



Universität Hamburg
Fakultät für Mathematik,
Informatik und Naturwissenschaften
Fachbereich Informatik
Verteilte Systeme und Informationssysteme

Diplomarbeit

Visuelle Unterstützung bei der EPL-Anfragestellung für Sensordatenströme

Hai-Minh Le

haiminh.le@webleko.com

Studiengang Informatik

Matr.-Nr. 5422322

Fachsemester 16

Erstgutachter: Professor Dr. W. Lamersdorf

Zweitgutachter: Professor Dr. N. Ritter

Dezember 2010

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Zielsetzung	3
1.3	Aufbau der Arbeit	4
2	Grundlagen	7
2.1	Sensornetze	7
2.1.1	Sensor	7
2.1.2	Sensornetz	8
2.1.3	Datenstrom	8
2.1.4	Metainformationen	8
2.2	Komplexe Ereignisverarbeitung	9
2.2.1	Event	9
2.2.2	Complex Event	9
2.2.3	Event Driven Architecture	9
2.2.4	Complex Event Processing	9
2.2.5	Event Processing Language	10
2.2.6	Rule Engine	11
2.2.7	CEP-System	11
2.3	Visualisierung (allgemein)	11
2.4	Visuelle Datenexploration	12
2.5	Zusammenfassung	12
3	Motivation für die Visualisierungsunterstützung bei der Anfragestellung	15
3.1	Das CEP-System Esper	15
3.1.1	Die Esper-EPL	16
3.1.1.1	Klauseln	16
3.1.1.2	Beispiele für grundlegende Esper-EPL-Statements	18
3.1.1.3	Pattern	19
3.1.1.4	Time Windows	20
3.1.1.5	Views	22
3.1.2	Die Esper-Engine	23
3.2	Anwendungsszenarien	24
3.2.1	Waldbrandgefahrenüberwachung	24

3.2.2	Smart Home	25
3.2.3	Weitere Anwendungsideen	26
3.2.4	Gemeinsamkeiten der Anwendungsszenarien	28
3.3	Probleme und Schwierigkeiten einer EPL-Anfrage	28
3.4	Zusammenfassung	30
4	Evaluation der Visualisierungsunterstützung bei der Anfragestellung	31
4.1	Grundlagen der Visualisierung und allgemeine Anforderungen	31
4.2	Spezifische Anforderungen des Visualisierungswerkzeugs	32
4.2.1	Funktionale Anforderungen	32
4.2.2	Nichtfunktionale Anforderungen	34
4.3	Abgeleitete Anforderungen aus Datenverarbeitungszielen	34
4.4	Gesamtanforderungen an das Visualisierungswerkzeug	35
4.4.1	Anforderungskatalog mit Gewichtung	35
4.4.2	Bewertungsmethode	36
4.5	Vorstellung der verschiedenen Visualisierungstechniken	38
4.5.1	Geometrische Techniken	38
4.5.1.1	Scatterplot-Matrizen	38
4.5.1.2	Prosection Views	40
4.5.1.3	Hyperslices	41
4.5.1.4	Parallele Koordinaten	41
4.5.1.5	Parahistogramme	43
4.5.1.6	Spiral Graphs	44
4.5.2	Ikonenbasierte Techniken	45
4.5.2.1	Bubble Graphs	45
4.5.2.2	Stick Figures	46
4.5.2.3	Shape Coding	48
4.5.2.4	Color Icons	49
4.5.2.5	Chernoff Faces	50
4.5.3	Pixelbasierte Techniken	52
4.5.3.1	Einfache Techniken	52
4.5.3.2	Raumfüllende-Kurven-Techniken	53
4.5.3.3	Zwei-Schritte-Techniken	53
4.5.3.4	Recursive-Pattern-Techniken	54
4.5.4	Hierarchische Techniken	56
4.5.4.1	Dimensional Stacking	56
4.5.4.2	Worlds-within-Worlds	58
4.5.4.3	Cone Trees	59
4.5.4.4	Treemaps	60
4.6	Zusammenfassende Bewertung der Visualisierungstechniken	61
4.7	Zusammenfassung	63

5 Entwurf und Implementierung des Visualisierungswerkzeugs für die Anfrageunterstützung	65
5.1 Bestimmung der geeigneten visuellen Unterstützungen	65
5.1.1 Auswertung der Bewertungsergebnisse	65
5.1.2 Ermittlung der Unterstützungspunkte	67
5.1.3 Einordnung der passenden Visualisierungen	69
5.2 Auswahl für den Entwurf des Visualisierungswerkzeugs	71
5.2.1 Wahl der zu entwerfenden visuellen Unterstützungen	71
5.2.2 Überblick vorhandener Visualisierungswerkzeuge und -projekte .	72
5.2.3 Grundlage für die Entwicklung des Visualisierungswerkzeug . . .	74
5.3 Prototypische Umsetzung	76
5.3.1 Architekturbild	76
5.3.2 Mock-ups	76
5.3.2.1 Mock-up der Programmoberfläche	76
5.3.2.2 Mock-ups einer Benutzerinteraktion	78
5.3.2.3 Mock-up einer weiteren Visualisierung	79
5.3.3 Prototypischer Implementationsansatz	79
5.4 Zusammenfassung	82
6 Zusammenfassung und Ausblick	83
6.1 Zusammenfassung	83
6.2 Ausblick	84
Literaturverzeichnis	87
Abbildungsverzeichnis	95
Tabellenverzeichnis	97
Listingverzeichnis	99
Eidesstattliche Erklärung	101

1 Einleitung

Die Anzahl von Sensoren jeglicher Art, die fortwährend Messdaten erzeugen und an eine zentrale Station übermitteln, nimmt tagtäglich zu. Laut dem AMA Fachverband für Sensorik e.V. wird nach Schätzungen von Marktforschungsinstituten in Deutschland jedes Jahr etwa 22 - 25 Mrd. Euro für die Produktion von Sensoren umgesetzt, mit einer steigenden Tendenz [Se10]. Durch die fortschreitende Verbreitung von Sensoren aller Art nehmen sie eine immer größere Rolle auch im privaten Bereich ein. In modernen Autos werden bis zu 100 Sensoren verbaut, das Haus der Zukunft, das sogenannte *Smart Home*, setzt auf vernetzte Sensoren und in Supermärkten werden Sensoren sogar für Marktforschungszwecke an Einkaufswagen angebracht [LÖ9]. Um mit diesen kontinuierlichen Mengen an Sensordaten zu arbeiten, bedarf es robuster und schneller Systeme, auf die mit einer einfachen Art Anfragen gestellt werden können. Die Erarbeitung der Möglichkeiten einer visuellen Unterstützung bei der Anfragestellung auf solche Systeme könnte dabei hilfreich sein.

1.1 Motivation

Sensoren sind in unserem Leben bereits allgegenwärtig geworden. Es handelt sich dabei um Messfühler, die bestimmte Zustände und Ereignisse aus der Umwelt wahrnehmen und diese als Werte in einem fortlaufenden Datenstrom weitergeben [WIK10d]. Wird diese riesige Anzahl und Vielfalt an Ereignisdaten an einem zentralen Knotenpunkt gebündelt, entsteht ein Fluss aus Datenströmen, aus denen durch Auswahl und Kombination höherwertige Informationen gewonnen werden können.

Da Sensoren implizit die Eigenschaft haben, den aktuellen Ist-Zustand ihrer Umgebung zu messen, und meist auch nur dieser von Interesse ist, gilt besonders bei Sensordatenströmen die direkte und zeitnahe Verarbeitung von Ereignis- und Zustandsdaten als sehr wichtiger Faktor. Beispielsweise könnte die Auswertung von Sensordatenströmen aus einem Waldgebiet dazu dienen, um vor einer drohenden Waldbrandgefahr zu warnen. Werden hier Temperatursensordaten erst zu einem späteren Zeitpunkt verarbeitet oder hierfür ältere Daten verwendet, ist das Ergebnis der Auswertung bereits bei der Erstellung überholt und für das gewünschte Ziel nicht mehr relevant.

Diese Verarbeitung von komplexen Ereignisdatenströmen, die sehr groß und vielfältig sein können, während sie auftreten, bezeichnet man als *Complex Event Processing*, kurz CEP genannt. Diese Technik erzeugt höherwertige Informationen durch das Filtern, Ag-

gregieren und Korrelieren von Ereignissen, sogenannten Events, direkt aus Datenströmen heraus, während sie ein *CEP-System* passieren [EB09].

Um nun eine bestimmte Auswertung von Datenströmen zu erhalten, stellt der Anwender mittels einer speziellen Anfragesprache (*Event Processing Language (EPL)*) eine Anfrage an ein CEP-System. Als Beispiel für ein weit verbreitetes CEP-System, das als Open Source Entwicklung frei verfügbar ist, ist *Esper* zu nennen.

Im folgendem Listing 1.1 wird das einfache Beispiel der Waldbrandüberwachung aufgegriffen und in der Anfragesprache *Esper-EPL* an das CEP-System *Esper* gestellt:

```
1 SELECT temp.ID
2 FROM TempSens.*.win:length(10) as temp,
3     HumiSens.*.win_length(10) as humi
4 WHERE temp.temperature > 30,0 AND humi.humidity < 10,0
5 AND (temp.locateX - humi.locateX < 10)
```

Listing 1.1: Beispiel einer EPL-Anfrage

Diese Anfrage erzeugt eine kontinuierliche Prüfung der Sensordatenströme auf Werte, in denen die gemessene Luftfeuchtigkeit weniger als 10% beträgt und gleichzeitig die Temperatur größer 30°C ist. Dabei werden immer genau nur die letzten zehn Ereigniswerte zum Vergleich herangezogen und nur die IDs herausgefiltert, dessen Sensorstandorte weniger als 10 Meter voneinander entfernt sind. Im Gegensatz zu klassischen Datenbankabfragen wird die Anfrage hier nicht nur einmalig zu einem Zeitpunkt ausgeführt, sondern immer wieder gegen die Datenströme wiederholt. Sobald die beschriebene Situation der Anfrage eintritt, löst das CEP-System eine gewünschte Aktion aus. Dies geschieht solange, bis der Anwender die Anfrage explizit aus dem CEP-System wieder entfernt.

Zur Gewinnung von Informationen aus Sensordatenströmen mittels eines CEP-Systems muss also eine EPL-Anfrage erzeugt und gestellt werden. Erst eine korrekt formulierte Anfrage löst eine kontinuierliche Überprüfung der Sensordatenströme aus. Durch die Komplexität der Datenströme ist das Erstellen einer solchen Anfrage aber nicht immer so einfach wie im obigen Beispiel. Auch wenn die verschiedenen existierenden EPL-Anfragesprachen meistens auf der bekannten Sprache *Structured Query Language (SQL)* aufbauen, kommen viele Besonderheiten von EPL hinzu. Zum einen können mit EPL Zeit- und Mengenfenster definiert und kausale bzw. zeitliche Zusammenhänge mittels eines *FOLLOWED-BY-Operators* (\rightarrow) dargestellt werden. Beides sind Anfrageanweisungen, die nur bei Datenströmen Sinn machen. Zum anderen erlaubt EPL zusätzlich die Verwendung beliebiger natürlicher Funktionen und regulärer Ausdrücke in den Anweisungen, was bei SQL standardmäßig nicht möglich ist.

Für eine EPL-Anfragestellung und Auswertung von Sensordatenströmen sind aber noch weitere Daten und Informationen nötig. Gemeint sind gewisse Informationen über

die Sensoren bzw. Datenströme selbst, die sogenannten *Metainformationen*. Dies sind Informationen derart wie das Wissen über die Existenz von Sensoren, deren Bezeichnungen und Standorte sowie die Art der gemessenen Sensordaten. Solche Informationen sind essentielle Metainformationen für eine Anfragestellung.

Aber auch ob Sensoren regel- oder steuerbar sind (Rückkopplung), ob ihre Datenmessung zeitlich begrenzt ist, oder ob sie sich bewegen oder stationär angebracht wurden, kann für die Anfrageformulierung relevant sein und gehört zu den Metainformationen über Sensoren.

Metainformationen über Sensordatenströme zu kennen ist, neben der Beherrschung der EPL-Anfragesprache, also ein sehr wichtiger Faktor. Hinzu kommt aber noch ein weiterer Faktor, nämlich Informationen über Werteskalen und Werteverläufe von Sensordaten zu haben. Zu Wissen, in welchen Wertespektren sich Sensordatenströme befinden und wie historische Verläufe der Sensorwerte aussehen, gibt einem zusätzlich wichtige Informationen für eine richtige und sinnvolle EPL-Anfragestellung.

In der Summe könnte eine gemeinsame Darstellung all dieser vielfältigen, verfügbaren Informationen über Sensoren und die Sensordaten selbst in einer einzigen Ansicht zu einer zusätzlichen (anwenderseitigen) visuellen Datenexploration führen. D.h. mittels einer adäquaten Visualisierung der vorhandenen Daten und Informationen könnten durch die Wahrnehmung der visualisierten Art insgesamt neue Anfragemöglichkeiten und Muster vom Anwender erkannt werden, die ohne eine gemeinsame Darstellung nicht bewusst wahrgenommen oder erkannt würden [Kei02].

1.2 Zielsetzung

Die angestrebte Arbeit befasst sich mit dem Gegenstand, den Anwender visuell bei der EPL-Anfragestellung an ein CEP-System speziell bei Sensordatenströmen zu unterstützen. Durch ein Visualisierungswerkzeug soll der Anwender in die Lage versetzt werden, schneller sinnvolle EPL-Anfragen zu formulieren. Das Visualisierungswerkzeug ist als eine Art visuelle Bedienoberfläche gedacht, die als eine Unterstützung bei der Anfrageformulierung von einfachen sowie komplexen EPL-Anfragen an ein CEP-System dienen soll.

Durch unterschiedliche Visualisierungen soll dem Anwender zudem eine visuelle Exploration der Daten ermöglicht werden. Das bedeutet mit Hilfe der verschiedenen Visualisierungen unter anderem eine Übersicht über die vorhandenen Sensoren und Metainformationen zu erhalten und mittels Interaktionen mit den visualisierten Daten wertvolle Schlussfolgerungen auch für die EPL-Anfrage zu erzielen.

Zusätzlich kann die visuelle Datenexploration erlauben, eine größere Vielfalt an möglichen EPL-Anfragevariationen zu erkennen, um die Fähigkeiten eines CEP-Systems besser auszunutzen. Die visualisierten Ansichten auf die Daten können insgesamt zu einer höheren Akzeptanz der EPL-Anfragesprache führen und ein schnelleres Verständnis für

Inhalte von Datenströmen bewirken.

Insgesamt entsteht dabei eine visuelle Unterstützung, die den Anwender bei der gesamten EPL-Anfrageformulierung helfen soll und ihn dazu befähigt, einfacher EPL-Anfragen erstellen zu können.

Im Verlauf der Arbeit wird ein Visualisierungswerkzeug entwickelt und abschließend anhand eines Entwurfs in einer prototypischen Implementation realisiert. Ziel der Diplomarbeit ist es, einen Gesamtüberblick über die verschiedenen Visualisierungstechniken und ihre Darstellungsmöglichkeiten für Sensordatenströme zu erhalten. Damit erhält der Anwender zum einen eine sinnvolle Übersicht aller visuellen Unterstützungsmöglichkeiten bei der EPL-Anfrageformulierung an ein CEP-System und zum anderen mit der prototypischen Implementierung zugleich eine praktische Umsetzung.

1.3 Aufbau der Arbeit

Die Diplomarbeit setzt sich aus zwei Bereichen zusammen, einem theoretischen und einem praktischen Teil, die in insgesamt sechs Kapitel gegliedert sind. Im theoretischen Teil wird im ersten Kapitel eine thematische Einleitung gegeben sowie in die Motivation und die Zielsetzung der Arbeit eingeführt. Im nachfolgendem Kapitel 2 werden die Grundlagen zu Sensoren und komplexer Ereignisverarbeitung sowie weitere wichtige Begriffe erklärt.

Im Kapitel 3 wird ausführlich auf die Motivation für die Entwicklung einer Visualisierungsunterstützung bei der Anfragestellung eingegangen. Hierfür wird der Aufbau einer EPL-Anfrageformulierung an ein CEP-System erläutert und aufgelistet, aus welchen Inhalten eine EPL-Anfrage bestehen kann. Anhand von Anwendungsszenarien werden Gemeinsamkeiten bei der EPL-Anfragestellung auf Sensordatenströme ausgearbeitet, um abschließend die Schwierigkeiten und Probleme, die bei einer EPL-Anfrageformulierung auftreten können, aufzuzeigen.

Im 4. Kapitel werden die allgemeinen Anforderungen an Visualisierungen erläutert und die speziellen Anforderungen für die Verwendung einer Visualisierung im geplanten Visualisierungswerkzeug aufgestellt. Anschließend findet die Vorstellung und Bewertung der verschiedenen Visualisierungstechniken anhand des zuvor entwickelten Anforderungskatalogs statt. Dafür werden einzelne Vertreter jeder Visualisierungstechnik mit einer Kurzbewertung vorgestellt und am Ende mit einer Gesamtbewertung für jede Gruppe von Visualisierungstechniken zusammengefasst.

In Kapitel 5 ist der Übergang von der theoretischen Evaluierung zur praktischen Umsetzung fließend. Beginnend mit der Auswertung der Bewertungsergebnisse wird unter anderem die Fragestellung aufgeworfen, inwieweit jede der einzelnen Visualisierungsarten aufgrund ihrer Darstellung für die EPL-Anfragestellung nützlich sein kann. Dies soll durch Ermittlung der möglichen Unterstützungspunkte und anschließender Einordnung der geeigneten Visualisierungstechniken zueinander geschehen.

Das Ergebnis dieser Untersuchung bildet zugleich auch die Basis für die Auswahl der zu entwerfenden visuellen Unterstützungen im nächsten Abschnitt. Dort wird zuerst ein Überblick über bereits vorhandene Visualisierungswerkzeuge und -projekte gegeben, anhand derer evaluiert wird, welche sich am besten als Grundlage für die eigene Ausarbeitung eignen.

Das Kapitel schließt ab mit dem Entwurf und der prototypischen Umsetzung des erarbeiteten Visualisierungswerkzeugs. Dafür wird das Architekturbild des Visualisierungswerkzeug vorgestellt und erläutert und das Visualisierungswerkzeugs selber mit Demonstrations-Prototypen, sogenannten Mock-ups, präsentiert.

Den Abschluss der Arbeit bilden in Kapitel 6 eine Zusammenfassung und ein Ausblick, welche weiteren möglichen EPL-Anfragemöglichkeiten neben den bereits evaluierten noch vorstellbar sind.

2 Grundlagen

Bei der Anfrageerzeugung für Sensordatenströme handelt es sich um einen komplexen Vorgang, der sich aus verschiedenen Elementen zusammensetzt. In diesem Kapitel werden die dafür notwendigen Begrifflichkeiten erklärt. Zunächst werden die Begriffe aus dem Bereich der Sensornetze vorgestellt. Danach erfolgt die Einführung in den Bereich der komplexen Ereignisverarbeitung. Abgeschlossen wird dieses Kapitel mit den Beschreibungen für Visualisierung und Visuelle Datenexploration.

2.1 Sensornetze

In diesem Abschnitt werden Begriffe, die im direkten Zusammenhang mit Sensornetzen stehen, nacheinander erläutert. Dies umfasst zunächst die Definition von Sensoren und Sensornetzen. Anschließend werden Datenströme und Metainformationen erklärt.

2.1.1 Sensor

Für einen Sensor gibt es verschiedene, aber ähnliche Auffassungen einer Definition. Im Brockhaus [Bro99] wird ein Sensor definiert als

„Messfühler, elektronisches Bauelement, das physikalische Größen, z.B. Temperatur, Druck und Dichte von Gasen, Lage und Bewegung eines Körpers im Raum ... erfasst und in elektronische Signale umsetzt.“

Dieser Definition schließt sich Wikipedia [WIK10d] an und beschreibt einen Sensor als

„... ein technisches (Bau-)Teil, das bestimmte physikalische oder chemische Eigenschaften .. seiner Umgebung qualitativ oder als Messgröße quantitativ erfassen kann. Diese Größen werden mittels physikalischer oder chemischer Effekte erfasst und in weiterverarbeitbare Größen (meist elektrische Signale) umgeformt.“

Weitergehend können bei Sensoren noch Unterteilungen in aktive und passive Sensoren sowie physische und virtuelle Sensoren vorgenommen werden. In dieser Arbeit wird auf diese genauere Aufteilung verzichtet und Sensoren entsprechend der obigen Ausführungen als Messwertaufnehmer verstanden, die ihre Ist-Zustände ihrer Umgebung als weiterverarbeitbare Größen wiedergeben.

2.1.2 Sensornetz

[SMZ07] bezeichnet ein drahtloses *Sensornetz*, auf Englisch *Wireless Sensor Node (WSN)*, folgendermaßen:

„... a WSN consist of a group of dispersed sensors (nodes) that have the responsibility of covering a geographic area (the sensor field) in terms of some measured parameter (...). Sensor nodes have wireless communication capabilities and some logic for signal processing, topology management (if and where applicable), and transmission handling (...).“

Ein Sensornetz oder drahtloses Sensornetz besteht also aus einer Gruppe von verteilten Sensoren, die Werte eines geographischen Gebietes messen und überwachen und miteinander kommunizieren können. Drahtlose Sensornetze nutzen dabei die kabellose Kommunikation.

Durch Vernetzung bieten Sensornetze also eine kontinuierliche Messung von geografisch sehr großen Flächen an.

2.1.3 Datenstrom

Im Zusammenhang mit Java wird ein Datenstrom als eine einfache Folge von Daten verstanden, die entweder eine Eingabequelle oder Ausgabeziel darstellen [Zak07]. Bezogen auf Netzwerktechnologien wird ein Datenstrom deutlich konkreter definiert und zwar als eine Abfolge von Nachrichten, die über dieselbe Quelle, dasselbe Ziel und dieselbe Dienstqualität verfügen [Cis01].

In dieser Arbeit soll ein Datenstrom¹ nach [Ste00] grundlegender verstanden werden als eine Sequenz bzw. zeitliche Abfolge einzelner Pakete. Die Pakete stehen dabei synonym für Informationen in Form von kontinuierlichen oder aber auch diskreten Inhalten. Wichtig dabei ist, dass Datenströme immer eine zeitliche Komponente enthalten.

Zum Beispiel sind Sensordatenströme von Wetterstationen oder Mautsystemen automatisch immer mit einem Messungszeitpunkt verbunden.

2.1.4 Metainformationen

Metainformationen werden auch als Metadaten bezeichnet und sind allgemein als Daten zu sehen, die Informationen über andere Daten enthalten. Dies können Eigenschaften von realen Objekten sein wie Farbe, Seitenanzahl und ISBN-Nummer eines Buches, aber auch Attribute von nicht greifbaren Gegenständen wie der Name, die Größe und der Ort einer elektronischen Datei.

Im Zusammenhang mit Sensoren können Metainformationen sämtliche Merkmale eines Sensors sein wie beispielsweise die ID, der Ort, das Datum oder die vergangene Zeit seit der letzten Messung.

¹Im Zusammenhang mit Complex Event Processing (siehe 2.2.4) werden die Begriffe „Datenstrom“ und „Ereignisdatenstrom“ in dieser Arbeit synonym zueinander verwendet.

2.2 Komplexe Ereignisverarbeitung

Im Folgenden wird in die Grundlagen der komplexen Ereignisverarbeitung eingeführt. Dafür werden die grundlegenden Begriffe Event, Complex Event, Event Driven Architecture, Complex Event Processing, Event Processing Language und Rule Engine sowie der neue eingeführte Begriff CEP-System erklärt.

2.2.1 Event

In [DEF⁺08] werden als Beispiele für ein *Event* Ereignisse wie etwa eine Flugbuchung, der Eingang eines Auftrags, eine Geldtransaktion oder das Ausliefern einer Ware aufgezählt, aber auch jede einzelne Kursänderung von Aktien an der Börse wird als ein Event aufgefasst. [LS08] definiert ein Event allgemein als

„Anything that happens, or is contemplated as happening.“

Ein Event ist also alles, was geschieht oder als ein Geschehnis, Ereignis betrachtet werden kann. Aus dem Bereich der Sensornetze können das Messwerte wie die Temperatur oder der Luftdruck an einem Ort oder auch gesendetes Signal eines Bewegungsmelders sein.

2.2.2 Complex Event

Complex Event [LS08] lautet übersetzt „komplexes Ereignis“ und steht für ein Ereignis, das aus einer Menge von Teilereignissen besteht. Ein Complex Event ist insofern komplex, da es eine Ereignismenge umfasst, dessen Einzelereignisse aber nicht mehr immer bestimmbar sind. Es entsteht also eine Menge, die sich nur noch als Kombination mehrerer Ereignisse erkennen lässt.

Als Beispiel hierfür dient das Ereignis eines Börsencrashes. Der Börsencrash selber ist in seiner Summe ein Complex Event. Es ist allerdings nicht möglich, eindeutig zu sagen, welche Einzelereignisse zu diesem Desaster geführt haben bzw. dazu gehören.

2.2.3 Event Driven Architecture

Event Driven Architecture wird *EDA* abgekürzt und bedeutet übersetzt „Ereignisgesteuerte Architektur“. Der Begriff *EDA* wurde erstmals 2002 von David Luckham an der Stanford University eingeführt [Luc02]. Es handelt sich um eine Softwarearchitektur, die auf die Ereignisverarbeitung ausgerichtet ist, also das Erzeugen, Entdecken und Verarbeiten einzelner Ereignisse oder ganzer Ereignisströme als zentrale Architekturkomponenten enthält [DEF⁺08].

2.2.4 Complex Event Processing

Complex Event Processing (CEP) wird von [LS08] folgendermaßen definiert:

„Computing that performs operations on complex events, including reading, creating, transforming, or abstracting them.“

Complex Event Processing bezeichnet also den Vorgang, komplexe Ereignisse (engl. *Complex Events* siehe 2.2.2) zu verarbeiten, d.h. zu lesen, zu erzeugen, zu transformieren oder zu abstrahieren. In [EB09] wird dieser Vorgang noch etwas genauer beschrieben und CEP folgendermaßen definiert:

„Complex Event Processing (CEP) ist ein Sammelbegriff für Methoden, Techniken und Werkzeuge, um Ereignisse zu verarbeiten während sie passieren, also kontinuierlich und zeitnah. CEP leitet aus Ereignissen höheres, wertvolles Wissen in Form von sog. komplexen Ereignissen, d.h. Situationen die sich nur als Kombination mehrerer Ereignisse erkennen lassen, ab.“

Der CEP-Prozess, der als Ergebnis immer Complex Events hat², lässt sich dabei in zwei Kategorien unterteilen. Zum einen in die Kategorie des Erkennens von a-priori bekannten Mustern in Ereignisdatenströmen, zum andern in das Finden von bisher unbekanntem Mustern.

Dabei besteht ein CEP immer aus zwei Teilen. Es setzt sich zusammen aus einer definierten Anfragesprache (siehe 2.2.5) und einer dazugehörigen *Rule Engine* (siehe 2.2.6), welche die Anfragen verarbeiten kann.

2.2.5 Event Processing Language

Mit *Event Processing Language (EPL)* werden Anfragesprachen für Datenströme bezeichnet, mit denen sich Ereignisse, Ereignismuster und Regeln beschreiben lassen. Die meisten EPLs lehnen sich an die Datenbankabfragesprache SQL an und nutzen SQL bekannte Strukturen und Konstrukte wie `SELECT`, `WHERE` und `INSERT INTO`, um komplexe Ereignisse zu filtern, zu aggregieren und zu erzeugen. Statt Datenbanktabellen finden hier Eingabeströme und *Views* (siehe 3.1.1.5) Verwendung. Mit Zeitfenstern (*Time Windows* siehe 3.1.1.4) und Längenfenstern (*Length Windows* siehe 3.1.1.5) lassen sich zudem die Mengen an betrachteten Ereignissen in einem Ereignisdatenstrom eingrenzen. Im Gegensatz zu Datenbankabfragen werden in EPL formulierte Anfragen nicht nur einmalig, sondern kontinuierlich ausgeführt. Zum Definieren von Ereignismustern in EPL können innerhalb der `WHERE`-Klausel Ereignistypen, Ereignisattribute und temporale Aspekte, also die zeitliche Abfolge von Ereignissen, beschrieben werden. Insbesondere ermöglicht EPL auch die Spezifizierung von Ereignissen, die innerhalb eines Zeitraums ausgeschlossen werden sollen, d.h. nicht stattgefunden haben [DEF⁺08].

Aufgrund der kontinuierlichen Abfrageeigenschaft an das CEP-System und der Anlehnung der Anfragesprache an SQL wurden und werden Anfragesprachen auf Daten-

²Ein CEP-Prozess erzeugt immer komplexe Ereignisse, also Complex Events, unabhängig davon, ob es sich bei den Ursprungsereignissen um einfache oder komplexe Ereignisse handelt. [LS08]

ströme auch als *Continuous Query Language (CQL)* [ABW03] oder *Event Processing Query Language (EQL)* [Esp06] bezeichnet.

2.2.6 Rule Engine

Die *Rule Engine* ist das zentrale Stück eines CEP. Es handelt sich dabei um die datenstromverarbeitende Komponente, die die Ereignisströme entsprechend der in EPL definierten Ereignismuster und -regeln verarbeitet. Die Rule Engine ist dabei speziell für die schnelle Abarbeitung von großen Datenströmen ausgelegt und daher an zwei Entwurfsprämissen geknüpft. Zum einen ist ein hoher Datendurchsatz von bis zu 1.000.000 gleichzeitig verarbeiteten Ereignissen pro Sekunde angestrebt, zum andern sollen die Latenzzeiten für die Reaktion auf Ereignisse möglichst gering sein [DEF⁺08].

2.2.7 CEP-System

In dieser Arbeit wird der Begriff *CEP-System* als ein System eingeführt, welches Complex Event Processing zur Ereignisverarbeitung verwendet. Diese Konkretisierung wird gemacht, um die allgemeine Definition für CEP von den Systemen, die CEP als Verarbeitungszentrale verwenden, zu unterscheiden. Zu den momentan vorhandenen CEP-Systemen gehören die kommerziellen Oracle CEP, Coral8, StreamBase, Aleri und RTM-Analyzer, die frei verfügbaren Studienprojekte Stream und StreamCruncher sowie das Open-Source Projekt Esper [DFH08][EB09]. Alle derzeit existierenden CEP-Systeme unterstützen primär die a-priori Suche nach bekannten Mustern und geben weniger eine Hilfe für das Erkennen unbekannter Muster.

2.3 Visualisierung (allgemein)

Der Begriff der *Visualisierung* wird im Allgemeinen als das Umwandeln symbolischer oder numerischer Informationen in visuelle Information wie Bilder, Graphiken und Karten verstanden. Nach [Jun98] dient es dabei dem Erfassen komplexer Zusammenhänge. Auch [Leh10] sieht es ähnlich und formuliert es folgendermaßen:

„Visualisierung ist ... ein Werkzeug zum phänomenologischen Verständnis von Zusammenhängen (...).“

Visualisierung ist also eine Art, Informationen oder Daten in eine bildhafte Form zu bringen, damit Zusammenhänge besser erfasst und verstanden werden können. Dabei übernimmt eine Visualisierung aber immer nur die Funktion der reinen Präsentation der Daten, und darf sie auf keine Art und Weise selbst weiter interpretieren [Sch08].

Eine weitere auf Datenmengen und Auswertung bezogene Definition von Visualisierung ist im Kapitel 4.1 zu finden.

2.4 Visuelle Datenexploration

Der Begriff der *visuellen Datenexploration* wird oft in Verbindung mit *Data Mining*³ genannt und von [Kei02] als *Visuelles Data-Mining-Verfahren* bezeichnet. Ihre Grundidee ist es, anhand von geeigneten visuellen Darstellungen von Daten, Einblicke in die Struktur der Daten zu erhalten. Durch Interaktionen mit der visuellen Form der Daten können zudem auch gehaltvolle Folgerungen aus der Darstellung geschlossen werden. Visuelle Datenexploration kann aber auch als das Aufstellen von Hypothesen aus einer Visualisierung heraus verstanden werden, das sich aus tieferen Verständnis der dahinter liegenden Daten ergibt.

Das Vorgehen der visuellen Datenexploration ist nach [Kei02] nur dann sinnvoll, wenn wenig über Daten bekannt ist und die Ziele der Untersuchung ungenau spezifiziert sind.

2.5 Zusammenfassung

Sensornetze sind riesige Zusammenschlüsse von Sensoren, die Unmengen an Daten und Informationen produzieren, die unentwegt an Basisstationen gesendet werden. Die Datenwerte dieser Datenströme können auch als Ereignisse, sogenannte Events verstanden werden, da sie Messwerte realer Geschehnisse bzw. deren Attribute repräsentieren. Der junge Zweig der Komplexen Ereignisverarbeitung beschäftigt sich mit der Verarbeitung dieser Ereignisströme und erlaubt die Anfrage an komplexe Situationen. Dabei wird der gesamte Vorgang der Verarbeitung als Complex Event Processing (CEP) bezeichnet.

Um mittels CEP nun Ereignisdatenströme zu verarbeiten, ist das Erstellen einer geeigneten Anfrage nötig. Dies geschieht mit der Event Processing Language-Anfragesprache (EPL-Anfragesprache). Die meisten EPLs sind dabei eine an SQL angelehnte Anfragesprache für Datenströme. Zusammen mit der verarbeitenden Rule Engine bilden sie den in dieser Arbeit eingeführten Begriff eines CEP-Systems.

Im letzten Abschnitt wurde in die Grundlagen der Visualisierung und der visuellen Datenexploration eingeführt. Visualisierungen wurden allgemein als eine Art Darstellung von Information in bildhafter Form bezeichnet, mit dem Ziel komplexe Zusammenhänge besser und einfacher erfassen zu können. Wichtig dabei ist, dass Visualisierungen ihre Daten nur präsentieren dürfen und keine Interpretation hinein legen. Bei einer geeigneten Wahl der Darstellung kann zudem mit einer Visualisierung eine visuelle Datenexploration betrieben werden. D.h mittels Visualisierungen von Daten und durch ihre bildhafte Darstellung neue Schlussfolgerungen zu erzielen.

In diesem Kapitel wurden die Grundlagen gelegt, um in den nächsten Kapiteln auf ein konkretes, ausgewähltes CEP-System und verschiedene Visualisierungen einzugehen,

³„Data Mining beschäftigt sich mit der Möglichkeit, in Datenbeständen versteckte Muster und Strukturen aufzudecken.“ [Wir]

mit dem Ziel, ein Visualisierungswerkzeug für die EPL-Anfrageunterstützung auf Sensordatenströme zu entwerfen.

3 Motivation für die Visualisierungsunterstützung bei der Anfragestellung

In diesem Kapitel wird erläutert, warum eine visuelle Unterstützung bei der Anfragestellung sinnvoll ist und welche Arten der Verwendung vorstellbar sind. Dafür wird im ersten Abschnitt das CEP-System *Esper* vorgestellt, welches für diese Arbeit ausgewählt wurde. Im zweiten Abschnitt werden die Probleme und Schwierigkeiten einer EPL-Anfrage behandelt. Der abschließende Abschnitt befasst sich mit Anwendungsszenarien, die deutlich machen sollen, welche Unterstützungsmöglichkeiten ein Visualisierungswerkzeug für die EPL-Anfragestellung bieten kann.

3.1 Das CEP-System Esper

In den folgenden Abschnitten wird das in dieser Arbeit verwendete CEP-System *Esper* von der Firma *EsperTech Inc* [Esp09a] eingeführt und erklärt. Dabei wird zunächst allgemein auf *Esper* eingegangen, um anschließend die einzelnen Bestandteile genauer vorzustellen. Anzumerken ist, dass sich prinzipiell alle Ausführungen der folgenden Abschnitte auf die gleiche oder ähnliche Art und Weise auch auf andere CEP-Systeme, deren Anfragesprache auf SQL basiert, übertragen lassen.

Esper [Espb] ist eines der wenigen CEP-Systeme, welches derzeit als Open Source frei verfügbar ist. Mit eines der ersten CEP-Systeme war es 2008 noch das einzige CEP Open Source-Projekt unter der *General Public Licence (GPL)* [Ope] und infolge dessen heute auch ein weit verbreitetes in der Open Source-Gemeinde [DEF⁺08]. Vertrieben wird *Esper* von der Firma *EsperTech Inc.*, die neben der Open Source-Variante auch noch eine kommerzielle Version von *Esper* (*Esper Enterprise Edition* [Espa]) vertreibt.

Für diese Arbeit wurde *Esper* ausgewählt, weil es ein typischer Vertreter von CEP-Systemen ist. Es benutzt wie die meisten CEP-Systeme als Anfragesprache eine SQL ähnliche und darauf aufbauende Sprache. Es ist als freie Variante verfügbar sowie auch in Java implementiert und besitzt einen großen Funktionsumfang. *Esper* macht insgesamt einen weit ausgereiften Eindruck und ist in stabilen Versionen vorhanden. Verwendet wird das *Release 3.2.0* von *Esper for Java* [Esp09b], das in dieser Version zum Zeitpunkt der Themenevaluation die Aktuellste war.

Die von Esper eingesetzte Anfragesprache heißt wie die Klasse der Datenstrom-Anfragesprachen selbst *Event Processing Language (EPL)* und wird zur besseren Unterscheidung in dieser Arbeit als *Esper-Event Processing Language (Esper-EPL)* bezeichnet. Die Rule Engine von Esper nennt sich *Esper-Engine* [Esp09a]. Diese Bestandteile von Esper werden in nachfolgenden Abschnitten genauer erklärt.

3.1.1 Die Esper-EPL

Die *Esper-EPL* ist eine Datenstrom-Anfragesprache die auf SQL basiert und die Elemente wie `SELECT`, `FROM`, `WHERE` oder `GROUP BY` ähnlich oder genauso wie SQL verwendet. Dabei fungieren als Quelldaten keine Datenbanken, sondern Datenströme, die einzelne Ereignisse oder Ereignis-Tupel enthalten. [Esp09a] beschreibt die Funktion einer EPL-Anfrage folgendermaßen:

„EPL statements are used to derive and aggregate information from one or more streams of events, and to join or merge event streams.“

Mit der Esper-EPL werden also Anfragen auf einen Datenstrom oder mehrere Datenströme gestellt, um Ereignisse zu selektieren, zu filtern oder zusammenzuführen und so neue Informationen abzuleiten. Das Ergebnis einer Anfrage ist dabei immer ein Complex Event, d.h. eine Aggregation von Ereignissen, selbst wenn dieses aus nur einem einzigen Ereignis besteht (siehe auch Kapitel 2.2.4).

Der Aufbau einer Esper-EPL-Anfrage ist im Listing 3.1 zu sehen. Anhand dessen werden im Folgendem die einzelnen Teile der Anfragestruktur genauer beschrieben und die Besonderheiten der Esper-EPL herausgestellt.

```
1 SELECT select_list
2 FROM stream_def [as name] [, stream_def [as name]] [,...]
3 [WHERE search_conditions]
4 [GROUP BY grouping_expression_list]
5 [HAVING grouping_search_conditions]
6 [OUTPUT output_specification]
7 [ORDER BY order_by_expression_list]
8 [LIMIT num_rows]
```

Listing 3.1: Aufbau einer Esper-EPL-Anfrage [Esp09a]

3.1.1.1 Klauseln

Esper-EPL ist, wie bereits beschrieben, eine an SQL angelehnte Anfragesprache, die gleiche bzw. erweiterte und ähnliche Klauseln verwendet. Die Möglichkeiten der Esper-EPL sind sehr groß und vielfältig, so dass nachfolgend nur die wichtigsten Esper-EPL-Klauseln kurz vorgestellt werden [Esp09a].

SELECT

Die `SELECT`-Klausel ist in allen Esper-EPL-Anfragen erforderlich. Mit ihr werden die Eigenschaften von Ereignissen oder Ereignisse spezifiziert, die abgerufen werden sollen. Gleichzeitig wird damit festgelegt, welcher Typ von Ereignissen als Ergebnis ausgegeben wird.

Ein Beispiel für eine `SELECT`-Klausel ist im Kapitel 3.1.1.2 (Listing 3.2) zu finden.

FROM

Auch die `FROM`-Klausel ist zwingend notwendig für eine Esper-EPL-Anfrage. Mit ihr werden die verwendeten Ereignisdatenströme bestimmt und es können zudem *Named Windows* definiert werden. Mit dem Schlüsselwort `AS` können Datenströme an einen neuen Namen gebunden werden. Die *Named Windows* sind für diese Arbeit nicht relevant und können unter [Esp09a] nachgelesen werden.

Ein Beispiel für eine `FROM`-Klausel ist im Kapitel 3.1.1.2 (Listing 3.2) zu finden.

WHERE

Die `WHERE`-Klausel ist eine optionale Klausel. Sie legt die Bedingungen für die Ereignisse oder Ereignis-Tupel fest, nach denen gesucht werden soll. Anhand von Vergleichsoperatoren wie etwa `=`, `<`, `>`, oder `!=` und den logischen Verknüpfungen `and` und `or` können Ereignisdatenströme verbunden und gefiltert und so Ergebnismengen eingeschränkt werden.

Ein Beispiel für eine `WHERE`-Klausel ist im Kapitel 3.1.1.2 (Listing 3.3) zu finden.

GROUP BY

Mit Hilfe der optionalen `GROUP-BY`-Klausel lässt sich die Ausgabe einer Esper-EPL-Anfrage in Gruppen einteilen. Die Gruppierung kann nach einem oder mehreren Eigenschaften der Ereignisse oder nach den Ergebnisausdrücken der Anfrage stattfinden.

Ein Beispiel für eine `GROUP-BY`-Klausel ist im Kapitel 3.1.1.2 (Listing 3.4) zu finden.

HAVING

Die `HAVING`-Klausel ist optional und erlaubt Gruppierungen nochmals einzuschränken. Genauso wie die `WHERE`-Klausel die Ergebnismenge der `SELECT`-Klausel reduziert, verringert das `HAVING` die erfassten Ergebnisse der `GROUP-BY`-Klausel. Im Gegensatz zur `WHERE`-Klausel sind bei der `HAVING`-Klausel Aggregationsfunktionen wie `sum` oder `avg` erlaubt. Mit der `HAVING`-Klausel können so Ereignisse aus Gruppierungsmengen bestätigt oder zurückgewiesen werden. Sie ist ohne ein `GROUP-BY` nicht erlaubt.

Ein Beispiel für eine `HAVING`-Klausel ist im Kapitel 3.1.1.2 (Listing 3.4) zu finden.

OUTPUT

Die `OUTPUT`-Klausel ist eine optionale Esper-EPL-Klausel. Mit ihr lässt sich die Geschwin-

digkeit und Ausgaberate der Ergebnismenge einer Anfrage stabilisieren und kontrollieren und Ereignisse in der Ergebnismenge unterdrücken.

Ein Beispiel für eine `OUTPUT`-Klausel ist im Kapitel 3.1.1.2 (Listing 3.5) zu finden.

ORDER BY

Die `ORDER BY`-Klausel ist optional und sortiert die Ausgabemenge nach Eigenschaften von Ereignissen oder nach Ausdrücken, die Ereigniseigenschaften enthalten.

Ein Beispiel für eine `ORDER BY`-Klausel ist im Kapitel 3.1.1.2 (Listing 3.5) zu finden.

LIMIT

Die `LIMIT`-Klausel wird in der Regel mit der `ORDER BY`- und `OUTPUT`-Klausel verwendet, um die Abfrageergebnisse innerhalb eines bestimmten Bereiches zu begrenzen. Es kann benutzt werden, um eine festgelegte Anzahl der ersten Ergebnisse oder eines ausgewählten Bereiches zu erhalten.

Ein Beispiel für eine `LIMIT`-Klausel ist im Kapitel 3.1.1.2 (Listing 3.5) zu finden.

3.1.1.2 Beispiele für grundlegende Esper-EPL-Statements

Im folgenden werden die Möglichkeiten der Esper-EPL anhand von Esper-EPL-Statement-Beispielen (kurz EPL-Statements) verdeutlicht und erläutert.

```
1 SELECT *
2 FROM House_Sensor
```

Listing 3.2: Beispiel einer einfachen `SELECT`-Anfrage

Das obige Listing 3.2 zeigt ein Minimalbeispiel für ein EPL-Statement. Es besteht nur aus einer `SELECT`-Abfrage gefolgt von einer `FROM`-Klausel. Dieses einfache EPL-Statement wählt alle Ereignisse des `House_Sensor`-Datenstroms aus, die ans CEP-System Esper gesendet werden.

```
1 SELECT ID
2 FROM House_Sensor AS Sens
3 WHERE Sens.Temperature > 30
```

Listing 3.3: Beispiel für ein EPL-Statement mit `WHERE`-Klausel

Das obige EPL-Statement im Listing 3.3 zeigt das Beispiel einer `WHERE`-Einschränkung. Hier werden die IDs aller herausgefilterten Ereignisse des `House_Sensor` angezeigt, dessen Temperaturwert größer als 30°C ist. Dabei wird mittels des `AS`-Ausdrucks dem `House_Sensor`-Stream der neue Name `Sens` gegeben.

```
1 SELECT ID, Temperature, Company
2 FROM House_Sensor AS Sens
3 WHERE Sens.Temperature <= 0
4 GROUP BY Company
5 HAVING Produced = 'DE'
```

Listing 3.4: Beispiel für ein GROUP-BY- und HAVING-EPL-Statement

Das EPL-Statement im Listing 3.4 verwendet die HAVING-Klausel um eine gruppierte Ergebnismenge an Ereignissen weiter einzuschränken. Angezeigt werden die ID, Temperatur und Herstellungsfirma der Sensoren, die eine Temperaturmessung unter Null Grad haben, und zwar nach Firmen (Company) gruppiert, dessen Sensoren aber nur in Deutschland produziert wurden (Produced = „DE“).

```
1 SELECT Temperatur, Humidity
2 FROM House_Sensor AS Sens
3 WHERE Sens.Humidity > 80
4 OUTPUT EVERY 60 Seconds
5 ORDER BY Humidity
6 LIMIT 100
```

Listing 3.5: Beispiel eines EPL-Statements mit OUTPUT-, ORDER BY- und LIMIT-Klausel

Bei Ereignisdatenströmen kann die Resultatmenge, die geliefert wird, aufgrund des stetigen Ereignisflusses sehr groß und fortwährend sein. Mit der OUTPUT-Klausel lässt sich die Ausgabe einschränken und reduzieren. Das Listing 3.5 zeigt ein Beispiel für die Verwendung der OUTPUT-Klausel mit ORDER BY und LIMIT. Mit dem Statement werden die Temperatur und Luftfeuchtigkeit der Haussensoren abgefragt, die eine Luftfeuchtigkeit von über 80% gemessen haben (Sens.Humidity > 80). Die Ausgabe geschieht wegen EVERY 60 Seconds nicht ständig, sondern nur alle 60 Sekunden. Die ORDER BY Humidity sortiert die Ausgabe nach der Luftfeuchtigkeit und LIMIT 100 begrenzt die Ausgabemenge auf 100 Elemente.

Bisher wurden nur die grundlegenden Esper-EPL-Statements vorgestellt, wie sie auch aus SQL größtenteils bekannt sind. Konstrukte, die über SQL hinaus gehen und die Besonderheiten der strombasierten Anfragesprache betonen, werden im folgendem Abschnitt genauer erläutert.

3.1.1.3 Pattern

Die Esper-EPL bietet die Definition von sogenannten *Pattern*, zu deutsch Muster, an, mit denen Beziehungen, vor allem auch zeitliche Beziehungen zwischen Ereignissen beschrieben werden können. Pattern-Ausdrücke bestehen dabei aus zwei Ausdrückstypen: *Pattern-Atomen* und *Pattern-Operatoren*.

Die *Pattern-Atome* bilden die Basis für den Pattern-Block. Pattern-Atome können Filterausdrücke, zeitbasierte *Event Observer*, zu deutsch Ereignisbeobachter, sein oder benutzerdefinierte *Plug-In Observer*. Letzteres ist für diese Arbeit nicht relevant und kann unter [Esp09a] nachgelesen werden. Mit Filterausdrücken kann ein Ereignis, nach dem gesucht wird, definiert werden, zeitbasierte Event Observer erlauben Zeitintervalle oder Zeitpläne zu spezifizieren.

Die *Pattern-Operatoren* kontrollieren und bestimmen die Lebensdauer der Pattern-Atome und kombinieren diese Ausdrücke dafür logisch oder zeitlich miteinander. Es gibt vier Typen von Pattern-Operatoren. Typ 1 sind die Operatoren `EVERY`, `EVERY-DISTINCT`, `[NUM]` und `UNTIL`, mit denen sich Wiederholungen von Pattern-Teilausdrücken steuern lassen. Der zweite Typ sind die logischen Operatoren `AND`, `OR` und `NOT`. Zu Typ 3 zählen die zeitlichen Operatoren wie `FOLLOWED-BY` (\rightarrow), die auf eine Ereignis-Abfolge hin arbeiten. Und Typ 4 sind *Guards*, zu deutsch Wächter, die die Lebensdauer von Teilausdrücken kontrollieren, beispielsweise `TIMER:WITHIN`.

Ein Pattern kann überall in der `FROM`-Klausel eines Esper-EPL-Statements vorkommen einschließlich in Joins und Unterabfragen. Dabei kann es mit fast allen Klauseln wie etwa `WHERE`, `GROUP BY` oder `HAVING` kombiniert und benutzt werden. Wird ein Pattern erfüllt, so wird dies auch als *Pattern Matching* bezeichnet [Esp09a].

```

1 SELECT a.custId, sum(a.price + b.price)
2 FROM pattern [every a=ServiceOrder ->
3             b=ProductOrder(custId = a.custId)
4             WHERE timer:within(1 min)
5             ].win:time(2 hour)
6 WHERE a.name in ("Repair", b.name)
7 GROUP BY a.custId
8 HAVING sum(a.price + b.price) > 100

```

Listing 3.6: Pattern Matching: Beispiel eines EPL-Statements mit Pattern (aus [Esp09a])

Das Listing 3.6 zeigt das Beispiel eines EPL-Statements mit Pattern. Das EPL-Statement berechnet und zeigt den Gesamtpreis für jeden Kunden samt ID an, der innerhalb einer Minute eine `ServiceOrder` gefolgt von einer `ProductOrder` im System ausgelöst hat. Das Ganze geschieht über einen Zeitraum von zwei Stunden, aber nur für die Kunden, dessen Gesamtpreis über 100 liegt. Zusätzlich wird die Ergebnisliste nach Kunden-ID gruppiert. So kann beispielsweise sofort erkannt werden, was ein Kunde innerhalb von zwei Stunden alles jeweils über den Wert von 100 in Auftrag gegeben hat [Esp09a].

3.1.1.4 Time Windows

Die Bedeutung von *Time Windows* lautet wörtlich übersetzt Zeitfenster und genauso sind diese auch als solche zu verstehen. Es gibt zwei Arten von Time Windows: *Time Window* und *Time Batch*.

Time Window ist eine Art bewegliches, gleitendes Zeitfenster, mit der ein Zeitintervall für die Menge an beispielsweise ankommenden Ereignissen festgelegt werden kann, die für eine Anfrage berücksichtigt werden. Damit wird also ähnlich wie bei einem Längenfenster (siehe 3.1.1.5) die Anzahl an betrachteten Ereignissen aus einem Datenstrom eingegrenzt.

```
1 SELECT account, avg(amount)
2 FROM Withdrawal.win:time(4 sec)
3 GROUP BY account
4 HAVING amount > 1000
```

Listing 3.7: Time Windows: Beispiel eines EPL-Statements mit Time Window (aus [Esp09a])

Im Listing 3.7 ist ein Beispiel für ein Time Window zu sehen. Das EPL-Statement bestimmt alle Konten, dessen Auszahlungsbeträge durchschnittlich größer waren als 1000 und zwar immer nur genau innerhalb der letzten 4 Sekunden.

Abbildung 3.1 soll das Time Window noch einmal an einem kürzeren Beispiel veranschaulichen. Das nachfolgende EPL-Statement im Listing 3.8 wird als Beispiel genommen. Es wählt kurzerhand immer alle Ereignisse im Zeitraum von vier Sekunden aus und gibt diese zu jeder Zeit als gleiche Ereignisse wieder aus.

```
1 SELECT *
2 FROM Withdrawal.win:time(4 sec)
```

Listing 3.8: Einfaches EPL-Statement mit Time Window (aus [Esp09a])

Die Abbildung 3.1 zeigt das passende Ausgabeschema des Time Window dazu an. Das Zeitfenster bewegt sich angefangen bei $t-4$ von links nach rechts in der Zeitachse auch immer weiter nach unten. Für diese Esper-EPL-Anfrage werden stets nur die Ereignisse berücksichtigt, die sich im gezeichneten Zeitrahmen der Abbildung befinden.

Das *Time Batch* ähnelt dem Time Windows im Zeitfenster. Es funktioniert so, dass es die ankommenden Ereignisse in einen definierten Zeitfenster puffert und sie in Zeitintervallen entsprechend der Zeitfenstergröße aktualisiert und weiter gibt. So entsteht ein Ereignisstrom im Takt des Time Batch.

```
1 SELECT *
2 FROM Withdrawal.win:time_batch(4 sec)
```

Listing 3.9: Time Windows: Beispiel eines EPL-Statements mit Time Batch (aus [Esp09a])

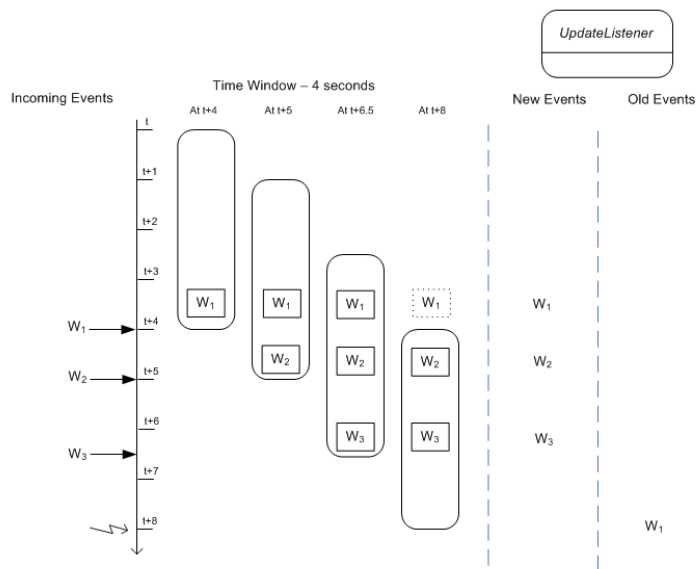


Abbildung 3.1: Schema der Ausgabe bei einem EPL-Statement mit Time Window [Esp09a]

Im Listing 3.9 ist ein einfaches EPL-Statement mit einem Time Batch zu sehen. Hier werden alle Ereignisse der jeweils letzten vier Sekunden gepuffert und als ein Ereignis alle vier Sekunden weitergeleitet.

Abbildung 3.2 veranschaulicht das Time Batch Beispiel vom Listing 3.9. Die Abbildung zeigt schematisch, wie das Time Batch-Zeitfenster erst alle vier Sekunden die gesammelten Ereignisse als ein komplexes Ereignis gebündelt ausgibt. Zu sehen ist in der Spalte $At\ t-4$ der Sprung des Zeitrahmens zu der nächsten Zeitspanne [Esp09a].¹

3.1.1.5 Views

Die Esper-EPL arbeitet mit Ereignisdatenströmen. Dennoch eignet sich nicht für jede Anfrage die Tatsache, dass sie konzeptionell auf eine unendliche Menge von Daten ausgeführt werden muss. Von daher bietet die Esper-EPL das Konzept der *Views*, übersetzt Sichten, an. Es ermöglicht, Ereignisse aus dem Datenstrom nach bestimmten Kriterien festzuhalten, damit Anfragen gegen die Menge von Ereignissen innerhalb der View gestellt werden können [Esp09a] [Jäg10].

Zwei der vielzähligen Views wurden bereits vorgestellt, die Zeitfenster Time Window und Time Batch. Esper besitzt aber noch eine Reihe weiterer Views wie z.B. die Längenfenster *Length Window* und *Length Batch Window*, die die Kriterien für die View anhand der Anzahl von gleichzeitig betrachteten Ereignissen festlegt. Da alle Views auch eine Kombination untereinander erlauben, die Menge an verfügbaren Views groß ist und teilweise Sonderfälle darstellen, wird an dieser Stelle für eine ausführliche Übersicht

¹In der Abbildung 3.2 des Time Batch-Schemas muss es in der ersten Zeitspalte richtigerweise „ $At\ t+2$ “ heißen.

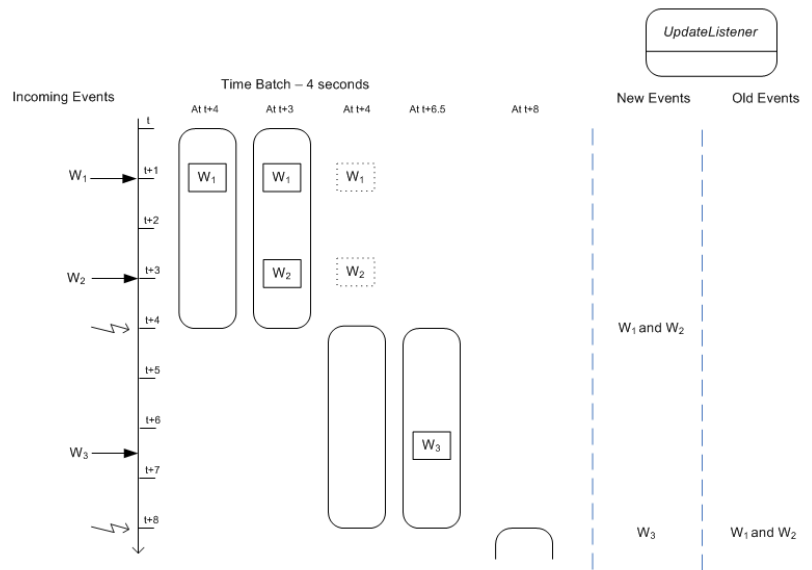


Abbildung 3.2: Schema der Ausgabe bei einem EPL-Statement mit Time Batch [Esp09a])

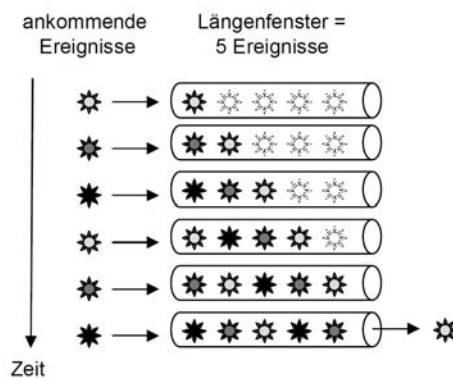


Abbildung 3.3: Schema von Längenfenster-Views [DEF⁺08])

und Vorstellung aller vorhandenen Views und Kombinationen auf das Esper-Referenz-Dokument [Esp09a] verwiesen.

Abbildung 3.3 veranschaulicht das Prinzip der Längenfenster-Views. Die Abbildung zeigt, wie ein Längenfenster der Puffergröße 5 bei einem vollen Puffer auf ein neues Ereignis reagiert. Nachdem fünf Ereignisse in den Puffer aufgenommen wurden, schiebt das neue Ereignis in der letzten Zeile nach einer bestimmten First-In-First-Out-Strategie (FIFO) das älteste Ereignis aus dem Zeitfenster und damit aus dem Betrachtungsraum für die Anfrage [DEF⁺08].

3.1.2 Die Esper-Engine

Die zentrale Verarbeitungseinheit des CEP-Systems Esper ist ihre Rule Engine, die sich bei Esper *Esper-Engine* nennt. Bei einer Anfragestellung an das CEP-System werden die formulierten EPL-Anfragen in der Esper-Engine registriert. Die Firma EsperTech Inc. be-

schreibt die Arbeitsweise ihrer Esper-Engine als eine Art umgedrehte Datenbank:

„The Esper engine works a bit like a database turned upside-down. Instead of storing the data and running queries against stored data, the Esper engine allows applications to store queries and run the data through. Response from the Esper engine is real-time when conditions occur that match queries. The execution model is thus continuous rather than only when a query is submitted.“

Die Esper-Engine ist also das Herzstück von Esper. Anstatt wie bei einer Datenbank Anfragen auf darauf gespeicherte Daten zu machen, werden beim CEP-System Esper die Anfragen in der Esper-Engine gespeichert und die Daten für eine Auswertung durch die Engine hindurch geschickt. Durch diese Weise werden Anfragen auf Daten nicht nur einmalig, sondern kontinuierlich gestellt. Bei Datenströmen sind so Antworten auf Anfragen in nahezu Echtzeit möglich, sobald die Bedingungen der Anfrage erfüllt sind.

Die Esper-Engine ist ausgelegt auf einen hohen Datendurchsatz. Nachrichten in der Größenordnung von 100.000 bis 1.000.000 pro Sekunde sind mit der Engine verarbeitbar. Gleichzeitig besitzt die Engine eine sehr niedrige Latenzzeit, so dass Anwendungen innerhalb von wenigen Millisekunden bis zu einigen Sekunden in beinahe Echtzeit auf Anforderungen reagieren können.

Eine weitere Ausführung und ausführliche Beschreibungen der Möglichkeiten der Esper-Engine ist unter [Esp09a] zu finden.

3.2 Anwendungsszenarien

In diesem Abschnitt werden verschiedene Anwendungsbeispiele vorgestellt, die verdeutlichen sollen, wie ein Visualisierungswerkzeug die EPL-Anfragestellung unterstützen kann. Dabei werden zwei Anwendungsszenarien genauer ausgeführt und vier weitere Anwendungsideen kurz skizziert.

3.2.1 Waldbrandgefahrenüberwachung

Zur Waldbrandüberwachung und -bekämpfung hat eine Firma einen Wald flächendeckend mit Sensoren ausgestattet, die kontinuierlich die Temperatur und Luftfeuchtigkeit messen und an eine zentrale Überwachungsstation senden. Dort laufen die Sensordatenströme auf dem CEP-System Esper zusammen, auf das EPL-Anfragen gestellt werden können. Die Firma bietet Anwohnern, Ferienhausbesitzern und dem Naturschutzbund (NABU) an, ihr Anfragen- und Auswertungssystem gegen eine monatliche Gebühr zu nutzen. Dafür laden sich die Anwender ein Softwarepaket herunter, das einen Zugriff auf das CEP-System erlaubt. Um die Anfragestellung zu vereinfachen, gibt es die Möglichkeit gegen eine geringe Gebühr ein zusätzliches Werkzeug zu erwerben. Das Werkzeug bietet dem Anwender eine visuelle Unterstützung bei der Auswahl und Bestimmung

von Anfrageparameter und kann vor allem bei nicht vertrauten Datenströmen hilfreich sein.

Der NABU hat sich dazu entschieden, den Service zu nutzen. Um den Mitarbeitern den Umgang mit dem Auswertungssystem zu erleichtern, wurde das Softwarepaket samt zusätzlichem Unterstützungs-Plugin erworben.

Nachdem die Mitarbeiter in die Anfragesprache Esper-EPL eingeführt wurden, können sie mit dem System umgehen. Möchte ein Mitarbeiter nun informiert werden, sobald eines der zehn Überwachungsstationen des NABU durch einen möglichen Waldbrand gefährdet ist, ruft er die Eingabemaske des CEP-Systems mit der visuellen Unterstützung auf. Diese besteht unter anderem aus einer geografischen Anzeige des Waldgebiets, mit der Übersicht aller verfügbaren Sensoren.

Über Zoom- und Schieberegler, wie sie in interaktiven Landkarten verwendet werden, sucht der Mitarbeiter über die visuelle Werkzeugunterstützung eines der Stationen aus. Mittels einer am oberen Rand erscheinenden Leiste mit Auswahlfeldern kann er, wenn gewünscht, eine andere Visualisierung der Karte auswählen.

Nun bekommt der Mitarbeiter beim Überfahren der ihn interessierenden Sensoren eine Übersicht über die aktuell verfügbaren Sensordatenarten angezeigt. Über das obige Auswahlmenü lässt er sich die verfügbaren Daten und Datenarten passend einblenden. Es erscheinen je nach Bereich der einzugebenden Daten für das CEP-System unterschiedliche Visualisierungen, die die Eingabemöglichkeiten verdeutlichen und aufzeigen. Beispielsweise kann sich der Mitarbeiter eine Visualisierung einblenden lassen, in der die Temperatur- und Luftfeuchtwerte eines oder mehrerer ausgewählter Sensoren für einen Zeitraum von einer Woche ansprechend angezeigt werden. Damit kann er auf einen Blick den sinnvollen Wertebereich für die Eingabe erkennen.

Hat der Mitarbeiter sich mittels der visuellen Unterstützung über mögliche Anfrageparameter informiert und entsprechend eine Eingabe ans CEP-System formuliert, kann er die erstellte EPL-Anfrage nun abschicken. Das CEP-System Esper überwacht fortan die ausgewählten Sensordatenströme entsprechend der eingestellten Anforderungen. Sobald in den Sensordatenströmen Ereignisse auftreten, die die Anfrageformulierung erfüllen, löst das System eine Mitteilung aus und der NABU wird per Kurzmitteilung aufs Handy über die Gefahrenstufe informiert.

3.2.2 Smart Home

Für die Überwachung und Bedienung von Smart Homes setzt ein Hausbau-Unternehmen das CEP-System Esper ein. Smart Home ist die Bezeichnung für zukunftsorientierte, intelligente und energieeffiziente Häuser, die u.a. mit einer Vernetzung von Haustechniken und -geräten sowie weiterer Sensoren ausgestattet sind. Durch ein sinnvolles Zusammenspiel der Geräte des vernetzten Hauses soll eine zugleich energiesparende wie komfortable und sichere Hausnutzung ermöglicht werden [Sma09].

Zur Bedienung und Regelung aller Einstellungen an einem Smart Home kann eine

zentrale Haussystemsteuerung eingesetzt werden. Dabei handelt es sich um eine zentrale Bedienungs- und Steuereinheit, über die sich Einstellungsänderungen im ganzen Haus vornehmen lassen [Res07].

In diesem Beispiel bietet die Hausentwicklungsfirma den Bewohnern ein Benutzer-Interface mit einem großen Touchscreen-Bildschirm an, der sich im Eingangsbereich des Hauses befindet. Über dieses bedienen die Bewohner das Smart Home und erstellen Steuer- und Regelbefehle, die Anfragen an das eingesetzte CEP-System Esper enthalten können. Unterstützt werden sie dabei durch ein Werkzeug, das verschiedene Visualisierungen als Hilfe für die einzugebenden Anfrage-Parameter anbietet.

Möchte ein Bewohner eine Urlaubsüberwachung seines Hauses starten, benutzt er dafür das Haussystem und wählt im Menü die Eingabe einer neuen Anfrage aus. Da alle Bewohner in einen Workshop der Hausbau-Firma in die Anfragesprache Esper-EPL eingeführt wurden, kennen sie sich mit dem Schema einer EPL-Anfrage gut aus und sind in der Lage, Anfragen zu formulieren. Dennoch wählen die Bewohner das Werkzeug zur visuellen Anfrageunterstützung gerne aus, um die Möglichkeiten der EPL-Anfragen visualisiert zu bekommen.

Zunächst wählt der Bewohner auf einer Grundrisskarte des Hauses die relevanten Fenster- und Türsensoren aus. Über das Visualisierungswerkzeug lässt er sich die Durchschnittswerte der Sensoren in einer geeigneten Darstellung anzeigen. So kann er schnell die richtigen Einstellungen für die EPL-Anfrage bestimmen.

Nachdem der Bewohner sich für die kombinierte Überwachung von Bewegungsmeldern und Temperaturfühlern an Fenster- und Türrahmen entschieden hat, möchte er zudem festlegen, dass die Überwachung ausgesetzt wird, solange er sich in der Nähe befindet. Er lässt sich erneut eine Grundrissvisualisierung anzeigen, in der er nun den Bereich auslotet, in dem die Überwachung bei Anwesenheit seiner RFID-ID-Karte vorübergehend deaktiviert werden soll. Die Visualisierung zeigt ihm dafür die Werte des markierten Bereichs für die Eingabe in die EPL-Anfrage an.

Hat der Bewohner die gewünschten Werte in die Anfrage übernommen, schließt er die Anfrage an das System mit einer Bestätigung ab und registriert somit die EPL-Anfrage beim CEP-System Esper. Sollte ein Einbrecher während der Urlaubszeit einen Türrahmen aufhebeln oder eine Fensterscheibe einschlagen, schlägt das CEP-System aufgrund der Positions- oder Temperaturveränderung an und löst eine Mitteilung aus.

3.2.3 Weitere Anwendungsideen

Drei weitere Anwendungsfälle, bei denen Visualisierungen bei der EPL-Anfragestellung nützlich sein können, werden hier kurz skizziert. Sie sollen die Variabilität der Hilfeleistung einer visuellen Unterstützung aufzeigen.

- **Patientenüberwachung im Krankenhaus**

Für die Überwachung von Patienten wird in einem Krankenhaus u.a. ein CEP-System verwendet. Möchte eine Krankenschwester unter gewissen Bedingungen

die Bewegungsfreiheit von Patienten überwachen und gegebenenfalls regulierend einschreiten, stellt sie eine EPL-Anfrage ans CEP-System. Sie möchte den Bereich, in dem sich ein Patient frei bewegen kann, bestimmen und ihm aber gleichzeitig bei sehr guten Vitalwerten und -funktionen erlauben, den Bereich für eine festgelegte Zeit zu verlassen.

Beispielsweise könnte ein Arzt anordnen, dass Patient X nur das Gebäude verlassen darf, wenn er Patient Y im Nebengebäude besucht. Dies aber auch nur dann, wenn sein Puls niedrig ist und sein Kreislauf währenddessen stabil bleibt. Der Besuch darf dabei nicht länger als 20 Minuten dauern.

Hier könnte ein Grundrissansicht von den Gebäuden und eine angemessene Visualisierung der Körperfunktionen die Anfrage erleichtern. Für eine Anfrageformulierung könnte der Grundriss mit allen verfügbaren Bewegungssensoren angezeigt werden. In dieser Ansicht könnte man sich die Körperfunktionen des Patienten der letzten Tage anzeigen lassen und daran definieren, was spezifisch bei diesem Patienten ein niedriger Puls und eine stabiler Kreislauf sind.

- **Unterstützung in Verkehrsleitzentralen**

In Verkehrsleitzentralen laufen sekundlich Unmengen an Daten zusammen, die aus verschiedensten Quellen stammen und sehr unterschiedlichen Inhalt haben können. Hier bieten sich erweiterte Visualisierungen bei einer Anfrage geradezu an, da einerseits sehr viele geografisch verteilte Datenströme zusammenlaufen, die sich z.B. gut in einer Landkarte visualisieren lassen.

Andererseits werden in Verkehrsleitzentralen häufig Anfragen getätigt, die aus den aktuellsten Verkehrsdaten stammen müssen. Zum Beispiel wenn Engpässe im Verkehr erkannt oder vermieden werden wollen. In diesen Fällen ist das Erfassen und Erkennen der neusten Datenlage die einzige Option. Solch eine Anzeige und Übersicht kann bei sehr großem Datenaufkommen nur eine vernünftige Visualisierung bieten.

- **Marktforschung in Einkaufszentren**

In einem Einkaufszentrum möchte das Center-Management seine Besucher überwachen, um aufgrund der Besucherzahlen und des Umsatzes der einzelnen Geschäfte die jeweilige Miete festzulegen. Dafür bedienen sie sich aller verfügbaren Messpunkte und Sensoren wie RFID-, Bluetooth- oder GPS-Geräte und Bewegungsmeldern. Um Einkäufe einem Kunden zuzuordnen, muss u.a. gemessen werden, wann ein Kunde ein Geschäftsbereich betritt und wann ein Einkauf stattgefunden hat einschließlich der dazugehörigen Summe.

Für die EPL-Anfrage an das genutzte CEP-System könnte in diesem Fall die Visualisierung des Grundrisses vom Einkaufszentrum eine Hilfe sein. Über eine Zoomfunktion könnten die Geschäfte mit den zugehörigen, verfügbaren Messfähigkeiten

angezeigt sowie Bewegungsbereiche und -aktionen visuell unterstützt festgelegt werden.

3.2.4 Gemeinsamkeiten der Anwendungsszenarien

In allen Szenarien wird ein visuelles Unterstützungswerkzeug zur einfachen, anwenderorientierten Eingabe einer EPL-Anfrage an das CEP-System Esper angeboten. Eine visuelle Darstellung der verfügbaren Sensoren und Sensordatenarten dient als Übersichtsgrundlage für die Anfrageformulierung. In allen Fällen bestehen die Datenströme aus Sensordatenströmen, die ähnliche Messwerte wie Temperatur, Beschleunigung oder Ortsdaten zur Basis haben. Eine Ortskarte könnte in allen Szenarien z.B. die aktuelle Lage aller Sensoren aufzeigen. Würden beispielsweise auf einer Karte die aktuellen Temperaturwerte in einer geeigneten Visualisierung dargestellt werden, könnte auf einen Blick die Wärmesituation in einem Gebiet aufgezeigt und eine entsprechende Anfrage durch den Anwender erstellt werden.

Allgemein könnten für alle Anwendungsszenarien Standardansichten für die visuelle Darstellung von (einheitlichen) Sensoren und Sensordatenströmen entwickelt werden, die einen Basisansichtensatz für die visuelle Anfrageunterstützung bilden.

Bei der Visualisierung der Daten selbst wird grundsätzlich bei allen Szenarien schnell deutlich, dass aufgrund der riesigen Mengen an anfallenden Datenvolumina eine Visualisierung der Daten auf verschiedene Art und Weisen für den Anwender sehr nützlich sein kann und eine sehr große Hilfe bei der Anfragestellung bietet.

In allen Fällen bietet das zu entwickelnde Visualisierungswerkzeug eine besondere Unterstützung bei der Wahrnehmung und Aufnahme von Wertespektren der Datenströme über die Zeit und dem Auffinden von zeitlichen Zusammenhängen, die bei der Anfragestellung an ein komplexes CEP-System ohne dies fehlen würde. Durch die verschiedenen Visualisierungen erhält der Anwender insgesamt eine sehr anschauliche Unterstützung und wird damit besser befähigt, sinnvolle EPL-Anfragen zu formulieren.

3.3 Probleme und Schwierigkeiten einer EPL-Anfrage

Die Esper-EPL bietet für die Esper-Engine ein sehr mächtige Anfragesprache im Bereich des Complex Event Processing. Da sie wie die meisten EPL-Anfragesprachen als Grundlage SQL verwendet, wird die Esper-EPL als Stellvertreter für alle anderen genommen.

Wie bereits vorgestellt, besteht eine Esper-EPL-Anfrage aus verschiedenen Elementen, die auf unterschiedliche Weise untereinander und auch teilweise mehrfach miteinander kombiniert und verschachtelt werden können. Das stellt den Anwender vor einige Schwierigkeiten bei der Anfrageformulierung.

Das besondere an EPL-Anfragen ist die Formulierung von Views wie z.B. den Zeit- und Längenfenstern. Hier vor allem ist eine visuelle Hilfe von Vorteil, da es den Begriff von Fenstern visuell fassbar macht. Anstatt Werte zu schätzen, könnte mittels einer an-

gebrachten Darstellung direkt Werte in einer Ansicht angezeigt werden, um etwa die Zeitspanne für sich wiederholende Muster festzulegen.

Für das Erstellen einer EPL-Anfrage, muss der Anwender aber auch wissen, welche Datenströme für eine Anfrage zur Auswahl stehen. Bei einer großen Zahl von Datenströmen wird beispielsweise eine einfache Auflistung in einer Tabellenansicht schnell unübersichtlich.

Dem Anwender muss bekannt sein, welche Arten von Daten vorhanden und verfügbar sind und es kann für ihn auch relevant sein, in welchen Zeitintervallen die Ereignisse erfasst bzw. vom Datenstrom geliefert werden. Für eine sinnvolle Anfrage ist aber besonders wichtig, den Wertebereich der Anfrageelemente zu kennen bzw. zu erkennen. Dafür ist eine Darstellung nötig, die z.B. ältere Daten und Verläufe über einen betrachteten Zeitraum angemessen aufzeigen kann oder eine Anzeige die eine Überlappung von Werten eines früheren Datensatzes mit dem aktuellen Datenstrom darstellt. Beides kann dabei helfen, das Wissen des Anwenders um die Anfragefähigkeiten und -möglichkeiten zu erweitern.

Zudem ist eine EPL-Anfrage auch immer mit dem Vorhaben verbunden, einen Datenstrom auszuwerten. D.h. es wird versucht, mittels der EPL-Anfrage aus Datenströmen Informationen zu gewinnen. Dabei kann ein Datenstrom aber sehr viele und auch komplexe Ereignisse enthalten. Die Möglichkeiten dabei, aus dem Datenstrom diverser Ereignisse und Ereignisarten die Interessanten auszuwählen, können sehr groß und bei der Betrachtung mehrerer Datenströme sehr unübersichtlich werden.

Um eine bessere Übersicht zu erhalten, könnte eine gefilterte Ansicht der Ereignisdatenströme bereits bei der Anfragestellung Vorteile bringen. Eine Filterung und Einschränkung von verfügbaren Ereignissen kann interessantere Ereignisse in den Vordergrund rücken und somit neue Anfragemöglichkeiten bewirken.

All die verschiedenen Problembereiche, in denen eine Visualisierung eine unterstützende Rolle spielen kann, sind nachfolgend nochmal zusammengefasst.

- **Anzeige und Bestimmung von Views**
 - **Anzeige und Bestimmung von zeitlichen Verläufen**
 - **Anzeige und Bestimmung von kausalen Abhängigkeiten**
 - **Auswahl und Anzeige der Ereignisdatenströme**
 - **Auswahl und Anzeige der Arten von Ereignissen**
 - **Anzeige älterer Werteverläufe von Ereignissen**
 - **Anzeige und Bestimmung der Wertebereiche von Ereignissen**
 - **Auswahl und Festlegung der Gruppierungen von Ereignissen**
 - **Auswahl und Anzeige von gefilterten Ereignissen**
-

Bei all den verschiedenen Problemen und Schwierigkeiten kann eine visuelle Unterstützung die EPL-Anfragestellung erleichtern und für den Anwender intuitiver gestalten.

3.4 Zusammenfassung

Die Motivation für eine mögliche Visualisierungsunterstützung bei der EPL-Anfragestellung wurde in diesem Kapitel ausführlich behandelt. Es wurde im ersten Abschnitt das in dieser Arbeit repräsentativ ausgewählte CEP-System Esper vorgestellt. Als Vertreter für alle anderen verwendet es wie die meisten CEP-Systeme eine SQL-basierte EPL-Anfragesprache. Die sogenannte Esper-EPL ist eine recht mächtige Anfragesprache, mit der sich neben den bekannten SQL-Klauseln und -Statements die für Datenströme relevanten Zeit- und Längenfenstern definieren lassen. Mit dem Konzept der Views lassen sich zudem Sichten auf die Datenströme bestimmen, so dass nicht immer ein kontinuierlicher Datenstrom betrachtet werden muss. Zusätzlich bietet die Esper-EPL Pattern an, mit denen sich vor allem zeitliche Beziehungen zwischen Ereignissen beschreiben lassen.

Die formulierten EPL-Anfragen werden von der Esper-Engine entgegen genommen und entsprechend der Anfrage verarbeitet. Die Esper-Engine arbeitet dabei derart, dass von ihr nicht ständig Anfragen auf Datenströme getätigt werden, sondern die Datenströme werden von der Esper-Engine verarbeitet, während sie durch das CEP-System Esper hindurch fließen.

Im zweiten Abschnitt wurden beispielhaft Anwendungsszenarien beschrieben, in denen die Möglichkeiten einer unterstützenden Visualisierung für die EPL-Anfragestellung skizziert sind. Anhand dieser Szenarien und weiteren Anwendungsfällen sind des weiteren im letzten Abschnitt die Probleme und Schwierigkeiten bei einer EPL-Anfrageformulierung beleuchtet worden. Dabei wurde deutlich, dass eine Visualisierung besonders für die Probleme beim Anzeigen, Auswählen und Eingrenzen von Ereignissen für die EPL-Anfragestellung eine Hilfe sein kann.

Mit dem Wissen um die Problematik bei der EPL-Anfragestellung können im nächsten Kapitel die möglichen Visualisierungsunterstützungen für das Visualisierungswerkzeug evaluiert werden.

4 Evaluation der Visualisierungsunterstützung bei der Anfragestellung

Im folgendem Kapitel werden zunächst Grundbegriffe zur Visualisierung erklärt, allgemeine Kriterien für Visualisierungen aufgelistet und danach die speziellen Anforderungen hinsichtlich der visuellen Unterstützung der Anfragestellung aufgeführt. Anschließend werden die verschiedenen Visualisierungstechniken vorgestellt und bewertet. In einer abschließenden Bewertung werden die Vor- und Nachteile der Techniken in Bezug auf die Visualisierungsunterstützung zusammengefasst.

4.1 Grundlagen der Visualisierung und allgemeine Anforderungen

Die Visualisierung hat nach [SM00] die Aufgabe, eine Datenmenge so visuell zu repräsentieren, dass damit eine effektive Auswertung möglich ist. Das Ziel einer Visualisierung wird folgendermaßen beschrieben:

„Das Ziel ist es, mit Hilfe der Visualisierung die Dinge so darzustellen, wie sie tatsächlich vorliegen und sich wirklich ereignet haben, und den Anwender in die Fähigkeit zu versetzen nicht nur zu sehen, sondern auch zu erkennen, zu verstehen und zu bewerten.“

Es wird also eine visuelle Repräsentation der Datenmenge erstellt, die durch verschiedene Sichtweisen auf die Daten und Analysen erlauben, verborgene Zusammenhänge aufzudecken und wichtige Informationen hervorzuheben.

Die Zielsetzung der Visualisierung lässt sich dabei in die drei Stufen Präsentations-, Konfirmations- und Explorationsvisualisierung einteilen. Bei der Präsentationsvisualisierung ist das Ziel, alle Daten in einer geeigneten Visualisierung darzustellen, damit Sachverhalte und Datenwerte veranschaulicht werden. Die konfirmative Visualisierung zielt auf das Überprüfen und Verifizieren von zuvor aufgestellten Hypothesen. Dafür kann zumeist das Betrachten von Ausschnitten oder Teilansichten auf die Datenmenge wichtig sein. Die Explorationsvisualisierung dient der (interaktiven und) ungerichteten Suche nach Informationen und Strukturen in den Daten. Durch die Visualisierung kön-

nen Beziehungen und Abweichungen zwischen Daten mittels optischer Auffälligkeiten aufgedeckt werden.

Aus den Zielen lassen sich folgende allgemeine Anforderungen an Visualisierungen ableiten:

A01: Expressivität Eine Visualisierung darf nur die in einer Datenmenge enthaltene Informationen und nur diese visualisieren. Expressivität ist primär von der Struktur und der Art der Datenmenge abhängig, die dargestellt werden soll, so dass sich nicht jede Visualisierungstechnik für jede Datenmenge eignet.

A02: Effektivität Die Effektivität einer Visualisierung wird als den Grad der optimalen Ausnutzung bei der Darstellung bezüglich der visuellen Wahrnehmungsfähigkeiten des Anwenders, den Anwendungszielen und der charakteristischen Eigenschaft des Ausgabegerätes bezeichnet. Erfüllen mehrere Visualisierungstechniken das Expressivitätskriterium, so repräsentiert das Effektivste die Datenmenge am besten.

A03: Angemessenheit Angemessenheit beschreibt das Verhältnis zwischen Aufwand und Nutzen einer Visualisierung. Es setzt das Maß für die Kosten für die visuelle Darstellung gegen die Effektivität derselbigen. Als Kosten werden Rechen- und Ressourcenaufwand aufgefasst, die zur Generierung der Visualisierung notwendig sind.

4.2 Spezifische Anforderungen des Visualisierungswerkzeugs

Aus den Anwendungsszenarien für die Erstellung einer EPL-Anfrage ergeben sich die folgenden Anforderungen an das Visualisierungswerkzeug. Sie teilen sich in funktionale und nichtfunktionale Anforderungen auf.

4.2.1 Funktionale Anforderungen

Das zu entwickelnde Visualisierungswerkzeug soll folgende funktionale Anforderungen bei der Anfragestellung erfüllen, wobei die ersten drei Punkte die zentralen Eigenschaften sind:

V01: Übersicht über die vorhandenen Sensoren Die Visualisierungsansicht soll eine praktische Übersicht über die vorhandenen Sensoren bieten.

V02: Übersicht aktueller Sensorwerte Die Visualisierungsansicht soll eine aktuelle Übersicht über die vorhandenen Sensorwerte geben.

V03: Übersicht historischer Datenwerte Die Visualisierungsansicht soll eine Übersicht über ältere Datenwerte, also deren Verlauf über die Zeit anzeigen können. (Anzeige aller Datenwerte einer Variablen)

-
- V04: Übersicht historischer Datensätze** Die Visualisierungsansicht soll eine Übersicht über ältere Datenwerte mehrerer Variablen gleichzeitig anzeigen können, also deren Verlauf mehrerer Variablen gleichzeitig über die Zeit anzeigen können. (Anzeige aller Datenwerte eines Datensatzes)
- V05: Möglichkeit logische Verknüpfungen zu setzen** Das Werkzeug soll dem Anwender ermöglichen, direkt aus der visuellen Darstellung logische Verknüpfungen setzen zu können, zum Beispiel durch das Markieren zweier Datenwerte.
- V06: Suchmöglichkeit** In der Visualisierungsansicht soll es die Möglichkeit geben, nach Daten, Sensoren oder Metainformationen zu suchen.
- V07: Hervorheben von Sensorwerten** Die Visualisierung soll das Hervorheben von Werten ermöglichen, zum Beispiel durch Färben von Werten.
- V08: Zoom-Möglichkeiten** Die Visualisierungsansicht soll Zoomen ermöglichen. Dadurch können zum Beispiel interessante Teile durch das Hineinzoomen in den Vordergrund vergrößert werden.
- V09: Filtermöglichkeiten** Die Visualisierungsansicht soll eine Filterung der angezeigten Daten ermöglichen. Dadurch können zum Beispiel nicht relevante Informationen durch das Filtern ausgeblendet und die interessanten Daten besser zusammen betrachtet werden.
- V10: Zeit- und Mengenfenster** Die Visualisierung soll das Erstellen von Zeit und Mengenfenstern unterstützen, zum Beispiel durch die Möglichkeit, Zeitbereiche oder Mengen von Daten zu bestimmen oder die dafür nötige Eingabewerte anzuzeigen.
- V11: Wechsel der Visualisierungsansicht** Es soll ermöglicht werden, innerhalb der Anwendung zwischen verschiedenen visualisierten Ansicht hin und her zu wechseln, die dieselben Daten auf eine andere Weise darstellen.
- V12: Angemessen schnelle Visualisierung** Die Visualisierung sollte in einer schnellen Ansicht die Datenwerte visualisieren. D.h. die Forderung des Anwenders, nicht lange warten zu müssen, erfüllen.
- V13: Aktualitätsanzeige der visualisierten Datenströme** In der Visualisierungsansicht soll stets erkennbar sein, wie aktuell die verwendeten Datensätze sind.
- V14: Überblick über die Metainformationen** In der Visualisierungsansicht soll der Anwender die Möglichkeit haben, sich Metainformationen von Sensoren sowie deren Sensordatenströme einblenden zu lassen.
- V15: Möglichkeit der Ergänzung von Kontextinformationen innerhalb der Visualisierung** Es soll dem Anwender möglich sein, weitere Kontextinformationen zu den visualisierten Daten hinzuzufügen. So können Anwender zum Beispiel eigenes Wissen
-

mit dem System verknüpfen und so durch neue Kontexterzeugung möglicherweise weitere Informationen gewinnen.

V16: Möglichkeit der rollenbasierten Benutzung Es soll möglich sein, verschiedene vordefinierte Ansichten auszuwählen, so dass zum Beispiel Administratoren mehr Auswahlmöglichkeiten erhalten als einfache Anwender.

V17: Nachvollziehbare Visualisierung (teilweise) Die Visualisierungsansicht soll für den Anwender möglichst nachvollziehbar sein. Es soll ersichtlich sein, wo die Daten herkommen, die visualisiert wurden.

V18: Rekonstruierbare Visualisierung (teilweise) Die Visualisierungsansicht soll ermöglichen, Rückschlüsse auf die verwendeten Daten zu geben und dafür mindestens die Anzeige der reinen Quelldaten anbieten können.

4.2.2 Nichtfunktionale Anforderungen

Des Weiteren sollen dem Visualisierungswerkzeug folgende nichtfunktionale Anforderungen an Software grundsätzlich zugesichert werden [WIK10a]:

Einfache Erweiterbarkeit Die Visualisierungsunterstützung soll zu einem späteren Zeitpunkt einfach erweiterbar sein.

Modularität Die Visualisierung soll modular erweiterbar sein.

Zuverlässigkeit Die Unterstützung soll zuverlässig arbeiten.

Benutzbarkeit Die Visualisierungsunterstützung soll einfach erlernbar und benutzbar sein.

Änderbarkeit Es soll möglich sein, die Visualisierungsunterstützung später einfach zu ändern.

4.3 Abgeleitete Anforderungen aus Datenverarbeitungszielen

Bei der Verarbeitung von Daten lassen sich nach [SM00] folgende Bearbeitungsziele als allgemeingültig ansehen. Aus diesem Grund werden von diesen Zielen Anforderungen für die Visualisierung abgeleitet. Sie werden als Datencharakteristische Anforderungen bezeichnet.

D01: Erkennung von Korrelationen Die Visualisierungsansicht soll die Anzeige und Erkennung von Zusammenhängen zwischen zwei oder mehreren Variablen oder zwischen Datenwerten im Beobachtungsraum über die Zeit unterstützen.

- D02: Erkennung von Clustern** Die Visualisierungsansicht soll die Anzeige von Datenwerten mit gemeinsamen Eigenschaften, also das Anzeigen und Erkennen von Clustern unterstützen.
- D03: Erkennung von Häufigkeiten** Die Visualisierungsansicht soll die Anzeige von Häufigkeitsverteilungen der Datenwerte unterstützen.
- D04: Erkennung durch Vergleiche** Die Visualisierungsansicht soll den Vergleich von einzelnen Datenwerten unterstützen.
- D05: Erkennung von Verteilungen** Die Visualisierungsansicht soll die Anzeige und Erkennung von Datenwertverteilungen wie zum Beispiel Extremwerte unterstützen.
- D06: Größe darstellbarer Datenmenge** Die Visualisierungsansicht soll eine möglichst große Menge an Datenwerten gleichzeitig darstellen können.

4.4 Gesamtanforderungen an das Visualisierungswerkzeug

Neben den allgemeinen grundsätzlichen Anforderungen an Visualisierungen kommen die speziellen Anforderungen hinzu, die sich aus den Aufgaben des zu entwickelnden Visualisierungswerkzeugs ergeben. Zusammen bilden diese die Gesamtanforderungen an das Visualisierungswerkzeug. Der nächste Abschnitt listet die Gesamtanforderungen in einem Anforderungskatalog mit Gewichtung auf, gefolgt von der Beschreibung der Bewertungsmethode für das Visualisierungswerkzeug.

4.4.1 Anforderungskatalog mit Gewichtung

Im folgenden werden die gesamten Anforderungen an das Visualisierungswerkzeug in tabellarischer Form in einem Anforderungskatalog zusammengefasst. Er setzt sich zusammen aus den drei bekannten Anforderungskategorien und deren Gewichtungen.

Die drei Anforderungskategorien lauten:

- **A Allgemeine Anforderung**
Das sind Anforderungen, die eine Visualisierung immer erfüllen muss. Diese Anforderungen werden nicht bewertet.
 - **V Visualisierungsanforderung (direkt)**
Diese Anforderungen werden direkt aus den spezifischen Anforderungen an das Visualisierungswerkzeug übernommen.
 - **D Datencharakteristische Anforderungen (indirekt)**
Dies sind Anforderungen, die sich indirekt aus der Visualisierung von Daten ergeben.
-

Die Wichtigkeit für die Verwendung im Visualisierungswerkzeug wird dabei nach folgendem Schema angegeben:

- **Wichtig**

Wichtig sind Anforderungen, die für das Ziel einer visuellen Unterstützung möglichst erfüllt werden sollen. Sie bilden die grundlegenden Bausteine für den Unterstützungswert einer Visualisierung im Visualisierungswerkzeug.

- **Optional**

Als optionale Anforderungen zu betrachten sind die, die den unterstützenden Wert des Visualisierungswerkzeugs nicht wesentlich verringern, falls sie nicht oder nur schlecht erfüllt werden. Werden sie hingegen gut erfüllt, werten sie Visualisierungstechniken zusätzlich auf.

- **Nicht gewichtbar**

Für das Visualisierungswerkzeug nicht gewichtbare Anforderung sind solche, die bei Visualisierungen prinzipiell vorausgesetzt werden. Diese werden bei Visualisierungen von Daten implizit vorgegeben, so dass sie hier nicht gewichtet werden.

Bei den allgemeinen Anforderungen an Visualisierungen (siehe Kapitel 4.1) handelt es sich um solche nicht gewichtbare Anforderungen. Sie werden aber weiterhin der Vollständigkeit halber mit aufgeführt.

Die Tabelle 4.1 zeigt den gesamten Anforderungskatalog an das Visualisierungswerkzeug mit Gewichtung in einer Übersicht.

4.4.2 Bewertungsmethode

Die Bewertung der Visualisierungstechniken geschieht in zwei Schritten. Zunächst wird zu jedem einzeln vorgestellten Vertreter der Visualisierungstechniken eine Kurzbewertung gegeben, in der besondere Vor- und Nachteile der Darstellung aufgezeigt werden. Dies geschieht in einer Auflistung mit folgendem Schema:

Schema der Kurzbewertung:

Wertung	Visualisierungstechnik
+	Eine Besonderheit der Visualisierung, die positiv zu bewerten ist.
-	Eine Besonderheit der Visualisierung, die negativ zu bewerten ist.

Anhand dieser Wertungen und weiterer Aspekte der unterschiedlichen Visualisierungstechniken wird abschließend eine zusammenfassende Gesamtbewertung der Visualisierungstechniken erstellt. Dabei werden die Visualisierungstechniken mittels der oben erstellten gewichteten Anforderungsliste bewertet.

Typ	Anforderung	Wichtig	Optional	Nicht gewichtbar
A01	Expressiv			✓
A02	Effektiv			✓
A03	Angemessen			✓
V01	Übersicht über Datenquellen	✓		
V02	Übersicht aktueller Datenwerte	✓		
V03	Übersicht historischer Datenwerte	✓		
V04	Übersicht historischer Datensätze	✓		
V05	Ansicht verknüpfter Datenwerte	✓		
V06	Identifizierbarkeit und Auswählbarkeit von Einzelwerten	✓		
V07	Hervorheben von Datenwerten	✓		
V08	Zoom-Möglichkeiten	✓		
V09	Filtermöglichkeiten	✓		
V10	Zeit- und Mengenfenster anzeigen	✓		
V11	Wechsel zwischen Visualisierungsansichten	✓		
V12	Schnelle Visualisierung	✓		
V13	Aktualitätsanzeige möglich		✓	
V14	Überblick über Metainformationen		✓	
V15	Möglichkeit der Datenergänzung		✓	
V16	Rollenbasierte Benutzung ermöglichen		✓	
V17	Nachvollziehbare Visualisierung		✓	
V18	Rekonstruierbare Visualisierung		✓	
D01	Erkennung von Korrelationen	✓		
D02	Erkennung von Clustern	✓		
D03	Erkennung von Häufigkeiten	✓		
D04	Erkennung durch Vergleiche	✓		
D05	Erkennung von Verteilungen	✓		
D06	Größe darstellbarer Datenmenge	✓		

Tabelle 4.1: Anforderungskatalog an das Visualisierungswerkzeug mit Gewichtung

4.5 Vorstellung der verschiedenen Visualisierungstechniken

Im Folgenden wird eine Übersicht über die vier grundsätzlich verschiedenen Visualisierungstechniken [SM00] für multivariate¹ und multidimensionale Daten gegeben und zu jeder Technik beispielhafte Vertreter mit einer Kurzbewertung vorgestellt.

Im Bereich Data Mining² wird u.a. von [Che99] oder [Oel02] noch die Gruppe der Graphbasierten Techniken zur Visualisierung aufgeführt. Da diese Techniken im Grunde als Basiseigenschaften in den anderen Visualisierungstechniken aufgehen und sich vor allem unter die Hierarchischen Techniken einordnen lassen, werden sie hier nur der Vollständigkeit halber aufgezählt, werden aber im Weiteren nicht mehr einzeln behandelt.

4.5.1 Geometrische Techniken

Geometrische Techniken benutzen geometrische Transformationen und Projektionen, um Daten und deren mögliche Beziehungen zueinander darzustellen. Zu dieser Klasse von Visualisierungstechniken gehören u.a. Scatterplot-Matrizen, Prosection Views, Hyperslices, Parallele Koordinaten und Spiral Graphs. Im Nachfolgendem werden die einzelnen Techniken vorgestellt.

4.5.1.1 Scatterplot-Matrizen

Ein *Scatterplot*, zu deutsch Streudiagramm, bezeichnet die bivariate Darstellung zweier Variablen eines m -dimensionalen Merkmalraumes, die von Interesse sind. Dazu werden die Werte der beiden Variablen in zwei senkrecht zueinander stehenden Achsen in einem Diagramm, entsprechend dem Wertebereich passend skaliert, aufgetragen. Jeder Punkt in dem durch die Achsen aufgespannten Raum entspricht genau einer möglichen Wertekombination. Werden Daten in das Diagramm abgebildet, so gibt die dabei entstehende Punktwolke einen visuellen Einblick auf die Werteverteilung und mögliche Korrelationen bezüglich des Merkmalraums der ausgewählten Variablen. Legt man für jede vorhandene Variablenkombination nun einen Scatterplot an und ordnet diese matrizenartig, so entsteht eine *Scatterplot-Matrix* [Cle93] mit (m^2-m) Scatterplots, die eine vollständige Darstellung des Merkmalsraums entspricht. In der Hauptdiagonalen steht anstelle der Kombination der Variablen mit sich selbst oft die Variablenbezeichnung. Es können aber beispielsweise auch die univariaten Datenwerte oder Histogramme der Variablen eingetragen werden.

Die Abbildung 4.1 zeigt ein Beispiel für eine Scatterplot-Matrix, in der vier Datenwerte von Autos verschiedener Typen aufgetragen sind. Jedes Paar von Werten der Autos

¹Bei multivariaten Daten hängt ein Ausdruck im Gegensatz zu univariaten Daten nicht nur von dem Wert einer, sondern gleich mehrerer Variablen ab (vgl. [Wik10e]).

²„Unter Data Mining ... versteht man die systematische Anwendung von Methoden, die meist statistisch-mathematisch begründet sind, auf einen Datenbestand mit dem Ziel der Mustererkennung.“ [WIK10b]

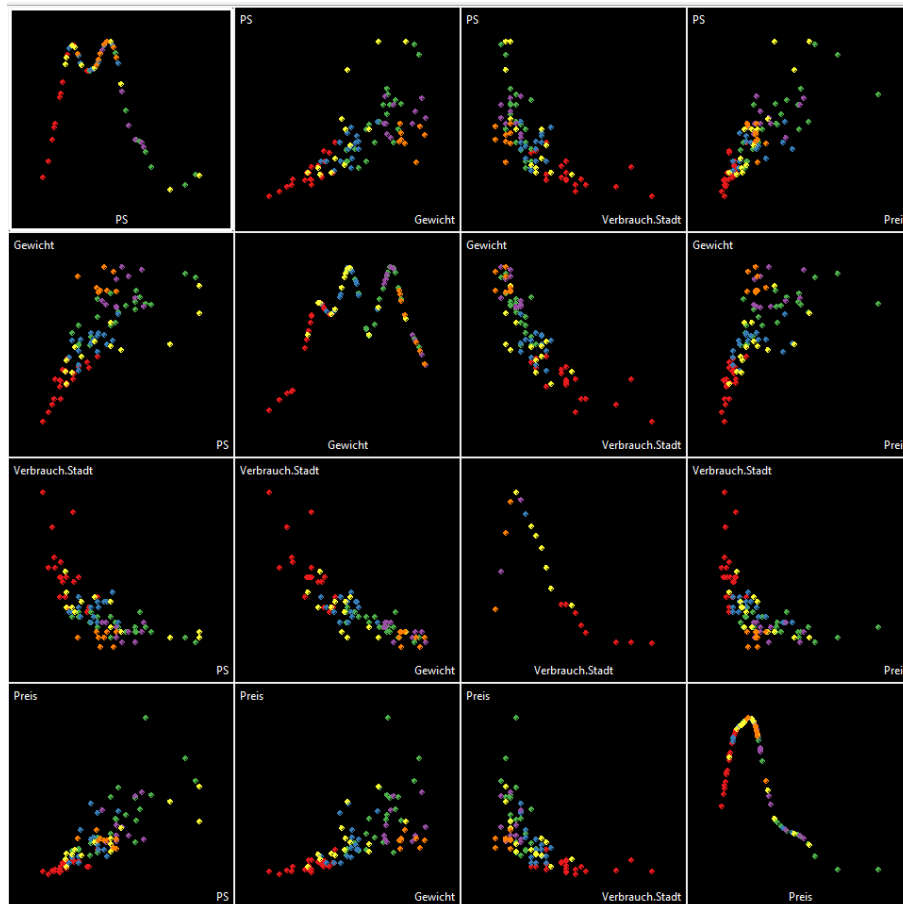


Abbildung 4.1: Beispiel einer Scatterplot-Matrix mit Fahrzeugdaten aus dem Jahre 1993 von 6 verschiedener Automobilklassen: rot = Kleinwagen, blau = Kompaktklasse, grün = Mittelklasse, lila = Oberklasse, orange = Mehrzweckfahrzeuge, gelb = Sportwagen (Eigene Darstellung erstellt mit Ggobi [Ggo08])

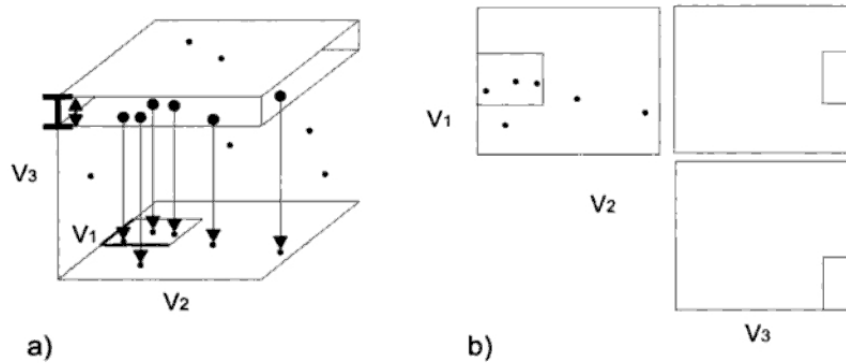


Abbildung 4.2: Veranschaulichung des Prinzips der Prosection View: a) Bildung von Teilsichten in einem 3-dimensionalen Raum; b) Bildung der dazugehörigen Dreiecksmatrix [SM00]

ist in zwei transponierten Matrizen als Punkt dargestellt. Durch die paarweise Verknüpfung der Eigenschaften und Visualisierung werden mögliche Zusammenhänge deutlich gemacht und statistische Besonderheiten hervorgehoben. Beispielsweise ist im oberen rechten Scatterplot schnell erkennbar, dass die PS-Stärke eines Autos stark mit dem Preis korreliert.

Wertung	Scatterplot-Matrizen
+	Durch eine Beschriftung der Achsen mit einer Skala ist die Identifikation von Werten möglich.
+	Das Erkennen von Korrelationen zweier Merkmale wird durch die paarweise Verknüpfung sehr erleichtert.
-	Durch die kompakte Darstellung entstehen leicht Überlappungen in der Ansicht.

4.5.1.2 Prosection Views

Die *Prosection Views* (vgl. [STDS95]) können als eine Erweiterung der Scatterplot-Matrizen verstanden werden. Sie kombinieren die Projektion in den Scatterplot-Matrizen mit der Selektion einer bestimmten Wertebereichseigenschaft und bieten somit eine Teilansicht auf die gesamten Datenmenge. In der Prosection View werden also nur die Punkte im Merkmalsraum angezeigt, die die Selektionsbedingung erfüllen. In Abbildung 4.2 wird das Prinzip der Prosection View veranschaulicht. Die Grafik a) zeigt dabei die Erstellung von Teilsichten in einem 3-dimensionalen Raum und b) die Bildung der dazugehörigen Dreiecksmatrix.

Wertung	Prosection Views
+	Es werden besonders die spezifische Untersuchungen multidimensionaler Wertekombinationen ermöglicht.
+	Die Visualisierung ist durch die Auswahl der Selektionsbedingungen im Besonderen als interaktive Darstellung sinnvoll nutzbar.
-	Es gibt keine vollständige Sicht auf die Datenwerte.

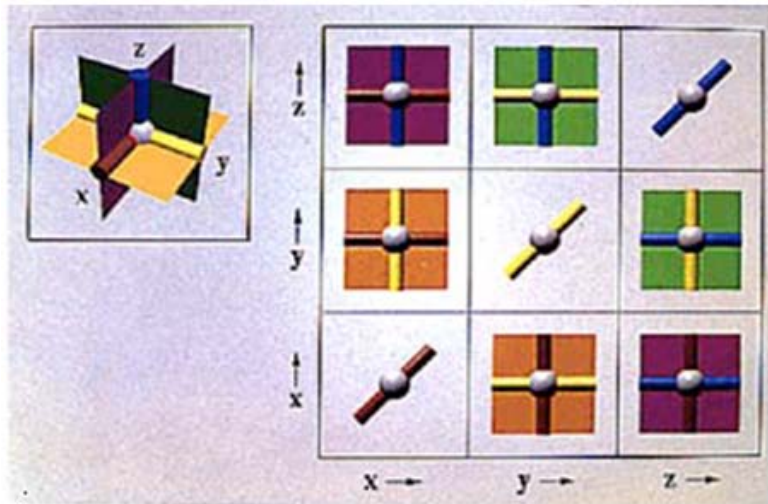
4.5.1.3 Hyperslices

Auch *Hyperslices* [WL93] ist eine Erweiterung der Scatterplot-Matrizen. Allerdings findet hier nur eine Selektion der Daten, die von Interesse sind, durch Schnitte im Raum statt und keine Projektion. Hyperslices sind (m^2-m) 2-dimensionale Schnitte durch einen m -dimensionalen Merkmalsraum, die sich in einem aktuellen Punkt, *Current Point* genannt, schneiden. Abbildung 4.3a zeigt das Konzept eines Hyperslices anhand eines 3-dimensionalen Merkmalsraum. Auf der linken Seite ist der Current Point der Hyperslices als Kugel im Raum dargestellt und rechts die dazugehörigen Schnitte, die durch Farbhervorhebung schnell zugeordnet werden können. Abbildung 4.3b zeigt wie die Visualisierung einer 4-dimensionalen Funktion mittels Hyperslices aussehen kann. Hier markiert der rote Punkt in den Ausschnitten den Current Point der Hyperslices.

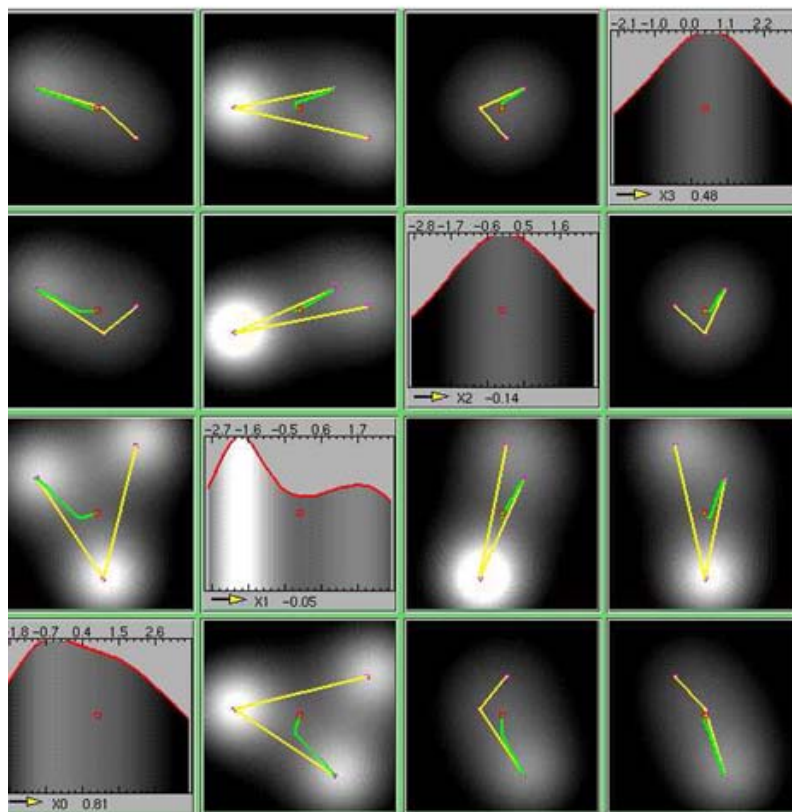
Wertung	Hyperslices
+	Durch die Beschriftung der Achsen ist eine Identifikation von Werten möglich.
+	Cluster und Verteilungen sind bei geeigneter Auswahl des Current Point gut sichtbar.
+	Korrelationen zweier Variablen können intuitiv wahrgenommen werden.
-	Korrelationen zwischen mehr als zwei Variablen sind schlecht erkennbar.
-	Aus der Ansicht kann keine eindeutige Aussage über Häufigkeitsverteilungen gemacht werden.
-	Es ist eine Erklärung und Einarbeitung nötig, um die Visualisierung zu verstehen und nutzen zu können.

4.5.1.4 Parallele Koordinaten

Bei *Parallele Koordinaten* [SM00] wird für jede Dimension eines Merkmalsraums eine Achse mit angepasst skaliertem Wertebereich konstruiert und in einem äquidistantem Abstand parallel aufgetragen. Merkmale einer Variablen werden als Punkte auf den Achsen aufgetragen und mit Linien verbunden. Diese Art der Darstellung erlaubt eine vollständige Visualisierung eines m -dimensionalen Merkmalsraums, schränkt aber durch Überdeckung identischer Werte die Vollständigkeit ein. In Abbildung 4.4 ist die Visualisierung



a)



b)

Abbildung 4.3: Veranschaulichung von Hyperslices: a) Konzept eines Hyperslices; [SM00] b) Hyperslices einer 4-dimensionalen Funktion [WL93]

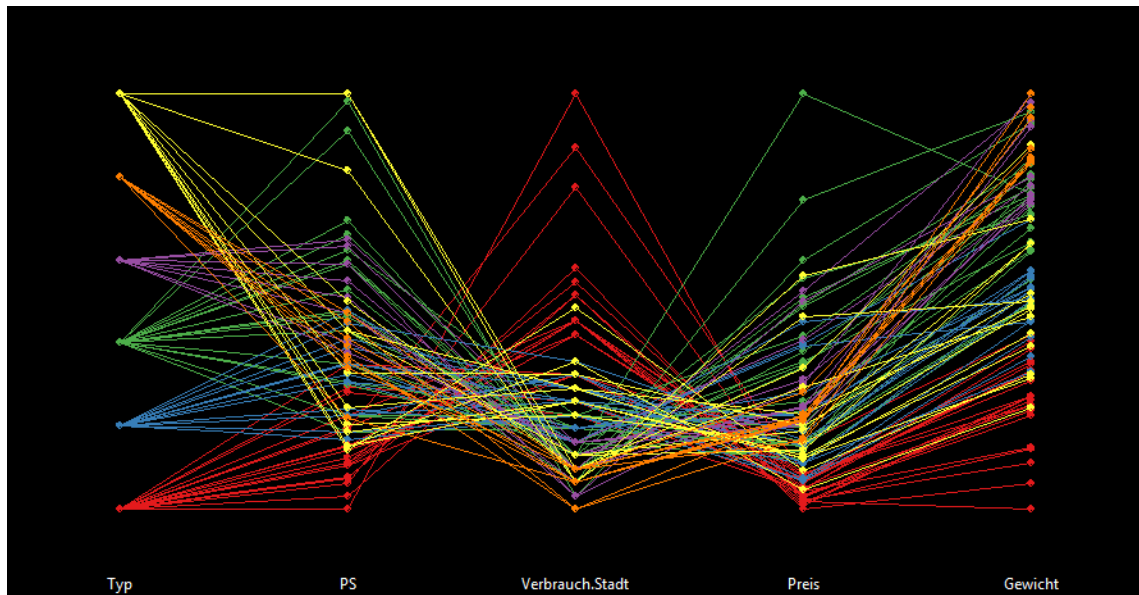


Abbildung 4.4: Darstellung der Fahrzeugdaten von Abbildung 4.1 als Parallele Koordinaten zusätzlich mit Fahrzeugtyp als erste Achse (Erstellt mit Ggobi [Ggo08])

der Fahrzeugdaten aus Abbildung 4.1 als Parallele Koordinaten zu sehen. Des Verständnisses halber ist der Fahrzeugtyp neben der vorhandenen Farbrepräsentation als Koordinatenachse mit aufgenommen.

Wertung	Parallele Koordinaten
+	Durch die Beschriftung der Achsen ist eine Identifikation von Werten möglich.
+	Eine vollständige Visualisierung eines m-dimensionalen Datensatzes ist mit dieser Technik möglich.
+	Korrelationen zwischen benachbarten Achsen lassen sich gut erkennen.
-	Überdeckungen identischer Datenwerte schränken die Vollständigkeit und Eindeutigkeit der Darstellung ein.
-	Darstellung wird bei vielen Merkmalen und sehr vielen Datensätzen schnell unübersichtlich.

4.5.1.5 Parahistogramme

Parahistogramme [SM00] sind eine Erweiterung der Parallelen Koordinaten und ergänzen die Parallele Koordinaten Darstellung um Histogramme in den Koordinatenachsen. Die Achsendarstellung der Variablenmerkmale weicht grafischen Balken, die die Verteilungsverhältnisse der Werte darstellen. Entsprechend der Anzahl an Werten, die durch den Balkenbereich verlaufen, wird der Bereich grafisch gefüllt. Haben beispielsweise alle Variablen dieselbe Merkmalsausprägung, so verlaufen sie alle durch denselben Punkt und der Balken ist vollständig gefüllt. In Abbildung 4.5 ist ein solches Beispiel zu sehen. Dar-

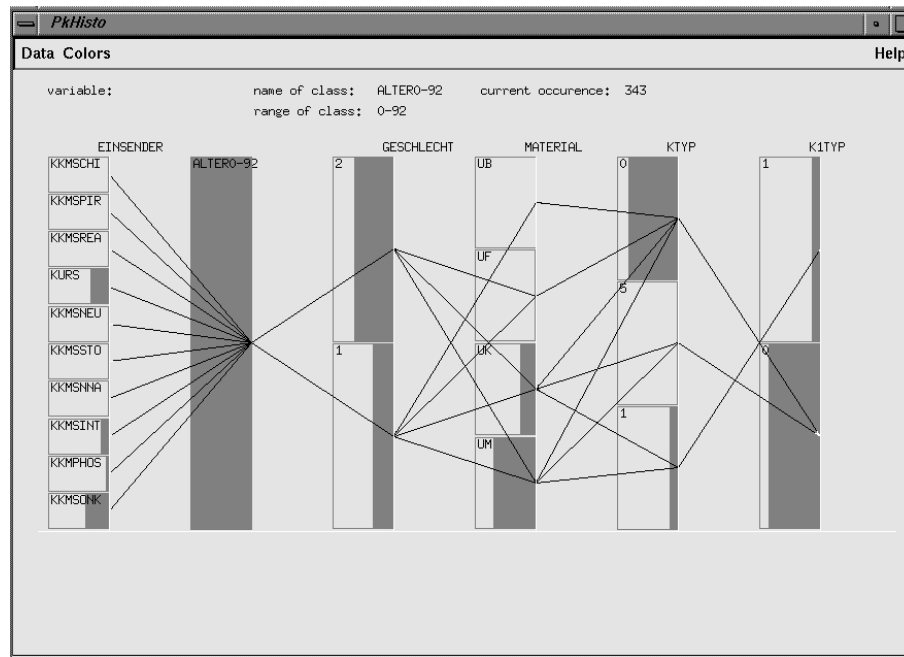


Abbildung 4.5: Parahistogramm einer mikrobiologischen Datenmenge [SM00]

gestellt ist eine mikrobiologische Datenmenge, dessen zweites Merkmal die Altersangabe ist. Da alle Datensätze das Merkmal „Alter zwischen 0 bis 92 Jahren“ erfüllen, wird der Balken „Alter0-92“ komplett gefüllt. Bei der Aufteilung nach Geschlecht ist ein Verhältnis 2:1 zu sehen, so dass hier die Balkenfüllung oben entsprechend zu Zweidrittel und die untere Hälfte nur zu einem Drittel geschehen ist.

Wertung	Name
+	Häufigkeitsverteilungen lassen sich intuitiv wahrnehmen.
-	Durch die breite Balkendarstellung anstelle der Achsen ist ein erhöhter Platzbedarf für die Darstellung im Gegensatz zu den Parallelen Koordinaten nötig.

4.5.1.6 Spiral Graphs

Die von [WAM01] vorgestellte Technik *Spiral Graphs* dient der Visualisierung (großer) zeitvarianter Datensätze. In dieser Darstellung werden Daten spiralförmig sequentiell in einem 1-dimensionalen-Diagramm angezeigt. Die einzelnen Daten werden durch Farben, Ikone oder der Linienstärke dargestellt. Durch die Spiraldarstellung werden besonders sich wiederholende Strukturen hervorgehoben und sichtbar gemacht.

In der Abbildung 4.6 ist die Sonnenscheinintensität eines Ortes über mehrere Tage visualisiert. Es ist leicht ersichtlich, wo im Spiral Graph die Tag-Nacht-Grenze verläuft (Tag unten, Nacht oben) und wann die Sonnenscheinintensität zugenommen hat.

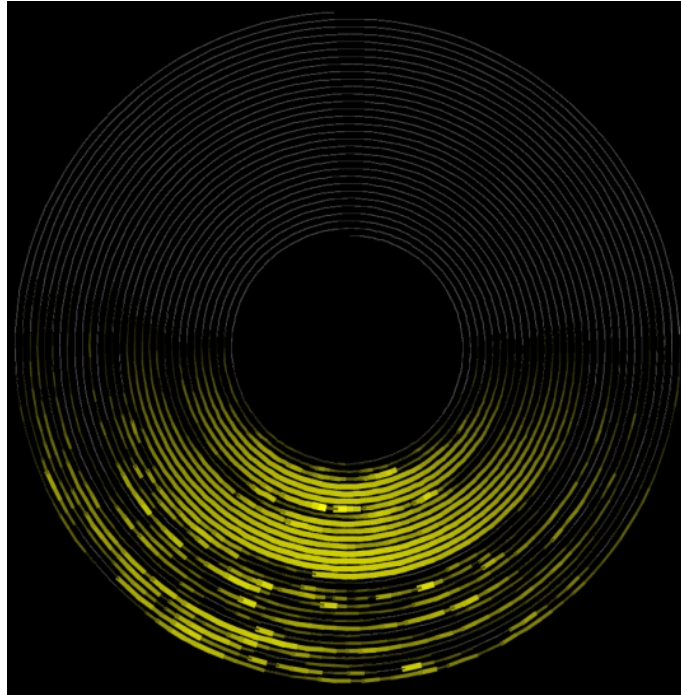


Abbildung 4.6: Spiral Graph Darstellung der Sonnenscheinintensität eines Ortes über mehrere Tage [WAM01]

Wertung	Spiral Graphs
+	Es können effizient sehr großen Datenmengen dargestellt werden.
+	Der Vergleich von Daten einer Nachbarschaft und über Zyklen hinweg wird unterstützt.
+	Periodisches Verhalten wird durch diese Visualisierung besonders gut wahrgenommen.
+	Eignet sich für die Visualisierung von zeitlichen Verläufen.

4.5.2 Ikonenbasierte Techniken

Die Ikonenbasierten Techniken beruhen auf dem Prinzip der Visualisierung von Datenwerten durch die Variation von Eigenschaften (z.B. Farbe, Transparenz) und Primitive (z.B. Form, Größe) graphischer Objekte, die exakt positioniert werden können. Zu diesen Techniken gehören u.a. Bubble Graphs, Stick Figures, Shape Coding, Color Icons und Chernoff Faces, die im Einzelnen kurz vorgestellt werden.

4.5.2.1 Bubble Graphs

Die *Bubble Graphs*-Technik [SM00] basiert auf der Darstellung von Datenmerkmalen als eingefärbter Kreis. Durch diese Darstellung können Merkmale auf insgesamt bis zu 4 Parametern (Position in der X und Y Achse, Größe und Farbe) gemappt und visualisiert werden. In der Abbildung 4.7 ist eine geographische Karte der USA zu sehen, in

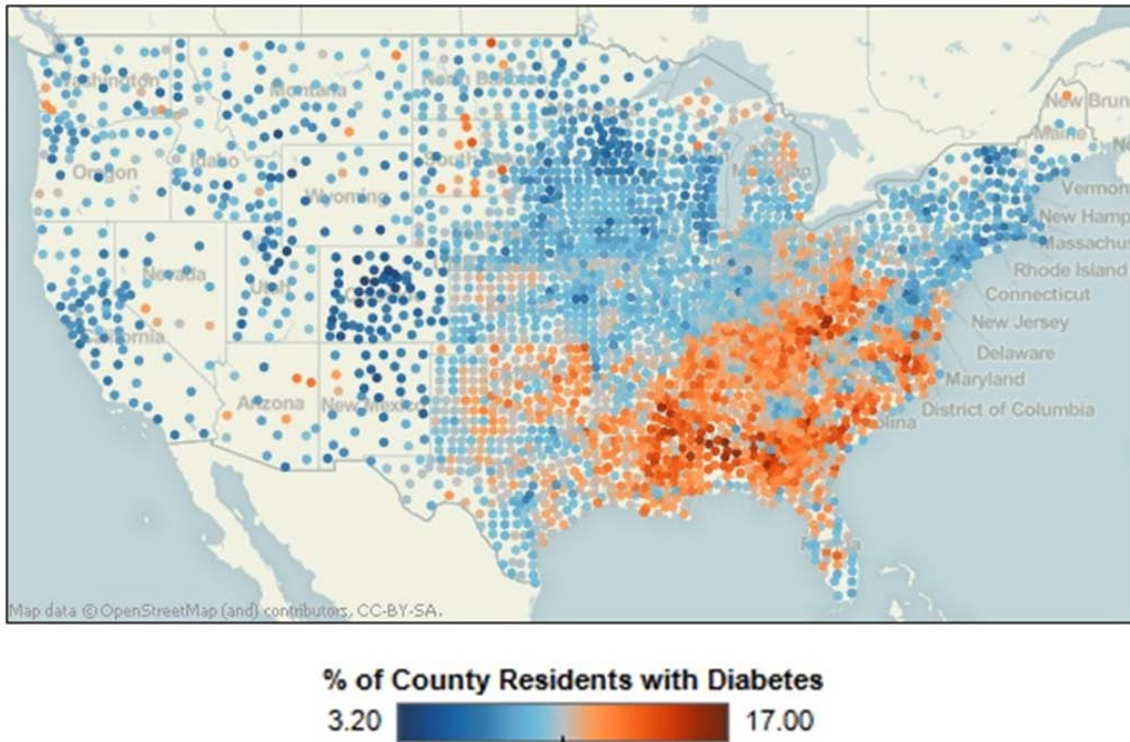


Abbildung 4.7: Bubble Graphs Karte der Diabeteskranken in den USA [tab10]

der durch Bubble Graphs und farblicher Kodierung der prozentuale Anteil an Diabetes Erkrankten visualisiert ist.

Wertung	Bubble Graphs
+	Erlaubt die Visualisierung des vollständigen Datensatzes.
-	Überlappungen beschränken die Darstellungsmöglichkeit auf relativ wenige Merkmale.

4.5.2.2 Stick Figures

Bei *Stick Figures* [PG88] werden Linien als grafische Primitive verwendet. Die einfachste Stick Figure-Ikone besteht aus einer Hauptachse, die durch ihre X- und Y-Position bereits zwei Parameter darstellen kann. Werden weitere Linien hinzugefügt, so ergibt sich ein Stick Figure-Ikone, die durch seine Attributvariierung der Linien z.B. in Farbe, Länge oder Winkel zueinander eine Vielzahl an Datenattributen darstellen kann.

Werden Datensätze mit Stick Figures visualisiert, entstehen durch die unterschiedlichen Charakteristika der Daten Strukturmuster in der Darstellung, die Rückschlüsse auf die Daten erlauben. Mit der Anzahl an darzustellenden Attributen verringert sich allerdings die Unterscheidungsfähigkeit der Datenwerte. Abbildung 4.8 zeigt Beispiele für den Aufbau verschiedener Stick Figure-Ikone. In Abbildung 4.9 ist die Visualisierung von fünfdimensionalen Infrarot-Bildern der großen Seen in den USA mittels dieser Technik zu sehen.

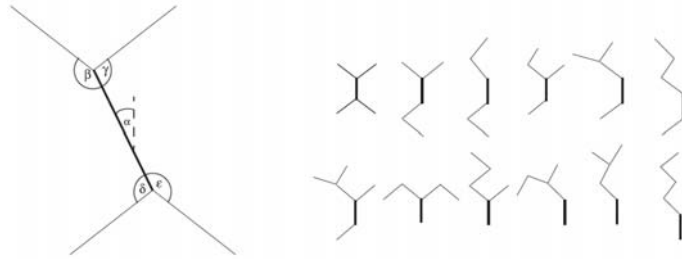


Abbildung 4.8: Beispiel für verschiedene Stick-Figures-Ikone [KK95]

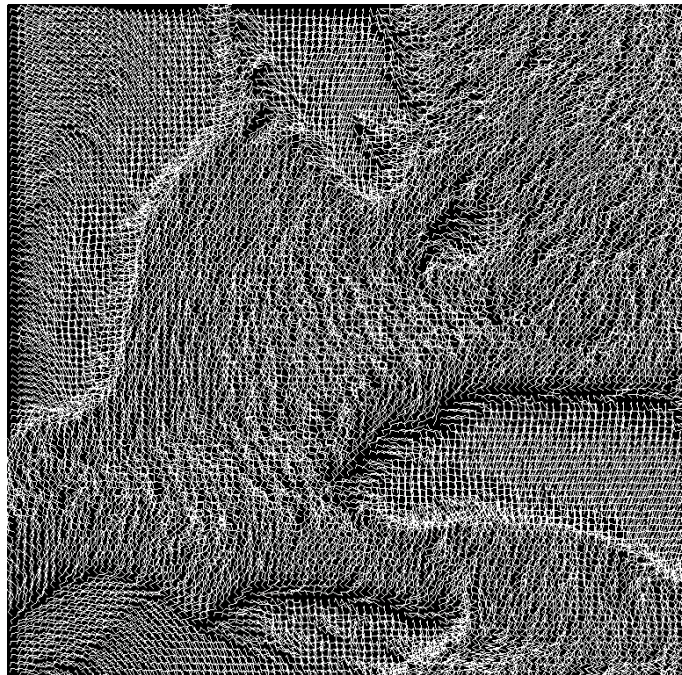


Abbildung 4.9: Stick Figures-Visualisierung der großen Seen in den USA [KK96]

Wertung	Stick Figures
+	Globale Verteilungen werden gut durch die Textur- und Strukturbildung dargestellt.
+	Die Anzahl an darstellbaren Attributvarianten ist theoretisch unbegrenzt.
-	Mit steigender Anzahl an dargestellten Attributen verschwinden die Unterscheidungsmerkmale und verfälschen die Darstellung.

4.5.2.3 Shape Coding

Die *Shape Coding*-Technik [Bed90] basiert auf der Visualisierung von Datenwerten in einem 2-dimensionalen Feld, welches aus kleinen Quadraten besteht. Jedes Kästchen im Feld steht durch seine binäre Codierung in schwarz/weiß für das Vorhandensein oder für das Nichtvorhandensein eines bestimmten Merkmals. Die Quadrate werden zeilenweise aufgetragen und ergeben ein Gitter aus Merkmalsausprägungen der dargestellten Datenwerte, eine sogenannte Shape Coding-Ikone. Diese Ikone werden im einfachsten Fall dann wiederum zeilenweise zu einem Gitternetz angeordnet.

Abbildung 4.10 veranschaulicht den Aufbau einer Shape Coding-Ikone, bei dem die Merkmale 2, 4 und 5 ausgeprägt und somit schwarz kodiert sind. In Abbildung 4.11 kann man die gesamte Datenmenge der mikrobiologischen Untersuchung aus Abbildung 4.5 als Shape Coding betrachten, der das Erkennen von möglichen Mustern und Korrelationen im Datensatz erleichtert.

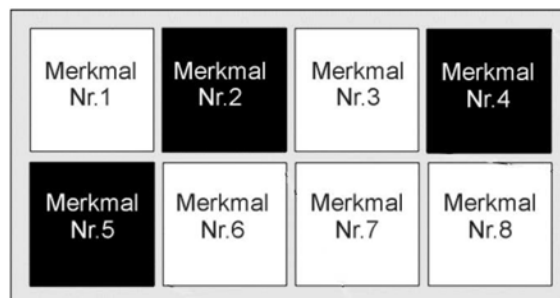


Abbildung 4.10: Aufbau einer Shape Coding Ikone [SM00]

Wertung	Shape Coding
+	Eignet sich besonders für Daten mit begrenzten nominalen Merkmalen, insbesondere für binäre Merkmalsausprägungen.
+	Bietet eine vollständige Gesamtansicht auf die Datenmenge.
+	Korrelationen großer Datensätze können intuitiv wahrgenommen werden.
+	Wahl der Anordnung der Shape Coding-Felder kann die Korrelationswahrnehmung positiv beeinflussen.

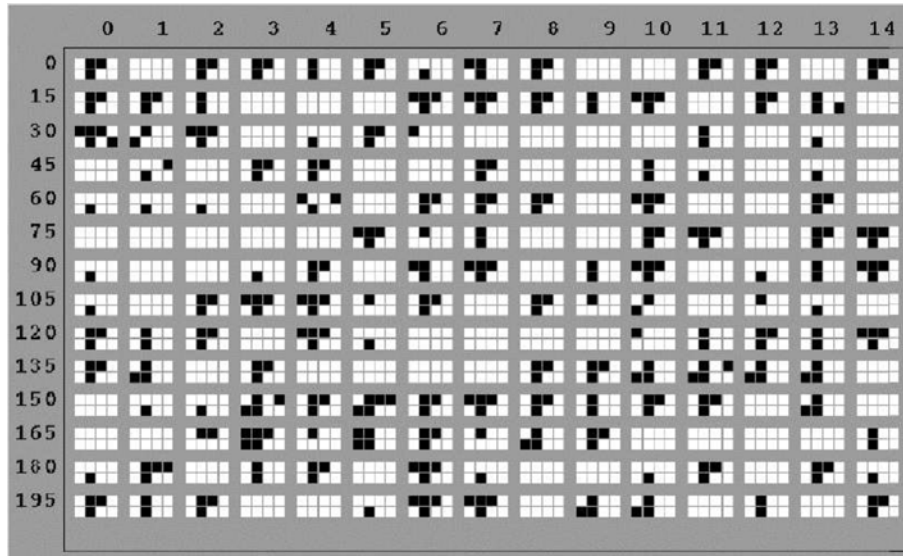


Abbildung 4.11: Shape Coding Visualisierung der mikrobiologischen Merkmale aus der Datenmenge von Abbildung 4.5 [SM00]

4.5.2.4 Color Icons

Color Icons [Lev88] verwendet Farb-Ikone und erweitert die Shape Coding-Technik um zusätzliche Farben in den Zellen und um Variationen in der Zellenteilung und Zellenanordnung. So können die Quadrate durch Linien unterschiedlich unterteilt werden. Anschließend erhalten die Kanten eine Farbe und die dazwischen liegenden Flächen werden entsprechend der anliegenden Kanten linear bzw. bilinear interpoliert eingefärbt. Die Zellen lassen sich dann je nach Relevanz in der Anordnung verändern.

Dieser Ansatz der Color Icon-Visualisierung setzt interpolierbare Werte voraus, um die Zwischenflächen einzufärben. Um diese Einschränkung aufzuheben, färbt ein zweiter Ansatz die Felder entsprechend der Werte direkt ein (vgl. [SM00] und [WB97]). Der erste Ansatz begünstigt die Visualisierung von fließenden Übergängen in den Datenwerten (z.B. kontinuierliche Veränderungen über die Zeit), wohingegen der zweite Ansatz die deutliche Separation von Merkmalsausprägungen unterstützt. Abbildung 4.12 zeigt ein Beispiel für das Aussehen eines Color Icons. Jede Teileinfärbungen des Color Icons steht für eine Ausprägung eines Datenattributes.

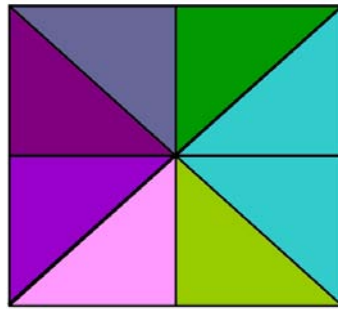


Abbildung 4.12: Beispiel eines Color Icons [SM00]

Wertung	Color Icons
±	Vor- und Nachteilen der Shape Coding-Technik gelten auch hier unter Berücksichtigung der nachfolgenden Einschränkungen und Ergänzungen.
+	Eignet sich zur Darstellung von kontinuierlichen Veränderungen, beispielsweise Verläufe über die Zeit.
+	Texturwahrnehmung wird effektiv unterstützt.
-	Wahl der richtigen Einfärbung der Flächen kann notwendig und kompliziert sein.

4.5.2.5 Chernoff Faces

Bei *Chernoff Faces* werden bis zu 12 Merkmale unter Verwendung eines komplexen Icons in Form eines Gesichtes visualisiert. Dafür werden die Gesichtsformen wie Nase, Ohr und Auge in ihrer Größe und Form entsprechend der Merkmalsausprägung geändert und so kodiert. Bei Chernoff Faces werden Gesichter benutzt, da dies die beim Menschen besonders ausgeprägte Fähigkeit kleinste Veränderungen von Mimiken wahrzunehmen nutzt, um Unterschiede zu verdeutlichen. Abbildung 4.13a zeigt beispielhafte Chernoff Faces mit zwölf unterschiedlichen Gesichtsausdrücken und 4.13b welche Gesichtsmimiken variiert werden können. Eine abgewandelte Variante der Gesichter ist auf der Karte in Abbildung 4.14 zu sehen, in der [TT80] die Lebensbedingungen in Los Angeles mittels Chernoff Faces visualisiert sind.

Wertung	Chernoff Faces
+	Besonders soziale Merkmale lassen sich mit dieser Technik gut visualisieren.
-	Interpretation ist nur anhand einer Legende möglich und kann bei vielen Attributen schwierig und langwierig sein.

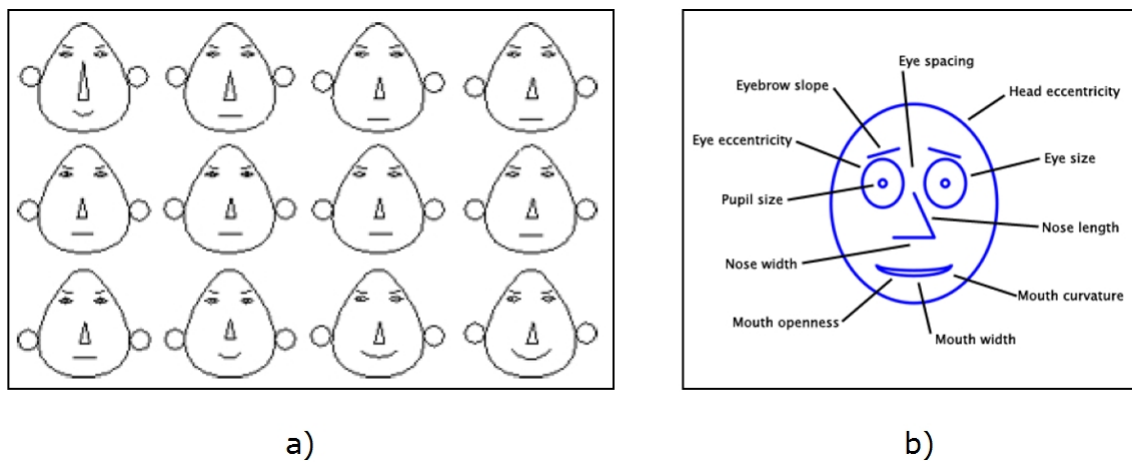


Abbildung 4.13: Beispiele für Chernoff Faces: a) zwölf verschiedene Gesichtsausdrücke [Che73]; b) Variierungspunkte der Gesichtsmimik [BRA10]

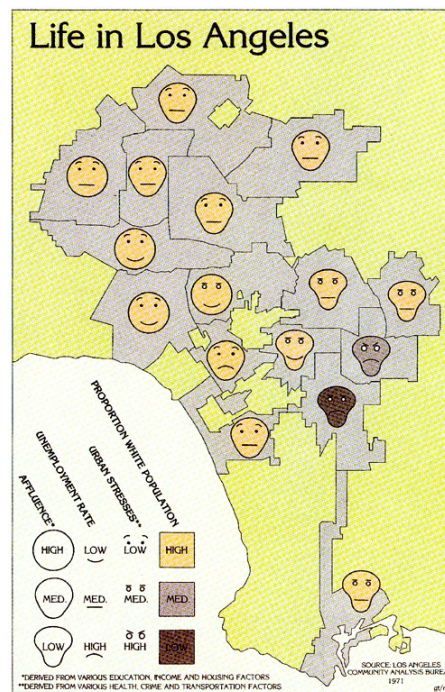


Abbildung 4.14: Visualisierung der Lebensbedingungen in L.A. mittels Chernoff Faces [TT80]

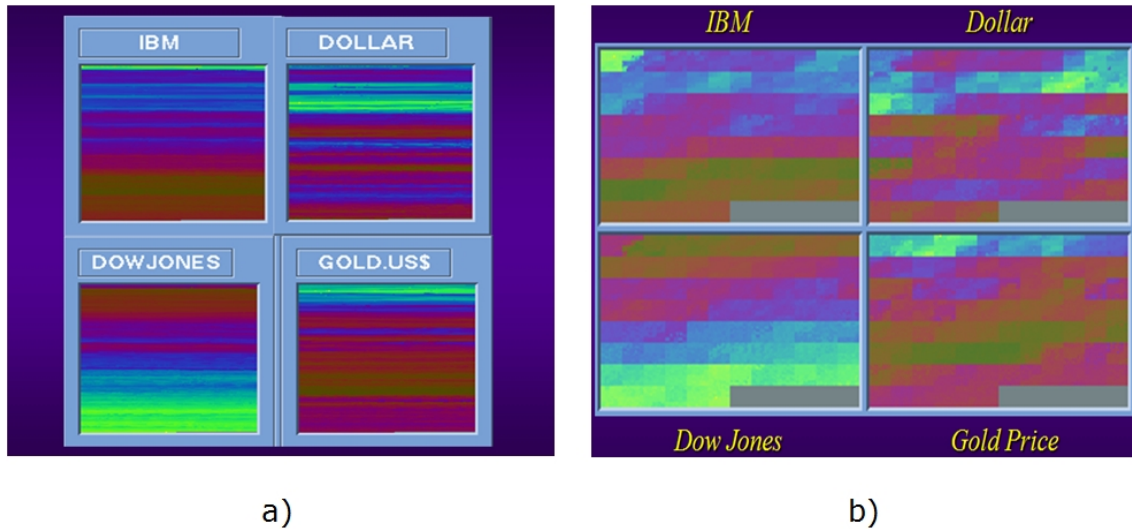


Abbildung 4.15: Zwei Vertreter der Pixelbasierten Techniken: a) Visualisierung von Finanzkursverläufen mittels Einfache Techniken und b) mittels Raumfüllende-Kurven-Techniken [Kei02]

4.5.3 Pixelbasierte Techniken

Die pixelbasierten Techniken beruhen auf dem Prinzip, jeden Datenwert einer Datenmenge auf genau einem Pixel zu projizieren und in der Farbe zu kodieren. Durch diese minimale Darstellung lassen sich mit diesen Techniken sehr große Datenmengen visualisieren. Diese Gruppe der Techniken unterscheiden sich ausschließlich in der Art der Reihenfolge der Abbildung der Datenwerte. Folgende Auswahl an pixelbasierten Techniken wird vorgestellt: Einfache Techniken, Raumfüllende-Kurven-Techniken, Zwei-Schritte-Techniken und Recursive-Pattern-Techniken.

4.5.3.1 Einfache Techniken

Bei den *Einfachen Techniken* [Kei02] werden die Datenwerte zunächst serialisiert und dann einfach zeilen- oder spaltenweise angeordnet. Abbildung 4.15a zeigt als Beispiel die Anordnung von Finanzdaten. Durch die Einfachheit der Visualisierungsgrundlage sind Darstellungen mittels dieser Technik oftmals nicht leicht interpretierbar und somit weniger erfolgversprechend.

Wertung	Einfache Techniken
+	Eignet sich zur Visualisierung sehr großer Datensätze, da auf die Darstellungsfläche heutiger Bildschirme weit mehr als eine Millionen Pixel passen.
+	Bietet i.A. eine einfache vollständige 2-dimensionale Ansicht auf Merkmalsausprägungen einer Datenmenge.
-	Datenwerte müssen für die Darstellung serialisiert werden.
-	Ist in der Regel schwer interpretierbar und dadurch im Nutzen relativiert.
-	Eine Identifikation und der Vergleich von Einzelwerten ist nicht möglich, da die Pixeldarstellung zur Unterscheidung von Werten zu klein sind.

4.5.3.2 Raumfüllende-Kurven-Techniken

Auch bei den *Raumfüllenden-Kurven-Techniken* [Kei02] werden die Datenwerte zuvor serialisiert. Anschließend werden sie hier aber entlang von Raumfüllende-Kurven angeordnet, die der Visualisierung ihre Vorteile aufsetzen wie beispielsweise die Zusammenziehung benachbarter und ähnlicher Datenwerte [Tie08]. In Abbildung 4.15b sind dieselben Finanzwerte von der Abbildung 4.15a der Einfachen-Techniken als Raumfüllende-Kurven-Technik dargestellt. Zu sehen ist die Ausrichtung der serialisierten Daten an einer Peano-Hilbert-Kurve.

Wertung	Raumfüllende-Kurven-Techniken
±	Vor- und Nachteilen der Einfachen Techniken gelten auch hier unter Berücksichtigung der nachfolgenden Ergänzung.
+	Benachbarte und ähnliche Datenwerte werden näher beieinander dargestellt.

4.5.3.3 Zwei-Schritte-Techniken

Zwei-Schritte-Techniken (*Grouping Technique* [KK94]) arbeiten dem Namen nach in zwei Schritten. Im ersten Schritt werden Datenwerte in Ikone gruppiert. Die Ikone bestehen ähnlich wie die Shape Coding Ikone aus einem Gitternetz von quadratischen Feldern mit dem Unterschied, das die Felder lediglich Pixelgröße besitzen. Auch hier wird durch ihre Farbausprägung der Datenwert kodiert. Im zweiten Schritt werden die Ikone dann nach einem bestimmten Schema angeordnet. Hierbei wird zwischen Spiral-, Achsen- und Gruppenanordnung unterschieden. Als Beispiel ist in Abbildung 4.16 eine Circle Segments Visualisierung [AKK96] zu sehen, welches die spiralförmig Anordnung der Datenfelder verwendet und die Attribute eines Pixels in Segmenten eines Kreises anordnet. Visualisiert sind hier die tagesgenauen Aktienkurse von 50 Aktienwerten des FAZ-Index über einen Zeitraum von 20 Jahren. Die Farbskala wurde dabei so gewählt, dass helle

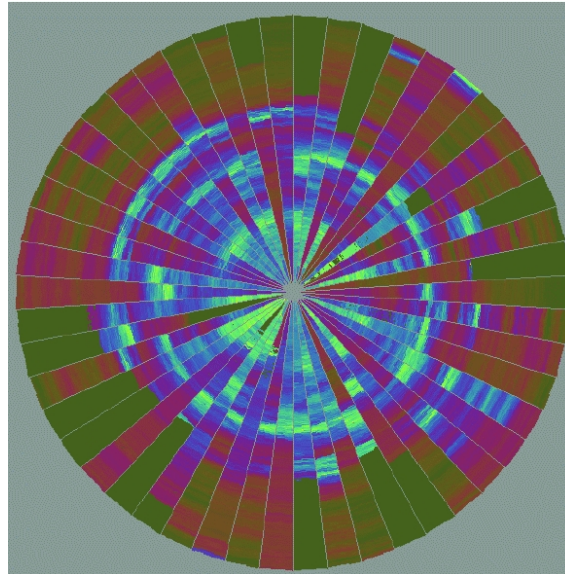


Abbildung 4.16: Ein Beispiel der Zwei-Schritte-Techniken: Das Circle Segment [AKK96]

Farben hohe Kurse und dunkle Farben niedrige Kurswerte bedeuten. Deutlich in der Visualisierung zu erkennen sind ähnliche Aktienverläufe, aber auch Aktien, die sich gegen den Trend verhalten.

Wertung	Zwei-Schritte-Techniken
+	Erlauben die Darstellung von sehr vielen Merkmalsausprägungen.
+	Trends können mit dieser Technik gut visualisiert und erkannt werden.
-	Ist nur mit einer gegebenen Legende oder Erläuterung der Visualisierung sinnvoll nutzbar.
-	Eine Identifikation und der Vergleich von Einzelwerten ist nicht möglich, da die Pixeldarstellung zur Unterscheidung von Werten zu klein sind.

4.5.3.4 Recursive-Pattern-Techniken

Bei den *Recursive-Pattern-Techniken* [KAK95] werden wie bei den Zwei-Schritte-Techniken zuerst Gruppen von Datenwerten angelegt, die durch Pixel dargestellt werden. Anschließend wird für jede Rekursionsstufe eine feste Anzahl Gruppen je Zeile w_i und Spalte h_i bestimmt. Somit besitzt die i -te Rekursionsstufe also $w_i * h_i$ Pixelgruppen der Rekursionsstufe $(i-1)$. Jede Gruppe von Pixeln in den Rekursionsstufen bezeichnet man auch als ein Pattern, sodass also jede Rekursionsstufe $w_i * h_i$ Pattern besitzt. Als Beispiel für diese Art der Visualisierung dient die Repräsentation von fünf verschiedenen Wetterdaten in Abbildung 4.17, die linear angeordnet sind. Dort werden die Daten als Parameter 1 bis

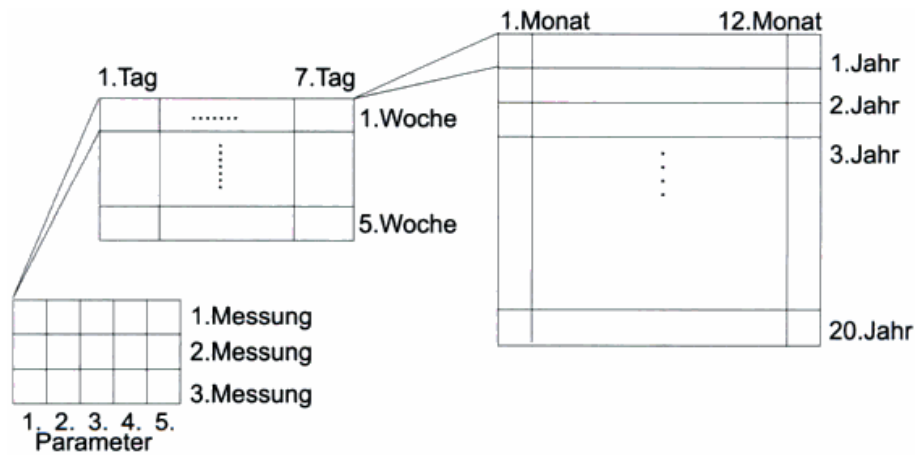


Abbildung 4.17: Veranschaulichung der Recursive Pattern-Techniken: Bildung von rekursiven Pattern [SM00]

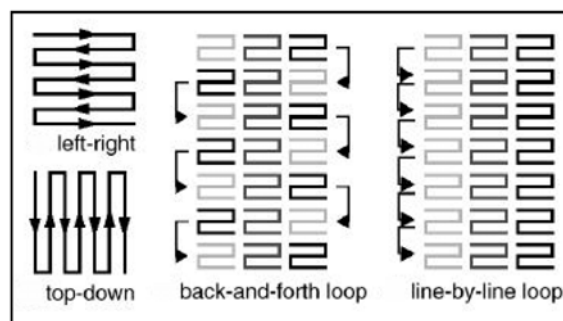


Abbildung 4.18: Vier Beispiele für Anordnungsschemata der Recursive Pattern-Techniken [KAK95]

5 bezeichnet und dreimal am Tag gemessen. In der ersten Rekursionsstufe werden 3×5 -Pattern gebildet, die die täglichen Messungen zeilenweise auflisten. In der zweiten Rekursionsstufe wird ein 5×7 -Pattern bestimmt, welches die Wochenansicht der Tage zeilenweise für einen ganzen Monat kodiert. In einem weiteren dritten Rekursionsschritt wird nun die Monatsansicht für 20 Jahre gebildet, indem ein 20×12 -Pattern gebildet wird. Jedes Pattern dieser Rekursionsstufe entspricht jetzt der Darstellung eines Monats. Anstelle der linearen Anordnung, welche im Beispiel verwendet wurde, können auch noch andere Anordnungsstrategien für die Recursive-Pattern-Techniken angewandt werden. Abbildung 4.18 zeigt vier weitere Anordnungsschemata, die die Aussagekraft der Darstellung verbessern können.

Wertung	Recursive Pattern-Technik
±	Vor- und Nachteilen der Zwei-Schritte-Techniken gelten auch hier unter Berücksichtigung der nachfolgenden Einschränkungen und Ergänzungen.
+	Durch gezielte Anordnung / Gruppierung sind gut interpretierbare Bilder erzeugbar.
+	Ermöglicht sehr gut die intuitive Erkennung von Häufigkeiten und Verteilungen.
+	Bietet durch die Rekursionsschichten eine kompakte Darstellung der Merkmalsausprägungen.
+	Wahl des richtigen Anordnungsschema kann die Aussagekraft der Visualisierung nochmals verstärken.
-	Das Rekursionsprinzip muss zum Verständnis der Visualisierung erläutert werden.

4.5.4 Hierarchische Techniken

Hierarchische Techniken visualisieren Daten durch das hierarchische Unterteilen in Unterstufen. Diese Unterteilung kann im Darstellungsraum oder im Merkmalsraum stattfinden. Die Techniken Dimensional Stacking und Worlds-within-Worlds verwenden die erstere Darstellung, Cone Trees und Treemaps letztere. Diese Techniken werden im Folgendem vorgestellt.

4.5.4.1 Dimensional Stacking

Beim *Dimensional Stacking* [LWW90] wird eine hierarchische Visualisierung im 2-dimensionalen Darstellungsraum vorgenommen. Das Prinzip beruht auf der Einbettung eines Koordinatensystems in ein anderes Koordinatensystem. Bei vorhandenen Variablen V_1 bis V_m , deren Wertebereiche durch die Kardinalzahlen K_1 bis K_m definiert sind, werden zunächst zwei (beliebige) Variablen V_i und V_j herausgenommen (mit $i, j \in \{1, 2, \dots, m\}$ und $i \neq j$). Diese beiden Variablen werden kombiniert und spannen ein $K_i * K_j$ Gitternetz als erstes Koordinatensystem auf, das den Darstellungsraum unterteilt.

Dieser Vorgang wird rekursiv solange wiederholt, bis alle Werte paarweise zugeordnet wurden und eine Verschachtelung der Gitternetze im Darstellungsraum entstanden ist. Falls die Anzahl der Variablen ungerade ist, wird eine Dummy-Variable eingeführt.

Das Ergebnis auf der letzten Ebene ist ein Koordinatensystem aus ineinander verschachtelten Koordinatensystemen, die alle Wertekombinationen der Datenwerte repräsentieren. Jede Gitterzelle des Koordinatensystems entspricht genau einer Datenwertkombination.

Abbildung 4.19 veranschaulicht die Dimensional Stacking-Technik schematisch. Abbildungen a) bis c) zeigen die Koordinatensysteme eines 6-dimensionalen Merkmalsraums.

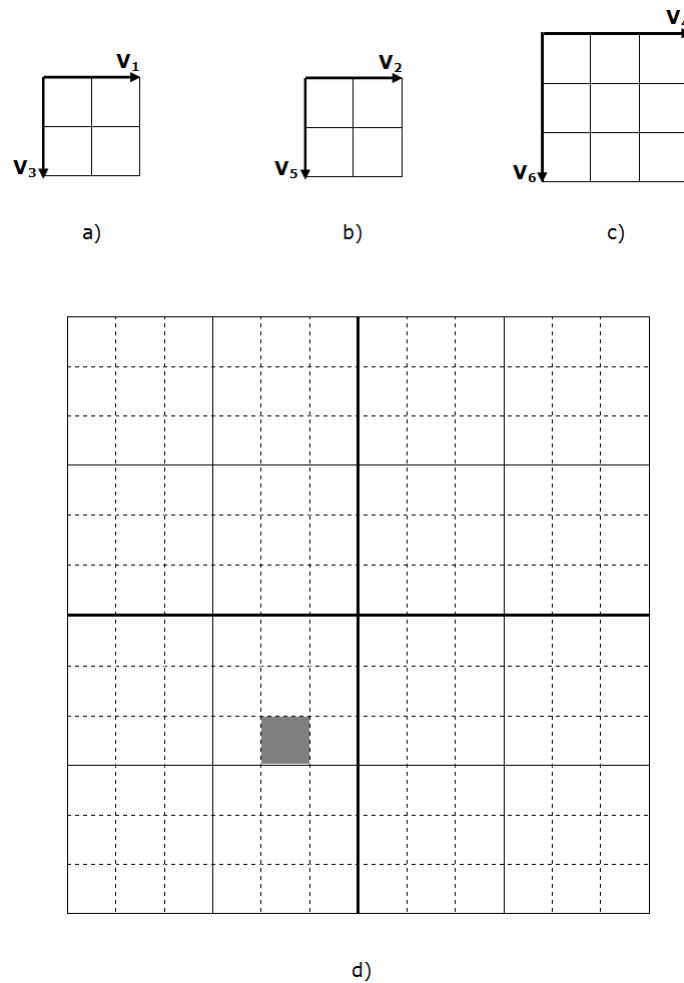


Abbildung 4.19: Schematischer Aufbau des Dimensional Stacking: a), b) und c) Koordinatensysteme eins, zwei und drei eines 6-dimensionalen Merkmalsraumes; d) Abschließende Ineinanderverschachtelung der Koordinatensysteme von links nach rechts (Graues Feld: Datenwertekombination (1, 2, 2, 1, 2, 3)) (Eigene Darstellung nach [SM00])

mes. Die zugehörigen Variablen V_1 bis V_6 haben die Kardinalzahlen $K_1 = 2$, $K_2 = 2$, $K_3 = 2$, $K_4 = 3$, $K_5 = 2$ und $K_6 = 3$. Daraus ergibt sich das auf Abbildung d) zu sehende Dimensional Stacking Koordinatensystem durch Ineinanderverschachtelung der Koordinatensysteme von links nach rechts. Die grau hinterlegte Gitterzelle darin entspricht der Datenwertekombination (1, 2, 2, 1, 2, 3) für die Variablen V_1 bis V_6 . In Abbildung 4.20 ist noch ein Beispiel für Dimensional Stacking zu sehen. Abgebildet sind verschiedene Attribute wie Längengrad, Breitengrad, Tiefe und Qualität von Ölförderdaten.

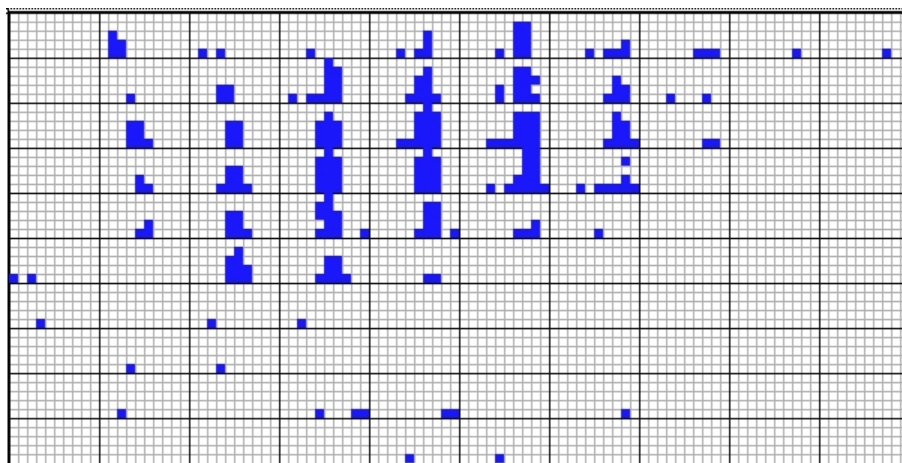


Abbildung 4.20: Dimensional Stacking von Ölförderdaten [Kei02]

Wertung	Dimensional Stacking
+	Bietet eine verlustfreie Darstellung multivariater Daten.
+	Kombination mit einer zusätzlichen Farbkodierung ermöglicht eine noch informativere Visualisierung.
-	Datenwerte der ersten Ebene dominieren die Darstellung und können zu irreführende und verfälschte Darstellungen führen.
-	Bei tiefer Verschachtelung verschwinden Unterteilungen und nur noch Trends bleiben sichtbar.
-	Mit steigender Größe der Mächtigkeit der Wertebereiche wird die Visualisierung unübersichtlicher.

4.5.4.2 Worlds-within-Worlds

Die Technik *Worlds-within-Worlds* [FB90] verwendet eine Verschachtelung im 3-dimensionalen Darstellungsraum. Dabei bilden hier drei Datenwerte aus der Datenwertmenge V_m ein 3-dimensionales Koordinatensystem. Die Skalierung des Koordinatensystems entspricht dem Wertebereich der jeweiligen Daten. Jeder Punkt im nun errichteten Koordinatensystem spannt ein weiteres Koordinatensystem mit drei Datenwerten auf. Dies wird rekursiv wiederholt bis alle Datenwerte ein Koordinatensystem gebildet haben, das letzte muss dabei nicht 3-dimensional sein.

Das innerste Koordinatensystem gibt einen Ausschnitt einer möglichen Wertebelegung der äußeren Datenvariablen auf Koordinaten im innersten System wider. Es entstehen so Gebilde, die Werteverteilungen darstellen können ähnlich wie beim Dimensional Stacking.

Die Worlds-within-Worlds-Technik ist explizit zur interaktiven Exploration von Datenmengen entwickelt worden und entspricht vom Prinzip her der selektiven Anfragestellung in klassischen Datenbanksystemen. Die Auswahl einer Position im Koordinatensystem steht für eine Selektion und das Resultat aller Anfragefolgen ist das innerste Koordinatensystem.

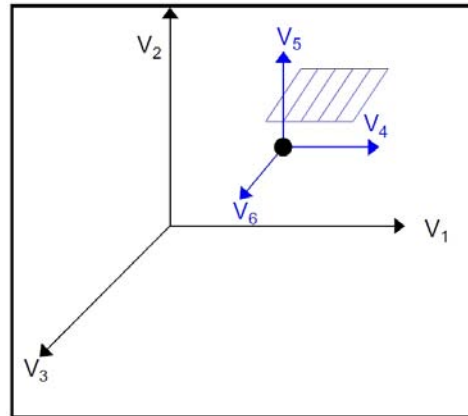


Abbildung 4.21: Prinzip der Worlds-within-Worlds-Visualisierung [SM00]

Abbildung 4.21 veranschaulicht die Worlds-within-Worlds-Technik. Die ersten drei Variablen V_1 bis V_3 spannen hierbei das äußere Koordinatensystem des 6-dimensionalen Beispielraums auf. In diesem lässt sich wiederum an einer beliebigen festen Koordinate ein inneres Koordinatensystem mit den Variablen V_4 bis V_6 erzeugen. In diesem innersten System bilden Punkte Werteverteilungen, die durch eine Schattierung hervorgehoben sind.

Wertung	Worlds-within-Worlds
+	Ideal für interaktive Systeme, wie zum Beispiel die Interaktion mit einem 3D Datenhandschuh, geeignet.
-	Bietet nur Teilsichten auf die Daten und keine Gesamtansicht.
-	Auswahl und Anzeige eines bestimmten Datenwerts ist schwierig.

4.5.4.3 Cone Trees

Cone Trees [RMC91] erzeugen eine Hierarchisierung im Merkmalsraum, also eine Unterteilung des Wertebereichs von Datenwerten. Bei dieser Technik werden Kegel zur Repräsentation verwendet. An der Kegelspitze sitzt ein Vaterknoten, auf der Mantelfläche des Kegels befinden sich die Nachfolgerknoten, die Kinderknoten. Besitzen Kinderknoten ebenso Nachfolgerknoten, so kommt die Spitze eines weiteren Kegels auf den Kinderknoten. Erst auf der letzten Mantelfläche werden die Datenwerte angezeigt. Jede Kegel-ebene steht für eine bestimmte Hierarchiestufe. Im Gegensatz zu anderen Visualisierungstechniken sind Manipulationsfunktionen wie das Drehen und Ausblenden von Kegeln integraler Bestandteil dieser Visualisierungstechnik.

Abbildung 4.22 zeigt eine Beispielvisualisierung von Cone Trees in euklidischer Form. Am oberen Rand befindet sich ein roter Vaterknoten, von dem mehrere grüne Kinderknoten durch Linien verbunden abgehen. Diese wiederum lassen jeweils weitere blaue Kinderknoten eine Stufe tiefer herab, auf dessen Mantelfläche dann die Datenwerte stehen. Die Linienverknüpfung der Cone Tree-Ikone in dieser Darstellung verringert Über-

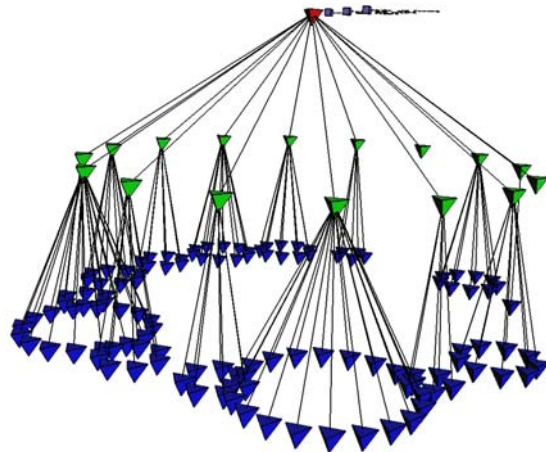


Abbildung 4.22: Euklidische Cone Trees Visualisierung [MB95]

lappungen und erhöht die Sichtbarkeit.

Wertung	Cone Trees
+	Ist eine auf Interaktion ausgelegte Visualisierung.
+	Strukturelle Zusammenhänge lassen sich gut darstellen.
+	Durch Hierarchisierung im Wertebereich (z.B. in Beobachtungsfälle oder Datensätze pro Beobachtungsfall) können Abhängigkeiten erkannt und analysiert werden.
-	Darstellung benötigt eine große Darstellungsfläche.
-	Bereits bei wenigen Daten treten bei einfacher Realisierung Verdeckungsprobleme auf.

4.5.4.4 Treemaps

Auch *Treemaps* [Shn92] wenden eine Hierarchisierung im Merkmalsraum an. Zur Visualisierung der Datenwerte werden Rechtecke genommen, dessen erste Rechteckfläche für den Wurzelknoten steht. Die Kinderknoten werden hierarchisch ineinander verschachtelt dargestellt und gefärbt. Auf der letzten Hierarchieebene befinden sich dann die reinen Datenwerte. Die Größe und die Größenverhältnisse der Datenwerte selbst werden dabei durch die Flächengröße der Rechtecke visualisiert.

Die Abbildung 4.23 zeigt die Visualisierung eines Festplattenlaufwerks als Treemap. Dabei werden Strukturen, Verteilungen und Größenverhältnisse auf einen Blick sichtbar. Deutlich zu sehen sind im rechten Bereich zwei im Verhältnis recht große Dateien und auf der linken Seite unzählig viele kleinere Dateien. Mit etwas Basiswissen und diesen Informationen lässt sich ableiten, dass Ersteres Auslagerungsdaten sind und letztere einen Großteil der kleinen Dateien *Dynamic Link Libraries (DLLs)* repräsentieren. Es handelt sich

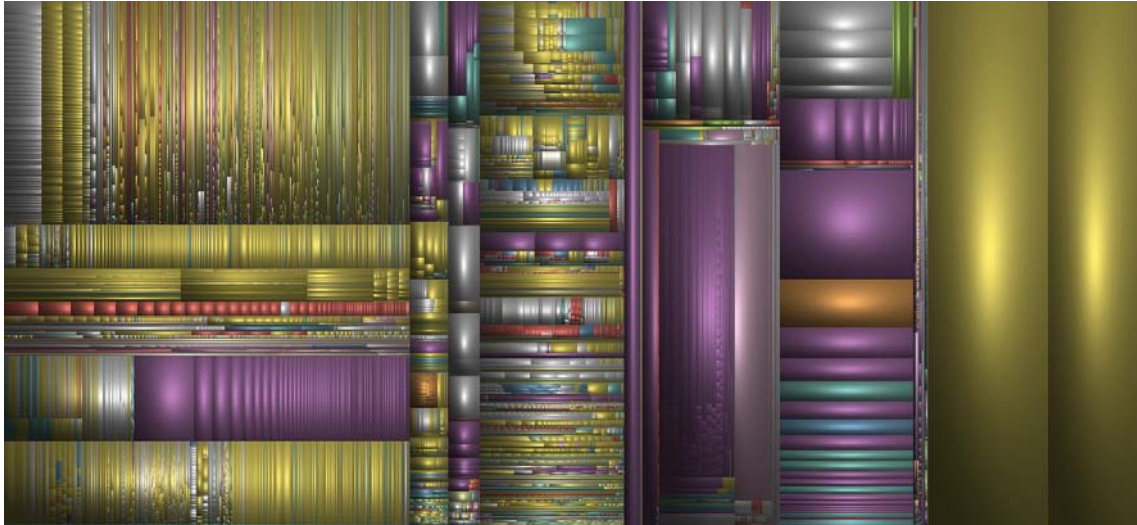


Abbildung 4.23: Treemap Ansicht einer Computerfestplatte (Eigene Darstellung erstellt mit SequoiaView [BGH⁺00])

also bei der Abbildung um die Visualisierung eines (Betriebs-)Systemlaufwerks.

Wertung	Treemaps
+	Größenverhältnisse zwischen Datenwerten werden gut sichtbar dargestellt.
+	Strukturen und Verteilungen werden intuitiv erkennbar gemacht.
+	Darstellung ist für eine Interaktion geeignet.
+	Bietet eine erfassbare vollständige Darstellung an.
-	Werden zu tiefe Verschachtelung visualisiert, sind nicht mehr Details, sondern nur noch grobe Verhältnisse erkennbar.

4.6 Zusammenfassende Bewertung der Visualisierungstechniken

Nachdem alle Visualisierungstechniken anhand einiger ihrer Vertreter vorgestellt und kurz bewertet wurden, erfolgt hier eine Gesamtbewertung mit einer abschließenden tabellarischen Auflistung aller Anforderungskriterien.

Geometrische Techniken

Geometrische Techniken bieten eine verlustfreie und eindeutige Abbildung des Merkmalsraumes in die 2-dimensionale Ebene oder den 3-dimensionalen Raum. Die Methode dieser Visualisierungstechnik ist es, mittels geometrischer Transformation und Projektion multidimensionale Daten und deren Beziehungen zueinander darzustellen. Das Ablesen einzelner Werte wird dabei grundsätzlich ermöglicht und das Erkennen von Korrelationen zweier Variablen unterstützt. Eine zusätzliche Farbkodierung beispielsweise bei den

Parallelen Koordinaten kann die Erkennungsrate erhöhen.

Da sich mit steigender Anzahl an zu visualisierenden Dimension die Abbildungsmöglichkeiten in den 2D- oder 3D-Raum erhöhen und bei schlechter Wahl als ungünstig erweisen können, ist eine zusätzliche Abbildungsstrategie erforderlich. Zudem wird die geometrische Darstellung mit zunehmender Anzahl an Dimensionen und gleichzeitiger Anzeige von sehr vielen Datensätzen immer unübersichtlicher. Auch wenn prinzipiell mit dieser Technik eine Visualisierungsfähigkeit von riesigen Datensätzen und Dimensionen gegeben wird, schränken Überlappungen in der Darstellungsfläche die Visualisierung auf etwa 1000 Datensätze ein.

Geometrische Techniken sind grundsätzlich für die interaktive Benutzung geeignet, da sie Selektion und Veränderung in der Darstellung unterstützen und keine Hindernisse für die Einblendung von Metainformationen aufstellen.

Ikonenbasierte Techniken

Die Ikonenbasierten Techniken erlauben in der Regel eine vollständige Präsentation der Datensätze. Sie ermöglichen die Visualisierung im Beobachtungsraum und verbinden multivariate Daten mit mehrdimensionalen Daten. Werte eines Datensatzes können kompakt und vergleichbar dargestellt werden, so dass Clusterbildungen und Gruppierungen auch ohne Filtermöglichkeit erkennbar werden. Ist die Datenmenge allerdings zu groß, treten Verdeckungen der Ikonen auf und die Identifikation von Einzelwerten ist schwierig oder wie bei Chernoff Faces nicht mehr möglich.

Die Ikonendarstellung ist eine recht spezifische Darstellungsart von Datenmerkmalen, so dass für eine effektive Nutzung eine genaue Einführung erforderlich sein kann.

Pixelbasierte Techniken

Pixelbasierte Techniken sind die kompakteste Darstellungsart vollständiger Datensätze. Alle Vertreter dieser Technik bieten Visualisierungen sehr großer Datenmengen an, allerdings sind sie sehr abstrakt in der Darstellung, da pro Datenwertrepräsentation nur genau ein Pixel verwendet wird. Das reduziert den Platzbedarf natürlich auf ein Minimum und erzeugt Übersichten über Datensätze, die ein intuitives Erkennen von Häufigkeiten und Verteilungen erlauben. Gleichzeitig ist die Einzelwertidentifikation oder der Vergleich von Werten aber schwierig, da Pixel für eine interaktive Auswahl zu klein sind.

Die kompakte Darstellung Pixelbasierter Techniken erlaubt im Gegensatz zu den anderen Techniken nicht nur eine Visualisierung von ca. 1.000 gleichzeitigen Datensätzen, sondern auf heutigen Darstellungsflächen eine Abbildung von sogar bis zu 1.000.000 Datensätzen.

Hierarchische Techniken

Die Hierarchischen Techniken unterstützen die Visualisierung unterschiedlicher Detailstufen und erlaubt durch diese Teilansichten das Erkennen von verschiedenen Trendverläufen auf unterschiedlichen Ebenen und Bereichen. Dafür ist eine Interaktion mit diesen

Techniken unumgänglich, um u.a. die richtige Hierarchisierung zu bestimmen. Da die primären Variablen in der ersten Hierarchieebene typischerweise eine höhere Auflösung haben und Korrelationen auf dieser Ebene einfacher erkannt werden, hat die Hierarchiestufenbestimmung einen direkten Einfluss auf die Effektivität der Visualisierung.

Tabellarische Übersicht der Bewertungen aller Visualisierungstechniken

Die Tabelle 4.2 gibt eine komplette Übersicht über die Bewertungen, wie gut welche Visualisierungstechniken die Anforderungen des Visualisierungswerkzeugs erfüllen. Das „+“ bedeutet dabei eine besonders gute Unterstützung, das „⊙“ eine durchschnittliche Eignung und das „-“ sagt aus, dass diese Technik diesen Punkt kaum bis gar nicht unterstützt. Die drei allgemeinen Anforderungen sind mit „*“ versehen, da sie für Visualisierungen vorausgesetzt sind. Eine Gewichtung dieser Anforderungen ist allerdings obsolet, da sie für Visualisierungen i.A. vorausgesetzt werden.

4.7 Zusammenfassung

Visualisierungen haben die impliziten Eigenschaften expressiv, effektiv und angemessen zu sein. Das heißt, sie sollten möglichst nur die Datenmenge visualisieren, die sie enthalten. Sie sollten die Darstellung so gestalten, dass u.a. den Anwendungszielen und den Eigenschaften des Ausgabemediums entsprochen wird. Und der Aufwand für die Visualisierung sollte in einem angemessenen Verhältnis zum gewonnenem Nutzen stehen.

Neben diesen vorgegebenen Zielen stellt das zu entwickelnde Visualisierungswerkzeug eine Reihe weiterer Anforderungen an Visualisierungen auf. Einige der Wichtigsten sind eine Übersicht über aktuelle Datenströme und -werte zu erhalten, historische Werteverläufe darstellen zu können und die Identifizierbarkeit und Auswählbarkeit von Einzelwerten zu ermöglichen.

Nachdem die Anforderungen an die Visualisierungen aufgestellt und in einem Anforderungskatalog zusammengefasst wurden, erfolgte im letzteren Teil dieses Kapitels die Vorstellung und Bewertung der verschiedenen Visualisierungstechniken für multivariate und multidimensionale Daten, die für eine Verwendung im Visualisierungswerkzeug in Frage kommen. Dabei wurden für jede Gruppe von Visualisierungstechniken einige Vertreter ausgewählt, anhand derer sie bewertet wurden. Die Ergebnisse der Einzelwertungen wurden gruppenweise den Visualisierungstechniken untergeordnet und abschließen in einer Gesamtbewertungsübersicht aufgelistet.

In diesem Kapitel wurde festgestellt, welche Anforderungen an das Visualisierungswerkzeug von welchen Visualisierungstechniken am besten erfüllt werden. Im nächsten Kapitel kann anschließend eine Auswertung dieser Ergebnisse stattfinden und darauf aufbauend eine prototypische Implementierung des Visualisierungswerkzeug zur EPL-Anfrageunterstützung angesetzt werden.

Typ	Anforderung	Geo- metrische Techniken	Ikonen- basierte Techniken	Pixel- basierte Techniken	Hier- archische Techniken
A01	Expressiv	*	*	*	*
A02	Effektiv	*	*	*	*
A03	Angemessen	*	*	*	*
V01	Übersicht über Datenquellen	⊖	+	-	-
V02	Übersicht aktueller Datenwerte	+	+	+	+
V03	Übersicht historischer Datenwerte	+	- + Color Icons	+	⊖
V04	Übersicht historischer Datensätze	+	-	+	-
V05	Ansicht verknüpfter Datenwerte	+	+	+	+
V06	Identifizierbarkeit und Auswählbarkeit von Einzelwerten	+	-	-	+
V07	Hervorheben von Datenwerten	+	-	-	⊖
V08	Zoom-Möglichkeiten	+	-	+	+
V09	Filtermöglichkeiten	+	⊖	-	-
V10	Zeit- und Mengenfenster anzeigen	+	-	-	-
V11	Wechsel zwischen Visualisierungsansichten	⊖	⊖	⊖	⊖
V12	Schnelle Visualisierung	+	+	⊖	⊖
V13	Aktualitätsanzeige möglich	⊖	⊖	⊖	⊖
V14	Überblick über Metainformationen	⊖	⊖	⊖	⊖
V15	Möglichkeit der Datenergänzung	⊖	⊖	⊖	⊖
V16	Rollenbasierte Benutzung ermöglichen	⊖	⊖	⊖	⊖
V17	Nachvollziehbare Visualisierung	⊖	⊖	-	⊖
V18	Rekonstruierbare Visualisierung	⊖	⊖	⊖	⊖
D01	Erkennung von Korrelationen	+	+	⊖	⊖ (+ Bei geeigneter Hierarchisierung)
D02	Erkennung von Clustern	+	⊖	+	⊖
D03	Erkennung von Häufigkeiten	- + Parahistogramme	⊖	+	+ Dimensional Stacking - Worlds-within-Worlds
D04	Erkennung durch Vergleiche	+	⊖	-	⊖
D05	Erkennung von Verteilungen	+	⊖	+	⊖
D06	Größe darstellbarer Datenmenge	+	+	+	⊖

Tabelle 4.2: Übersicht über die Bewertungen der Visualisierungstechniken hinsichtlich einer visuellen Unterstützung

5 Entwurf und Implementierung des Visualisierungswerkzeugs für die Anfrageunterstützung

Die Konzeption und die Implementierungsintention des Visualisierungswerkzeugs für die visuelle Unterstützung bei Esper-EPL-Anfragen sind Inhalt dieses Kapitels, welches aus drei Abschnitten besteht. Nachdem im vorherigen Kapitel die verschiedenen Visualisierungstechniken ausführlich bewertet wurden, werden im ersten Abschnitt dieses Kapitels die geeigneten visuellen Unterstützungspunkte bei der Esper-EPL-Anfragestellung ermittelt. Im zweiten Abschnitt wird dann basierend auf der Auswertung die Auswahl für den Entwurf des Visualisierungswerkzeugs vorgenommen. Der dritte und letzte Abschnitt befasst sich mit der Umsetzung des Werkzeugentwurfs.

5.1 Bestimmung der geeigneten visuellen Unterstützungen

Das Ergebnis der Untersuchungen im Kapitel 4 ist eine tabellarische Übersicht, die aufzeigt, welche Visualisierungstechniken die Anforderungen des Visualisierungswerkzeugs gut erfüllen und welche sich weniger gut dafür eignen (siehe Tabelle 4.2). Im Nachfolgenden wird eine Auswertung dieser Bewertungsergebnisse vorgenommen und die dazu passenden Unterstützungspunkte bei der Esper-EPL-Anfrageerstellung ermittelt. Anschließend findet die Zuordnung der nützlichen Visualisierungen zu den möglichen Unterstützungspunkten statt.

5.1.1 Auswertung der Bewertungsergebnisse

Die Bewertungen der Visualisierungstechniken hinsichtlich einer visuellen Unterstützung bei der EPL-Anfragestellung haben ergeben, dass sich vor allem Geometrische Visualisierungstechniken eignen. Im Detail haben die Bewertungen der verschiedenen Visualisierungstechniken folgendes ergeben:

Die Bewertungssymbole vor den Auswertungen sollen zusätzlich auf einen Blick das Ergebnis der Auswertung verdeutlichen. Die Symbole entsprechen den Bewertungssymbolen aus Tabelle 4.2, wobei die Größe der Symbole die Anzahl der erreichten Punkte mit dieser Wertung repräsentieren. Dabei wurde auch die Gewichtung der Anforderungen berücksichtigt.

⊕ ⊗ ⊖ **Geometrische Techniken** Die Visualisierungen aus dem Bereich der Geometrischen Techniken erfüllen nahezu alle Anforderungen aus dem Gesamtanforderungskatalog des Visualisierungswerkzeugs. So bieten alle vorgestellten Techniken durchgängig gute Übersichten auf die Daten, eine gute Identifizierbarkeit von Einzelwerten und unterstützen die Erkennung von Korrelationen sehr gut. Auch Cluster und Verteilungen lassen sich mit diesen Techniken gut wahrnehmen und Vergleiche von Datenwerten gut vornehmen. Die Menge an darstellbaren Daten ist groß und erfüllt somit sehr gut ein wichtiges Kriterium für das Visualisierungswerkzeug.

Insgesamt gibt es bis auf die Erkennung von Häufigkeiten keine Anforderung, die gegen die Benutzung von Geometrischen Techniken im Visualisierungswerkzeug spricht. Und selbst dabei gibt es mindestens eine Visualisierungsausnahme, die Parahistogramme, die auch dieses Kriterium gut erfüllen.

⊕ ⊗ ⊖ **Ikonenbasierte Techniken** Bei den Ikonenbasierten Techniken spricht genauso viel für wie gegen eine Verwendung im Visualisierungswerkzeug. Einerseits bieten sie eine gute Korrelationserkennung und unterstützen die Ansicht verknüpfter Datenwerte. Andererseits sind die Übersichten über ältere Datenwerte und Verläufe sowie die eindeutige Auswählbarkeit von einzelnen Datenwerten schlecht bis gar nicht möglich. Alle sonstigen Anforderungen werden lediglich nur durchschnittlich erfüllt.

Ikonenbasierte Techniken sind also für manche Untersuchungsvorhaben gut nutzbar, für andere weniger gut zu gebrauchen. Es hängt vom Anwendungsfall ab, ob eine Visualisierung mit Ikonen anderen Visualisierungsmöglichkeiten vorzuziehen ist.

⊕ ⊗ ⊖ **Pixelbasierte Techniken** Pixelbasierte Techniken eignen sich bedingt für das zu entwickelnde Visualisierungswerkzeug. Sie erfüllen viele Anforderungen aus dem Bereich der Datencharakteristika sehr gut, wie die Erkennung von Clustern, Häufigkeiten und Verteilungen. Auch bieten sie gute Übersichten über Datenwerte und historische Verläufe an und deutlich positiv hervorzuheben ist, dass sie als einzige Visualisierungstechniken besonders gut die Erkennung von Häufigkeiten unterstützen. Mit Pixelbasierten Techniken lassen sich bis zu eine Million Datensätze gleichzeitig visualisieren, diese Größe ist nur mit diesen Techniken möglich. Andererseits ist die Identifizierung und Auswahl von einzelnen Datenwerten hier nur schlecht möglich und eine Filterung damit nur wenig aussagekräftig. Pixelbasierte Techniken erfordern zudem oft aufgrund ihrer Rekursivität eine längere Berechnungszeit, so dass das Kriterium der schnellen Visualisierung nur mit befriedigend bewertet werden kann.

Zusammengefasst können diese Techniken nur für eine spezielle oder zusätzliche Verwendung in Betracht gezogen werden, wenn sich eine andere Visualisierung

nicht oder schlechter eignet.

- ⊕ ⊖ **Hierarchische Techniken** Die Auswertung der Hierarchischen Techniken fällt durchschnittlich aus. Diese Gruppe der Visualisierungstechniken bieten zwar interessante Arten der Darstellungen an, im Zusammenhang mit einer visuellen Unterstützung für EPL-Anfragen sind sie allerdings nicht hervorzuheben. Die hierarchischen Visualisierungen erhalten fast ausschließlich eine mittelmäßige Bewertung bezüglich der Anforderungen. Weder Übersichten über die Daten noch eine Erkennung von Datencharakteristika werden besonders gut unterstützt.

Hierarchische Techniken werden nicht bevorzugt, aber auch nicht sehr benachteiligt, um im geplanten Visualisierungswerkzeug Einsatz finden zu können. Insofern bieten sie sich als eine gute Alternative zu anderen Visualisierungen an, falls eine zusätzliche Darstellungsart gewünscht ist.

Geometrische Techniken sind also die erste Wahl für die Verwendung im Visualisierungswerkzeug. Da es sich dabei um eine Gruppe von Visualisierungstechniken handelt, sind einzelne Vertreter daraus noch auszuwählen. Auch wenn sich die anderen Gruppen von Visualisierungstechniken teilweise weniger gut eignen, können auch dort einige Vertreter der Visualisierungstechniken in Frage kommen. Anzumerken ist, dass die optionalen Anforderungen von fast allen Techniken durchweg durchschnittlich bewertet wurden und dort somit keine Begünstigungen zu finden sind. Eine Durchschnittsbewertung ist allerdings in diesem Kontext als positiv zu betrachten, da es bedeutet, dass alle Visualisierungstechniken beispielsweise die Einblendung und Anzeige von Metainformationen gleich gut unterstützen und dem nicht im Wege stehen.

5.1.2 Ermittlung der Unterstützungspunkte

Der Aufbau einer Esper-EPL-Anfrage ist im Listing 3.1 zu sehen. Anhand dessen werden im Folgendem die einzelnen Teile der Anfragestruktur genauer beschrieben und die Besonderheiten der Esper-EPL herausgestellt.

Um die möglichen Unterstützungspunkte in einer Esper-EPL-Anfrage aufzufinden, wird das Syntaxschema der Esper-EPL-Anfrage aus Kapitel 3.1.1 wieder aufgegriffen.

```
1 SELECT select_list
2 FROM stream_def [as name] [, stream_def [as name]] [,...]
3 [WHERE search_conditions]
4 [GROUP BY grouping_expression_list]
5 [HAVING grouping_search_conditions]
6 [OUTPUT output_specification]
7 [ORDER BY order_by_expression_list]
8 [LIMIT num_rows]
```

Listing 5.1: Syntaxschema einer Esper-EPL-Anfrage [Esp09a]

Mit der `SELECT`-Klausel in der ersten Zeile werden die Eigenschaften von Ereignissen und Ereignisse der Datenströme ausgewählt, die angezeigt werden sollen. Diese Auswahl könnte beispielsweise unterstützt werden durch eine Anzeige aller verfügbarer Eigenschaften von Ereignissen oder aller vorhandenen Ereignisse eines Datenstroms. Wichtig ist dabei vor allem die Fähigkeit, viele Informationen darzustellen, damit eine große Auswahlmöglichkeit visualisiert werden kann.

Die Zeile `FROM stream_def [as name] [, stream_def [as name]] [, ...]` spezifiziert den oder die Datenströme, die in einer Esper-EPL-Anfrage betrachtet werden sollen, und belegt ihn, wenn gewünscht, mit einem neuen Namen. Hier könnte eine visuelle Unterstützung durch eine z.B. geografische Anzeige aller existierenden Datenströmen erfolgen, so dass der Anwender auf einen Blick alle Quellen für Datenströme sofort erkennen kann..

In der `[WHERE search_conditions]`-Zeile werden die zuvor ausgewählten Datenströme mittels Konditionen genauer bestimmt. Es wird festgelegt, unter welchen Bedingungen Ereignisse aus den Datenströmen betrachtet werden.

Mit einer übersichtlichen Darstellung aller verfügbaren Merkmale kann die Esper-EPL-Anfrageformulierung hier erleichtert werden. Durch eine Übersicht können mögliche Bedingungen besser gesetzt werden, um die Anfrage auf die interessantesten Merkmale zu konzentrieren.

Die `GROUP BY`- und `HAVING`-Klauseln in der vierten und fünften Zeile aggregieren und gruppieren die Ergebnismenge der Ereignisse zu Gruppen mit besonderen Bedingungen. Auch hier bietet sich eine Übersicht über die Ereignisdaten an, die aber eine Filterung ermöglichen sollte. So könnte beispielsweise eine Filterung in der Visualisierungsansicht bereits eine Voransicht auf die Gruppen darstellen.

Das `OUTPUT output_specification`-Statement legt u.a. die Menge und Häufigkeit der Ergebnisausgabe einer Esper-EPL-Anfrage fest. Hier ist eine Unterstützung aus visueller Sicht nicht erforderlich.

Mit der `ORDER BY`-Klausel kann die Ausgabe der komplexen Ereignisse sortiert werden. Wenn gewünscht, wird eine Ergebnismenge so nach einem bestimmten Merkmal von Ereignissen sortiert ausgegeben. Hier können die gleichen Visualisierungen ähnlich wie bei der `GROUP BY`-Klausel in Zeile vier eine Unterstützung bei der Esper-EPL-Anfrage bieten.

Mit dem `LIMIT num_rows`-Statement in der letzten Zeile wird die Ergebnismenge auf ein bestimmtes Maximum beschränkt. Eine Visualisierung bringt hier keinen Mehrnutzen.

Eine Esper-EPL-Anfrage bietet neben den bisher genannten Unterstützungspunkten noch weitere bezüglich den EPL-Anfrage-Besonderheiten wie Zeitfenstern, Pattern und

Views.

Zum Beispiel könnte in einer Visualisierung zur Bestimmung von Zeitfenstern angegeben werden, dass nur für eine definierte Zeitspanne Ereignisse auf der Karte direkt angezeigt werden. So wird sofort erkannt, wie viele mögliche Ereignisse in einer Zeitspanne überhaupt auftreten.

Bei Pattern können visualisierte Übersichten verschiedener Datenmerkmale nützlich sein. Sie können das Erkennen der Merkmalsvielfalt für die Pattern-Verwendung erleichtern und bei der Verknüpfung von Zusammenhängen unterstützen.

Und auch Views können Vorteile aus den Visualisierungen der Ereignisdatenströme ziehen. Mittels Filterung und Markierungen in Übersichtskarten könnten etwa Längfenster direkt bestimmt werden. Aber auch sonst könnten andere der vielfältigen Views Nutzen aus Visualisierungen ziehen. Die Anzeige der Ereignisse bei einem Mausklick, also eine Momentaufnahme, könnte für die Definition von Views wie „Erstes Ereignis“ oder „Letztes Ereignis“ für eine Esper-EPL-Anfragestellung hilfreich sein.

Die Fülle an möglichen Unterstützungspunkte werden im nächsten Unterabschnitt wieder aufgegriffen und mit passenden Visualisierungen verknüpft.

5.1.3 Einordnung der passenden Visualisierungen

Nachdem die für eine Esper-EPL-Anfragestellung in Frage kommenden Visualisierungstechniken bestimmt und visuelle Unterstützungspunkte gefunden wurden, erfolgt hier eine sinnvolle Zuordnung zueinander.

Allgemein

Für die meisten der gefundenen Unterstützungspunkte sind gute Übersichten über die vorhandene Ereignisdaten und deren Eigenschaften die beste Unterstützung bei der Formulierung einer Esper-EPL-Anfrage. Dafür kommen die visualisierten Ansichten der *Scatterplot-Matrizen* und *Parallele Koordinaten* aus der Gruppe der **Geometrischen Techniken** in Frage. Sie bieten eine übersichtliche Darstellung an und erlauben dabei gleichzeitig eine Visualisierung von enorm großen Datenmengen.

Zeitfenster und Pattern

Speziell für die Definition von Zeitfenstern und Anzeige von Daten über die Zeit eignen sich vor allem *Spiral Graphs* von den **Geometrischen Techniken**. Auf Ihnen können auf der einen Seite gut Zeitintervalle bestimmt werden, auf der anderen Seite können mit ihnen Zeitverläufe von Ereignisdaten sehr gut angezeigt werden.

Aber auch für Pattern sind *Spiral Graphs* eine dankbare Visualisierung. Anhand ihrer können sehr gut Pattern-Operatoren wie `EVERY` oder `EVERY-DISTINCT` (siehe 3.1.1.3)

definiert werden, die sich beispielsweise an sich wiederholenden Mustern oder periodisch auftretenden Ereignissen oder Merkmalen orientieren.

SELECT- und WHERE-Klausel

Für die `SELECT`- und `WHERE`-Klausel können auch noch *Bubble Graphs* Visualisierungen der *Geometrischen Techniken* herangezogen werden. Sie bieten auf geografisch basierten Karten eine gute grobe Übersicht auf Ereignisdaten oder Ereignisdatenströme.

Gesamtübersicht

Um eine gute Gesamtübersicht vieler Merkmale zu erhalten und möglich Unterschiede oder Veränderungen von Ereignissen oder Ereignisdaten über die Zeit angezeigt zu bekommen, sind noch die *Stick-Figures*-Visualisierungen der *Ikonenbasierten Techniken* zu erwähnen. Sie bieten zwar mit Zunahme der dargestellten Datenmenge eine immer schwindendere Unterscheidungsmöglichkeit der Einzelmerkmale und Einzelereignisse, jedoch sind mit dieser Technik sehr große Datenmengen und Datensätze kompakt visualisierbar.

Alternativen

Neben den bereits vorgestellten Visualisierungstechniken gibt es zu manchen Unterstützungspunkten noch ein, zwei alternative, aber nicht unbedingt gleichwertige Visualisierungsmöglichkeiten.

Als eine Alternative für die *Parallelen Koordinaten* und gleichzeitige Erweiterungsmöglichkeit sind die *Parahistogramme* zu erwähnen. Da sie die Schwäche ihrer Gruppe von *Geometrischen Techniken* aus hebt, indem sie statistische Häufigkeiten mit in ihre Darstellung von Datenwerten aufnimmt. Dies macht die Parahistogramme allerdings auch sehr speziell. Durch ihre Integration von Häufigkeitsverteilungen in ihrer Visualisierung schwindet die Anzahl an gleichzeitig darstellbaren Daten und Wertespektren sehr stark.

Color Icons aus den *Ikonenbasierten Techniken* eignen sich bedingt gut als Alternative für die Anzeige von zeitlichen Verläufen. Einerseits können theoretisch mehrere gekoppelte Zeitverläufe verschiedener Datenmengen gemeinsam visualisiert werden. Die Schwierigkeit dabei besteht darin, die richtigen Farbkodierungen für eine gute Auswertung zu bestimmen und zudem, das Wissen zu besitzen diese Auswertung vornehmen zu können.

Für die Bestimmung von Alternativen ist bei der Gruppe der *Pixelbasierten Techniken* anzumerken, dass Pixel die kleinste Einheit auf dem Bildschirm darstellen und so eine Einzelidentifikation von Werten in der Regel ausgeschlossen erscheint. Selbst bei wenigen Werten wird die Interpretation der Visualisierung tendenziell nicht einfacher.

Einzig bei riesig großen Datenmengen mit globalen Veränderungen sind **Pixelbasierte Techniken** wie etwa **Recursive-Pattern-Techniken** möglicherweise als Alternative verwendbar.

Die Ausführungen hier sind nicht abschließend, da es eine Reihe weiterer Vertreter für jede Gruppe von Visualisierungstechniken gibt, deren Vorstellung den Umfang dieser Arbeit überschreitet.

5.2 Auswahl für den Entwurf des Visualisierungswerkzeugs

Im vorherigen Abschnitt wurde eine Reihe möglicher visueller Unterstützungen für eine Esper-EPL-Anfragestellung gefunden. Nachfolgend findet daraus eine sinnvolle Auswahl statt, die im Visualisierungswerkzeug prototypisch implementiert werden soll. Danach wird ein kurzer Überblick über bereits vorhandene Visualisierungswerkzeuge und -projekte gegeben, der Aufschluss darüber geben soll, welche Implementationsbasis für die Esper-EPL-Anfrageunterstützung verwendet werden soll.

5.2.1 Wahl der zu entwerfenden visuellen Unterstützungen

Auch wenn alle Visualisierungsarten bezogen auf einen jeweiligen Anwendungsfall eine sinnvolle Unterstützung bieten könnte, beschränkt sich der zu entwickelnde Prototyp lediglich auf einige wenige, allgemein anwendbare Visualisierungen als „*Proof of Concept*“.

Die Evaluierung der möglichen Visualisierungen für einen unterstützenden Ansatz bei der Esper-EPL-Anfragestellung hat ergeben, dass sich wenige Visualisierungen für viele Unterstützungsmöglichkeiten anbieten. Deshalb werden in der prototypischen Implementierung alle sinnvollen Visualisierungen eingebaut, die vorgeschlagen wurden.

Im Detail bedeutet dies, dass die vier bis acht geeignetsten Visualisierungsarten für die prototypische Implementierung in Frage kommen. Diese Visualisierungstechniken lauten wie folgt:

Aus der Gruppe der *Geometrischen Techniken* sind es die

- *Scatterplot-Matrizen*
- *Spiral Graphs*
- *Parallele Koordinaten* und
- *Parahistogramme*

Aus der Gruppe der *Ikonenbasierten Techniken* sind es die

- *Bubble Graphs* und

- *Stick Figures*

Aus der Gruppe der *Hierarchischen Techniken* sind es die

- *Cone Trees*

Von den *Pixelbasierten Techniken* wurden keine Vertreter für eine prototypische Umsetzung favorisiert.

5.2.2 Überblick vorhandener Visualisierungswerkzeuge und -projekte

Es gibt bereits eine große Anzahl an reinen Visualisierungswerkzeugen für allgemeine Zwecke, die unter verschiedenen Lizenzen und Technologien verfügbar sind. Ein Teil davon wurde in kleinen Gruppen und Projekten an Universitäten entwickelt und steht somit größtenteils zur freien Verfügung. Nachteilig dabei ist, dass universitäre Projekte aufgrund der Abgänge von Absolventen nach einer gewissen Zeit eingestellt oder nicht mehr weiterentwickelt werden. Einerseits soll dies kein Hindernis für die Verwendung in dieser Arbeit sein, da allein die Funktionsfähigkeit und Mächtigkeit des Visualisierungswerkzeugs von entscheidender Bedeutung ist. Andererseits schränkt es die Auswahl etwas ein, da beendete Visualisierungsprojekte bei Problemen oder Fragen keinen oder nur noch mangelhaften Support anbieten.

In dieser Arbeit wird daher nur eine Auswahl an frei verfügbaren Visualisierungswerkzeugen kurz vorgestellt, die den eben genannten Anforderungen genügen, um als Basis für diese prototypische Implementation zu dienen.

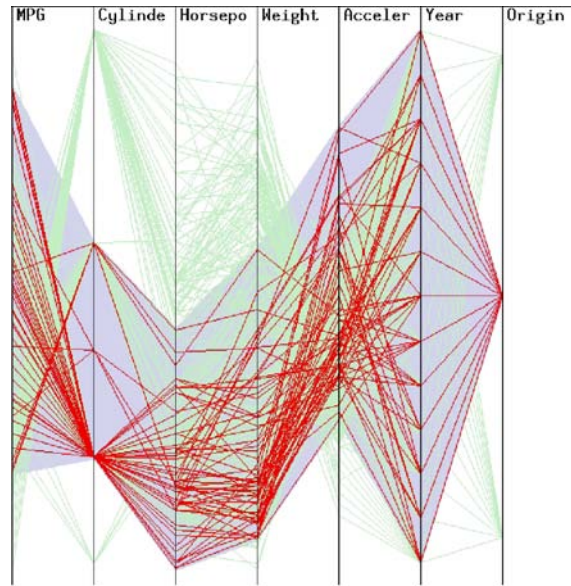
Die Kurzvorstellung der existierenden Visualisierungswerkzeuge erfolgt anhand von den folgenden fünf Merkmalen in Tabellenform:

- **Angebotene Visualisierungen**
 - **Interaktionen**
 - **Programmiersprache**
 - **Lizenz**
 - **Letzte Aktualisierung**
-

Merkmal	GGobi [Ggo08]
Angebotene Visualisierungen	Scatterplot, Scatterplot-Matrizen, Parallele Koordinaten, Time Series, Bar-charts, Average Shifted Histogramm, Dotplot, Balkendiagramm, Spineplot, Grand tour, Projection Pursuit
Interaktionen	Diverse (u.a. Skalieren, Färben, Identifizieren, Transformieren, Manipulieren, Markieren)
Programmiersprache	GTK 3.0
Lizenz	GNU Public License
Letzte Aktualisierung	12 August 2008

Merkmal	XmdvTool [The]
Angebotene Visualisierungen	Scatterplot, Scatterplot-Matrizen, Star Glyphs, Parallele Koordinaten, Dimensional Stacking, Pixel-oriented Display
Interaktionen	Diverse (u.a. N-Dimensionales Einfärben, Hierarchisches Clustern, Filterung)
Programmiersprache	OpenGL und Tcl/Tk
Lizenz	Open Source
Letzte Aktualisierung	20. Oktober 2010

Merkmal	Walrus - Graph Visualization Tool [Cai]
Angebotene Visualisierungen	Graphen im 3-dimensionalem Raum
Interaktionen	Diverse (u.a. Auswählen, Markieren, Färben, Zoomen)
Programmiersprache	Java
Lizenz	GNU Lesser GPL
Letzte Aktualisierung	November 2002

Abbildung 5.1: Visualisierungsbeispiel vom Visualisierungswerkzeug *XmdvTool* [The]

Merkmal	The InfoVis Toolkit [Jea]
Angebotene Visualisierungen	Scatterplot, Time Series, Parallel Koordinaten, Tabellen-Matrizen; Node-Link diagrams, Icicle trees, Treemaps; Adjacency Matrices, Node-Link diagrams und weitere Variationen
Interaktionen	Diverse (u.a. Zoom, Markieren, Färben, Verschieben)
Programmiersprache	Java
Lizenz	Open Source
Letzte Aktualisierung	30 November 2005

Neben den hier vorgestellten Visualisierungswerkzeugen gib es viele weitere, deren Auflistung den Rahmen dieser Arbeit übersteigt. Bei einem Großteil davon handelt es sich allerdings um kommerzielle Produkte. Für eine ausführlichere Vorstellung und den Vergleich von Visualisierungswerkzeugen wird auf die Arbeit [Hel07] verwiesen. Eine Reihe weiterer Visualisierungswerkzeuge sind direkt über die Dokumente [Pot08] und [Pre07] zu finden.

5.2.3 Grundlage für die Entwicklung des Visualisierungswerkzeug

Als Grundlage für die Entwicklung des Visualisierungswerkzeugs werden die frei verfügbaren Ggobi und XmdvTool gewählt. Zum einen ist durch die Java-Programmierung des Werkzeugs die Erweiterbarkeit zugesichert. Zum anderen bietet diese die meisten gewünschten Visualisierungen an.

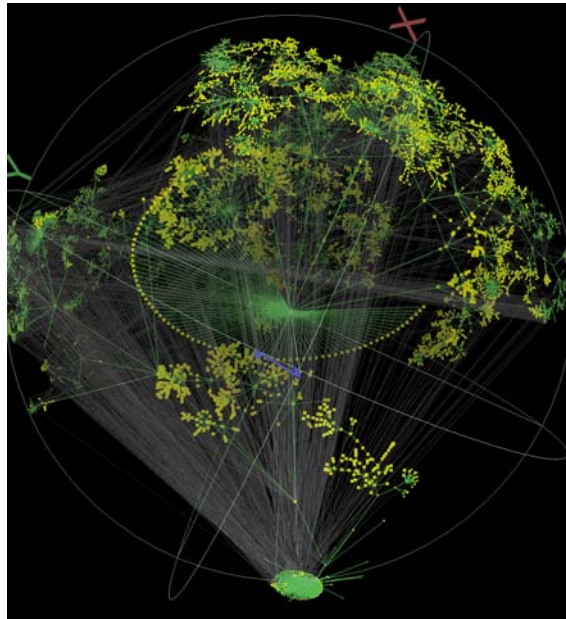


Abbildung 5.2: Visualisierungsbeispiel vom Visualisierungswerkzeug *Walrus - Graph Visualization Tool* [Cai]

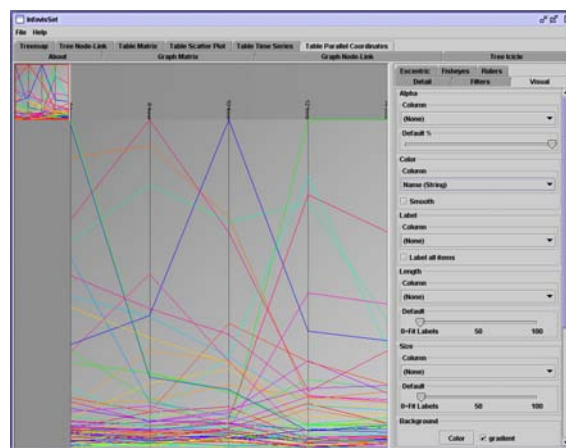


Abbildung 5.3: Visualisierungsbeispiel vom Visualisierungswerkzeug *The InfoVis Toolkit* [Jea]

5.3 Prototypische Umsetzung

Nach der Ausarbeitung der Übersicht über die Möglichkeiten einer visuellen Unterstützung bei der EPL-Anfragestellung folgt nun mit der prototypischen Umsetzung des Werkzeugs das letzte Teilziel dieser Arbeit.

In den folgenden beiden Unterabschnitten wird zunächst die Architektur des Werkzeugs aufgezeigt und anschließend *Mock-ups* des Visualisierungswerkzeugs gezeigt und erläutert. Abschließend wird auf den prototypischen Implementationsansatz für das Visualisierungswerkzeug eingegangen.

5.3.1 Architekturbild

Im Architekturbild Abbildung 5.4 ist unten links der Anwender als Strichfigur angedeutet. Darüber befindet sich das Visualisierungswerkzeug, das verschiedene Visualisierungen anbietet. Der Anwender kann direkt mit dem Visualisierungswerkzeug interagieren. Dies soll mittels des Doppelpfeils (hin und zurück) symbolisiert werden.

Bei einer Anfrage des Anwenders an das Visualisierungswerkzeug eine Visualisierung von Sensordatenströmen anzuzeigen, startet das Werkzeug eine Anfrage an die Middleware. Die Middleware, das CEP-System, ist dabei für das Visualisierungswerkzeug eine Art Vermittler und Zwischenspeicher für Sensordaten zugleich [Bad09].

Die Middleware bekommt kontinuierlich die Datenwerte von den Sensordatenströmen übermittelt. Sobald sie die durch den Anwender ausgelöste Anfrage vom Visualisierungswerkzeug erhalten hat, filtert es die gewünschten Datenströme aus, holt gegebenenfalls zusätzliche historische Daten aus ihrem Langzeitspeicher und übermittelt die Datenwerte zurück an das Visualisierungswerkzeug. Dies geschieht von nun an genau solange, bis die Anfrage aus der Middleware wieder entfernt wird.

Das Visualisierungswerkzeug empfängt die Datenwerte und gibt die Visualisierung entsprechend der Anwenderaufforderung aus.

5.3.2 Mock-ups

Im Folgenden wird anhand von ersten Demonstrations-Prototypen, sogenannten *Mock-Ups*¹ die zu entwickelnde Bedienoberfläche des Visualisierungswerkzeugs und deren Elemente skizziert.

5.3.2.1 Mock-up der Programmoberfläche

Das Mock-up in Abbildung 5.5 zeigt, wie die Programmoberfläche des Visualisierungswerkzeugs aussehen soll. Es teilt sich dabei in einen oberen und in einen unteren Bereich ein.

¹„Ein Mock-up in der Softwareentwicklung bezeichnet einen rudimentären Wegwerfprototyp der Benutzerschnittstelle einer zu erstellenden Software. Mock-ups werden ... eingesetzt, um Anforderungen an die Benutzeroberfläche ... besser ermitteln zu können.“ [WIK10c]

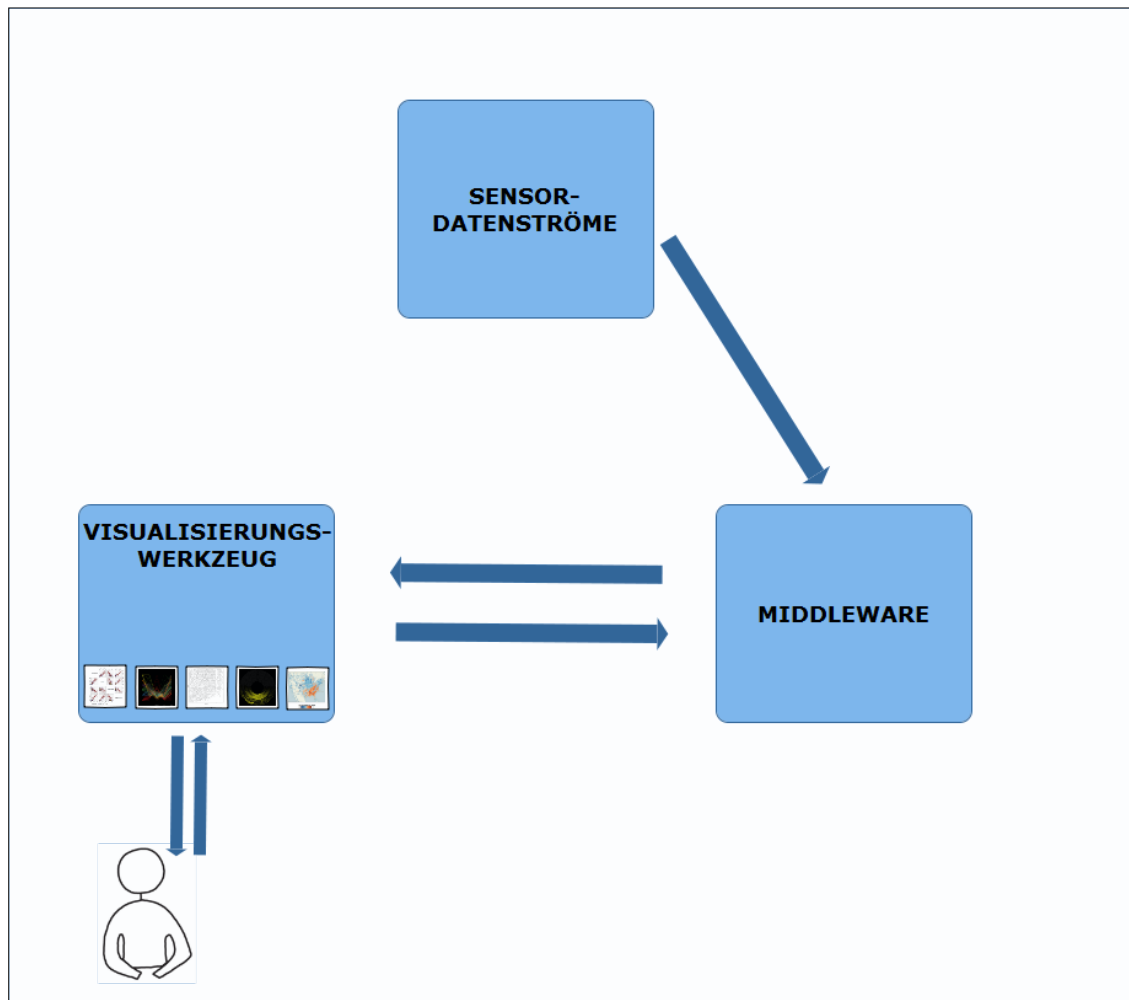


Abbildung 5.4: Architektur des Visualisierungswerkzeugs (Eigene Darstellung)

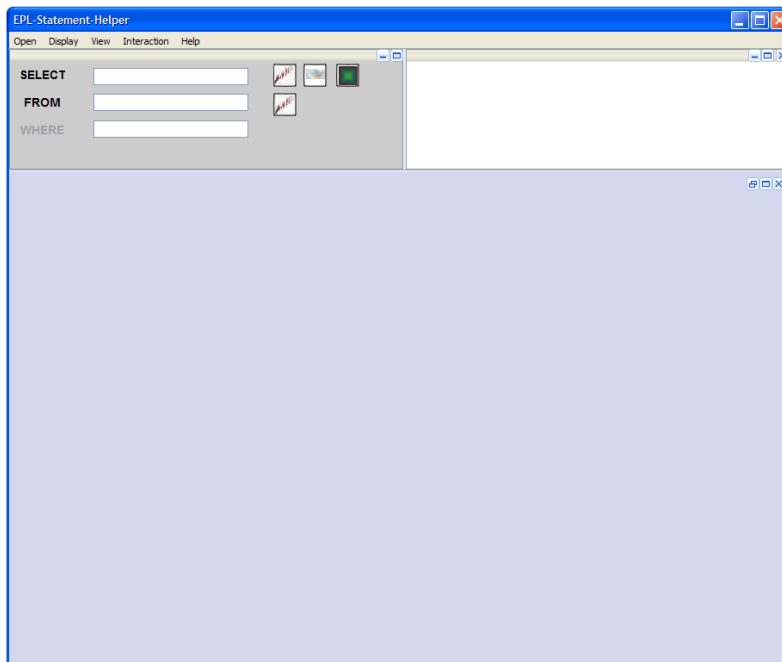


Abbildung 5.5: Mock-up der Programmoberfläche (Eigene Darstellung)

Im oberen Bereich werden links die nutzbaren Klauseln angezeigt. Nicht oder noch nicht auswählbare Klauseln sind grau hinterlegt oder werden durch die untere Fläche verdeckt. Die Eingabefelder, um die Esper-EPL-Anfrage aufzufüllen befinden sich rechts davon, gefolgt von Ikonen, die die verschiedenen möglichen Visualisierungen für den nebenstehenden Klauselbereich symbolisieren. Ganz rechts werden Systeminformationen zu den ausgewählten Visualisierungen und Metainformationen angezeigt. Im Mock-up sind diese Inhalte dort nur mit Blindtexten versehen.

Bei dem unterem Bereich handelt es sich genauso wie bei den beiden Fenstern oben um verschiebbare Fenster (*Internal Frames*), welche dennoch fest im Programmfenster integriert sind. Je nach Situation verschiebt sich dieser Bereich nach oben oder nach unten oder kann vom Anwender frei innerhalb der Anwendung verschoben werden. In diesem Bereich werden die verschiedenen Visualisierungen aus der obigen Auswahl angezeigt.

5.3.2.2 Mock-ups einer Benutzerinteraktion

Mit der nachfolgenden kleinen Serie von Mock-ups soll eine Benutzerinteraktion skizzenhaft angedeutet werden.

Das Mock-up 1 in Abbildung 5.6 zeigt die Auswahl einer Visualisierung. Der Anwender hat hierbei bereits die erste Visualisierungsart durch das Anklicken der ersten Ikone in der SELECT-Klausel ausgewählt. Diese Ikone stellt die Visualisierung als Scatterplot-Matrizen dar, was durch eine Miniaturabbildung auf der Ikone angedeutet wird.

Die nun aktive Ikone ist farblich markiert worden und das untere Programmfenster hat sich bis zur aktuellen EPL-Anfragezeile hoch geschoben, um den Platz für die Visua-

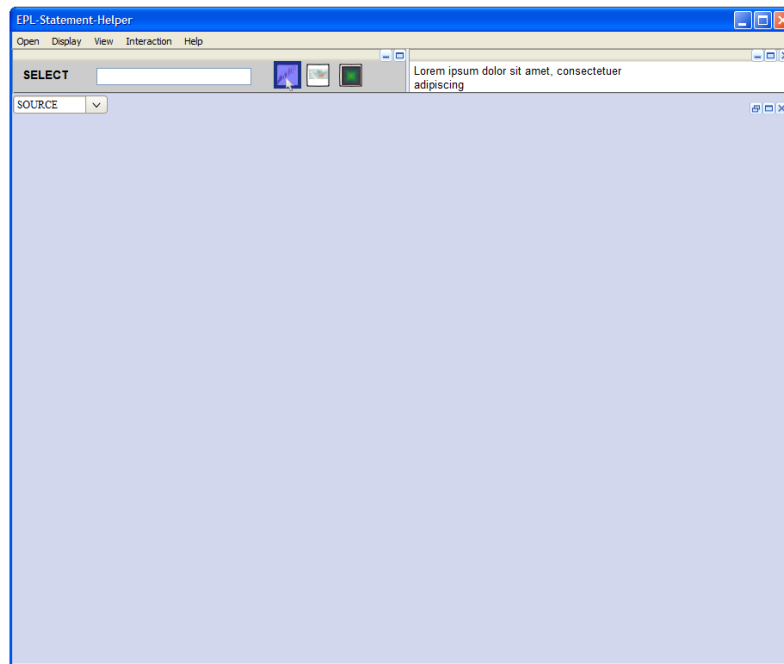


Abbildung 5.6: Mock-up 1 der Benutzerinteraktion: Auswahl einer Visualisierung (Eigene Darstellung)

lisierung darunter zu maximieren. Im unteren Fenster erscheint zudem ein Auswahlfeld, in dem eine Quelle für die Visualisierung ausgewählt wird.

Mock up 2 in Abbildung 5.7 zeigt nun die ausgewählte Ansicht der Scatterplot-Matrizen. Der Anwender befindet sich darin gerade mit dem Mauszeiger bzw. mit dem Finger auf dem Touchscreen über dem oberen mittleren Scatterplot rechts.

Mock-up 3 in Abbildung 5.8 zeigt abschließend die Auswahl von Daten innerhalb der Visualisierung. Durch die Handabbildung wird die mögliche interaktive Bedienung mittels eines Datenhandschuhs oder der Finger-Bedienung auf einem großen Touchscreen-Display angedeutet.

5.3.2.3 Mock-up einer weiteren Visualisierung

In diesem Mock-up wird eine weitere ausgewählte Visualisierungsansicht noch kurz skizziert. Die Abbildung 5.9 zeigt die Ansicht einer ausgewählten Bubble Graphs-Visualisierung. Das Mock-up soll andeuten, dass zwischen verschiedenen Visualisierungen hin und her gewechselt werden kann.

5.3.3 Prototypischer Implementationsansatz

Die Algorithmen für Visualisierungen, wie sie im geplanten Visualisierungswerkzeug Verwendung finden sollen, stellen sich unter anderem als sehr komplex dar, so dass sie für diese Arbeit nicht erneut programmiert werden. Die Implementierungsintention ist so gelegt, dass für das Visualisierungswerkzeug Anleihen bei anderen Visualisierungs-

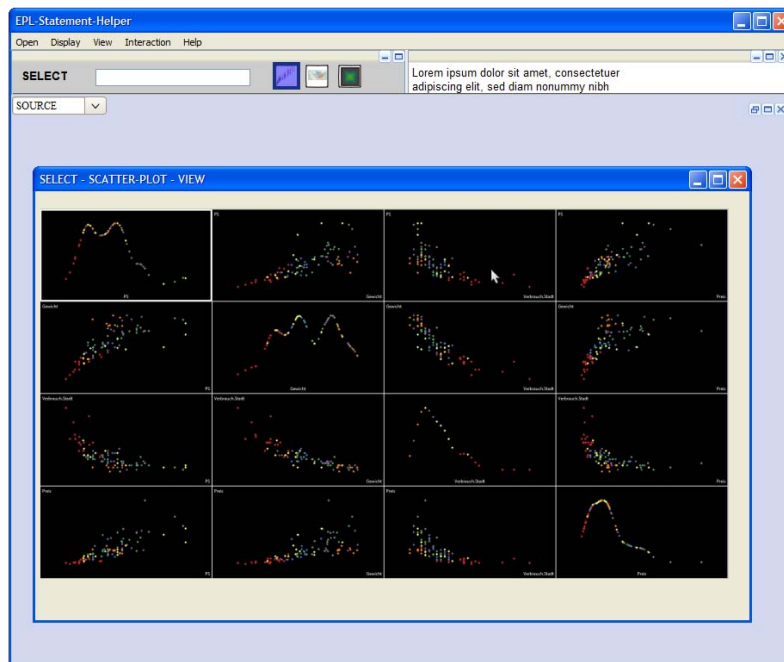


Abbildung 5.7: Mock-up 2 der Benutzerinteraktion: Anzeige einer Visualisierung (Eigene Darstellung)

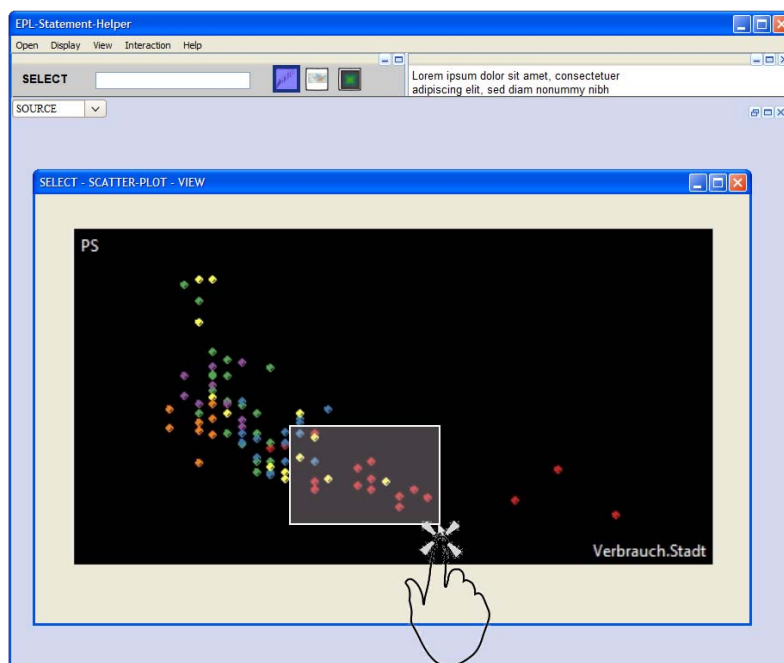


Abbildung 5.8: Mock-up 3 der Benutzerinteraktion: Auswahl von Daten in einer Visualisierung (Eigene Darstellung)

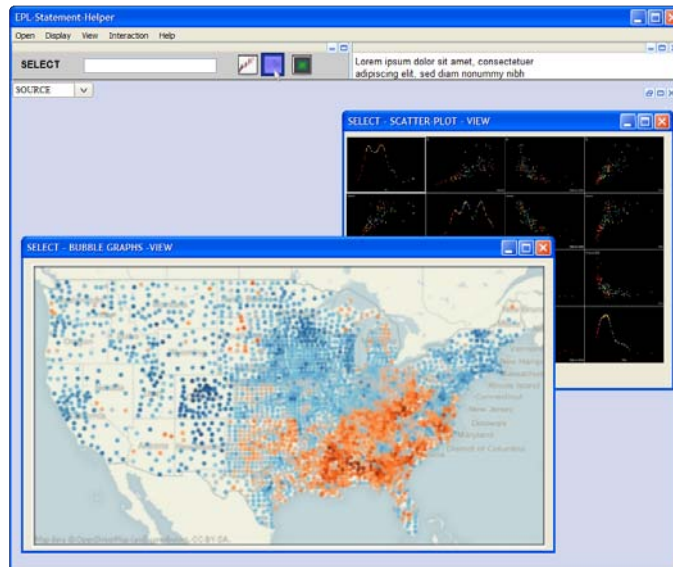


Abbildung 5.9: Mock-up zweier überlappender Visualisierungen (Eigene Darstellung)

projekten gemacht werden. So kann Bestehendes genutzt werden und an das Visualisierungswerkzeug adaptiert und erweitert werden.

Das Werkzeug wurde schließlich als eigenständige Applikation in der Programmiersprache Java geschrieben. Die bereits im Kapitel 5.2.3 für die Implementierung ausgewählten Visualisierungsprojekte GGobi und XmdvTool stellen - mit geringen Modifikationen - eine geeignete Basis für die reine Datenvisualisierung dar. Die von diesen Projekten gebotenen Visualisierungen finden sich als *Internal Frames* (siehe Mock-up im Kapitel 5.3.2.1) in der Applikation wieder.

Um an die zu visualisierenden Sensordatenströme zu gelangen, verbindet sich der Prototyp über das Internet mit einer laufenden Instanz der *JESPA-Middleware* (*Jadex Event Stream Processing Architecture*) [BL10], einer Infrastruktur zur Vermittlung von Kontextdaten zwischen Daten-Produzenten (Sensoren) und -Konsumenten (zum Beispiel dem Visualisierungswerkzeug) und ruft die Metadaten aller in JESPA eingehenden Sensordatenströme ab (siehe Kapitel 5.3.1).

Zur Visualisierung eines vom Benutzer ausgewählten Datenstroms verbindet sich das Werkzeug erneut mit JESPA um einerseits die von JESPA aufgezeichneten historischen Zeitreihen des Datenstroms abzurufen und andererseits um sich ggf. für Notifikationen bei neu eintreffenden Sensor-Datensätzen zu registrieren, was eine fortlaufende Aktualisierung der Visualisierung erlaubt.

Nachdem der Benutzer, unterstützt durch die Visualisierungen, eine EPL-Anfrage formuliert hat, verbindet er sich erneut mit der JESPA-Middleware um nun seine Anfrage beim System zu registrieren.

5.4 Zusammenfassung

In diesem Kapitel wurde eine Auswertung aller Bewertungen der Visualisierungstechniken vorgenommen, die für eine Implementierung im Visualisierungswerkzeug in Betracht kommen. Das Ergebnis der Untersuchungen ist, dass die Visualisierungstechniken aus der Gruppe der Geometrischen Techniken sich am besten für eine visuelle Unterstützung bei der EPL-Anfragestellung eignen.

Mit diesem Resultat wurden die möglichen Unterstützungspunkte bei der EPL-Anfragestellung evaluiert und mit den geeigneten Visualisierungen verknüpft. Dabei zeigte sich, dass sich einige Visualisierungen gleich für mehrere Unterstützungspunkte am besten anbieten. Geometrische Techniken wie etwa Scatterplot-Matrizen können gleichzeitig als helfende Unterstützung für die `SELECT`-, `FROM`- und `WHERE`-Klausel nützlich sein.

Das Kapitel schließt ab mit dem Abschnitt der prototypischen Umsetzung des Visualisierungswerkzeugs. Darin wurde die Architektur des Visualisierungswerkzeugs vorgestellt und anhand von Demonstrations-Prototypen die Bedienoberfläche des Programms sowie Benutzerinteraktionen skizziert.

Die Bestimmung und der Entwurf des Visualisierungswerkzeugs sowie ein prototypischer Implementationsansatz haben gezeigt, dass Visualisierungen bei der EPL-Anfragestellung eine Hilfe sind. Eine Gesamtzusammenfassung und einen kurzen Ausblick, welche weiteren Unterstützungen noch möglich sind, liefert das letzte Kapitel dieser Arbeit.

6 Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurden die Möglichkeiten einer visuellen Unterstützung bei der EPL-Anfragestellung auf Sensordatenströme untersucht. Im folgenden wird eine Zusammenfassung und ein Ausblick dieser Arbeit gegeben, der interessante Ansatzpunkte für einer weitergehende Evaluierung enthält.

6.1 Zusammenfassung

Ausgangspunkt für diese Arbeit war die EPL-Anfragestellung auf Sensordatenströme mittels einer geeigneten Visualisierung zu unterstützen. Dieses Vorhaben wurde dadurch motiviert, dass die Verarbeitung von multivariaten Sensordatenströmen in der Regel einen recht komplexen Vorgang darstellen und deshalb die Formulierung einer Anfrage auf solche Datenströme möglichst so einfach und intuitiv wie möglich gehalten werden sollte.

Als Basis für dieses Vorhaben wurde nach der Einleitung im zweiten Kapitel in die Grundlagen der komplexen Ereignisverarbeitung und Visualisierung eingeführt. Es folgte darauf die Einarbeitung in das beispielhafte Complex Event Processing-System Esper, welches in dieser Arbeit stellvertretend für eine Reihe ähnlicher CEP-Systeme ausgewählt wurde.

Die Ausführungen zu Esper haben gezeigt, dass die verwendete Esper-Event Processing Language zusammen mit der Esper-Engine eine sehr mächtige Anfragesprache darstellt. Sie bietet die gewohnten SQL-Klauseln als Basis für ihre eigene Anfragesprache, erweitert diese aber um grundlegende Mengen- und Zeitaspekte wie Zeit- und Mengenfenster. Hinzu kommen auch noch Views, die Sichten auf Datenströme erlauben, so dass Anfragen darauf ähnlich wie in Datenbanken gestellt werden können. Den Abschluss der Erweiterungen bilden u.a. noch die Pattern, die das Definieren von zeitlichen bzw. kausalen Zusammenhängen zwischen Ereignisdaten erlauben.

Im Anschluss sind anhand von Anwendungsszenarien mögliche Anwendungsideen für EPL-Anfragen auf Sensordatenströme vorgestellt worden. Mit deren Hilfe wurden daraufhin die Problematik und die Schwierigkeiten bei der EPL-Anfragestellung aufgezeigt. Es hat sich herausgestellt, dass besonders der Mangel an Übersichten zu den Sensordatenströmen und das Fehlen von Anzeigen zeitlicher Verläufe der Datenwerte, ein Problem für die EPL-Anfragestellung darstellen.

Dies wurde in den darauf folgenden Kapiteln näher behandelt, indem evaluiert wurde, mit welchen Visualisierungstechniken welche unterstützenden Ansätze bei der EPL-

Anfragestellung am besten zusammenpassen.

Dafür wurden zunächst die grundsätzlichen Anforderungen an Visualisierungen vorgestellt und anschließend die visualisierungswerkzeug-spezifischen Anforderungen an die zur Verwendung in Frage kommenden Visualisierungstechniken aufgestellt. Das Ergebnis daraus ist ein Bewertungskatalog, anhand derer die Bewertungen für die Vertreter der einzelnen Visualisierungstechniken vorgenommen wurden.

Mit der Auswertung der Bewertungsergebnisse wurde im nächsten Kapitel begonnen. In diesem hat sich gezeigt, dass einige wenige Visualisierungstechniken, aber vor allem die der Gruppe der Geometrischen Techniken, sich einerseits einen am besten, aber gleichzeitig auch für mehrere Unterstützungspunkte bei der EPL-Anfragestellung eignen.

Nachdem passende Unterstützungspunkte für die Visualisierungen gefunden wurden, erfolgte die Auswahl für die Basis des Visualisierungswerkzeugs. Dies wurde so realisiert, dass ein Überblick über verschiedene bereits vorhandene Visualisierungswerkzeuge und -projekte ausgearbeitet und vorgestellt wurde. Daraufhin wurde aus der Liste an Visualisierungswerkzeugen eine Auswahl als mögliche Grundlage für die Implementierung bestimmt.

Abgeschlossen wurde das Vorhaben, eine visuelle Unterstützung für die EPL-Anfragestellung an CEP-Systeme zu entwickeln, mit der prototypischen Umsetzung eines Entwurfs des Visualisierungswerkzeugs.

Die Realisierung fand in der Form einer Vorstellung des Architekturbildes des Visualisierungswerkzeug und der Präsentation der Programm- und Bedienoberfläche des Visualisierungswerkzeugs anhand von Demonstrations-Prototypen, sogenannten Mockups statt. Es wurde angedeutet, wie sich das Visualisierungswerkzeug bedienen lässt und auf Interaktionen reagiert.

Zusammenfassend lässt sich feststellen, dass eine Visualisierung bei der EPL-Anfrageformulierung eine große Unterstützung darstellt und das Finden von sinnvollen EPL-Anfragen deutlich erleichtert.

6.2 Ausblick

In dieser Arbeit wurden eine Vielzahl an Visualisierungstechniken auf einen unterstützenden Ansatz für die EPL-Anfragestellung untersucht. Zum einen kann eine solche Untersuchung nicht abschließend sein. Von daher gibt es viele weitere Visualisierungstechniken, deren Unterstützungswert noch evaluiert werden könnte.

Zum anderen wurden bisher ausschließlich die Eingaben in eine EPL-Anfrage für eine mögliche Unterstützung untersucht. Denkbar wäre es hier, auch die zurückkommende

EPL-Anfrageantwort, also die komplexen Ereignisse zu visualisieren, um mit deren Hilfe die semantische Korrektheit der gestellten EPL-Anfrage zu validieren.

Insgesamt bildet die Anzahl an fortlaufend neu und weiter entwickelten Visualisierungstechniken immer wieder neue Möglichkeiten dieses Gebiet der Complex Event Processing erneut zu evaluieren.

Literaturverzeichnis

- [ABW03] ARASU, Arvind ; BABU, Shivnath ; WIDOM, Jennifer: CQL: A Language for Continuous Queries over Streams and Relations. In: *DBPL*, 2003, 1–19
- [AKK96] ANKERST, Mihael ; KEIM, Daniel A. ; KRIEGEL, Hans-Peter: Circle Segments: A Technique for Visually Exploring Large Multidimensional Data Sets, 1996 (Proc. Visualization'96)
- [Bad09] BADE, Dirk: Verteilte Abfragebearbeitung von Sensordaten. In: 8. *GI/ITG KuVS Fachgespräch Sensornetze, Hamburg, Germany* (2009), 8. http://www.ti5.tu-harburg.de/events/fgsn09/proceedings/fgsn_013.pdf
- [Bed90] BEDDOW, J.: Shape coding of multidimensional data on a microcomputer display. In: *Visualization, 1990. Visualization '90., Proceedings of the First IEEE Conference on*, 1990, S. 238–246, 478
- [BGH⁺00] BRULS, Mark ; GEERLINGS, Johan ; HAM, Frank van ; HUIZING, Kees ; MELBY, Elisabeth ; WETERING, Huub van d. ; WIJK, Jarke J.: *SequoiaView - Exhibit your hard drive!* http://w3.win.tue.nl/nl/onderzoek/onderzoek_informatica/visualization/sequoiaview//. Version: 11 2000
- [BL10] BADE, Dirk ; LAMERSDORF, Winfried: An Agent-based Event Processing Middleware for Sensor Networks and RFID Systems. In: *Computer Journal, Special Issue on Agent Technologies for Sensor Networks*" (2010). <http://dx.doi.org/10.1093/comjnl/bxp103>. – DOI 10.1093/comjnl/bxp103
- [BRA10] BRANDYANDKATHY: *So What's a Chernoff Face, Anyway?* <http://bradandkathy.com/software/faces.html>. Version: 2010
- [Bro99] BROCKHAUS ; 1 (Hrsg.): *Reader's Digest - DER BROCKHAUS - IN ZWEI BÄNDEN*. F.A. Brockhaus GmbH, 1999
- [Cai] CAIDA.ORG: *Walrus - Graph Visualization Tool*. <http://www.caida.org/tools/visualization/walrus/>
- [Che73] CHERNOFF, Herman: The Use of Faces to Represent Points in K-Dimensional Space Graphically. In: *Journal of the American Statistical Association* 68 (1973), Nr. 342, 361–368. <http://dx.doi.org/10.2307/2284077>. – DOI 10.2307/2284077. – ISSN 01621459
-

- [Che99] CHEN, C.: *Information Visualisation and Virtual Environments*. Springer-Verlag, 1999. – 223 S.
- [Cis01] CISCO SYSTEMS GMBH: *Handbuch Netzwerktechnologien*. Markt-und-Technik-Verl., 2001 <http://books.google.de/books?id=d-c5-XLm4eEC>. – ISBN 9783827260802
- [Cle93] CLEVELAND, William S.: *Visualizing data*. Summit, NJ : Hobart Press, 1993
- [DEF⁺08] DUNKEL, Jürgen ; EBERHART, Andreas ; FISCHER, Stefan ; KLEINER, Carsten ; KOSCHEL, Arne: *Systemarchitekturen für Verteilte Anwendungen*. München : Hanser, 2008. – 124–139 S. – ISBN 978–3–446–41321–4
- [DFH08] DUSCH, Sebastian ; FONDEN, Christian ; HARBICH, Florian: *Datenstrommanagementsysteme*, Universität Stuttgart, Studie, 2008. http://w3studi.informatik.uni-stuttgart.de/~fondencn/fs_dsms_dusch_fonden_harbich.pdf
- [EB09] ECKERT, Michael ; BRY, François: Aktuelles Schlagwort: Complex Event Processing (CEP). In: *Informatik Spektrum* 32 (2009), Nr. 2, 163–167. <http://www.pms.ifi.lmu.de/publikationen/#PMS-FB-2009-5>
- [Espa] ESPERTECH INC.: *Esper Enterprise Edition: Enterprise ready Event Processing and CEP platform*. <http://www.espertech.com/products/espereee.php>
- [Espb] ESPERTECH INC.: *Event Stream Intelligence with Esper and NEsper*. <http://esper.codehaus.org/>
- [Esp06] ESPERTECH INC.: *Esper Reference Documentation - Version 0.7.5*. http://esper.sourceforge.net/esper-0.7.5/doc/reference/en/pdf/esper_reference.pdf. Version: 2 2006
- [Esp09a] ESPERTECH INC.: *Esper Reference Documentation - Version 3.2.0*. http://esper.codehaus.org/esper-3.2.0/doc/reference/en/pdf/esper_reference.pdf. Version: 9 2009
- [Esp09b] ESPERTECH INC.: *Index of /esper/prior-releases esper-3.2.0.zip*. <http://dist.codehaus.org/esper/prior-releases/>. Version: 09 2009
- [FB90] FEINER, S. K. ; BESHES, Clifford: Visualizing n-dimensional virtual worlds with n-vision. In: *Proceedings of the 1990 symposium on Interactive 3D graphics*. New York, NY, USA : ACM, 1990 (I3D '90). – ISBN 0–89791–351–5, 37–38
- [Ggo08] GGOBI: *GGobi - See what's going on in your data*. <http://www.ggobi.org/>. Version: 08 2008
-

-
- [Hel07] HELBERG, Anne: *Analyse und vergleichende Bewertung von Visualisierungstoolkits*, Universität Rostock Fakultät für Informatik und Elektrotechnik, Diplomarbeit, 08 2007. http://vcg.informatik.uni-rostock.de/assets/publications/theses_mas/DA_Helberg2007.pdf
- [Jea] JEAN-DANIEL FEKETE: *The InfoVis Toolkit*. <http://ivtk.sourceforge.net/>
- [Jäg10] JÄGER, Pascal: *Umsetzung der Kommunikationsmodellierungssprache ModoCommit Esper*, Hochschule für Angewandte Wissenschaften Hamburg - Department Informatik, Bachelorarbeit, 8 2010. <http://opus.haw-hamburg.de/volltexte/2010/1050/pdf/Bachelorarbeit.pdf>
- [Jun98] JUNG, Volker: *Integrierte Benutzerunterstützung für die Visualisierung in Geo-Informationssystemen*, Technischen Universität Darmstadt, Diss., 1998. <http://tuprints.ulb.tu-darmstadt.de/1127/1/dissvolk.pdf>
- [KAK95] KEIM, Daniel A. ; ANKERST, Mihael ; KRIEGEL, Hans-Peter: Recursive Pattern: A Technique for Visualizing Very Large Amounts of Data. In: *Proceedings of the 6th conference on Visualization '95*. Washington, DC, USA : IEEE Computer Society, 1995 (VIS 95). – ISBN 0–8186–7187–4, 279–
- [Kei02] KEIM, Daniel A.: Datenvisualisierung und Data Mining. In: *Datenbank-Spektrum 2* (2002), Nr. 2, 30–39. <http://www.datenbank-spektrum.de/pdf/dbs-02-30.pdf>
- [KK94] KEIM, Daniel A. ; KRIEGEL, Hans-Peter: Visdb: Database exploration using multidimensional visualization. In: *IEEE Computer Graphics and Applications 14* (1994), S. 40–49
- [KK95] KEIM, Daniel A. ; KRIEGEL, Hans-Peter: Visualisierungstechniken zur Exploration und Analyse sehr großer Datenbanken. In: *BTW*, 1995, 262-281
- [KK96] KEIM, Daniel A. ; KRIEGEL, Hans-Peter: Visualization Techniques for Mining Large Databases: A Comparison. In: *IEEE Transactions on Knowledge and Data Engineering 8* (1996), S. 923–938. <http://dx.doi.org/http://doi.ieeecomputersociety.org/10.1109/69.553159>. – DOI <http://doi.ieeecomputersociety.org/10.1109/69.553159>. – ISSN 1041–4347
- [Lö9] LÖFKEN, Jan O.: Auto 2020: Das rollende Rechenzentrum. In: *VDE-Dialog - Zeitschrift für VDE-Mitglieder 600* (2009), Nr. 4, 6-7. <http://www.vde.de/de/InfoCenter/VDE-Informationen/Magazine/VDE%20Dialog/Documents/dialog%204-2009%20WEB.pdf>
- [Leh10] *Visualisierung und Analyse multi-dimensionaler Datensätze*. 2010
-

- [Lev88] LEVKOWITZ, Haim: *Color in computer graphic representation of two-dimensional parameter distributions*. Philadelphia, PA, USA, Diss., 1988. <http://portal.acm.org/citation.cfm?id=914429>. – AAI8824759
- [LS08] LUCKHAM, David C. ; SCHULTE, Roy: *Event Processing Glossary - Version 1.1*. Online Resource. <http://complexevents.com/2008/08/31/event-processing-glossary-version-11/>, July 2008. – Last visited: October 2009.
- [Luc02] LUCKHAM, David: *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Professional, 2002 <http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0201727897>. – ISBN 0201727897
- [LWW90] LEBLANC, Jeffrey ; WARD, Matthew O. ; WITTELS, Norman: Exploring N-dimensional databases. In: *Proceedings of the 1st conference on Visualization '90*. Los Alamitos, CA, USA : IEEE Computer Society Press, 1990 (VIS '90). – ISBN 0-8186-2083-8, 230-237
- [MB95] MUNZNER, Tamara ; BURCHARD, Paul: Visualizing the Structure of the World Wide Web in 3D Hyperbolic Space. In: *VRML, 1995*, S. 33-38
- [Oel02] OELLIEN, Frank: *Algorithmen und Applikationen zur interaktiven Visualisierung und Analyse chemiespezifischer Datensätze*, Universität Erlangen-Nürnberg, Diplomarbeit, 2002. http://www2.chemie.uni-erlangen.de/services/dissonline/data/dissertation/Frank_Oellien/html/pdfs/Dissertation.pdf. – Kapitel 5 Data Mining und Datenvisualisierung
- [Ope] OPEN SOURCE INITIATIVE: *GNU General Public License, version 2 (GPLv2) The GNU General Public License (GPL) Version 2, June 1991*. <http://www.opensource.org/licenses/gpl-2.0.php>
- [PG88] PICKETT, R. M. ; GRINSTEIN, G. G.: Iconographic Displays For Visualizing Multidimensional Data. In: *Systems, Man, and Cybernetics, 1988. Proceedings of the 1988 IEEE International Conference on* Bd. 1, 1988, 514-519
- [Pot08] POTT, Brigitte: *Beispiele für Visualisierungstools (Linkliste)*. Online Dokument. <http://www.ibi.hu-berlin.de/infokomp/materialien/tools/linktools>. Version: 03 2008. – Humboldt-Universität zu Berlin
- [Pre07] PREHOFER, Michaela: *Abhandlung über SW-Visualisierungstools*. Online Dokument. http://www.splendit.at/downloads/dl_dok.html. Version: 09 2007. – Splendit IT-Consulting GmbH
- [Res07] RESSEL, Christian: *Haussysteme-Bedienung heute und in Zukunft*. Version: 2007. http://www.inhaus-zentrum.de/_uploads/media/88_
-

Haussysteme-Bedienung%20heute%20und%20in%20Zuk.pdf. In: *Electronic home-Jahrbuch 2007* Bd. 2007. Electronic home, 2007, 107-109

- [RMC91] ROBERTSON, George G. ; MACKINLAY, Jock D. ; CARD, Stuart K.: Cone Trees: animated 3D visualizations of hierarchical information. In: *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*. New York, NY, USA : ACM, 1991 (CHI '91). – ISBN 0-89791-383-3, 189-194
- [Sch08] *Kapitel 7*. In: SCHELER, Dipl.-Ing. I.: *Analyse und Visualisierung raumplanerischer Prozesse mit Hilfe von Voronoi-Diagrammen*. Fachbereich Informatik der Technischen Universität Kaiserslautern, 2008, 85-111
- [Se10] SENSORIK E.V., AMA F.: *AMA Fachverband für Sensorik e.V.* <http://www.ama-sensorik.de/site/de/289/wirtschaftliche-bedeutung.html>. Version: 02/2010
- [Shn92] SHNEIDERMAN, Ben: Tree Visualization with Tree-Maps: 2-d Space-Filling Approach. In: *ACM Trans. Graph.* 11 (1992), Nr. 1, S. 92-99
- [SM00] SCHUMANN, Heidrun ; MÜLLER, Wolfgang: *Visualisierung: Grundlagen und allgemeine Methoden*. Springer Verlag, 2000
- [Sma09] SMARTHOME PADERBORN ; REDAKTIONSBÜRO, GO (Hrsg.): *SmartHome Broschüre*. <http://www.smarthomepaderborn.de/Pages/upload/daten/docs/1237893554.pdf>. Version: 3 2009
- [SMZ07] SOHRABY, Kazem ; MINOLI, Daniel ; ZNATI, Taieb: *Wireless Sensor Networks: Technology, Protocols, and Applications*. Wiley-Interscience, 2007. – ISBN 0471743003
- [STDS95] SPENCE, B. ; TWEEDIE, L. ; DAWKES, H. ; SU, Hua: Visualization for functional design. In: *Information Visualization, IEEE Symposium on 0* (1995), 4. <http://dx.doi.org/http://doi.ieeecomputersociety.org/10.1109/INFVIS.1995.528680>. – DOI <http://doi.ieeecomputersociety.org/10.1109/INFVIS.1995.528680>. ISBN 0-8186-7201-3
- [Ste00] STEINMETZ, R.: *Multimedia-Technologie: Grundlagen, Komponenten und Systeme*. Springer, 2000 http://books.google.de/books?id=jXRenVdVG_cC. – ISBN 9783540673323
- [tab10] TABLEAUSOFTWARE: *The Geography of Diabetes*. <http://www.tableausoftware.com/public/gallery/geography-diabetes>. Version: 2010
-

- [The] THE XMDVTOOL PROJECT: *XmdvTool*. <http://davis.wpi.edu/~xmdv/index.html>
- [Tie08] TIEBLER, Daniel: *Optimierung multidimensionaler Bereichsanfragen mittels raumfüllender Kurven in Peer-to-Peer-Netzen*, Universität Stuttgart, Diplomarbeit, 8 2008. http://elib.uni-stuttgart.de/opus/volltexte/2009/4408/pdf/DIP_2731.pdf. – 9–10 S.
- [TT80] TURNER, D. W. ; TIDMORE, F. E.: FACES-A FORTRAN program for generating Chernoff-type faces on a line printer. In: *American Statisticians* 34 (1980), 187. <http://www.jstor.org/pss/2683888>
- [WAM01] WEBER, Marc ; ALEXA, Marc ; MÜLLER, Wolfgang: Visualizing Time-Series on Spirals. In: *Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*. Washington, DC, USA : IEEE Computer Society, 2001. – ISBN 0-7695-1342-5, 7-14
- [WB97] WONG, Pak C. ; BERGERON, R. D.: 30 Years of Multidimensional Multivariate Visualization, IEEE Computer Society Press, 1997, 3–33
- [WIK10a] WIKIPEDIA: *Anforderung (Informatik) - Wikipedia, Die freie Enzyklopädie*. [http://de.wikipedia.org/w/index.php?title=Anforderung_\(Informatik\)&oldid=70544602](http://de.wikipedia.org/w/index.php?title=Anforderung_(Informatik)&oldid=70544602). Version: 2010
- [WIK10b] WIKIPEDIA: *Data Mining*. http://de.wikipedia.org/wiki/Data_Mining. Version: 2010
- [WIK10c] WIKIPEDIA: *Mock-up - Wikipedia, Die freie Enzyklopädie*. <http://de.wikipedia.org/wiki/Mock-up>. Version: 2010
- [WIK10d] WIKIPEDIA: *Sensor - Wikipedia, Die freie Enzyklopädie*. <http://de.wikipedia.org/w/index.php?title=Sensor&oldid=70739100>. Version: 2010
- [Wik10e] WIKIPEDIA: *Univariat — Wikipedia, Die freie Enzyklopädie*. <http://de.wikipedia.org/w/index.php?title=Univariat&oldid=79084449>. Version: 2010
- [Wir] WIRTSCHAFTSLEXIKON24: *Data Mining*. <http://www.wirtschaftslexikon24.net/d/data-mining/data-mining.htm>
- [WL93] WIJK, Jarke J. ; LIERE, Robert van: HyperSlice: visualization of scalar functions of many variables. In: *Proceedings of the 4th conference on Visualization '93*. Washington, DC, USA : IEEE Computer Society, 1993 (VIS '93). – ISBN 0-8186-3940-7, 119–125
-

- [Zak07] ZAKHOUR, J.R.M.H.I.R.T.R.S.H.S.: *Das Java Tutorial*. Addison-Wesley, 2007 <http://books.google.de/books?id=wAq1QilPfmwC>. – ISBN 9783827326652
-

Abbildungsverzeichnis

3.1	Schema der Ausgabe bei einem EPL-Statement mit Time Window [Esp09a)	22
3.2	Schema der Ausgabe bei einem EPL-Statement mit Time Batch [Esp09a)	23
3.3	Schema von Längenfenster-Views [DEF ⁺ 08]	23
4.1	Beispiel einer Scatterplot-Matrix mit Fahrzeugdaten aus dem Jahre 1993 von 6 verschiedener Automobilklassen	39
4.2	Veranschaulichung des Prinzips der Prosection View	40
4.3	Veranschaulichung von Hyperslices: a) Konzept eines Hyperslices; [SM00] b) Hyperslices einer 4-dimensionalen Funktion [WL93]	42
4.4	Darstellung der Fahrzeugdaten von Abbildung 4.1 als Parallele Koordinaten zusätzlich mit Fahrzeugtyp als erste Achse (Erstellt mit Ggobi [Ggo08])	43
4.5	Parahistogramm einer mikrobiologischen Datenmenge [SM00]	44
4.6	Spiral Graph Darstellung der Sonnenscheinintensität eines Ortes über mehrere Tage [WAM01]	45
4.7	Bubble Graphs Karte der Diabeteskranken in den USA [tab10]	46
4.8	Beispiel für verschiedene Stick-Figures-Ikone [KK95]	47
4.9	Stick Figures-Visualisierung der großen Seen in den USA [KK96]	47
4.10	Aufbau einer Shape Coding Ikone [SM00]	48
4.11	Shape Coding Visualisierung der mikrobiologischen Merkmale aus der Datenmenge von Abbildung 4.5 [SM00]	49
4.12	Beispiel eines Color Icons [SM00]	50
4.13	Beispiele für Chernoff Faces	51
4.14	Visualisierung der Lebensbedingungen in L.A. mittels Chernoff Faces [TT80]	51
4.15	Zwei Vertreter der Pixelbasierten Techniken: a) Visualisierung von Finanzkursverläufen mittels Einfache Techniken und b) mittels Raumfüllende-Kurven-Techniken [Kei02]	52
4.16	Ein Beispiel der Zwei-Schritte-Techniken: Das Circle Segment [AKK96]	54
4.17	Veranschaulichung der Recursive Pattern-Techniken: Bildung von rekursiven Pattern [SM00]	55
4.18	Vier Beispiele für Anordnungsschemata der Recursive Pattern-Techniken [KAK95]	55
4.19	Schematischer Aufbau des Dimensional Stacking (Eigene Darstellung nach [SM00])	57

4.20	Dimensional Stacking von Ölförderdaten [Kei02]	58
4.21	Prinzip der Worlds-within-Worlds-Visualisierung [SM00]	59
4.22	Euklidische Cone Trees Visualisierung [MB95]	60
4.23	Treemap Ansicht einer Computerfestplatte (Eigene Darstellung erstellt mit SequoiaView [BGH ⁺ 00])	61
5.1	Visualisierungsbeispiel vom Visualisierungswerkzeug <i>XmdvTool</i> [The] . .	74
5.2	Visualisierungsbeispiel vom Visualisierungswerkzeug <i>Walrus - Graph Visualization Tool</i> [Cai]	75
5.3	Visualisierungsbeispiel vom Visualisierungswerkzeug <i>The InfoVis Toolkit</i> [Jea]	75
5.4	Architektur des Visualisierungswerkzeugs (Eigene Darstellung)	77
5.5	Mock-up der Programmoberfläche (Eigene Darstellung)	78
5.6	Mock-up 1 der Benutzerinteraktion: Auswahl einer Visualisierung (Eigene Darstellung)	79
5.7	Mock-up 2 der Benutzerinteraktion: Anzeige einer Visualisierung (Eigene Darstellung)	80
5.8	Mock-up 3 der Benutzerinteraktion: Auswahl von Daten in einer Visualisierung (Eigene Darstellung)	80
5.9	Mock-up zweier überlappender Visualisierungen (Eigene Darstellung) . .	81

Tabellenverzeichnis

4.1	Anforderungskatalog an das Visualisierungswerkzeug mit Gewichtung .	37
4.2	Übersicht über die Bewertungen der Visualisierungstechniken hinsichtlich einer visuellen Unterstützung	64

Listingverzeichnis

1.1	Beispiel einer EPL-Anfrage	2
3.1	Aufbau einer Esper-EPL-Anfrage [Esp09a]	16
3.2	Beispiel einer einfachen <code>SELECT</code> -Anfrage	18
3.3	Beispiel für ein EPL-Statement mit <code>WHERE</code> -Klausel	18
3.4	Beispiel für ein <code>GROUP-BY-</code> und <code>HAVING-EPL-Statement</code>	19
3.5	Beispiel eines EPL-Statements mit <code>OUTPUT-</code> , <code>ORDER BY-</code> und <code>LIMIT-Klausel</code>	19
3.6	Pattern Matching: Beispiel eines EPL-Statements mit Pattern (aus [Esp09a])	20
3.7	Time Windows: Beispiel eines EPL-Statements mit Time Window (aus [Esp09a])	21
3.8	Einfaches EPL-Statement mit Time Window (aus [Esp09a])	21
3.9	Time Windows: Beispiel eines EPL-Statements mit Time Batch (aus [Esp09a])	21
5.1	Syntaxschema einer Esper-EPL-Anfrage [Esp09a]	67

Eidesstattliche Erklärung

Ich versichere, dass ich die vorstehende Arbeit selbstständig und ohne fremde Hilfe angefertigt und mich anderer als der im beigefügten Verzeichnis angegebenen Hilfsmittel nicht bedient habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht.

Ich bin mit einer Einstellung in den Bestand der Bibliothek des Departments Informatik einverstanden.

Hamburg, den _____ Unterschrift: _____