FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Visual Analytics of Clinical Data and Treatment Processes for Cohorts

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Medizinische Informatik

eingereicht von

Jürgen Unger, BSc
Matrikelnummer 0827133

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Mag.rer.soc.oec. Dr.rer.soc.oec. Silvia Miksch
Mitwirkung: Projektass. Paolo Federico MSc

Wien, 26.11.2014

(Unterschrift Verfasserin) (Unterschrift Betreuung)

Visual Analytics of Clinical Data and Treatment Processes for Cohorts

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Medical Informatics

by

Jürgen Unger, BSc
Registration Number 0827133

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Ao.Univ.Prof. Mag.rer.soc.oec. Dr.rer.soc.oec. Silvia Miksch
Assistance: Projektass. Paolo Federico MSc

Vienna, 26.11.2014

(Signature of Author) (Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Jürgen Unger, BSc
Hochmühlstrasse 24, 2514 Traiskirchen

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)	(Unterschrift Verfasserin)

Acknowledgements

First and foremost I would like to thank my parents for both their continuing and mental support through my studies. A big thank you to my father who continuously proof read this thesis and improved its readability vastly. The German abstract would not be nearly as good without my mother, who provided her invaluable expertise in punctuation in the German language. The next acknowledgements go out to all my friends, who provided me with ideas, inspiration and balance in my life. Kudos also go to my test subject Sebastian Stasek, who bravely solved all user tasks and gave additional and meaningful input. Last but not least I would like to thank my advisors Paolo Federico, Albert Amor Amos and Silvia Miksch for their continuing scientific input, patience, support, proof reading and testing of this thesis and prototypes.

Abstract

Recommendations for the medical treatment of patients, by a physician, can be provided as so called Clinical Guidelines. A lot of previous and current research focuses on how to integrate the computerized form of Guidelines, Computer Interpretable Guidelines into clinical practice. Such an integration could be used to provide real time-decision support. The retrospective analysis of adherence to a guideline is also an area of interest. Such an analysis provides insight into guideline quality in terms of treatment success and the general acceptance of the guideline by healthcare professionals. Medical studies are often conducted with a cohort, which is a group of patients who share a common characteristic. In case of this thesis, the commonality is that all patients of the cohort received the same treatment recommended by the guideline we want to evaluate.

The goal of our thesis is to enable a retrospective visual analysis of the treatment data of a whole cohort. Furthermore the information is not only visualized but also interaction techniques are provided to the user, to enable exploratory analysis.

We extended techniques for single patients and also implemented new approaches for the visualization of all measurements of certain clinical parameters as well as the executed actions and adherence to a guideline by healthcare professionals. For both distinct types of data we developed one visualization technique that aggregates the information and another one that keeps as much detail as possible for each patient. State of the art visualizations for guideline and statistical compliance information were extended to handle the accumulated data within a cohort.

The result is a fully functional prototype based on the Java Programming Language and the Prefuse visualization framework. All developed visualization techniques were implemented in the prototype and are available for use. In an evaluation with a domain expert we assessed the usability of the techniques and visual encodings. We found out, that they are mostly intuitive and understandable. Some of them are harder to grasp but only short introductions were necessary for the expert to properly use them as well. During the conduction of the thesis and the evaluation we were able to identify approaches and ideas for subsequent future research and present them briefly.

Kurzfassung

Empfehlungen für medizinische Behandlungen von Patienten durch medizinisches Fachpersonal können in Form von medizinischen Leitlinien zur Verfügung gestellt werden. In die Forschung an der Integration von medizinischen Leitlinien in die klinische Praxis wurde und wird viel Aufwand investiert. Diese kann Empfehlungen für zu setzende klinische Aktionen während der Behandlung durch Computer ermöglichen, aber auch eine retrospektive Analyse der Einhaltung der Leitlinien ist von großem Interesse für Mediziner und Forscher. Diese kann Auskunft über die Qualität der Leitlinie bezüglich deren Behandlungserfolg, aber auch über die Akzeptanz der Leitlinie durch medizinisches Fachpersonal geben. Klinische Studien werden in vielen Fällen als Kohortenstudie durchgeführt. Eine Kohorte ist eine Gruppe von Patienten die zumindest ein gemeinsames Merkmal aufweisen. Im Falle dieser Arbeit ist die Gemeinsamkeit, dass alle Patienten die gleiche medizinische Behandlung, welche durch eine medizinische Leitlinie empfohlen wurde, erfahren haben.

Ziel dieser Arbeit ist es eine retrospektive visuelle Analyse der Behandlungsdaten einer ganzen Kohorte zu ermöglichen. Neben der Visualisierung dieser Informationen, stellen wir dem Benutzer Interaktionstechniken zur Verfügung, welche eine explorative Analyse der Daten ermöglichen.

Dazu passen wir Visualisierungen welche für die Daten eines einzelnen Patienten entworfen wurden an. Weiters implementieren wir neue Ansätze für die Visualisierung von Messergebnissen klinischer Werte, die angewandten klinischen Aktionen und die Einhaltung der Leitlinie durch medizinisches Personal. Sowohl zur Visualisierung der Messdaten als auch der dokumentierten klinischen Aktionen haben wir jeweils einen Ansatz entwickelt, welcher die Informationen aggregiert darstellt und einen weiteren, der es ermöglicht, Details für alle Patienten zu erkennen. Zusätzlich haben wir bereits entwickelte Visualisierungen für medizinische Leitlinien und statische Information über die Befolgung dieser zur Visualisierung der Informationen einer ganzen Kohorte erweitert.

Das Resultat ist ein voll funktionaler Prototyp, welcher in Java entwickelt wurde und auf dem Visualisierungsframework Prefuse basiert. Alle entwickelten Visualisierungen wurden im Prototypen implementiert und stehen für den Benutzer zur Verfügung. Bei einer anschließenden Evaluierung, die wir gemeinsam mit einem Fachexperten durchgeführt haben, konnten wir feststellen, dass der Großteil der entwickelten Methoden und visuellen Darstellungen der Informationen intuitiv und schnell zu verstehen sind.

Contents

CHAPTER 1

Introduction

1.1 General Introduction

Physicians aim to achieve the best possible treatment for the individual patient by using the current best known evidence provided by medical research. This evidence might be provided in the form of clinical guidelines, which are a set of schematic plans for management of patients who suffer from a particular clinical condition [55]. Clinical guidelines can be provided as Computer Interpretable Guidelines (CIGs), with the goal of a deeper integration of recommended treatment plans in clinical practice.

When the applied clinical actions in a treatment process are recorded, it is possible to compare the actual process for a patient with the recommendations presented by the guideline. Raw clinical parameters can be recorded over the course of the treatment process and presented in a time-aligned manner with the sequence of applied clinical actions. Another possibility to obtain such data is to retrospectively extract treatment data from collected electronic health records. Connecting both sets of data, enables the observation of effects of the applied actions on the evolution of the patient's clinical parameters, and enables us, to determine the overall adherence of the actions to a provided guideline.

Many medical studies are conducted on groups of patients, called cohorts to enable comparison of the effects of different treatments by statistical methods. An example is the evaluation of new drugs, where it is necessary to show the possible treatment improvement, by comparing a group of patients which are treated with the tested drug with another group of patients who are given a placebo. Extending guideline compliance and treatment analysis to cohorts, presents an approach to find patterns in the treatment process of the cohort itself and also for further research and improvement of clinical guidelines.

While statistic aids in the numerical proof of benefits or lack thereof, some patterns in the data are often not discovered by such methods. Visual Analytics and Information Visualization offer an alternative approach to potentially discover such patterns through properly visualizing the gathered information and employing the human cognitive mind to solve the task. Information visualization is used to present information in an intuitive manner, enable clinicians to gain insight about a patient's current health status and track state changes, trends and patterns over time.

1.2 Background and Motivation

While Information Visualization and Visual Analytics are relatively new fields, many techniques and systems for the visualization of medical data already exist. In fact medicine is one of the biggest application domains for Information Visualization. This is most likely the fact, because the amount of collected medical data (tests, patient monitoring, treatment data, meta-data) is quite vast. When treating patients, clinicians have to take all data they posses about a patient under consideration and make their treatment decisions accordingly.

Modern information technology already enables easy and fast access to medical data in the form of Electronic Health Records (EHRs). With the implementation and advance of EHRs the amount of data a clinician has to review, can potentially increase enormously, but they are also able to provide access to huge amounts of clinical data for scientific research. A possible and commonly used form of data presentation is in the form of spreadsheets, which are difficult to interpret for the human mind. Representing data in an appropriate visual form yields far shorter perception times in humans and therefore might decrease decision time of physicians in life critical situations.

A partial task in solving this problem is the presentation of Computer Interpretable Guidelines in a visual form. Computer Interpretable Guidelines are recommendations for the conduction of a treatment provided by healthcare authorities, while researchers aim to integrate them in clinical practice which might result in the following advantages:

- Increase the overall treatment quality
- Reduce the risks for patients
- Increase the satisfaction of patients
- Increase the quality of the healthcare system
- Reduce healthcare costs

The adherence to a guideline can be an indicator of how well the patient's treatment has been fulfilled and the treatment process in general, but it can also give insight about the quality of a guideline. If overall adherence to the guideline is low, the recommendation is not commonly followed by physicians, which might indicate that the guideline is impractical for some reason.

Properly analyzing the adherence to guidelines and detecting patterns for larger groups of patients is a challenging task due to the sheer amount of data.

When clinical practice guidelines are applied frequently, treatment information for large groups of patients can be collected. This information might contain interesting patterns, which are not observable in the data collected from a single patient. Most of the state-of-the-art approaches are however limited to the presentation of the data of a single, or a few patients in parallel. Finding visualizations, which are able to properly represent the evolution of the clinical parameters of whole cohorts and also the overall compliance to the treatment plan might enable the visual detection of such patterns.

The finite amount of screen space and the limitations of the human cognitive mind, set strong boundaries on the amount of simultaneously presentable information, which renders it impossible to simultaneously visualize all data points and attributes for even a medium sized cohort. Abstracting and aggregating the data for the cohort into a more general form for visualization presents a promising approach. It is the goal of this thesis is to research and develop interactive visualization methods, which are able to adequately visualize the aggregated data for a cohort and aid medical experts in the visual recognition of interesting and relevant patterns.

1.3 Description of the Addressed Scientific Problem

Developing meaningful, interactive and useful visualization techniques for cohorts in the context of treatment processes, yields a set of several scientific problems.

The first important partial problem is to find meaningful aggregations and presentation of the contained data. Visualizing all the collected data for a large cohort would result in meaningless visual clutter. Such techniques should enable the user to gain insight into the clinical development of the presented cohort, while some detail about single patients should still be perceptible. Finding applicable aggregation techniques is a non-trivial task, because aggregated information might hide many trends in the cohort in average values. Different techniques will be suitable for conveying different attributes of the data, therefore a single optimal technique for all upcoming issues does not exist. Since the compliance to a guideline is compared to the evolution of parameters, situations might arise, where these two different data-sets cancel each other out and trends are not observable anymore.

While details about single patients are important, an overall trend for the whole cohort should be visible at a glance and directly comparable to another cohort (e.g. for direct comparison of guideline quality, when comparing with a cohort treated according to another guideline). The clinical trend as well as the total compliance to the guideline need to be presented in an intuitive way. Further applications include the comparison of the evolution of the overall health status of cohorts with different level of guideline adherence.

Another challenge in developing visualizations, is the mapping of the attributes of the data to appropriate spatial positions, color and marks. Since we deal with time-oriented data we will most likely stick to the paradigm of assigning time to the x-axis. All other attribute assignments are however largely debatable. Also an extensive point of research will be, how to properly visualize multivariate treatment data (multiple clinical parameters)

Interaction techniques are able to vastly increase the usability and usefulness of visualization techniques. Especially the information seeking mantra introduced by Shneiderman: overview first, zoom, details on demand [56] is a useful recommendation and widely used. Therefore an additional research question is, how the end-user should be able to interact with the developed visualizations. Solutions to questions like, should she be able to zoom in the data, filter it in some way, or reorganize it altogether to another point of view, need to be found.

1.4 Research Questions

Based on the problem description the following research questions, of this thesis, are formulated as follows:

RQ: How to aid medical experts in the analysis of the treatment data of a cohort, by means of visual analysis?

Several additional questions arise from the main research question:

Rq1: Which approaches are suitable to abstract and aggregate the treatment and compliance data of a cohort?

Rq2: Which visualization techniques provide the best insight into the aggregated data, while preserving information about single patients?

Rq3: Which interaction techniques are suitable for these visualizations?

After the development of experimental techniques, the visualizations will be evaluated against the following qualitative research question:

Rq4: Are the developed visualization techniques suitable for solving the problem, and how well can they be understood and used?

1.5 Research Methodology

The methodological approach chosen to research the given problem, is comprised of the following parts:

1. **Literature Review and Research:**

   The first step is the research and selection of relevant literature. The selected literature should especially cover one of the following topics which are relevant to the problem at hand:

   - Clinical Guidelines
   - Time-Oriented Information Visualization
   - Information Visualization in the medical domain
   - Abstraction and Aggregation of clinical data
   - Visualization techniques for cohorts

   The thesis will contain a survey of the most relevant literature encountered in this step.

2. **Selection, refinement and development of suitable visualization techniques**

   Suitable information visualization techniques will be selected from the literature and adjusted to properly suit the given task. Also new techniques will be developed by combining different approaches, which show promising characteristics.

3. **Prototype Development**

   A prototype will be developed which will include the most promising techniques determined in the previous step. This implementation will integrate and extend the *CareCruiser* [23] and *VisuExplore* [46] prototypes described in section 3.4 Preceeding Work and should allow the evaluation of the implemented visualization techniques for clinical data of cohorts, together with a medical expert.

4. **Evaluation**

   In the last step an evaluation of the developed information visualization methods preferably by a medical expert will be conducted.
   These results are collected and used to the determine if the developed techniques show potential in aiding physicians to detect patterns in the treatment data of a cohort of patients. The general usability of the implemented techniques will be the second important evaluation criterion we focus on.

1.6 Structure of this Thesis

This thesis is structured into the following chapters:

- **Chapter 2 Methods and Concepts:** In the second chapter we present the methods and concepts which are important to the topic and relevant to this thesis. We start by giving general introductions to Time Oriented Information Visualization and Visual Analytics. In center sections of the chapter, we give an introduction to Clinical Practice Guideline (CPG), their computerized form: Computer Interpretable Guidelines (CIG) and the plan specification language Asbru we use for our prototype. The last two sections contain brief introductions to Action Compliance and the concept of Cohorts as used in this thesis.
- **Chapter 3 Related and Previous Work:** This chapter provides an overview of work that is related to our thesis. We start out by presenting Related Overview Literature and describe the Search Methodology we used to find relevant literature. We continue by presenting Related Work in Guideline Visualization, Compliance Analysis, Visualization of Guideline Compliance Over Time and Techniques for Temporal Abstraction of Clinical Parameters. We end the chapter by describing Selected Systems with Visualization Techniques for Cohorts. The chapter ends with an introduction to the Preceeding Work our prototypes are based on and a Discussion of the Presented Systems and Techniques.
- **Chapter 4 Implementation**: At the start of the fourth chapter we describe the Frameworks our prototypes are built on. We then describe The Integration of CareCruiser and VisuExplore and the Integration and Extensions for Cohorts in the new prototype. We conclude the chapter with the Technical Documentation of the created software prototype.
- **Chapter 5 Evaluation:** In this chapter we assess the usability of the visualization techniques we developed. We provide a short survey of Evaluation Techniques for Information Visualizations and continue by describing the Evaluation Method we used. At the end of the chapter we present the Results we obtained.
- **Chapter 6 Discussion and Future Research:** In this section we discuss the findings of our work and outline the ideas for subsequent research.
- **Chapter 7 Conclusion:** We finish the thesis by a short summary and conclusion of our work and results.

CHAPTER 2

Methods and Concepts

This section describes the approaches and methods used to conduct our thesis. We start by giving introductions to Information Visualization (section 2.1 Information Visualization), the more specific topic of the concept of time in information visualization (section 2.2 The Dimension of Time in Information Visualization) and Visual Analytics (section 2.3 Visual Analytics) are given.
The following three sections describe Clinical Practice Guidelines (CPG) (section 2.4 Clinical Practice Guideline (CPG)), their computerized integration Computer Interpretable Guidelines (CIG) (section 2.5 Computer Interpretable Guidelines (CIG)) as well as a concrete implementation of such guidelines provided by the Asbru plan language (section 2.6 Asbru).
We conclude this chapter with definitions and a description of the concept of cohorts (section 2.8 Cohorts).

2.1 Information Visualization

Information visualization is the study of visually representing abstract data to enable cognitive insight into the data by employing the human mind. The insight or understanding is acquired, by looking at these graphics. Spence [58] therefore emphasizes that visualization is solely a human cognitive activity and has nothing to do with computers.

Information visualization is a sub-field of visualization, while it has to be distinguished from the related field of scientific visualization. A visualization is characterized as scientific, when the spatial representation is given and not chosen.

The two following definitions are used throughout the literature:

Visualize: To form a mental model or mental image of something [58].

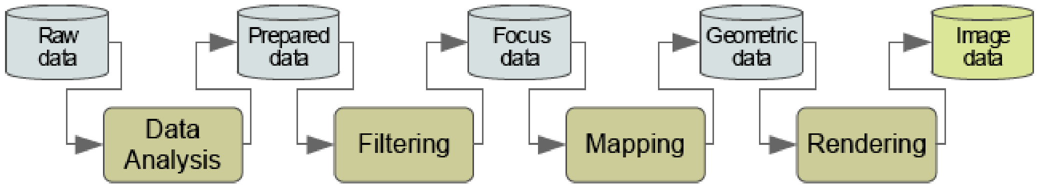

Figure 2.1: The extended Information Visualization Pipeline as presented in [18] and [3].

Information Visualization: The process of forming a mental model of information.

A more comprehensive description is given by Thomas and Cook in [63]:
Visual representations and interaction techniques take advantage of the human eye's broad bandwidth pathway into the mind to allow users to see, explore, and understand large amounts of information at once. Information visualization focused on the creation of approaches for conveying abstract information in intuitive ways.

Information visualization has already been and is increasingly applied to aid users in solving problems in many domains like scientific research, finance, market analysis, medicine and manufacturing. Some applications also include time-critical problems, like the proper response of emergency teams to a threat. The goal of a well developed visualization is therefore to provide insight into the abstract data at a glance and representing it in a form as to make the solution transparent to the user [58].

Development of visualizations is however a non-trivial task. Often the amount of collected data is vast, and representing it all is an impossible task. Also the proper and intuitive use of mappings of data to space and the variety of possible visual attributes like colors symbols or structures, might represent a demanding challenge even for experienced designers.

Visualizations of abstract data have been already present for centuries [58], but the emergence of computer technology has enabled massive advances in the field. Computer technology provides crucial abilities like storing vast amounts of data, powerful and fast computations for the selection and filtering of subsets of data. The rapidly evolving display technology, with ever increasing screen spaces and resolutions increases the amount of data which can be simultaneously represented. Of even greater benefit is the ability to create interactive visualizations, which intuitively react to user input. They offer the possibility to the user to actively change and steer the visualization in an intended direction.

A popular and important concept in the field of information visualization is the information visualization pipeline shown in figure 2.1.

The following steps are described in the process for producing a visual representation of the raw data introduced in the visualization pipeline [3].

1. *Data Analysis:* Preparation of Data for visualization (e.g. filtering, smoothing interpolation, normalization, scaling)

2. *Filtering:* Selection of the subset of the data which will be visualized

3. *Mapping:* Assigning visual attributes to data values, like the geometric primitives color, position and size

4. *Rendering:* Transformation of the geometric primitives to visual image data

While the first two steps might be omitted in case the raw data is already correctly prepared and filtered, the last two steps are always necessary to produce a visualization. Due to the ever growing amount of data which is often also heterogeneous this however only applies to a fraction of the applications for information visualization. While we mentioned, that the proper assignment of visual attributes to dimensions in the data is of immense importance for the expressiveness of a designed visualization, proper analysis and filtering of the data sets also becomes more and more important. Solving this problem may be accomplished by the proper application of appropriate data mining algorithms [63].

2.2 The Dimension of Time in Information Visualization

Time is one of the four dimensions humans live in and are able to perceive. Unlike in space, we are not able to navigate through time by our choice. For example we have to let it slip by while patiently waiting for an event to happen. In most cases we perceive time as linear and we move through it into the direction of the future. In Time-Oriented Information Visualization, time can also have cyclic components, like cycles of the moon or seasons. Such components can be used to find cyclical patterns which reoccur over certain intervals of time.

It is a common dimension present in everyday life as well as in many domains like business, engineering, science, biography, history, planning or project management. [9]. We are often able to learn from past events to detect reoccurring patterns and predict the future with some certainty.

The sub-domain of time-oriented information visualization deals with methods to present time in an explorable and analyzable manner. Medical treatment data is almost always time-oriented. Treatment actions are conducted one after another, drugs are applied in certain intervals and clinical parameter measurements are taken at some point in time. Exceptions might be the static data like a patients name or the connection to his relatives.

For a comprehensive survey of the methods and concepts in time-oriented information visualization, we kindly refer to reader to [9].

2.3 Visual Analytics

The concept of Visual Analytics as a distinct field has been established in 2004 by Wong and Thomas in [68]. They describe Visual Analytics as an outgrowth and combination of information visualization and scientific visualization, and define it as: *“the science of analytical reasoning facilitated by visual interactive interfaces“*. Automated analysis is combined with interactive visualizations for effective understanding, reasoning and decision making on the basis of very large and complex data sets [63].

In [26] Keim et. al. get into more detail and identify the main goal of Visual Analytics which is to *“create tools and techniques which enable people to:“*

- Synthesize information and derive insight from massive, dynamic, ambiguous and often conflicting data
- Detect the expected and discover the unexpected
- Provide timely, defensible and understandable assessments
- Communicate assessment, effectively for action

Information visualization and Visual Analytics are however not two easily separable fields, because both have some overlapping concepts. The most notable difference is that Visual Analytics includes the use of analytical processes such as data mining or the use of statistics and enables interactive exploration, whereas Information Visualization rather deals with questions like how attributes of the data should be mapped to appropriate geometric structures and color. Scientific visualization is the third overlapping field, distinguished by the presence of a geometric structure or natural spatial mapping. This could for example be the part of the body in a CT scan, or a spatial structure in Computational Fluid Dynamics analysis.

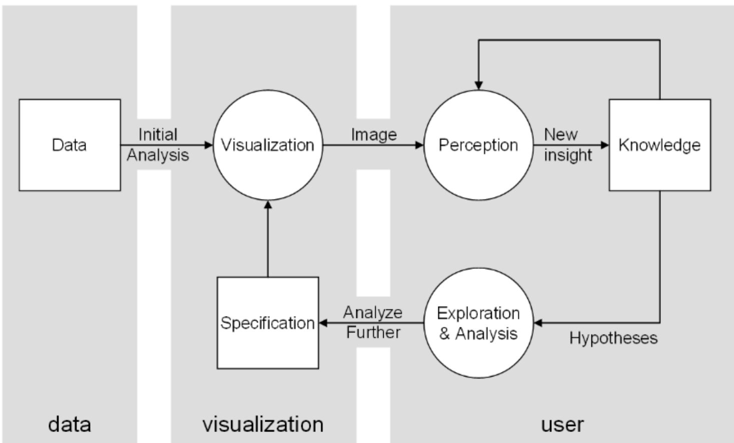

Figure 2.2: The sense making loop presented in [26].

As stated in the definition by Wong and Thomas [68] the interfaces are interactive and the user is able to steer the visualization in their desired direction. This process is more formally named

Visual Analytics or knowledge discovery process. It is structured by the sense-making-loop presented in figure 2.2. After an initial analysis of the data (filtering of irrelevant data, statistical analysis, aggregation), the precomputed data is visualized and presented to the user. She is then able to gain cognitive insight by visual examination of the data. This might induce the discovery of previously unknown knowledge leading to new hypotheses which have previously not been fought off. Through further exploration and analysis, the user can interactively control and alter the visualization, while starting a new iteration of the loop if further analysis is required.

2.4 Clinical Practice Guideline (CPG)

Clinical Practice Guidelines (CPG) are documents, which are used to provide treatment plans and actions for specific clinical conditions and situations to clinicians. Guidelines in different forms have existed for a long time. Academic medicine today is based on the best-known evidence provided by the current state-of-the-art knowledge in the field.

A commonly used definition according to Cronenboorg et. al. [48] reads as follows:
"Guidelines are systematically developed statements to assist practitioners and patient decisions about appropriate healthcare for specific circumstances".

Guidelines are developed in cooperation of many practitioners and scientists in the field, medical organizations or governmental bodies, and are the result of a consensus about the best clinical treatment practice in the given situation as shown in figure 2.3. They represent the best known evidence for the treatment in the form of a published guideline. The goal is to standardize medical care, provide a best practice treatment for all patients and optimize cost-effectiveness of the treatment.

For a clinician, who is treating his patient, this represents a recommendation she should follow over the course of the treatment. A general guideline, is however only able to cover the average patient and might not be applicable in certain abnormal or unusual situations. Even a well developed guideline can not substitute the intuition and implicit knowledge of an experienced clinician.
Often guidelines are subject to continuing change, because new found knowledge, drugs or concepts might present alternative treatment courses and alter the best-known evidence.

2.5 Computer Interpretable Guidelines (CIG)

Clinical Practice Guidelines are often provided in the form of large paper-based documents, which are difficult to read and apply in the care process [17]. Information systems can be employed to provide Clinical Practice Guidelines in the digital form of so called Computer Interpretable Guidelines (CIGs). CIGs can be instantiated for every single patient and automatically compared to the actual treatment processes conducted by the clinician. With visualization methods we are also able to present them in several different and suitable forms applicable to different

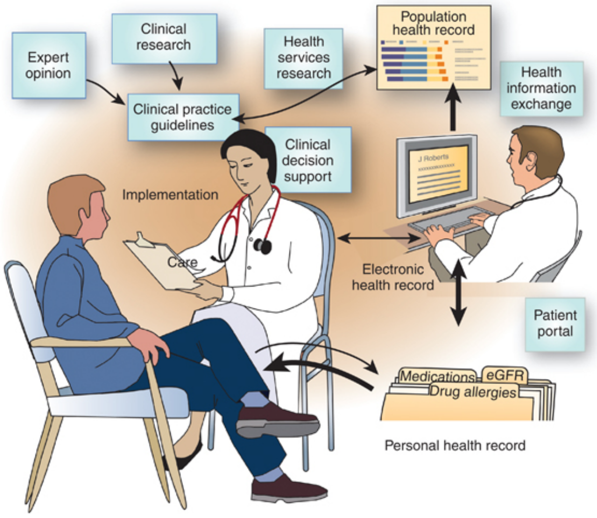

Figure 2.3: Sources which influence Clinical Practice Guidelines according to [64].

tasks.

Deviations from the given recommended guideline in the treatment can be recognized and presented to interested parties [17]. Guidelines in computerized form can also be implemented in Decision Support Systems (DSS), which can be used to actively support physicians in making decisions during a treatment. The knowledge needed for providing suggestions, is provided by the given guideline [37].

A variety of different approaches for the computer-based formalization of CIGs exist, which have been surveyed by Peleg et. al in [38]. For the prototypes developed in our thesis, we use the plan language developed in the Asgaard project [55] by Shahar et. al. named Asbru.

2.6 Asbru

Asbru is a modeling language for plans or Computer Interpretable Guidelines which we use for the conduction of this thesis. The language was created as part of the Asgaard project, which was started in 1996 [4], [55].

The project was set up with the intention of developing a set of techniques, that support the design and execution of skeletal plans, while the plans can be designed and executed by two different individuals. Next to the problem of developing the plan language, other research areas of the project included the development of a critiquing module and a visualization tool for guidelines called AsbruView, which we both present in section 3.3.

In [50] the creators of Asbru describe the language in the following way: *"Asbru is a time-oriented, intention-based, skeletal plan-specification representation language that is used in the Asgaard Project to represent clinical guidelines and protocols in XML"*. Such skeletal plans can be instantiated for each single treatment and enable the reuse of given domain-specific knowledge, but leave room for flexibility for the physician to achieve the intended goals during the execution of the treatment.

The concepts of Asbru are not strictly limited to medicine, but can be used to express clinical guidelines as skeletal plans which are reusable and instantiable for each treatment of a single patient. We describe some of the concepts and features of the Asbru plan language in the following subsections.

Language Elements

A plan represented in the Asbru plan language can consist of the following elements, which determine specific properties like goals intentions costs or conditions and allow a flexible execution of the plan [4].

- **Arguments**: These are values which can be passed from an invoking plan to a sub-plan it executes.
- **Preferences**: Preferences are used to set constraints on resources or costs and influence the selection of plans.
- **Intentions**: Intentions model the high level goals of a plan. The are described by the creator and are modeled by temporal patterns of actions and states that should be maintained achieved or avoided.
- **Conditions**: Conditions are used to define values, at which the state of a plan during execution is altered. Examples are abort or complete conditions, which when fulfilled stop the execution of the plan or filter conditions which determine if the plan is applicable.
- **Effects**: Effects can be used to describe known relationships between plan arguments and measurable patient parameters by mathematical functions or in qualitative manners.
- **Plan body**: The plan body describes the actions or sub-plans to be executed.

Structure of the Plan Body

The plan body is a hierarchical tree structure, started by a parent node which can contain any set of sub-plans and so on. The leafs of the tree represent elemental actions of the plan. The body of each plan can be executed in different patterns and orders:

There are three different types of plans, while only one of them is allowed in a single plan body [33]:

- **Sequential**: The specified set of plans is executed in sequential order. All plans have to be successfully completed for the continuation of the next step in the parent plan.
- **Concurrent**: The plans can be executed in parallel or any order. Ordering constraints (start together, execute in any order, execute in total order) and continuation conditions (all plans should be completed to continue, some plans should be completed to continue) can be specified to determine the conditions for execution and completion.
- **Cyclical**: A cyclical plan can be repeated. If and at which frequency this happens, is determined by the optional temporal and continuation arguments. Examples include the number of attempts, the number of completions and temporal patterns.

Time Annotations in Asbru

As the problem covered in this thesis, many treatment plans and actions are time-oriented or dependent. In some cases only the order of events in a plan might be important, while other plans might rely on more complex time annotations which are able to model temporal uncertainty. Figure 2.4 shows the set of parameters a time annotation in Asbru consists of.

The most important variable is the reference point. This point can be a specific date or an occurring event during the execution of the guideline like the start of the plan itself or a specific applied action. Most of the other variables are time ranges which are defined in relation to the reference point. The Earliest and Latest starting shift parameters can be used to define the interval in which the action or plan can be started, while the Earliest and Latest Finishing shift define when it should end. The Minimum and Maximum duration parameters allow the definition of an interval which specifies the execution time of a plan or an action.

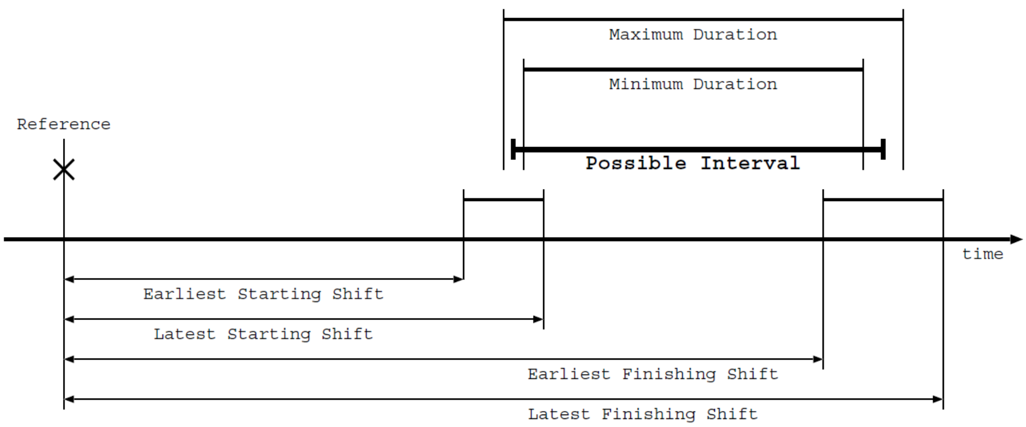

Figure 2.4: Time annotations in Asbru as presented in [4].

Plan States

In terms of the Asbru language the plan interpreter is a finite state machine and can reside in one of several different plan states during the selection and execution of plans. Figure 2.5, shows the possible states which we also describe shortly:

The states on the left in figure 2.5 show the states before an actual execution of a plan takes place, which are called plan-selection states. Plan selection states are used to determine if the plan is applicable in the given situation, according to certain parameters. First the plan is filtered according to criteria and is considered possible if they match, in a second step the plan setup is conducted and the state switches to ready. In all steps the plan is filtered and if some filtering step fails the plan switches to the rejected state, where it is considered inapt and will not be executed. Afterwards the plan can be executed manually or automatically.

In the right part of figure 2.5, the possible steps during the execution of the plan are illustrated. There are four possible steps which are: *activated* (initial step when the plan starts and is actually executed), *suspended* (the suspend condition is met, the plan will however be reactivated if the reactivate condition is satisfied), *aborted* (the plan gets aborted if the abort condition is met and can be reached from both the activated and suspended state), *completed* (the plan gets completed if the complete criteria are met during the activated state). Note that both, *aborted* and *completed* are terminal states, however the sooner was reached due to some problems in the treatment, while the later successfully concludes and finishes the treatment according to the plan.

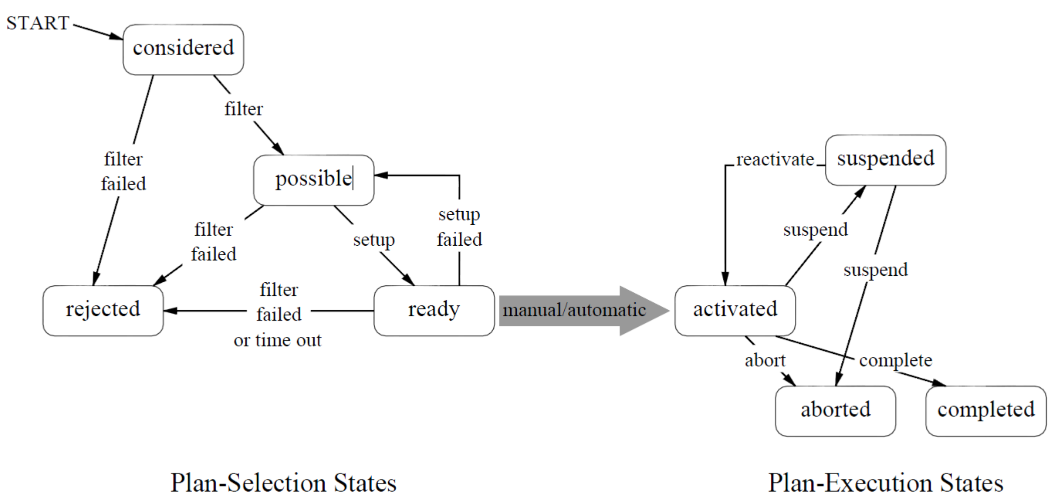

Figure 2.5: Possible states of an Asbru plan according to [4].

2.7 Action Compliance

In his master thesis [11] Bodesinsky identified and used the concepts of action definition described in this section, which are based on the previous work on the topic of compliance analysis.

During a treatment actions like the application of drugs or the ventilation of the patient are executed, which can be in compliance with a given CIG, falsely applied in some manner or even completely unrelated to it. To evaluate the compliance to a guideline, every action taken is recorded and the compliance of the conducted treatment is checked against the selected CIG. The evaluation is conducted on the level of atomic clinical actions, which are similar to events.

In terms of compliance an executed action can have three types [11]:

- **Valid Action:** The action was applied correctly
- **Invalid Action:** An action which has been applied, but should not have been applied (wrong action, conditions do not hold)
- **Missing Action/Missing Interval:** An action should have been applied, but it was not. If the action should be applied during a continuous interval in time, this time span represents a missing interval during which the action should have been applied.

Executed actions, which are not part of the guideline, are automatically marked as invalid. Since missing actions or missing intervals are actually non-existent in the data, they are computed and generated as events/intervals during compliance checking.

Furthermore Bodesinsky identified differences in the computation of compliance in different types of plans. He describes the compliance analysis for a guideline covering the artificial ventilation of newborn infants. This guideline consists of initial plans for setting the parameters where the plan consists of actions which have to be conducted in a sequential order. Another part of the guideline are however ventilation plans, which depend on time intervals and time relations. Therefore time-oriented concepts like minimum and maximum delays between actions have to be considered, which yields different conditions for compliance analysis.
We summarize the rules for compliance analysis used by Bodesinsky as follows:

- **Valid Action**
  - *Initial plans:* The condition for the action was true and it was executed in the correct order.
  - *Ventilation plans:* The conditions for the execution of the actions hold, while the action was also executed after the minimum delay between the last conducted action has passed.
- **Invalid Action**
  - *Initial plans:* An action was applied but the conditions were not met.

  - *Ventilation plans:* The action is marked as invalid if the conditions have not been fulfilled or the minimum time delay has been undercut or the action has been applied after the abort or complete state has been reached.

- **Missing Action/Missing Interval**
  - *Initial plans:* If an action is applied, while other actions should have been applied before the given action but have not the later are generated as missing actions
  - *Ventilation plans:* A missing interval is generated if the maximum delay to a previously applied action has been reached and the conditions for the action are true. The missing interval spans from the described point in time, until the conditions for the interval are not applicable anymore, if the correct action is applied or if the abort or complete conditions are fulfilled.

In [36] Panzarasa et. al. introduced two different decision support systems for the execution of clinical practice guidelines. Careflow aims to support real-time decision support during the treatment, without analyzing compliance. The second system is called Reasoning on Medical Action (RoMA) with the intended use as off-line tool for non-compliance detection.

RoMA is executed at discharge of the patient and enables physicians to analyze the care process when treatment has been finished. A report is created which consists of missing data according to the minimum requirements for guideline interpretation. Furthermore the report contains a list of recommendations a patient was eligible for, as well as the list of non-compliances which occurred and were found during analysis of the conducted treatment. Non-compliances are detected by violation of the rules. For example the criteria for the rule have been met during the treatment and therefore recommended to the physician, however he has not acted according to the recommendations given by the guideline. Users are also able to provide motivations for non-compliances and comment on them in free text fields for subsequent analyses of treatments by other users.

2.8 Cohorts

Several different definitions of the word cohort exist depending on the covered topic. The word was historically used in different domains, while relevant domains for us include medicine, epidemiology and statistics.
Therefore definitions of cohorts which are applicable in the context of this thesis are:

Epidemiology: *"A group of individuals sharing a common characteristic and observed over time in the group."* [1]

Statistics: *"A collection or sampling of individuals who share a common characteristic, such as members of the same age or the same sex."* [1].

The common characteristic in our case is the conduction of the same treatment and therefore the use of the same guideline. This does however not guarantee that the patients actually have the same disease, but the same diagnosis was made by a physician or the guideline was applicable because the patient parameters met the filter criteria as described in Asbru. While this might represent an interesting research approach (for example comparing the development of parameters for two different cohorts, who received treatments with different guidelines applied), this does however not represent the normal concept used.

The most relevant and frequently used application of the concept of cohorts is the conduction of so called cohort studies. Cohort studies are prospective longitudinal studies, in which a large group of patients, which have been exposed in different manners, but are not sick yet is observed over a longer time-frame. An example would be how smoking affects the incidence rate of lung cancer for smokers in their future lives and also their lifespan. Often vast amounts of data are collected and statistical methods are used to show if the results are statistically significant.

CHAPTER 3

Related and Previous Work

This chapter describes and presents the results of our literature and state of the art survey.
In the first part of the chapter (3.1 Related Overview Literature) we summarize the overview literature we identified, which already included an enormous amount of different techniques and systems related to our problem.
In the second part of the chapter (3.2 Search Methodology) we present the search methodology we used, for conducting our literature research and finding relevant literature in scientific online databases.
The third part of this chapter (section 3.3 Related Work) presents a collection of visualization techniques and systems, which cover one or more of the related concepts for this thesis. These especially include systems which focus on visualizing cohorts and the aggregation of data collected from multiple patients, which might present readily adaptable as well as inspiring approaches to our problem. Further focus lies on techniques for visualizing clinical guidelines, the analysis and visualization techniques for compliance to clinical guidelines, as well as temporal abstraction techniques for clinical parameters which are also relevant for the conduction of our thesis.
Section 3.4 Preceeding Work, describes the history and state of the art version of the *VisuExplore* and *CareCruiser* prototypes, which the prototype of this thesis will be based on, while in the last section (3.5 Discussion of the Presented Systems and Techniques) we compare and discuss the presented prototypes according to relevant criteria in our thesis.

3.1 Related Overview Literature

In this section we present the existing overview literature discovered in our literature survey. We give a short overview and the main results of both selected sources and point out relevant topics, which where not discussed.

Visualizing Time-Oriented Data - A Systematic View

In Visualizing Time-Oriented Data [9] written by Aigner, Miksch, Schumann and Tominski, provide the reader with an overview of techniques and the state of the art in the visualization of time-oriented data. In the first chapters the terminology and the history of relevant concepts are covered. The later chapters describe a systematic view of important visualization aspects and both approaches for interaction in an analytical as well as a supporting sense.

The largest part of the book (Chapter 7) is dedicated to an enormous survey of a variety of different approaches and systems for solving specific problems of time-oriented information visualization. A total of 101 different techniques are presented, which are divided by the criteria: Data frame of reference (abstract vs. spatial), data variables (univariate vs. multivariate), time arrangement (linear vs. cyclic), time primitives (instant vs. interval), visualization mapping (static vs dynamic) and visualization dimensionality (2D vs. 3D).

The first set of visualizations are very basic techniques like Line Plots and Bar Graphs. More specific systems are presented towards the end of the chapter including the following visualization systems for medical data: *IPBC* [16], *LifeLines* [42], *LifeLines2* [66], *Similian* [70], *CareCruiser* [23], *VIE-VISU* [25], *Gravi++* [24], *TimeRider* [44], *PatternFinder* [19], *Midgaard* [10], *VisuExplore* [46] and *KNAVE II* [54].

While the authors do not focus on problems related to the medical domain, it is evident that visualization systems for medical data represent a large fraction of the presented techniques. Due to the large amount of systems each one is only described on a single page. No emphasis is placed on the medical domain and no distinction according to criteria like single patient vs. multiple patients were made. Some approaches described in the book, which are not a part of the medical domain, might be relevant to our problem, since they might also present promising approaches to our research problem.

Interactive Information Visualization to Explore and Query Electronic Health Records

In [47], Rind et. al. conducted a comprehensive survey of information visualization systems, for exploring and querying medical data from electronic health records. This state-of-the-art survey was updated in 2013 and contains a systematic overview of the current systems for clinical data from academic literature. Also a review and comparison for 14 selected systems is conducted, while many more are mentioned in the paper.

These systems are compared according to the following criteria: Data types covered, multivariate analysis support, number of patient records used (one or multiple) and user intents addressed. Also specific focus lies on systems which combine views for treatment data as well as clinical guidelines. The results section is splits the covered systems into two subgroups: Techniques for the visualization of a single patient record and techniques for a collection of patient records.

The systems that are most interesting to us, are found in the later part of the chapter. The majority of these systems allow the user to align multiple patient records on a common time axis. Future research directions for the application of information visualization systems in Electronic Health Records are pointed out in the end.

All systems mentioned in the previously described book [9] are also included in this paper. Short descriptions of the following systems are included which are of interest to us: *MIVA* [20], *WBIVS* [39], *VISITORS* [27], *Caregiver* [13], *MSTA* [35], *LifeFlow* [29], *OutFlow* [69], *VisCare-Trails* [32], *EventFlow* [34] and *VisPap* [61].

Both sources, while providing visualization techniques for medical information, do not present details or cover criteria for the visualization of cohorts. Some of the described systems provide applicable approaches to our problem. Other techniques found in these sources, which cover other domains may also inspire alternative approaches.

3.2 Search Methodology

This section describes the methodology and approach we used to find relevant sources for our thesis. We give an overview of the literature we received at the start and the scientific databases, the keywords we used for our search and the results we got. We finish the section by listing the criteria for selecting the sources relevant to our thesis.

For the start of the literature research for this thesis, we were provided with the following articles:

- Visual Analysis of Compliance with Clinical Guidelines [12]
- CareCruiser: Exploring and Visualizing Plans, Events, and Effects Interactively [23]
- Challenges of Time-oriented Data in Visual Analytics for Healthcare [7]
- Interactive Information Visualization to Explore and Query Electronic Health Records [47]

The first three papers provide an overview of the state of the art research in medical visualization at Vienna University of Technology. This research mainly focused on the visualization of CIGs, evolution of patient parameters over time and the visualization compliance of a patients treatment to a guideline. In conjunction with [12], Bodesinsky conducted his master thesis [11] where he researched and developed visualization methods for the compliance information to medical guidelines. In his thesis, he also described the latest version of *CareCruiser* [23] prototype in which he integrated his methods.

As described in the previous Chapter 3.1 in [47], Rind et. al. provided an extensive overview over visualization techniques for EHRs, containing a whole section for visualization techniques of multiple patients. Many systems described in this section have been selected as interesting approaches, while the focus lay on systems including techniques for the time-oriented aggregation of patient or abstract clinical parameter.

The referenced papers gave an indication, which journals and conference proceedings should be searched for further literature. Table 3.2 summarizes the results of a subset of the conducted searches. It presents the selected sources, used keywords and amount of obtained results. Not all keywords we searched are presented and we focused on the ones we received the best results for. Most of the selected sources were already obtained from the search through the literature we were given, because research groups in the field are highly interconnected.

Table 3.1: Sample Results of the Literature Search

Keywords	**Source**	**Results**	**Selected**
Information Visualization, Cohort	**Google Scholar**	121.000	[44] [47] [69] [15]
	PubMed	36	-
	Springer Link	7.622	[44] [45]
	AIiM	9	-
Information Visualization, Medical	**Google Scholar**	1.200.000	[44] [56] [14] [10] [42] [24] [15]
	PubMed	1428	[37] [67] [53] [27] [39]
	Springer Link	55.127	[44] [45]
	AIiM	162	[27] [8] [59] [15]
Information Visualization, Temporal	**Google Scholar**	880.000	[14] [24] [10] [8] [65] [42] [70] [9] [53]
	PubMed	367	[66] [41] [60]
	Springer Link	37.958	[9]
	AIiM	84	[60] [15]
Information Visualization, Clinical	**Google Scholar**	885.000	[42] [53] [54] [52] [10] [9] [8] [24] [15]
	PubMed	367	[41] [66] [60]
	Springer Link	48.495	[9]
	AIiM	144	[60] [53] [8] [15]

Selection Criteria

The main criteria for the selection of sources were, that they contain at least one of the following concepts:

- Time-Oriented information visualization
- Visualization of data in the medical/clinical domain
- Aggregation and temporal abstraction techniques for parameters
- Techniques for handling and visualizing data for cohorts
- Analysis and visualization of compliance to clinical guidelines

We excluded papers which focus on scientific visualization of medical data, like the rendering of spatial data from medical images of different modalities (Computed Tomography (CT), Magnetic Resonance Imaging (MRI)) and focused on visualization of data without spatial components.

If we found sources for systems, which are used in multiple domains, we only focused on the papers describing the applications for medicine. Some systems and concepts were described in more than one source, because they were improved over time. In such cases, we focused only on the latest versions presented and neglected older papers.

3.3 Related Work

This section describes the selected approaches for the given problems we found during our search. We start out by giving an overview on guideline visualization methods. The following section continues by giving an overview of compliance analysis and the visualization of the computed results. The last two subsections contain techniques for the temporal abstraction of parameter measurements, as well as a survey of relevant existing systems, which contain techniques for the visualization of multiple patients or cohorts.

Guideline Visualization

We were able to find a variety of different approaches to guideline visualization. We describe most of them briefly and present details about the views we want to extend for the use with cohorts in our thesis.

PlanStrips were introduced by Seyfang et. al in [49]. *PlanStrips* focus on visualizing the the hierarchy of Asbru plans as a set of nested strips and preserving the timely ordering of actions in the treatment plan. Synchronizations with other sub-plans are visualized, for example if they are to be executed in serial, parallel, cyclical or alternative to each other and also if the plan is not synchronized with any other plan at all. To show the hierarchical relation of plans, child plans are stacked on top of their parent plans with an inset. Color is used to represent the kind of synchronization of the plans. Selected plans are highlighted by an increased saturation and brightness. The general concept for the layout of the visualization of PlanStrips is shown in figure 3.1. A study with 4 computer scientists was conducted, which showed, that even users which were not previously acquainted with clinical guidelines were able to grasp and use the prototype in a short amount of time. An interview was conducted after an analysis task of dependencies in the plan, where all users positively answered to questions about the usability of the prototype.

Glare [62] stands for Guideline Acquisition, Representation and Execution. A physician is able to supply and develop the guideline with the support of syntactic and semantic tests which verify the “well-formedness “of the guideline. [62]. As figure 3.3 shows, Glare uses a graph-like representation for the guideline where nodes represent atomic actions and edges represent the control flow between them. Different types of actions are encoded by different symbols and col-

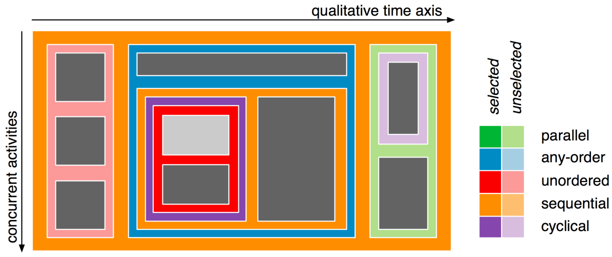

Figure 3.1: Concept for the visualization of plan hierarchies in *PlanStrips* [49].

ors. The tree on the left in figure 3.3 shows the hierarchy of the selected plans, while sub-plans can be extended and reduced. Glare is not just a simple visualization and is also able to execute the guidelines, provide decision support for the physician by conducting a what-if analysis based on the current state of the patient and also temporal reasoning in both, acquisition and execution.

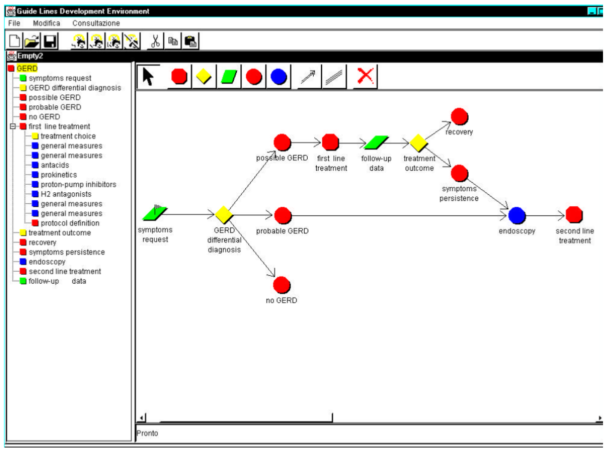

Figure 3.2: *Glare*: [62].

AsbruView was introduced by Kosara et. al. in [28] as part of the Asgaard [50] project and employs a different approach to visualize Asbru Guidelines. The intention is to provide access to the complex plan format for physician and give an overview of the plan hierarchy. The guidelines are visualized as 3D diagrams, while width, height and depth dimensions each represent different information. While time is mapped to the horizontal axis of the visualization, sub-plans are stacked above their parent plans and parallel plans are shown next to each other. Visual metaphors are used to provide intuitive understandings of visualizations for plans. Running Tracks represent different plans, while Traffic Signs and other symbols from the domain of traffic control are used to represent different concepts, like a red light for an abort condition, or a finishing flag for the complete condition. The interface provides interaction techniques like a scroll bar, which enables the user to move through the hierarchy of the plan while the interface shows the currently displayed plan levels.

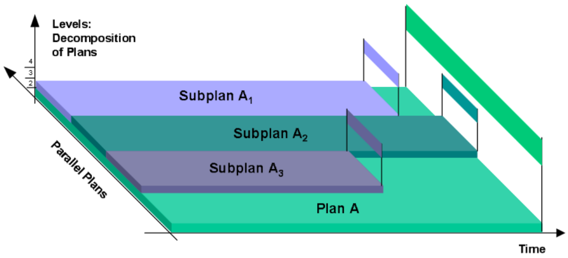

Figure 3.3: *AsbruView*: [28].

We devote the largest part of this chapter to describe the guideline visualization technique and view used in the *CareCruiser* prototype [23]. This specific interface was also implemented to visualize plans represented in the Asbru [50] plan language, as developed in the Asgaard project [55]. The first version of the logical view for guidelines was already included in *CareVis* [8]. The hierarchical view was added in the *CareCruiser* prototype [23] and Bodesinsky further extended this view in his master thesis [11], by adding additional techniques which show statistical information of guideline compliance. As described above, these techniques have lately been migrated to the *VisuExplore* [46] prototype. Figure 3.4 shows the latest version of both views.

The left part of figure 3.4 shows the logical detail view of a selected plan. It visualizes from top to bottom: Path from the highest level of the plan to the sub-plan, name of the plan, abort conditions for the plan in the red strip on the top ($FiO2 > 90$, $PIP > 25$ or $PaCO2 > 100$), the intentions of the plan ($90 \leq tcSO2$ AND $tcSO2 \leq 92$) and the conditions and the actions needed to be conducted if the condition is applicable in sequential order.

On the right hand side the hierarchical view of the whole guideline is shown in the form of a tree. Grey bars represent the whole plan as well as the sub-plans of it. The colored diamonds, at the leafs of the tree, represent elemental actions of the above plan (each type of action represented by a different color).

The logical view is connected to this view by linking and brushing. When the user selects a sub-plan, the borders are colored in red and the sub-plan is shown in the logical view.
If the mouse is moved over a plan element (action or plan) a tool-tip appears, which shows aggregated counts of guideline compliance information (valid, invalid and missing actions) for the patient, in the form of a stacked bar. Compliance information of actions is also shown in a

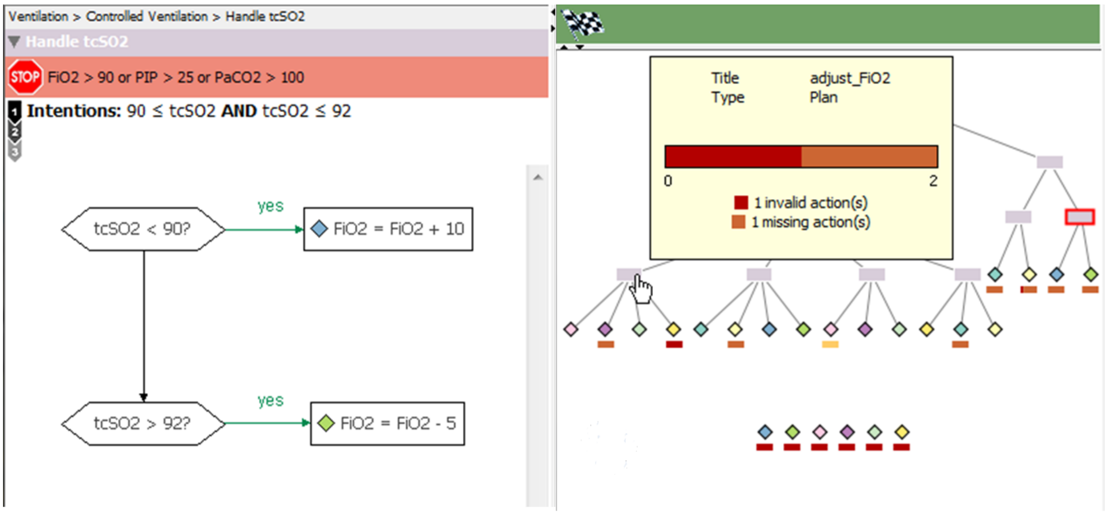

Figure 3.4: The latest guideline visualization prototype as developed by Bodesinsky in [11].

smaller version below the diamond representing them.

Compliance Analysis

Compliance analysis has the objective of computing the compliance to a given guideline for a conducted treatment. This application is also referred to as critiquing or plan-based critiquing in the literature [22]. Such approaches try to identify differences between the conducted treatment and the recommendation provided by the guideline and provide feedback or critique for the attending physician or other health care specialists.

Compliance to a guideline can however be interpreted on different levels of granularity and might also differ between specific applications. While in this thesis, we use an approach which analyzed compliance on the level of atomic actions, other approaches assess compliance up to the level of the goals and intentions of the treatment.

As part of the Asgaard project [4], Advani et. al [5] developed a methodology and tool for the critiquing of guideline-oriented medical care. They describe, how the care given to a patient during a treatment according to a plan in the Asbru language, can be retrospectively compared up to the level of the intentions of the plan. Next to the patient states recorded during the treatment and the recommendations and specifications in the guideline, the recommendations and interventions of the physician are also taken into account. The comparisons of these three input sources are also made on 6 different levels, which reach from the fine grained analysis of recorded actions to an analysis based on an abstracted state of the patient and the intentions of the physician. The system is therefore able to adjust its critiquing of the conducted actions in terms of the overall goal of the treatment, to allow physicians to react to occurring variations of the patient state or conditions and allow him the flexibility of treating the patient according to his experience. If the physician deviates from the treatment recommended by applying a non-recommended action, the system is able to consider the effects of the action on the overall goal of the plan. Actions not recommended in the plan can be computed as compliant with the overall intentions of the guideline and receive positive critiquing.

In [22] Groot et. al. propose a model checking as well as a satisfaction set approach for the analysis of compliance. The guideline is used as input providing a state transition system or solution for the model checking algorithm. The symptoms of the patient represent the problem the algorithm should solve. Groot et. al. used a patient description (symptoms, test outcomes measured) and a treatment (actions performed by the practitioner) description as inputs for the algorithm. They categorize non-compliance of actions in a fine grained manner, while they also describe the concept of non-compliant findings. The 'T' type is used to describe differences between the treatment suggested by the guideline and the actual treatment and is used to find non-compliant actions. On the other hand the 'F' type considers the required patient findings, clinical or laboratory-based for choosing a specific action [22]. It is used to critique decisions or actions which are assumed to be compliant with the guideline but can be non-compliant with findings in the patient parameters. For example, if a patient has been treated in some manner, but the choice of the treatment is not supported by any findings in the patient parameters.

Bodesinsky [11] used the compliance analysis method we described in section 2.7 Action Compliance, which he describes as partially inspired by the work conducted by Chesani et. al in [22]. He stated that he used a reduced form of the method, since he focused mainly on the visualization of compliance and not the analysis task. Chesani et. al. describe Guideline PRocess cOnformance VErification (*GPROVE*) which is a set of tools for the specification and a-posterior verification of care-flow process executions. Guidelines in *GPROVE* are modeled in the graphical language called *GOSpeL* and translated to a formal language called *SCIFF*. A *SCIFF* proof procedure is used to check for violations to the modeled guideline in a given event log and report the flow of events. The authors describe two forms of non-compliance (non-compliant action ordering and non-compliant actions which are actions that can not be prescribed at all for the given patient) and state that due the fact that *GPROVE* uses temporal logic it is naturally able to support temporal constraints in guidelines.

Visualization of Guideline Compliance Over Time

Research for visualizing guideline compliance is rare in the literature and we were not able to find any other approaches than the ones developed by Bodesinsky in his master thesis [11]. In this section we briefly describe his approaches for the time-oriented views, while we already described compliance information in the hierarchical guideline view in section 3.3.

Figure 3.5 shows the different encoding of action instances in the plan execution view. An atomic action is marked as a diamond, while the color encodes the type of the action. Element (a) shows a valid action, while invalid actions are marked by an X within the diamond. Elements

(c) and (d) show valid and invalid actions with a time-span, indicated by the attached horizontal bars. The bar in sub-figure (e) shows the whole duration, in which an action should have been applied, but was not. In (f) an invalid action, which occurs during the interval of a missing action, is visualized.

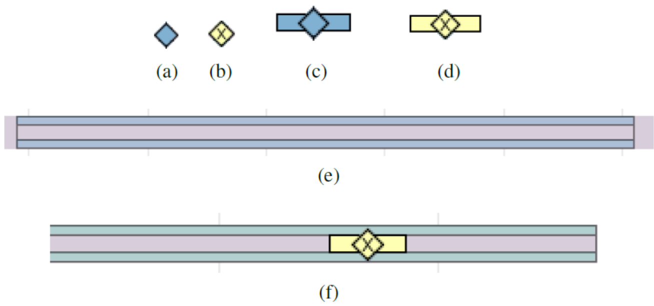

Figure 3.5: The encoding for visualizing different action instances as introduced by Bodesinsky in [11].

In figure 3.6, an example instance of a compliance visualization to the plan presented in figure 3.4 is shown. The information for the two sub-plans are shown separately in (1) and (2) in this case. The large strip for the upper plan indicates, that there was a missing action for the whole time-span the view covers. The circle with the number 3 points to a bar, which shows a missing interval. The interval is ended by the execution of the action represented by the yellow diamond. Actions which are not part of the guideline, but were conducted in the treatment process, are marked by diamonds containing an X (6).

Bodesinsky also did a small study with a single medical expert. He first introduced the expert to the problem and tasks he should solve. He used the Thinking Aloud method [31] in the process of solving the tasks. The expert was recorded on audio and video and the study finished with a final interview. Bodesinsky concludes that overall the developed techniques were intuitive to use and able to aid the user in accomplishing the described task. He pointed out that minor issues were found during the evaluation.

Techniques for Temporal Abstraction of Clinical Parameters

As it will not be possible to present all value points for all patients of a cohort on the screen without presenting the user with visual clutter, methods for temporal parameter abstraction present

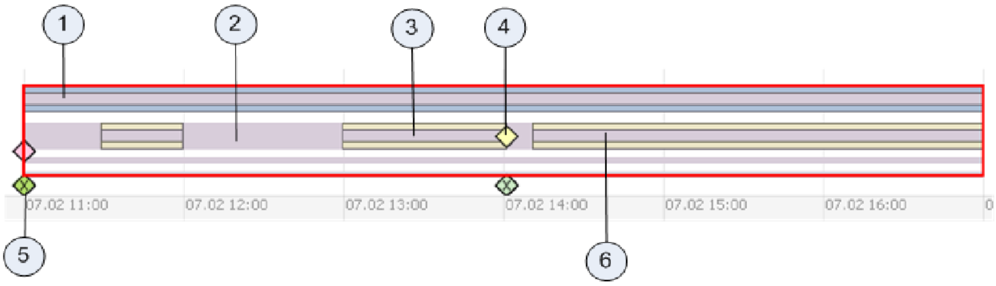

Figure 3.6: Example for the time-oriented visualization of compliance information, developed by Bodesinsky in [11].

promising approaches for reducing the screen spaced used. Such methods have been introduced and described by Shahar in [51]. The intention is to provide context-sensitive abstractions and interpretations of higher level concepts from time-stamped data. In his article, Shahar specifically mentions the application to the concept for monitoring therapy plans during execution and creating high-level summaries of medical records, which are related problems for our thesis.

This can for example be reducing a quantitative parameters values over time to qualitative ranges which could show for example interval-based development of the clinical parameter. A parameters range could be split up in certain qualitative ranges like very low, low, medium, high, very high and also other factors like the change to the previous value (improvement or worsening) can be of interest which can be represented in a different manner than the absolute data value.

In this section we give an overview of the methods we found in our literature search and short assessments of their usefulness:

In [10], Bade et. al. present several time-oriented approaches for the representation of a fever curve: Initially the range of the body temperature is split up into 3 distinct qualitative groups: *Normal* [< 38.5], *Elevated* [> 38.5, <41.0] and *Highly Elevated* [> 41.0].

Figure 3.7 shows a visualization if the parameter in the form of height adjusted bars which present the progression over time. A Normal temperature level is indicated by the small bar, an Elevated level is indicated by a thicker bar and a small plus sign and Highly elevated temperature by the thickest bar and a larger plus sign.

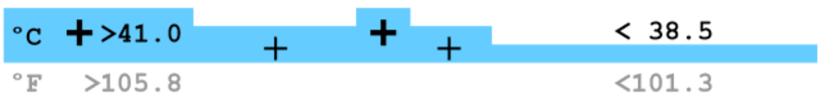

Figure 3.7: Height coded representation of a fever curve presented by Bade et. al. in [10].

Figure 3.8: Color coded representation of a fever curve presented by Bade et. al. in [10].

In figure 3.8 the same groups are used, but they are mapped to color. Green is used for normal, yellow for elevated and red for highly elevated. In both approaches the quantitative absolute values are also presented in the visualization. The user is able to get an overview of the fever development and at a second glance she can see the exact value behind the abstraction.

Both mappings of qualitative ranges for fever are able to convey the qualitative information of the patients fever to the viewer, however the colored fever curve seems to be easier to grasp in our opinion. The representation in the height has the disadvantage of using variable sreenspace. We can easily stack the colored representation for multiple patients when using it in a cohort study. The height mapping on the other hand, is far worse to read when stacked multiple times.

In figure 3.9, the technique is extended by showing the quantitative curve of a parameter (in this case the blood pressure value for multiple patients) and coloring the areas below the curve according to qualitative criteria like before. All curves are stacked and shown in juxtaposition which uses up less space.

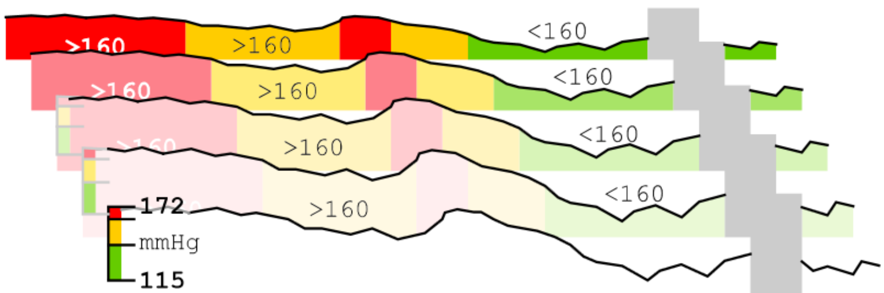

Figure 3.9: Example for a Continous Whiskers Plot presented by Bade et. al. in [10].

This technique has the advantage of shown all the present information, while it may however be possible that if the visualization is badly calibrated, that features occlude each other (very low value hidden by a very high value). For a large number of patients the visualization will still use a lot of screen space and the tendency to visual clutter is high.

In [23] Gschwandtner et. al present several techniques where they use color for emphasizing different developments in quantitative values. In all methods, the range of the parameter is indicated by the dark horizontal lines.

Figure 3.10, accentuates the difference of a measured clinical parameter to an intended value by underlying the dot plotted parameter curve over time with color. Different saturations of magenta are used in this case. The higher the saturation, the greater the distance from a defined intended value.

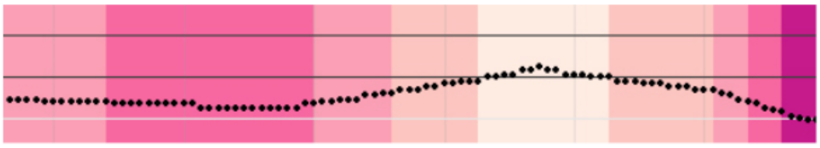

Figure 3.10: Color determined by the distance to an intended value [23].

Figure 3.11 shows a colored encoding, where the color gives insight into the qualitative progression of the treatment. The initial value of the parameter is marked by a white background. When the parameter development is bad in terms of the health status of the patient, the background will change from white into a red hue. A positive progression is indicated by blue colors.

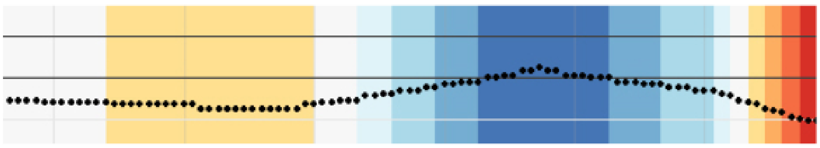

Figure 3.11: Color determined by the qualitative progress of the treatment [23].

The last figure 3.12, highlights the slope of the visualized parameter. Turquoise indicates a drop of the value, while brown indicates a rise. The saturation of the color represents the magnitude of the change, with the value determined by the mean of 7 data points for robustness.

All of the above methods are able to give physicians insight about the impact of the treatment process on the development of the visualized parameter at a quick glance. Generally the colors can be exchanged, but in the progress and slope methods complementary colors should be used. The omission of the dot plot and horizontal lines for the parameter, leaves the color-coded information and again would enable us to reduce the amount of screen space used per patient. Stacking and ordering such a set of colored bars, might enable use to convey general trends of

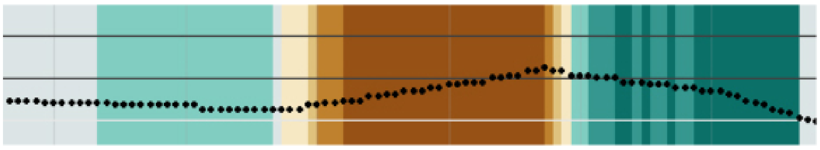

Figure 3.12: Color determined by direction and magnitude of change [23].

the effects of a treatment on a cohort.

In [21] Federico et. al. developed a technique called Qualizon Graphs, which also combine quantitative data with qualitative abstractions. Qualizon Graphs are based on Horizon Graphs described in [43]. The range of the parameter is split up into an even number of equally sized (according to parameter range) bands, along the middle of the parameters range, the so called horizon. The different bands are differently colored, with a divergent color scheme, which also skips at the horizon. Afterwards the lower half of the graph is mirrored along the x-axis, bringing the lower bands next to the upper bands and thereby reducing the allocated screen space by half, while containing the information. In the last step, the bands are collapsed on the x-axis with a technique called two-tone pseudo coloring, again reducing the needed screen space, while keeping the information in the color. A restriction of Horizon Graphs is however the number of bands used. The reference shows 6 bands, but the higher the number, the more time the user generally needs to grasp the information.

In the case of Qualizon bands, the parameter range is still split up in bands, while the technique looses the restrictions for equal size and also an odd number of bands can be present. The are as many bands as qualitative categories with the corresponding different sizes. If the number of bands is even and the parameter range is split up equally the Qualizon Graph is indistinguishable from a Horizon Graph of the same data series [21].

Federico et. al. [21] also conducted a user study, which showed, that Qualizon Graphs are equally fast and accurate for quantitative data and offer an alternative to other visualization techniques, while offering a trade-off between speed and accuracy.

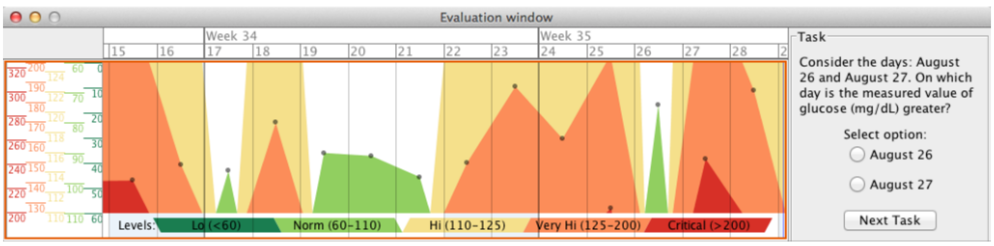

Figure 3.13: The prototype used for the evaluation of Qualizon Graphs by Federico et. al. in [21].

Selected Systems with Visualization Techniques for Cohorts

TimeRider - Visualization of Correlations with Animation

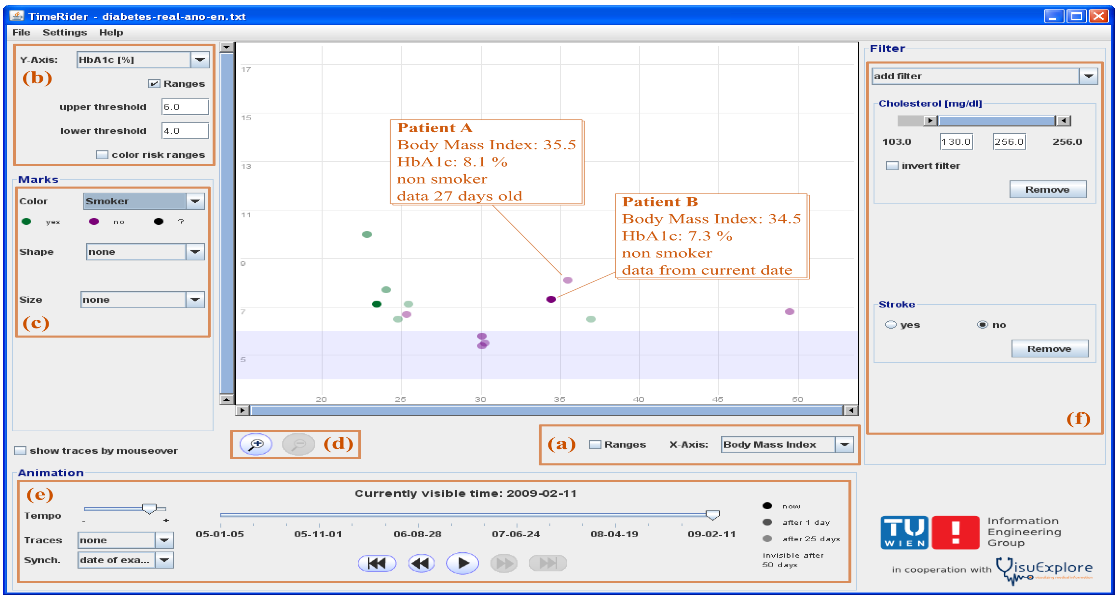

Figure 3.14: The TimeRider Interface as presented by Rind et. al in [44].

In [44], Rind et. al. approached the visual analysis of trends and patterns in cohorts by the use of animated scatter-plots and visualizing the development of the variables over time and implemented the *TimeRider* prototype. Two selectable variables of the data are mapped to the x- (a) and y- (b) axis, while the user is able to control time. The temporal component of the data is mapped to animation, while data is split in frames of single days and the user is able to navigate through time. A data point for a single patient at a given point in time, is represented as mark in the Cartesian coordinate system in the corresponding time-frame.

Since clinical data might not be provided on a daily basis, Rind et. al. also used concepts named transparency and traces. In transparency mode marks fade out more and more, as the time in the animation, fades further away from the time the data point was collected, by reducing the opacity of the mark. In trace mode, the position of missing marks is interpolated between two actual data points. At each frame the patient mark is drawn on a linear trajectory between the previous and next known position.

Interaction with time is possible in multiple ways. The user can hit the play button (e) to start the animation and watch the events unfold, as the interfaces animates the collected data over time. She can also steer the animation by back and forward buttons (e) or manually dragging the slider.

Since patient data are not collected at the same absolute point in time, TimeRider enables temporal alignment or synchronization of all patients data sets, by calendar date, patient age, start of treatment or end of treatment.

Panel (f) provides controls in which the user is enabled to add filters for neglecting patients with certain characteristics and conduct their investigation only on relevant subsets of the cohort.

To evaluate the method, a small study with ten physicians was conducted. At first the participants were introduced to the domain and software. Afterwards they had to solve four tasks while thinking out aloud, while the screen and the voice were captured. In the end interviews about user experiences were conducted and also recorded.

The results of the study showed that many participants had problems with the user interface and the usability, but were able to identify trends in the data which was not the case in similar systems. This led the authors to the conclusion that the visualization is generally able to support physicians in finding patterns.

Lifelines2

LIFELINES2 [65] is an interactive visualization system designed to search and explore event sequences in multiple records of temporal categorical data. The records of multiple patients are stacked vertically on a shared horizontal time-line as annotated by (a) in figure 3.15. Each record is able to show a number of categorical variables, which are represented as triangles, color coded according to category, and horizontally aligned if they are of the same instance.

On the right panel (c), the user is able to vertically align all records by a specific event present in the records. In figure 3.15 the first occurrence of radiology contrast is used to show this functionality. Records which do not contain the selected event are filtered out. The time scale also switches from an absolute calendar scale to a relative scale showing with the selected event as reference point. The alignment enables the user to search for patterns which might occur commonly after a specific event.

The interface also includes rank and filter functionality. Records can be ranked for instance by the number of occurrences of a specific event in the record. Furthermore they can be filtered by specific criteria, like temporal patterns and sequences of event occurrences including the absence of specific events.

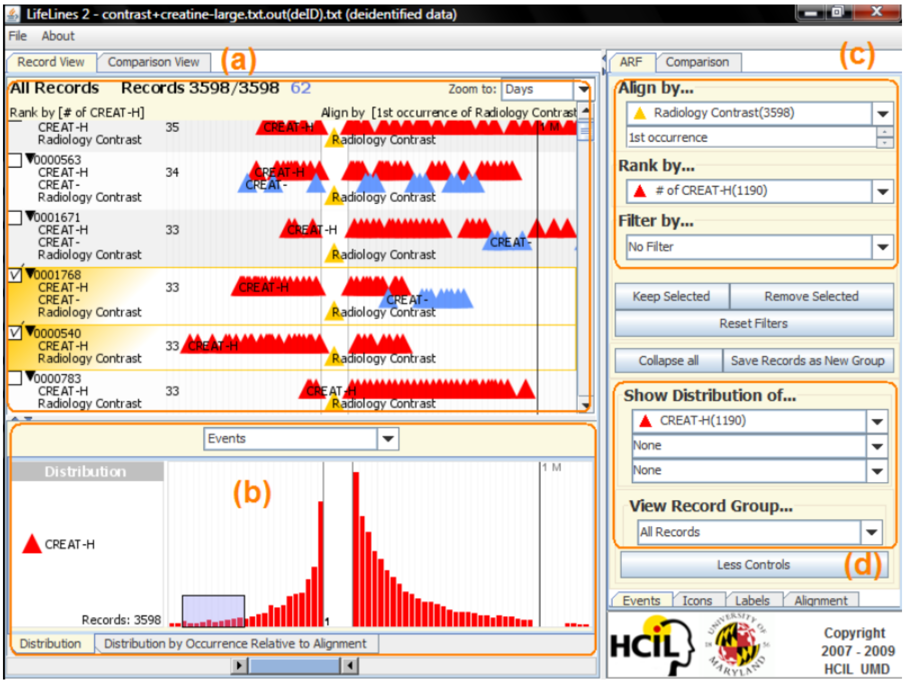

Figure 3.15: The LIFELINES2 Interface [65].

In the bottom left panel (b), a temporal histogram is shown which shows the the temporal distribution of a specific event in the aligned records. This visualization is also interactive and enables the selection of a time span. Records which do not contain the selected event in this time-span are filtered out, and LIFELINES2 allows to save the results of such an interactive query as new group of patients. Afterwards subsequent filtering can be applied to explore subgroups of patients.

Outflow

Outflow [69] combines the information from multiple patient records into a graph-based temporal event sequence visualization, which is analogous to a state-diagram.

To aggregate the data, the user first selects an alignment point. This point could, for example, be a state where all patients have the same exact symptoms. Starting from this point an Outflow graph is constructed. A state in the Outflow graph represents a unique combination of symptoms a patient shows and is annotated by a rectangle with a height proportional to the cardinality of patients. The graph is vertically sliced into layers, where the number of the layer equals the number of symptoms for the patients in the cohort. The leftmost layer represents the patients state with no symptoms and therefore always contains only one single node. Colors in the graph

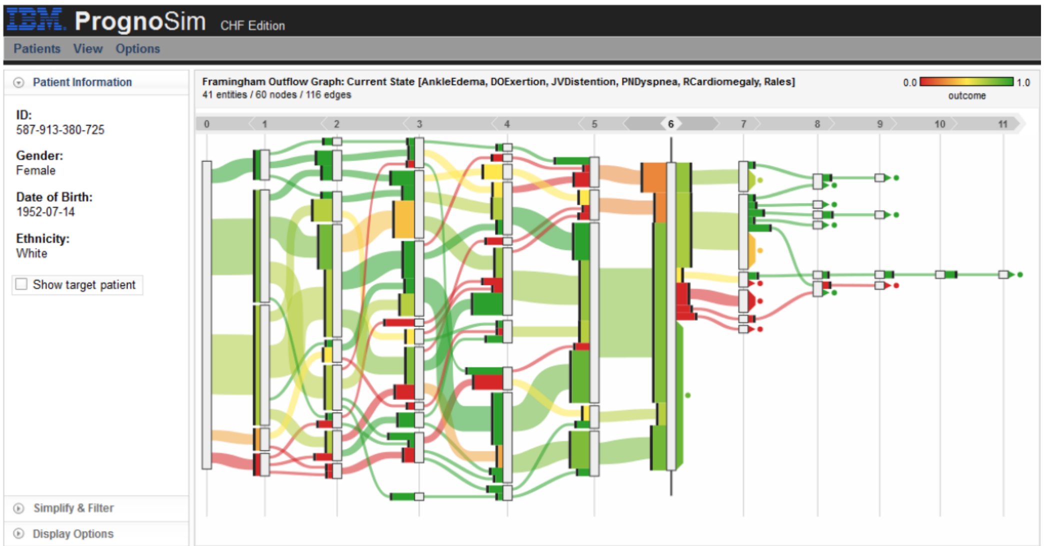

Figure 3.16: The Outflow interface as introduced by Wongsuphasawat et. al. in [69].

represent the average outcome in terms of health, with greener colors representing the best and red colors representing the worst possible outcomes.

Edges convey the transitions between different sets of symptoms. They are annotated with the number of patients with the same transition, average time gap and outcome of the patient groups between the state.

The following interaction techniques are supported: Panning & Zooming (uncover more detail about the graphs structure), Filtering (filter nodes and edges based on the number of patients), Symptom Selection (select symptom types which are used to construct the graph), Brushing (hovering the mouse over a node highlights all paths passing through the corresponding point), Tool-tips (hovering over a node shows more detail about the condition in a tool-tip).

The graph captures all event patients event paths which led to the selected alignment point and all events that occur afterwards, enabling the user to select a single patient and exploring the most probable outcomes for the selected state.

The paper does not include any study with individuals, but finishes with a preliminary analysis of the visualization. The authors accentuate, that the example in figure 3.16 shows a *Leading Indicator*. The bottommost transition is strongly red and therefore indicates that if a patient undergoes this transition, the outcome is likely to be bad. They conclude that the presented method is a promising approach for giving a prediction for a single patient reaching a specific set of symptoms.

VISITORS

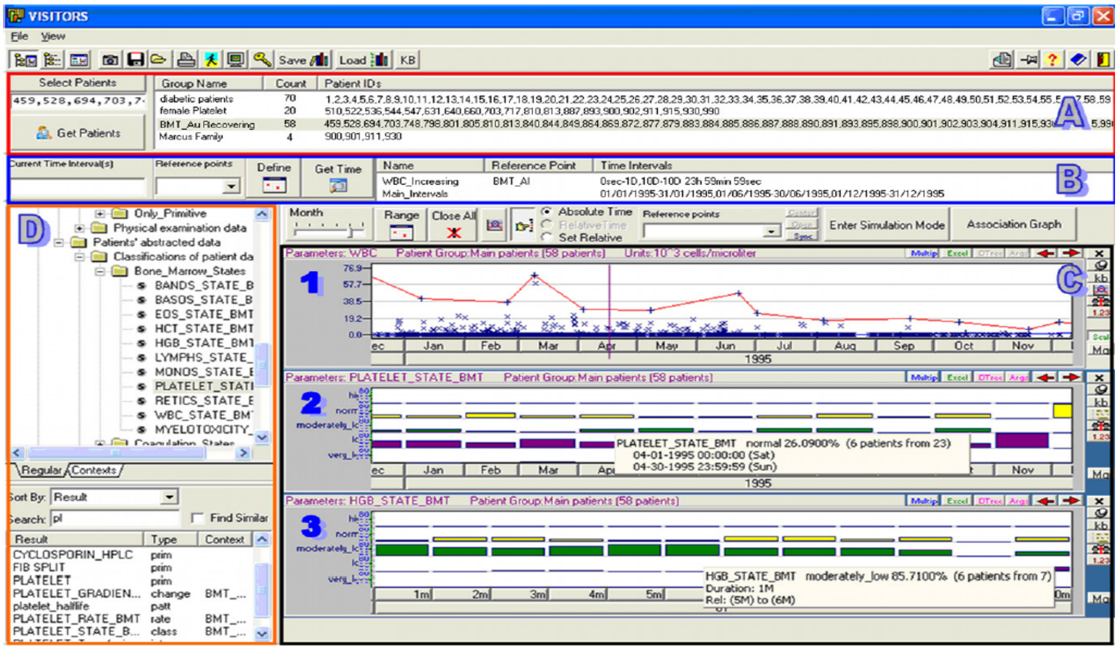

Figure 3.17: The VISITORS Interface [27].

The *VISITORS* system by Klimov et. al. [27] combines a clinical knowledge base with visualizations, to enable the user to explore the data of multiple cohorts. Domain ontologies are used, to define higher level abstractions of given raw temporal data. Abstractions of numerical values to ordinal scales are derived and aggregated, to show the evolution of parameters for larger groups of patient.

The interface shown in figure 3.17 shows a number of 58 selected patients. In panel (A) the user is able to select specific groups of patients, which are included in the aggregation of parameters. Panel (B) enables the setting of filter parameters, for example specifying absolute time intervals or intervals, in which specific parameter changes take place. In (C), the evolution of selected parameters is shown in an aggregated form for the whole group in dot plots (top visualization) and the lower panels show parameters in higher abstraction in the form of bars. The height of the bars shows the aggregated number of patients whose abstracted parameter values fall within in the ordinal abstraction ranges. On the left of the screen (D) medical ontologies are listed, which encode the abstractions for the parameters shown in (C).

An expressive query language is offered, which allows the user to search for both, raw clinical parameter values as well as aggregated data in the group of patients. Queries can be specified for non temporal meta-data (gender or race of a patient), time (relative as well as absolute) and value constraints (e.g parameter is low, normal or elevated).

Caregiver

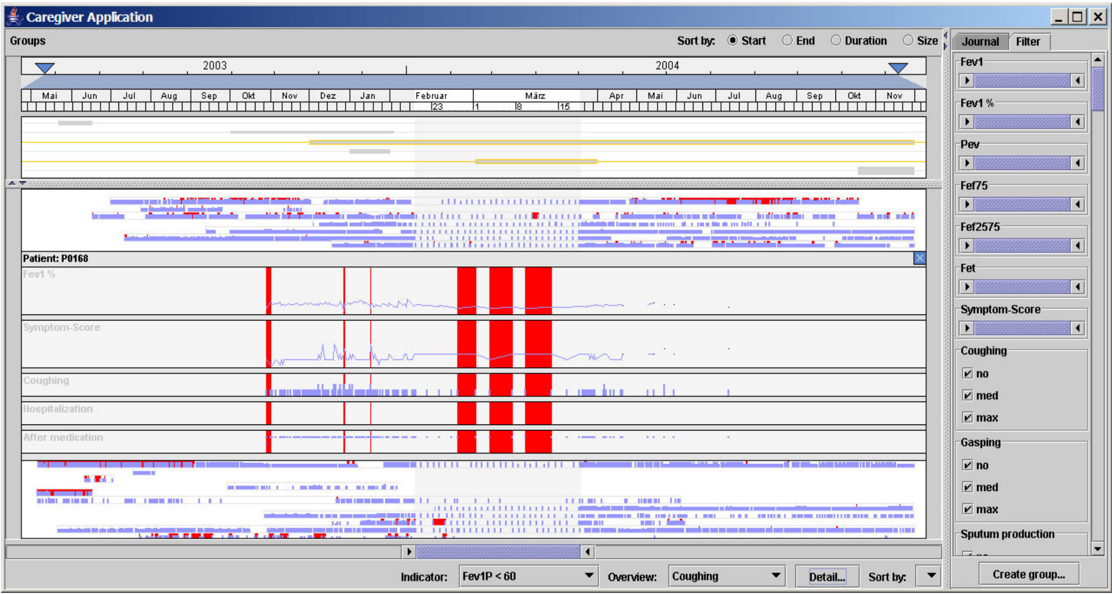

Figure 3.18: The Caregiver Interface [13].

In [13] Brodbeck et. al. introduced *Caregiver*, with the goal of supporting therapeutic decision making. *Caregiver* has three main visualizations, which are all placed on same horizontal time line. The interface is presented in figure 3.18.

The top panel of the interface visualizes multiple cohorts of patients, which are represented by solid rectangles on the time-line. The length of these rectangles indicates the duration over which treatment data has been collected. The height correlates with the number of patients in the cohort. In the middle panel, detailed data about the evolution of a single patients' clinical parameters is visualized in the form of bar charts and line plots. On the bottom an overview of the treatment data of all patients in all cohorts is shown, where each patients data is visualized on a small horizontal strip of space. The evolution of a single chosen variable is displayed for the treatment process of each patient, in the form of bar charts.

A brushing and linking approach is used, to interactively connect the three views. When the user selects one of the cohorts in the top panel, all the patients of the selected cohort are highlighted in the bottom panel and vice versa. If a patient is selected in the bottom, his detailed information about his treatment process is shown in the middle. Also a focus and context technique was implemented, for showing a selected time period in more detail while other periods are shown in a compressed form.

The right side of the interface features controls, that can be used to query for patients which depict the selected status. A downside is that the interface can not be used to search for certain developments of patients' parameters. Last but not least, an indicator can be defined, which highlights the temporal regions showing the specified characteristics (designated by the red regions in 3.18).

3.4 Preceeding Work

This section describes two prototypes especially relevant to our thesis, since they contain most of the state of the art visualization techniques we want to extend and adapt for our thesis and are also open for enhancements.

VisuExplore

VisuExplore [46] is a prototype for interactive visual exploration of medical data. It was designed to fit the following key requirements: simple user interface, flexible for various medical parameters, time-oriented data, Multiple patients, Interactivity. Figure 3.19 shows the time-aligned visualization of different clinical parameters for a diabetes patient.

Different visualization techniques can be selected for each parameter. The example includes: event chart, line plot, bar chart, and timeline chart. Parameters can be added and removed on the fly. Also several interaction techniques are offered, like resizing the parameter view. Interactions with the time dimension are also possible, by dragging the mouse, or resizing the time windows by zooming. The software architecture of the prototypes aims for easy integration of newly developed techniques and is easily extendable.

CareVis and CareCruiser

In [8] Miksch and Aigner introduced *CareVis*, which was developed with the goal of combining views for patient observations, treatment data and CIGs and is based on the Prefuse prototype [3].

Gschwandtner et. al. [23] developed extensions to *CareVis* and renamed the new prototype to *CareCruiser*. *CareCruiser* supports a step-wise interactive exploration of a single patient's condition and the effects of applied treatment guidelines.

In particular the following features are included.

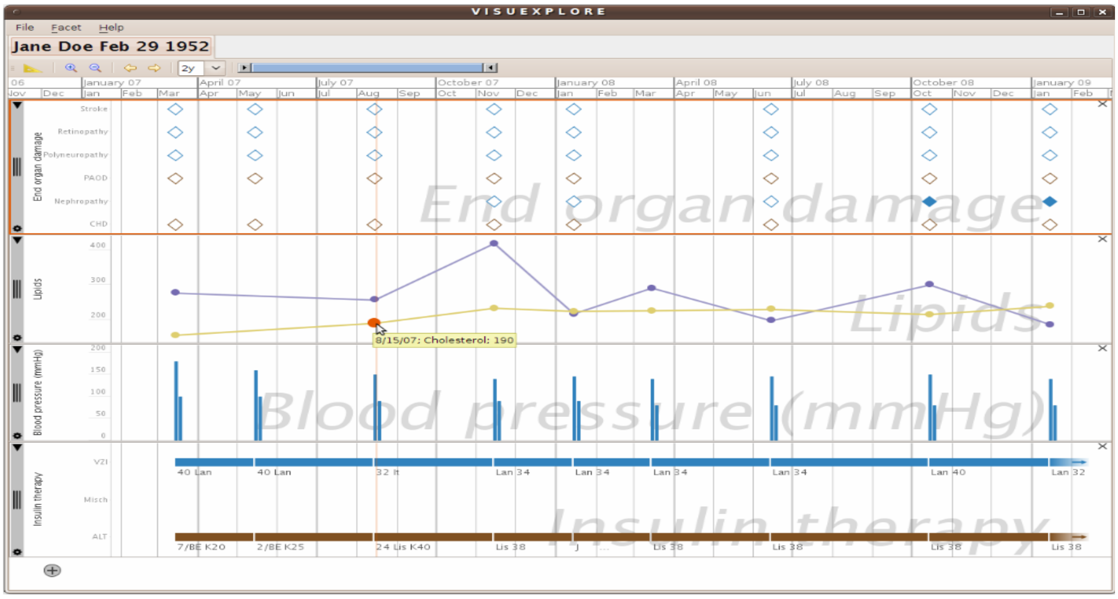

Figure 3.19: The evolution of different types of parameters over time in the *VisuExplore* interface [46].

- Visualizing patient data in combination with applied treatment plans and clinical actions
- Aligning treatment plans and clinical actions for comparison
- Color-coded highlighting of interesting events of the patient's parameter values development
- Filtering of highlighted events
- Providing focus and context information to support the detection of effects' patterns

In his master thesis, Bodesinsky [11] further expanded *CareCruiser* especially focusing on integrating the visualization of compliance information for an executed treatment plan.

The main components of the extended *CareCruiser* [23] interface (figure 3.20) are:

Panel (a) shows a time-oriented visualization of the temporal development of clinical parameters relevant for the treatment guideline. The area between the two thick lines for each parameter visualizes the target range of the clinical parameter observed. Also the maximum and minimum of the plotted parameter is shown by the two outer thin lines. The whole time spans in which the measurements of the clinical parameters are recorded can be highlighted by using the techniques described in [23], we presented in section 3.3, which can be adjusted by the controls above the temporal view. Panel (b) shows the modified plan execution view for the treatment plans of both parameters, which are vertically stacked. A combination of automated data analysis and information visualization has been implemented as prototype by Bodesinsky et. al described in [11] and [12]. In his approach he first precomputes and then visualizes the adherence of a treatment process to a plan, on a level of single clinical actions. We already presented the compliance view in detail in section 3.3. Panel (c) contains general information about the observed patient, statistical information about action compliance and controls for altering options for the visualization. In (d) the logical and hierarchical views of the used guideline as describe in section 3.3 are shown.

The *CareCruiser* interface offers manifold possibilities for interaction with the data: When selecting an action in the temporal view, all actions of the same type are also highlighted in the plan execution view. Filtering is supported by the range slider over the temporal view, which can be used to restrict the highlighting of parameters to specific ranges. Finally focus and Context is provided by a focus lens which can be dragged over the temporal view so highlighting is only conducted inside within the borders of the lens.

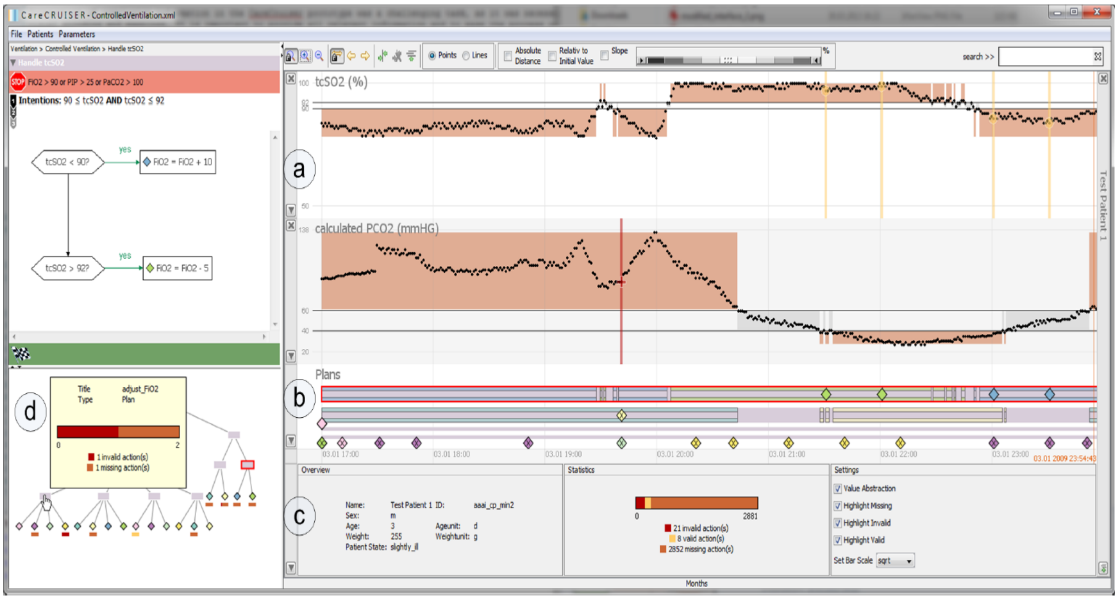

Figure 3.20: The Interface of the latest *CareCruiser* prototype as presented by Bodesinsky in [11].

3.5 Discussion of the Presented Systems and Techniques

In this section we summarize and discuss the features and characteristics of the systems presented in the previous section. The techniques and approaches are compared according to different criteria and to which degree they are able to solve the given problem described in section 1.3.

The guideline visualization technique which includes information of compliance information, described in section 3.3, is very useful for visualizing the data of cohorts in its current form. The advantage to AsbruView, presented by Shahar et. al in [28], is that it is a more compact representation. It offers the same functionality through interaction and navigation and integrates compliance information into the plan views. Also the extension to cohorts seems relatively easy. Aggregations of valid invalid and missing actions can be shown by the same technique used for single patients.

The compliance visualization technique developed by Bodesinsky [11], as presented in section 3.3, is the only time-oriented compliance visualization technique we found in the literature. It offers useful capabilities for visualizing compliance information over time for a single patient. Extending the technique for the visualization of compliance information might be difficult. Overlapping intervals and actions on a single strip, will most likely result in visual clutter. Aggregating the information might yield meaningless and generic results. In fact this seems to be one of the more difficult tasks in the thesis.

In section 3.3 we presented approaches for temporal abstraction of parameter measurements. Especially approaches using color and a non-variable amount of screen space should prove quite useful in conveying information about trends in a cohort and also single patients.

In section 3.3 we provided an overview of systems, which show approaches to problems similar to ours:

TimeRider [44] approaches the visualization of cohort data by animated scatter-plots. Mapping the time dimension to time instead of space leaves another dimension for visualizing data, however visualization of data by animation yields the perceptual effect of change blindness. Also the system shows only the development of parameters without any compliance or guideline information.
LIFELINES2 [65] offers great capabilities for filtering, aligning and searching for relevant patients. However the only aggregation of data takes place for the temporal histogram and *LIFELINES2* only supports point-based events and no intervals (e.g. ongoing treatments). Also no compliance information is offered, since it does not offer any features for visualizing Computer Interpretable Guidelines.
Outflow [69] offers great capabilities in showing outcomes for certain symptoms a patient shows. The visualization gives a good idea about the most probable outcomes and is easily interpretable. The disadvantages are that time between symptom transitions of patients is averaged and no idea is given about how this time can vary. Also the system covers only outcomes that resulted from

symptoms a patient showed and does not include information about the quality of the treatment which lead to the transitions between those symptoms. Developing ideas about how to include such options (e.g. mapping the treatment which took place in in the transition to visual attributes) is a promising idea.
VISITORS [27] uses a promising approach to abstract and aggregate the data for evolution of raw clinical parameters of a group of patients. Especially the aggregated line charts and bar plots in panel (C) represent suitable techniques, while the visualized treatment processes of the patients are not aligned in any way. No information about the guideline compliance of treatment actions is provided by the visualizations in this system.
Caregiver [13] also presents promising aggregation and interaction techniques, while it also does not align the the treatment processes. Also no features regarding guideline compliance are included.
VisuExplore [46] is only able to show a multitude of parameters for a single patient.
CareCruiser [23] only supports the visualization of raw parameters and compliance information for one ore a small number of patients. In figure 3.20, the whole screen space is used for a single patient. Although the size of the views for the patient can be reduced, we get an idea about the limits when visualizing multiple patients. In fact the prototype contains no techniques for visualizing or managing a cohort.

Mapping of Time

With the exception of *TimeRider* [44], which uses animation, all presented systems map time to the X-axis in visualizations. This method is so commonly used for a long time now, that it is strongly etched in our minds. We would not think of mapping time to the y-axis like in old visualization approaches.

Handling of Cohorts

Except *CareCruiser* [23] and *VisuExplore* [46] all described systems are able to manage at least one cohort. *VISITORS* [27] and *Caregiver* [13] are even able to handle multiple cohorts, intersections thereof and manage the creation and arrangement of patients within them. *LIFELINES2* [65], *Outflow* [69] and *TimeRider* [44] can only depict a single cohort, at a given point in time.

Aggregation

The systems differ largely in terms of aggregation techniques and show a multitude of approaches. *TimeRider* [44] does not provide any approach to aggregation of parameters. In *LIFELINES2* [65] the sequences of patients are all stacked above each other. No aggregation of clinical parameters is implemented. An aggregation of the temporal distribution of events is however given in the bottom view. *Outflow* [69], uses a high degree of aggregation because it averages almost everything (time intervals, symptoms) and hides a lot of detailed information. On the other hand, it gives a very good idea about general trends. *VISITORS* [27] also provides high levels of aggregation, since it is designed to handle huge numbers of patients. The

Parameter views in *VISITORS* divide parameters into qualitative classes, aggregate the number of patients in the class and show them in the form of bars. This bars are sized according to the number of patients in the class at the given point in time. This visualization method is able to indicate the general development of a parameter in a cohort over time. *Caregiver* [13] shows information aggregated on all granularity levels. Meta information about cohorts is given in the top view. Highly detailed information of the development of parameters and all individual patients is presented on a small view on the bottom.

Time

In terms of conveying time, the most common method is to use the absolute timescale provided by the point of event occurrence or parameter measurement in real world and calculate relative times to a selectable sentinel event like the start of the treatment. As previously described an alignment point has to be used for cohort studies, to make treatment processes comparable, when they did not take place at the same time. All presented systems which are able to handle groups of patients offer such a mechanism.

Clinical Guidelines

Visualizing clinical guidelines was an early goal in the development of the *CareVis* and *Care-Cruiser* systems. Over time many features were added to the views and the were continuously improved. While we were able to identify other approaches for the visualization of guidelines, we found no integrated views in any of the other systems we encountered in our survey.

Compliance to Guidelines

In this subsection, we cover both, the visualization of compliance as well as the computational analysis of compliance. The *CareCruiser* prototype, extended by Bodesinsky in his master thesis, integrates both features and represents the state of the art solution covering this problem. During our search we were able to identify one other approach of integrating compliance in information systems implemented by Panzarasa et. al. in [36]. The developed prototype has the disadvantage of only giving a textual description of the clinical recommendations within the guideline a patient was eligible for. Non-compliance are also just shown as single instances with a textual description, while no time dependent information in guideline execution similar to the concept of missing intervals is considered. The authors also conducted no approach to graphically visualize the information in their system.

Details for Single Patients

While all systems offer details on demand about single patients, the clinical process of the single patient is not observable in all systems. Often a trade-off between conveying detail for single patients and the overall trend in the cohort has to be considered. In *LIFELINES2* [65], *Care-Cruiser* [23] and *VisuExplore* [46], the events and parameters of the single patient are easily observable. However a large group of patients is just stacked and only a small number of patients is present on the screen at a given point in time. *VISITORS* [27] is the hugest counterpart, because it aggregates all parameters and a patient is lost in the abstracted and aggregated statistics. As previously stated *Caregiver* [13] has views for all granularity levels and the details for single patients are visualized in the bottom view.

In Table 3.2, we give a general comparison of attributes of the techniques as described in this section.

Table 3.2: Summary and Comparison of Systems for Cohort Visualization

	Mapping of Time	**Manage Groups**	**Add/Remove Parameters**	**Add/Remove Patients**	**Sort Items**	**Parameter Abstraction**	**Visualization of Guidelines**	**Patient Status**	**Development over Time**	**Brushing**	**Details on Demand**	**Filtering of Patients**
TimeRider	Animation		●					●	●		●	●
LIFELINES2	X-Axis	●	●	●	●			❍	●	●	●	●
Outflow	X-Axis				●	●		●	●	●	●	●
VISITORS	X-Axis	●	●	❍		●		●	●		●	●
Caregiver	X-Axis	●			●			●			●	●
Carecruiser	X-Axis		●	●		●	●	●	●	●	●	●
VisuExplore	X-Axis		●	●		●		●	●			●

Table 3.3: *

●: Full Support, ❍: Partial Support, ' ': No Support

CHAPTER 4

Implementation

This section describes the integration of prototypes developed during the conduction of this thesis.
We start out by giving brief descriptions of the user interface and visualization frameworks we used in section 4.1 Frameworks.
In section 4.2 The Integration of CareCruiser and VisuExplore, we continue by giving reasons for the selection of *CareCruiser* and *VisuExplore* as base for our prototype and describe their integration into a single prototype.
The developed techniques, visual artifacts and encodings of attributes are presented and described in 4.3 Integration and Extensions for Cohorts.
We conclude this section by providing a high level documentation of the architecture and classes we implemented, to provide the functionality of our prototypes in section 4.4 Technical Documentation.

4.1 Frameworks

In this section we give a short introduction on the existing frameworks, which were used to implement the developed and extend the already existing prototypes for this thesis. Both frameworks are based on the JAVA programming language, since Prefuse was built with the presented Swing GUI [2] and the state of the art *VisuExplore* [46] prototype is again based on the Prefuse [3] framework.

Swing

The popular Swing API [2] is a comprehensive framework for implementing graphical user interfaces in the JAVA programming language.
Because Swing as a whole was programmed in Java, it inherits the convenient platform independence feature from JAVA and can run on any system with an installed Java Runtime Environment. The components of Swing, some of which we will describe briefly later, are extensible

and the inheritance feature can be used to easily extend or override default implementations of the framework, which enables programmers to provide altered or additional functionality.

The GUI components are painted using the Java 2D APIs and do not rely on any native OS GUI components. However, Swing is based or related on the AWT framework, which enables it to interact with the GUI management framework of the underlying operating system. This enables swing components to access the interaction functionality of the OS (user interactions, screen-device mappings), while they are responsible for rendering themselves when a call to the paint() method of the component is made. As is the case for all objects in Java, the uppermost class in the Spring framework inherits the *Object* class. Every Swing component which can be painted on the screen inherits from this class, while the components form a hierarchical structure. They are always contained within a top-level container like *JFrame*, *JDialog* or *JComponent*. Figure 4.1 shows an example hierarchy for a sample menu. Top level containers can again be comprised of containers, or concrete element like JMenuBar or JPopupMenu.

Like many state of the art GUI toolkits, Swing makes use of the popular Model-View-Controller design pattern. This pattern decouples the the data from the view and the controls thereof, which enables easy substitution of different views (e.g. a web view or a thick client GUI) on the data or the data model itself.

Swing components are able to receive user input after implementing the *ActionListener* or *MouseMoveListener* interfaces, providing the basic components needed for interactive visualizations and the implementation of the prototypes.

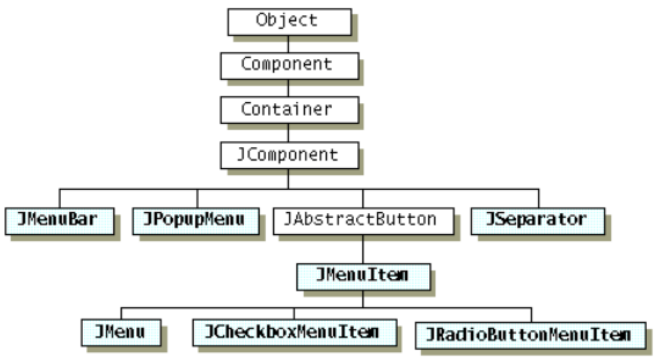

Figure 4.1: A sample hierarchy (menu components) in the Java Swing framework [2].

Prefuse

Prefuse [3] is a set of software tools for creating rich interactive data visualizations. It is an open source framework and the authors have opened the development of the framework to the community for some time now. Since it is based on the Swing framework the components can easily be integrated into Swing based applications and also web applets [3]. An overview of the the Prefuse packages is given in figure 4.2.

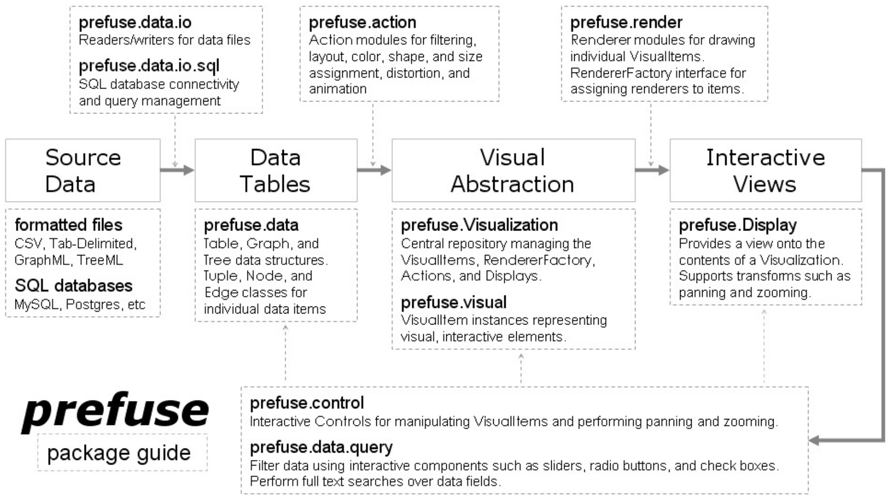

Figure 4.2: Prefuse packages arranged along the Information Visualization pipeline [3].

The framework implements and enables the use of the basic form of the information visualization reference model or pipeline as presented in figure 4.2.

Several classes are provided to read data of different formats. Most notably are data from SQL databases and formatted files like .csv delimited files. The data is read out and can be stored in different in-memory formats like tables, graphs or tree structures, which are selectable by the user.

Prefuse.action modules are used to filter data in the model or alter the layout, table or size of data and provide them to classes of the visual abstraction components. They contain the packages necessary for creating and managing the visual items and structures of which a visualization consists of and map the data obtained from the action modules to a visual form.

At the end of the pipeline, a set of renderer classes for distinct visual items are used to paint the visual attributes on a *prefuse.Display* instance. A display object provides a view on the data and also supports user interactions like panning and zooming.

The display instance is also used the handle the user input for the *prefuse.control* and the *prefuse.data.query* packages. As the names suggest the control package handles the input, while the query components are used to search and filter the data structures for elements that match the criteria defined by the user. She is able to set this criteria by using the controls of the interface. As shown in figure 4.2 these components are able to alter the instances of data structures, view elements or displays. Alteration of these objects in the pipeline results in the re-execution of parts of the information visualization pipeline and also alters the resulting visualization.

The implementation contains all components which are necessary for completing the sense-making loop described in section 2.3 Visual Analytics.

4.2 The Integration of CareCruiser and VisuExplore

For the integration and development of the prototypes of this thesis, we needed to find a suitable base to build on. *CareCruiser* has been continuously improved and extended and contains support for guidelines, as well as the most recent visualization techniques for compliance, it's disadvantage is the rather rigid software architecture. The biggest drawback is the lack of a feature for dynamically adding and removing facets, which contain the selected visualizations. For this reasons we decided to integrate the relevant features of *CareCruiser* into the most recent version of *VisuExplore*.

VisuExplore offers the dynamic loading and removal of different kinds of facets as mentioned above, whereby each facet can contain any visualization offered by the framework. Also the more flexible overall software architecture enables an easier integration of new types of visualizations. It however lacked support for clinical guidelines in both, data handling (parsing and interpreting a given guideline in its .xml format), as well as the visualization thereof as shown in figure 3.20 on the left hand side of the screen.

Both prototypes did not exactly meet the requirements we asked for, but a combination of both seemed desirable to meet our goals. We decided to migrate all guideline related features, as well as the visualization techniques which have been developed so far, into a single prototype based on the latest version of *VisuExplore*.

A screenshot of the newly integrated prototype, also including all newly developed visualization techniques, is shown in figure 4.3. We presented the old prototypes in figures 3.20 (*CareCruiser*) and 3.19 (*VisuExplore*) for comparison.

In the following paragraphs we give brief descriptions of the elements and their functionality shown in figure 4.3, while in following subsections we focus on details of the developed techniques:

(A) Guideline View Panel: The main window of *VisuExplore* has been split in two distinct panes, allowing the dynamic addition and removal of structural and logical guideline visualizations on the left, which can be achieved by clicking on the second symbol in the menu bar. A list of all guidelines present in the guideline folder is presented and the user is able to select the applicable one. The visualization technique for guidelines was reintegrated from the latest version of the *CareCruiser* prototype developed by Bodesinsky in [11]. It also contains the extensions related to compliance like statistics bars and customized tool-tips. The guideline is loaded in a

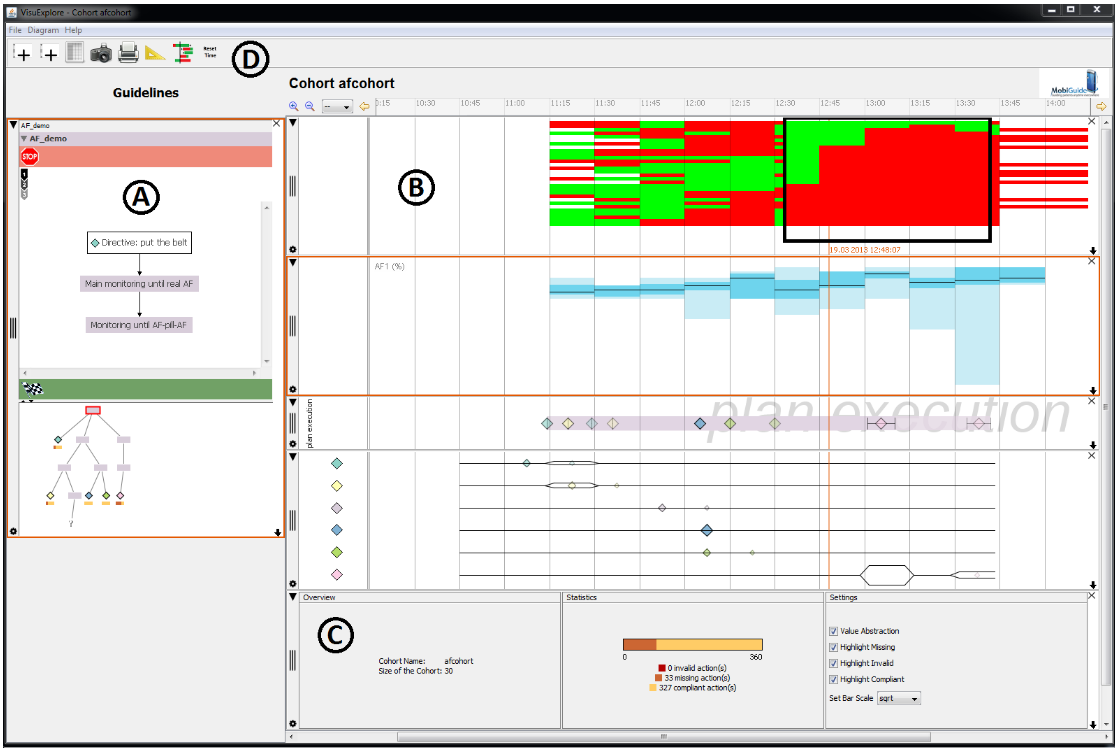

Figure 4.3: The fully integrated *VisuExplore* prototype, showing all previously developed, altered and newly developed visualization techniques for cohorts.

facet like all other visualizations in the prototype, while in the guideline visualization pane only facets which contain guideline visualizations can be added.

(B) Time Oriented View Panel: The right side of the interface was mainly taken over from *VisuExplore*. It allows the dynamic addition and removal of time oriented visualizations each contained within a facet. Facets can also be moved and reordered within the right pane. The panel shows all visualizations developed during the conduction of this thesis, while we provide descriptions about them in the following sections.

(C) Statistics Facet: The statistics facet was also reintegrated from the latest version of *Care-Cruiser*. It represents an exception to the rule of only adding time oriented visualizations to the right panel. For the sake of a full reintegration we however violated this restriction.

(D) Menu Bar: Several key functions are offered by interaction with the menu bar. By clicking the first button the user is able to add a new diagram to the time oriented view. The second button allows the addition of guidelines on the left. The button with the spreadsheet like icon opens a tabular view of the data that is currently visualized in the facet selected on the right. Only one facet can be selected at any time, while the borders of the selected facet are surrounded by an orange box. The camera icon offers to take a screenshot of the current view, while the print icon opens the standard print dialog of the operating system. A tool for measuring the time between two points in the time oriented visualizations is offered by the button depicting the triangle. The last two buttons are related to the temporal alignment feature described in section 4.3

Adding Visualizations

As described in the caption of figure 4.3, the user is able to initiate the dynamic loading of time oriented visualizations by use of the leftmost button in the menu bar. After a click on the button the dialog presented in figure 4.4 appears:

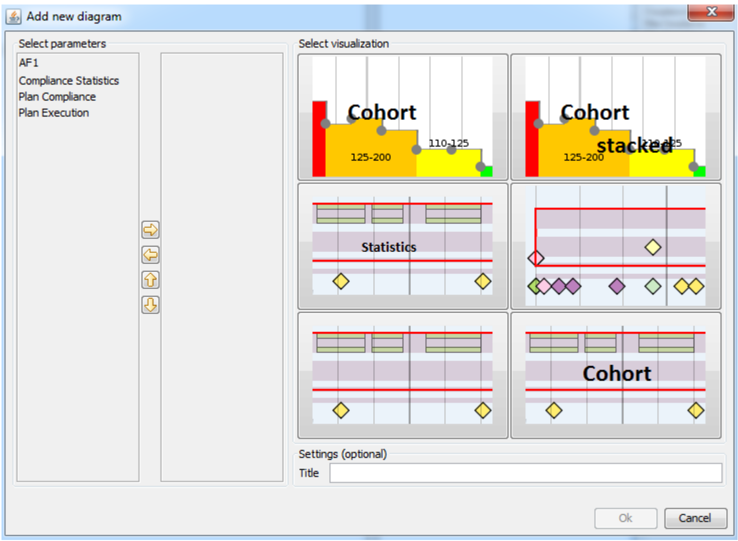

Figure 4.4: The new facet dialog for the addition of time oriented facets. The statistics facet is also selectable. Available parameters are shown on the left, while toggle buttons represent the visualization techniques offered by the prototype on the right.

The list on the left of the dialog shows the parameters available for visualization. Plan execution, plan compliance and compliance statistics are present by default, while all other parameters are loaded from the patient or cohort loaded in *VisuExplore*. By double clicking a parameter in the list, or selecting it and pressing the yellow button pointing to the right, it is moved from the left to the right list box.

After the addition of parameters, one of the buttons on the right, each representing a different kind of visualization indicated by its icon can be selected. Multiple parameters can be loaded, since some visualization techniques are able to visualize more than one parameter at a time.

The dialog itself conducts a validation, if the selected visualization is able to handle the selected parameters and only disables the OK button if this is not the case. Therefore only valid parameter and visualization combinations are selectable by the user.

4.3 Integration and Extensions for Cohorts

In this section we describe all the data related, as well as visual features implemented, to provide the functionality for the visual analysis of cohorts. We focus only on things related to the usage and interpretation of these features while neglecting specific technical details.

Loading of Parameter Measurements, Actions and Demographic Data

The data for each patient so far consisted of a set of files. The first one contains a set of parameter measurements as well as the related points in time. A separate file contains the actions executed for the patient, while yet another file encloses the demographic information.

To offer the proper handling of this potentially large set of files for a cohort, we decided that the paths to all related files should be handled within a single .xml file. This .xml compliant file consists of any number of records (one for each patient). Each record provides the path to the files containing the relevant parameter measurements, conducted actions and demographic data of the patient.

In theory cohorts of any size can be loaded into the prototype. A limit might be given by the memory of the machine, while the more probable case is that a proper visualization of all the loaded data is limited by the finite screen space.

```
<cohort name="afcohort">
    <patient>
        <path>patients</path>
        <filename>parameters_1.txt</filename>
        <actionfilename>actions_1.txt</actionfilename>
        <demofilename>demographics_1.txt</demofilename>
    </patient>
```

Figure 4.5: A sample entry in the .xml file provided to load the data for a cohort into the prototype.

A sample entry is presented in figure 4.5. The name of the cohort can be given as attribute in the cohort tag. Then one ore more several patient records can be provided inside the cohort tag, each one comprised by the following entries:

- **path:** Path to the folder the data files are stored in
- **filename:** the name of the file containing the parameter measurements
- **actionfilename:** the file containing the actions executed for this patient
- **demofilename:** name of the containing demographic information like age, sex and name of the patient

Hierarchical Guideline View

For the structural guideline view we reintegrated the techniques as developed by Bodesinsky in his master thesis [11].

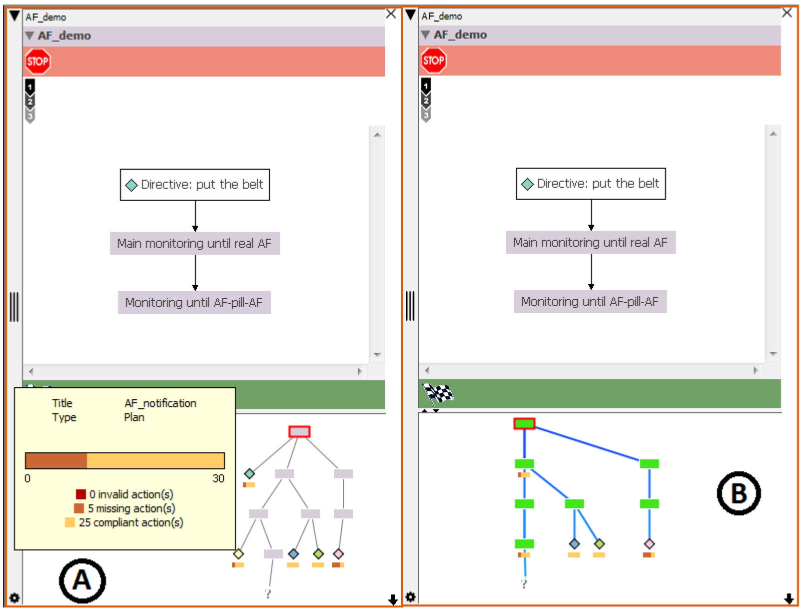

Figure 4.6: (A): The reintegrated guideline view contained within a facet. Stacked bars and tool tips are used to visualize compliance information of the cohort. **(B):** The settings dialog of the facet offers the user to switch to the guideline order visualization developed by Michael Fischer.

Extending the techniques for cohorts was fairly straight forward. They are equally applicable without adjustment of the visual elements and artifacts. As can be observed in figure 4.6 by the high amount of actions, the aggregated action counts of all patients in the cohort are visualized.

We also integrate a newly developed technique for visualizing guideline execution order, developed by Michael Fischer. The further away the color of the plan body node moves from green, the later the action should be applied according to the guideline. If the guideline order has not been followed properly this is observable by a wrong ordering of color in the view. Times between executions of the cohort are averaged and provided to the view, so it is able to also provide visualization of execution order for cohorts. The thickness of the edge between two nodes indicates the amount of compliant actions in each of the sub plans.

We enable the user to dynamically switch between the two visualizations, by selecting and deselecting the check box in the settings dialog of the guideline facet opened by a click on the cogwheel icon at the bottom left of the facet.

Compliance Statistics Facet

As stated in the description of figure 4.3, we also reintegrated the statistics panel developed by Bodesinsky. The statistics visualization is technically not a time oriented visualization, but it can also be added as facet to the time oriented view.

Figure 4.7 shows the reintegrated statistics panel used to visualize and present the action and compliance related data calculated for the cohort.

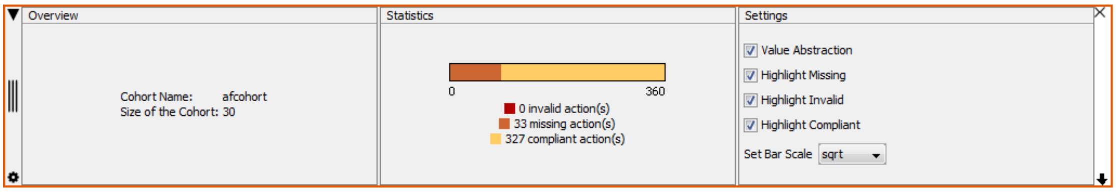

Figure 4.7: The reintegrated Statistics Facet from the *CareCruiser* prototype. Counts of compliant, valid and invalid actions conducted during the treatments of the cohort are visualized.

We made a slight adjustment on the left side of the panel, which was initially used to provide a patients demographics to show information related to the cohort.
In the middle of the facet all actions executed during the treatment of all patients have been aggregated and are visualized in the form of a stacked bar, while the controls on the right hand side are again fully functioning and the guideline view is instantly updated when the settings are altered by the user.

Visualization of Parameter Development

We split the time oriented visualization techniques used for this thesis into visualizations that provide insight into the development of parameters and visualizations for the conducted treatment actions and compliance information within them. In the following section we describe visualizations for the first type.

Stacked LifeLines

As we described in section 3.3, temporal abstractions of parameters can be used to increase virtual resolution, thus potentially enabling us to use the available screen space for the visualization of a large number of patient treatments.

With the Stacked LifeLines visualization we make a basic attempt to implement this functionality, by using the most convenient form of temporal abstractions: Color-Coded LifeLines. In our case there are two distinct qualitative areas indicated by the colors red and green, whereby green of course denotes the default range for the parameter value.

Figure 4.8 presents the prototype we implemented.

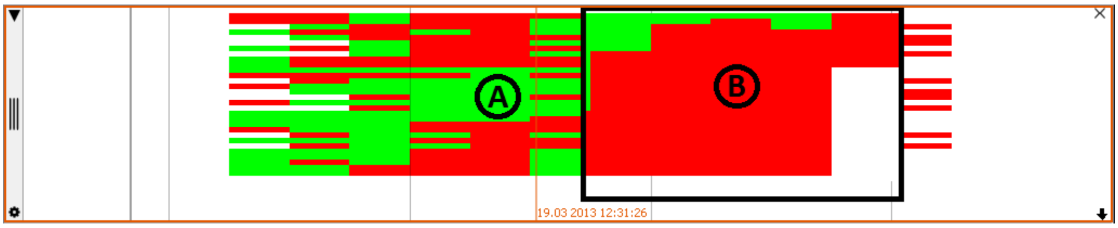

Figure 4.8: Visualization of the development of a parameter for all patients within a cohort, by stacking LifeLines of all patients above each other **(A):** The LifeLines along the X-Axis calculated from the parameter measurements taken during the treatment of each patient over time. **(B):** The magic lens provides easier insight into the current qualitative parameter state of the whole cohort by ordering the lines within the lens according to their qualitative area.

The computed LifeLines are stacked above each other with the intention of giving a general indication of the development of the visualized parameter in the cohort. A fixed amount of space is assigned to each LifeLine determined by the height of the display in the facet and the number of patients. If the facet is resized, the LifeLines are also resized so the visualization always uses all the available space within the facet.

A technical limit for this method is set by the screen resolution. For drawing each LifeLine we need at least 2 pixels of screen space (one for the line, one used to provide a small gap between LifeLines). Therefore the upper bound of patients that can be visualized is determined by the display's height and therefore ultimately by the screen resolution, if the facet is resized to use all vertical space available.

While we aimed to provide a general indication of the development and still show some detail for the single patient, rapid switching between the color of the LifeLines in an interval might be hard to interpret properly by the user. By providing a so called magic lens for this visualization, we tried to provide an interaction technique to solve this problem, and make it easier for the user to interpret the data in such cases.

The functionality of the magic lens is shown inside the black lines in figure 4.8. The user is able to drag the lens across the LifeLines by moving the cursor inside the lens, clicking and holding the left mouse button and moving the mouse pointer across the screen. The horizontal size of the lens can be adjusted by dragging the left and right lines. All LifeLines, currently placed under the lens, are ordered (the default qualitative area is the first area) according to their current qualitative abstraction for each time interval and realigned.

Cases might occur where treatments did not take place at the same time and there are no LifeLines for some of the patients. In such a case all available LifeLines are ordered as before and stacked directly above each other, while the remaining space determined by the amount of missing LifeLines is filled with white.

Streaming Boxplots

The Stacked LifeLines visualization aims to provide all available information to the user by visualizing it in the available screen space. On the contrary, with the Streaming Boxplots technique, we approach the problem by visualizing the results of a statistical aggregation method executed for each distinct interval in the data points.

The Streaming Boxplots visualization consists of a series of single box plots for each given interval. The black line in each interval indicates the mean value for the parameter in the given interval. The range from the 0.25 percentile to the 0.75 percentile, is visualized by the darker blue color, while the whole range of the values is visualized by the light blue area. Therefore a statistical abstraction is calculated for each interval clearly indicating information about the median and the spread of the parameter value within the given interval.

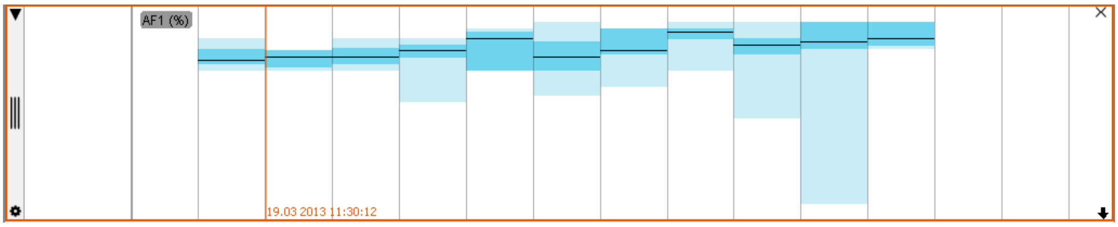

Figure 4.9: Visualization of the overall development for the visualized parameter measurement values as Streaming Boxplots.

Because the provided method aggregates the data, details about single patients are lost or at least hard to grasp. Streaming Boxplots might however be able to better suit the visualization of large scale trends and concentrated important events. This might for example be a declining universal development of measurements of the visualized parameter, after a lot of recommended actions were not executed in the treatment process of the cohort.

Visualization of Plan Execution and Compliance

In this section we describe the methods we developed for the visualization of executed actions and compliance in the treatment process. We also developed two distinct visualization methods for this problem. While the first one again keeps the actual absolute time points and shows all the data, the second one is again based on aggregation of data within time intervals as used in the Streaming Boxplots visualization.

Alpha Blended Visualization of Plan Execution and Compliance

In a first approach we attempted to visualize the execution and compliance of and to a guideline by using the visualization technique developed by Bodesinsky as described in section 3.3.

We kept the original visual encoding as well as methods, while we only adjust the alpha channel of the color for each visual item. The alpha for a single item is calculated by the maximum alpha value divided by the number of patients in the cohort. Therefore, if all of the actions in the cohort have been executed at the exact same point in time all actions are stacked above each other and shown in exactly the same way as a single action executed in the previous version for one patient.

A sample visualization of the executed actions in a cohort by this method is presented in figure 4.10.

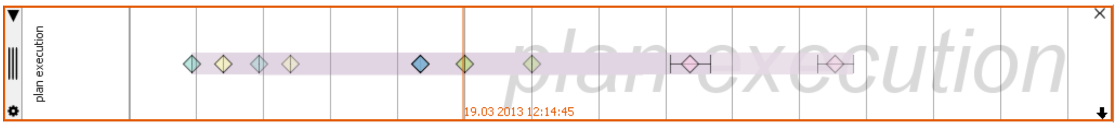

Figure 4.10: Visualization of the executed actions and compliance of the cohort by overlapping executed actions with alpha blending.

An action executed at the same point in time for roughly two thirds of the cohort would show two thirds of the full alpha. Single lonely actions would be almost invisible. In theory this method is able to show clusters of executed actions, while it will however fail if the execution points of actions are distributed more uniformly.

We also implemented a simple approach for interaction with this visualization. As described the less actions there are at the same point, the less overall of opaqueness is contained within the visualization. While calculating the alpha value for a single patient, might be applicable in cases where lots of actions are executed at the same point in time, some hardly visible clutter might be shown in cases where actions are spread equally over time.

By scrolling the mouse wheel up, the user of the visualization is able to increase the alpha value and therefore the visibility of all items, up to a point where each item's alpha reaches the maximum alpha value. By use of this technique she is able to better grasp instances of plan executions of cohorts, with larger spread of action execution throughout time.

Aggregated Visualization of Plan Execution and Compliance

Our second approach to the visualization of executed actions and the compliance for cohorts also uses aggregation of time intervals as basis. The original representation of actions as diamonds is kept, while visual characteristics of each diamond are altered to encode attributes acquired from the aggregation of the data.

We developed an algorithm, that iterates through all time intervals in which actions have been executed. The intervals are equally sized as in the above techniques. In each interval all of the actions of each type are iterated over and information about plan execution is aggregated.

Each distinct action type is assigned an equally sized amount of vertical space on the display.

The number of validly executed actions, as well as missing actions, for each distinct action type in the cohort is counted, resulting in three distinct dimensions within the treatment data for each time interval.

The following listing describes the dimensions, and the visual encoding assigned to each of them:

- **Number of patients:** Size of the diamond
- **Number of validly applied actions:** Opaqueness of the diamond
- **Number of missing actions:** Distance of the line above and below the diamonds. The greater the distance the higher the amount of missing actions in corresponding interval.

An example for the visualization of executed actions and their compliance information by the described technique is shown in figure 4.11.

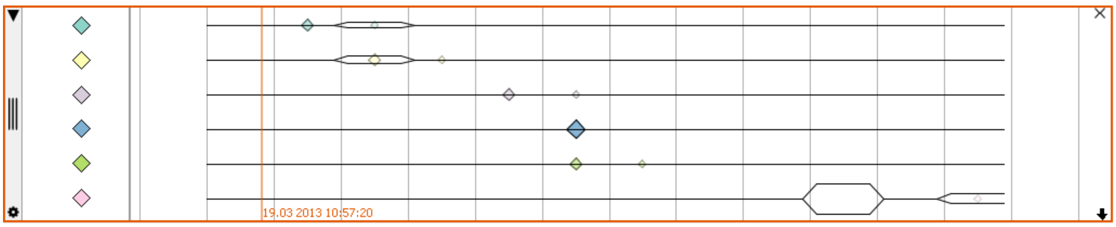

Figure 4.11: Visualization of the executed actions and compliance of the cohort. Actions are split according to type and counts of valid, invalid and missing action instances are aggregated for each interval. The larger a diamond, the more patients the actions has been executed for. The opaqueness indicates the number of total instances of the action in the interval. The amount of missing actions in the interval is visualized by the lines above and below each series of diamonds for each action.

This visualization also scales to size like the Stacked LifeLines method. If the size is decreased, the diamonds and items get smaller to fit within the available screen space. However small details might be not recognizable any more when the facet's size is set too small.

Temporal Alignment

As discussed in section 3.5, temporal alignment is an important feature for comparing data of treatment processes of multiple patients. In clinical practice a treatment starts when it is indicated by the patient's condition. Therefore the measurements taken and actions recorded are different absolute points in time and often the data is temporally scattered over years. The courses of the different treatments can however be compared, when time is calculated relative to a specific point within the treatment. This could for example be the first measurement of a parameter or the first action conducted of the guideline.

We offer temporal alignment of treatment, plan execution and compliance data within *VisuExplore* in the following way:

The two leftmost button in the Menu Bar of the application as shown in figure 4.3, enable the use of the feature. The first button is a toggle button. When enabled, it allows the user to align all treatments of the cohort, by clicking on an action in the structural overview graph of the guideline panel.

After a click on the action, all data points are recalculated in reference to the selected action. While the intervals between the points remain unaltered, the first occurrence of the selected action acts as zero reference point for all treatments. Instantly after this all visualizations are updated and present the aligned data as shown in figure 4.12.

Figure 4.12: The header and Stacked LifeLines visualization after alignment to the first occurrence of an action, executed within each treatment.

As can be observed, in the time scale header at the top and at the vertical orange line drawn within the displays, the application also uses relative time labels after alignment. All time scales (header in the controls of the time oriented panel, background of the visualizations, time units and string drawn next the vertical orange line drawn on all displays) and time points within the application are rendered in relativity to the selected reference time point given by the first occurrence of the selected action in each treatment.

After a click on the second button, the application switches back into the absolute time mode. All data points are reset to their initial temporal state, provided by the data within the loaded data files.

Dynamic Aggregation of Data for Visualized Time Intervals

Treatments are normally not temporally. Furthermore some measurements might have been taken in sampling rates, which are too short or too long for a proper visual analysis of the treatment within the examined time spans. For example they might have been taken in 5 minute intervals, while the users intention is to analyze the treatments a time span of several weeks. For such cases, the given data will likely not fit the visualization. A big amount of alternating visual items might be present which results in visual clutter.

To enable the proper analysis in such cases, we introduced the dynamic scalable aggregation of the treatment data in equidistant time intervals in *VisuExplore.java*.

Figure 4.13 shows how the data points for the cohort are aggregated. The time interval length for which the data is aggregated, is given by the time units visualized within the header. In the instance shown in figure 4.13 we are at 10 minutes intervals which also determine the aggregation interval duration within the visualizations.

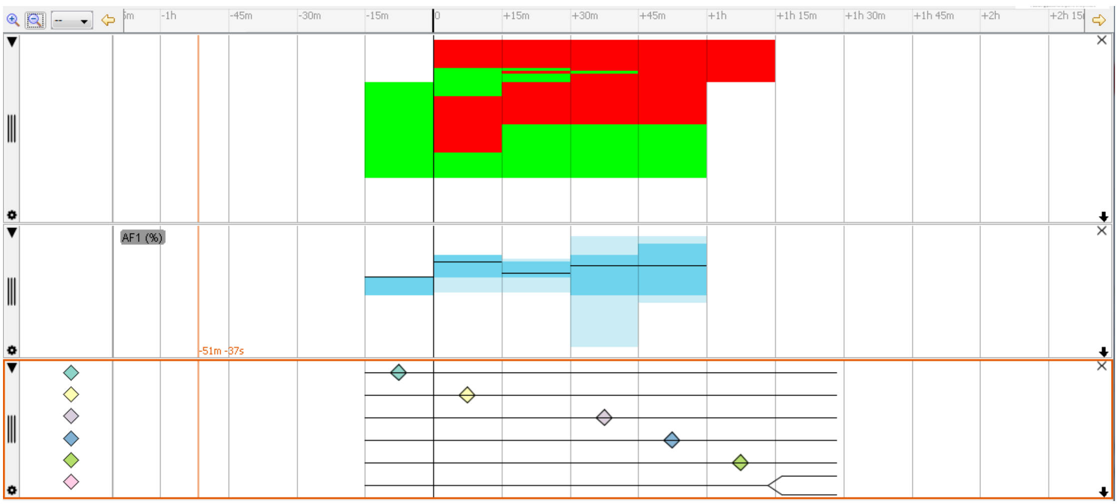

Figure 4.13: The Stacked LifeLines, Streaming Boxplots and Aggregated Plan Execution and Compliance visualizations after alignment and reduction of the time unit interval to 15 minutes.

For both, the Stacked LifeLines as well as the Streaming Boxplots visualizations, a single data point is calculated for each interval. If there is no point within the interval, the last value is also used until the next original data point occurred. In the case where multiple points fall within the calculated interval, the data point is calculated as duration weighted average from all data points within and the data point before. The result is a recalculated data table, containing a single averaged data point for each patient in each time interval.

In case of the Aggregated Plan Execution and Compliance Visualization, the dimensions for each action are also calculated for each time interval as described in 4.3. The diamonds are drawn in the center of the time interval. Actions that were executed exactly at the time corresponding to an interval border, are counted to the previous interval. We decided to handle the specific case in this way, because the execution of an action affects the parameters values development in the following time interval.

In relative mode the reference date also acts as reference point for the recalculation of intervals, while in the absolute time mode aggregation is started from the first parameter measurement taken for the first patient in the cohort.

Selection and Highlighting of a Single Patient

Finally we implemented a brushing and linking approach to enable the user to gain insight into details for a single patient.

The interaction technique is offered as additional mode in the settings of the Stacked LifeLines visualization. If enabled, the user is able to hover the mouse over any of the LifeLines of the cohort as shown in figure 4.14. The selected LifeLine is enlarged and separated from the others by adjacent lines colored in magenta.

In all other visualizations visual items related to the patient selected by the LifeLine, are highlighted in different ways. If a Streaming Boxplots visualization for the cohort is present, the patient's value, if available for the visualized interval, is drawn as horizontal magenta colored line within the interval.

In both plan execution and compliance related visualizations, the action items to which the patient's treatment data contributed to are highlighted. The borders of the affected items are drawn in magenta like the lines in the Streaming Boxplots visualization. In case of the aggregated visualizations, an action is highlighted if the action was conducted for the selected patient within this interval.

With this method we aim to give a general indication of the treatment conducted for the selected patient, while it represents a simple and intuitive approach.

Figure 4.14: All visual items in each visualization, that are related to the treatment of the selected patient are highlighted. In the Streaming Boxplots visualization the parameter measurements for the patient are highlighted as magenta lines. In the plan execution visualization and aggregated compliance visualizations, items for which the patients treatment contributed to are highlighted by coloring their borders in magenta.

4.4 Technical Documentation

In this section we provide a high level documentation of the technical aspects of the newly integrated prototype. We focus on the large scale functionality of the important classes, while we skip a detailed description which would exceed the scope of this thesis.
To gain insight into the exact details, the developed code contains java doc conform comments, which provide a low level description of the code.
We present class diagrams which present the interactions and relations between relevant classes.

Startup and Initialization

The prototype developed during this thesis is started by executing the class *VisuExplore.java* in the *visuexplore* package. While the old functionality let the user load a comma separated file with the patient's data, a cohort can now be loaded by providing an .xml extended file compliant with the format we described in section 4.3. If an .xml file is provided, the main class determines that the loaded patient must be a cohort and acts like that. The Cohort constructor of *Cohort.java* in *visuexplore.data* is invoked to load the data for the patients as specified in the given file.

The Cohort and Patient Data Classes

The class diagram shown in figure 4.15 shows all classes used to handle the loading of data, as well as providing other functionality related to data handling for cohorts and patients.

For loading the data from an .xml file with the structure presented in figure 4.5 the Java Architecture for XML Binding (JAXB) is used. While an instance of *XMLPatient.java* is created for each patient entry in the cohort, *XMLCohort.java* contains a list of all loaded patients as well as the name of the cohort. The *Cohort.java* class invokes the process and then uses the previously present .csv file reading functionality, to parse all files and create and store an instance of *CsvPatient.java* for each patient in the file.

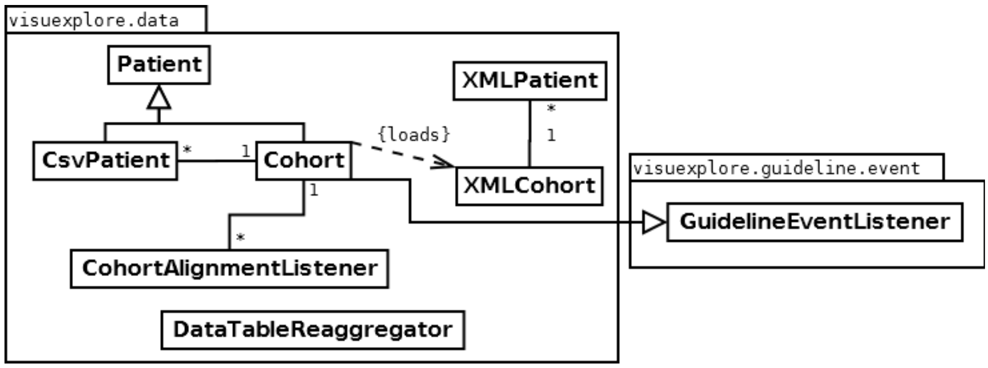

Figure 4.15: The *visuexplore.data* package. All relevant classes and relationships for the handling of the cohorts data are shown.

Each *CsvPatient.java*, contains the data of measured parameters, the executed actions, as well as the used file names and demographic data.

As visualized in figure 4.15, both *CsvPatient.java* and *Cohort.java* are inherited from the abstract *Patient.java* class. This seems ambiguous and is indeed a workaround. Several parts of the code in *VisuExplore.java* expect an instance of patient. Therefore the *Cohort.java* class also provides all these functions, which is reasonable since the cohort also has a list of parameters and a name. Functions that would provide a data table for the single patient have been rewritten to return a joined table of all parameter records. The functions append the patients id as given by the file name as well as a unique incremental patient number for identification and uniqueness of the tuples in the joined data tables.

The *Cohort.java* class now also handles the task of the initial calculation of compliance data. In the *CareCruiser* prototype the compliance was actually computed, but this functionality has been neglected for this integration. Compliance information is given by a column within the file containing the executed actions for a patient. An instance of *ComplianceModel.java* is created and the counts for valid, invalid and missing actions in the cohort are saved within the model. The functionality and code remained the same as developed by Bodesinsky [11]. This is done on start-up of the application, after parsing the data files, since counted compliance information for the cohort is not alterable in the prototype. The calculated model provides the basis for the stacked bar visualizations in the guideline view as well as the Statistics Facet.

The data classes also handle the recalculation of their dates as needed by the time alignment feature presented in section 4.3. *Cohort.java* offers all alignment options and the alignment can be invoked from there. The class has access to all patients and computes a difference timespan for each of the patients. The *Patient.java* class itself handles the recalculation of all dates for a patient by the given timespan.

CohortAlignmentListener.java provides an interface for classes that need to react on updates after temporal alignment or change of the aggregation interval. For the time being, these classes are all four visualizations for time oriented treatment and action data as described above. Each class that implements the interface needs to subscribe to the singleton *Cohort.java* instance at runtime, to be able to react on updates for the visualized data. The interface offers a second method that is invoked by the *Cohort.java* instance if a patient is highlighted in the Stacked LifeLines visualization.

DataTableReaggregator.java provides the functionality to aggregate the data table containing the parameter measurements, obtained from *Cohort.java* and visualized by the Streaming Boxplots as well as the Stacked LifeLines visualizations. A reference date needs to be provided as starting point, while a value for a fixed length time span denotes the granularity of the aggregation.

Reintegration and Extensions of Compliance Visualization in the Guideline Visualization

A goal of the thesis was to reintegrate the state of the art compliance visualization techniques in the developed prototype and extend them for cohorts. While most of the parts have been kept as they were in terms of their functionality (The structure in the classes has not changed much), the different architecture of the *VisuExplore* prototype required some changes within relationships between them.

Figure 4.17 shows an overview of the reintegrated classes and their relationships.

After the start up of the application, a singleton instance of *ComplianceModel.java* is created in the main class *VisuExplore.java*. This hinders the possibility of multiple compliance models within the prototype and guarantees, that all views access the same information. The calculation of the data is however handled in the *Cohort.java* class. The structure of the model stayed the same, the only difference are the higher counts within the model.

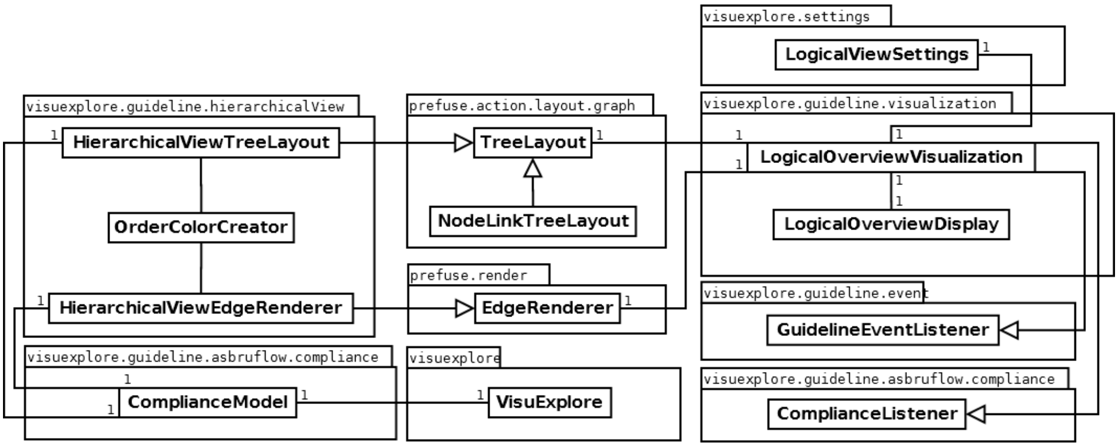

Figure 4.16: Relationships between the guideline view and the reintegration of classes related to visualization of compliance.

LogicalOverViewVisualization.java, from the *visuexplore.guideline.visualization package*, handles the structural view of the guideline. The node renderer was extended by Bodesinsky with the *BarOverviewShapeRenderer.java* class, which we reuse to handle the rendering of the bars. Furthermore the extended tool tip has been reintegrated to show the extra compliance information by use of the stacked bar decorator.

To receive updates for the re-rendering of the bars in case of selections made in the statistics facet, the *LogicalOverViewVisualization.java* class implements the *ComplianceListener.java* interface. It only needs to repaint the view if the function is invoked, since the information for rendering is contained within the singleton *ComplianceModel.java* in the application.

Another interface implemented by *LogicalOverViewVisualization.java*, is the *GuideLineEventListener.java* class. Classes that implement this interface are able to register at the *PlanExecutionSelectControl.java* class in the *visuexplore.guideline.control* package to receive updates if and which plan elements are selected.

The classes containing the functionality for the the additional view on guideline execution order (presented in section 4.3) are now contained within the*visuexplore.guideline.hierarchicalView* package. Switching between the views is realized by exchanging the node and edge renderers in *LogicalOverViewVisualization.java* and re-rendered the visualization. While the edge renderer for the execution order view (*HierarchicalViewEdgeRenderer*) is an extension of the previous edge renderer (*EdgeRenderer.java*) from the Prefuse framework, the node renderer is an additional extension of the *TreeLayout.java* class provided by the Prefuse framework. Both have however equal super types and are therefore easily interchangeable.

Temporal Visualizations and the Time Oriented Panel

In this subsection, we describe the general architecture used in the *VisuExplore* prototype to add visualizations to the time oriented panel. All time-oriented visualizations described, inherit and reuse this functionality.

The most important classes related to the handling and addition of visualizations within facets are presented in figure 4.17.

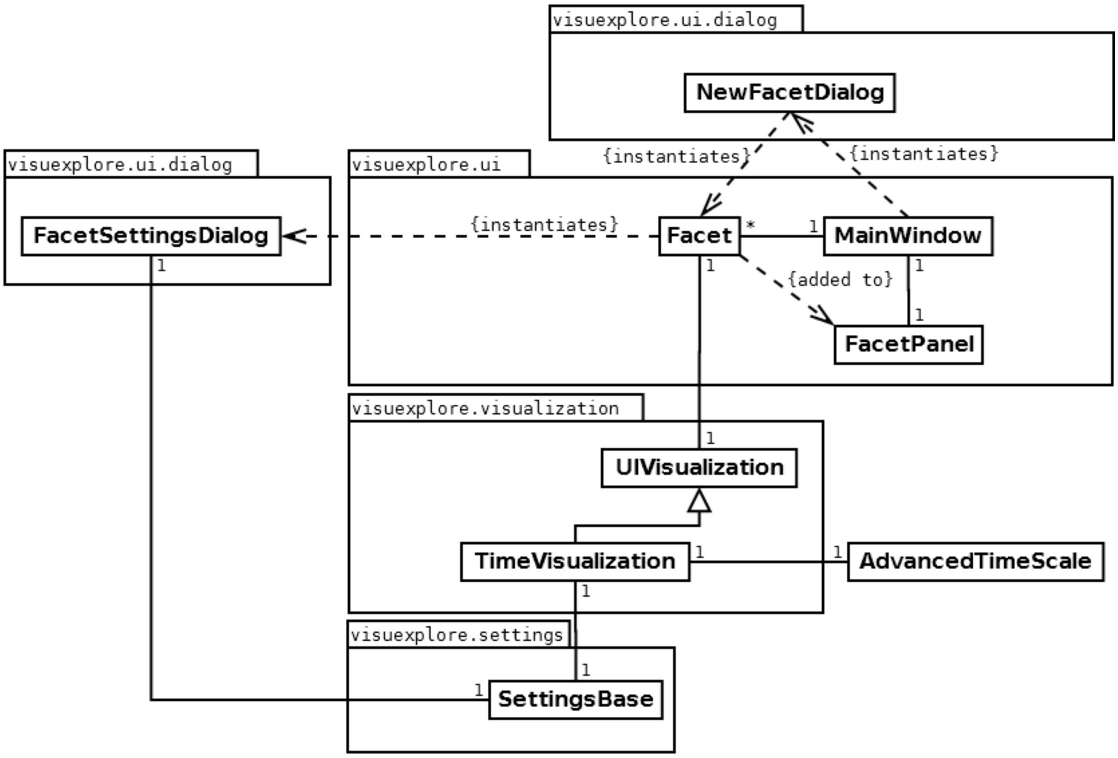

Figure 4.17: Classes related to VisuExplore's software architecture for the addition of visualizations to the time oriented panel.

The process is started from the *MainWindow.java* class, which instantiates and opens a new facet dialog providing the functionality as described in section 4.2. After selection of the parameters and the visualization method a *Facet.java* is invoked, which is added to the facet panel (*FacetPanel.java*). Each facet holds a class that implements the *UIVisualization.java* interface as visualization, which is added to the panel in *VisuExplore.java*, by creating an instance of a Settings class for the corresponding method. All Settings classes are inherited from *SettingsBase.java* which provides a skeleton for the readout and alteration of the settings specific to the visualizations.

In almost all cases, the instance of *UIVisualization.java* contained within the facet is an instance of *TimeVisualization.java*.

Relative Time

The classes used for switching to relative mode and rendering relative times are are presented in figure 4.18. Classes used for rendering the relative time labels are all extended from their original versions provided by the TimeView framework described in [6]. *RelativeMultiMouseTracker.java*, *RelativeTimeScaleHeader.java*, *RelativeTimeScalePainter.java* all provide methods, to switch between rendering absolute times and times relative to a given reference point.

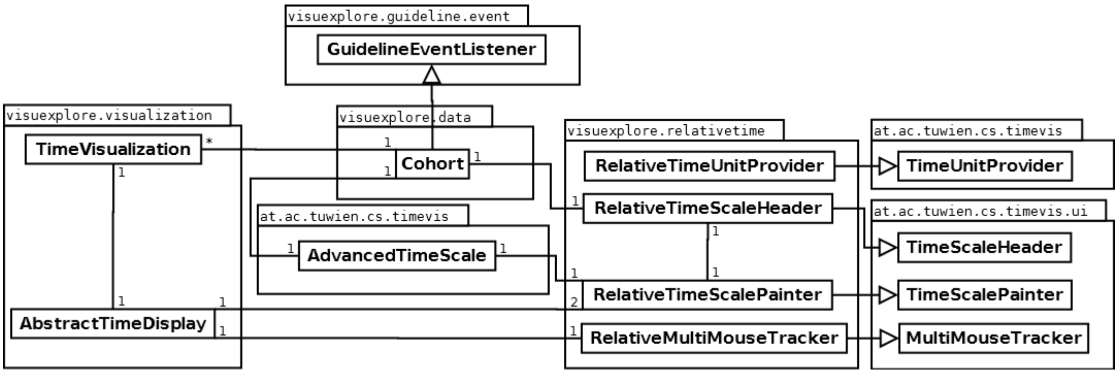

Figure 4.18: Classes and their relationships responsible for rendering relative times.

Cohort.java implements the interface *GuidelineEventListener.java* to act on selections made in the structural guideline view if the toggle button in the menu bar is enabled. If a selection is made in active mode, the class invokes a series of function calls that set all classes and renderers into the according mode and also provide them with the determined reference time point. All visualizations within are set to the selected mode and consequently set their *AbstractTimeDisplay.java* instances into relative mode. The display objects continue by setting the time scales in the back of the displays as well as the mouse tracker into the selected temporal mode.

The mode of *RelativeTimeScaleHeader.java* is directly set by the cohort class, which subsequently invokes the method of the *RelativeTimeScalePainter.java* instance it holds.

RelativeTimeUnitProvider.java extends the original *TimeUnitProvider.java* by offering a static method that creates a *TimeUnitProvider.java* instance better suited for visualizing relative times.

Compliance Statistics

The Compliance Statistics facet represents an exception to the above rule, since it is no time oriented visualization in any sense. Therefore the visualization is not an instance of *TimeVisual-*

ization.java, and just an implementation of the *UIVisualization.java* interface.

Related classes for providing the compliance statistics functionality and the relationships between them are shown in figure 4.19.

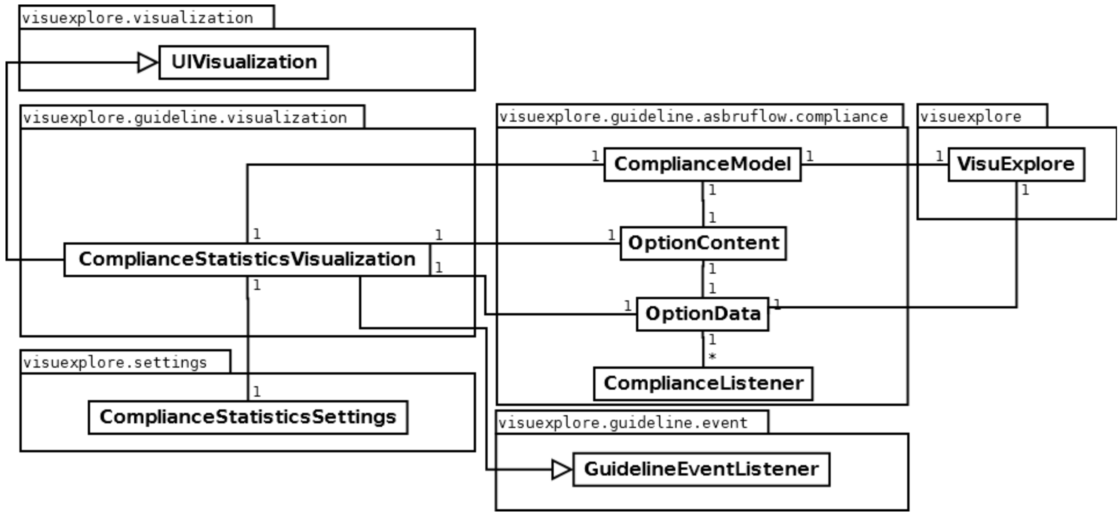

Figure 4.19: Classes providing the functionality for the reintegrated Compliance Statistics facet.

The Visualization class itself (*ComplianceStatisticsVisualization.java*) can be found in the package *visuexplore.guideline.visualization*. It contains an object of the type *OptionContent.java* which remained unaltered and was taken from the *CareCruiser* prototype. It provides the controls, panels and interactive functionality for the facet. The static main class *VisuExplore.java* provides the singleton instances of the *ComplianceModel.java* and the *OptionData*.

The *ComplianceStatisticsSettings.java* instance enables ComplianceListeners to subscribe, which need to react on selections made in the controls of the facet.

Streaming Boxplots

We described the Streaming Boxplots visualization, the Semantic Zoom package provides, in section 4.3. For reasons of clarity and comprehensibility we did not include all classes in the UML diagram presenting the relationships in figure 4.20. We especially left out axis and label renderers and focus on classes used for the visualization. Also some relationships were reduced and are not UML compliant (if a relationship is drawn from a class to a package, the class has a relationship to each of the contained classes of the package).

The class *SemanticZoomVisualization.java* which is derived from *TimeVisualization.java* manages all components needed. On start-up the settings are initialized (*SemanticZoomSettings.java*

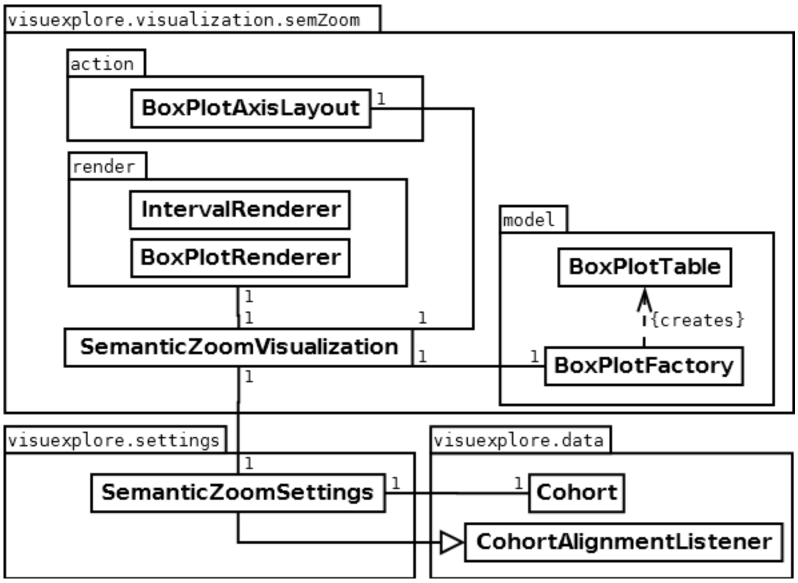

Figure 4.20: The Semantic Zoom package, that contains the Streaming Boxplots implementation, with all relevant classes and relationships.

in *visuexplore.settings*) and contain the Cohort instance with the data to visualize. The settings class also implements *CohortAlignmentListener.java*, so it is able react to temporal recalculations of the data in the cohort and highlighting of a patient in the Stacked LifeLines visualization.

At the start, the parameter table of the cohort is given to the visualization, which is aggregated according to the current time interval by *DataTableReaggregator.java*. *SemanticZoomVisualization* then invokes *BoxPlotFactory.java* which calculates all visual items for the box plots calculated from the the given parameter table. The visual items are stored as tuples in an instance of (*BoxPlotTable.java*).

The rendering of the box plots and the intervals is handled in *BoxPlotRenderer.java* as well as *IntervalRenderer.java* in the *visuexplore.visualization.semZoom.render* package.

Stacked LifeLines

The Stacked LifeLines visualization package *visuexplore.visualization.cohort.lifeLines* consists of just a few classes. Some are also reused from the *visuexplore.visualization.semZoom* package described before. Figure 4.21 shows a diagram of the set of classes and their relationships.

LifeLinesCohortSettings.java contains the settings for the visualization, the cohort with the data to visualize and also implements the interface *CohortAlignmentListener.java*, to re-render on temporal alignment and re-aggregation of the data. *LifesLinesVisualizationCohort.java* holds

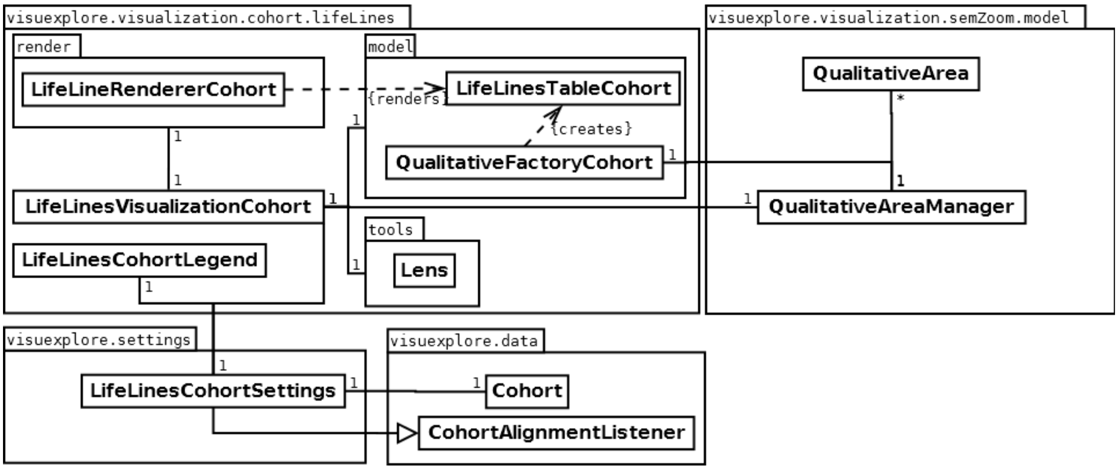

Figure 4.21: The Cohort Lifelines package with all relevant classes and relations, reusing the qualitative area classes from the Semantic Zoom package.

references to instances of all other classes. The *QualitativeFactoryCohort.java* class conducts the temporal abstractions and calculates the LifeLines for each patients, storing them in a newly created instance of *LifeLinesTableCohort.java*. The visualization reuses the *QualitativeArea.java* and *QualitativeAreaManager.java* classes from the Semantic Zoom package to enable the use of the same qualitative areas for parameters. LifeLines stored in the instance of *LifeLinesTable.java* are rendered by *LifeLinesRendererCohort.java*.

The functionality for the magic lens is provided within the *visuexplore.cohort.lifeLines.tools* package by the class *Lens.java*. It implements the *MouseInputAdapter.java* interface and is registered with the Display of the visualization to react on mouse events for interaction with the lens.

Alpha Blended Plan Execution and Compliance

The alpha blended compliance visualization is derived from the plan execution visualization for single patients. Only the loading stage has been altered to join all single step graphs of the cohort. In this loading process items that occurred at the same instance in time and are of the same type are joined and counted, to reduce the amount of visual items and improve the performance of the visualization. The edges are also removed from the graphs, since keeping them created cycles within the graphs, which resulted in high computational effort.

Figure 4.22 shows an overview of classes providing the functionality for alpha blended compliance visualization of a cohort.

Unlike in the visualizations described before, actually two Settings classes for this visualization

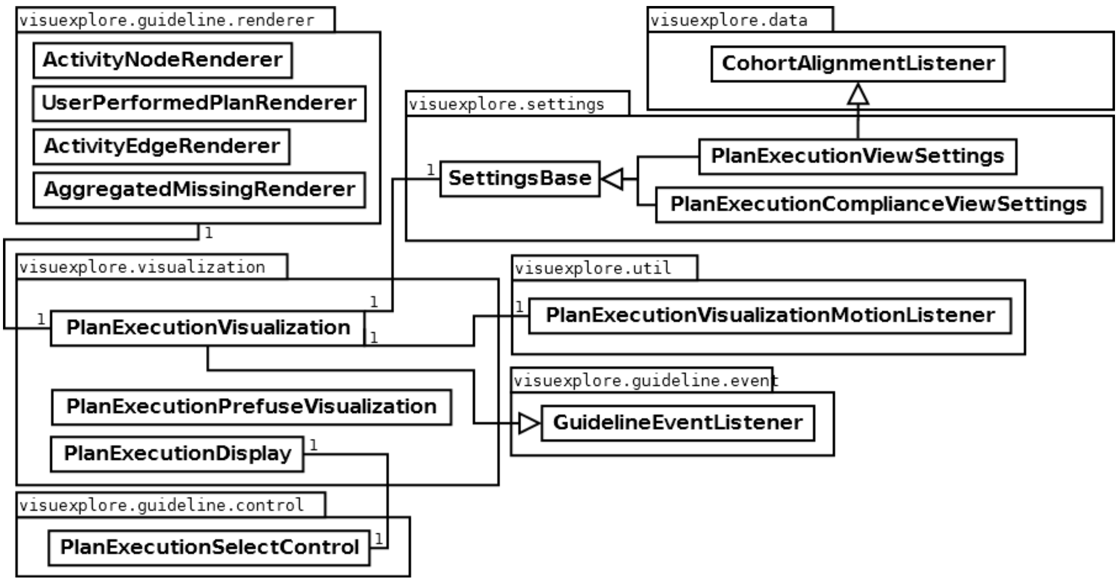

Figure 4.22: Packages and classes used to provide the functionality for the Alpha Blended Compliance visualization.

(*PlanExecutionViewSettings.java* and *PlanExecutionComplianceViewSettings.java*) exist. The difference between them is that, depending on which one is instantiated, compliance information is visualized or not. Switching between the techniques is achieved by just setting a single boolean in the constructor of the visualization to switch between the two modes.

PlanExecutionVisualization.java implements the *GuideLineEventListener.java* interface to react on selections of actions or plans made in other visualizations by highlighting the corresponding visual items. The highlighting and selection is handled in the additional class *PlanExecutionDisplay.java* for this package, which extends the *TimeDisplay.java* class used by the other time oriented visualizations. In the package the Visualization class is also extended by *PlanExecutionPrefuseVisualization.java* which holds the data structure to be visualized.

The functionality of adjusting the alpha values by scrolling the mouse wheel, is provided by the class *PlanExecutionVisualizationMotionListener*. It extends MouseInputAdapter from the Java SE framework for receiving the according mouse events.

A variety of renderer classes handler the different visual items. The single edge in figure 4.22 between the *PlanExecutionVisualization.java* and the renderers specifies that the visualization contains one reference to an instance of each renderer, which are reused from the *visuexplore.guideline.renderer* package.

The visualization also needs to redraw on temporal alignment of data and highlighting of a patient selected in the Stacked LifeLines visualization. Therefore the *PlanExecutionViewSettings.java* class implements *CohortAlignmentListener.java*.

Aggregated Plan Execution and Compliance

The last set of classes we describe, handles the time oriented visualization of aggregated compliance and is mainly located in the package *visuexplore.visualization.cohort.compliance*. An overview of the modules and their relationships is presented in figure 4.23.

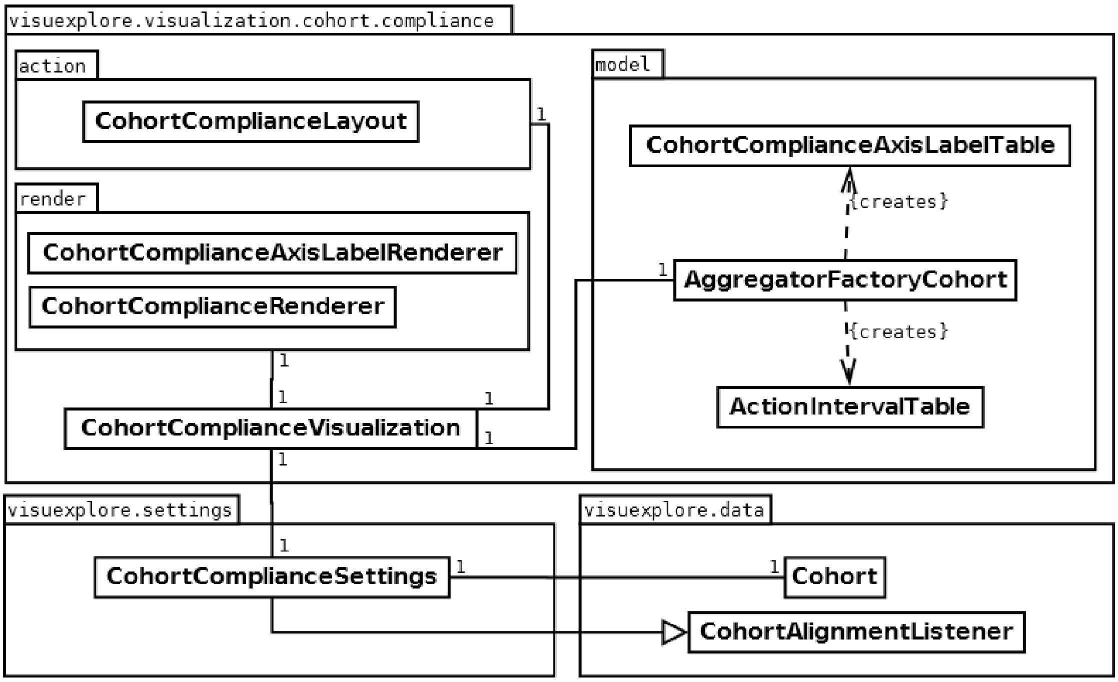

Figure 4.23: The Cohort Compliance package with all relevant classes and relationships.

Initialization and start-up is handled in the corresponding settings class *CohortComplianceSettings.java*. It is constructed and invoked with instance of *Cohort.java*, containing the plan execution and compliance data to be visualized. The class obtains all execution graphs of the patients and also builds an overview graph needed to identify all actions that can occur for visualizing them in the legend. It also implements the *CohortAlignmentListener.java* interface to rebuild the visualization if temporal alignment or re-aggregation of times are executed.

The main class for the visualization is named *CohortComplianceVisualization.java*. On instantiation with a joined action table it instantiates the factory class *AggregatorFactoryCohort.java*, which builds both a table with the tuples containing the information about aggregated actions in the intervals (*ActionIntervalTable.java*) and a table for the items in the legend which consists of all actions in the plan (*CohortComplianceAxisLabelTable.java*).

The factory method is able to aggregate actions for any size of intervals to any given reference date, obtained from the *Cohort.java* instance. The start of the first interval is set to the date before the earliest action found corresponding to the reference date minus the necessary amount of time spans needed to find an earlier date. An additional time interval before this point is added, while actions exactly executed at this starting point are counted within this interval.

The rendering of visual items or tuples, present within the created tables, is handled by the corresponding renderer classes *CohortComplianceRenderer.java* and *CohortComplianceAxisLabelRenderer*. The second class is responsible for drawing a diamond for each possible item in the legend. In the first renderer the visual items from the aggregated actions table are drawn. The opaqueness is determined by the total amount of actions in this interval, while the size is determined by the number of patients.

For the layouting of aggregated compliance items we used the standard *TimeLayout.java*. We needed to implement the additional layout: *CohortComplianceAxisLabelLayout.java* for a proper layouting of the legend's labels on the left display of the facet.

CHAPTER 5

Evaluation

This chapter describes concepts and the exact method we used for the evaluation of our prototype. The observations and results gained from this chapter are especially relevant for answering research question number 4 in section 1.4.
In section 5.1 Evaluation Techniques for Information Visualizations we provide a short survey of techniques for information visualization evaluation we found during our literature research. We follow up by giving a detailed description of the evaluation method we used in 5.2 Evaluation Method and conclude the chapter with the results we obtained from the evaluation procedure in 5.3 Results.

5.1 Evaluation Techniques for Information Visualizations

A detailed survey of evaluation techniques would exceed the scope if this thesis. We try to give insight about the variety of approaches for the evaluation of information visualizations by briefly describing selected methods. The selections are intended to give an indication of the variety of the methods.

Evaluation of visualization methods is also a non-trivial task. It is hardly possible, to assign exact measures to obtained results, while the results themselves are often also not generalizable. While an comparison approach can be used if more than one method is present, the evaluation of novel approaches with no comparable alternative is even harder to conduct, since no referenceable results or studies exist.

Also a variety of methods exist, trying to evaluate different attributes of the given visualization. In [40] Plaisant et. al. identified four different general classes of evaluation methods:

1. Controlled experiments comparing design elements

2. Usability evaluation

3. Controlled experiments
4. Case studies of tools in realistic settings

Methods can also be distinguished by the amount of test subjects they require to be executed. Heuristic or analytical methods can be performed with one to just a few test subjects, while empirical studies require a larger number of test users to acquire a significant amount of data points.

Both types of methods are used frequently. When selecting one of them the choice should also be made according to the stage of the development. A large scale empiric evaluation during the design phase will not be reasonable. The creators of the visualization are at this stage more interested if the visual elements and metaphors in the visualization can be understood at all and if it is therefore worth to continue the development. However for the comparison of two existing tools an empirical approach is more appropriate. It can be used to provide a significant amount of data points which are statistically comparable.

In [30], Lam et. al. present a scenario based look at empirical studies in information visualization. They did an extensive survey of over 850 publications in the field of visualization. They found 361 papers which contained evaluations. Afterwards they categorized them and identified seven different main scenarios for evaluations. Suitable evaluation questions and methods are presented for each of the seven scenarios, including their goals and possible output.

A very commonly used and well known analytical approach for evaluation of visualization techniques was described by Lewis in [31], which he called the Thinking Aloud method. Today the method is used in product design, development and many other fields to gather data about the usability of product prototypes. During conducting the Thinking Aloud method the test subject is asked to perform a specified set of tasks. During the process of solving these tasks, the user is asked to think aloud and express his thoughts and everything he hears sees and feels. The expressed thoughts are recorded in some manner and can be analyzed after the evaluation to gain ideas about the usability of the product or in our case the visualization.

In [71] Zuk et. al. conducted a case study in which they compared three different heuristic methods against certain aspects. In general the three methods approach the problems from different directions, also different obstacles and problems are encountered. The authors also state, that some methods are largely dependent on the level of domain specific knowledge brought in by the selected test subjects. Therefore the selection of test subjects has a high impact on the results of the evaluation.

A method for large scale empirical evaluations of visualization methods was introduced by Shneiderman et. al in [57]. It is named *“Multi-dimensional In-depth Long-term Case studies“* (MILCs). Such a study is conducted over a long period, and starting it in early stages of development of the tool is a feasible strategy. The aim is to gain detailed insight about the usability of techniques by observing a small number of individuals working on problems in their normal environment and use the tool to solve them [57]. These individuals should be domain experts, which are the intended target user group for the tool. It is possible for the study period to span over 1-3 years for ambitious projects, while a multitude of basic evaluation techniques can be used to assess different aspects of the tool during this time. The authors also suggest that an increased number of test users is beneficial since more data is acquired. This type of study is likely one of the most beneficial ones, since insight into usability is gained over long periods. However it is also one of the most expensive to conduct, since continuous meeting, training and evaluation with the individuals needs to be done.

5.2 Evaluation Method

Due to the experimental character of our prototypes and the scope of this thesis, we opted to conduct a very small scale study with just a single test user. The techniques we developed represent early prototypes. At early prototyping stages, we consider an analytical method more appropriate for an evaluation considering the literature described above.

We continually evaluated and improved the techniques during the development and implementation phase. We however had no access to domain specific expert knowledge during this stage. Our prototype has a very specific intended application area for analysis of trends within treatment data of a cohort of patients. Therefore we opted to select a domain expert, to test them and receive feedback about their usability from him. This can for example be a practising doctor or soon to graduate medicine student.

Evaluation Procedure and Tasks

The evaluation procedure consisted of the following steps:

- **Introduction:**
  - Explain the purpose and motivation of this work
  - Introduction to general concepts
  - Ask for previous experience with visualization tools and computers
  - Explanation of visual encodings
  - Explanation of the test procedure
- **Test**:
  - Introduction to the Interface
  - Solving of presented tasks
  - Exploratory testing of the software
- **Interview and Summary**:
  - Issues in the prototype and techniques

- Ideas for improvement
- Overall impression

At the beginning of the evaluation we introduce the user to the methods and concepts in this work, like information visualization and visual analytics. We present the previous state of the art, our motivation for this work and what we aim to achieve. Since the user is an expert in the medical domain, he is acquainted with the medical concepts, while he is not familiar with information visualization or visual analytics. After explaining all concepts, we ask him if he had any previous contact with visualization tools and if he used them in the past. We will also ask him for his experience with computers and for which purposes he uses them to gain an idea of his overall experience. As next step we give an introduction to the techniques and visual encodings used to visualize these concepts in the past. We conclude the introductory phase by explaining the procedure of the following test.

In the second phase of the evaluation we conduct the actual user test. We start up the prototype and introduce the test person to the elements of the interface. Afterwards the mouse is handed to the expert. From this point on the screen as well as the speech of the user is recorded. Then he is asked to solve the set of tasks presented in User Tasks, while sharing his thoughts and feelings during the process. Each single task consists of a description that depicts what the user should do. The preconditions that need to be fulfilled before the task can be started are also specified. We also state the expected result, that should be achieved if the user properly solves the task. After finishing a task, the user is asked if he could solve the task as he expected or if he wanted to approach the task in a different manner and we write down or state unexpected approaches. Also everything else that influences results is recorded by us, for example, if we noticed that the user had any kind of problems while solving the tasks. After solving all tasks, the user is allowed to freely explore the interface as long as he wants to.

As last stage of the evaluation we carry out a concluding interview, where we mainly ask the user about his overall impression of the prototype and visualizations. Furthermore we ask him if he has any additional ideas for extending the techniques and if he is able to identify future research possibilities. He is also motivated to mention any additional things that attracted his attention in a positive, as well as a negative manner.

Equipment

As test set up we used a single laptop, which we connected to a large 24' monitor, to increase the available screen space and resolution for the evaluation. Furthermore we also used a pen and paper to draw and illustrate some concepts if necessary. The screen was captured by a software installed on the laptop while the speech was recorded with a smart phone.

Test Data

While we attempted to obtain parameter measurement and action execution data for real treatments, we did not receive them until the day of the conduction of this evaluation. For the guideline we used a demo guideline intended to recommend the clinical strategy in case of an elevated risk of atrial fibrillation for a patient. Also no suitable treatment data related to this guideline could be found. To be able to perform the evaluation we created a dummy data set. This set contained data points that enabled the unambiguous resolution of all tasks, which would maybe would not have been possible for a real data set. Using this constructed data set with certain characteristics however suited our evaluations design approach. We generally aimed to focus on evaluating the visual metaphors, mappings and items we used.

5.3 Results

The person who performed the evaluation for us was found in the extended personal environment of the author. He was working on his thesis for his degree in medicine at the time of the evaluation. We were occupied with the evaluation for about 100 minutes and used all the equipment described above.

As a medicine student he was familiar with the concepts of clinical guidelines and cohorts, which we did not need to explain to him. Furthermore he even was familiar with the sample guideline for atrial fibrillation we used and also had expert knowledge within this area. We explained the other concepts to him briefly. These covered information visualization, visual analytics and compliance to guidelines and the plan representation language Asbru. When asked for his previous experience with visualization tools, he responded that he does not have any experience with visualization tools in medicine. Finally we explained the test procedure and introduced him to the basic functions of the interface.

User Test

We started the user test with the first two tasks in User Tasks, that were intended to check if the user can grasp and get an idea of the overall state of the cohort within the intervals with and without the lens. Both tasks share the same goal. The difference was, that in the second task the user was allowed to use the magic lens. In the first task the user identified two intervals that could contain the best overall state, which actually met our expectations, since there were two intervals that almost contained the same overall state. At the beginning of the second task it was unclear to the user what the magic lens was and its functionality. He tried to use zoom in button at first. After explaining that the magic lens is depicted by the black lines, he dragged the lens over all intervals and identified the single interval with the best state. He however kept the initial size of the magic lens and did not resize it over the whole time frame of the parameter recordings. After further explaining the functionality he also resized the lens as intended. We subsequently asked him if there was any way in which he could more easily grasp the available interactions with the lens. He was however not able to express any ideas.

Tasks 3 through 5 were used to test the expressiveness of the Streaming Boxplots visualization. At the beginning the test subject himself identified the visualization as boxplots, since he was familiar with the method from his studies. The user was easily able to solve the three tasks in a short amount of time. This suggests that boxplots are also usable in this extended form, for a consecutive set of intervals.

Task 6 was devised to see if the user could grasp the concept of overlapping items by alpha blending and identify the largest cluster of items. While at first he was confused by the mix of aggregated and non-aggregated visualizations, he was able to identify the time instance where the same action has been conducted for all patients.

We followed by tasks 7 and 8 with which we intended to check if the user could understand the visual mappings in the Aggregated Compliance visualization. He quickly noticed that the distinct types of actions were split along the y-axis and pointed to the legend on the left. Initially he did not understand the visual mappings, while after some checking of the detailed values in the tool tips he found out that the largest diamond corresponded to the highest amount of involved patients in task seven. For task 8 he looked for missing action instances in the Alpha Blended Compliance visualization. This actually showed that he made the connection between the blended and aggregated visualization and was aware that they contained the same data. He did however not intuitively grasp that the distance of the lines to each other was calculated from the missing actions. He however moved over the interval and found the tool tip where no visual item was shown, which showed that he understood that there was a data point.

Task 9 was intended to test if the user could align all treatments to the first occurrence of an action, that is part of the the guideline, by himself. He quickly identified the toggle button, but did have no idea what to do afterwards. He tried to align actions by selecting an executed action instance in the plan execution view. After pointing him to the structural guideline overview, he was able to make the connection. This however suggests that additional interactivity with items in the Alpha Blended Compliance visualization and the Aggregated Compliance visualization could help users to achieve the second step necessary for alignment. The concept of alignment and what happened to the treatments were understood easily.

Tasks 10 and 11 were used to check if the expert understands the temporal concepts in the prototype. He was instantly able to state that the current interval and the correct ending time of parameter measurements relative to the first occurrence of the selected action.

The final two tasks (12 and 13) were used to check if the reintegrated statistical compliance information can also be used with all executed actions from a cohort as input. At first the user checked the Aggregated Compliance visualization, while he quickly noticed the counts in the statistics facet below. Then he correctly stated the amount of missing actions in the cohort. In task 13 he instantly moved the mouse to the structural guideline view and checked the tool tips of the distinct action types for counts. While he did not immediately point to the correct action, he checked all actions that had instances of missing actions and was able to state the correct one.

Interview

After the user finished solving all tasks we started a short concluding interview. We first asked the expert for his general opinion on the prototype and techniques. He stated the he liked the overall structure of the prototype and visualizations and thinks the architecture is logically structured. He noted, that he would have liked a longer introduction to the techniques and encodings. When asked, he repeated that he would have liked tool tips for the Plan Execution visualization. Such a tool tip should state the exact execution time of the action and the amount of stacked action instances. He explained that exact times could be relevant in case of time sensitive guidelines.

Next we asked him if he would have liked additional interactivity other than the tool tips. He answered that it would be better to keep prototypes simple, since the techniques might not be universally applicable. If he had to make himself familiar with other prototypes he would like that each of them is as simple as possible. Next we showed him the highlighting of patients, which was not part of the evaluation initially because of its yet more experimental character. He also understood the functionality easily and imagined that it could be useful if he would like to see details. He further noted that some substitute element should be present in an interval of the Aggregated Compliance visualization that shows interaction is possible for an interval that contained only missing actions.

Finally he stated that he thinks the techniques are logical and usable as is, even if they are still in an experimental state. He however showed concerns about their universal applicability to all different types of clinical guidelines.

Summary

The evaluation showed that the medical expert was easily able to recognize the medical concepts like guidelines and cohorts in the prototype. He however had no previous experience with visualization tools and did not identify the meaning of the visual mappings at first glance. Yet after short introductions he was always able to solve the tasks quickly and understand our intended ideas behind the visual metaphors.

Both techniques for the visualization of the development of measurements values turned out to be intuitive for the expert. He was able to work with the Stacked LifeLines instantly, while he also appreciated the magic lens after explanation. Due to his studies he was also familiar with boxplots and understood the aggregation in the distinct intervals of the visualization. Both of the compliance visualizations took him longer to understand. We think, that this results out of the lack of any comparable visualizations or previous techniques, which are intuitive to the brain like the mapping of time to the x-axis. He however noticed that the same data is presented in different forms in both views and used the two of them in combination to solve one of the tasks. On the contrary he very easily understood the time related concepts within the prototype, which included alignment, relative times and time intervals used. The solution times for the tasks related to statistical compliance visualization in the structural guideline view as well as the statistics facet were very short, which we think proves that these techniques are equally applicable for cohorts.

The most substantial issues the expert mentioned during the evaluation were that he missed tool tips for the plan execution view and that the prototype and techniques can possibly not be generalized to all clinical guidelines. We note that all guidelines, that are representable in the Asbru plan specification language, should also be analyzable in our prototype. The general applicability of the prototype is therefore linked to the expressiveness of Asbru and the verification for which types of guidelines the prototype is usable is subject to future research.

CHAPTER 6

Discussion and Future Research

In the first part of this chapter (6.1Discussion) we answer our research questions and try to give reason to them by use of the developed prototypes and the evaluation conducted with them. The second part (6.2Future Reserach) present possibilities and ideas for future research.

6.1 Discussion

While we presented a multitude of approaches to the research questions in this thesis, we were not able to place all of the information into one single time oriented visualization. A proper analysis of the treatment data as we asked for in the main research question is, considering the development and evaluation, possible by using a combination of multiple techniques whereby each one handles a different problem. The first division we made was by the type of data in which we split the parameter measurement data from the executed actions and compliance to them. Splitting these sets made it possible to divide the task into two smaller problems, which we could tackle with different concepts. We made a further distinction by approaching the visualization of each of the data sets with aggregation and without. Both aggregated visualization are condemned to loose the detail about a single patient, while the non-aggregated version are able to visualize details to some degree. Therefore the main research question was answered by providing a multitude of visualizations that are time aligned on the same axis and are stacked above each other for simultaneous visual analysis of the data.

Research questions 1 and 2 can be answered considering the conducted evaluation. The test subject was easily able to understand the temporal abstractions into qualitative areas that are implemented in the Stacked LifeLines visualization. He also easily understood the aggregation of the data of all patients into time intervals of equal duration. Therefore the temporal aggregation of data represents a suitable approach to make the data points comparable and analyzable. The Streaming Boxplots visualization turned out to be an even more promising approach, since the medical expert recognized that it consists of a series of box plots, which is also well known to visualize statistical information in the medical domain. The compliance visualization methods were more difficult to understand for the subject. This most likely stems from the fact, that there is only limited previous knowledge of compliance visualization as well as techniques approaching the problem. In the evaluation the mappings used were understandable and mostly intuitive, while comparable techniques do not exist. Also unlike in the Stacked LifeLines visualization, overlaying the actions by alpha blending also loses the details for the single patient. Overall we consider the methods to be usable for the analysis of compliance to the guideline within the cohort.

Research question 3 is answered by a look at the second task related to the magic lens as well as the question regarding the highlighting of patients. Considering that the user was able to distinctly identify the interval with the best overall state of the whole cohort only after use of the magic lens, it is a valid approach for providing insight into the overall progress at the loss of the exact parameter development of a single patient. The Alpha Blended compliance visualization in fact loses the detail for a single patient, because some items overlap each other and it is not clear when the actions for a selected patient were executed. The interaction technique for highlighting a single patient presented in section 4.3, enables the retrieval of some details in all the presented visualizations and therefore also seems like a promising approach.

For answering research question number 4 we again need to consider the conducted evaluation. The results indicate that the presented and implemented techniques are well suited for the analysis to the problem. However they are in an early experimental stage and the evaluation also spawned some ideas for improvement of the techniques as well as some issues. Considering that the expert instantly recognized the medical concepts in the prototype the visualizations are well suited to the problem, since he was able to easily identify the concepts and most of the ideas that led to the visual encodings.

As stated before a minor issue was that the user expected details on demand within the Plan Execution visualization by tool tips. A possible larger issue he mentioned is that the software could not be generalizable to all types of guidelines, since the complexity of some might bust the expressiveness of the used visualization techniques.

While we were able to gain interesting data from the evaluation with a single medical experts, the small scale of the conducted evaluation yields the issue of a smaller significance of the study. Finding a larger variety of test subject might result in the finding of other and more diverse issues in the developed techniques and the prototype.

6.2 Future Reserach

During our research and the evaluation we were able to identify the following suggestions for future research:

- **Handling of multiple cohorts:** While we focuses to integrate visualizations for a single cohort in this work, the integration of the visualization of multiple cohort also seems to be promising. A simple approach is to just load two or more cohorts and stack visualizations of them above one another to, for example, compare the development of parameters. A yet more elaborate method would be to show correlations or differences between the cohorts in a single visualization or facet. The Caregiver prototype we presented in section 3.3 already enables comparison of different cohorts.

- **Temporal abstractions methods:** From all methods for the temporal abstraction of parameters we presented in section 3.3, we just implemented a simple version of LifeLines. While types of temporal abstractions that do not change their height and the same color is shown on all of the vertical space seem easily usable, the intended order inside the magic lens is harder to define. Height encoded techniques use more screen space and are unable to convey the same amount of information at equal screen size. Techniques in which also use the vertical space to encode information like Horizon or Qualizon Graphs will theoretically need an even larger amount of screen space than all other techniques.

- **Alignment methods**: For the temporal alignment of data within our prototype, we only implemented the basic technique of aligning the data to the first occurrence of an action. Many other and more complex methods have been presented in the literature we found during our survey. Furthermore we also never altered the time spans between two actions. One could consider to examine the parameter development between the execution of two actions and distort the time spans between them to gain equally sized intervals in terms of screen space.

- **Long term usability study of the visualizations:** While we conducted a low scale evaluation described in chapter 5, a long term usability study like Shneiderman et. al. described in [57] helps to further improve the techniques and make them more accessible for a broad range of user.

CHAPTER 7

Conclusion

Clinical Practice Guidelines (CIGs) are used to provide the current best known evidence from medicine and research to provide recommendations for the treatment of patients, conveying a specific sickness. The integration of CIGs in their computerized from Computer Interpretable Guidelines (CPGs) into clinical practice might yield several benefits for all involved parties. Physicians can follow the recommendations and have a clear guideline, while the overall treatment quality is improved with the continuous improvement of guidelines. Risk for patients might be reduced, while the satisfaction could be increased. Also the overall healthcare system might benefit since unnecessary parts of treatments, like too many drug applications, might be identified and rendered obsolete. For the improvement of a guideline it is often necessary to examine data obtained from a large number of treatments. These group of patients, that received the same treatment as recommended by the given guideline can be considered as a so called cohort. The amount of treatment data recorded for all patients within the cohort is often vast and heterogeneous in terms of treatment times and time intervals within the data. The data also consisted of two distinct sets, measurement data of clinical parameters as well as the executed actions and their compliance to the given guideline. An analysis of the data in textual form seems quite challenging and even impossible for some instances. The goal of this work was to extend state of the art techniques and develop new ones that approach the problem by visualizing the raw treatment data. Furthermore the user should be enabled to interact with the visualizations, to gain further insight into the attributes of the data from different perspectives.

Before we actually approached the problem, we conducted a survey of previous work that contained approaches for the visualization of larger group of patients. We selected five systems that were of interest to our work and also discussed previous work that was done on the visualization of parameter measurements as well as compliance to a guideline. The selected systems were presented in detail, while we also discussed them according to the features that we considered to be interesting for our work. While some of the systems visualizing data of cohorts showed promising approaches and concepts for the visualization of parameters and other attributes for multiple patients, none of the creators of them made attempts to integrate the visualization of guidelines, guideline execution or compliance. The systems visualizing data for single patients however contained related approaches to all of these problems, while they did not attempt to cover the visualization of multiple patients.

To approach the problem we integrated the state of the art visualization techniques for guidelines, guideline execution and compliance as well as the initial techniques for visualizing a patient's parameter measurement into the VisuExplore prototype, which has the advantage of a superior and easier extendable software architecture. As a next step we also reintegrated the statistics facet developed by Peter Bodesinsky from the CareCruiser prototype and altered both of his techniques to show accumulated compliance data for cohort. For both distinct data set in the cohort, which are parameter measurements and guideline execution and compliance data, we developed an approach that still contains all the information and a second one that allows aggregated visualization of the data.
For the non-aggregated visualization of parameter measurements over time we developed the Stacked LifeLines visualization. A single LifeLine is computed for each patient, showing the development of parameter measurement values and stack the LifeLines for all patients above each other. The aggregated version is called Semantic Zoom and aggregates the information within defined time intervals. It shows a series of boxplots called Streaming Boxplots. Each single boxplot shows the largest, lowest, 0.25 quantile, 0.75 quantile and median for the values of all patients in each interval.
We also extended the first version of guideline and compliance execution from the initial version developed by Peter Bodesinsky. Each action of each patient is visualized, while the items are made transparent. The larger the cohort the more transparent each item gets. Time areas where actions were executed frequently will show interesting information about the spread of applied actions. The aggregated method vertically split the view and assigns an equal amount of space to each distinct type of action. For each type of action in each interval of all treatments aggregated counts for valid and missing actions are computed. Also the number of patients for which actions were conducted in the interval is accumulated. The valid actions are visualized by diamonds, whereby the size scales according to the amount of patients involved, while the opaqueness is linearly related to the count of maximum actions in all intervals.
Since the data sets are as stated heterogeneous we integrated methods to aggregate the data points according to time intervals given by the selected timescales in the prototype. The presented time unit always specifies the aggregation intervals for the aggregated visualizations of parameter measurements and guideline execution. Since treatments start at different points in time, keeping the data in absolute times might make the analysis infeasible. We provide a method all data points within the cohort relative to the first occurrence of a distinct type of action within each treatment.

To assess the quality of our newly developed techniques we conducted a small scale qualitative evaluation with a single medical expert. He was quickly able to identify the medical concepts in the application and grasp most of the visual metaphors easily. Also the new time related concepts like aligning treatments and fixing all aggregation intervals to the current time unit were clear to him. Some techniques and encoding were however not initially comprehensible for him.

He stated, that a short introduction would solve the problem for him.

In the last part of thesis we discussed if the techniques are applicable to solve the research problems, considering the results we obtained from the evaluation. We conclude that, in general, the techniques are applicable. Some minor issues were raised during the evaluation which can be solved by improving the prototype. Finally we gave short overview of future research possibilities we identified during our work. These include the handling and comparison of multiple cohorts, additional methods for temporal abstraction and alignment and a long term usability study of the prototype and visualizations.

APPENDIX A

User Tasks

This appendix provides descriptions of the tasks, which were used during the second phase of the evaluation described in section 5.

Task 1

Task Description:	Try to identify the time interval, in which the visualized parameter of the cohort has the best overall quality. Achieve this without using the magic lens.
Preconditions:	Constructed cohort data is loaded. The guideline and all developed visualizations are visible. The lens is visible, but is shown to the left of the start of visualized parameters. VisuExplore is in absolute time mode and 15 minute intervals.
Result:	The test user points to the time interval, in which he thinks that the cohort is in the best overall state.

Task 2

Task Description:	Try to identify the time interval, in which the visualized parameter of the cohort has the best overall quality. Achieve this with help of the magic lens.
Preconditions:	Constructed cohort data is loaded. The guideline and all developed visualizations are visible. The lens is visible, but is shown to the left of the start of visualized parameters. VisuExplore is in absolute time mode and 15 minute intervals.
Result:	The user resizes the lens to cover the whole range of parameters and points to the time interval, in which he thinks that the cohort is in the best overall state.

Task 3

Task Description:	Try to identify the time interval with the lowest overall parameter measurement value in the Semantic Zoom visualization.
Preconditions:	Constructed cohort data is loaded. The guideline and all developed visualizations are visible. VisuExplore is in absolute time mode and 15 minute intervals.
Result:	The test user points to the interval with the lowest overall parameter measurement value of the visualized parameter.

Task 4

Task Description:	Try to identify the time interval with the highest median parameter measurement value in the Semantic Zoom visualization.
Preconditions:	Constructed cohort data is loaded. The guideline and all developed visualizations are visible. VisuExplore is in absolute time mode and 15 minute intervals.
Result:	The test user points to the interval with the largest median parameter measurement value of the visualized parameter.

Task 5

Task Description:	Try to identify the time interval with the smallest spread in parameter measurement values in the Semantic Zoom visualization.
Preconditions:	Constructed cohort data is loaded. The guideline and all developed visualizations are visible. VisuExplore is in absolute time mode and 15 minute intervals.
Result:	The test user points to the interval with the smallest spread in parameter measurement value of the visualized parameter.

Task 6

Task Description:	Point to the time instance at which the highest amount of actions has been conducted in the Alpha Blended Plan Execution Visualization.
Preconditions:	Constructed cohort data is loaded. The guideline and all developed visualizations are visible. VisuExplore is in absolute time mode.
Result:	The test user points to the instance where the highest amount of actions has been conducted in the plan execution visualization.

Task 7

Task Description:	Point to the action type and interval, where the highest amount of patients has been involved in the Aggregated Compliance visualization. Also state the number of patients that contributed to the visual item.
Preconditions:	Constructed cohort data is loaded. The guideline and all developed visualizations are visible. VisuExplore is in absolute time mode and 15 minute intervals.
Result:	The test user points to the distinct action and time interval where the highest amount of patients has been involved. The tool tip pops up and he states the exact number of patients.

Task 8

Task Description:	Point to the action type and interval, where the highest amount of missing action instances occurred. Also state the exact number of missing actions within the interval.
Preconditions:	Constructed cohort data is loaded. The guideline and all developed visualizations are visible. VisuExplore is in absolute time mode and 15 minute intervals.
Result:	The test user points to the distinct action and time interval where the highest amount of missing actions instances occurred. The tool tip pops up and he states the exact number of missing action instances.

Task 9

Task Description:	Switch into the relative time mode by temporally aligning all data to the first occurrence of action type: $AF\_notification$.
Preconditions:	Constructed cohort data is loaded. The guideline and all developed visualizations are visible. VisuExplore is in absolute time mode and 15 minute intervals.
Result:	VisuExplore is in relative time mode and all visualizations are aligned to the first occurrence of action type: $AF\_notification$.

Task 10

Task Description:	Look for and state the currently used time interval.
Preconditions:	Constructed cohort data is loaded. The guideline and all developed visualizations are visible. VisuExplore is in relative time mode and 15 minute time intervals.
Result:	The user states the current time unit, he maybe points to the time scale header.

Task 11

Task Description:	Point to the time at which all parameter measurements have ended. State the relative amount of time that has expired to the reference point.
Preconditions:	Constructed cohort data is loaded. The guideline and all developed visualizations are visible. VisuExplore is in absolute time mode and 15 minute intervals.
Result:	The user states the time relative to the reference point when the last time interval ends.

Task 12

Task Description:	State the overall amount of missing action instances in the cohort and point to the screen coordinates where this information can be found.
Preconditions:	Constructed cohort data is loaded. The guideline and all developed visualizations are visible (Compliance Statistics Visualization needed).
Result:	The user states the total amount of missing action instances in the cohort and points to the stacked bar in the Compliance Statistics Facet.

Task 13

Task Description:	Point to the type of action with the most missing action instances in the structural guideline view. State the amount of missing actions in the cohort for this type of action.
Preconditions:	Constructed cohort data is loaded. The guideline and all developed visualizations are visible (Structural Guideline visualization needed).
Result:	The user points to the type of action with most missing action instances in the cohort. A tool tip appears and the user is able to read and state the exact number of instances.

Bibliography

[1] Medical Dictionary. `http://medical-dictionary.thefreedictionary.com/cohort`. Accessed: 22.11.2014.

[2] Oracle Swing Tutorial. `http://docs.oracle.com/javase/tutorial/uiswing/`. Accessed: 22.11.2014.

[3] Prefuse Manual. `http://prefuse.org/doc/manual/`. Accessed: 22.11.2014.

[4] The Asgaard Project. `http://asgaard.tuwien.ac.at/`. Accessed: 22.11.2014.

[5] A. Advani, K. Lo, and Y. Shahar. Intention-Based Critiquing of Guideline-Oriented Medical Care. In *In Proceedings of the AMIA Annual Symposium*, pages 483–487, 1998.

[6] W. Aigner. TimeViewer/TimeVis - Visualizing Temporal Data using Prefuse. `http://www.cvast.tuwien.ac.at/projects/timeviewer`. Accessed: 22.11.2014.

[7] W. Aigner, P. Federico, T. Gschwandtner, S. Miksch, and A. Rind. Challenges of Time-oriented Data in Visual Analytics for Healthcare. In *IEEE VisWeek Workshop on Visual Analytics in Healthcare*, page 4. IEEE, IEEE, 2012.

[8] W. Aigner and S. Miksch. CareVis: Integrated Visualization of Computerized Protocols and Temporal Patient Data. *Artificial Intelligence in Medicine*, 37(6):203–218, 2006.

[9] W. Aigner., S. Miksch, W. Mueller, H. Schumann, and C. Tominski. *Visualizing Time-Oriented Data - A Systematic View*. Human Computer Interaction, Springer Verlag, 2011.

[10] R. Bade, S. Schlechtweg, and S. Miksch. Connecting Time-Oriented Data and Information to a Coherent Interactive Visualization. *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2004*, New York, NY, USA:105–112, 2004.

[11] P. Bodesinsky. Visualization of Compliance with Medical Guidelines. Master's thesis, Vienna University of Technology, 2013.

[12] P. Bodesinsy, P. Federico, and S. Miksch. Visual Analysis of Compliance with Clinical Guidelines. In *in Proceedings of the 13th International Conference on Knowledge Management and Knowledge Technologies, Graz Austria*, pages 12:1–12:8. ACM, 2013.

[13] D. Brodbeck, R. Gasser, M. Degen, S. Reichlin, and J. Luthiger. Enabling Large-Scale Telemedical Disease Management through Interactive Visualization. *Connecting Medical Informatics and Bio-Informatics*, In Proceedings of the International Congress Eur. Fed. Med. Inform. (MIE):1172–1177, 2005.

[14] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Academic Press, January 1999.

[15] L. Chittaro. Information visualization and its application to medicine. *Artificial Intelligence in Medicine*, 22(2):81–88, 5 2001.

[16] L. Chittaro, C. Combi, and G. Trapasso. Data Mining on Temporal Data A Visual Approach and its Clinical Application to Hemodialysis. *Journal of Visual Languages and Computing*, 14(6):591–620, 2003.

[17] P. de Clercq, K. Kaiser, and A. Hasman. Computer-interpretable Guideline Formalisms. *Stud Health Technol Inform*, 139:22–43, 2008.

[18] S. dos Santos, K. Brodlie, and K. Gaining. Gaining understanding of multivariate and multidimensional data through visualization. *Computers & Graphics*, 28:311–325, 2004.

[19] J. Fails, A. Karlson, L. Shahamat, and B. Shneiderman. A Visual Interface for Multivariate Temporal Data: Finding Patterns of Events across Multiple Histories. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 167–164, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[20] A. Faiola., C. Boston-Clay, J. Jones, M. Downey, and C.M. Newlon. Managing Patient Health Across Diverse Spaces: Using Activity Theory to Model Persuasive Decision Support. In *Proceedings of the CHI2012 Workshop Briding Clinical and Non-Clinical Health Practice Opportunities and Challenges*, 2012.

[21] P. Federico, S. Hoffmann, A. Rind, W. Aigner, and S. Miksch. Qualizon Graphs: Space-Efficient Time-Series Visualization with Qualitative Abstractions. In *proceedings of AVI*, 2014.

[22] P. Groot, A. Hommersom, P. J. F. Lucas, R.-J. Merk, A. ten Teije, F. van Harmelen, and R. Serban. Using model checking for critiquing based on clinical guidelines. *Artifical Intelligence in Medicine*, 46(1):19–36, May 2009.

[23] T. Gschwandtner, W. Aigner, K. Kaiser, S. Miksch, and A. Seyfang. CareCruiser: Exploring and Visualizing Plans, Events, and Effects Interactively. In *Proceedings of the 4th IEEE Pacific Visualization Symposium Hongkong*, pages 43–50, 2011.

[24] K Hinum, S. Miksch, W. Aigner, S. Ohmann, C. Popow, M. Pohl, and M. Rester. Gravi++ Interactive Information Visualization of Highly Structured Temporal Data. *Journal of Universal Computer Science, Special Issue on Visual Data Mining*, 11:1792–1805, 2005.

[25] W. Horn, C. Popow, and L. Unterasinger. Support for Fast Comprehension of ICU Data: Visualization using Metaphor Graphics. *Methods of Information in Medicine*, 40(5):421–424, 2001.

[26] D. Keim, G. Adrienko, J.D. Fekete, C. Goerg, J. Kohlhammer, and G. Melancon. *Information Visualization*. Springer-Verlag Berlin Heidelberg, 2008.

[27] D. Klimov, Y. Shahar, and M. Taieb-Maimon. Intelligent Visualization and Exploration of Time-Oriented Data of Multiple Patients. *Journal of Intelligent Information Systems*, 35:261–300, 2010.

[28] R. Kosara, S. Miksch, P.D. Johnson, and Y. Shahar. AsbruView Capturing Complex, Time-oriented Plans – Beyond Flow-Charts. In *The Second Workshop on Thinking with Diagrams 1998 (TwD-98)*, pages 119–126. The University of Wales, 1998.

[29] K.Wongsuphasawat, J. A. Guerra Gomez, C. Plaisant, T. D.Wang, M. Taieb-Maimon, and B. Shneiderman. LifeFlow: Visualizing an Overview of Event Sequences. In *Proceedings of the 2011 Annuals Conference on Human Factors in Computing Systems (CHI 2011)*, pages 1747–1756, 2011.

[30] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. Empirical Studies in Information Visualization: Seven Scenarios. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1520–1536, September 2012.

[31] C.H. Lewis. Using the "Thinking Aloud"Method In Cognitive Interface Design. Technical Report RC-9265, IBM, 1982.

[32] L. Lins, M. Heilbrun, J. Freire, and C. Silva. VisCareTrails Visualizing Trails in the Electronic Health Record with Timed Word Trees, a Pancreas Cancer Use Case. In J. J. Caban and D. Gotz, editors, *in Proceedings of the IEEE VisWeek Workshop on Visual Analytics in Health Care*, pages 13–16, 2011.

[33] S. Miksch, Y. Shahar, and P. D. Johnson. Asbru: A Task-Specific, Intention-Based, and Time-Oriented Language for Representing Skeletal Plans. In *7th Workshop on Knowledge Engineering: Methods & Languages (KEML-97)*, 1997.

[34] M. Monroe, K. Wongsuphasawat, C. Plaisant, B. Shneiderman, J. Millstein, and S. Gold. *Exploring point and interval event patterns: Display Methods and Interactive Visual Query*. Human Computer Interaction Lab, University of Maryland, 2012.

[35] P. Ordonez, T. Oates, M.E. Lombardi, G. Hernandez, K. W. Holmes, J. Fackler, and C.U. Lehmann. Practical Visualization of Multivariate Time Series Data in Neonatal ICU. *IBM Journal of Research and Development*, 56(5):7:1–7:12, 2012.

[36] S. Panzarasa, S. Quaglini, A. Cavallini, G. Micieli, S. Marcheselli, and M. Stefanelli. Technical Solutions for Integrating Clinical Practice Guidelines with Electronic Patient Records. In *Knowledge Representation for Health-Care. Data, Processes and Guidelines*,

volume 5943 of *Lecture Notes in Computer Science*, pages 141–154. Springer Berlin Heidelberg, 2010.

[37] M. Peleg. Computer-interpretable Clinical Guidelines: A Methodological Review. *Journal of Biomedical Informatics*, 46:744–763, 2013.

[38] M. Peleg, S. Tu, J. Bury, P. Ciccarese, J. Fox, R. A. Greenes, R. Hall, P. D. Johnson, N. Jones, A. Kumar, S. Miksch, S. Quaglini, A. Seyfang, E. H. Shortliffe, and M. Stefanelli. Comparing Computer-interpretable Guideline Models: A Case-study Approach. *J Am Med Inform Assoc.*, 10:52–68, Jan-Feb 2003.

[39] D. Pieczkiewicz, S. M. Finkelstein, and M.I. Hertz. Design and Evaluation of a Web-Based Interactive Visualization System for Lung Transplant Home Monitoring Data. In *Proceedings of the AMIA 2007 Symposium*, pages 589–602, 2007.

[40] C. Plaisant. The Challenge of Information Visualization Evaluation. *ACM*, Proceedings of the working conference on Advanced visual interfaces, New York, NY, USA, 2004. ACM:109–116, 2004.

[41] C. Plaisant, S. Lam, B. Shneiderman, M. S. Smith, D. Roseman, G. Marchand, M. Gillam, C. Feied, J. Handler, and H. Rappaport. Searching Electronic Health Records for Temporal Patterns in Patient Histories: A Case Study with Microsoft Amalga. In *Proceedings of the AMIA 2008 Symposium*, pages 601–605. AMIA, November 2008.

[42] C. Plaisant, R. Mushlin, A. Snyder, J. Li, D. Heller, and B. Shneiderman. LifeLines: Using Visualization to Enhance Navigation and Analysis of Patient Records. In *Proceedings of the AMIA Symposium*, pages 76–80, 1998.

[43] H. Reijner. The Development of the Horizon Graph. In *Proceedings of the Vis08 Workshop from Theory to Practice*, 2008.

[44] A. Rind, W. Aigner, S. Miksch, S. Wiltner, M. Pohl, F. Drexler, B. Neubauer, and N. Suchy. Visually Exploring Multivariate Trends in Patient Cohorts using Animated Scatter Plots. In *Proceedings of the International Conference on Human-Computer Interaction (HCI-I)*, Berlin, Germany, 2011. Springer.

[45] A. Rind, W. Aigner, S. Miksch, S. Wiltner, M. Pohl, T. Turic, and F. Drexler. Visual Exploration of Time-oriented Patient Data for Chronic Diseases: Design Study and Evaluation. In *Proceedings of USAB 2011: Information Quality in e-Health*, pages 301–320, 2011.

[46] A Rind, S. Miksch, W. Aigner, T. Turic, and M. Pohl. VisuExplore: Gaining New Medical Insights from Visual Exploration. In *Proceedings of the 1st International Workshop on Interactive Systems in Healthcare (WISH@CHI2010)*, pages 149–152, New York, NY, USA, 2010. ACM Press.

[47] A. Rind, T.D. Wang, W. Aigner, S. Miksch, K. Wongsuphasawat, C. Plaisant, and B. Shneiderman. Interactive Information Visualization to Explore and Query Electronic Health Records. *Foundations and Trends in Human-Computer Interaction*, 5(3):207–298, 2013.

[48] K. Rosenbrand, J. Van Croonenborg, and J. Wittenberg. Guideline Development. *Studies in Health Technology and Informatics*, 139:3–21, 2008.

[49] A. Seyfang, K. Kaiser, T. Gschwandtner, and S. Miksch. Visualizing complex process hierarchies during the modeling process. In *Business Process Management Workshops*. Springer, 2013.

[50] A. Seyfang, R. Kosara, and S. Miksch. Asbrus Reference Manual, Asbru Version 7.3. Technical Report, 2002.

[51] Y. Shahar. A Framework for Knowledge-Based Temporal Abstraction. *Artifical Intelligence*, 90(1-2):79–133, 1997.

[52] Y. Shahar, D. Boaz, G. Tahan, M. Galperin, D. Goren-Bar, H. Kaizer, L.V. Basso, S.B. Martins, and M.K. Goldstein. Interactive visualization and exploration of time-oriented clinical data using a distributed temporal-abstraction architecture. In *Proceedings of the 2003 AMIA Annual Fall Symposium*, 2003.

[53] Y. Shahar and C. Cheng. Intelligent Visualization and Exploration of Time-Oriented Clinical Data. *Artificial Intelligence in Medicine*, 49:11–31, 2010.

[54] Y. Shahar, D. Goren-Bar, M. Galperin, D. Boaz, and Gil Tahan. KNAVE-II: A Distributed Architecture for Interactive Visualization and Intelligent Exploration of Time-Oriented Clinical Data. In *Proceedings of the workshop on Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP) 2003*, 2003.

[55] Y. Shahar, S. Miksch, and P. Johnson. The Asgaard Project: A Task-Specific Framework for the Application and Critiquing of Time-Oriented Clinical Guidelines. *Artificial Intelligence in Medicine*, pages 29–51, 1998.

[56] B. Shneiderman. The Eyes Have It A Task by Datatype Taxonomy for Information Vizualisations. In *In Proccedings of Visual Languages*, pages 1–8, 1996.

[57] B. Shneiderman. and C. Plaisant. Strategies for evaluating information visualization tools: Multi-dimensional in-depth long-term case studies. In *proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*, 2006.

[58] R. Spence. *Information Visualization*. Pearson Prentice Hall, 2007.

[59] M. Spenke. Visualization and Interactive Analysis of Blood Parameters with InfoZoom. *Artificial Intelligence in Medicine*, 22(2):159–172, May 2001.

[60] M. Stacey and C. McGregor. Temporal abstraction in intelligent clinical data analysis: A survey. *Artificial Intelligence in Medicine*, 39:1–24, 2007.

[61] M. Steenwijk, J. Milles, M. A. van Buchem, J. H. Reiber, and C. Botha. Integrated Visual Analysis for Heterogeneous Datasets in Cohort Studies. In *IEEE VisWeek Workshop on Visual Analytics in Healthcare*, 2010.

[62] P. Terenziani, S. Montani, A. Bottrighi, M. Torchio, G. Molino, and G. Correndo. The GLARE Approach to Clinical Guidelines: Main Features. *Studies in health technology and informatics*, 101:162–166, 2004.

[63] J. J. Thomas and K. A. Cook. *Illuminating the Path - The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Center, 2005.

[64] J. A. Vassalotti. The cycle of development, publication, and implementation of clinical practice guidelines for CKD. `http://www.nature.com/ki/journal/v81/n12/fig_tab/ki201234f1.html`. Accessed: 22.11.2014.

[65] T. D. Wang, C. Plaisant, A. J. Quinn, R. Stanchak, S. Murphy, and B. Shneiderman. Aligning Temporal Data by Sentinel Events: Discovering Patterns in Electronic Health Records. In *Proceedings of the Annual SIGCHI Conference on Human Factors in Computing Systems, CHI 2008*, pages 457–466. ACM, 2008.

[66] T. D. Wang, C. Plaisant, B. Shneiderman, N. Spring, D. Roseman, G. Marchand, V. Mukherjee, and M. Smith. Temporal Summaries: Supporting Temporal Categorical Searching, Aggregation and Comparison. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1049–1056, 2009.

[67] T. D. Wang, K. Wongsuphasawat, C. Plaisant, and B. Shneiderman. Extracting Insights from Electronic Health Records Case Studies, a Visual Analytics Process Model, and Design Recommendations. *Journal of Medical Systems*, 35(5):1135–1152, May 2011.

[68] P. C. Wong and J. Thomas. Visual Analytics. *IEEE Computer Graphics and Applications*, 24:20–21, 2004.

[69] K. Wongsuphasawat and D. H. Gotz. Outflow: Visualizing Patient Flow by Symptoms and Outcome. In *Proceedings of the IEEE VisWeek Workshop on Visual Analytics in Health Care*, pages 25–28, 2011.

[70] K. Wongsuphasawat and B. Shneiderman. Finding Comparable Temporal Categorical Records: A Similarity Measure with an Interactive Visualization. In IEEE, editor, *in Proceedings of the IEEE Symposium Visual Analytics Science and Technology 2009 (VAST 2009)*, pages 27–34, 2009.

[71] T. Zuk, L. Schlesier, P. Neumann, M. Hancock, and S. Carpendale. Heuristics for Information Visualization Evaluation. *In Proceedings of BELIV '06*, pages 55–60, 2006.