised. A ~ is necessary if the original data-format differs from the format of the ↑ visualisation.

**Visualisation:** In the context of this thesis ~ describes the task to visualise data by creating visual representations (images, diagrams, maps or animations) as a necessary step in ↑ information visualisation.

**Visualisation-Transform-Model:** Describes a process model for ↑ visualisation transformation. Examples are the ↑ data-flow model and the ↑ data-state model.

**Windows Presentation Foundation:** The ~ (WPF) is the foundation of graphical user interfaces in the ↑ Microsoft .NET Framework. The ~ provides a declarative language (XAML) to define visual interfaces. WPF applications typically implement the ↑ model-view-viewmodel design pattern.



## Abbreviations

CAN	Controller Area Network
CSCW	Computer Supported Cooperative Work
CSV	Comma separated values
CT/CAT	Computer tomography/Computer axial tomography
DQM	Data Quality Management
DSM	Domain Specific Model
DWH	Data Warehouse
ECU	Electronic Control Unit
EKN	Epidemiological Cancer Registry Lower Saxony (Epidemiologisches Krebsregister Niedersachsen)
FSM	Finite State Machine
GIS	Geographic information system
ICN	In-Car Communication Network
IV	Information Visualisation
KDD	Knowledge Discovery in Databases
KnoVA MM	KnoVA Meta Model
KnoVA PM	KnoVA Process Model
KnoVA	Knowledge-Based Visual Analytics
KRG	Cancer Registry Legislation (Krebsregistergesetz)
LIN	Local Interconnect Network
MD	Medical Director
MDSD	Model Driven Software Development
MOST	Media Oriented Systems Transport
MRI	Magnetic resonance imaging
MRT	Medical Records Technician
MSC-View	Message-Sequence Chart View
Mustang	Multi-dimensional statistical data analysis engine
MVVM	Model-View-Viewmodel
OLAP	Online Analytical Processing
P-Set	Parameter Set
RA	Reference Architecture
RST	Registry Party (Registerstelle)
SciVis	Scientific Visualisation
SOL-View	State Machine Overview List View
STL-View	State-Transition List View
SUS	System Usability Scale
TLX	Task Load Index (short for NASA Task Load Index)
TOAD	Touch and Decide

---

UCD	User centred design process
UI	User interface
UML	Unified Modelling Language
UTM	Universal Transverse Mercator
VA	Visual Analytics
VAST	Visual Analytics Science and Technology
VAT	Visual Analytics Transformation (System)
VDET	Visual Data Exploration Taxonomy
VizQL	Visual Query Language
VMTS	Visual Modelling and Transformation System
VST	Trusted Party (Vertrauensstelle)
WPF	Windows Presentation Foundation
XAML	Extensible Application Markup Language
XML	Extensible Markup Language

## Symbols

### THE VISUAL ANALYTICS PROCESS

---

$F$	VA process.
$S$	State in $F$ representing data sources.
$V$	State in $F$ representing visualisations.
$H$	State in $F$ representing hypothesis.
$I$	State in $F$ representing insight.
$f \in \{D_W, V_X, H_Y, U_Z\}$	Function defining transitions between the states in $F$ .
$D_W$	Functions representing data pre-processing with $W \in \{T, C, SL, I\}$ .
$D_T$	Functions for data transformations.
$D_C$	Functions for data cleaning.
$D_{SL}$	Functions for data selections.
$D_I$	Functions for data integration.
$V_X$	Functions representing visualisation with $X \in \{S, H\}$ .
$V_S$	Functions for visualising data.
$V_H$	Functions for visualising hypothesis.
$H_Y$	Functions representing the hypothesis generation process with $Y \in \{S, V\}$ .
$H_S$	Functions for hypothesis from data.
$H_V$	Functions for hypothesis from visualisations.
$U_Z$	Functions representing user interactions with $Z \in \{V, H, CV, CH\}$ .
$U_V$	Functions for user interaction effecting visualisations.
$U_H$	Functions for user interaction effecting hypothesis.
$U_{CV}$	Functions for user interaction leading to insight from visualisations.
$U_{CH}$	Functions for user interaction leading to insight from hypothesis.

---

### DATA-FLOW MODEL

---

$VTN = (V_T, E)$	Visualisation transformation network.
$V_T$	Set of visualisation stages.
$E$	Data-Flow Edge.
$v_i \in V_T$	Specific visualisation stage $v_i = (IN_i, OUT_i, g_i)$ .
$IN_i$	Set of inputs.
$OUT_i$	Set of outputs.
$out_k \in OUT_i$	Specific output.
$g_i : IN_i \rightarrow OUT_i$	Function mapping inputs to outputs.

---

## DATA-STATE MODEL

---

$DVP = (VD, ED)$	Data-state visualisation pipeline.
$VD$	Set of data-states.
$ED$	Set of operations.
$vd_i \in VD$	Specific piece of data.
$ed_i \in ED$	Visualisation operation with $ed_i = (vd_{from}, vd_{to}, h_i)$
$h_i : vd_{from} \rightarrow vd_{to}$	Function for a visualisation operation.

---

## P-SET MODEL

---

$VIS = (T, P, R, SR)$	Visualisation session.
$T$	Set of visualisation transformations.
$P$	Set of parameter sets.
$R$	Set of visualisation transform result types.
$SR$	Set of visualisation session results.
$D_i, i \in \{1 \dots n\}$	Specific data sets.
$P_k, k \in \{1 \dots m\}$	Specific visualisation transformation parameter types.
$t$	Visualisation transformation function $t : D_1 \times \dots \times D_n \times P_1 \times \dots \times P_m \rightarrow R$ .
$p_j$	Parameter set (p-set) with $p_j = \{p_j(1), \dots, p_j(o)\}$ and $p_j(l) \in P_k$ .
$s = (p, r, ts, d)$	Visualisation session result with $p \in P$ and $r \in R$ .
$ts$	Time stamp.
$d$	Specific relation of the parameter derivation calculus.

---

## KNOVA REFERENCE ARCHITECTURE

---

$KPM$	KnoVA Process Model with $KPM := \{\sigma_h \in \Sigma \mid \sigma_h < \sigma_{h+1} \forall h \in \mathbb{N}\}$ .
$\Sigma$	Set of analysis steps.
$\sigma := (as_k, \tau)$	Analysis step $\sigma \in \Sigma$ .
$AS$	Set of analysis states.
$\tau := \{as_{k+1}, \dots, as_{k+l}\}$	Successors of $as_k$ with $\tau \in AS^k$ with $k, l \in \mathbb{N}$ .
$as := \{p_1, \dots, p_m\}$	Analysis state $as \in AS$ with $\{p_1, \dots, p_m\} \in AP$ and $m \in \mathbb{N}^*$ .
$AP$	Set of analysis parameters.
$ap_i \in AP$	Tuples of data with values with $ap_i = (a_1, \dots, a_n)$ and $a_1, \dots, a_n \in A$ and $n \in \mathbb{N}^*$ .
$A \in DS$	Attributes of a data source $DS$ .
$DS$	A specific data source.
$\rho := \{as_i, m_{as_i}\}$	Reference analysis state with $as_i \in AS$ and $m_{as_i} \in M$ .
$\mathcal{RS}$	Set of all reference analysis states.
$m_{as_i} \in M$	Meta model instance valid for analysis state $as_i$ .
$M$	Set of meta model instances.

---

$\Theta$	Set of knowledge items.
$\vartheta_{12} := \{\rho_1, \rho_2\}$	Knowledge item with $\rho_1, \rho_2 \in \mathcal{RS}$ .
$mme \in m$	Element of a meta model instance $m \in M$ .
$MT$	Set of all meta model types.
$mmt \in MT$	meta model type of a meta model element $mme$ .
$\Delta$	Set of generalisation rules.
$\delta_i \in \Delta$	Specific generalisation rule.

---



## Figures

1.1	Schematic Overview over the Visual Interface of CARELIS. . . . .	5
1.2	Overview of the Structure of the Thesis. . . . .	8
2.1	The Information Visualisation Cycle. . . . .	12
2.2	The Visual Analytics Process. . . . .	14
2.3	The Visual Data Exploration Taxonomy. . . . .	19
4.1	Screenshot of the TaP System. . . . .	49
4.2	Exemplary Analysis Path to illustrate the VA Process. . . . .	52
4.3	Screenshot of the 3D-Cube System. . . . .	54
4.4	Screenshot of Mustang. . . . .	56
4.5	Screenshot of the POLARIS UI. . . . .	57
4.6	Two users working with Cambiera. . . . .	59
4.7	Screenshot of Cardiogram. . . . .	61
5.1	Fundamental Structural Architecture of VA Systems. . . . .	81
5.2	Abstract Representation the Analysis Path shown in Figure 4.2. . . . .	86
5.3	UML class diagram of the KnoVA Process Model. . . . .	87
5.4	KnoVA Reference Architecture including the KnoVA Process Model. . . . .	88
5.5	Example of Knowledge Application in Mustang. . . . .	90
5.6	UML class diagram of the KnoVA Meta Model. . . . .	93
5.7	Thematic Map Meta Model as Application of the KnoVA Meta Model. . . . .	97
5.8	Changes in the KnoVA Meta Model upon Knowledge Application. . . . .	100
5.9	KnoVA Reference Architecture including the KnoVA Meta Model. . . . .	101
5.10	UML class diagram of the classes for Knowledge Extraction and Application. . . . .	104
5.11	KnoVA Reference Architecture including Concepts for Knowledge Extraction. . . . .	107
5.12	The KnoVA Reference Architecture. . . . .	111
6.1	Screenshot of the TOAD System for the VA of ICNs. . . . .	118
6.2	Two Users in a VA session at the Multitouch Surface Computer. . . . .	119
6.3	Smart Filtering in the TOAD System. . . . .	123
6.4	Integration of the KnoVA RA into the TOAD System. . . . .	125
6.5	Elements of the Visual Interface of CARELIS. . . . .	129
6.6	Rule Derivation in the CARELIS. . . . .	131
6.7	Integration of the KnoVA RA into the CARELIS System. . . . .	133
7.1	Expertise of the Participants in the Engineering Survey. . . . .	152
7.2	Average Results of the the Engineering Survey. . . . .	153
7.3	Results of the the Engineering Survey ordered by Group and Answer. . . . .	154

---

7.4	Individual Results of the SUS Questionnaire in the CARELIS Evaluation.	157
7.5	SUS Score in the CARELIS Usability Evaluation. . . . .	158
7.6	Average Results of the TLX in the CARELIS Usability Evaluation. . . . .	160
7.7	Detailed Results of the TLX in the CARELIS Usability Evaluation. . . . .	161
7.8	Number of Aggregations per Rule. . . . .	163
B.1	The System Usability Scale Questionnaire. . . . .	181
B.2	The NASA Task Load Index Questionnaire. . . . .	182



## Tables

2.1	Qualitative Comparison of existing Process Models for Visualisation Exploration. . . . .	20
3.1	Categorisation of Requirements and Design Implications. . . . .	41
4.1	KnoVA Taxonomy, Class: Data To Be Visualised, Subclass: Data Characteristic. . . . .	65
4.2	KnoVA Taxonomy, Class: Data To Be Visualised, Subclass: Data Structure. . . . .	67
4.3	KnoVA Taxonomy, Class: Data To Be Visualised, Subclass: Data Order. . . . .	69
4.4	KnoVA Taxonomy, Class: Visualisation Technique. . . . .	70
4.5	KnoVA Taxonomy, Class: Interaction Technique. . . . .	71
4.6	KnoVA Taxonomy, Class: Exploration Technique. . . . .	73
4.7	KnoVA Taxonomy, Class: Dynamic Representation. . . . .	75
4.8	Overview of Properties of the KnoVA Taxonomy. . . . .	76
4.9	The Knowledge based Visual Analytics Taxonomy. . . . .	77
5.1	Relationship between Design Implications and Concepts. . . . .	80
5.2	Qualitative State-of-the-Art Comparison of the KnoVA RA. . . . .	114
7.1	Overview of the Evaluation of the KnoVA RA. . . . .	142
A.1	Mustang Example Data Set. . . . .	179
C.1	Results of the SUS Questionnaire in the CARELIS Scenario. . . . .	183
C.2	Results of the NASA TLX Questionnaire in the CARELIS Scenario. . . . .	183
C.4	Results of the Engineering Survey. . . . .	184
C.3	Matrix of Rules (ID) and their Fire Frequency in the CARELIS Field Study. . . . .	185



## References

- [ABM<sup>+</sup>07] AIGNER, W. ; BERTONE, A. ; MIKSCH, S. ; TOMINSKI, C. ; SCHUMANN, H.: Towards a conceptual framework for visual analytics of time and time-oriented data. In: *WSC '07: Proceedings of the 39th conference on Winter simulation*. Piscataway, NJ, USA : IEEE Press, 2007. – ISBN 1-4244-1306-0, S. 721–729
- [AMST96] APPELRATH, H.-J. ; MICHAELIS, J. ; SCHMIDTMANN, I. ; THOBEN, W.: Empfehlung an die Bundesländer zur technischen Umsetzung der Verfahrensweisen gemäß Gesetz über Krebsregister (KRG). In: *Informatik, Biometrie und Epidemiologie in Medizin und Biologie* 27 (1996), Nr. 2, S. 101–110
- [AN95] AAMODT, A. ; NYGÅRD, M.: Different Roles and Mutual Dependencies of Data, Information, and Knowledge - An AI Perspective on their Integration. In: *Data & Knowledge Engineering* 16 (1995), Nr. 3, S. 191–222
- [BBB<sup>+</sup>08] BIEHL, J. T. ; BAKER, W. T. ; BAILEY, B. P. ; TAN, D. S. ; INKPEN, K. M. ; CZERWINSKI, M.: IMPROMPTU: A New Interaction Framework for Supporting Collaboration in Multiple Display Environments and Its Field Evaluation for Co-located Software Development. In: *Proceedings of Conference on Human factors in computing systems (CHI)* ACM, 2008, S. 939–948
- [Ber83] BERTIN, J.: *Semiology of graphics*. University of Wisconsin Press, 1983. – ISBN 0299090604
- [BG07] BOSCH GMBH, Robert: *Kraftfahrzeugtechnisches Taschenbuch*. 26. Vieweg&Sohn Verlagsgesellschaft, 2007
- [Bou06] BOURBAKI, N.: *Théorie des ensembles*. Bd. 1: *Éléments de mathématique Première Partie*. Berlin : Springer Verlag, 2006. – ISBN 3540340343
- [Bro96] BROOKE, J.: A quick and dirty usability scale. In: JORDAN, P. (Hrsg.) ; JORDAN, W. (Hrsg.) ; MCCLELLAND, I. (Hrsg.): *Usability Evaluation in Industry*, Taylor and Francins, 1996. – ISBN 978-0199286706, S. 189–194
- [Bro06] BROY, M.: Challenges in automotive software engineering. In: *Proc. Software Engineering (ICSE)* ACM, 2006, S. 33–42
- [Bun94] BUNDESTAG, Deutscher: *Gesetz über Krebsregister (Krebsregistergesetz KRG)*. Drucksachen 12/6478, 12/7726, 12/8287, Bonn, 1994

- [Car03] CARLSON, C.N.: Information overload, retrieval strategies and Internet user empowerment. In: HADDON, Leslie (Hrsg.): *The Good, the Bad and the Irrelevant (COST 269)* Bd. 1, Media Lab UIAH, 2003, S. 169–173
- [Car08] CARPENDALE, S.: Evaluating Information Visualizations. In: *Information Visualization*. Berlin, Heidelberg : Springer-Verlag, 2008. – ISBN 978–3–540–70955–8, S. 19–45
- [CCS93] CODD, E.F. ; CODD, S.B. ; SALLEY, C.T.: *Providing OLAP to User-Analysts: An IT Mandate*. 1993
- [Chi00] CHI, E.: A Taxonomy of Visualization Techniques Using the Data State Reference Model. In: *INFOVIS '00: Proceedings of the IEEE Symposium on Information Visualization 2000*. Washington, DC, USA : IEEE Computer Society, 2000. – ISBN 0–7695–0804–9, S. 69–75
- [Chi02] CHI, E.: Expressiveness of the data flow and data state models in visualization systems. In: *AVI '02: Proceedings of the Working Conference on Advanced Visual Interfaces*. New York, NY, USA : ACM, 2002. – ISBN 1–58113–537–8, S. 375–378
- [CM97] CARD, S.K. ; MACKINLAY, J.: The Structure of the Information Visualization Design Space. In: *Proceedings of the Symposium on Information Visualization '97*, 1997, S. 92–99
- [CMS99] CARD, S. ; MACKINLAY, J. ; SHNEIDERMAN, B.: Using vision to think. In: *Readings in information visualization*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1999. – ISBN 1–55860–533–9, S. 579–581
- [CR98] CHI, E. ; RIEDL, J.: An Operator Interaction Framework for Visualization Systems. In: *INFOVIS '98: Proceedings of the 1998 IEEE Symposium on Information Visualization*. Washington, DC, USA : IEEE Computer Society, 1998. – ISBN 0–8186–9093–3, S. 63–70
- [EN09] ELMASRI, R.A. ; NAVATHE, S.B.: *Fundamentals of Database Systems*. 5th. New York, NY, USA : Pearson Studium, 2009 (Pearson Studium). – ISBN 321–36957–2
- [FA11] FLÖRING, S. ; APPELRATH, H.-J.: KNOVA: INTRODUCING A REFERENCE MODEL FOR KNOWLEDGE-BASED VISUAL ANALYTICS. In: GABRIELA CSURKA, José B. Martin Kraus K. Martin Kraus (Hrsg.) ; INSTICC (Veranst.): *Proceedings of the International Conference on Information Visualization Theory and Applications (IVAPP) 2011* Bd. 3/2011. Vilamoura, Portugal : SciTePress – Science and Technology Publications, 2011, S. 230–235

- [FH09] FLÖRING, S. ; HESSELMANN, T.: TAP: visual analytics on surface computers. In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. New York, NY, USA : ACM, 2009 (ITS '09). – ISBN 978-1-60558-733-2, S. 11:1–11:1
- [FH10] FLÖRING, S. ; HESSELMANN, T.: TaP: Towards Visual Analytics on Interactive Surfaces. In: ISENBERG, P. (Hrsg.) ; SEDLMAIR, M. (Hrsg.) ; BAUR, D. (Hrsg.) ; ISENBERG, T. (Hrsg.) ; BUTZ, A. (Hrsg.) ; LMU Media Informatics (Veranst.): *Collaborative Visualization on Interactive Surfaces - CoVIS '09*. München : LMU, 2010 (Technical Reports 2), S. 9–12
- [FH09] FLÖRING, S. ; HESSELMANN, T. ; TEIKEN, Y. ; APPELRATH, H.-J.: Kollaborative visuelle Analyse multidimensionaler Daten auf Surface-Computern. In: *Datenbank-Spektrum* 9 (2009), Dezember, Nr. 31, S. 17–25. – ISSN 1618–2162
- [Fow02] FOWLER, M.: *Patterns of Enterprise Application Architecture*. Addison-Wesley Longman Publishing Co.,Inc., 2002. – ISBN 3–211–2742–0
- [Fra06] FRANK, U.: Towards a Pluralistic Conception of Research Methods in Information Systems Research / University of Duisburg-Essen. 2006 (7). – ICB-Research Report. – ISSN 1860–2770. –
- [Fra07] FRANK, U.: *Evaluation of Reference Models*, University of Duisburg-Essen, Diss., 2007
- [Gar11] GAROFALO, R.: *Building Enterprise Applications with Windows Presentation Foundation and the Model View ViewModel Pattern*. Microsoft Press, 2011 (Microsoft Press Series). – ISBN 9780735650923
- [GEGC98] GERSHON, N. ; E. GERSHON, G. S. ; CARD, S.: Information Visualization. In: *Interactions* 5 (1998), March, S. 9–15. – ISSN 1072–5520
- [GGS97] GERSHON, N. ; GERSHON, E. ; STEPHEN, G.: Information Visualization. In: *IEEE Transactions on Visualization and Computer Graphics* 17 (1997), Nr. 4, S. 29–31. – ISSN 0272–1716
- [Han06] HANRAHAN, P.: VizQL: a language for query, analysis and visualization. In: *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*. New York, NY, USA : ACM, 2006 (SIGMOD '06). – ISBN 1–59593–434–0, S. 721–721
- [Hei05] HEINECKE, H.: Automotive system design-challenges and potential. In: *Proc. Design, Automation and Test (DATE) IEEE*, 2005, S. 656–657

- [HFS09] HESSELMANN, T. ; FLÖRING, S. ; SCHMITT, M.: Stacked Half-Pie menus: navigating nested menus on interactive tabletops. In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. New York, NY, USA : ACM, 2009 (ITS '09). – ISBN 978–1–60558–733–2, S. 173–180
- [HLC91] HABER, R.B. ; LUCAS, B. ; COLLINS, N.: A data model for scientific visualization with provisions for regular and irregular grids. In: *VIS '91: Proceedings of the 2nd conference on Visualization '91*. Los Alamitos, CA, USA : IEEE Computer Society Press, 1991. – ISBN 0–8186–2245–8, S. 298–305
- [HM90] HABER, R.B. ; MCNABB, D.A.: Visualization idioms: A conceptual model for scientific visualization systems. In: NIELSON, G.M. (Hrsg.) ; SHRIVER, B.D. (Hrsg.) ; ROSENBLUM, L. (Hrsg.): *Proceedings of IEEE Visualization, Visualization in Scientific Computing* Bd. 74, IEEE Computer Society Press, 1990, S. 74–93
- [HMW<sup>+</sup>03] HUTCHINSON, H. ; MACKAY, W. ; WESTERLUND, B. ; BEDERSON, B.B. ; DRUIN, A. ; PLAISANT, C. ; BEAUDOUIN-LAFON, M. ; CONVERSY, S. ; EVANS, H. ; HANSEN, H. u. a.: Technology probes: inspiring design for and with families. In: *Proceedings of Conference on Human factors in computing systems (CHI)* ACM, 2003. – ISBN 1581136307, S. 17–24
- [HSS88] HART, S. G. ; STAVENLAND, L. E. ; STAVENLAND, L. E.: Development of NASA-TLX (task load index): Results of empirical and theoretical research. In: *Human Mental Workload*, 1988, S. 139–183
- [IC07] ISENBERG, P. ; CARPENDALE, S.: Interactive Tree Comparison for Co-located Collaborative Information Visualization. In: *IEEE Transactions on Visualization and Computer Graphics* 13 (2007), Nr. 6, S. 1232–1239
- [IF09] ISENBERG, P. ; FISHER, D.: Collaborative Brushing and Linking for Co-located Visual Analytics of Document Collections. In: *Computer Graphics Forum Proceedings of EuroVis* 28 (2009), Nr. 3, S. 1031–1038. – ISSN 1467–8659
- [IF11] ISENBERG, P. ; FISHER, D.: Cambiera: Visual Analytics on the Surface. In: *Videos of the ACM Conference on Computer Supported Cooperative Work (CSCW)*. Hangzhou, China : ACM, 2011, S. 0
- [IFM<sup>+</sup>10] ISENBERG, P. ; FISHER, D. ; MORRIS, M.R. ; INKPEN, K. ; CZERWINSKI, M.: An exploratory study of co-located collaborative visual analytics around a tabletop display. In: *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on*, 2010, S. 179–186

- 
- [Ins09] INSELBERG, A.: *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer Science + Business Media, 2009. – ISBN 978-0-387-21507-5
- [IWR<sup>+</sup>10] ISENBERG, T. ; WIJK, J. ; ROERDING, J. ; TELEA, A. ; WESTENBERG, M.: Chapter 8 – Evaluation. In: KEIM, D.A. (Hrsg.) ; KOHLHAMMER, J. (Hrsg.) ; ELLIS, G. (Hrsg.) ; MANSMANN, F. (Hrsg.): *Mastering The Information Age - Solving Problems with Visual Analytics. Chapter 8 – Evaluation*. Eurographics, 2010. – ISBN 978-3-905673-77-7, S. 131–144
- [Jai99] JAIN, S.: Simulation in the next millennium. In: *WSC '99: Proceedings of the 31st conference on Winter simulation*. New York, NY, USA : ACM, 1999. – ISBN 0-7803-5780-9, S. 1478–1484
- [JK03] JANKUN-KELLY, T.J.: *Visualizing Visualization: A Model and Framework for Visualization Exploration*, Center for Image Processing and Integrated Computing, University of California, Davis, Diss., Juni 2003
- [JKMG07] JANKUN-KELLY, T.J. ; MA, K. ; GERTZ, M.: A Model and Framework for Visualization Exploration. In: *IEEE Transactions on Visualization and Computer Graphics* 13 (2007), Nr. 2, S. 357–369. – ISSN 1077-2626
- [KBT<sup>+</sup>08] KLANTEN, R. ; BOURQUIN, N. ; TISSOT, T. ; EHMANN, S. ; HEERDEN, F.v.: *Data Flow: Visualising Information in Graphic Design*. 1st. Berlin : Die Gestalten Verlag, 2008. – ISBN 978-3-89955-217-1
- [Kei01] KEIM, D.A.: Visual exploration of large data sets. In: *Commun. ACM* 44 (2001), Nr. 8, S. 38–44. – ISSN 0001-0782
- [Kei02a] KEIM, D. A.: Datenvisualisierung und Data Mining. In: *Datenbank-Spektrum* 2 (2002), S. 30–39
- [Kei02b] KEIM, D.A.: Information Visualization and Visual Data Mining. In: *IEEE Transactions on Visualization and Computer Graphics* 8 (2002), Nr. 1, S. 1–8. – ISSN 1077-2626
- [KKM<sup>+</sup>10] KEIM, D.A. ; KOHLHAMMER, J. ; MAY, T. ; WANNER, F. ; MANSMANN, F.: Visual Analytics. In: KEIM, D.A. (Hrsg.) ; KOHLHAMMER, J. (Hrsg.) ; ELLIS, G. (Hrsg.) ; MANSMANN, F. (Hrsg.): *Mastering The Information Age - Solving Problems with Visual Analytics*. Eurographics, 2010. – ISBN 978-3-905673-77-7, S. 7–18
- [KMS<sup>+</sup>08] KEIM, D.A. ; MANSMANN, F. ; SCHNEIDEWIND, J. ; THOMAS, J. ; ZIEGLER, H.: Visual Analytics: Scope and Challenges. In: SIMOFF, Simeon J. (Hrsg.) ; BÖHLEN, Michael H. (Hrsg.) ; MAZEIKA, Arturas

- (Hrsg.): *Visual Data Mining*. Berlin, Heidelberg : Springer-Verlag, 2008. – ISBN 978–3–540–71079–0, S. 76–90
- [KMSZ09] *Kapitel 10*. In: KEIM, D. A. ; MANSMANN, F. ; STOFFEL, A. ; ZIEGLER, H.: *Visual Analytics*. Springer, 2009
- [Kna01] KNAUER, U.: *Diskrete Strukturen, kurz gefasst*. Spektrum Akademischer Verlag, 2001. – ISBN 3827410215
- [Kra10] KRAMER, M.: *3D-Cube: Visualisierung und Selektionsmenü für multidimensionale Daten*. Oldenburg, Germany, Universität Oldenburg, Master Thesis, May 2010
- [KSS<sup>+</sup>06] KEIM, D.A. ; SCHELWIES, T. ; NIETZSCHMANN A. ; SCHNEIDEWIND, J. ; SCHRECK, T. ; ZIEGLER, H.: A Spectral Visualization System for Analyzing Financial Time Series Data. In: *EuroVis '06*, 2006, S. 195–202
- [KW02] *Kapitel 11*. In: KEIM, D.A. ; WARD, M.: *Visualization*. 2nd. Berlin Heidelberg : Springer Verlag, 2002
- [LLMC05] LEVENDOVSKY, T. ; LENGYEL, L. ; MEZEI, G. ; CHARAF, H.: A Systematic Approach to Metamodeling Environments and Model Transformation Systems in VMTS. In: *Electronic Notes in Theoretical Computer Science*, 2005, S. 65–75
- [Mac10] MACDONALD, M.: *Pro WPF in C# 2010: Windows Presentation Foundation in .NET 4*. Apress, 2010. – ISBN 978–1–4302–7205–2
- [McC09] MCCANDLESS, D.: *Information is Beautiful*. 1st. London, UK : Harper-Collins, 2009. – ISBN 978–0–00–729466–4
- [MMC09] MÉSZÁROS, T. ; MEZEI, G. ; CHARAF, H.: Engineering the Dynamic Behavior of Metamodeled Languages. In: *Simulation, Special Issue on Multi-Paradigm Modeling* 89 (2009), S. 793–810
- [MTWS08] MULLER-TOMFELDE, C. ; WESSELS, A. ; SCHREMMER, C.: Tilted tabletops: In between horizontal and vertical workspaces. In: *Proceedings of Workshop on Horizontal Interactive Human Computer Systems (TABLETOP)*, 2008, S. 49–56
- [Mun09] MUNZNER, T.: A Nested Process Model for Visualization Design and Validation. In: *IEEE Transactions on Visualization and Computer Graphics* 15 (2009), Nr. 6, S. 921–928. – ISSN 1077–2626
- [Pal87] PALMQUIST, S.: Knowledge and Experience - An Examination of the Four Reflective 'Perspectives' in Kant's Critical Philosophy. In: *Kant-Studien* 78 (1987), January, Nr. 1-4, S. 170–200



- 
- [PHP03] PFITZNER, D. ; HOBBS, V. ; POWERS, D. M. W.: A Unified Taxonomic Framework for Information Visualization. In: *InVis.au*, 2003, S. 57–66
- [Pre09] PRESS, Microsoft: *.NET Application Architecture Guide, 2nd Edition*. Microsoft Press, 2009. – ISBN 978–0735627109
- [RFF<sup>+</sup>08] ROBERTSON, G. ; FERNANDEZ, R. ; FISHER, D. ; LEE, B. ; STASKO, J.: Effectiveness of Animation in Trend Visualization. In: *IEEE Transactions on Visualization and Computer Graphics* 14 (2008), Nr. 6, S. 1325–1332. – ISSN 1077–2626
- [RL04] ROGERS, Y. ; LINDLEY, S.: Collaborating around vertical and horizontal large interactive displays: which way is best? In: *Interacting with Computers* 16 (2004), Nr. 6, S. 1133–1152. – ISSN 0953–5438
- [RM90] ROTH, S.F. ; MATTIS, J.: Data characterization for intelligent graphics presentation. In: *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*. New York, NY, USA : ACM, 1990 (CHI '90). – ISBN 0–201–50932–6, S. 193–200
- [SB03] SHNEIDERMAN, B. ; BEDERSON, B.: *The Craft of Information Visualization: Readings and Reflections*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 2003. – ISBN 1558609156
- [SBM08] SIMOFF, S. J. ; BÖHLEN, M. H. ; MAZEIKA, A.: Visual Data Mining: An Introduction and Overview. In: *Visual Data Mining*. Springer, 2008. – ISBN 978–3–540–71079–0, S. 1–12
- [Sch07] SCHNEIDEWIND, J.: *Scalable Visual Analytics: Solutions and Techniques for Business Applications*, Dissertation Thesis, University of Konstanz, Konstanzer Online-Publikations-System (KOPS), 2007
- [SEI10] SVETACHOV, P. ; EVERTS, M. H. ; ISENBERG, T.: DTI in Context: Illustrating Brain Fiber Tracts In Situ. In: *Comput. Graph. Forum* 29 (2010), Nr. 3, S. 1023–1032
- [SH02] STOLTE, C. ; HANRAHAN, P.: Polaris: A System for Query, Analysis and Visualization of Multi-dimensional Relational Databases. In: *IEEE Transactions on Visualization and Computer Graphics* 8 (2002), S. 52–65
- [Shn83] SHNEIDERMAN, B.: Direct Manipulation: A Step Beyond Programming Languages. In: *Computer* 16 (1983), Nr. 8, S. 57–69. – ISSN 0018–9162
- [Shn96] SHNEIDERMAN, B.: The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In: *VL*, 1996, S. 336–343

- [SIB<sup>+</sup>11] SEDLMAIR, M. ; ISENBERG, P. ; BAUR, D. ; MAUERER, M. ; PIGORSCH, Ch. ; BUTZ, A.: *Cardiogram: visual analytics for automotive engineers*. In: *Proceedings of the 2011 annual conference on Human factors in computing systems*. New York, NY, USA : ACM, 2011 (CHI '11). – ISBN 978–1–4503–0228–9, S. 1727–1736
- [SIBB10] SEDLMAIR, M. ; ISENBERG, P. ; BAUR, D. ; BUTZ, A.: *Evaluating Information Visualization in Large Companies: Challenges, Experiences and Recommendations*. In: *Proceedings of Conferenece on Beyond Time and Errors: Novel Evaluation Methods for Information Visualization (BELIV)* ACM, 2010, S. 79–86
- [Spe07] SPENCE, R.: *Information Visualization: Design for Interaction (2nd Edition)*. Upper Saddle River, NJ, USA : Prentice-Hall, Inc., 2007. – ISBN 0132065509
- [Ste46] STEVENS, S.S.: *On the Theory of Scales of Measurement*. In: *Science* 103 (1946), June, Nr. 2684, S. 677 – 780
- [STH02] STOLTE, C. ; TANG, D. ; HANRAHAN, P.: *Query, analysis, and visualization of hierarchically structured data using Polaris*. In: *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA : ACM, 2002. – ISBN 1–58113–567–X, S. 112–122
- [STT06] SNOWDEN, R. ; THOMPSON, P. ; TROSCIANKO, T.: *Basic Vision: An Introduction to Visual Perception*. 1st. OUP Oxford, 2006. – ISBN 978–0199286706
- [TAS94] THOBEN, W. ; APPELRATH, H.-J. ; SAUER, S.: *Record Linkage of Anonymous Data by Control Numbers*. In: GAUL, W. (Hrsg.) ; PFEIFER, D. (Hrsg.): *From Data to Knowledge: Theoretical and Practical Aspects of Classification, Data Analysis and Knowledge Organisation*. Springer-Verlag, 1994, S. 412–419
- [TC05] THOMAS, J.J. ; COOK, K.A.: *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Center, 2005
- [TF08] TEIKEN, Y. ; FLÖRING, S.: *A Common Meta-Model for Data Analysis based on DSM*. In: GRAY, Jeff (Hrsg.) ; SPRINKLE, Jonathan (Hrsg.) ; TOLVANEN, Juhha-Pekka (Hrsg.) ; ROSSI, Matti (Hrsg.) ; ACM SIGPLAN, DSM FORUM (Veranst.): *Proceedings of the 8th DSM Workshop at OOPSLA 2008*. New York, NY, USA : ACM, 2008, S. 35–38

- 
- [Tho97] THOMSEN, E.: *OLAP solutions: building multidimensional information systems*. New York, NY, USA : John Wiley & Sons, Inc., 1997. – ISBN 0-471-14931-4
- [TIC09] TOBIAS, M. ; ISENBERG, P. ; CARPENDALE, S.: Lark: Coordinating Co-located Collaboration with Information Visualization. In: *IEEE Transactions on Visualization and Computer Graphics* 15 (2009), Nr. 6, S. 1065–1072. – ISSN 1077-2626
- [TRM10] TEIKEN, Y. ; ROHDE, M. ; MERTENS, M.: MUSTANG: Realisierung eines Analytischen Informationssystems im Kontext der Gesundheitsberichterstattung. In: *GI Jahrestagung*, 2010, S. 253–258
- [Tuf83] TUFTE, E.R.: *The Visual Display of Quantitative Information*. Graphics Press, 1983
- [Tuk65] TUKEY, J.W.: The Technical Tools of Statistics. In: *The American Statistician* 19 (1965), Nr. 2, S. pp. 23–28. – ISSN 00031305
- [Uph10] UPHOFF, A.: *Entwicklung einer Abstraktionsschicht zwischen Darstellungen, Eingabeformen und Datenquellen zur visuellen Analyse*. Uhlhornsweg 49-55, 26129 Oldenburg, April 2010
- [VPF06] VALIATI, E. ; PIMENTA, M. ; FREITAS, C.: A taxonomy of tasks for guiding the evaluation of multidimensional visualizations. In: *BELIV '06: Proceedings of the 2006 AVI workshop on BEyond time and errors*. New York, NY, USA : ACM, 2006. – ISBN 1-59593-562-2, S. 1–6
- [WL90] WEHREND, S. ; LEWIS, C.: A problem-oriented classification of visualization techniques. In: *VIS '90: Proceedings of the 1st conference on Visualization '90*. Los Alamitos, CA, USA : IEEE Computer Society Press, 1990. – ISBN 0-8186-2083-8, S. 139–143
- [ZSAL08] ZUDILOVA-SEINSTRAS, E. ; ADRIAANSEN, T. ; LIERE, R.v.: *Trends in Interactive Visualization: State-of-the-Art Survey*. 1. Springer Publishing Company, Incorporated, 2008. – ISBN 1-848-00268-8