

Prototypische Entwicklung einer Trend basierten Datenbank-Suchmaschine

vorgelegt von
Diplom-Psychologe Jörg Huss
aus Potsdam

von der Fakultät III - Prozesswissenschaften
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr.-Ing. –

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr. G. Tsatsaronis

Berichter: Prof. Dr. G. Wozny

Berichter: Prof. Dr. L. Urbas

Tag der wissenschaftlichen Aussprache: 25. April 2008

Berlin 2008

D83

Vorwort

Die vorliegende Arbeit stellt einen Überblick meiner wissenschaftlichen Arbeit als Doktorand am Graduiertenkolleg *prometei* dar. Dieses Kolleg hat sich der prospektiven Gestaltung von Mensch-Technik-Interaktion in verschiedenen Anwendungsbereichen verschrieben. Als Forschungsprojekt des Zentrums Mensch-Maschine-Systeme (Technische Universität Berlin) vereint das Kolleg seit 2004 die verschiedensten Fachgebiete und Disziplinen.

Um die Idee des interdisziplinären Austausches persönlich zu verfolgen, habe ich mich als Psychologe in die Obhut zweier Ingenieure gewagt. An dieser Stelle gilt mein ganz besonderer Dank meinen Betreuern und Gutachtern Prof. Wozny (Fachgebiet Dynamik und Betrieb technischer Anlagen, TU-Berlin) und Prof. Urbas (Professur für Prozessleittechnik, TU-Dresden). Durch ihr nachhaltiges Vertrauen haben sie mir eine Reihe von Erfahrungen ermöglicht, die weit über die fachliche Bearbeitung der Mensch-Technik-Interaktion in der Prozessführung hinausgehen.

Eine wesentliche Erfahrung war die fachliche Zusammenarbeit mit Kollegen verschiedenster (nichtpsychologischer) Fachrichtungen und Nationalitäten. Am erfolgreichsten gestaltete sich diese Kooperation ab Sommer 2006 mit Alexander Mohammed Hatri, zurzeit Mathematikstudent der Humboldt-Universität zu Berlin. Neben der notwendigen Implementierungsleistung technischer Komponenten hat mir der intensive gemeinsame Austausch über Konzepte, Probleme und Lösungen der Signalverarbeitung gezeigt, wie sich ein echtes interdisziplinäres Verständnis aufbauen und effektiv umsetzen lässt. Ich danke Alex für seinen Enthusiasmus, seine Unvoreingenommenheit und seine Kollegialität.

Den Weg für diese Zusammenarbeit haben im Jahr 2005 zu einem wichtigen Teil Christopher Lörken (Kognitionswissenschaften, Universität Osnabrück) und Oliver Frey (Physikalische Ingenieurwissenschaften, TU-Berlin) geebnet. Beide haben viele Ideen und Artefakte in ein gemeinsames Kooperationsprojekt eingebracht, welches mit der BASF AG Ludwigshafen durchgeführt wurde. Martin Schwibach gebührt als Ansprechpartner und Abteilungsleiter an dieser Stelle der Dank unseres gesamten Teams für seine Wertschätzung unserer universitären Forschungsarbeit.

Tilman Barz (Energie- und Verfahrenstechnik, TU-Berlin), mein Doktorandenkollege am Forschungsschwerpunkt „Prozessmonitoring“ des GRK *prometei* hat mir in vielen geduldigen Gesprächen zu einem Grundverständnis thermischer Trennprozesse verholfen. Darüber hinaus war er mir über die gesamte Zeit ein verlässlicher Partner und menschlicher Zimmergenosse. Martin Christof Kindsmüller möchte ich für die Vorarbeit seiner erstklassigen Dissertation zur *Trend-Literacy* bedanken. Nicht nur die Lektüre war eine Freude und Bereicherung, auch die gemeinsamen Gespräche in der Anfangsphase meines Promotionsprojektes haben viel für mein Verständnis der Aufgabendomäne beigetragen.

Einige Kollegen des Zentrum Mensch-Maschine-Systeme haben ihren Anteil am Gelingen dieser Arbeit getragen, indem sie mir immer wieder eine menschlich und fachlich motivierte Freude an der Arbeit auf dem TU-Campus vermittelten. Ganz besonders denke ich da an Boris Gauss, Katja Karrer, Thorsten Zander, Mario Lasch, Christiane Steffens und Birgit Trogisch. Den drei letztgenannten bin ich zudem für ihre technische und administrative Unterstützung zu Dank verpflichtet, welche an vielen Stellen des universitären Alltags weit über das übliche Maß hinaus ging. Anne Klostermann hat mir als Psychologin in gemeinsamen Gesprächen und Kleinprojekten immer wieder rückgemeldet, wie weit ich auf meinem Weg zum Ingenieur bereits von der Denkweise der Psychologie entfernt hatte – und mich damit vor Schlimmeren bewahrt. Ihren Teil beigetragen haben auch meine Studentischen Mitarbeiter und Praktikanten Dorota Szymanska (BWL, Europa Universität Viadrina, Frankfurt/Oder), Oliver Walter Schreiber (Sozialwissenschaften, Universität Erfurt) und Zakaria Laaroussi (Informatik, TU-Berlin).

Meiner Schwester Antje gilt mein besonderer Dank für ihr unkompliziertes und selbstverständliches Exil, welches sie mir für den Abschluss der komplizierten Kapitel gewährt hat. Fernab von Alltagsverpflichtungen und –freuden hatte ihr großer Esstisch die nötige Stabilität für das schwere Gewicht an Büchern und Gedanken. Juliane Domeinski, Ex-Kommilitonin, Kollegin und Freundin danke ich für die selbstverständliche Hilfe bei der Endkorrektur meiner schriftlichen Ausarbeitungen.

Parallel zur Arbeit an meinem Promotionsprojekt habe ich meine Partnerin Sandra Scholz kennen und lieben gelernt und mit ihr eine Familie gegründet. Zusammen mit unserer Tochter Juna Philine

hat sie mich in dieser wichtigen und schwierigen Zeit begleitet. Unsere Familie wird nachhaltiger und auf eine ganz andere Art und Weise mein Schaffen und Wirken beeinflussen, als es die Erfahrungen meiner Promotionszeit getan hat. Darum widme ich diese Dissertation meiner jungen Familie in Hoffnung auf gutes Gelingen.

Potsdam, Januar 2008

Jörg Huss

Zusammenfassung

Die Dynamik einzelner Variablen nichtlinearer Prozesse erschließt sich für den Experten unmittelbar aus deren zeitlichen Trendverläufen. Kindsmüller (2006) hat erstmals eine psychologisch fundierte Theorie des Trendverstehens (Trend-Literacy) aufgestellt und empirisch überprüft. Er fokussierte dafür die Anwendungsdomäne der verfahrenstechnischen Prozessführung. Aufgrund zunehmender Komplexität und Koppelung derartiger Prozesse, fortschrittlicher Messinstrumente und kostengünstiger Datenspeicherung wird gerade in diesem Bereich eine kontinuierlich steigende Menge an Prozessdaten in Form von Zeitreihen zur Verfügung gestellt. Allerdings ist ein effizienter und unmittelbarer Zugang zu diesen Daten nicht gewährleistet. Ziel der vorliegenden Arbeit war es deshalb, eine prototypische, Trend basierte Suchmaschine für die Zeitreihenanalyse zu entwickeln. Dafür wurde die Synthese von effizienten Methoden der Signalverarbeitung mit Mechanismen des menschlichen Trendverstehens angestrebt.

Inhalt der schriftlichen Ausarbeitung ist die vollständige Beschreibung des Entwicklungsprozesses. Ausgehend von einer Problemdarstellung wird die anfängliche Design-Idee skizziert. Darauf folgt eine Funktionsanalyse, welche sowohl den aktuellen Forschungsstand zu Transformations- und Vergleichsverfahren von Zeitreihen aufarbeitet als auch einige Anwendungsbeispiele in den Domänen der Prozessführung und der Finanzanalyse aufzeigt. Inhalt einer parallel durchgeführten Aufgabenanalyse war die empirische Erfassung des Nutzungskontextes von Kurvenbildern sowie des aktuellen Unterstützungsbedarfs. Anhand exemplarischer Aufgabenszenarien wurde zudem ein früher Prototyp des Nutzerdialogs evaluiert.

Das Systemdesign erfolgte in zwei parallelen Entwicklungspfaden. Ein erster Pfad fokussierte die rein technische Entwicklung eines effizienten Suchalgorithmus. Zwei nacheinander erstellte Entwürfe werden vorgestellt und begründet. Die Vor- und Nachteile werden aufgezeigt, wobei der zweite Entwurf nach einer zufriedenstellenden analytischen Evaluation einem normativen Testverfahren unterzogen wurde, dessen Verlauf und Ergebnisse abschließend dokumentiert werden. Der zweite Pfad beinhaltete die Konzeption eines gebrauchstauglichen Nutzerdialogs. Die Dissertations-

schrift stellt den dabei durchlaufenen Entwurfsprozess inklusive Expertenevaluation und Re-Design dar und gibt einen Überblick über die wesentlichen Dialogprinzipien.

Mit dem so entwickelten, voll funktionsfähigen Prototyp war die Grundlage für eine ganzheitliche Systembewertung im Rahmen von experimentellen Nutzertests geschaffen. Die Ergebnisse der summativen Evaluation bestätigten die erfolgreiche Konzeption der Trend basierten Suchmaschine. Darüber hinaus wurden im gesamten Entwicklungsprozess wichtige und interessante Möglichkeiten der Weiterentwicklung deutlich, welche im abschließenden Ausblick thematisiert werden. Darüber hinaus wird die Generalisierbarkeit der Entwicklungsmethodik und der prototypischen Entwicklung auf Anwendungsdomänen innerhalb und außerhalb der Prozessführung diskutiert.

Summary

The dynamic of particular variables in nonlinear processes is perceived and interpreted by experts directly from trend displays. Kindsmüller (2006) has hypothesized and empirically proven a psychological theory of Trend-Literacy while focusing on the application domain of chemical process engineering. Caused by an increase of complexity and coupling of chemical processes, progressive measurements, and cost efficient data storage, an increasing amount of process data is provided in terms of time series. However, no efficient or direct access is possible yet. Thus, the goal of the present design study is the development of a trend-based search engine for expert-oriented matching and analysis of time series. A synthesis of efficient methods of signal processing and expert abilities in Trend-Literacy has been pursued.

The aim of this thesis is an exhaustive description of the entire design process. Starting with a problem statement, the initial design idea is outlined. The following function analysis gives a brief overview of recent publications on data mining methods and their applications. Additionally, a comprehensive benchmark analyses the pros and cons of already existing trend-based search engines from different domains. An empirical task analysis has also been conducted regarding the contextual use of trend charts and the need for potential support systems. Finally, an early prototype of the user dialog is evaluated through scenario-based requirement analysis.

The system design has been distinguished into two parallel design paths. The first path focuses on the development of a search algorithm. Two subsequent drafts are described and justified in this thesis. The evaluation of the second draft by normative test scenarios is reported. The second path of design process involves the development of a user dialog. The main interaction principles are exemplified and an evaluation by usability experts is provided.

The resulting high fidelity prototype has been evaluated by user experiments. This was done using artificial scenarios of time series. The results show that users are able to learn effective interaction with the user dialog and to find satisfying results in the data base. Additionally, some interesting opportunities for future related work are reported within the closing section of this paper. Finally, the generalization of methodology and prototype is discussed.

Inhaltsübersicht

VORWORT	2
ZUSAMMENFASSUNG	5
SUMMARY	7
INHALTSÜBERSICHT	8
INHALTSVERZEICHNIS	9
I EINFÜHRUNG	12
I.1 Zielstellung	12
I.2 Vorgehensweise.....	14
II SYSTEMANALYSE	19
II.1 Funktionsanalyse – Stand der Technik.....	21
II.2 Aufgabenanalyse – Stand des Wissens	37
II.3 Zusammenfassung.....	52
III SYSTEMDESIGN.....	54
III.1 Suchalgorithmus.....	56
III.2 Nutzerdialog	100
III.3 Implementierung.....	122
IV SYSTEMBEWERTUNG	125
IV.1 Untersuchungsmaterial	126
IV.2 Durchführung.....	131
IV.3 Ergebnisse.....	132
IV.4 Zusammenfassung und Diskussion	140
V SYSTEMKORREKTUR - AUSBLICK.....	143
LITERATURVERZEICHNIS	146
BEGRIFFSGLOSSAR	153
ZEICHENGLOSSAR.....	157
ANHANG.....	163

Inhaltsverzeichnis

VORWORT	2
ZUSAMMENFASSUNG	5
SUMMARY	7
INHALTSÜBERSICHT	8
INHALTSVERZEICHNIS	9
I EINFÜHRUNG	12
I.1 Zielstellung	12
I.2 Vorgehensweise.....	14
I.2.1 Verwendung von Prototypen	14
I.2.2 Parallel-iterative Entwicklung	16
I.2.3 Schriftliche Ausarbeitung.....	17
II SYSTEMANALYSE	19
II.1 Funktionsanalyse – Stand der Technik.....	21
II.1.1 Transformation von Zeitreihen	22
II.1.2 Methoden für den Vergleich von Zeitreihen	28
II.1.2.1 Segmentieren.....	28
II.1.2.2 Indexieren	30
II.1.3 Anwendungen von Zeitreihenvergleichen.....	31
II.1.3.1 Zustandsdiagnose	31
II.1.3.2 Fehlerdiagnose	32
II.1.3.3 Datenbanksuche	34
II.2 Aufgabenanalyse – Stand des Wissens	37
II.2.1 Das Kurvenbild als Werkzeug der Prozessüberwachung und -analyse	39
II.2.1.1 Feldstudie	39
II.2.1.2 Zusammenfassung.....	43
II.2.2 Technischer Unterstützungsbedarf	43
II.2.3 Nutzerexpertise bei der Prozessüberwachung und -analyse.....	45
II.2.4 Entwurf potentieller Aufgabenszenarien.....	46
II.2.5 Aufgabenanalyse anhand einer Prototypen-Evaluation.....	50
II.3 Zusammenfassung.....	52

III	SYSTEMDESIGN.....	54
III.1	Suchalgorithmus.....	56
III.1.1	Entwurf I.....	56
III.1.1.1	Grundkonzept.....	56
III.1.1.2	Vorverarbeitung.....	58
III.1.1.2.1	Identifikation linearer Episoden.....	58
III.1.1.3	Suchprozedur.....	59
III.1.1.3.1	Suche nach linearen Episoden.....	59
III.1.1.3.2	Suche nach gekrümmten Episoden.....	60
III.1.1.4	Bewertung von Entwurf I.....	61
III.1.1.4.1	Explorative Testergebnisse.....	61
III.1.1.4.2	Probleme von Episodenübergängen.....	62
III.1.1.4.3	Inkompatibilitäten mit Nutzerdialog.....	63
III.1.2	Entwurf II.....	65
III.1.2.1	Grundkonzept.....	65
III.1.2.2	Vorverarbeitung.....	68
III.1.2.2.1	Identifikation gekrümmter Episoden.....	69
III.1.2.2.2	Identifikation linearer Episoden.....	70
III.1.2.3	Suchprozedur.....	73
III.1.2.3.1	Suchraumeinschränkung.....	77
III.1.2.3.2	Suche der Startepisoden.....	78
III.1.2.3.3	Suche der Folgeepisoden.....	80
III.1.2.3.4	Eliminierung von Redundanzen.....	82
III.1.2.3.5	Ähnlichkeitsschätzung.....	84
III.1.2.4	Konfigurationsheuristiken.....	88
III.1.2.4.1	Parameter der Vorverarbeitung.....	88
III.1.2.4.2	Parameter der Suchprozedur.....	90
III.1.2.5	Explorative Evaluation.....	91
III.1.2.5.1	Testszenarien.....	91
III.1.2.5.2	Testkonfiguration.....	94
III.1.2.5.3	Ergebnisse.....	95
III.2	Nutzerdialog.....	100
III.2.1	Entwurfsprozess.....	100
III.2.2	Prototyp.....	103
III.2.2.1	Dialogprinzipien.....	103
III.2.2.2	Referenzkurvenauswahl.....	104
III.2.2.3	Kurvenfeld.....	107
III.2.2.4	Kurvennavigation.....	107

III.2.2.5	Trendbeschreibung.....	108
III.2.2.6	Expertenmodus.....	108
III.2.2.7	Ergebniskurvenauswahl	110
III.2.2.8	Ergebniskurvennavigation	111
III.2.3	Expertenevaluation	112
III.2.3.1	Evaluationsstudie.....	114
III.2.3.2	Ergebnisse	118
III.2.4	Re-Design.....	120
III.3	Implementierung	122
IV	SYSTEMBEWERTUNG	125
IV.1	Untersuchungsmaterial	126
IV.1.1	Testsets und Testszenarien.....	126
IV.1.2	Parametereinstellungen.....	130
IV.2	Durchführung	131
IV.3	Ergebnisse.....	132
IV.3.1	Abweichungen von normativer Trendbeschreibung.....	133
IV.3.2	Ergebnisgüte in Abhängigkeit von der Trendbeschreibung.....	136
IV.3.3	Einfluss der Episodenanzahl auf die Ergebnis-güte	138
IV.3.4	Nutzerevaluation der Ähnlichkeitsschätzung	139
IV.4	Zusammenfassung und Diskussion	140
V	SYSTEMKORREKTUR - AUSBLICK.....	143
	LITERATURVERZEICHNIS	146
	BEGRIFFSGLOSSAR	153
	ZEICHENGLOSSAR.....	157
	ANHANG.....	163

I Einführung

I.1 Zielstellung

Die Idee zu dem hier verfolgten Promotionsprojekt entstand vor dem Hintergrund jüngster Technologieentwicklungen auf dem Gebiet der Prozessindustrie. Diese unterliegt seit Jahrzehnten einem stetigen Zuwachs an Komplexität und Kopplung von Prozessautomatisierung (Huber, 2001), Verfahrenstechnik und Datenmanagementsystemen (Pieters & Krätzner, 2006). Zudem steigt der Umfang an verfügbarem Speicherplatz bei gleichzeitig fallenden Preisen. Die Nutzung dieses Speichervolumens wird durch Entwicklungen auf dem Gebiet der Messtechnik und Prozesssimulation beschleunigt. Damit einher geht eine zunehmende Anzahl verfügbarer Steuer- und Regelgrößen. Zudem werden komplexe Prozessmodelle zur Kosten senkenden Prozessoptimierung eingesetzt (Schuler, 2006). Mit der steigenden Quantität, Komplexität und Diversität der vorhandenen Daten wird der Einsatz effizienter Methoden und Werkzeuge zur Informationsverarbeitung notwendig (Treier et al. 2006).

Die aktuellen Überwachungs- und Analysesysteme werden diesen Anforderungen jedoch nur teilweise gerecht. So wird dem Prozessexperten ermöglicht, anhand von nominalen Daten wie Schichtwechsel, Alarme und Steuereingriffe in der Prozesshistorie zu recherchieren (siehe Tabelle 5 in Kapitel II.2.2). Ein führender Anbieter von Prozessleitsystemen wirbt zum Beispiel mit folgendem Angebot: *„Archived history may be accessed automatically by simply scrolling to, or directly entering, the appropriate time and date.“* (Honeywell Inc., 2005, S. 9). Eine Suche in Datenbanken anhand verschiedener Suchkriterien wird im Werbetext eines weiteren Marktführers beschrieben: *„You can search ... for previous processes by product, procedure, cycle time, or any other designation. You can even search for batches with a specific common variable.“* (OSIsoft Inc., 2007, S. 1). Die intensive Nutzung von Kurvenbildern und Prozessdynamiken bei der Prozessüberwachung und -analyse durch Operateure und Prozessingenieure (vgl. Kapitel II.2) wird durch die beschriebenen Unterstützungswerkzeuge bisher jedoch nur indirekt und unzureichend unterstützt.

Diese Lücke soll durch die Entwicklung einer Trend basierten Suchmaschine geschlossen werden, welche

- einen unmittelbaren und gebrauchstauglichen Informationszugriff durch die Nutzung signifikanter Prozessdynamik als explizites Suchkriterium, und
- die Kombination von Fähigkeiten der Prozessexperten im Kurvenlesen mit effizienten mathematischen Vergleichsmethoden ermöglicht.

Die Interaktion mit der Suchmaschine erfolgt dabei in Analogie zu bekannten Text basierten Suchmaschinen (Tabelle 1). Abfolgen zeitlicher Episoden bestimmter Dynamik, sogenannte Trends, dienen als Schlüsselwörter. Neben der Konzeption des Nutzerdialogs beinhaltet die Arbeit den Entwurf und die prototypische Entwicklung eines Algorithmus zum Vergleich dieser Trends anhand signifikanter Merkmale. Zudem wurde ein besonderes Augenmerk auf die ganzheitliche Überprüfung des prototypisch entworfenen Mensch-Technik-Systems gerichtet.

Tabelle 1 Aufgabenablauf bei Datenbankrecherche^a

Funktions-träger	Trend basierte Suche	Text basiert Suche
Mensch	Kurvenlesen & Mustererkennung	Textlesen & Inhaltsanalyse
Mensch	Trendbeschreibung	Schlagwortdefinition
Technik	Trendsuche	Schlagwortsuche
Technik	Ausgabe von Ergebniskurven	Ausgabe von Ergebnistexten
Mensch	Recherche in Kurvenausschnitten	Recherche in Textauswahl

^a Da einige Begriffe bisher noch nicht eingeführt wurden, sei auf das Begriffsglossar am Ende der Dissertationsschrift verwiesen.

1.2 Vorgehensweise

Im folgenden Kapitel wird die systematische Vorgehensweise bei der Erstellung einer Entwicklungslösung erläutert. Wesentlicher Aspekt ist dabei die Verwendung von Prototypen in parallel iterativen Entwurfszyklen, mit welcher die prospektive Gestaltung von gebrauchstauglicher Mensch-Technik-Interaktion erreicht werden soll. Die inhaltliche Gliederung der Dissertationsschrift orientiert sich direkt an dem entworfenen Entwicklungsprozess und wird zum Ende des Einführungskapitels erläutert.

1.2.1 Verwendung von Prototypen

Bereits in den 80er Jahren steigerte sich die Popularität von Prototypen in der Entwicklung von Informationssystemen. Ausgehend von folgender Begriffsdefinition wurde von Alavi (1984) eine Untersuchung zum Nutzen dieser Entwicklungsmethode vorgestellt: *„An information systems prototype is an early version of a system that exhibits the essential features of the later operational system.“* (Alavi, 1984, S. 556). Auf Basis von Interviews mit Projektmanagern und Systemanalysten verschiedener Anwendungsdomänen wurde eine Reihe von Vorteilen des Prototyping festgestellt. Zum einen liefern Nutzerbefragungen anhand von greifbaren Systementwürfen vollständigere und sinnvollere Rückmeldungen. *“The users are extremely capable of criticizing an existing system but not too good at articulating or anticipating their needs.“* (Alavi, 1984, S. 557). Zum anderen fördert die Bereitstellung von Prototypen die Partizipation von Nutzern im Entwicklungsprozess. Des Weiteren ermöglichen Prototypen zu einem frühen Zeitpunkt die Evaluation und Korrektur von Leistungsmängeln, bevor hohe Kosten und viele Ressourcen für die Entwicklung von komplett integrierten und industriell gefertigten Systemen aufgewendet werden müssen.

Nach Pomberger und Blaschek (1996) kann zwischen vertikalem und horizontalem Prototyping unterschieden werden. Beim *vertikalen Prototyping* wird ein ausgewählter Teil des Systems durch alle Ebenen hindurch implementiert. Im Gegensatz dazu wird im *horizontalen Prototyping* eine spezifische Systemebene möglichst vollständig dargestellt, z. B. bei der Realisierung eines Nutzerdialogs ohne die Implementierung technischer Funktionalitäten.

Auf einer zweiten Unterscheidungsdimension werden *explorative*, *experimentelle* und *evolutionäre* Arten von *Prototyping* unterschieden (Pomberger und Blaschek, 1996). Das *explorative Prototyping* dient einem besseren Problemverständnis und dem Aufdecken von Spezifikationslücken. Dazu wird sich meist auf eine horizontale Betrachtung der Systemfunktionalitäten konzentriert. Die Effizienz des Prototypen ist dabei von zweitrangiger Bedeutung. Das *experimentelle Prototyping* wird zu Forschungszwecken bzw. zur Suche nach Realisierungsmöglichkeiten angewandt. Mit einem experimentellen Prototyp werden klassischer Weise sehr umfangreiche Problemanalysen auf vertikaler Ebene durchgeführt. Die dabei gewonnenen Erkenntnisse können anschließend zur Produktentwicklung verwertet werden. Beim *evolutionären Prototyping* wird die Funktionalität lauffähiger Prototypen schrittweise bis zur Produktreife erweitert. Dabei werden in regelmäßigen Iterationen Feedbacks der zukünftigen Nutzer bzw. des Auftraggebers eingeholt.

Die Einordnung des bei der Entwicklung der Trend basierten Datenbanksuchmaschine angewandten *Prototyping* ist abhängig von der Definition der Systemgrenzen. Betrachtet man das Gesamtsystem *Leitwarte*, stellt die Funktionalität der Suchmaschine einen Teilaspekt dar, welcher in vollständiger Tiefe von der Bildschirmeingabe bis zum Quellcode des Suchalgorithmus prototypisch umgesetzt wird. Innerhalb des prozessnahen Fertigungsmanagementsystems (engl. *Manufacturing Execution System*, MES) kann jedoch von keiner vertikalen Integration gesprochen werden. Der Prototyp wird zwar lauffähig für ein Microsoft Betriebssystem gestaltet, die Anforderungen einer direkten Einbindung in das MES liegen jedoch außerhalb der Systemanforderungen.

Auch auf der zweiten Unterscheidungsdimension nach Pomberger & Blaschek (1996) bildet das in dieser Arbeit verwendete *Prototyping* eine Kombination aus den dort abgebildeten Varianten. So werden einzelne Problemlösungen für Suchalgorithmen und Nutzerdialog in explorativen Untersuchungen auf ihre Tauglichkeit getestet. Das Gesamtsystem lässt sich dagegen in Form eines voll funktionsfähigen Prototypen experimentellen Nutzertests unterziehen. Betrachtet man die Vorgehensweise der Implementierung von Nutzeroberfläche und Suchalgorithmus, kann jedoch auch von einem evolutionären *Prototyping* die Rede sein. Eine kontinuierliche Versionsverwaltung liefert die Möglichkeit regelmäßiger Überprüfungen und Korrekturen.

Am Ende der Entwicklung steht somit ein funktional ausgereifter Prototyp, ein sogenannter *High-Fidelity-Prototype*. Nur mit einem derart ganzheitlichen Ansatz lässt sich die Komplexität abbilden, welche den betrachteten Prozessen eigen ist. Außerdem wird ein Zugang zu professionellen Nutzern ermöglicht, deren Verfügbarkeit einerseits eingeschränkt und kostspielig ist, und denen andererseits klassische Erhebungsmethoden wie Papierprototypen oder Interviews zu aufwendig und nicht überzeugend erscheinen. Da Nutzer häufig von existierenden, ihnen bekannten Systemen mental geprägt sind, haben Entwickler Schwierigkeiten bei der verständlichen Präsentation neuer Lösungsansätze. Ein *High-Fidelity-Prototype* ermöglicht einen direkten und damit einfachen Zugang, unabhängig von Terminologie und Domänen-spezifischem Hintergrund der beteiligten Versuchspersonen. Beispiele prototypischer Teilimplementierungen von Mensch-Technik-Schnittstellen aus der Prozessindustrie finden sich bei Wittenberg (2004) sowie Komischke und Burmeister (2000). Ein weiterer Grund für die Entwicklung eines *High-Fidelity-Prototype* ist die Zielstellung, die zu entwerfende Technik (Suchalgorithmus) in einem möglichst realitätsnahen Nutzungskontext zu überprüfen.

1.2.2 Parallel-iterative Entwicklung

Bei der Gestaltung von neuartigen Arbeitssystemen kann es zu Konflikten zwischen Human-Factors-Experten und Automatisierungstechnikern hinsichtlich deren Design-Ziele kommen. So versuchen erstere oft, „... die Auslegung der technischen Systemkomponenten und der Arbeitsaufgaben so zu gestalten, dass, entsprechende Qualifikation vorausgesetzt, interessante, motivierende und beanspruchungsgerechte Arbeitstätigkeiten ermöglicht werden.“ (Urbas et al., 2006, S. 40). Ziel der Automatisierungstechniker dagegen ist häufig „... the intelligent systems revolution, as a result of which, it is now possible to have man-made systems that exhibit ability to reason, learn from experience and make rational decisions without human intervention.“ (Kaynak, 2005, S. 14). Zugespielt formuliert wird der Nutzer in der Auseinandersetzung zwischen Human-Factors-Experten und Automatisierungstechnikern entweder als zentrales Zielobjekt oder als vermeidbare Fehlerquelle gesehen. Im Sinne einer partnerschaftlich interdisziplinären Systemgestaltung empfehlen

Timpe und Kolrep (2000), Urbas und Timpe (2004) sowie Karrer et al. (2005) einen *Parallel-Iterativen Entwicklungsprozess (PIE)*. Dessen wesentlichen Phasen bestehen aus:

1. Systemanalyse
2. Systemdesign
3. Systembewertung und -korrektur

Der PIE bildet technische und menschliche Komponenten des zu entwickelnden Gesamtsystems in zwei parallel zu bearbeitenden Pfaden ab, welche in regelmäßigen Iterationen überprüft und abgeglichen werden. Die Pfade teilen sich auf in eine reine Technikentwicklung und eine Schnittstellenentwicklung. Während das Ursprungsmodell von Timpe und Kolrep (2000) als Life-Cycle Modell vorgestellt wurde, endet das hier angewandte Prozessmodell mit der Freigabe von lauffähigen Prototypen bereits vor der Serienproduktion und der Anwendung durch Endnutzer im Tagesgeschäft (Abbildung 1).

1.2.3 Schriftliche Ausarbeitung

Die vorliegende Dissertationsschrift orientiert sich in ihrer Gliederung an einem gekürzten und spezifizierten Modell des Parallel-Iterativen Entwicklungsprozess. Im einführenden Kapitel I.1 wurde bereits die Design-Idee vorgestellt und begründet. Im Kapitel II wird die Systemanalyse dokumentiert. Hauptaugenmerk liegt dabei auf die Funktions- und Aufgabenanalyse. Diese Analysen sind überwiegend theoretischer Natur und bilden den Stand der Technik zu den Methoden der Zeitreihenvergleiche sowie den Stand des Wissens zur Kurvennutzung in der Prozessführung ab. Abgeschlossen wird das Kapitel mit einer Zusammenfassung der Systemanalyse, welche die Grundlage für den Systementwurf im darauf folgenden Kapitel III liefert. Im weiteren Verlauf werden die Detailentwürfe des technischen Pfades (III.1 Suchalgorithmus) und des Pfades der Schnittstellenentwicklung (III.2 Nutzerdialog) vorgestellt. In beiden Fällen wird versucht, den iterativen Verlauf des Entwurfsprozesses darzustellen. Am Ende der Bewertung von Teilergebnissen und deren Integration steht ein voll funktionsfähiger Prototyp, welcher eine ganzheitliche Systembewertung (Kapitel IV) in Form von Nutzerexperimenten ermöglicht. Positive Aspekte der Bewer-

tung ermöglichen schlussendlich den Ausblick auf mögliche Praxisanwendungen, negative Aspekte zeigen den Bedarf an weiterer Forschung auf (V Systemkorrektur).

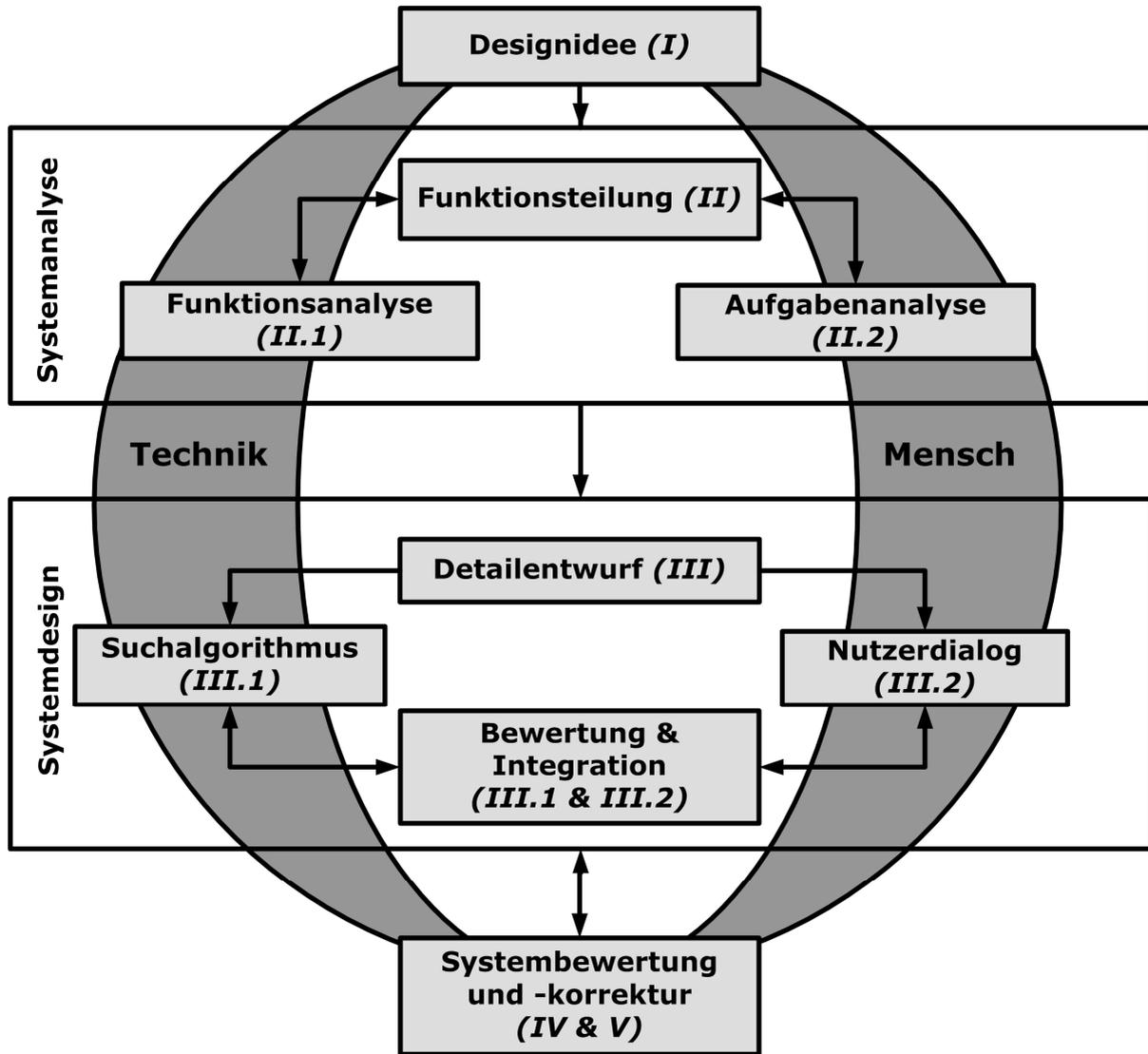


Abbildung 1 Entwicklungsmo­dell inklusive der Kapitelangaben für die einzelnen Entwicklungsphasen

II Systemanalyse

Im Rahmen der Systemanalyse wurde eine Funktionsteilung in iterativer Abstimmung mit detaillierten Analysen der einzelnen technischen Funktionen und menschlichen Aufgaben angestrebt. Am Anfang stand ein relativ grober Entwurf der Funktionsteilung (vgl. Tabelle 1 aus Kapitel I.1), der sich zum Teil an Heuristiken wie der sogenannte MABA-MABA Liste orientiert (Fitts, 1951). Dort werden beispielsweise Aussagen getroffen, wie: "men are better at: perceiving patterns of light or sound, ..., exercising judgment" und "machines are better at reasoning deductively" (Fitts, 1951, zitiert nach Sheridan, 2000, S. 204). Tabelle 1 zeigt ausgewählte Inhalte aus Fitts' MABA-MABA Liste, welche auf die Aufgabe der Trend basierten Datenbanksuche angepasst wurden.

Tabelle 2 Auf Kontext der Datenbanksuchmaschine angepasste MABA-MABA Liste

Mensch	Technik
Mustererkennung	Kurze Antwortzeiten
Improvisieren	Schnelle Informationsspeicherung
Genauigkeit im Abrufen von Informationen aus dem Langzeitspeicher	Vollständiges Löschen von Informationen
Induktives Schließen	Deduktives Schließen
Entscheiden unter Unsicherheiten	

Sheridan (1997) empfiehlt als Basis der Funktionsallokation die Definition eines Lösungsraums. Darin stellen die Vollautomatisierung und die Nullautomatisierung die zwei Extremlösungen dar. Zur Vollautomatisierung der Fehlererkennung anhand von dynamischen Verlaufsmustern haben Forscher wie Vedam und Venkatasubramanian (1997) sowie Haimowitz und Kohane (1993)

künstliche Intelligenz in Form von neuronalen Netzen und Fuzzy-Logik verwendet. Das genaue Gegenteil wird in den meisten der herkömmlichen Prozessleitwarten praktiziert. Dort obliegt die Störungssuche und -analyse den Anlagenfahrern, ohne diesen dafür adäquate technische Hilfsmittel zur Verfügung zu stellen.

Die Trend basierte Datenbanksuchmaschine ist zwischen den beschriebenen Extremen einzuordnen. Das Wissen der Prozessexperten wird dabei während der direkten Aufgabenbearbeitung genutzt, statt es im Vorfeld zu erheben und in ein rein technisches System zu integrieren. Nach der Einteilung von Sheridan (1997) lässt sich die Suchmaschine dem Konzept des *Trading* zuordnen. Mensch und Computer verhandeln danach eine Aufgabe, indem sie einzelne Unteraufgaben bearbeiten und sobald diese erfolgreich beendet wurde, die Hauptaufgabe an den jeweils anderen Akteur übergeben. Im Kontrast dazu steht das Konzept des *Sharing*, wobei beide Akteure dieselbe Unteraufgabe ausführen und die Ergebnisse (Entscheidungen, Handlungen) miteinander vergleichen.

Innerhalb der folgenden beiden Kapitel sollen die Potentiale und Beschränkungen beider Subsysteme – Mensch & Technik – verdeutlicht werden. Am Ende wird ein Interaktionsmodell für das iterative *Trading* während der gemeinsamen Aufgabe *Kurvensuche* abgeleitet und vorgestellt.

II.1 Funktionsanalyse – Stand der Technik

Um die Entwicklung einer Trend basierten Datenbank-Suchmaschine zu ermöglichen, wurde auf geeignete Methoden der Mustervergleiche von Zeitreihen zurück gegriffen (Kapitel II.1.2). Da verschiedene Prozessdynamiken in unterschiedlichen Frequenzspektren der Rohdaten-Zeitreihe enthalten sind (vgl. Abbildung 2), wurde zudem nach effektiven Transformationsmethoden gesucht (Kapitel II.1.1). Die recherchierten Verfahren werden primär an naturwissenschaftlich-mathematischen Fakultäten der weltweiten Forschergemeinde entwickelt. Kapitel II.1.3 beleuchtet die Anwendung derartiger Verfahren für verschiedene Automatisierungstechniken der Verfahrenstechnik. Anwendungen im Sinne einer Nutzer zentrierten Suchmaschine sind in dieser Domäne jedoch nicht auffindbar. Ein Blick in eine Reihe von Fremddomänen schließt deshalb das Kapitel der Funktionsanalyse ab.

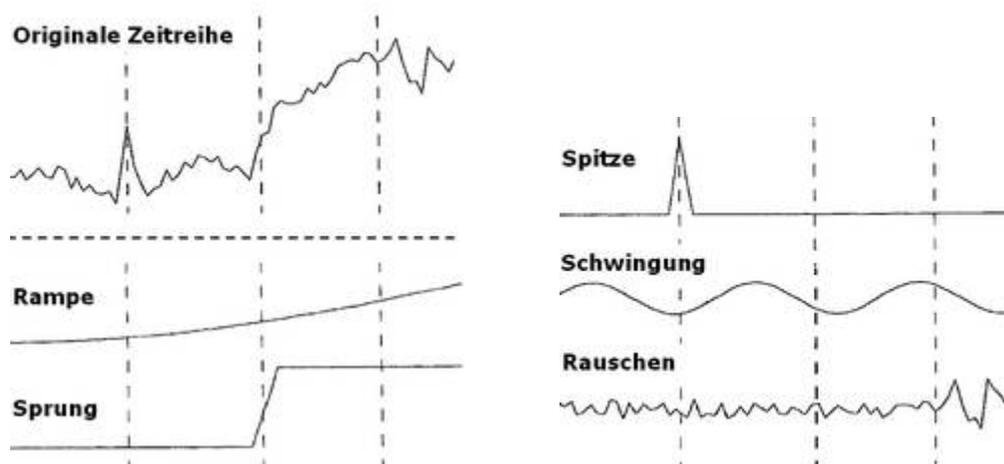


Abbildung 2 Komponenten von Prozesssignalen in Kurvenform (entnommen aus Bakshi et al., 1994, S. 180)

11.1.1 Transformation von Zeitreihen

Die wohl bekannteste Form der Dekomposition einer Zeitreihe in deren Frequenzbereiche ist die *Fourier-Transformation* (Agrawal et al. 1993). Gleichung 1 zeigt die kontinuierliche Fourier-Transformation. $F(\omega)$ gibt dabei die Amplitude einer jeden Frequenz $\omega \in \mathbb{R}$ aus der Originalfunktion $f(t)$ an.

Gleichung 1
$$F(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) * e^{i\omega t} dt$$

Das moderne Verfahren der *Wavelet-Transformation* (Gleichung 2)^a hat jedoch gegenüber der *Fourier-Transformation* den Vorteil der zeitlichen Lokalität der Basisfunktionen sowie der geringeren Komplexität (Shahabi et al., 2000; Wang & Wang, 2000). Um die Informationen der originalen Zeitreihe auf verschiedenen Frequenzbändern abzubilden, wird die Transformation auf unterschiedlichen Skalen σ durchgeführt. Mallat und Zhong (1992) schlagen eine dyadische Abstufung der Skalen vor (Gleichung 3).

Gleichung 2
$$f_{\sigma,\tau}(t) = \frac{1}{\sqrt{\sigma}} \int_{-\infty}^{+\infty} f(t) \Psi_{\sigma}(t - \tau) dt$$

Gleichung 3
$$\sigma = 2^n \text{ mit } n \in \mathbb{N}^+$$

Für unterschiedliche Anwendungen werden verschiedenste *Mother-Wavelets* verwendet (Abbildung 3). Alle Mother-Wavelets verhalten sich asymptotisch entlang der Abszisse und bilden ein konstantes, d. h. von ihrer Breite und Höhe unabhängiges Integral.

^a Der Faktor vor dem Integral dient der Normierung.

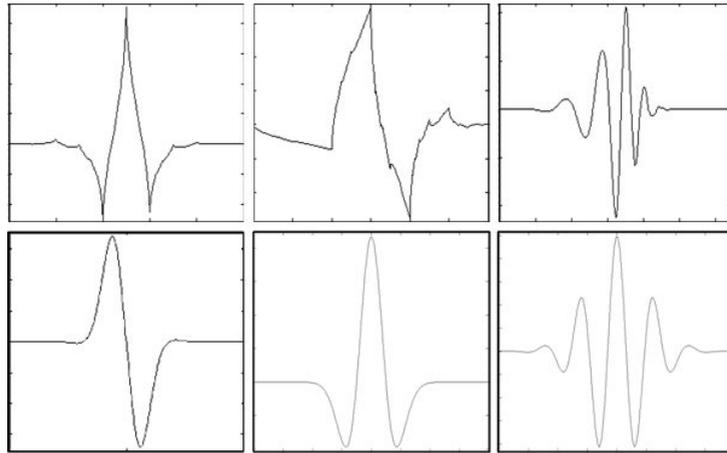


Abbildung 3 Darstellung verschiedener Mother-Wavelets (Frey, 2006, S. 17), von links oben nach rechts unten: Coiflet; Daubechies 2; Daubechies 7; 1. Ableitung der Gaußfunktion; Mexican Hat; Morlet ABB115

Die Wavelet-Transformation ist ein Spezialfall der *Faltung* (eng. *convolution*). Diese wird allgemeiner als die Verknüpfung zweier beliebiger Funktionen verstanden (Gleichung 4)^a. Die Faltung ist eine Operation, welche die Funktion $g(t)$ anhand des Translationsparameters τ über die Funktion $f(t)$ „schiebt“ (Abbildung 4). Als Integrationsbereich wird der komplette Definitionsbereich D beider Funktionen genutzt. Um die Unabhängigkeit des Ergebnisses vom Startzeitpunkt der Transformation (Translationsinvarianz) zu gewährleisten, wird eine kontinuierliche Form der Faltung benötigt (Bakshi & Stephanopoulos, 1994). Dabei wird mit der minimal möglichen Schrittweite für den Translationsparameter τ gearbeitet. Da somit $\tau = \{1, \dots, n\}$, kann die vereinfachte Schreibweise aus Gleichung 5 genutzt werden.

Gleichung 4
$$f_{\sigma, \tau}(t) = \int_D f(\tau) g_{\sigma}(t - \tau) d\tau$$

Gleichung 5
$$f_{\sigma}(t) = \int_D f(\tau) g_{\sigma}(t - \tau) d\tau$$

^a Wird die Faltung wie in Abbildung 5 zur Glättung des Originalsignals genutzt, kann auf den Normierungsfaktor vor dem Integral verzichtet werden.

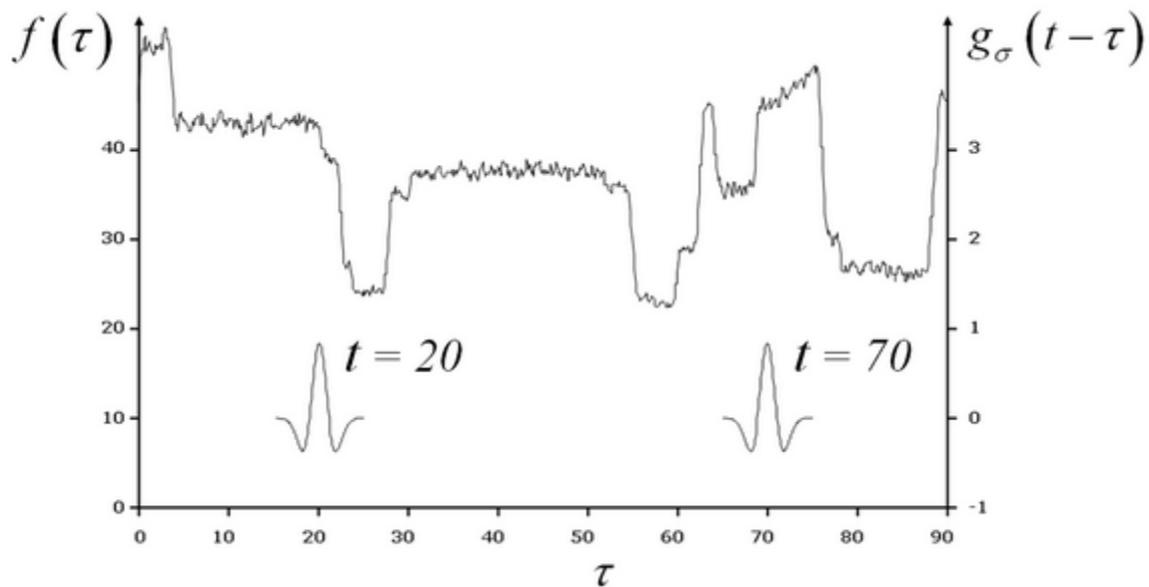


Abbildung 4 Visualisierung des Translationsparameters τ

Witkin (1983) hat die Faltung mit Gaußfunktionen (Gleichung 6)^a unterschiedlicher Standardabweichung^b untersucht. Dabei entsteht eine Abfolge von immer stärker geglätteten Zeitreihen. Die geglätteten bzw. gefalteten Zeitreihen werden auf verschiedenen Faltungsskalen abgebildet (vgl. Abbildung 5).

Gleichung 6
$$g_{\sigma}(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2\sigma^2}x^2}$$

Die Wahl der Gaußfunktion erfüllt einige der Anforderungen^c, die für die Multiskalenanalyse gestellt werden:

- Vollständige („lückenlose“) Abdeckung des gesamten Zeit- und Frequenzraum
- Stabilität und Vollständigkeit bei der Rekonstruktion
- minimale Informationsredundanz zwischen den Faltungsskalen (Skalenunabhängigkeit)
- effiziente Numerik bei Analyse und Merkmalsextraktion

^a Der Faktor vor dem Integral dient der Normierung.

^b Die Standardabweichung der Normalverteilung ist als Skalierungsparameter zu verstehen.

^c Die Punkte 1 und 2 sind dabei konfliktär zu den Punkten 3 und 4. Ein optimaler Kompromiss muss gefunden werden.

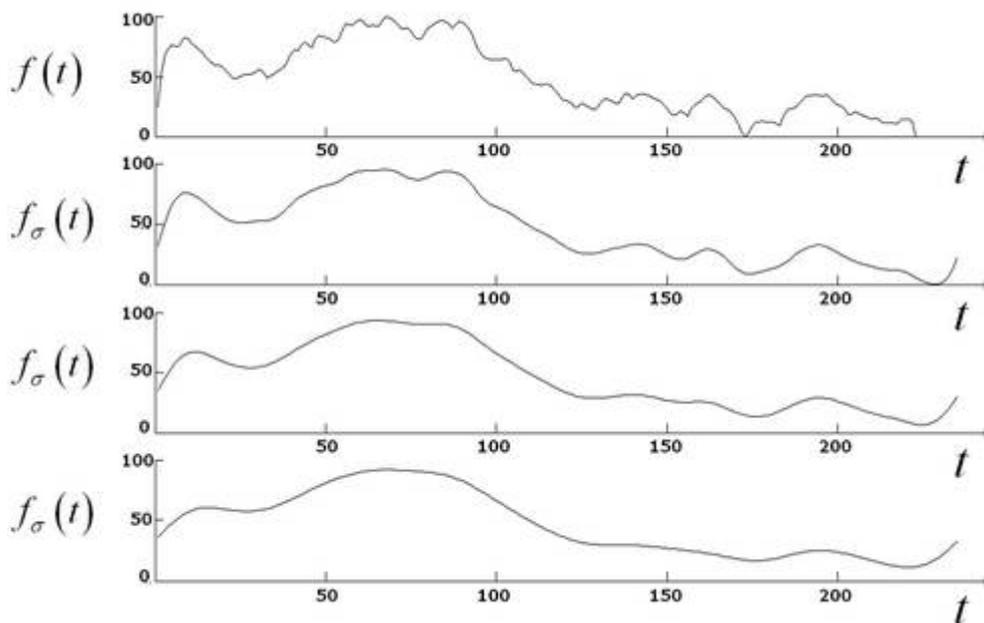


Abbildung 5 originale Zeitreihe (oben) und die entsprechenden Approximationen auf den verschiedenen Faltungsskalen. Werte von Sigma von oben nach unten: $\sigma = \{3, 6, 9\}$

Abbildung 5 macht zudem deutlich, dass die Faltung (Gleichung 4) einer beliebigen Zeitreihe mit der Gaußfunktion (Gleichung 6) eine gute visuelle Approximation der ursprünglichen Zeitreihe darstellt, und zwar über alle Faltungsskalen hinweg. Als Resultat der Faltung erhält man pro Faltungsskala^a σ eine unendlich oft differenzierbare Funktion $f_{\sigma}(t)$. Die erste und zweite Ableitung sowie deren markante Punkte (Extremstellen & Nullstellen) zeigt Abbildung 6.

Das so genannte *Mexican-Hat-Wavelet* (MHW) ist proportional zu der zweiten Ableitung der Gaußfunktion (Abbildung 3). Bei einer Wavelet-Transformation mit dem MHW, bilden die Wavelet-Transformierten die Krümmungsstärke der originalen Zeitreihe ab (Abbildung 6). Je größer die Amplitude, desto stärker ist zu diesem Zeitpunkt die Krümmung der geglätteten Zeitreihe. Man kann die Amplituden der Wavelet-Transformierten auch als Indikation der Ähnlichkeit zwischen Zeitreihe und MHW zum Zeitpunkt t verstehen. Eine Amplitude von 1 entspricht dabei einer vollständigen Überlappung, bei -1 ist die Zeitreihe die exakte Inversion des MHW.

^a Der Skalenindikator σ entspricht hier der Standardabweichung der Gaußfunktion.

Durch die Wavelet-Transformation erhöht sich die Anzahl an Datenpunkten um den Faktor σ . Für eine effiziente Datenreduktion werden deshalb lediglich die bedeutungsvollen Datenpunkte genutzt. Mallat und Zhong (1992) zeigten unter Verwendung des *Mexican-Hat-Wavelet*, dass auf Basis der lokalen Extremstellen der Wavelet-Transformierten eine originale Zeitreihe fast vollständig rekonstruiert werden kann. Da die Extremstellen über alle Wavelet-Skalen teilweise große Redundanzen aufweisen, können diese zu E-Ketten (Extremstellen-Ketten) verknüpft werden. Zudem lässt die Länge einer E-Kette auf die Stabilität der zugrundeliegenden Dynamik schließen. Abbildung 7 zeigt sehr anschaulich, welche Information in einer stabilen, d.h. über viele Wavelet-Skalen verlaufenden E-Kette stecken.

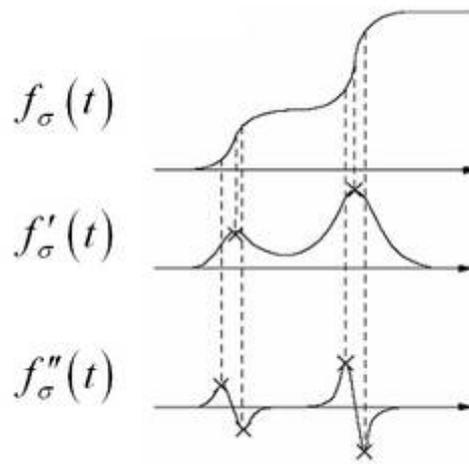


Abbildung 6 Zeitreihe mit korrespondierenden Maxima und Minima der 1. und 2. Ableitung

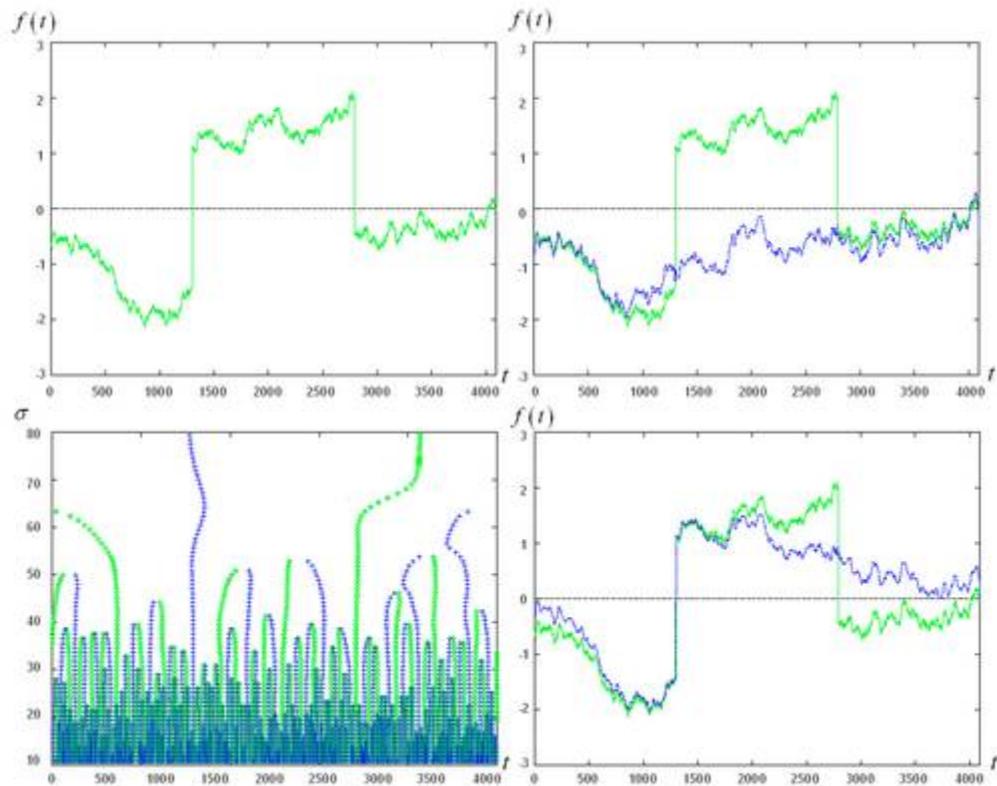


Abbildung 7 Entnommen aus Struzik und Siebes (1998, S. 10).

Oben links: Originalsignal

Unten links: E-Ketten über alle Wavelet-Skalen

Oben rechts: Rekonstruktion (blaue Linie) ohne die beiden längsten E-Ketten

Unten rechts: Rekonstruktion (blaue Linie) ohne die zweite (grüne) der beiden längsten E-Ketten

11.1.2 Methoden für den Vergleich von Zeitreihen

In einem Überblicksartikel von Keogh und Kasetty (2003) werden vier verschiedene Arten von Vergleichen zwischen Zeitreihen beschrieben:

- Segmentieren (engl. *segmentation*): Konstruiere das Modell einer beliebigen Zeitreihe aus einer Anzahl an Segmenten, welche deutlich geringer ist, als die Anzahl an Datenpunkten der Zeitreihe. Das Modell sollte annähernd die originale Zeitreihe abbilden.
- Indexieren (engl. *indexing*): Ordne die Zeitreihen einer Datenbank nach ihrer Ähnlichkeit mit einer Referenz-Zeitreihe. Nutze dazu bestimmte Ähnlichkeitsmaße.
- Gruppieren (engl. *clustering*): Finde Gruppen von Zeitreihen einer Datenbank. Die Elemente einer Gruppe sollen sich hinsichtlich bestimmter Kriterien möglichst ähnlich sein. Die Elemente verschiedener Gruppen sollen sich im Bezug auf diese Kriterien so weit möglich unterscheiden.
- Klassifizieren (engl. *classification*): Ordne eine beliebige Zeitreihe einer aus zwei oder mehreren vordefinierten Klassen zu.

Bei der in dieser Arbeit verfolgten Prototypenentwicklung wurden die Verfahren des Segmentierens und Indexierens umgesetzt, welche im Folgenden näher erläutert werden sollen. Die Methoden des Gruppierens und Klassifizierens werden als vielversprechend für weitere Anwendungsfälle erachtet, sollten jedoch erst nach einer erfolgreichen Erprobung der ersten beiden Verfahren verfolgt werden (vgl. Kapitel V).

11.1.2.1 Segmentieren

Das Segmentieren von Zeitreihen führt zu einer Menge an zeitlich definierten Episoden, deren Anzahl kleiner ist als die Anzahl an Datenpunkten der originalen Zeitreihe (Keogh & Kasetty, 2003). Dadurch wird die Effizienz Computer gestützter Datenspeicherung, -übertragung und -verarbeitung deutlich erhöht. Das Segmentieren kann auf Basis von Polynomen beliebiger Ordnung geschehen. Die Linearapproximation (Abbildung 8) stellt dabei die am häufigsten durchgeführte Methode dar.

Aber auch signifikante Punkte der klassischen Kurvendiskussion (Nullstellen der ersten und zweiten Ableitung) können als Segmentierungshilfen genutzt werden (vgl. Kapitel II.1.3.2). In Anlehnung an Mallat und Zhong (1992) kann auch auf die Nullstellen der dritten Ableitung (Punkte maximaler Krümmung in Originalzeitreihe) zurück gegriffen werden.

Die verschiedenen Vorgehensweisen beim Segmentieren können in drei große Gruppen geordnet werden:

- *sliding windows*. Episode wird entlang der Zeitachse schrittweise bis zu dem Erreichen eines Abbruchkriteriums generiert. Das folgende Segment beginnt beim Datenpunkt, welcher einen Schritt nach dem letzten Datenpunkt der zuvor gebildeten Episode folgt.
- *top down*. Die gesamte Zeitreihe bzw. deren Episoden werden schrittweise bis zu dem Erreichen eines Abbruchkriteriums halbiert.
- *bottom up*. Ausgehend von den kleinstmöglichen Segmenten, werden diese schrittweise bis zu dem Erreichen eines Abbruchkriteriums zusammen geführt.



Abbildung 8 Originale Zeitreihe und deren Linearapproximation (entnommen aus Keogh & Kasetty, 2003, S. 356)

II.1.2.2 Indexieren

Das einfachste Maß für die Ähnlichkeit zweier Zeitreihen $x = (x_1, \dots, x_n)$ und $y = (y_1, \dots, y_n)$ mit n Datenpunkten ist das euklidische Abstandsmaß d .

Gleichung 7
$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

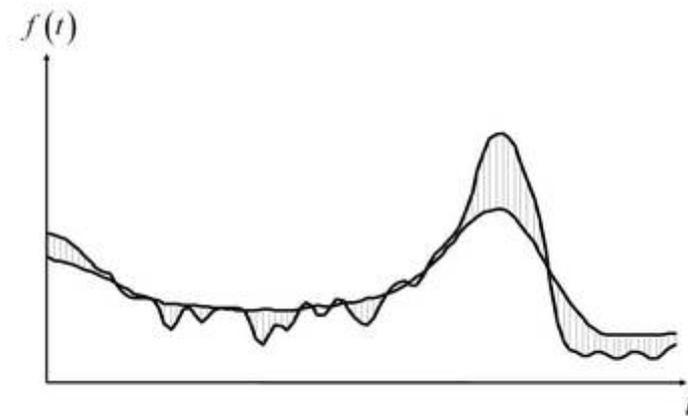


Abbildung 9 Euklidisches Abstandsmaß
(entnommen aus Keogh et al., 2001, S. 153)

Bei großen Datenmengen wird diese Form des Vergleichs jedoch rechentechnisch recht aufwendig, wenn auch mit einer linearen Komplexität $O(n)$. Der Aufwand verstärkt sich, sobald die Grenzen der Zeitreihenfenster nicht festlegbar sind. Ein Beispiel für den Vergleich fester Zeitfenster ist der Vergleich von Zeitreihen verschiedener Jahre. Die Grenzen fallen weg, sobald eine Zeitreihe von fünf Stunden mit jedem beliebigen Zeitabschnitt eines bestimmten Jahres verglichen werden soll, unabhängig von Tageszeiten oder Wochentagen. Die Komplexität erhöht sich in diesem Fall auf $O(n * k)$, wobei k die Anzahl der Zeitschritte innerhalb der fünf Stunden darstellt. Als Möglichkeit zur Komplexitätsreduktion kann auf die Methoden des Segmentierens von Zeitreihen zurück gegriffen werden.

Dennoch wird für die Bewertung der so entstandenen Datenmodelle oft auf das euklidische Abstandsmaß zurück gegriffen. Keogh et al. (2001) nennen als ein Kriterium für den Vergleich der

verschiedenen Methoden das Verhältnis zwischen dem geschätzten Abstand zweier transformierter Zeitreihen und dem euklidischen Abstand der korrespondierenden originalen Zeitreihen. Ein Verhältnis nahe Eins lässt auf eine äußerst genaue Methode schließen. Die Autoren zeigen in experimentellen Untersuchungen, dass qualitative Unterschiede in den Implementierungen (*implementation bias*) der vorgeschlagenen Algorithmen sowie in den angewandten Datensätzen (*data bias*) einen großen Einfluss auf die Ergebnislage hat, welche mit den unterschiedlichen Transformationsmethoden erreicht werden können.

„Implementation bias is the conscious or unconscious disparity in the quality of implementation of a proposed approach, vs. the quality of implementation of the competing approaches.“ (Keogh et al., 2001, S. 3)

„Data bias is the conscious or unconscious use of a particular set of testing data to confirm a desired finding.“ (Keogh et al., 2001, S. 4)

Dies lässt die Schlussfolgerung zu, dass „objektive“ externe Evaluationen von Recheneffizienz und Genauigkeit an fehlenden Benchmarks scheitern und „subjektive“ interne Evaluationen immer im Kontext des Implementierungs- und Datenbias zu betrachten sind.

11.1.3 Anwendungen von Zeitreihenvergleichen

Im folgenden Kapitel werden einige Beispiele für die praktische Anwendung von Zeitreihenvergleichen vorgestellt. Dabei wird sich zwar die Domäne der Prozessführung hauptsächlich, jedoch nicht ausschließlich fokussiert. Zum einen soll damit die Relevanz der Methoden für die Prozessüberwachung und -analyse verdeutlicht werden. Zum anderen zeigen die Beispiele Ansätze auf, welche vor dem Hintergrund der Entwicklung des geplanten Suchalgorithmus von Interesse sind.

11.1.3.1 Zustandsdiagnose

Flehmg et al. (1998) und Alsmeyer (2005) haben ein Werkzeug entwickelt, welches die Identifikation stationärer und linearer Zustände anhand eines Segmentierungsalgorithmus ermöglicht. Derartige Zustände sind vor allem für die Regelungstechnik und die Prozessanalyse von Bedeutung.

Die Autoren nutzen die Wavelet-Transformation zur Rauschreduktion und Polynome zur Identifikation einzelner Kurvenabschnitte. Liegen die Koeffizienten dieser Polynome unter gewissen Schwellwerten, werden die Segmente als stationär oder linear identifiziert (vgl. Abbildung 10). Dabei werden k fixe Zeitintervalle verschiedener Ausdehnung l betrachtet. Die Auflösung dieser Ausdehnungen wurde in Analogie zur Multiskalenanalyse von Mallat und Zhong (1992) in dyadischen Schrittweiten variiert.

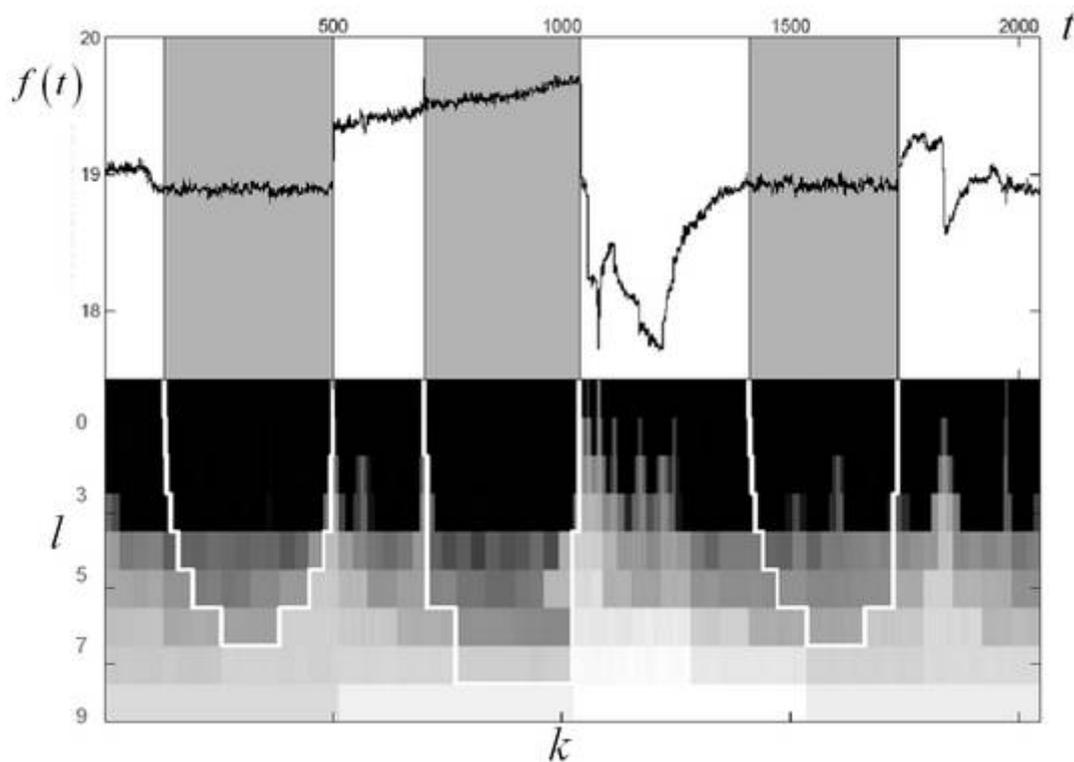


Abbildung 10 Identifikation linearer Episoden (entnommen aus Flehmig, 1998, S. 496). Oben: Originale Zeitreihe mit identifizierten linearen Episoden (grau). Unten: dyadisch skalierte Zeitintervalle mit Farb-Code: hell = großer Approximationsfehler, dunkel = kleiner Approximationsfehler

II.1.3.2 Fehlerdiagnose

Verschiedene Wissenschaftler beschäftigt sich seit beinahe zwei Jahrzehnten mit der automatischen Erkennung von Prozess-Anomalitäten, Verschiebung von Arbeitspunkten u. ä. (Janusz & Venkatasubramanian, 1991; Maurya et al., 2007; Rengaswamy & Venkatasubramanian, 1995; Vedam &

Venkatasubramanian, 1997). Ziel ist die Entlastung der Anlagenfahrer durch Überwachungssysteme basierende auf künstlicher Intelligenz. Mithilfe qualitativer Trendanalysen sollen bedeutungsvolle Muster in Mess-Signalen identifiziert und interpretiert werden. Im Wesentlichen modellieren bzw. segmentieren die entworfenen Verfahren die Dynamiken bzw. Trends von Zeitreihen und ordnen die generierten Modelle bekannten Fehlerklassen zu.

Der Segmentierungsalgorithmus arbeitet dabei mit Merkmalen der klassischen Kurvendiskussion. Zur Beschreibung dynamischer Charakteristika von Zeitreihen werden diese in Episoden unterteilt. Nach Cheung und Stephanopoulos (1990) beginnt und endet eine Episode mit der Änderung des Vorzeichens der ersten oder zweiten Ableitung (vgl. Abbildung 11). Bei gekrümmten Episoden entspricht dies den Extremstellen bzw. Wendestellen. Die erste Ableitung einer Extremstelle und die zweite Ableitung einer Wendestelle sind gleich Null, und werden darum auch Nullstellen erster und zweiter Ordnung genannt. Sind die erste oder zweite Ableitung einer Zeitreihe über einen bestimmten Zeitbereich konstant Null, so werden diese als lineare Episoden definiert. Ein Toleranzband um den Wert Null ist denkbar und bei verrauschten Daten zwingend notwendig. Der komplexen Dynamik von verfahrenstechnischen Prozessgrößen mit der Verwendung der Wavelet-Transformation Rechnung getragen. Der relevante Frequenzbereich wird dabei anhand von Neuronalen Netzwerken, also künstlicher Intelligenz identifiziert. Das Ergebnis besteht in einer Zuordnung aktueller Trends zu den bekannten Fehlern einer Fehlerdatenbank.

Kritisch zu bewerten ist das Paradigma von automatisierten Fehlererkennungssystemen jedoch vor dem Hintergrund sich ständig wandelnder und entwickelnder Prozessanlagen. Für Probleme der Aktualität der Fehlerdatenbank und des Wissensmanagement müssen Lösungen gefunden werden. Ein alternatives Paradigma, welches auch in dieser Arbeit verfolgt wurde, vertraut auf die Flexibilität und Lernfähigkeit von gut ausgebildeten Prozessexperten. Diese verfügen über eine gewachsene und vertiefte Expertise im Kurvenlesen (vgl. Kapitel II.2.3). Über deren Expertise in der Beurteilung von modernen, auf künstlicher Intelligenz basierenden Fehlererkennungssystemen gibt es jedoch bisher keine optimistisch stimmenden Studien.

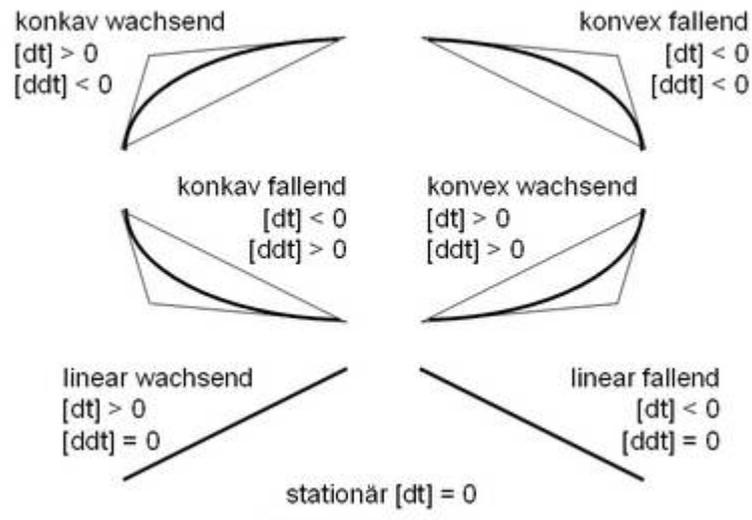


Abbildung 11 Trendlet-Alphabet nach Cheung und Stephanopoulos (1990, S. 504)

II.1.3.3 Datenbanksuche

Die Erforschung und Entwicklung von Nutzer zentrierten Suchanfragen für Werkzeuge des Zeitreihenvergleichs wird primär in der Finanzdomäne verfolgt. Als typische Aufgabenszenarien werden von Wattenberg (2001) u. a. folgende Beispiele genannt.

- Welche Technologieaktien sind im April preislich gestiegen?
- Welche Aktien sind über die vergangenen Jahre dramatisch gefallen und beginnen nun seit kurzem wieder zu steigen?

Eine Beantwortung dieser Fragen soll durch das Java Applet *QuerySketch* unterstützt werden (Wattenberg, 2001). Dieses macht das Anfertigen einer Freihandskizze zur Spezifikation der Suchanfrage möglich. Die Ähnlichkeitsschätzung wird dann auf Basis des Euklidischen Abstandsmaßes vorgenommen. Fraglich bleibt, inwieweit die schematische Natur einer Freihandskizze mit der detailgetreuen Berechnung des Euklidischen Abstandsmaßes harmonisiert. Zumindest erfordert dieses Dialogkonzept eine hohe Form von Genauigkeit in der Linienführung beim Skizzieren. Eine andere Möglichkeit ist die Nutzer zentrierte Spezifikation einer Suchanfrage anhand von bereits vorhandenen Beispiel- bzw. Referenzkurven (Buono et al., 2004).

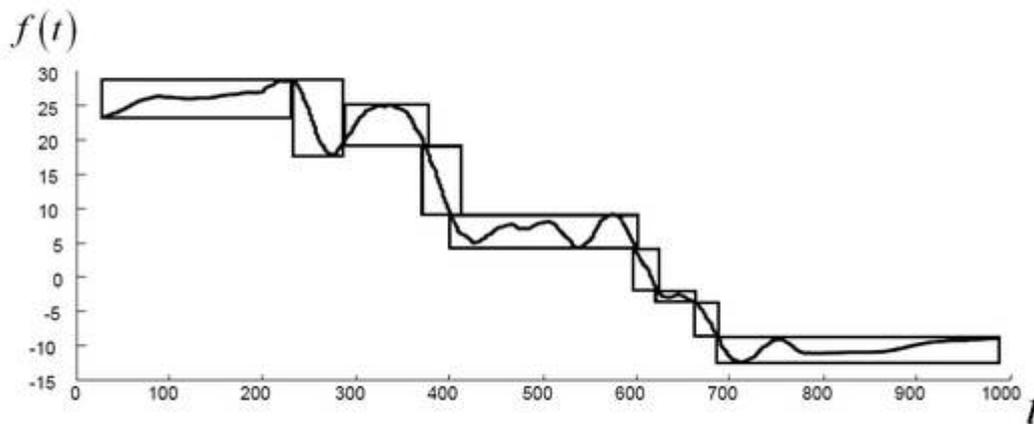


Abbildung 12 Beispiel für Trendbeschreibung einer graphisch unterstützten Suchmaschine für Zeitreihen (entnommen aus Keogh et al., 2002, S. 245)

Um jedoch die Unabhängigkeit von existierenden Referenzkurven zu gewährleisten, bieten sich stärker abstrahierte Suchanfragen an, wie sie zum Beispiel von Hochheiser & Shneiderman (2002) und Keogh et al. (2002) in Form von Zeitfenstern vorgeschlagen werden. Ein Zeitfenster (eng. *timebox*) wird durch die Punkte (x_{\min}, y_{\min}) und (x_{\max}, y_{\max}) definiert (Gleichung 8 und Gleichung 9). Die Selektion von Ergebniszeitreihen erfolgt durch die Kombination von aufeinander folgenden Zeitfenstern (Abbildung 12). Diese dürfen überlappen oder Abstände zueinander aufweisen.

Gleichung 8
$$x_{\min} \leq x \leq x_{\max}$$

Gleichung 9
$$y_{\min} \leq y \leq y_{\max}$$

Eine alternative Erweiterung der Zeitfenster wurde von Hochheiser und Shneiderman (2004) mit der Anstiegsspezifikation einer Referenz-Episode vorgestellt. Dabei wird von einem Nutzer für ein bestimmtes Zeitintervall (t_{\min}, t_{\max}) ein Toleranzband an Anstiegen mit den Winkeln $(\alpha_{\min}, \alpha_{\max})$ festgelegt (vgl. Abbildung 13). Die Winkelbedingung wird für jeden Zeitschritt des ausgewählten Zeitintervalls überprüft. Sowie ein Teilstück außerhalb des spezifizierten Anstiegs liegt, wird die entsprechende Zeitreihe als Ergebnis verworfen. Je größer jedoch das Messrauschen im Verhältnis

zur Anstiegshöhe ist und je größer die Anzahl an Zeitschrittweiten je Episode^a, desto anfälliger wird diese Bedingungsvariation.

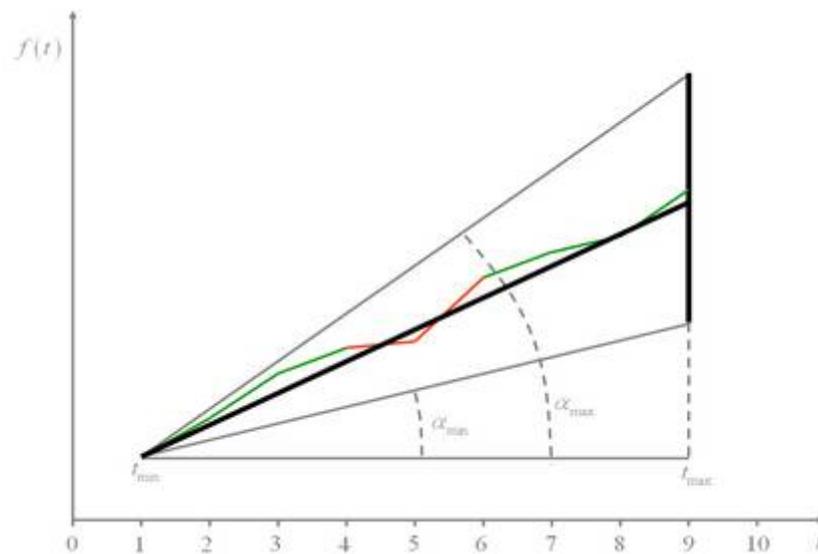


Abbildung 13 Episodenbeschreibung auf Basis einer Winkelbedingung. Grün: Zeitintervalle innerhalb der Toleranzen. Rot: Zeitintervalle außerhalb der Toleranzen

Sowohl die Dialogmetapher der Zeitfenster als auch die der Winkelbedingung werden primär aus mathematisch analytischen Überlegungen heraus begründet. Die Autoren geben keinerlei Hinweise, ob sie etwa auf Modellen der Wahrnehmung und Interpretation von Zeitreihen durch die Domänenexperten basieren. Hochheiser (2003) leitet analytisch zwar eine umfangreiche Sammlung an hypothetischen Aufgabenstellungen ab. Jedoch empfiehlt er für zukünftige Arbeiten eine intensive Überprüfung der Praxisrelevanz aller bereitgestellten Funktionalitäten. Zwei weitere Punkte führen zudem zu einer Einschränkung für die Anwendung der hier aufgeführten Entwürfe und Ansätze auf komplexe und unabhängig von festen Zeitfenstern auftretende Dynamiken der Prozessführung. Zum Ersten fehlt eine Datenreduktion im Sinne von Indexierung oder Segmentierung in der Vorverarbeitung. Zum Zweiten fehlt eine Betrachtung der Rohdaten auf verschiedenen Frequenzbändern.

^a Die Autoren bieten ihr Werkzeug für die Finanzdomäne an. Dort ist mit maximalen Zeitschrittweiten von $N = 12$ (Jahreszyklus) bis $N = 30$ (Monatszyklus) zu rechnen.

11.2 Aufgabenanalyse – Stand des Wissens

Parallel zu der detaillierten Funktionsanalyse wurde eine Aufgabenanalyse durchgeführt, welche sich aus analytischen und empirischen Teilen zusammensetzte. Zunächst wurde die Verwendung von Kurvenbildern bei der Prozessüberwachung und -analyse recherchiert (Kapitel 0). Dabei wurde die Prozessbeobachtung als Voraussetzung für den unmittelbaren, zumeist regelbasierten Eingriff aus der prozesstechnischen Leitwarte verstanden. Die Prozessanalyse wurde dagegen als wissensbasierte Tätigkeit definiert, die auch außerhalb der Leitwarte stattfindet und deren Resultate oft mittelfristige Konsequenzen für die verfahrenstechnische Prozessführung haben.

Im Rahmen einer Interviewstudie wurde dann der technische Unterstützungsbedarf in diesen Arbeitssituationen erfragt (Kapitel 11.2.2). Um die kognitiven Fähigkeiten und das domain spezifische Vorwissen zu definieren, über welches die potentielle Nutzergruppe für die Interaktion der zu entwickelnden Unterstützungskomponenten verfügt, wurde vor allem auf Vorarbeiten von Kindsmüller (2006) zurückgegriffen (Kapitel 11.2.3). Zur Spezifikation des Systementwurfs wurde der Ansatz der Szenario basierten Anforderungsanalyse verfolgt (Kapitel 11.2.4), wie ihn unter anderem Sutcliffe (2003) vorschlägt. Ganz im Sinne des Parallel-Iterativen Entwicklungsprozess wurde die Aufgabenanalyse zu einem Zeitpunkt wiederholt, zu dem bereits ein erster Prototyp des Nutzerdialogs vorlag. Erst dann war es möglich, gemeinsam mit potentiellen Nutzern detaillierte Design-Entscheidungen zu reflektieren und zu überprüfen (11.2.5).

Tabelle 3 Überblick über die Schritte der Aufgabenanalyse

Kapitel	Fragestellung
<p>0</p> <p>Das Kurvenbild als Werkzeug der Prozessüberwachung und -analyse</p>	<p>Für welche Aufgaben der Prozessüberwachung und –analyse nutzen Experten die derzeitigen technischen Umsetzungen von Kurvenbildern?</p>
<p>II.2.2</p> <p>Technischer Unterstützungsbedarf</p>	<p>Für welche Aufgaben der Prozessüberwachung und –analyse brauchen Prozessexperten zusätzliche Unterstützung?</p>
<p>II.2.3</p> <p>Nutzerexpertise bei der Prozessüberwachung und -analyse</p>	<p>Über welche kognitiven und wissensbasierten Voraussetzungen für die Aufgabenbewältigung verfügen die Prozessexperten?</p>
<p>II.2.4</p> <p>Entwurf potentieller Aufgabenszenarien</p>	<p>Wie soll der Experte bei der Bewältigung konkreter Aufgaben durch das zu entwerfende System unterstützt werden?</p>
<p>II.2.5</p> <p>Aufgabenanalyse anhand einer Prototypen-Evaluation</p>	<p>Welche zusätzlichen Aufgaben entstehen für den Nutzer aufgrund des neuartigen technischen Unterstützungssystems?</p>

11.2.1 *Das Kurvenbild als Werkzeug der Prozessüberwachung und -analyse*

Dem Kurvenbild kommt neben den übrigen graphischen und alphanumerischen Darstellungsformen von Prozesszuständen eine besondere Bedeutung zu. Erfahrene Prozessexperten erleben das Kurvenbild oft als „das Fenster zum Prozess“ (Krick, 2003, S. 22). Das Kurvenbild unterstützt Prozessingenieure wie Operateure bei Fragen wie:

- Ist ein stationärer Zustand erreicht?
- Wo kommt der Prozess her?
- Wie wird sich der Prozess entwickeln?

Antworten auf diese Fragen können bei geeigneter Wahl der Darstellung häufig direkt aus dem Kurvenbild abgelesen werden (Wandke, 1997). Um die Bedeutung von Kurvenbildern bei der Prozessüberwachung und -analyse nochmals im Kreis potentieller Nutzer zu bestätigen, wurde eine Interviewstudie durchgeführt, die im Folgenden kurz dargestellt werden sollen.

11.2.1.1 Feldstudie

Experten aus dem Bereich Forschung und Entwicklung der BASF AG Ludwigshafen (N = 7) und des Fachgebiets Dynamik und Betrieb technischer Anlagen der Technischen Universität Berlin (N = 2) wurden in Einzelsitzungen zur Online- und Offline-Nutzung von Kurvenbildern bei der Prozessüberwachung und -analyse befragt. Dies erfolgte anhand von strukturierten Interviews, Bewertungsskalen und Dominanz-Paarvergleichen (Bortz & Döring, 2006, S. 157 u. f.; vgl. Anhang A.1.1.2). Die Auswertung der Gesprächsanteile erfolgte anhand von qualitativen Inhaltsanalysen der Interview-Transkripte.

Um eine Einschätzung zur Verwendung des Kurvenbilds in Relation zu anderen gängigen Prozessvisualisierungen zu erhalten, wurden die Experten um einen Dominanz-Paarvergleich der in Abbildung 14 aufgelisteten Kategorien gebeten. Konform zu den qualitativen Aussagen wird dem Kurvenbild eine zentrale Bedeutung für die Prozessüberwachung beigemessen. Die Eindeutigkeit

des Ergebnisses wird durch eine zufriedenstellende Urteiler-Übereinstimmung ($W = 0.69^{**}$) bestätigt^a.

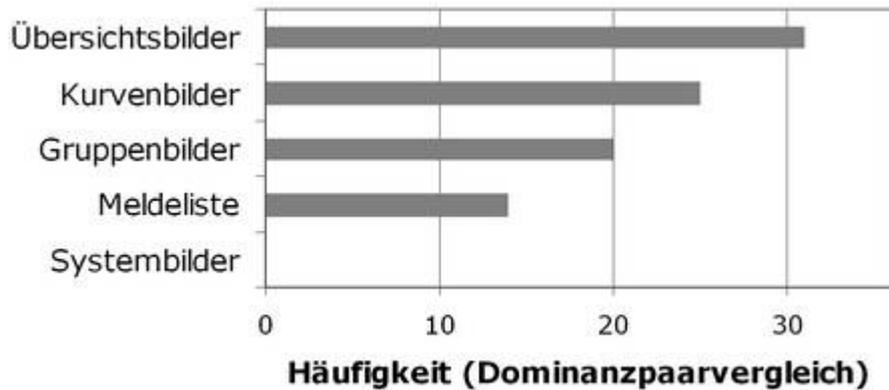


Abbildung 14 Anwendung verschiedener Prozessvisualisierungen

Tabelle 4 Beschreibung typischer Aufgaben bei der Nutzung von Kurvenbildern

Gesamteindruck	Wie hat sich der Prozess über den Zeitraum der letzten Schicht verhalten?
Ablesen von Messwerten	Was ist/war der konkrete Messwert zu einem konkreten Zeitpunkt?
Wirkkontrolle	Wie hat sich ein manueller Eingriff ausgewirkt?
Fehler- und Störungsdiagnose	Was ist ohne direkten Eingriff an Auffälligkeiten geschehen & warum?
Recherche in Prozesshistorie	Recherche von Kurvenverläufen von abgeschlossenen Ereignissen in der Prozess- bzw. Kurvenhistorie ^b
Qualitätskontrolle	Erfüllt mein Produkt die geforderten Spezifikationen?
Prädiktion	Vorhersage vom weiteren Verlauf aktueller Prozessereignisse

^a W kennzeichnet Kendalls Konkordanz-Koeffizient (Bortz & Lienert, 2003)

^b Die Kurvenhistorie wird hier in Anlehnung an die VDI/VDE Richtlinie 3699 (1997, Blatt 4) von der Kurvenvorgeschichte unterschieden. Die konkreten Definitionen finden sich im Begriffsglossar am Ende dieser Dissertationsschrift.

Auf Basis von Vorarbeiten (Kindsmüller, 2006) konnte auf eine Liste von Aufgaben zurückgegriffen werden, die Anlagenfahrer mithilfe von Kurvenbildern bewältigen (Tabelle 4). Die Experten wurden gebeten, anhand eines Dominanz-Paarvergleichs einzuschätzen, für welche Aufgaben dieser Liste Kurvenbilder am häufigsten durch die Anlagenfahrer der von ihnen betreuten Betriebe verwendet werden.

In Abbildung 15 sind die Häufigkeiten abgetragen, mit denen die einzelnen Aufgabentypen den jeweils anderen vorgezogen wurden. Erwartungskonform scheint das Kurvenbild an erster Stelle dem Gesamteindruck zu dienen. Seltener wird es dagegen für die Recherche in der Prozesshistorie („Blick zurück“) und die Prozess-Prädiktion („Blick voraus“) genutzt. Ursache dafür könnte die mangelhafte Effizienz derartiger Tätigkeiten sein, welche durch geeignete Tools wie der angestrebten Suchmaschine gesteigert werden kann. Allerdings waren sich die Experten in ihren Einschätzungen uneinig, wie die geringe Urteiler-Übereinstimmung ($W = 0.18$) ganz deutlich zeigt. Eine Ursache hierfür kann in den heterogenen Erfahrungen der Experten in ihren Kundenbetrieben vermutet werden.

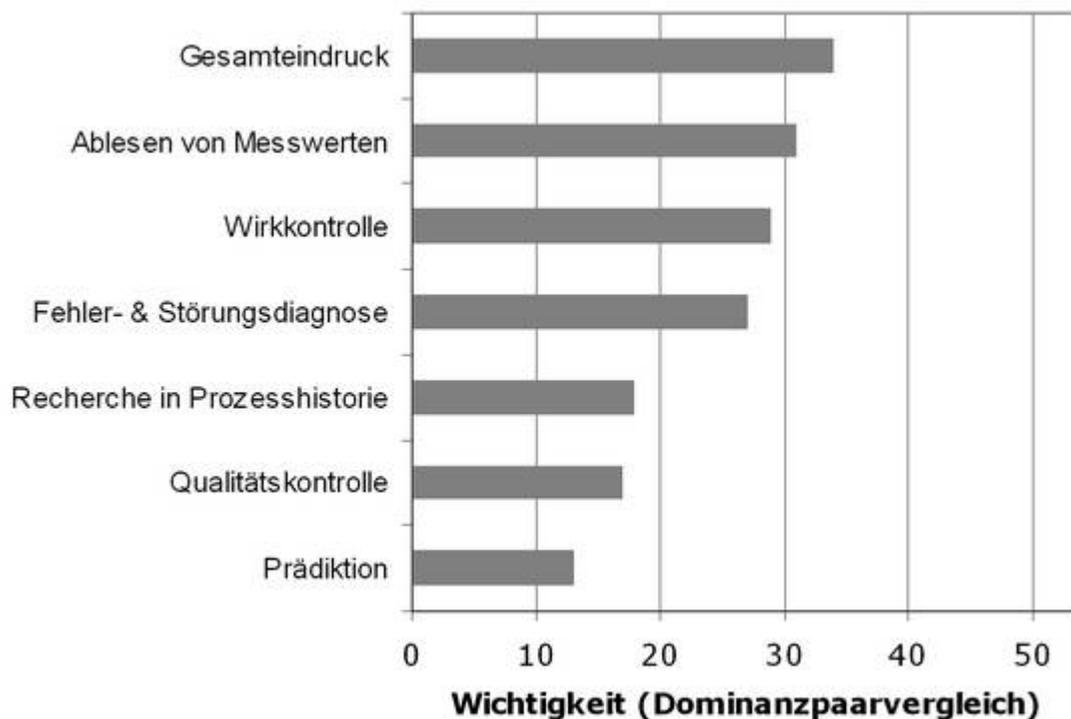


Abbildung 15 Anwendungsbereiche des Kurvenbilds im operativen Anlagenbetrieb

Die Experten wurden weiterhin gefragt, für welche fachlichen Tätigkeiten sie in ihrer eigenen Berufstätigkeit^a die Kurvendarstellung^b nutzen. Zusammenfassend können auf Basis der Interview-Transkripte folgende Anwendungsgebiete identifiziert werden:

- Analyse von Dynamiken und Schwingungen zur Unterstützung des Regler-Tuning
- Gegenüberstellung von Batch-Kurven aus verschiedenen Anlageneinheiten und aus verschiedenen Zeitintervallen
- Darstellung berechneter Kenngrößen zur Früherkennung von potentiellen Anlagenausfällen
- Störungsanalyse nach Instandsetzungsarbeiten
- Ereignissuche zur Untermauerung von Hypothesen über anormales Anlagenverhalten

Als alternative Analysemethoden bzw. –werkzeuge wurden von den Experten genannt:

- Fourier-Transformation zur Analyse des Schwingungsverhaltens
- Schichtbücher zur Rekapitulation von manuellen Eingriffen

Weitere Methoden wurden eher kritisch gesehen. So wurden Wissensdatenbanken kontrovers diskutiert. Entscheidend für die Nutzer-Akzeptanz ist danach die Handhabung der Schnittstelle und die Integration in den Arbeitsablauf. Die Brauchbarkeit statistischer Korrelation einzelner Prozessvariablen wurde teilweise geringer eingeschätzt als die der Kurvenanzeige. Bei der Faktorenanalyse stellte sich für die Experten die Problematik der Datenaufbereitung, z.B. beim Abgleich unterschiedlicher zeitlicher Ausdehnungen von Batches^c. Zudem müssen derartige PCA-Systeme (PCA: Principal Component Analysis) kontinuierlich „trainiert“ werden, vor allem bei häufigen Änderungen in der Auslegung und Fahrweise der verfahrenstechnischen Anlage. Wenn der Anwenderbetrieb dafür extra Spezialisten benötigt, wirkt dies zusätzlich erschwerend. Man erhält Signale über anormales Verhalten und einen Hinweis auf die Prozessgrößen, die dieses Verhalten verursacht haben. Das Ganze funktioniert jedoch hauptsächlich im stationären Bereich. Zuletzt ist die Interpretation durch Nicht-Spezialisten fehleranfällig.

^a Dies muss eindeutig unterschieden werden von der Frage nach dem Tätigkeitsfeld der Anlagenfahrer aus den von den Experten betreuten Betrieben.

^b Zur begrifflichen Unterscheidung von „Kurvenbild“ und „Kurvendarstellung“ sei auf das Begriffsglossar verwiesen.

^c Im Gegensatz zur kontinuierlichen Betriebsweise werden in einem Batch (eng. Stapel) mehrere Einzeloperationen sequentiell abgearbeitet.

11.2.1.2 Zusammenfassung

Die durchgeführte Expertenevaluation hat zu einem frühen Zeitpunkt der Prototypenentwicklung Einblicke in die Anwendungsmöglichkeiten des vorgeschlagenen Unterstützungswerkzeugs geliefert. Die Bedeutung des Kurvenbildes für das Prozessverständnis der Ingenieure und Anlagenfahrer wurde in dieser Untersuchung unterstrichen. Die Einschätzung, dass die Anlagenfahrer weniger in die Vergangenheit und Zukunft des Prozesses schauen, als vielmehr das aktuelle Prozessgeschehen betrachten, kann auf zwei Ursachen zurückgeführt werden. Zum einen sind diese kognitiven Leistungen weniger zur operativen Prozessführung notwendig, zum anderen fehlen adäquate Werkzeuge und Unterstützungsmöglichkeiten.

11.2.2 Technischer Unterstützungsbedarf

Im Rahmen der im Kapitel 11.2.1.1 beschriebenen Feldstudie wurden die Prozessexperten nach bekannten bzw. erwünschten Innovationen auf dem Gebiet der Kurvendarstellung bzw. -nutzung befragt. Die Befragten konnten jedoch nur eine geringe Anzahl an Aussagen generieren. Wie bereits von Alavi (1984) aufgezeigt, fiel es den Anwendern unter Ermangelung an konkreten bzw. greifbaren Konzepten oder Geschichten über potentielle Unterstützungssysteme schwer, kreative Neuerungswünsche äußern. Die Ergebnisse der offenen Befragung lassen sich deshalb auf die folgenden Punkten reduzieren:

- Interesse an Darstellung statistischer Kennwerte bzw. spezieller Ereignisse in den Kurvenverläufen
- Interesse an Mustersuche in der Vergangenheit bei gleichzeitiger Skepsis gegenüber den Umsetzungsmöglichkeiten
- Skepsis gegenüber 3D-Darstellungen Prozessinformationen
- Herausforderung für Darstellung von berechneten statt gemessenen Prozessgrößen
- Interesse an Messgrößen-Darstellung in Polardiagrammen

Tabelle 5 Softwareangebote der Prozessüberwachung und –analyse

Firma	Produkt	statistische Analyse	Zusätzliche Daten der Prozessführung
ABB	DigiVis	nein	Schicht- und Ereignisprotokolle
automation	aXdas	Urwert-, Mittelwert- und Streuwertkarte	
Copa-Data	Extended Trendmodule (ETM)	Minimal-, Maximal- und Mittelwert	Status
Citect	CitectSCADA	nein	Alarme
emation	Wizcon	Grenzwerte	
Honeywell	PlantScape Vista 400	nein	Alarme
Honeywell	Trendmanager	nein	Warnungen, Alarme und Bedienerkommentare Batch-Informationen (Start, Ende, Pause, Abbruch)
Honeywell	SDA	nein	Alarme, Ereignisse, Diagnosen
OSIsoft	PI Batch View	nein	Produktnamen, Prozeduren, Batches
Siemens	SIMATIC WinCC	ja	
Yokogawa	Exaquantum	ja	

Parallel zur Expertenbefragung wurde eine vergleichende Recherche durchgeführt, welche das aktuelle Angebot an Softwarekomponenten zur Prozessüberwachung und –analyse aufzeigt. Die recherchierten Produkte sind teilweise integriert in moderne Prozessleitsysteme, teilweise werden sie jedoch als unabhängige Analysewerkzeuge angeboten. Tabelle 5 zeigt eine Auflistung der aktuellen Angebote zur statistischen Analyse und der Informationsrecherche anhand von Daten der Prozessführung, welche simultan zum Kurvenbild abgebildet werden können.

11.2.3 *Nutzerexpertise bei der Prozessüberwachung und -analyse*

Während die kognitionswissenschaftliche Grundlagenforschung die Fähigkeit von Menschen, Kurvenbilder effizient zu lesen als Untersuchungsbereich weitgehend ignoriert hat (Kindsmüller, 2006), betonen verschiedene Autoren von Algorithmen aus der Klasse der qualitativen Trendanalyseverfahren (engl. *Qualitative Trend Analysis, QTA*), dass ihre Verfahren als „emulation of human perception and reasoning capabilities“ konzipiert sind (Kivikunnas, 1999, S. 6). Kindsmüller (2006) kommt anhand empirischer Untersuchungen zur Kurvenlesefähigkeit zu dem Schluss, dass die QTA-Verfahren zwar ein Ansatz zur Emulation perzeptiver Bottom-Up-Prozesse darstellen. Der Einfluss von Top-Down-Prozessen, die sich auch in Leistungsparametern wahrnehmungsnaher Prozesse beim Kurvenlesen deutlich niederschlagen, kann durch die QTA-Verfahren jedoch nicht erklärt werden.

Dies mag auch ein Grund dafür sein, dass sich die QTA-Verfahren bisher nicht durchsetzen konnten. Die rein datengetriebene Trendanalyse scheitert daran, dass für eine erfolgreiche Analyse eine Verbindung von Kontextwissen und Datenebene notwendig wäre. Erfolgreich meint hier, dass das Verfahren nur relevante Ereignisse und diese möglichst vollständig findet. In der Terminologie der Signalerkennungstheorie (Green & Swets, 1988; Wickens et al., 2004) ist eine hohe Wahrscheinlichkeit für das Erkennen von relevanten Stellen und eine hohe Wahrscheinlichkeit für eine korrekte Zurückweisung von irrelevanten aber ähnlichen Stellen zu fordern. Ziel des Ansatzes einer adäquaten Aufgabenteilung zwischen Mensch und Technik ist es, die Vorteile computergestützter Bottom-Up-Prozesse und Experten gesteuerter Top-Down-Prozesse zu kombinieren.

Für die Nutzung der Top-Down-Prozesse lässt sich auf die Metapher der Trendbeschreibung zurück greifen, welche im Wesentlichen einer Theorie des Kurvenlesens (*Trend Literacy*) folgt. Diese wurde von Kindsmüller (2006) analytisch hergeleitet und empirische überprüft. Danach sind Prozessexperten in der Lage, komplexe Prozessdynamiken in zeitliche Episoden zu unterteilen, und diese anhand geometrischer Primitiva, sogenannter *Trendlets* (vgl. Abbildung 11 auf S. 34) zu klassifizieren. Feldstudien des Autors zeigten, dass über achtzig Prozent aller Prozessexperten Kurven lesen,

indem sie diese spontan anhand von dynamischen Primitiva segmentieren. Bei diesen ist das Segmentieren anhand von Extrem- und Wendestellen weit verbreitet. Unabhängig von der Art der Trendlet-Typen kommen die Prozessexperten mit 4 bis 7 verschiedenen Trendlets aus.

11.2.4 Entwurf potentieller Aufgabenszenarien

Während die Aufgabenanalyse des bestehenden Arbeitssystems noch stark Nutzer zentriert erfolgen konnte (Kapitel 0 bis II.2.3), kam dieser Ansatz bei der Analyse der Aufgabenbearbeitung mithilfe eines hypothetischen Unterstützungssystems an seine Grenzen. Dieser Effekt ist verstärkt zu beobachten bei hoch innovativen Produkten, wie die Trend basierte Datenbanksuchmaschine eines darstellt (Alavi, 1984). Die hier dokumentierte Szenario basierte Aufgabenanalyse wurde deshalb zunächst in einem Entwickler zentrierten Ansatz durchgeführt. Damit wurde jedoch das Design eines Nutzerdialogs in Form von interaktiven Klick-Prototypen möglich (siehe Kapitel III.2). Mit einem frühen Prototypen wurde die geplante Anwendung dann für die potentiellen Nutzer greifbar. Sie brauchten sich weniger in abstrakten, spekulativen Szenarien bewegen und waren damit in der Lage, konstruktive und realistische Wünsche und Anmerkungen zu äußern (vgl. Kapitel II.2.5).

Die Szenario basierte Aufgabenanalyse wird von Sutcliffe (2003) als Kontinuum beschrieben. Auf der einen Seite dieses Kontinuums stehen narrative^a Beschreibungen von Aufgabenbearbeitungen (Carroll, 2000). Auf der anderen Seite stehen formale Modelle, wie sie im Objekt orientierten Systemdesign verwendet werden (Cockburn, 2001; Jacobson et al., 1992). Eine zweite Unterscheidungsdimension betrifft die Art der Aufgabe, welche analysiert wird. Diese variieren von Aufgaben aus dem existierenden, realen Arbeitskontext hin zu Interaktionsspezifikationen von Artefakten zukünftiger Arbeits- und Unterstützungssysteme. Szenarios wirken damit als Inspiration und Testmaterial für die Prototypenentwicklung (Misra et al., 2005; Sutcliffe, 2002). Sie werden als beispielhafte *stories* entworfen und im Laufe des Systemdesigns in funktionale Modell und Prototypen transformiert.

^a narrativ = in erzählender Form darstellend

Während der Überwachung einer Destillationskolonne wird ein spezieller Trend in der Kondensattemperatur beobachtet. Mit den bekannten Überprüfungen kann der Prozessexperte keine Ursache für den Trend finden. Er fragt sich, ob es ein ähnliches Prozessverhalten schon einmal gegeben hat. Denn wenn er ein Ereignis mit einem ähnlichen Trend findet, kann er dieses hinsichtlich weiterer Prozesseigenschaften wie Bauteilausfälle, Steuereingriffe und Umgebungsbedingungen überprüfen. Aus der Analyse der Vergangenheit kann er dann auf die aktuelle Situation schließen.

Er stellt sich die Frage, wie nach ähnlichen Trends gesucht werden kann. Mit der zu entwickelnden Suchmaschine startet der Prozessexperte mit einer reduzierten Beschreibung des Trends, welcher ihm charakteristisch für die aktuelle Dynamik erscheint. Er definiert dafür eine Sequenz aufeinander folgender Episoden, und spezifiziert diese hinsichtlich deren Länge, Höhe, und Dynamik. Mit dieser Trendbeschreibung ist die Suchmaschine in der Lage, Kurvenverläufe mit vergleichbaren Trends zu präsentieren.

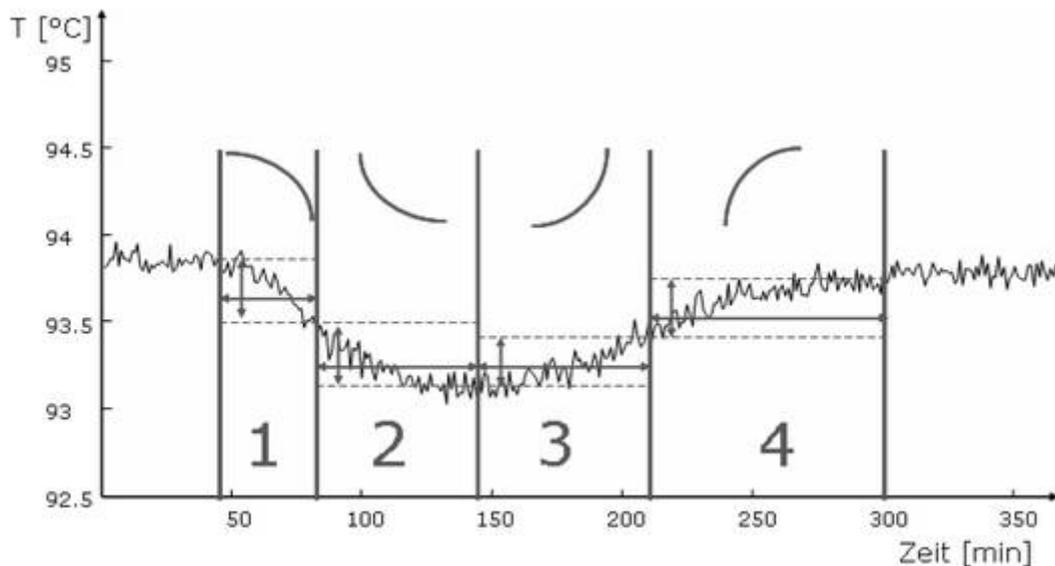


Abbildung 16 Szenario-Beispiel, Teil 1

Im hier dokumentierten Entwicklungsprozess wurde sich der Szenario basierten Aufgabenanalyse bedient. Abbildung 16 und Abbildung 17 zeigen ein beispielhaftes Anwendungsszenario (Ursachenanalyse bei Prozess-Anomalitäten), welches in ein weiter abstrahiertes Interaktionsmodell überführt wurde (Abbildung 18). Dieses bildet einen hypothetischen *Workflow* bei der Nutzung der Trend basierten Datenbanksuchmaschine zur Diagnose von Prozess-Anomalitäten ab. Die Aufgabe startet mit der Beobachtung eines anomalen Prozessverhaltens. Die Fragen, welche sich der Prozessexperte stellen mag, lauten:

- Was ist die Ursache für das anomale Prozessverhalten?

- Welcher Fehler bzw. welche Störung verursacht dieses Verhalten?

Ist der Experte in der Lage, eine konkrete Hypothese zu formulieren, kann er durch eine Suche anhand von Schlüsselwörtern unterstützt werden (Schritt 3 & 6). Darüber hinaus kann er seine Hypothese durch gezielte Anlageninspektionen überprüfen (Schritt 5). Wenn er allerdings keine solche explizite Hypothese generieren kann, wird ihm durch eine Trendbasierte Suchmaschine ermöglicht, vergleichbare Trends in der Datenbank zu finden (Schritte 8, 10 & 14). Sein Ziel ist es dabei, auf Basis der gefundenen Ereignisse Ursachenhypothesen für das aktuelle Ereignis zu generieren (Schritt 15).

Drei andere Kurven werden gefunden, die sich alle hinsichtlich der Länge und Höhe unterscheiden. Bei der Analyse der Suchergebnisse stellt der Prozessexperte fest, dass sie alle bei einem unterschiedlichen Arbeitspunkt (Durchflussrate im Kolonnenzufluss) stattgefunden haben. Gleichzeitig haben sie jedoch alle eine gemeinsame Ursache, nämlich einen Sprung in der Zuflusskonzentration. Eine zeitnahe Erkenntnis dieses Zusammenhangs bringt Vorteile, da Konzentrationsmessungen in der Praxis nur in längeren zeitlichen Abständen oder ganzen Schichten durchgeführt werden. Dies ist in der teuren Technik automatischer Konzentrationsmessungen begründet. Die Messungen werden deshalb meist per Hand durchgeführt. Eine vorgezogene, außerhalb der festen vorgenommene manuelle Probe bietet sich also in diesem Fall an, um die aus den historischen Fällen generierte Hypothese zu verifizieren.

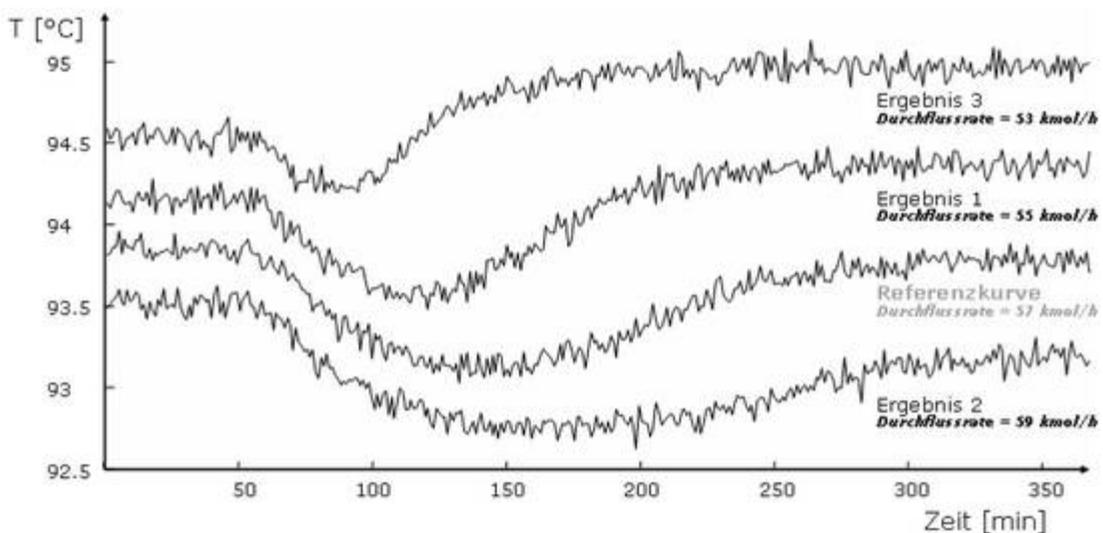


Abbildung 17 Szenario-Beispiel, Teil 2

11.2.5 Aufgabenanalyse anhand einer Prototypen-Evaluation

Ziel der Expertenevaluation war die Erkundung möglicher Einsatzgebiete einer Trend basierten Suchmaschine sowie die Überprüfung des aktuellen Entwicklungsstands vom Nutzerdialog. Dazu wurde erfahrenen Prozessexperten aus dem Bereich Forschung und Entwicklung ein früher Klick-Prototyp der Trend basierten Datenbank-Suchmaschine vorgeführt (Anhang A.1.2.1). Darauf aufbauend erfolgte eine Diskussion^a von Einsatzmöglichkeiten bzw. –beschränkungen und weiteren wünschenswerten Funktionalitäten (Anhang A.1.2.2). Der Prototyp war in seinem grundlegenden Nutzerdialog der finalen Version aus Kapitel III.2.2 bereits recht ähnlich, verfügte jedoch noch über keinerlei Suchfunktionalitäten.

Als potentielle Anwender führten die Experten nicht den Operateur im Schichtdienst auf, sondern vorwiegend Betriebsleiter, Betriebsingenieure, Techniker oder Meister. Bei diesen wurden Aufgabenfelder genannt wie die Betrachtung der Dynamik und der Prozesshistorie, die Störungs- und Fehlerdiagnose und der Automatisierung. Statt in alten und bekannten Produktionsanlagen sollte es nach Expertenmeinung vor allem in Forschungsanlagen und neuen Betrieben erhöhten Bedarf an derartigen Kurvenvergleichen geben.

Alle Experten waren sich einig, dass bei der Weiterentwicklung des Prototypen auf die Einfachheit der Funktionalitäten geachtet werden solle. In diesem Sinne wurde die reduzierte Form des vorgestellten Trendlet-Alphabet begrüßt. Eine Anpassung an spezielle Prozessmerkmale wurde jedoch für notwendig erachtet, ebenso wie die Eingrenzung der Suchanfrage mithilfe von Prozess spezifischen Rahmenbedingungen, die Gewichtung der verschiedenen Ähnlichkeitsdimensionen und die Kombinationsmöglichkeit von verschiedenen Variablen. Wichtig ist den potentiellen Anwendern nicht zuletzt auch das Vorhandensein einer Speicherfunktion sowie die Reduktion von Alarmen.

^a Es wurden Audioaufnahmen der Gespräche erstellt, im Anschluss nieder geschrieben und inhaltsanalytisch ausgewertet.

Tabelle 6 Nutzerprognose zum Einsatz einer Trend basierten Datenbanksuchmaschine

Einsatzmöglichkeiten	Einsatzbeschränkungen	Einsatzvoraussetzungen
Störungsanalyse	Nicht für Anlagenfahrer im Schichtbetrieb geeignet	Erweiterung des Trendlet- Alphabets um Sprünge und komplexe Muster wie Einschwingen und Aufschwüngen
Störungsprognose	Erfahrenes Personal mit guter Kommunikationskultur notwendig	direkte, Maus gesteuerte Zoomfunktion (drag & drop)
Fehlerreproduktion	Kein Mehrwert bei bekannten Anlagen mit geringem Optimierungspotential	editierbare, Anlagen spezifischen Bibliothek an Trendlet-Kombinationen
Alarm-Management		Speicherfunktion für Trendbeschreibungen
Analyse von (zeitversetzten) Zusammenhängen zwischen Prozessvariablen		Kombination mit Betriebsdaten zur Spezifikation des Suchraumes
Prozessoptimierung		Beibehaltung der simplen Interaktion
Vergleiche zwischen Simulations- und Realdaten		
Batch-Vergleiche		
Template-Erstellung für Batch-Prozesse		

II.3 Zusammenfassung

Aus den im Kapitel II beschriebenen Vorarbeiten lassen sich einige Ansätze für die weitere Spezifikation eines gebrauchstauglichen Nutzerdialogs sowie eines effektiven Suchalgorithmus nutzen.

Die Faltung im Allgemeinen und die Wavelet-Transformation im Speziellen wurden als etablierte und effektive Methoden der Multiskalenanalyse vorgestellt. Sie erscheinen als viel versprechende Grundlage für das Segmentieren und Indexieren von Zeitreihen hinsichtlich ihrer Dynamiken auf verschiedenen Frequenzbändern. Das Verfahren des Segmentierens wird als essentiell notwendig erachtet, um eine nachvollziehbare und konsistente Verknüpfung zu der menschlichen Wahrnehmung und Interpretation von Dynamiken in Kurvenverläufen herzustellen. Die Methode des Indexierens soll auf den Ähnlichkeiten der Zeitreihenmodelle basieren. Das Euklidische Abstandsmaß zwischen den Originalzeitreihen erscheint als Indexierungskriterium für ungeeignet. Nutzer vergleichen Trends in Zeitreihen nicht in Form von Abständen zwischen einzelnen Mess-Datenpunkten.

Zwei verschiedene Ansätze zum Segmentieren von Zeitreihen anhand von Episoden wurden vorgestellt (Kapitel II.1.3.1 & II.1.3.2). Beide wurden bereits erfolgreich wissenschaftlich erprobt und exemplarisch in der Praxis angewandt. Beide ermöglichen das Segmentieren anhand von geometrischen Primitiva, welche nach der Theorie der Trend-Literacy (Kinds Müller, 2006) konform mit den mentalen Repräsentationen von Prozessexperten sind. Die jeweilige analytische Ratio für das Computer gestützte Segmentieren wurde bewertet. Die Identifikation gekrümmten Episoden anhand der relevanten Punkte aus der Kurvendiskussion (Extrem- und Wendestellen) kann als sehr vielversprechend eingeschätzt werden. Die Linearapproximation anhand von Polynomen scheint ebenfalls plausibel. Allerdings ist damit kein kontinuierliches und vollständiges Segmentieren einer Zeitreihe möglich.

In einer Feldstudie wurde die Relevanz der Kurvenbilder für die Prozessüberwachung und –analyse bestätigt. Darüber hinaus wurde der Unterstützungsbedarf von Prozessexperten bei der Analyse

der Prozesshistorie identifiziert. Die Nutzerexpertise im Kurvenlesen wurde als wesentliches Nutzermerkmal für die Konzeption des Nutzerdialogs bestimmt. Aufgabenszenarien wurden als Unterstützungswerkzeuge für den Design-Prozess erstellt. Eine frühe Evaluation des Nutzerdialogs anhand einer Prototypenbewertung lieferten wichtige Hinweise für die weitere Entwicklung.

III Systemdesign

In Anlehnung an die erste Ablaufskizze der zu gestaltenden Mensch-Technik-Interaktion aus Tabelle 1 und dem Aufgabenszenario aus Kapitel II.2.4 wurde ein grober Systementwurf erstellt. Dieser lässt sich recht anschaulich anhand der Abbildung 19 verdeutlichen. Über einen zu spezifizierenden Dialog greift der Experte auf Referenzkurven zu, deren signifikante Dynamik er interpretiert und in Form von Trends beschreibt. Mit der Trendbeschreibung ist der Suchalgorithmus in der Lage, eine Datenbankanfrage zu stellen. Dafür nutzt er einen offline zu erstellenden Index. Auf Basis dieses Index beurteilt der Algorithmus die Ähnlichkeit der Vergleichskurven mit dem beschriebenen Trend. Die Ergebnisse werden an den Experten zurück gegeben.

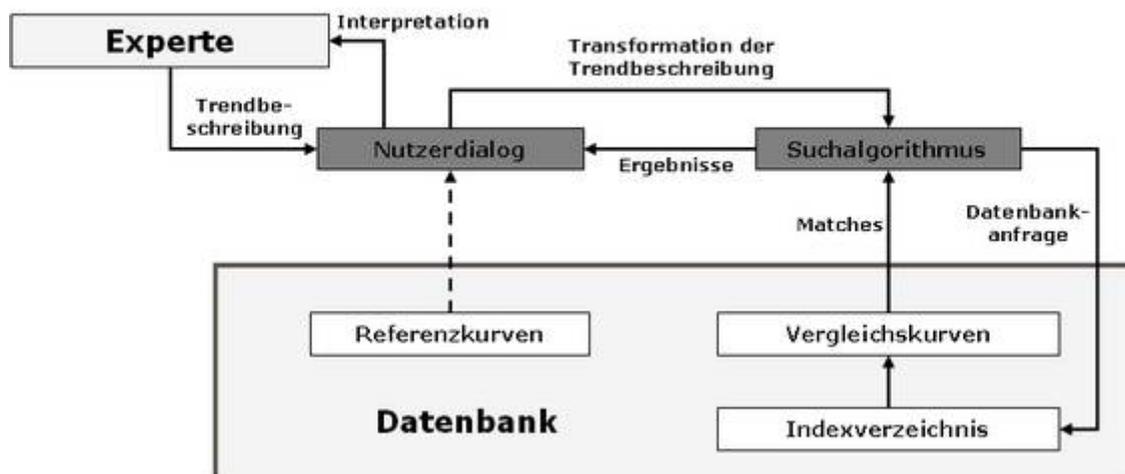


Abbildung 19 Interaktionsmodell zum Systementwurf

Weitere, in einer frühen Phase des Systemdesigns festgelegte Spezifikationen sind in Tabelle 7 aufgelistet. Dort findet sich bereits die Unterscheidung in zwei voneinander unabhängige Indexe für lineare und gekrümmte Episoden. Die Hauptkriterien der Ähnlichkeitsschätzung wurden ebenfalls definiert. Beim Kurvenbild als Basis für den Nutzerdialog wurde sich für eine Orientierung an VDI Richtlinien und einem Marktführer auf dem Gebiet der Prozessleitsysteme entschieden. Erste

Anforderungen an den Informationsgehalt der Trendbeschreibung und der Einbindung der Ergebnisse in das klassische Kurvenbild wurden definiert (Huss & Barz, 2005). Ausgehend von diesen Spezifikationen wurde parallel an dem Suchalgorithmus und dem Nutzerdialog gearbeitet. Das iterative Vorgehen und die Ergebnisse dieser Entwicklungsarbeit sollen in den folgenden Kapiteln III.1 und III.2 weiter erläutert werden.

Tabelle 7 Überblick über Detailentwurf

Suchalgorithmus	Nutzerdialog
<p>Daten-Vorverarbeitung</p> <ul style="list-style-type: none"> • Multiskalenanalyse anhand von Faltungen auf unterschiedlichen Frequenzbändern • Index für lineare Episoden • Index für gekrümmte Episoden 	<p>Kurvenbild</p> <ul style="list-style-type: none"> • Freelance 2000 (ABB) • Richtlinie VDI/VDE 3699-4 (1997)
<p>Suche</p> <ul style="list-style-type: none"> • Sequentielle Suche nach Episodenkombinationen • Ein bis ca. 7 Episoden pro Trend 	<p>Trendbeschreibung</p> <ul style="list-style-type: none"> • Episodendefinition • Trendlet-Alphabet • Längen- und Höhenparameter
<p>Ähnlichkeitsschätzung</p> <ul style="list-style-type: none"> • Abweichungen entlang der Zeit- und Wertachsen 	<p>Ergebnispräsentation</p> <ul style="list-style-type: none"> • Flexibilität im Kurvenvergleich • Flexibilität in Gesamtzahl angezeigter Kurven

III.1 Suchalgorithmus

Neben den häufigen Zyklen aus Konzept, Entwurf, Implementierung, Test, Evaluation und Re-Design innerhalb der rein technischen Entwicklung hat sich der konzeptuelle Abgleich zwischen den beiden Hauptentwicklungspfaden *Suchalgorithmus* und *Nutzerdialog* bewährt. Um diese Form der Iteration zu verdeutlichen, wird im Anschluss an die Beschreibung und Evaluation von Entwurf I im Kapitel III.1.1 detailliert auf Entwurf II eingegangen (Kapitel III.1.2), welcher die Anpassung an den parallel entworfenen Nutzerdialog darstellt.

III.1.1 Entwurf I

Auf Basis der oben dargestellten Anforderungsanalyse wurde ein erster Entwurf eines Suchalgorithmus entworfen. In den folgenden Kapiteln wird zunächst das verwendete Grundkonzept verdeutlicht. Danach wird auf die Vorverarbeitung der Datenbank zur Gewinnung eines effizienten Suchraums dargestellt. Im Anschluss wird die Vorgehensweise zur Suche nach linearen und gekrümmten Episoden erläutert. In einer abschließenden Bewertung von Entwurf I werden Probleme bei der konzeptuellen Kompatibilität mit dem Nutzerdialog aufgezeigt. Diese Bewertung bietet den Ausgangspunkt für den in Kapitel III.1.2 dokumentierten Entwurf II.

III.1.1.1 Grundkonzept

Zur Abbildung einer originalen Zeitreihe auf unterschiedlichen Frequenzbändern wird die Wavelet-Transformation als spezielle Form der Faltung verwendet (vgl. Kapitel II.1.1). Mit dyadischen Skalierungsparametern werden dabei Wavelet-Transformierte erstellt (Abbildung 20), deren Extremstellen in einem nächsten Schritt zu Ketten zusammen gefasst und in dieser Form als Suchindex gespeichert werden (Abbildung 21). Die so generierten E-Ketten bilden den Krümmungsverlauf der originalen Zeitreihe auf zwei Dimensionen ab: der Zeit und den Amplituden. Betrachtet man das Auftreten starker Krümmungen als das Abweichen von Linearität, können die E-Ketten darüber hinaus zur Identifikation linearer Episoden verwendet werden.

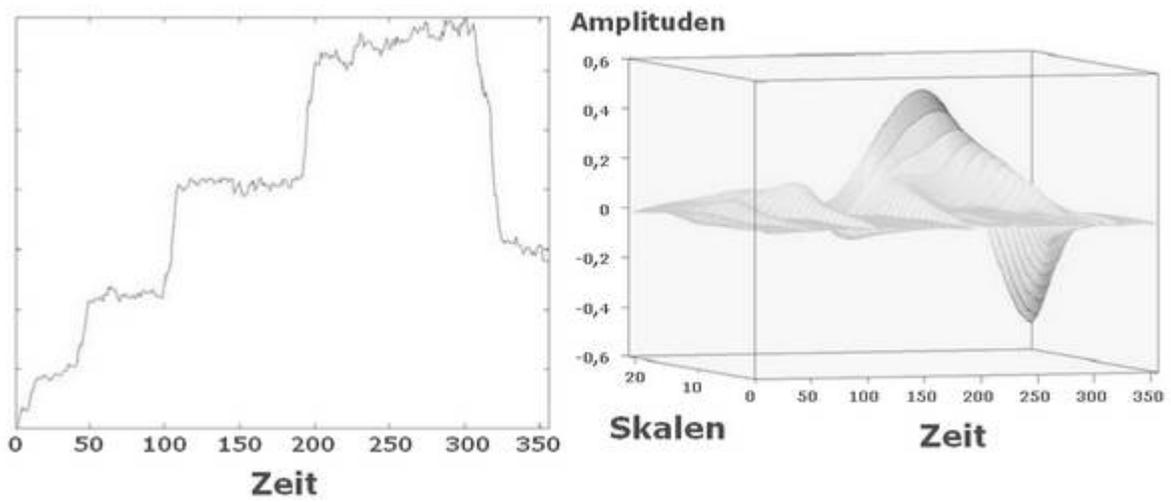


Abbildung 20 Zeitreihe (links) und korrespondierende Amplituden der Wavelet-Transformierten (rechts)

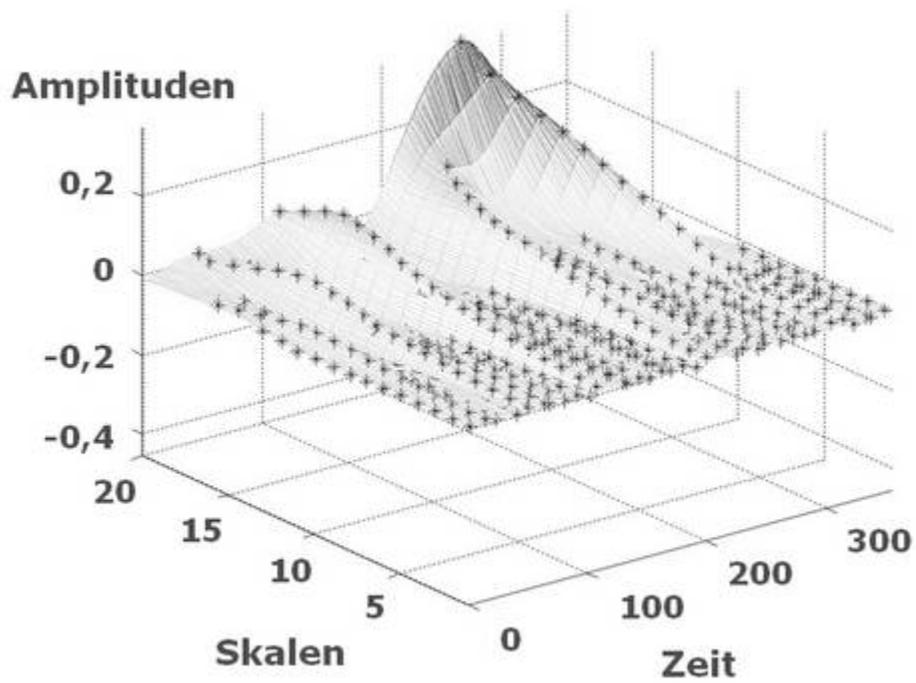


Abbildung 21 Datenreduktion in Form von E-Ketten (Kreuze)

III.1.1.2 Vorverarbeitung

Zunächst werden alle vorhandenen Zeitreihen mit dem Mexican-Hat-Wavelet gefaltet (vgl. Kapitel II.1.1). Da die absoluten Amplituden in Abhängigkeit vom Skalierungsparameter ansteigen (vgl. Abbildung 20 und Abbildung 21), ist für die Verwendung generell anwendbarer Schwellwerte eine Normierung der Amplituden am Skalenmaximum notwendig (Gleichung 10). Sodann werden für alle einzelnen Wavelet-Transformierten die Extremstellen \mathbf{m} extrahiert (Gleichung 11). Die Bildung der E-Ketten startet dann mit der Extremstelle $\mathbf{m}_{\sigma,t}$ der mit $\max(\sigma)$ erstellten Wavelet-Transformierten^a. Auf der Wavelet-Transformierten mit dem nächst kleineren Skalierungsparameter σ wird nun die auf der Zeitachse nächstliegende Extremstelle gleichen Vorzeichens $\mathbf{m}_{\sigma-1}$ gesucht und der Startstelle zugeordnet. Sind zwei Extremstellen gleich weit entfernt, wird dasjenige gewählt, welches die höhere Amplitude aufweist. Diese Prozedur wird bis zur untersten Skala durchgeführt. Kann eine Extremstelle keiner bestehenden E-Kette \mathbf{M} zugeordnet werden, wird eine neue Kette gebildet.

$$\text{Gleichung 10} \quad \hat{f}_{\sigma}(t) = \frac{f_{\sigma}(t)}{\max(f_{\sigma}(t))}$$

$$\text{Gleichung 11} \quad \mathbf{m}_{\sigma,t} = \left\{ t_{m_{\sigma}}, \hat{f}_{\sigma}(t_m) \right\} \left| \begin{array}{l} \hat{f}_{\sigma}(t_m - 1) < \hat{f}_{\sigma}(t_m) > \hat{f}_{\sigma}(t_m + 1) \\ \hat{f}_{\sigma}(t_m - 1) > \hat{f}_{\sigma}(t_m) < \hat{f}_{\sigma}(t_m + 1) \end{array} \right.$$

$$\text{Gleichung 12} \quad \mathbf{M} = \{ \mathbf{m}_1, \dots, \mathbf{m}_{\sigma} \} \left| \min(|t_{m_{\sigma}} - t_{m_{\sigma-1}}|) \right.$$

III.1.1.2.1 Identifikation linearer Episoden

Bereits in der Vorverarbeitung werden zugunsten einer geringen Antwortzeit während der Suchanfrage die linearen Episoden auf Basis der Amplituden aller Extremstellen aller E-Ketten bestimmt und gekennzeichnet. Der Zeitpunkt t_{m_1} der Extremstelle auf der untersten Wavelet-Skala einer E-Kette kennzeichnet den Start bzw. das Ende (t_{lin}) einer linearen Episode, sobald ihre durchschnittlichen normierten Amplituden über einem Schwellwert $\mathbf{m}_{\text{linTol}} = \{0, \dots, 1\}$ liegen (Gleichung 13).

^a Nur bei einer Wavelet-Transformation mit einem maximalen Skalierungsparameter kommt es zur Reduktion auf nur eine Extremstelle. In den anderen Fällen wird mit mehr als einer Extremstelle gestartet.

Gleichung 13

$$t_{lin} = t_{m_1} \left| \frac{\sum_{\sigma=1}^{\max(\sigma)} \hat{f}_{\sigma}(t)}{\max(\sigma)} \geq m_{linTol} \right.$$

Für die zwischen allen möglichen Paaren an aufeinander folgenden Zeitpunkten t_{lin} liegenden Intervalle werden in einem nächsten Schritt die Regressionsgeraden berechnet. Um die Flexibilität des Suchalgorithmus gegenüber unterschiedlichen Suchanfragen zu gewährleisten, wird diese Prozedur für unterschiedliche Linearitätsschwellwerte durchgeführt. Zwei beispielhafte Ergebnisse finden sich in Abbildung 22.

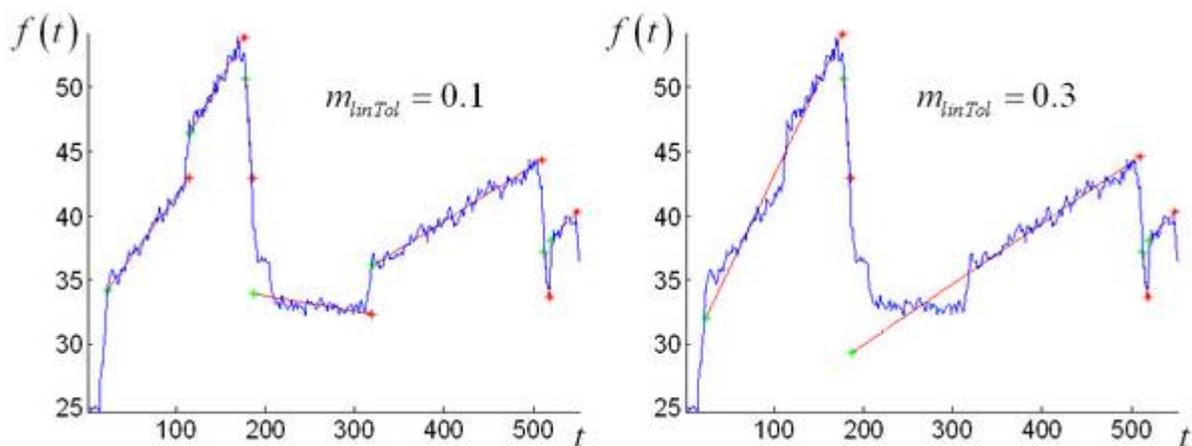


Abbildung 22 Lineare Approximationen auf Basis verschiedener Schwellwerte

III.1.1.3 Suchprozedur

Die Referenzzeitreihe muss einer Vorverarbeitung wie unter Kapitel III.1.1.2 unterzogen werden. Dies kann zum Zeitpunkt der Suchanfrage oder bereits offline erfolgen. Eine online bzw. dynamische Transformation ist zwar möglich, aufgrund der Randeffekte allerdings nicht besonders hilfreich. Die Konzepte von Suchheuristiken für lineare und gekrümmte Episoden werden in den folgenden Kapiteln erläutert.

III.1.1.3.1 Suche nach linearen Episoden

Es werden zwei E-Ketten \mathbf{M}_{st} und \mathbf{M}_{end} in der Referenzzeitreihe identifiziert, welche den tatsächlichen Startzeitpunkt $t_{m_1} \in \mathbf{M}_{st}$ und Endzeitpunkt $t_{m_1} \in \mathbf{M}_{end}$ der vom Nutzer anhand der geschätz-

ten Zeitpunkte t_{st} und t_{end} ausgewählten linearen Referenz-Episode markieren (Gleichung 14). Auf Basis der durchschnittlichen genormten Amplituden wird dann der spezifische Schwellwert \widehat{m}_{linTol} berechnet (Gleichung 15). Nach vergleichbaren linearen Episoden wird nun im Index der Vorverarbeitung gesucht, welcher auf Basis einer Linearitätstoleranz m_{linTol} entstanden ist, die möglichst nah an der empirisch ermittelten Toleranz \widehat{m}_{linTol} liegt. Diese dürfen in ihrer Länge und Höhe nur innerhalb von Toleranzen von der Referenz-Episode abweichen (Gleichung 16 und Gleichung 17).

$$\emptyset M_{st} = \frac{\sum_{\sigma=1}^{\max(\sigma)} \widehat{f}_{\sigma}(t)}{\max(\sigma)} \left| \begin{array}{l} \widehat{f}_{\sigma}(t) \in M_{st} \\ t_{st} - (0.2 * |t_{end} - t_{st}|) < t_{m_1} < t_{st} + (0.2 * |t_{end} - t_{st}|) \\ t_{m_1} \in M_{st} \end{array} \right.$$

Gleichung 14

$$\emptyset M_{end} = \frac{\sum_{\sigma=1}^{\max(\sigma)} \widehat{f}_{\sigma}(t)}{\max(\sigma)} \left| \begin{array}{l} \widehat{f}_{\sigma}(t) \in M_{end} \\ t_{end} - (0.2 * |t_{end} - t_{st}|) < t_{m_1} < t_{end} + (0.2 * |t_{end} - t_{st}|) \\ t_{m_1} \in M_{end} \end{array} \right.$$

Gleichung 15 $\widehat{m}_{linTol} = 0.5 * (\emptyset M_{st} + \emptyset M_{end})$

Gleichung 16 $t_{end} - t_{st} - \Delta t < \overline{t_{end}} - \overline{t_{st}} < t_{end} - t_{st} + \Delta t$

Gleichung 17 $f(t_{end}) - f(t_{st}) - \Delta h < \overline{f}(t_{end}) - \overline{f}(t_{st}) < f(t_{end}) - f(t_{st}) + \Delta h$

III.1.1.3.2 Suche nach gekrümmten Episoden

Definiert der Nutzer einen Zeitabschnitt einer Referenzzeitreihe als gekrümmte Episode, wird für diesen Zeitbereich zunächst die dominante E-Kette identifiziert. Dafür wird die stabilste E-Kette extrahiert, deren Zeitkoordinate auf der untersten Wavelet-Skala zwischen dem vom Nutzer markierten Startzeitpunkt t_{st} und Endzeitpunkt t_{end} liegt.

Es werden nun lediglich die E-Ketten aus der Datenbank identifiziert, welche eine ähnliche Stabilität aufweisen, d.h. sie verlaufen bis zu einer Wavelet-Skala, welche nicht weiter als s Skalierungsparameter von der höchsten Wavelet-Skala abweicht, auf welcher die E-Kette der Referenz-Episode endet. Je höher der Parameter s gewählt wird, desto ungenauer werden die Ergebnisse. Je stärker sich s dem Wert Null annähert, desto mehr passende Ergebnisse werden abgewiesen. Die

Abstände zwischen den Orten (Zeitunterschied) bzw. Amplituden (Amplitudenunterschied bzw. Krümmungsstärke) der Extremstellen zweier E-Ketten bilden die Kriterien für den Vergleich der Zeitreihen und deren Ähnlichkeitsschätzung.

Zur Überprüfung der Episodenähnlichkeit werden zunächst die Zeitpunkte der Extremstellen beider E-Ketten (Referenz-Episode: t_{m_σ} ; potentielle Ergebnis-Episode: $\overline{t_{m_\sigma}}$) auf der kleinsten Wavelet-Skala $\sigma = 1$ auf einen gemeinsamen relativen Zeitpunkt gelegt. Bleibt der durchschnittliche zeitliche Abstand aller auf den weiteren Wavelet-Skalen liegenden Extremstellen-Paare unter einem Schwellwert $m_{\text{gekr,t}}$ (Gleichung 18), und bleibt gleichzeitig der durchschnittliche Amplitudenabstand aller Extremstellen $\hat{f}_\sigma(t_{m_\sigma})$ (Referenz-Episode) und $\hat{f}_\sigma(\overline{t_{m_\sigma}})$ (potentielle Ergebnis-Episode) unter einem Schwellwert $m_{\text{gekr,f}}$ (Gleichung 19), kennzeichnet die verglichene E-Kette eine Ergebnis-Episode.

$$\text{Gleichung 18} \quad \frac{\sum_{\sigma=1}^{\max(\sigma)} | \overline{t_{m_\sigma}} - t_{m_\sigma} |}{\max(\sigma)} < m_{\text{gekr,t}}$$

$$\text{Gleichung 19} \quad \frac{\sum_{\sigma=1}^{\max(\sigma)} | \hat{f}_\sigma(t_{m_\sigma}) - \hat{f}_\sigma(\overline{t_{m_\sigma}}) |}{\max(\sigma)} < m_{\text{gekr,f}}$$

III.1.1.4 Bewertung von Entwurf I

Nach einem kurzen Verweis auf Ergebnisse explorativer Test (Kapitel III.1.1.4.1) werden die Schwächen von Entwurf I bei der Kombination einzelner Episoden verdeutlicht (Kapitel III.1.1.4.2). Zentrale Komponente der iterativen Systembewertung ist jedoch die Überprüfung der Passung zwischen Technik und Mensch bzw. zwischen Suchalgorithmus und Nutzerdialog (Kapitel III.1.1.4.3).

III.1.1.4.1 Explorative Testergebnisse

Auf Basis einer prototypischen Implementierung des vorgestellten Entwurfs waren erste explorative Tests möglich. Deren Ergebnisse wurden von Barz et al. (2006) der verfahrenstechnischen

Forschungsgemeinschaft vorgestellt. Die Performanz bei der Identifikation linearer Episoden zeigte bereits Abbildung 22 aus Kapitel III.1.1.2.1. Abbildung 23 veranschaulicht dagegen zwei beispielhafte Ergebnisse bei der Suche nach einer gekrümmten Episode.

Darüber hinaus wurde auf weitere systematische Tests verzichtet, da die analytische Bewertung (Kapitel III.1.1.4.2 und III.1.1.4.3) des Entwurfs Probleme aufzeigte, deren Lösung zur Erarbeitung von Entwurf II führte (Kapitel III.1.2).

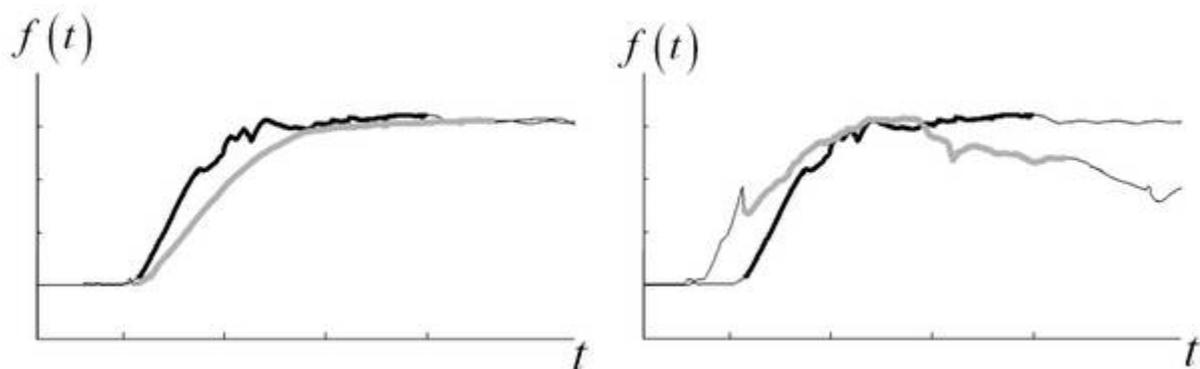
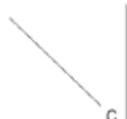
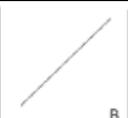
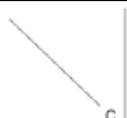
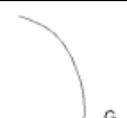
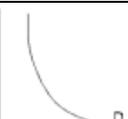
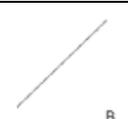
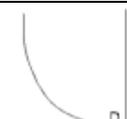
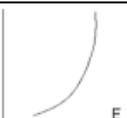
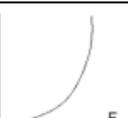
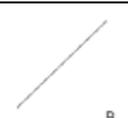
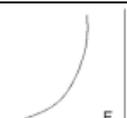
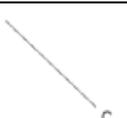
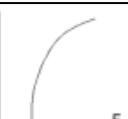
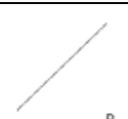
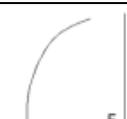
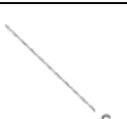
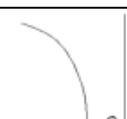
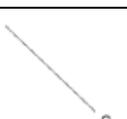


Abbildung 23 Ergebnisse für die Suche nach einer gekrümmter Episode.
Schwarz: Referenzzeitreihe
Grau: Ergebniszeitreihe

III.1.1.4.2 Probleme von Episodenübergängen

Bei der Kombination von linearen und gekrümmten Episoden kann es zu glatten Übergängen kommen. Ein Übergang ist „glatt“, wenn er ohne eine starke Krümmung erfolgt, welche eine stabile E-Kette aufweisen würde. Tabelle 8 zeigt die Möglichkeiten für glatte Übergänge beim Wechsel zwischen linearen und gekrümmten Episoden auf. In diesen Fällen ist es sehr wahrscheinlich, dass die E-Ketten, welche die Start- bzw. Endzeitpunkte der linearen Episoden markieren, mit den E-Ketten identisch sind, welche die angrenzende gekrümmte Episode charakterisieren. Es kommt zu einer starken Überlappung der beiden Episoden. In diesen Fällen ist also eine eindeutige Unterscheidung der aufeinander folgenden Episoden nicht möglich.

Tabelle 8 Möglichkeit eines glatten Episodenübergangs

  Nein	  Nein	  Ja
  Ja	  Ja	  Nein
  Nein	  Ja	  Nein
  Ja	  Nein	  Ja
  Ja	  Nein	  Ja
  Nein	  Ja	  Nein
  Ja	  Ja	  Nein
  Nein	  Nein	  Ja

III.1.1.4.3 Inkompatibilitäten mit Nutzerdialog

Im Rahmen einer interdisziplinären Bewertung des ersten Entwurfs und Abgleich mit dem parallel entstehenden Nutzerdialog wurden einige Inkonsistenzen zwischen Nutzerdialog und Suchalgorithmus deutlich (vgl. Tabelle 9).

Die von Kindsmüller (2006) vorgestellte und empirisch überprüfte Theorie zur *Trend-Literacy* wurde nicht konsequent umgesetzt. Während dort zwischen mindestens sieben verschiedenen Primitiven eines Trendlet-Alphabets (vgl. Kapitel II.2.3) unterschieden wird, ermöglicht der zunächst erstellte Entwurf I lediglich die Unterscheidung des Experten zwischen Linearität und Nicht-Linearität. Darüber hinaus ist keine weitere Spezifikation der Episodendynamik nötig, da diese implizit in den

Rohdaten der vom Nutzer beschriebenen Zeitreihe enthalten sind. Durch einen Vorzeichenvergleich der E-Ketten wird implizit zwischen konvexen und konkaven Trends unterschieden. Ob eine Episode aufsteigend oder abfallend verläuft, wird implizit durch den Zeitverlauf der E-Kette festgelegt.

Tabelle 9 Differenzen zwischen Entwürfen des Suchalgorithmus und Nutzerdialogs

Vergleichskriterium	Suchalgorithmus	Nutzerdialog
Abhängigkeit von Referenzkurve	ja	nein
Unterscheidung der Episoden	linear vs. gekrümmt	7 Trendlets
Berücksichtigung der Episodenlänge	nein (gekrümmt) indirekt (linear)	ja
Berücksichtigung der Episodenhöhe	nein (gekrümmt) indirekt (linear)	ja

Neben dieser konzeptuellen Abweichung vom Nutzerdialog bringt der Entwurf I noch eine weitere wesentliche Einschränkung mit sich. Durch die Orientierung an den Rohdaten der Referenzzeitreihe ist dem Nutzer keine abweichende oder abstrahierte Beschreibung der von ihm als relevant erachteten Prozessdynamik möglich. Er zeigt lediglich einen Zeitabschnitt auf, teilt diesen in einzelne Episoden ein und weist diesen eine Grunddynamik (linear vs. gekrümmt) zu. Einige Aussagen aus der Aufgabenanalyse (vgl. Kapitel II.2.5) zeigten jedoch ein starkes Interesse an einer Trendbeschreibung unabhängig von der Referenzzeitreihe. Schlussendlich ignoriert der Algorithmus die Längen- und Höhenspezifikation des Nutzers weitgehend. Lediglich die Längenin-formation wird annäherungsweise genutzt.

Zusammenfassend muss dem ersten Entwurf des Suchalgorithmus trotz guter Ansätze bei der Identifikation linearer und gekrümmter Episoden einer beliebigen Zeitreihe eine mangelhafte Abstimmung mit dem parallel entworfenen Interaktionskonzept des Nutzerdialogs bescheinigt werden. Aus diesem Grund wurde sich für eine konzeptuelle Überarbeitung des Suchalgorithmus entschieden, welche in der Erstellung des zweiten Entwurfs ihre Umsetzung fand (siehe Kapitel III.1.2).

III.1.2 Entwurf II

Auf Basis der Bewertung von Entwurf I (Kapitel III.1.1) wurde entschieden, den Entwurf des parallel entwickelten Nutzerdialogs beizubehalten und den Suchalgorithmus zu verändern. Analog zum Kapitel III.1.1 wird im Folgenden zunächst das Grundkonzept des Entwurf II dargestellt (Kapitel III.1.2.1), gefolgt von einer Dokumentation der Vorverarbeitung und der Suchprozedur (Kapitel III.1.2.2 und III.1.2.3).

Die Entwicklung des zweiten Entwurfs für den Suchalgorithmus stellte einen kontinuierlichen Prozess dar, in dessen Verlauf ca. 40 verschiedene testbare Versionen erstellt wurden. Der überwiegende Teil dieser Versionen diente zunächst der Veranschaulichung von Effekten der Faltung auf verschiedenste Zeitreihen. Spätere Versionen stellten Vorschläge verschiedener Such- und Vergleichsalgorithmen dar. Alle Versionen ermöglichten sowohl einen visuellen Zugang als auch einen statistischen Zugang zu den Zwischenergebnissen. Ersterer ermöglichte die Ableitung einiger Konfigurationsheuristiken (Kapitel III.1.2.4). Letzterer führte zu Ergebnissen explorativer Testläufe, welche im abschließenden Kapitel III.1.2.5 vorgestellt werden.

III.1.2.1 Grundkonzept

Die grundlegenden Zielstellungen des zweiten Entwurfs war zum ersten die Unabhängigkeit der Suchanfrage von den Rohdaten der Referenzzeitreihe und zum zweiten die sinnvolle Verwendung des kompletten Trendlet-Alphabets. Um diese Ziele zu erreichen, wurde sowohl die Identifikation der linearen als auch gekrümmten Episoden konzeptuell verändert. In Abbildung 24 ist eine, mit einer Gaußkurve gefaltete Zeitreihe zu sehen. Auf Basis der ersten und zweiten Ableitung lassen sich hierfür die Wendestellen und Extremstellen identifizieren. Die Menge dieser Punkte bildet den Index für die gekrümmten Episoden. Für die Bildung eines Index der linearen Episoden wird die gefaltete Zeitreihe einer einfachen Linearapproximation unterzogen.

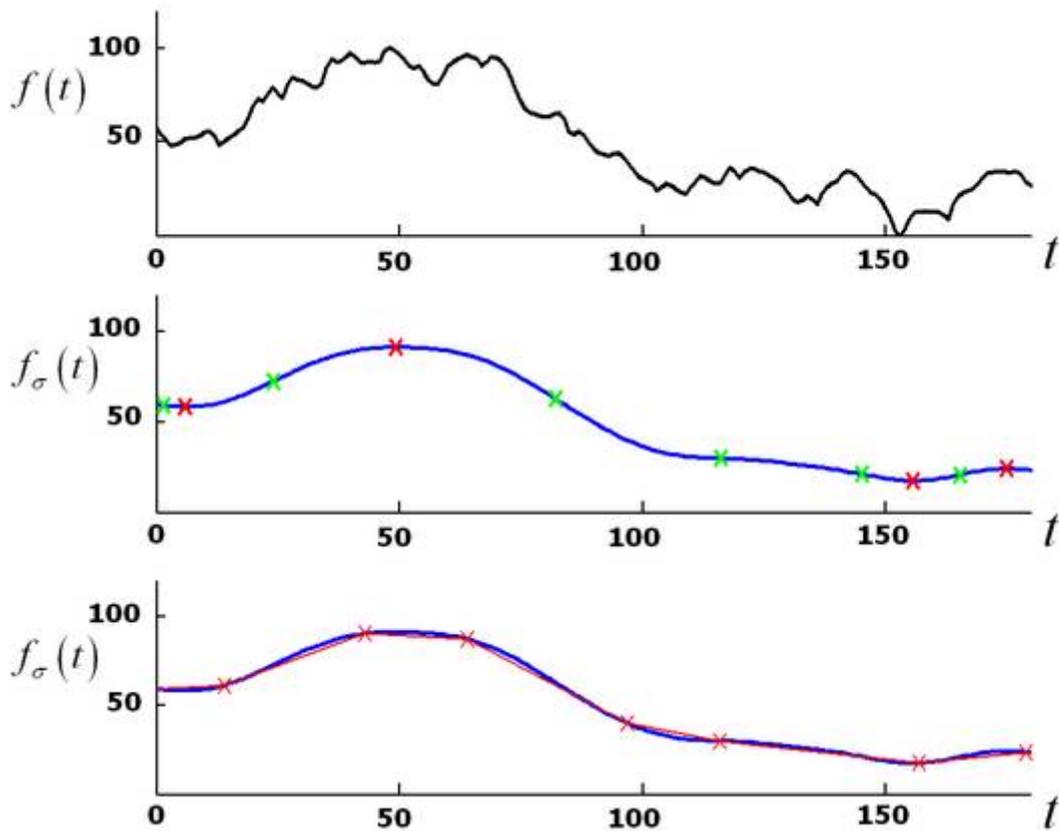


Abbildung 24 Oben: originale Zeitreihe. Mitte: gefaltete Zeitreihe mit Extremstellen (rot) und Wendestellen (grün).
 Unten: gefaltete Zeitreihe mit Linearapproximationen (rote Linien) und deren Start- und Endpunkte (rote Kreuze)

Je höher die Faltungsskala, desto mehr weichen die Amplituden der Extremstellen von den Amplituden der entsprechenden Originalfunktion ab. Auf der höchstmöglichen Faltungsskala stellt die resultierende Zeitreihe eine horizontal und monoton verlaufende Kurve dar. Da jedoch die Episodenhöhen als charakteristisches Vergleichskriterium zwischen Referenztrend und Ergebnistrend gebraucht werden, muss eine Lösung gefunden werden, welche die Daten der Originalkurve mit berücksichtigt. Eine Variante wäre, die Amplitude direkt aus dem originalen Zeitreihe zu entnehmen. Bei großem Rauschanteil führt dies jedoch zu einer stark zufälligen Variation der Episodenhöhen in Abhängigkeit der entsprechenden Längen. Die Bildung eines Mittelwertes für einen bestimmten Bereich um den Start- und Endpunkt einer Episode könnte dieses Problem lösen. Allerdings wäre ein weiterer, zunächst willkürlich festzulegender Parameter in Form einer prozen-

tualen zeitlichen Ausdehnung dieser Bereiche in Relation zur Ergebnis-Episode notwendig. Eine stabilere und elegantere Lösung ist die Standardisierung der gefalteten Funktionen an der originalen Zeitreihe (Gleichung 20).

Gleichung 20

$$\hat{f}_\sigma(t) = \frac{f_\sigma(t) - \bar{f}_\sigma}{SDf_\sigma} * SDf + \bar{f}$$

$$\text{mit } \bar{f}_\sigma = \frac{\sum_{t=1}^n f_\sigma(t)}{n}$$

$$\text{und } SDf_\sigma = \sqrt{\frac{\sum_{t=1}^n (f_\sigma(t) - \bar{f}_\sigma)^2}{n}}$$

$$\text{und } \bar{f} = \frac{\sum_{t=1}^n f(t)}{n}$$

$$\text{und } SDf = \sqrt{\frac{\sum_{t=1}^n (f(t) - \bar{f})^2}{n}}$$

Die Effekte dieser Standardisierung werden in Abbildung 25 verdeutlicht. Im oberen Diagramm bilden die Amplituden der gefalteten Zeitreihe (Faltungsskala 9) noch eine sehr gute Approximation der originalen Zeitreihe ab. Allerdings kann aufgrund des starken Rauschens keine Episode des Trendlets G identifiziert werden, welches ungefähr zwischen den Zeitpunkten 65 und 120 zu finden wäre. Eine solche Episode bietet sich jedoch für das menschliche Auge an, wie man im unteren Diagramm erkennen kann. Die mathematische, Computer gestützte Identifikation dieser Episode ist jedoch erst auf einer recht hohen Faltungsskala (in diesem Fall Faltungsskala 33) möglich. Mit der Faltung auf einer hohen Skala geht jedoch eine starke Glättung der originalen Zeitreihe einher (rote Linie). Damit geht die ursprüngliche Höheninformation verloren. Als Lösung dieses Dilemmas bietet eine Standardisierung nach Gleichung 20 wiederum eine sehr gute Approximation der Amplituden der originalen Zeitreihe.

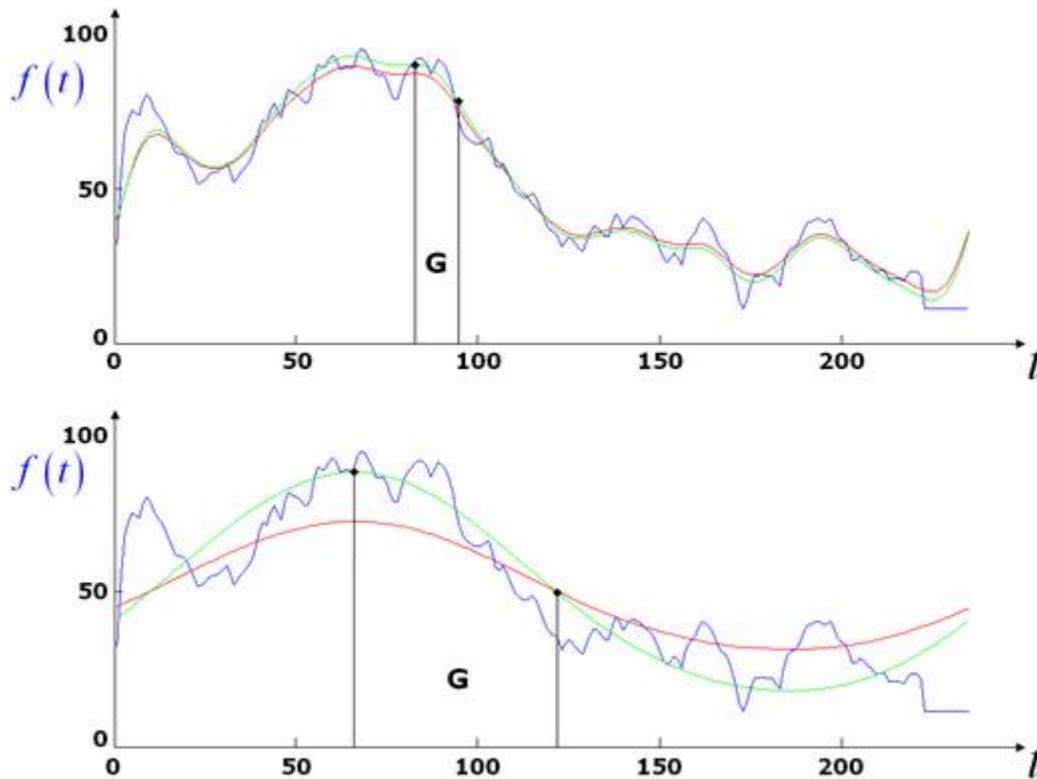


Abbildung 25 blau: originale Zeitreihe; rot: Zeitreihe auf originaler Faltungsskala; grün: standardisierte gefaltete Zeitreihe; oben: $\sigma = 9$; unten: $\sigma = 33$

III.1.2.2 Vorverarbeitung

Im Rahmen der Vorverarbeitung einer beliebigen Datenbank wird eine darin gespeicherte Zeitreihe $f(t)$ mit $t = \{1, \dots, n\}$ der Faltung aus Gleichung 4 auf allen Skalen eines spezifischen Skalenraums ζ unterzogen (Gleichung 21)^a. Für die gefalteten Zeitreihen jeder dieser Skalen wird nun je ein Index^b Z_σ generiert, welche die Start- und Endzeitpunkte gekrümmter Episoden, deren Trendlets sowie Amplituden enthält (Gleichung 23). Pro Faltungsskala σ wird für alle Linearitätskoeffizienten aus der Menge Λ (Gleichung 22) je eine Index ${}_\lambda L_\sigma$ an Start- und Endzeitpunkten, Trendlets und Amplituden linear Episoden erstellt (Gleichung 24). Die Gesamtheit aller Indexe Z_σ und ${}_\lambda L_\sigma$ ergibt den vollständigen Suchraum Γ_ζ (Gleichung 25).

Gleichung 21 $\zeta = \{\sigma_1, \dots, \sigma_p\}$ mit $p \in \mathbb{N}^*$

Gleichung 22 $\Lambda = \{\lambda_1, \dots, \lambda_q\}$ mit $p \in \mathbb{N}^*$ und $q \in \mathbb{N}^*$

^a Konfigurationsheuristiken für den Skalenraum der Vorverarbeitung finden sich im Kapitel III.1.2.4.1.

^b Unter Index wird hier ein Verzeichnis verstanden.

Gleichung 23 $Z_\sigma = \{T_\sigma, W_\sigma, H_\sigma\}$

Gleichung 24 ${}_\lambda L_\sigma = \{{}_\lambda W_\sigma, {}_\lambda H_\sigma\}$

Gleichung 25 $\Gamma = \{Z_\sigma, {}_\lambda L_\sigma\}$

III.1.2.2.1 Identifikation gekrümmter Episoden

Die Nullstellen der ersten und zweiten Ableitung $f'_\sigma(t)$ und $f''_\sigma(t)$ kennzeichnen die Start- und Endzeitpunkte gekrümmter Episoden, und werden für jede Faltungsskala σ in dem Index W_σ zusammen gefasst (Gleichung 26). Aufgrund der diskreten Schrittweite von $f'_\sigma(t)$ und $f''_\sigma(t)$ werden oftmals Nullstellen übersprungen. Deshalb müssen diese Nullstellen nach Gleichung 27 und Gleichung 28 konstruiert werden. Basierend auf dem Index W_σ wird dann der Index T_σ konstruiert (Gleichung 29 bis Gleichung 31). Dieser enthält die Vorzeichen der ersten und zweiten Ableitung für alle Episoden auf einer bestimmten Faltungsskala σ . Zur vollständigen Abbildung der originalen Zeitreihe in Form von Episoden wird zuletzt der Index H_σ generiert (Gleichung 32), welcher die Amplituden der einzelnen Episoden enthält. Dafür wird auf die standardisierte gefaltete Zeitreihe aus Gleichung 20 zurück gegriffen.

Gleichung 26 $W_\sigma = (\bar{t}_1, \dots, \bar{t}_m) \mid \bar{t} \in \{1 \cup \{t \in \{2, \dots, n-1\} \mid f'_\sigma(t) * f''_\sigma(t) = 0\} \cup n\}$

Gleichung 27 $f'_\sigma(t) = \begin{cases} f'_\sigma(t) & \text{falls } f'_\sigma(t) * f'_\sigma(t+1) \geq 0 \\ 0 & \text{falls } f'_\sigma(t) * f'_\sigma(t+1) < 0 \end{cases}$

Gleichung 28 $f''_\sigma(t) = \begin{cases} f''_\sigma(t) & \text{falls } f''_\sigma(t) * f''_\sigma(t+1) \geq 0 \\ 0 & \text{falls } f''_\sigma(t) * f''_\sigma(t+1) < 0 \end{cases}$

Gleichung 29 $T_\sigma = \left(\begin{array}{l} \left(\text{sgn}(f'_\sigma(\bar{t}_1 + 1)), \text{sgn}(f''_\sigma(\bar{t}_1 + 1)) \right), \\ \dots, \\ \left(\text{sgn}(f'_\sigma(\bar{t}_{m-1} + 1)), \text{sgn}(f''_\sigma(\bar{t}_{m-1} + 1)) \right) \end{array} \right) \mid \bar{t} \in W_\sigma$

Gleichung 30 $\text{sgn}(f'_\sigma(\bar{t} + 1)) = \begin{cases} +1 & \text{falls } f'_\sigma(\bar{t} + 1) > 0 \\ 0 & \text{falls } f'_\sigma(\bar{t} + 1) = 0 \\ -1 & \text{falls } f'_\sigma(\bar{t} + 1) < 0 \end{cases}$

$$\text{Gleichung 31} \quad \text{sgn}\left(f''_{\sigma}(\bar{t}+1)\right) = \begin{cases} +1 & \text{falls } f''_{\sigma}(\bar{t}+1) > 0 \\ 0 & \text{falls } f''_{\sigma}(\bar{t}+1) = 0 \\ -1 & \text{falls } f''_{\sigma}(\bar{t}+1) < 0 \end{cases}$$

$$\text{Gleichung 32} \quad \mathbf{H}_{\sigma} = \left(\hat{f}_{\sigma}(\bar{t}_1), \dots, \hat{f}_{\sigma}(\bar{t}_m) \right) \mid \bar{t} \in \mathbf{W}_{\sigma}$$

III.1.2.2.2 Identifikation linearer Episoden

Die Start- und Endzeitpunkte linearer Episoden werden durch sogenannte *dominant points* gekennzeichnet (Horst & Beichl, 1996). Nach folgender Prozedur werden diese Punkte ermittelt und für alle Linearitätskoeffizienten λ auf allen Skalen des Skalenraums σ in den Indexen ${}_{\lambda}\mathbf{W}_{\sigma}$ und ${}_{\lambda}\mathbf{H}_{\sigma}$ gespeichert (Gleichung 37 und Gleichung 38):

1. Speichere in den Indexen ${}_{\lambda}\mathbf{W}_{\sigma}$ und ${}_{\lambda}\mathbf{H}_{\sigma}$ die Koordinaten des Startpunktes der Zeitreihe $\hat{f}_{\sigma}(t)$ als ersten *dominant point* \bar{t}_1 .
2. Berechne für die folgenden Punktepaare $(\hat{f}_{\sigma}(t-1), \hat{f}_{\sigma}(t))$ die jeweiligen Sehnenlängen \bar{c} nach Gleichung 33.
3. Kumuliere die Sehnenlängen \bar{c} zur Bogenlänge s (Gleichung 34) und berechne gleichzeitig die Gesamtsehnenlänge c des jeweiligen Zeitreihenabschnitts (Gleichung 35).
4. Sobald das Verhältnis zwischen Bogenlänge s und Gesamtsehnenlänge c den Wert eines bestimmten Linearitätskoeffizienten λ übersteigt (Gleichung 36), speichere in den Indexen ${}_{\lambda}\mathbf{W}_{\sigma}$ und ${}_{\lambda}\mathbf{H}_{\sigma}$ die Koordinaten des voran gegangenen Punktes $t-1$ als den nächsten *dominant point* \bar{t}_2 (dieser Punkt ist gleichzeitig der Endzeitpunkt der erstellten linearen Episode und Startzeitpunkt der nächstfolgenden linearen Episode).
5. Wiederhole die Schritte 1 bis 4 bis zum Ende ($t = n$) der standardisierten Faltungsfunktion $\hat{f}_{\sigma}(t)$.

Eine graphische Abbildung der Prozedur findet sich in Abbildung 26.

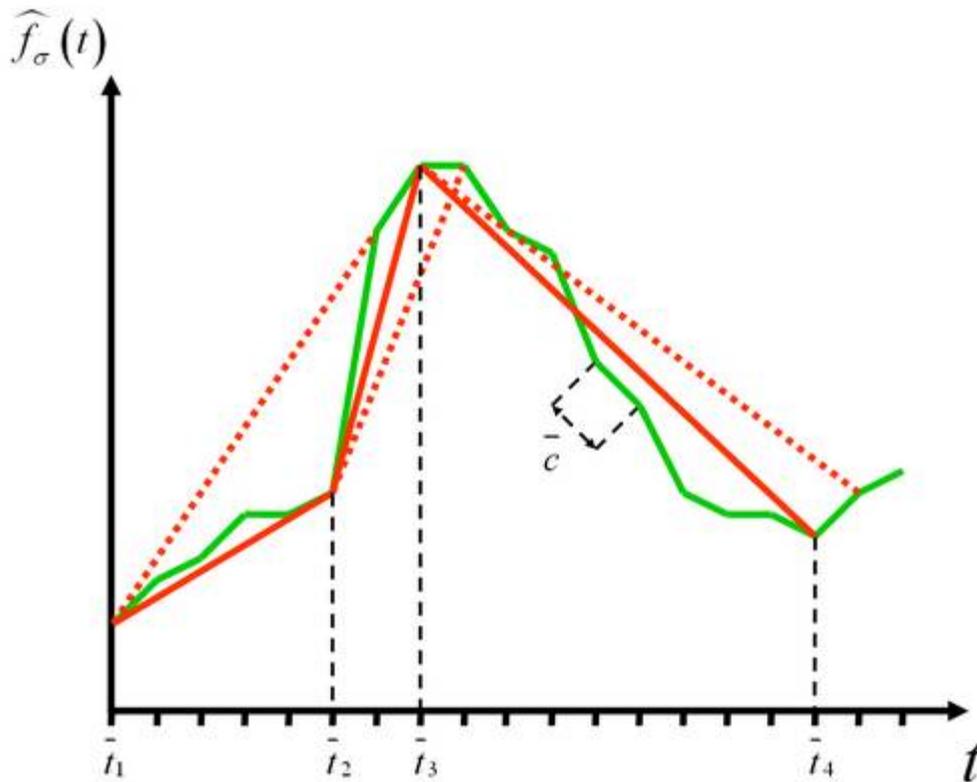


Abbildung 26 Linearapproximation. rot/gedoppelt: c mit $\lambda_t > \lambda$ rot: c mit $\lambda_t \leq \lambda$
grün: s

Gleichung 33
$$\bar{c} = \sqrt{1 + (\hat{f}_\sigma(t) - \hat{f}_\sigma(t-1))^2}$$

Gleichung 34
$$s = \sum_{k=t}^t \bar{c}_k$$

Gleichung 35
$$c = \sqrt{(t - \bar{t})^2 + (\hat{f}_\sigma(t))^2}$$

Gleichung 36
$$\bar{t} = t - 1 \text{ sobald } \frac{1}{2} \sqrt{s^2 - c^2} > \lambda$$

Gleichung 37
$${}_\lambda W_\sigma = (\bar{t}_1, \dots, \bar{t}_m)$$

Gleichung 38
$${}_\lambda H_\sigma = \{\hat{f}_\sigma(\bar{t}) \mid \bar{t} \in {}_\lambda W_\sigma\}$$

Die Wahl verschiedener Linearitätskoeffizienten ist notwendig, da eine eindeutige Definition von Linearität nicht in allgemein gültiger Form gegeben werden kann. Vielmehr sollte es dem Nutzer überlassen werden, diese Definition anhand eines eindimensionalen Parameters zu stellen (vgl.

Kapitel III.1.2.4.2). Abbildung 27 verdeutlicht, welchen Einfluss verschiedene Linearitätskoeffizienten auf verschiedenen Faltungsskalen bei der Bildung einer Episodenfolge haben. Die Approximationen bieten für alle drei Linearitätskoeffizienten jeweils Ergebnisse an, die für das menschliche Auge als die lokal beste Approximation gelten kann (violett: $\lambda = 3$; orange: $\lambda = 5$; grün: $\lambda = 10$).

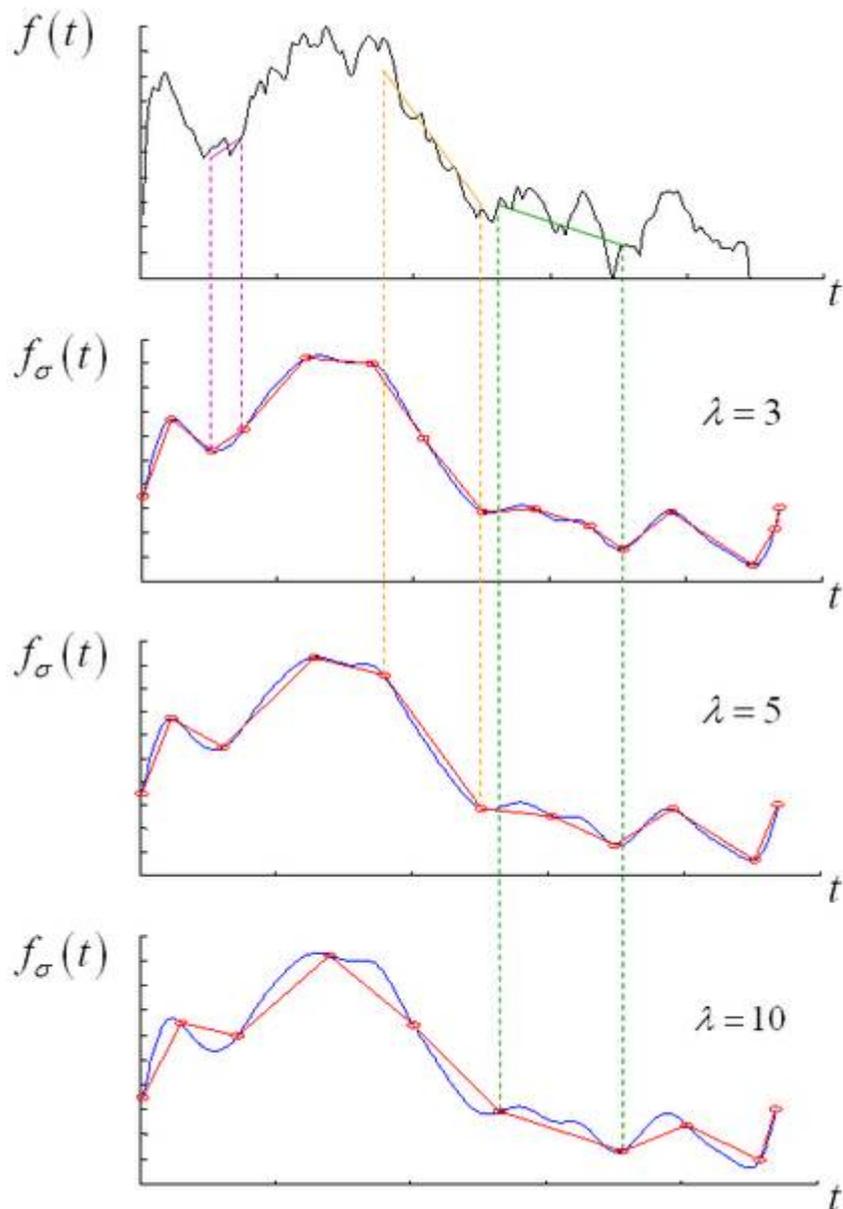


Abbildung 27 Einfluss der Linearitätskoeffizienten auf Linearapproximation.
 Oben: originale Zeitreihe
 Blau: gefaltete Zeitreihe auf Faltungsskala 6
 Rot: Linearapproximationen für unterschiedliche Linearitätskoeffizienten

III.1.2.3 Suchprozedur

Zwei verschiedene Informationseingänge werden für die Suchprozedur (Abbildung 28) gebraucht. Der erste Eingang ist der Suchraum Γ als Ergebnis der Vorverarbeitung (Kapitel III.1.2.2). Der zweite Eingang besteht aus dem Inhalt der Suchanfrage^a. Diese enthält:

- den Referenztrend Ω (Gleichung 39 und Gleichung 40)
- den Linearitätsparameter λ_{Ω} (Gleichung 41)
- die Menge Φ an prozentualen Überlappungs-, Abstands-, und Verschiebungstoleranzen benachbarter Episoden (Gleichung 46 bis Gleichung 48)
- die Gewichtungs- und Sensitivitätsparameter für alle Ähnlichkeitskriterien (Gleichung 53 und Gleichung 54)
- die Redundanztoleranz r_{tol} (siehe Kapitel III.1.2.3.4)

Ein Referenztrend Ω wird durch eine Abfolge an Episoden gekennzeichnet^b. Die Beschreibung Ω_e jeder Episode enthält ein Trendlet, die Episodenlänge w_e , die Episodenhöhe h_e , deren Toleranzen w_{ε_e} und h_{ε_e} , sowie im Falle einer gekrümmten Episode die Krümmungsstärke v_e (Gleichung 40).

Gleichung 39
$$\Omega = (\Omega_1, \dots, \Omega_{\max(e)}) \text{ mit } e \in \mathbb{N}^+$$

$$\Omega_e = \{T_e, w_e, w_{\varepsilon_e}, h_e, h_{\varepsilon_e}, v_e\} \text{ mit } T_e = \{A, B, C, D, E, F, G\}$$

$$\text{und } w_e, w_{\varepsilon_e}, h_{\varepsilon_e} \in \mathbb{R}^+$$

Gleichung 40

$$\text{und } w_e > w_{\varepsilon_e}$$

$$\text{und } h_e \in \mathbb{R}$$

$$\text{und } v_e \in \{0, \dots, 1\}$$

Der für die Einschränkung des Suchraums (Kapitel III.1.2.3.1) genutzte Linearitätsparameter λ_{Ω} wird für den gesamten Referenztrend angegeben.

Gleichung 41
$$\lambda_{\Omega} > 0 \text{ und } \lambda_{\Omega} \in \mathbb{R}^+$$

^a Alle diese Daten werden vom Nutzer generiert bzw. als Voreinstellung übernommen.

^b Der Laufindex e kennzeichnet hier die Stelle der Episode im Gesamttrend.

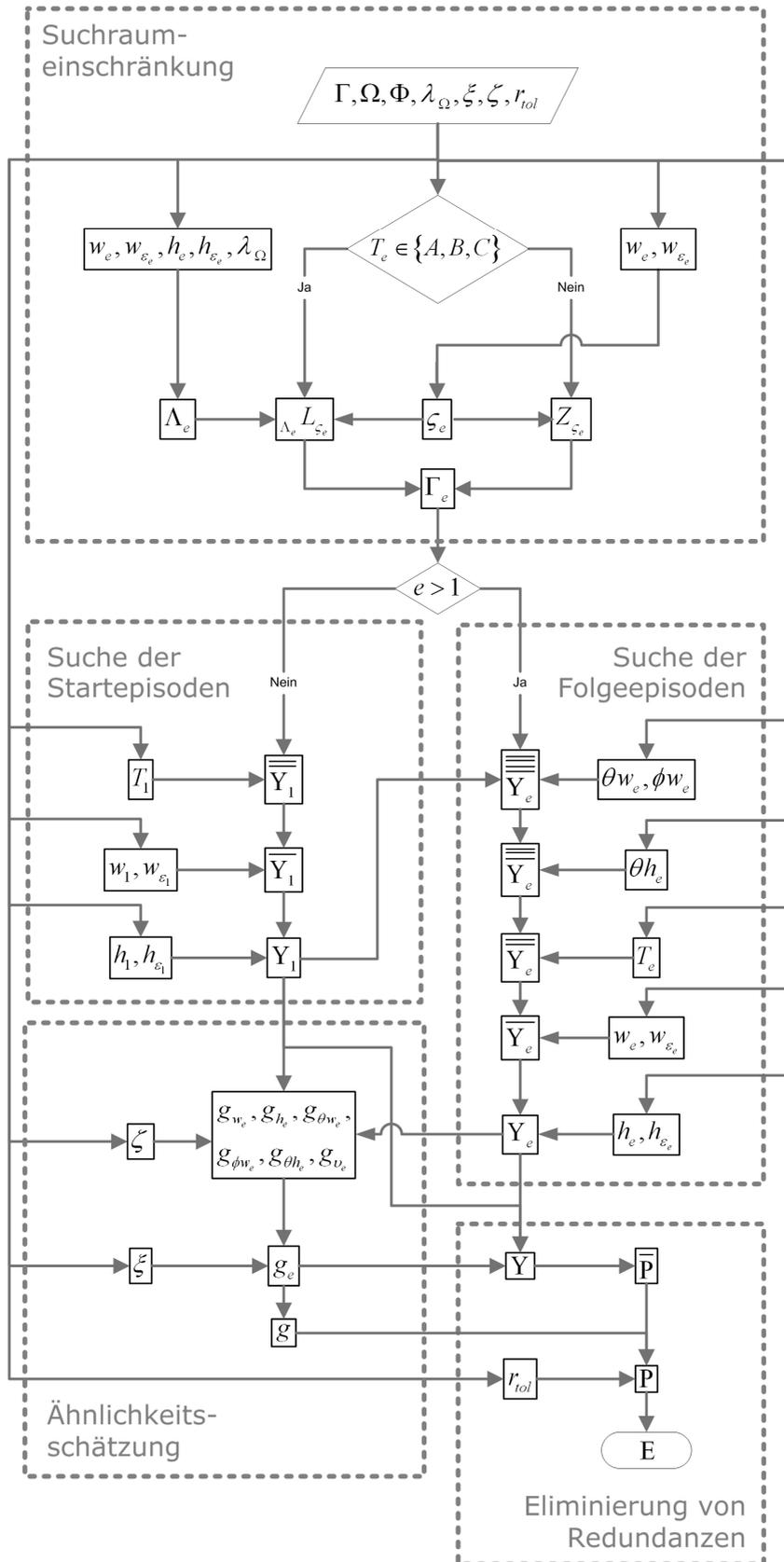


Abbildung 28 Überblick über die Suchprozedur^a

^a Eine Kurzbeschreibung aller Formelzeichen findet sich im Zeichenglossar am Ende dieser Dissertationsschrift.

Lambda ist nicht automatisch generierbar, da die Unabhängigkeit der Suchanfrage von der Referenzzeitreihe gewahrt bleiben muss (vgl. Kapitel III.1.1.4.3). Lambda wird darum als Parameter eingestellt und kann durch Nutzer verändert werden (siehe Kapitel III.1.2.4). Allerdings ist eine Anpassung an die Längen und Höhen der einzelnen Episoden notwendig, da das Ergebnis aus Gleichung 36 stark mit dem Größenmaßstab der Episoden variiert. Tabelle 10 macht dies an einem Rechenbeispiel deutlich.

Tabelle 10 Variation des Größenmaßstabs

w_e	h_e	$c_e = \sqrt{(w_e)^2 + (h_e)^2}$	$\max(s_e) = w_e + h_e$	$\frac{1}{2} \sqrt{(\max(s_e))^2 - (c_e)^2}$
2	1	2,2	3	1,0
20	10	22,4	30	10,0
200	100	223,6	300	100,0

Hinzu kommen die Variationsmöglichkeiten aufgrund der Längen- und Höhentoleranzen einzelner Referenz-Episoden. Auf Basis der Suchparameter $w_e, w_{\epsilon_e}, h_e, h_{\epsilon_e}$ und λ_Ω werden deshalb für alle Referenz-Episoden spezifische Schwellwerte $\{\min(\lambda_e), \max(\lambda_e)\}$ für die Auswahl der Linearitätskoeffizienten des eingeschränkten Suchraums erstellt (Gleichung 45 bis Gleichung 42).

Gleichung 42
$$\min(\lambda_e) = \lambda_\Omega * \min(c_e)$$

Gleichung 43
$$\min(c_e) = \sqrt{(w_e - w_{\epsilon_e})^2 + (h_e - h_{\epsilon_e})^2}$$

Gleichung 44
$$\max(\lambda_e) = \lambda_\Omega * \max(c_e)$$

Gleichung 45
$$\max(c_e) = \sqrt{(w_e + w_{\epsilon_e})^2 + (h_e + h_{\epsilon_e})^2}$$

Die Toleranzen, welche die möglichen Position der Folgeepisoden auf die zuvor identifizierte Ergebnis-Episode einschränken, werden im Rahmen der Suchanfrage alle in Prozent angegeben, wobei 100% in diesem Fall durch den Wert Eins gekennzeichnet wird.

Gleichung 46
$$\theta w = (\theta w_2, \dots, \theta w_{\max(e)}) \text{ mit } \theta w \in \{0, \dots, 1\}$$

Gleichung 47 $\phi w = (\phi w_2, \dots, \phi w_{\max(e)})$ mit $\phi w \in \{0, \dots, 1\}$

Gleichung 48 $\theta h = (\theta h_2, \dots, \theta h_{\max(e)})$ mit $\theta h \in \{0, \dots, 1\}$

Gleichung 49 $\Phi = \{\theta w, \phi w, \theta h\}$

Die prozentual angegebene horizontale Überlappungstoleranz wird auf die kürzere der beiden aufeinanderfolgenden Referenz-Episoden bezogen und nach Gleichung 50 in die absolute Toleranz $\widehat{\theta w}_e$ umgewandelt.

Gleichung 50
$$\widehat{\theta w}_e = \begin{cases} w_e * \theta w_e & \text{falls } w_e \leq w_{e-1} \\ w_{e-1} * \theta w_e & \text{falls } w_e > w_{e-1} \end{cases}$$

Die prozentuale Toleranz für den maximal zulässigen Abstand zweier aufeinander folgender Episoden wird an der längeren zweier aufeinander folgender Referenz-Episoden relativiert. Der maximale absolute Episodenabstand $\widehat{\phi w}_e$ errechnet sich aus Gleichung 51.

Gleichung 51
$$\widehat{\phi w}_e = \begin{cases} w_e * \phi w_e & \text{falls } w_e > w_{e-1} \\ w_{e-1} * \phi w_e & \text{falls } w_e \leq w_{e-1} \end{cases}$$

Zur Berechnung der absoluten vertikalen Überlappungstoleranz $\widehat{\theta h}_e$ wird auf den größeren Betrag der Episodenhöhe zweier aufeinander folgender Episoden heran gezogen.

Gleichung 52
$$\widehat{\theta h}_e = \begin{cases} |h_e| * \theta h_e & \text{falls } h_e > h_{e-1} \\ |h_{e-1}| * \theta h_e & \text{falls } h_e \leq h_{e-1} \end{cases}$$

Für die Schätzung der Ähnlichkeit von Ergebnis- und Referenz-Episoden wird auf die Kriterien Episodenlänge, Episodenhöhe, Episodenüberlappung, Episodenabstand, vertikaler Episodenverschiebung sowie Krümmungsstärke (nur bei gekrümmten Episoden) zurück gegriffen. Alle diese Kriterien können unterschiedlich starken Einfluss auf die Ähnlichkeitsschätzung in Form einer Gewichtung **G** haben (Gleichung 53), welche durch den Nutzer im Rahmen der Suchanfrage definiert wird^a. Neben der Gewichtung kann zudem die Sensitivität **S** der einzelnen Ähnlichkeitskri-

^a Gewichtungen werden beim Prototypen in Form einer Voreinstellung vorgeschlagen und sind optional vom Nutzer veränderbar.

terien unabhängig voneinander eingestellt werden (Gleichung 54)^a. Auf die Effekte unterschiedlicher Sensitivitäten auf die Berechnung der Ähnlichkeitsindikatoren wird in Kapitel III.1.2.3.5 näher eingegangen.

$$\text{Gleichung 53} \quad \xi = \{G_w, G_h, G_{\theta_w}, G_{\phi_w}, G_{\theta_h}, G_v\} \quad \text{mit } G = \{0, \dots, 1\}$$

$$\text{Gleichung 54} \quad \zeta = \{S_w, S_h, S_{\theta_w}, S_{\phi_w}, S_{\theta_h}, S_v\} \quad \text{mit } S = \{0, \dots, 1\}$$

III.1.2.3.1 Suchraumeinschränkung

Für jede Episode des Referenzrends $\bar{\Omega}$ wird zunächst der Suchraum Γ_e in Abhängigkeit vom Grund-Typ des verwendeten Trendlets (linear vs. gekrümmt) bestimmt^b.

$$\text{Gleichung 55} \quad \Gamma_e = \{\gamma_1, \dots, \gamma_u\} = \begin{cases} \Lambda_e L_{\zeta_e} & \text{falls } T_e \in \{A, B, C\} \\ Z_{\zeta_e} & \text{falls } T_e \in \{D, E, F, G\} \end{cases}$$

Der Skalenraum ζ aus der Vorverarbeitung wird zur Steigerung der Sucheeffizienz weiter eingegrenzt, und zwar auf Basis der Länge der Referenz-Episode Ω_e . Der resultierende Skalenraum ζ_e enthält alle Skalen σ aus dem Skalenraum ζ , für die mindestens eine indexierte Episode vorliegt, welche unter Berücksichtigung der Längentoleranzen länger ist als die kürzeste mögliche Ergebnis-Episode und kürzer als die längste mögliche Ergebnis-Episode.

$$\text{Gleichung 56} \quad \zeta_e = \left\{ \sigma \in \zeta \mid \exists \bar{t} \in W_\sigma \mid (w_e - w_{\varepsilon_e}) \leq (\bar{t} - (\bar{t} - 1)) \leq (w_e + w_{\varepsilon_e}) \right\}$$

Der Parameter ζ definiert den eingeschränkten Suchraum sowohl für die gekrümmten als auch für die linearen Episoden. Im Falle einer linearen Episode e wird der Suchraum jedoch zusätzlich durch die Menge an Linearitätskoeffizienten bestimmt, welche anhand der Berechnungen aus Gleichung 45 bis Gleichung 42 weiter eingegrenzt werden kann.

^a Sensitivitätsparameter werden beim Prototypen wie auch die Gewichtungen voreingestellt und können fakultativ vom Nutzer verändert werden.

^b Der Index u kennzeichnet die maximale Anzahl an Suchindexen des eingeschränkten Suchraums Γ_e . Dieser wird durch die Parameter ζ_e und Λ_e definiert.

Tabelle 11 Beispiel für Suchraumeinschränkung

	Vorverarbeitung	Suchraum- einschränkung (lineare Episode)	Suchraum- einschränkung (gekrümmte Episode)
Skalenraum	$\zeta = \{5, 10, \dots, 40\}$	$\zeta_e = \{15, 20, 25, 30\}$	$\zeta_e = \{25, 30, 25, 40\}$
Anzahl an Faltungsskalen	$p = 8$	$p = 4$	$p = 4$
Linearitätsindex	$\Lambda = \{5, 10, 15, 20\}$	$\Lambda_e = \{5, 10\}$	$\Lambda_e = \emptyset$
Anzahl an Linearitätskoeffizienten	$q = 4$	$q = 2$	$q = \emptyset$
Gesamtanzahl an Suchindizes	$u = p * q = 32$	$u = p * q = 8$	$u = p = 4$

Der eingeschränkte Linearitätsindex Λ_e enthält alle Linearitätskoeffizienten, die zwischen dem oberen und unteren Schwellwert $\{\min(\lambda_e), \max(\lambda_e)\}$ liegen. Existiert kein solcher Linearitätskoeffizient, werden die beiden Linearitätskoeffizienten gewählt, welche in der direkten Nachbarschaft von $\{\min(\lambda_e), \max(\lambda_e)\}$ liegen. Eine beispielhafte Suchraumeinschränkung zeigt Tabelle 11.

Gleichung 57

$$\Lambda_e = \begin{cases} \min(\lambda_e) \leq \{\lambda\} \leq \max(\lambda_e) \\ \{\min(\lambda) \mid \lambda > \max(\lambda_e), \max(\lambda) \mid \lambda < \min(\lambda_e)\} \end{cases}$$

mit $\lambda \in \Lambda$

III.1.2.3.2 Suche der Startepisoden

Innerhalb des eingeschränkten Suchraums $\Gamma_1 = \{\gamma_1, \dots, \gamma_u\}$ werden zunächst alle Episoden gesucht, die das Trendlet der entsprechenden Referenz-Episode Ω_1 aufweisen. Startzeitpunkt und Trendlet einer potentiellen Ergebnis-Episode werden in der Menge \overline{Y}_1 zusammen getragen.

$$\text{Gleichung 58} \quad \overline{\overline{Y_1}} = \{T_1, a_1\} \left| \begin{array}{l} a_1 \in W_{\varepsilon_1} \quad \text{falls } T_1 \in \{D, E, F, G\} \\ a_1 \in {}_{\Lambda_1}W_{\varepsilon_1} \quad \text{falls } T_1 \in \{A, B, C\} \\ T_1 \in T_{\varepsilon_1} \quad \text{falls } T_1 \in \{D, E, F, G\} \\ \\ T_1 = \begin{cases} (-1, +1) \text{ falls } T_1 = D \\ (+1, +1) \text{ falls } T_1 = E \\ (+1, -1) \text{ falls } T_1 = F \\ (-1, -1) \text{ falls } T_1 = G \end{cases} \\ \\ T_1 = \begin{cases} (0, 0) \quad \text{falls } \alpha_1 < \alpha_{\text{tol}} \\ (+1, 0) \quad \text{falls } \alpha_1 > 0 \\ (-1, 0) \quad \text{falls } \alpha_1 < 0 \end{cases} \end{array} \right.$$

Obwohl zu diesem Zeitpunkt die Episodenlänge noch nicht überprüft wird, kann auf Basis der im jeweiligen Suchindex ${}_{\Lambda_1}W_{\varepsilon_1}$ verzeichneten Folgepunkte b_1 potentieller Startzeitpunkte a_1 der Steigungswinkel α_1 berechnet werden.

$$\text{Gleichung 59} \quad \alpha_1 = \arctan\left(\frac{\hat{f}_{\sigma}(b_1)}{b_1 - a_1}\right) \quad \text{mit } b_1 = (a_1 + 1) \in {}_{\Lambda_1}W_{\varepsilon_1}$$

Im Falle einer stationären Episode mit dem Trendlet A definiert das Verhältnis aus Höhentoleranz und Episodenlänge (inklusive Toleranz) die erlaubte Abweichung von der Steigung Null.

$$\text{Gleichung 60} \quad \alpha_{\text{tol}} = \arctan\left(\frac{h_{\varepsilon_1}}{w_1 - w_{\varepsilon_1}}\right)$$

Anschließend wird die potentielle Ergebnis-Episode auf Basis der Parameter Episodenlänge (Gleichung 61) und Episodenhöhe (Gleichung 62) überprüft und gegebenenfalls verworfen.

$$\text{Gleichung 61} \quad \overline{Y_1} = \{T_1, a_1, b_1\} \left| \begin{array}{l} \{T_1, a_1\} \in \overline{\overline{Y_1}} \\ b_1 = (a_1 + 1) \in {}_{\Lambda}W_{\varepsilon} \cup W_{\varepsilon} \\ (w_1 - w_{\varepsilon_1}) \leq (b_1 - a_1) \leq (w_1 + w_{\varepsilon_1}) \end{array} \right.$$

$$\text{Gleichung 62} \quad Y_1 = \left\{ T_1, a_1, b_1, \hat{f}_\sigma(a_1), \hat{f}_\sigma(b_1) \right\} \left| \begin{array}{l} \{T_1, a_1, b_1\} \in \overline{Y_1} \\ (\hat{f}_\sigma(a_1), \hat{f}_\sigma(b_1)) \in {}_\Lambda H_\zeta \cup H_\zeta \\ (h_1 - h_{\varepsilon_1}) \leq (\hat{f}_\sigma(b_1) - \hat{f}_\sigma(a_1)) \leq (h_1 + h_{\varepsilon_1}) \end{array} \right.$$

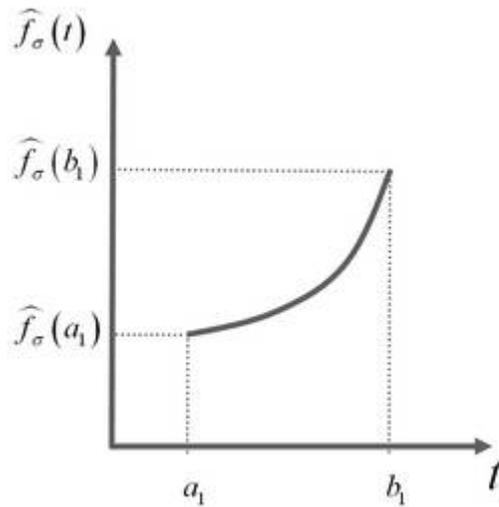


Abbildung 29 Veranschaulichung einzelner Komponenten aus Gleichung 62

III.1.2.3.3 Suche der Folgeepisoden

In Analogie zu Kapitel 0 wird zunächst der Suchraum für die Suche nach der nächstfolgenden Episode gewählt, und zwar unabhängig von der Wahl des Suchraums der vorangegangenen Episodensuche. Dort werden potentielle Folgeepisoden in Analogie zur Identifikation der Startepisode lokalisiert. Allerdings werden potentielle Startpunkte nun nicht mehr im gesamten Index, sondern unter Berücksichtigung der Überlappungs- und Abstandstoleranzen in der unmittelbaren Umgebung der bereits identifizierten Startepisoden gesucht (Gleichung 63 und Gleichung 64). Dafür werden die bereits transformierten Toleranzen $\widehat{\theta w}_e$ (Episodenüberlappung), $\widehat{\phi w}_e$ (Episodenabstand) und $\widehat{\theta h}_e$ (vertikale Episodenverschiebung) genutzt (vgl. Gleichung 50 bis Gleichung 52). Die Menge an potentiellen Folgeepisoden wird nun weiterhin sukzessive eingegrenzt. Zunächst wird die Übereinstimmung der Trendlets (Gleichung 65) und danach die Längen- und Amplitudenübereinstimmung (Gleichung 66 und Gleichung 67) überprüft. Kann keine Folgeepisode gefunden

werden, wird die Gesamte Kette an vorangegangenen Episoden als potentieller Ergebnistrend verworfen.

$$\text{Gleichung 63} \quad \overline{\overline{\overline{Y_e}}} = \{a_e\} \left| \begin{array}{l} a_e \in W_{c_e} \cup \wedge_e W_{c_e} \\ b_{e-1} - \widehat{\theta w_e} \leq a_e \leq b_{e-1} + \widehat{\phi w_e} \end{array} \right.$$

$$\text{Gleichung 64} \quad \overline{\overline{\overline{Y_e}}} = \{a_e, \widehat{f}_\sigma(a_e)\} \left| \begin{array}{l} a_e \in \overline{\overline{\overline{Y_e}}} \\ \widehat{f}_\sigma(a_e) \in \wedge H_c \cup H_c \\ |\widehat{f}_\sigma(a_e) - \widehat{f}_\sigma(b_{e-1})| \leq \widehat{\theta h_e} \end{array} \right.$$

$$\text{Gleichung 65} \quad \overline{\overline{\overline{Y_e}}} = \{T_e, a_e, \widehat{f}_\sigma(a_e)\} \left| \begin{array}{l} \{a_e, \widehat{f}_\sigma(a_e)\} \in \overline{\overline{\overline{Y_e}}} \\ T_e \in T_{c_e} \text{ falls } T_e \in \{D, E, F, G\} \\ \\ T_e = \begin{cases} (-1, +1) \text{ falls } T_e = D \\ (+1, +1) \text{ falls } T_e = E \\ (+1, -1) \text{ falls } T_e = F \\ (-1, -1) \text{ falls } T_e = G \end{cases} \\ \\ T_e = \begin{cases} (0, 0) \text{ falls } \alpha_e < \alpha_{tol} \\ (+1, 0) \text{ falls } \alpha_e > 0 \\ (-1, 0) \text{ falls } \alpha_e < 0 \end{cases} \end{array} \right.$$

Die Berechnung von α_e und α_{tol} erfolgt analog zu Gleichung 59 und Gleichung 60, welche zur Suche der Startepisoden verwendet wurden.

$$\text{Gleichung 66} \quad \overline{\overline{\overline{Y_e}}} = \{T_e, a_e, b_e, \widehat{f}_\sigma(a_e)\} \left| \begin{array}{l} \{T_e, a_e, \widehat{f}_\sigma(a_e)\} \in \overline{\overline{\overline{Y_e}}} \\ b_e \in \wedge W_c \cup W_c \\ (w_e - w_{\varepsilon_e}) \leq (b_e - a_e) \leq (w_e + w_{\varepsilon_e}) \end{array} \right.$$

$$\text{Gleichung 67} \quad Y_e = \left\{ \begin{array}{l} T_e, a_e, b_e, \\ \widehat{f}_\sigma(a_e), \widehat{f}_\sigma(b_e) \end{array} \right\} \left| \begin{array}{l} \{T_e, a_e, b_e, \widehat{f}_\sigma(a_e)\} \in \overline{\overline{\overline{Y_e}}} \\ \{\widehat{f}_\sigma(a_e), \widehat{f}_\sigma(b_e)\} \in \wedge H_c \cup H_c \\ (h_e - h_{\varepsilon_e}) \leq (\widehat{f}_\sigma(b_e) - \widehat{f}_\sigma(a_e)) \leq (h_e + h_{\varepsilon_e}) \end{array} \right.$$

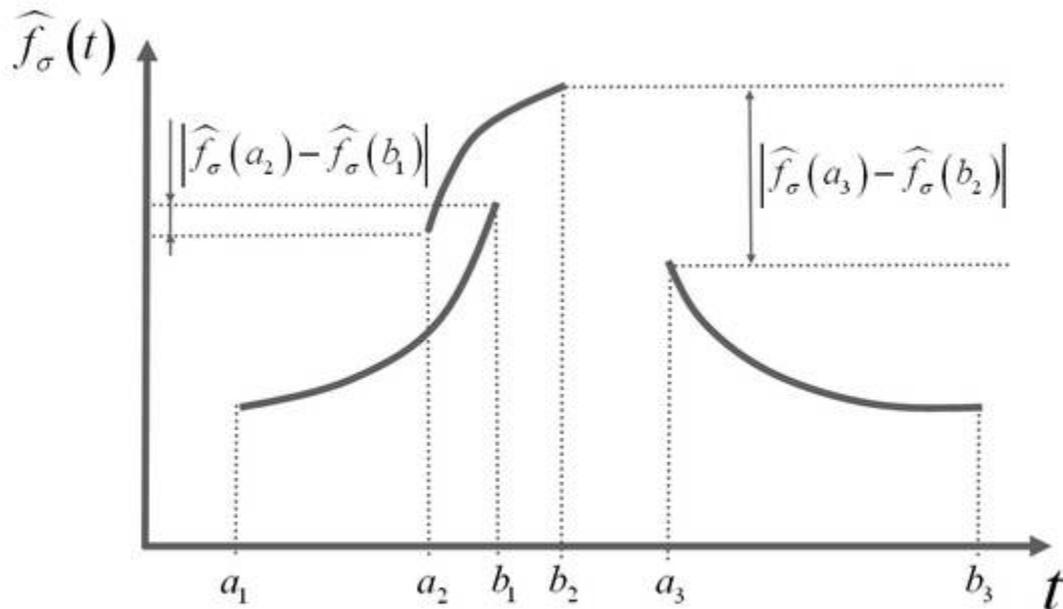


Abbildung 30 Veranschaulichung der Verknüpfung mit Folgeepisoden

III.1.2.3.4 Eliminierung von Redundanzen

Es ist anzumerken, dass für jede Episode Y_{e-1} mehr als eine potentielle Folgeepisode Y_e gefunden werden kann. Dies liegt darin begründet, dass die Kombination aufeinander folgender Episoden über den kompletten eingeschränkten Suchraum erlaubt ist. Bei hohen Längen- und Höhentoleranzen können somit auf verschiedenen Faltungsskalen und für verschiedene Linearitätskoeffizienten Folgeepisoden werden, welche der Definition aus der Suchanfrage entsprechen. Bei hohen Überlappungs- und Abstandstoleranzen können selbst auf einer Faltungsskala und für einen Linearitätskoeffizienten mehr als eine potentielle Folgeepisode gefunden werden^a.

Bei mehr als einer identifizierten Folgeepisode wird diejenige ausgewählt, welche den höchsten Ähnlichkeitsindikator mit der entsprechenden Referenz-Episode aufweist (vgl. Kapitel III.1.2.3.5).

Die Gewichtung der Ähnlichkeitskriterien hat somit nicht erst Einfluss auf die Reihenfolge der

^a Da sich die Abstandstoleranz auf die längere von zwei aufeinanderfolgenden Episoden bezieht, können für die kürzere (nachfolgende) Referenzepisode mehr als eine Episode innerhalb der Abstandstoleranz liegen.

Ergebnistrends in der Ergebnisliste, sondern bereits auf die Identifikation potentieller Ergebnistrends.

Als Zwischenresultat erhält man den Ergebnistrend \bar{Y} . Alle i Ergebnistrends werden in der Menge \bar{P} zusammen gefasst.

Gleichung 68
$$\bar{Y} = (Y_1, \dots, Y_{\max(e)})$$

Gleichung 69
$$\bar{P} = \{ {}_i Y, \dots, {}_i Y \} \text{ mit } i = \{ 1, \dots, j \}$$

Eine weitere Redundanzreduktion erfolgt auf Basis von Vergleichen ganzer Ergebnistrends. Ein jeder Ergebnistrend erstreckt sich über ein bestimmtes Zeitintervall. Gegeben ein Ergebnistrend im Intervall I und ein Ergebnistrend im Intervall J und eine Überlappung beider Intervalle größer als eine Redundanztoleranz r_{tol} , wird derjenige Ergebnistrend eliminiert, welcher den kleineren Gesamtähnlichkeitsindikator aufweist. Gleichung 70 berücksichtigt den Fall, dass zwei Ergebnistrends vollständig überlappen, jedoch aufgrund ihrer großen Längenunterschiede nicht als redundant gewertet werden können^a. Diese Prozedur wird so lange fortgeführt, bis alle Redundanzen eliminiert sind. Die verbleibenden Ergebnistrends werden in der geordneten Menge E zusammen mit den dazu gehörenden Ähnlichkeitsindikatoren g als endgültiges Ergebnis ausgegeben (Gleichung 71).

Gleichung 70
$$P \in \bar{P} \left| \frac{2 * |I \cap J|}{|I| + |J|} > r_{tol} \text{ mit } 0 \leq r_{tol} \leq 1 \right.$$

Gleichung 71
$$E = \{ \{ {}_i Y, {}_i g \}, \dots, \{ {}_i Y, {}_i g \} \} \mid {}_i Y \in P$$

^a Ein solcher Fall kann nur bei extrem großen Längen- und/oder Überlappungstoleranzen bei der Trendbeschreibung auftreten.

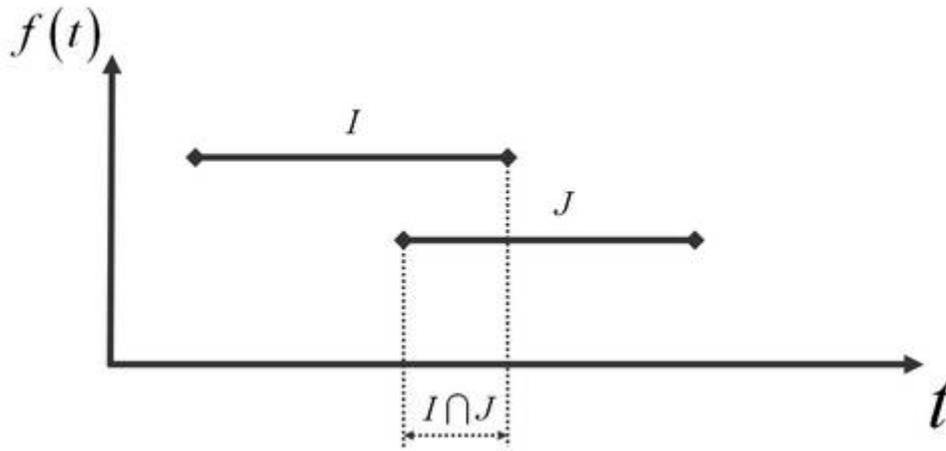


Abbildung 31 Veranschaulichung der Gleichung 70

III.1.2.3.5 Ähnlichkeitsschätzung

Die Ähnlichkeit der Suchergebnisse mit der Trendbeschreibung aus der Suchanfrage wird zunächst für alle Episoden des Referenzrends einzeln geschätzt (Gleichung 73 und Gleichung 74). Der Gesamtähnlichkeitsindikator g eines Ergebnistrends wird dann auf Basis der Ähnlichkeitsindikatoren g_e aller einzelnen Episoden berechnet (Gleichung 72).

Gleichung 72
$$g = \frac{\sum_{e=1}^{\max(e)} g_e}{\max(e)}$$

Gleichung 73
$$g_e = \frac{\left(G_{w_e} * g_{w_e} + G_{h_e} * g_{h_e} + G_{\theta_{w_e}} * g_{\theta_{w_e}} \right) + G_{\phi_{w_e}} * g_{\phi_{w_e}} + G_{\theta_{h_e}} * g_{\theta_{h_e}} + G_{v_e} * g_{v_e}}{G_{w_e} + G_{h_e} + G_{\theta_{w_e}} + G_{\phi_{w_e}} + G_{\theta_{h_e}} + G_{v_e}}$$

Für $e = 1$ gilt folgende Gleichung.

Gleichung 74
$$g_1 = \frac{(G_{w_1} * g_{w_1} + G_{h_1} * g_{h_1} + G_{v_1} * g_{v_1})}{G_{w_1} + G_{h_1} + G_{v_1}}$$

Zur Berechnung der Ähnlichkeitsindikatoren g wird dabei eine Gaußverteilung genutzt, wobei im Allgemeinen der Sensitivitätsparameter S die Standardabweichung der Gaußfunktion bestimmt (Gleichung 75). Den Effekt unterschiedlicher Sensitivitäten verdeutlicht Abbildung 32. Eine

Veränderung von S führt zu einer veränderten Ähnlichkeit der Ergebnisse. Ein kleiner Sensitivitätsparameter reduziert die Ähnlichkeit aller nicht hundertprozentig ähnlichen Ergebnisse. Ein großer Sensitivitätsparameter homogenisiert die Ergebnisähnlichkeit. Eine unterschiedliche Ausprägung der Sensitivitätsparameter einzelner Ähnlichkeitsindikatoren kommt damit einer impliziten Gewichtung gleich, wie beispielhaft in Tabelle 12 aufgezeigt wird.

Gleichung 75
$$g(x, \mu) = e^{-\frac{1(x-\mu)^2}{2(S*\mu)^2}}$$

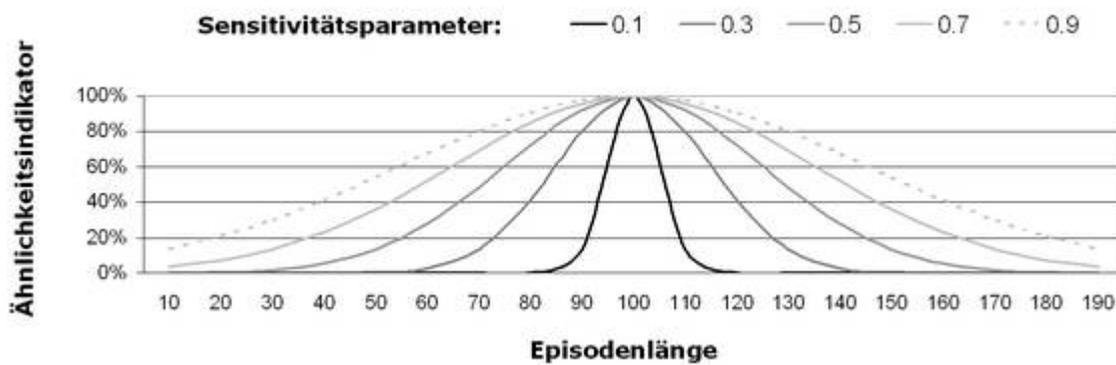


Abbildung 32 Beispiel zur Veranschaulichung des Sensitivitätsparameters S auf die Ähnlichkeitsschätzung am Beispiel einer Referenz-Episode mit der Länge von 100 Zeiteinheiten.

Tabelle 12 Effekte des Sensitivitätsparameters

	Episode A	Episode B	Episode C	Sensitivität
Länge	0,5	0,5	1	0,9
Höhe	0,5	1	0,5	0,6
Abstand	1	0,5	0,5	0,3
Ähnlichkeitsindikator	0,26	0,18	0,11	

Um die Anpassung des prozentualen Sensitivitätsfaktors S an den absoluten Wert des Ähnlichkeitskriteriums der jeweiligen Referenz-Episode zu gewährleisten, werden die folgenden Gleichungen genutzt.

$$\text{Gleichung 76} \quad g_{w_e} = e^{-\frac{1}{2} \frac{(\widehat{w} - w_e)^2}{(S_w * w_e)^2}} \quad \text{mit} \quad \widehat{w} = b_e - a_e$$

Die absolute Sensitivität (Standardabweichung der Gaußverteilung) des Ähnlichkeitsindikators g_{w_e} (Episodenlänge) ergibt sich aus dem Produkt der relativen Sensitivität S_w mit der Länge w_e der entsprechenden Referenz-Episode e .

$$\text{Gleichung 77} \quad g_{h_e} = e^{-\frac{1}{2} \frac{(\bar{h} - h_e)^2}{(S_h * h_e)^2}} \quad \text{mit} \quad \bar{h} = \widehat{f}_\sigma(b_e) - \widehat{f}_\sigma(a_e)$$

Die absolute Sensitivität (Standardabweichung der Gaußverteilung) des Ähnlichkeitsindikators g_{h_e} (Episodenhöhe) ergibt sich aus dem Produkt der relativen Sensitivität S_h mit der Höhe h_e der entsprechenden Referenz-Episode e .

$$\text{Gleichung 78} \quad g_{\theta w_e} = e^{-\frac{1}{2} \frac{(a_e - b_{e-1})^2}{(S_{\theta w} * \max(\widehat{\theta w}_e))^2}} \quad \text{mit} \quad a_e - b_{e-1} \leq 0$$

$$\text{und} \quad \max(\widehat{\theta w}_e) = \begin{cases} w_e & \text{falls } w_e \leq w_{e-1} \\ w_{e-1} & \text{falls } w_e > w_{e-1} \end{cases}$$

Die absolute Sensitivität (Standardabweichung der Gaußverteilung) des Ähnlichkeitsindikators $g_{\theta w_e}$ (Episodenüberlappung) ergibt sich aus dem Produkt der relativen Sensitivität $S_{\theta w}$ mit der maximal möglichen Episodenüberlappung $\max(\widehat{\theta w}_e)$. Diese entspricht jeweils der Länge der kürzeren zweier aufeinanderfolgender Episoden.

$$\text{Gleichung 79} \quad g_{\phi w_e} = e^{-\frac{1}{2} \frac{(a_e - b_{e-1})^2}{(S_{\phi w} * \max(\widehat{\phi w}_e))^2}} \quad \text{mit} \quad a_e - b_{e-1} \geq 0$$

$$\text{und} \quad \max(\widehat{\phi w}_e) = \begin{cases} w_e & \text{falls } w_e \geq w_{e-1} \\ w_{e-1} & \text{falls } w_e < w_{e-1} \end{cases}$$

Die absolute Sensitivität (Standardabweichung der Gaußverteilung) des Ähnlichkeitsindikators $g_{\phi_w_e}$ (Episodenabstand) ergibt sich aus dem Produkt der relativen Sensitivität S_{ϕ_w} mit dem maximal möglichen Episodenabstand $\max(\widehat{\phi_w}_e)$. Dieser entspricht jeweils der Längensumme zweier aufeinanderfolgender Episoden.

Gleichung 80
$$g_{\theta h_e} = e^{-\frac{1}{2} \frac{(\widehat{f}_\sigma(a_e) - \widehat{f}_\sigma(b_{e-1}))^2}{(S_{\theta h} * \max(\widehat{\theta h}_e))^2}}$$
 mit
$$\max(\widehat{\theta h}_e) = \begin{cases} h_e & \text{falls } h_e \geq h_{e-1} \\ h_{e-1} & \text{falls } h_e < h_{e-1} \end{cases}$$

Die absolute Sensitivität (Standardabweichung der Gaußverteilung) des Ähnlichkeitsindikators $g_{\theta h_e}$ (vertikale Episodenverschiebung) ergibt sich aus dem Produkt der relativen Sensitivität $S_{\theta h}$ mit der maximal möglichen vertikalen Episodenverschiebung $\max(\widehat{\theta h}_e)$. Dieser entspricht jeweils den addierten Beträgen der Höhendifferenzen zweier aufeinanderfolgender Episoden.

Gleichung 81
$$g_{v_e} = e^{-\frac{1}{2} \frac{(\bar{v} - v_e)^2}{(S_v * v_e)^2}}$$
 mit
$$\bar{v} = \frac{\frac{(\widehat{f}_\sigma(b_e) - \widehat{f}_\sigma(a_e)) * (b_e - a_e)}{2} - \left(\int_{a_e}^{b_e} \widehat{f}_\sigma(t) dt - \widehat{f}_\sigma(a_e) * (b_e - a_e) \right)}{\frac{(\widehat{f}_\sigma(b_e) - \widehat{f}_\sigma(a_e)) * (b_e - a_e)}{2}}$$

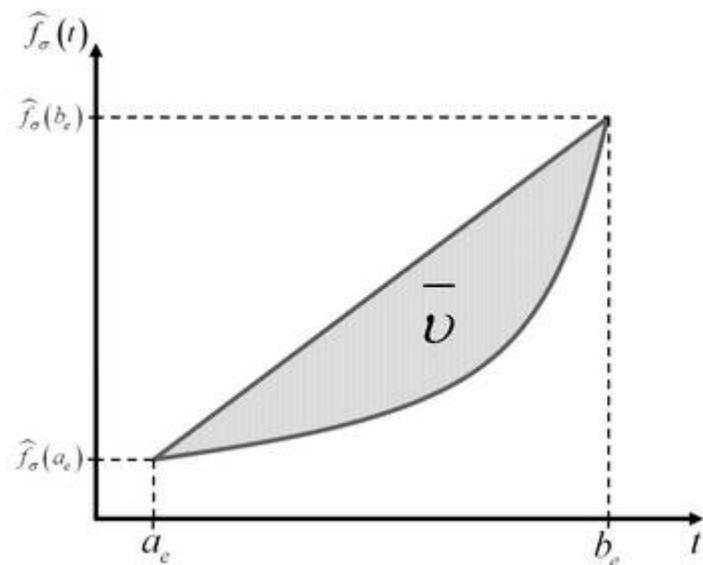


Abbildung 33 Veranschaulichung von Komponenten aus Gleichung 81

Die absolute Sensitivität (Standardabweichung der Gaußverteilung) des Ähnlichkeitsindikators g_{v_e} (Episodenkrümmung) ergibt sich aus dem Produkt der relativen Sensitivität S_{v_e} mit der Krümmungsstärke v_e der entsprechenden Referenz-Episode e . Die Krümmungsstärke berechnet sich aus der Differenz zwischen dem Integral einer linearen Verknüpfung von Episodenstart und Ende und dem Integral der tatsächlichen gefalteten Zeitreihe $\hat{f}_\sigma(t)$ (vgl. Abbildung 32).

III.1.2.4 Konfigurationsheuristiken

Je weniger Suchparameter vom Nutzer zur Trendbeschreibung eingestellt werden müssen, desto einfacher wird das Erlernen und Handhaben der Suchmaschine ausfallen. Je mehr Suchparameter jedoch bereits im Rahmen der Vorverarbeitung bekannt sein müssen, desto unflexibler wird die Suchmaschine auf Unterschiede der Prozessdynamiken und der Interessen der Experten eingestellt werden können. Um das Dilemma zwischen Einfachheit und Flexibilität zu lösen, wurden möglichst viele fakultative Konfigurationsmöglichkeiten implementiert. Nichts desto trotz bleiben zwei wesentliche Parameter, welche es gilt im Rahmen der Vorverarbeitung zu definieren den Skalenraum ζ und den Linearitätsindex Λ .

III.1.2.4.1 Parameter der Vorverarbeitung

Für die Parametrisierung des Skalenraums wird häufig eine dyadische Abstufung vorgeschlagen (vgl. Kapitel II.1.1). Folgt man dem Gedankengang, dass auf kleinen Skalen vor allem kurze Episoden zu finden sind, und die Länge der identifizierten Episoden mit der Höhe der Skala ansteigt, dann würde eine dyadische Skalenauflösung (Gleichung 82) die kurzen Episoden überproportional häufig repräsentieren. Werden die signifikanten Dynamiken jedoch auf höheren Skalen gesucht, dann zeigt eine lineare Skalierung (Gleichung 83) eine deutlich feinere Auflösung (vgl. Abbildung 34). Noch empfehlenswerter ist eine Verteilung der Skalierungsparameter, die sich an einer angenommenen (oder empirisch erhobenen) Gaußverteilung der Episodenlängen orientiert, welche zur Trendbeschreibung verwendet werden.

Gleichung 82 $\sigma = 2^n$ mit $n \in \mathbb{N}^+$

Gleichung 83

$$\sigma = 100 * n - 50 \text{ mit } n \in \mathbb{N}^+$$

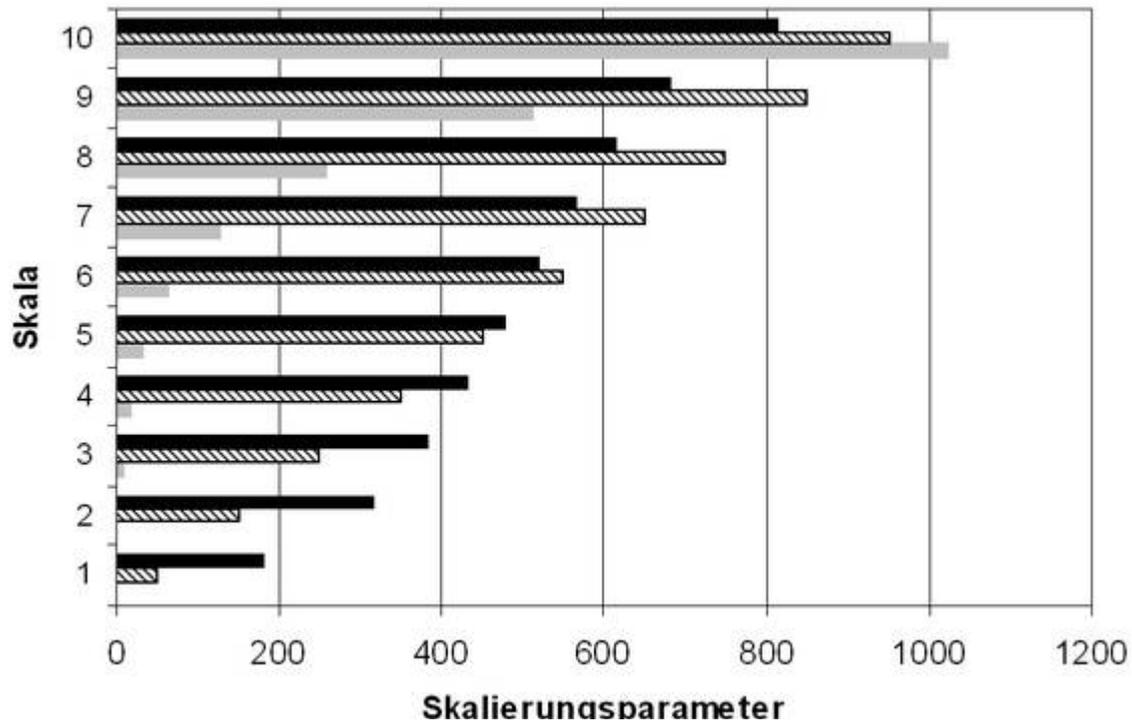


Abbildung 34 Visuelle Größenverteilung der dyadisch (hellgrau), linear (gestrichelt) und Gauß (schwarz) verteilten Skalierungsparameter

Eine genaue Einstellung des Skalierungsparameters lässt sich jedoch auch dann nicht generalisieren. Auf Basis von analytischen Überlegungen und explorativen Tests können aber folgende Heuristiken herleiten:

- Je feiner die Skalenauflösung, desto genauer sind die Suchergebnisse.
- Je breiter der Skalenraum, desto vollständiger sind die Suchergebnisse.
- Je größer die Skalenauflösung, desto schneller können Vorverarbeitung und Suche berechnet werden.

In explorativen Studien konnten keine Effekte einer systematischen Kombination von Skalierungsparameter und Linearitätskoeffizienten auf die Güte der Linearapproximationen gefunden werden.

Allgemein lässt sich jedoch sagen:

- Je kleiner der Linearitätskoeffizient, desto anfälliger ist der Algorithmus gegen hochfrequentes Rauschen.

- Je größer der Linearitätskoeffizient, desto größer ist die Verzögerung, mit der Start- und Endzeitpunkte von linearen Episoden erkannt werden, und desto stärker wird der Fehler bei den Höhenunterschieden.
- Je weniger Linearitätskoeffizienten für die Vorverarbeitung genutzt werden, desto schneller können Vorverarbeitung und Suche berechnet werden.

III.1.2.4.2 Parameter der Suchprozedur

Neben der eigentlichen Trendbeschreibung ist es dem Nutzer möglich, eine Reihe von Toleranzen zu definieren, für die allgemein gilt:

- Je kleiner die Toleranzen, desto weniger unpassende Ergebnisse werden gefunden.
- Je größer die Toleranzen, desto eher werden zufrieden stellende Ergebnisse trotz Überlagerungen und Artefakten gefunden.

Der Längen- und Abstandstoleranz kommt durch die glättenden Eigenschaften der Vorverarbeitung eine besondere Bedeutung zu.

- Je stärker der Übergang zwischen zwei Episoden gebrochen verläuft (vgl. Tabelle 8 aus Kapitel III.1.1.4.2), und je stärker der Rauschanteil einer Zeitreihe, desto größer müssen die Längen- und/oder Abstandstoleranzen gewählt werden.

Episoden unterschiedlicher Länge werden bei starkem Rauschen und starker Umgebungsdynamik nur auf unterschiedlichen Skalen identifizierbar. Damit wird eine Überlappungstoleranz erforderlich.

- Je größer die Skalenraumauflösung ist, desto größer sollte die Überlappungstoleranz sein.
- Je größer die Heterogenität der Episodenlängen, desto größer sollte die Überlappungstoleranz sein.

III.1.2.5 Explorative Evaluation

Mit einer Bewertung und Optimierung der Suchalgorithmen als technisches Subsystem lässt sich bereits in frühen Projektphasen eine prospektive Schätzung der Gebrauchstauglichkeit durchführen. Der Suchalgorithmus sollte möglichst vollständige, genaue, robuste und zuverlässige Ergebnisse liefern, und zwar unabhängig von den Fertigkeiten im Kurvenlesen und dem Prozesswissen der Prozessführungsexperten. In den folgenden Abschnitten werden zunächst ausgewählte Testszenarien sowie Konfigurationen des Algorithmus erläutert. Danach werden die Ergebnisse präsentiert und im Hinblick auf die Verwendbarkeit des Algorithmus für die weitere Systementwicklung diskutiert.

III.1.2.5.1 Testszenarien

Die Testzeitreihen wurden auf Basis geometrischer Primitiva und mit Hilfe von MATLAB R2006b erstellt. Die Primitiva entsprechen dabei den Episoden des vom Suchalgorithmus verwendeten Trendlet-Alphabets. Durch eine systematische Variation einzelner Episodenparameter (vor allem Episodenlängen und -höhen) lassen sich Testsets erstellen, deren einzelnen Testszenarien unterschiedlich stark von einem frei wählbaren Referenzszenario abweichen. Anhand der verwendeten Episodenparameter lässt sich eine normative Ähnlichkeit aller Testszenarien mit dem entsprechenden Referenzszenario schätzen.

Auf Basis dieser artifiziell erstellten Testsets lässt sich das technische Subsystem unter Ausschluss der vom menschlichen Subsystem verursachten Varianz überprüfen. Erst wenn der Algorithmus bei diesen Tests eine hinreichende Genauigkeit und Robustheit aufweist, kann sinnvoller Weise zur nächsten Stufe der Systementwicklung übergegangen werden.

Zur Überprüfung der endgültigen und im Kapitel III.1.2 vorgestellten Version des Suchalgorithmus wurden jeweils ein Testset für gekrümmte und lineare Episoden gewählt. Anhand von Quell-Code 1 sei die Erstellung der insgesamt 121 Testszenarien für das Testset der gekrümmten Episoden erläutert (vgl. auch Abbildung 35). In den Zeilen 16 bis 23 kennzeichnen die Argumente **(L,E,F,G,D,E,F,L)** die Trendlets der aufeinander folgenden Episoden. Der Buchstabe **L**

umfasst alle drei linearen Trendlets $\{A, B, C\}$. In der nachstehenden Klammer werden Länge und Höhe der jeweiligen Episoden spezifiziert. Zu Beginn und zum Ende eines jeden Testszenarios wurden stationäre Episoden eingesetzt. Diese relativ langen stationären Start- und Endepisoden (Zeilen 16 & 23) dienen der Vermeidung von Randeffekten, welche durch den kontinuierlichen Charakter der Transformation aus der Vorverarbeitung entstehen (vgl. Kapitel II.1.1). Über alle Testszenarios wurden die Längen und Höhen der Zwischenepisoden (Zeilen 17 bis 22) innerhalb festgelegter Intervalle und in Form von festgelegten Schrittweiten variiert (Zeilen 10 & 11). Das gesamte Testset wurde mit einem weißen Rauschen überlagert, welches mit der gleich verteilten Zufallsfunktion $f(x)$ erstellt wurde. Die Amplitude des Rauschens wurde mit dem Faktor X an der Amplitudendifferenz Δh des gesamten Testszenarios relativiert. Im Testset der gekrümmten Episoden betrug der Faktor X den Wert 0.1 (Zeile 15). Der so relativierte Rauschanteil wurde mit der Basisfunktion $f(t)$ addiert (Gleichung 84).

Quell-Code 1: Testset gekrümmter Episoden

```

10  range_1 = (0.5 : 0.1 : 1.5);
11  range_2 = (0.5 : 0.1 : 1.5);
12  for index_1 = range_1
13      for index_2 = range_2
14          T = wxalphabet( ...
15              wxnoise(0.1, ...
16                  L (2,0), ...
17                  E (index_1,0.5), ...
18                  F (index_1,index_1), ...
19                  G (index_2,(2-(index_1))), ...
20                  D (index_2,(2-(index_1))), ...
21                  E ((2*index_1),(index_1)), ...
22                  F ((2*index_1),0.5), ...
23                  L (2,0) ...
24              );
...
50      end
51  end

```

Gleichung 84
$$\hat{f}(t) = f(x) * X * \Delta h + f(t)$$

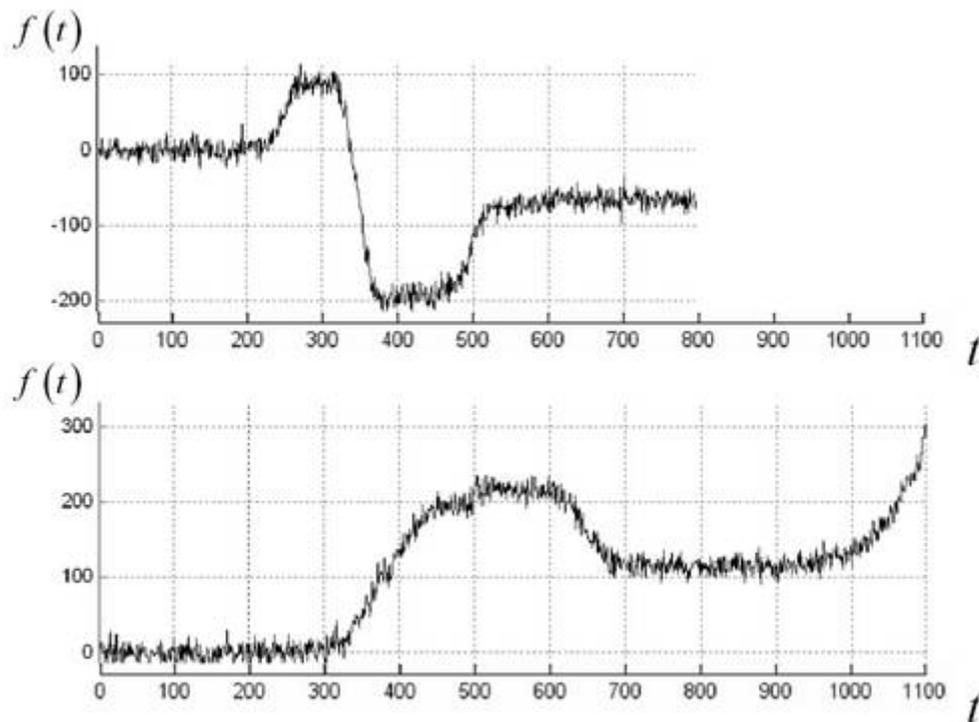


Abbildung 35 Bandbreite an Testszenarien für das Testset gekrümmter Episoden

Für ein Testset aus linearen Episoden wurden 36 Testszenarien anhand von Quell-Code 2 erstellt. Die Lesart erfolgt analog zum Testset der gekrümmten Episoden. Einen visuellen Eindruck der Bandbreite an Testszenarien vermittelt Abbildung 35. Darin ist sowohl das Testszenario mit den kürzesten und niedrigsten Episoden als auch das Testszenario mit den längsten und höchsten Episoden abgebildet.

Quell-Code 2: Testset linearer Episoden

```

10 range_1 = (1.0 : 0.2 : 2.0);
11 range_2 = (1.0 : 0.2 : 2.0);
12 for index_1 = range_1
13     for index_2 = range_2
14         T = wxalphabet( ...
15             wxnoise(0.1, ...
16                 L (2,0), ...
17                 L (index_1,index_2), ...
18                 L (index_2,0), ...
19                 L (index_1,index_2), ...
20                 L (2,(-2*index_2)), ...
21                 L (2,0) ...
22             );
...
50     end
51 end

```

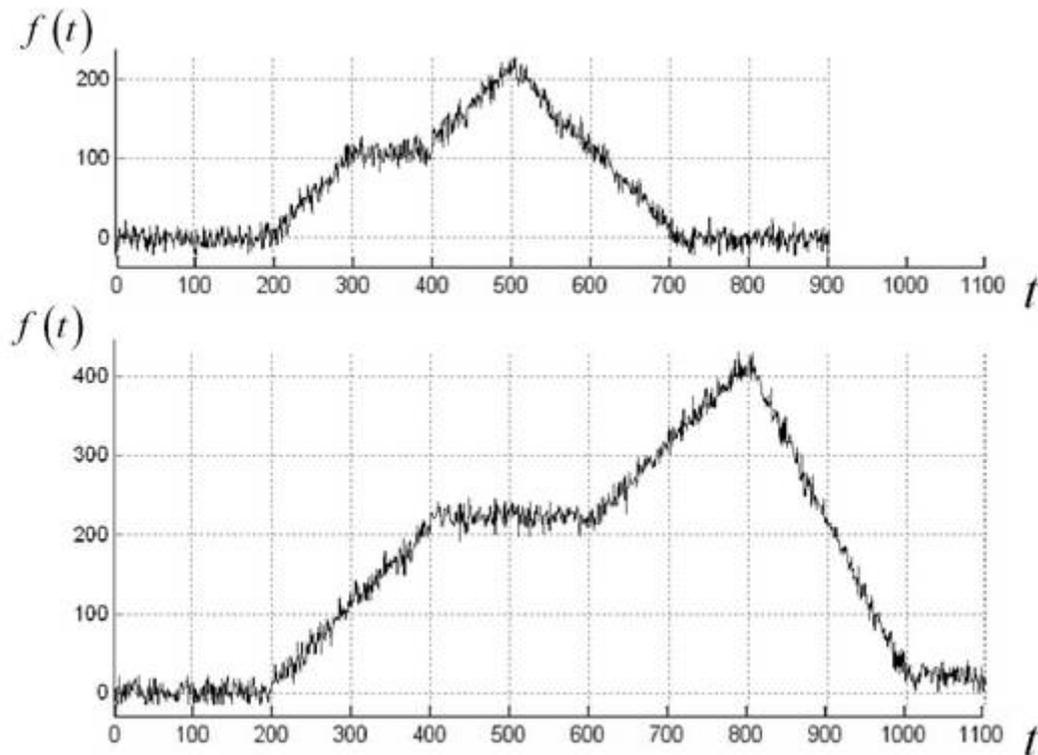


Abbildung 36 Bandbreite an Testszenarien für das Testset linearer Episoden

III.1.2.5.2 Testkonfiguration

An die beiden ausgewählten Testsets wurden zwei verschiedene Suchanfragen gerichtet. Da zu dieser Phase der Systementwicklung keine expliziten Recheneffizienzen empirisch erhoben werden sollten, wurde für beide Testsets eine vergleichbare Rechendauer angestrebt. Dies wurde durch eine übereinstimmende Anzahl von acht Suchindexen im Suchraum Γ_c ermöglicht (Tabelle 13). Die Komplexität der Suchanfragen wurde auf die Spezifikation und Gewichtung der Episodenlängen und -höhen beschränkt. Die Toleranzen wurden so großzügig gewählt, dass theoretisch alle Testszenarien vollständig erfasst werden konnten. Auf Basis von vorab durchgeführten Einzelfallversuchen wurde ein Linearitätsparameter von $\lambda_\Omega = 0.2$ gewählt.

Tabelle 13 Konfiguration von Vorverarbeitung und Suchanfrage

Testset	gekrümmte Episoden	lineare Episoden
Vorverarbeitung	$\zeta = \{5, 10, 15, \dots, 45\}$	$\zeta = \{5, 10, \dots, 20\}$ $\Lambda = \{5, 10\}$
Suchanfrage	$\Omega_1 = (G, 150, 120, -100, 80)$ $\Omega_2 = (D, 150, 120, -100, 80)$ $\lambda_{\Omega} = 0.2$ $\{\theta w_2, \phi w_2, \theta h_2\} = 20$	$\Omega_1 = (B, 200, 120, 200, 120)$ $\Omega_2 = (A, 200, 120, 0, 50)$ $\Omega_3 = (B, 200, 120, 200, 120)$ $\lambda_{\Omega} = 0.2$ $\{\theta w_2, \phi w_2, \theta h_2, \theta w_3, \phi w_3, \theta h_3\} = 20$
Parameter	$r_{tol} = 0$ $\{G_w, G_h\} = 1$ $\{G_{\theta w}, G_{\phi w}, G_{\theta h}, G_v\} = 0$ $\left\{ \begin{matrix} S_w, S_h, S_{\theta w} \\ S_{\phi w}, S_{\theta h}, S_v \end{matrix} \right\} = 0, 3$	

III.1.2.5.3 Ergebnisse

Die Suchergebnisse werden anhand der in Tabelle 14 aufgeführten Kriterien bewertet. Diese Kriterien wurden unter anderem von Venkatasubramanian et al. (2003) vorgeschlagen. Die Suche nach gekrümmten sowie linearen Episoden war in beiden Fällen annähernd vollständig und gleichzeitig genau im Sinne des Signal-Entdeckungs-Paradigmas (Abdi, 2007). Eine statistische Absicherung dieser Ergebnisse ist nicht möglich, da die korrekter Weise abgelehnten Ergebnistrends nicht quantifiziert werden können (Tabelle 15). Die Anzahl liegt jedoch weit über der Anzahl an korrekter Weise identifizierter Ergebnistrends, sodass ein entsprechender Vierfelder-Chi-Quadrat-Test (Bortz & Lienert, 2003, S. 86 u. f.) hochsignifikant ausfallen würde. Andererseits stehen die Häufigkeiten aus Tabelle 15 in direkter Abhängigkeit zu der Höhe der

Toleranzen aus der Suchanfrage und dem Rauschanteil aus der Szenarienerstellung. Letzteres sorgt zum Beispiel für inkorrekte Treffer, wie er in Abbildung 37 zu sehen ist. Dieser wird auf Basis der systematischen Szenarienerstellung nicht als Normtreffer gewertet, obwohl er einem menschlichen Betrachter als Ergebnis der Suchanfrage innerhalb der dort spezifizierten Toleranzen erscheint.

Tabelle 14: Evaluationskriterien

Vollständigkeit	Identifikation aller ähnlichen Trends
Genauigkeit	Ausschluss aller unähnlichen Trends
Robustheit	Unabhängig der Suchqualität gegenüber Störeinflüssen (Messrauschen, Arbeitspunktverschiebung oder irrelevante dynamische Muster)
Zuverlässigkeit	Wiederholbarkeit der Suchanfragen bzw. Re-Test-Reliabilität

Tabelle 15 Aufstellung der Anzahl identifizierter Testszenarien nach dem Signal-Entdeckungs-Paradigma.

Schwarz: korrekte Treffer bzw. Abweisungen.

Grau: inkorrekte Treffer bzw. Abweisungen.

Signal-Entdeckungs-Matrix		Testset			
		gekrümmte Episoden		lineare Episoden	
		Normtreffer		Normtreffer	
		ja	nein	ja	nein
empirischer Treffer	ja	118	8	30	0
	nein	3	?	6	?

Weitaus nachhaltiger lässt sich die Genauigkeit der Suchergebnisse (gekrümmte Episoden) anhand der Ähnlichkeitsschätzung bestimmen. Der substantielle Zusammenhang von $r = 0.81^{**}$ (Bortz, 1999, S. 197 u. f.) zwischen dem Ähnlichkeitsindikator G aller Ergebnistrends und der normativ

induzierten Abweichung der entsprechenden Testszenarien weist auf eine genaue und trotz des hohen Rauschfaktors robuste Ähnlichkeitsschätzung hin. Der Zusammenhang mit den vom Suchalgorithmus extrahierten Abweichungen fällt mit $r = 0.93^{**}$ noch etwas deutlicher aus. Abbildung 38 vermittelt die Ursache für diesen Unterschied. Der hohe Rauschanteil bewirkt vor allem bei den Episodenlängen eine Veränderung der normativ induzierten Abweichungen. Zusammenfassend lässt sich zur Robustheit des Algorithmus gegenüber weißem Rauschen sagen, dass das Auffinden tatsächlich vorhandener Trends bei entsprechend hohen Toleranzen recht unbeeinflusst erfolgt, die Genauigkeit der Ähnlichkeitsschätzung jedoch mit zunehmendem Rauschfaktor abnimmt.

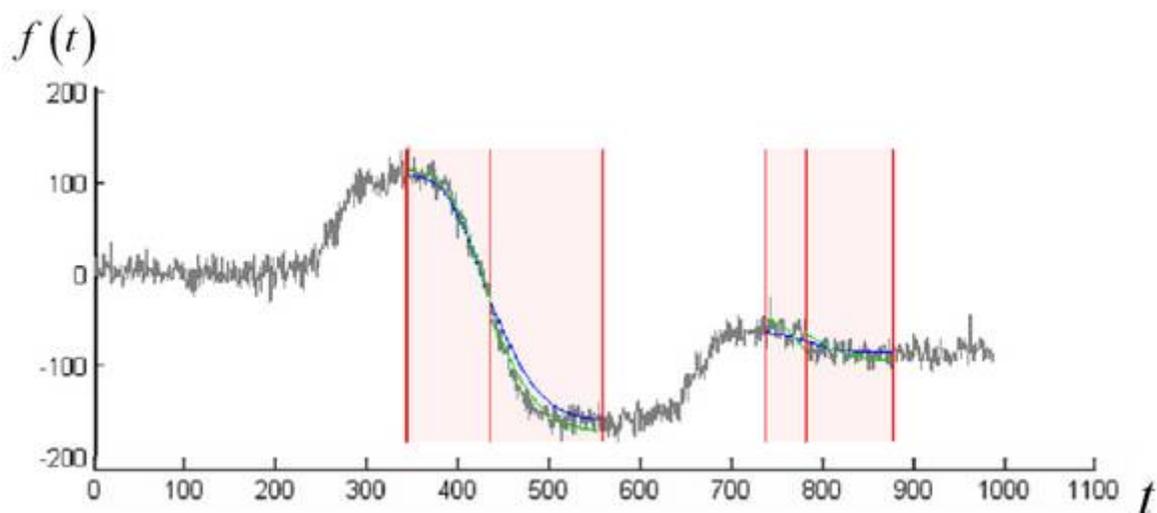


Abbildung 37 Beispiel für ein normativ korrekt (Mitte) und ein normativ inkorrekt (rechts) identifiziertes Ergebnis^a

Beim Testset der linearen Episoden verhält es sich vergleichsweise ähnlich (Abbildung 39). Allerdings wirkt hier der Rauschfaktor auf Episodenlängen und –höhen gleichermaßen. Insgesamt wirkt er jedoch weniger, was sich in den etwas näher beieinander liegenden Korrelationskoeffizien-

^a Zum Verständnis der Attribute *korrekt* und *inkorrekt* sei die Tabelle 15 konsultiert.

ten wider spiegelt (Ähnlichkeitsindikator G & normative induzierte Abweichung: $r = 0.88^{**}$; Ähnlichkeitsindikator G & empirisch generierte Abweichung: $r = 0.96^{**}$).

Die Zuverlässigkeit im Sinne einer Re-Test-Reliabilität wurde überprüft, indem ein und dieselbe Suchanfrage in voneinander unabhängigen Versuchen an ein und dieselbe Datenbank gestellt wurde. Die Ergebnisse und Ähnlichkeitsindikatoren waren in zehn verschiedenen Versuchen zu hundert Prozent identisch. Dies ist jedoch nicht weiter verwunderlich, da der Algorithmus in seinem Konzept keinerlei Unbestimmtheiten (z.B. in Form von Fuzzy-Logik) enthält. Alle Varianz in der Ergebnisgüte entsteht aus den Unterschieden der Datenbasis und/oder der Suchanfrage.

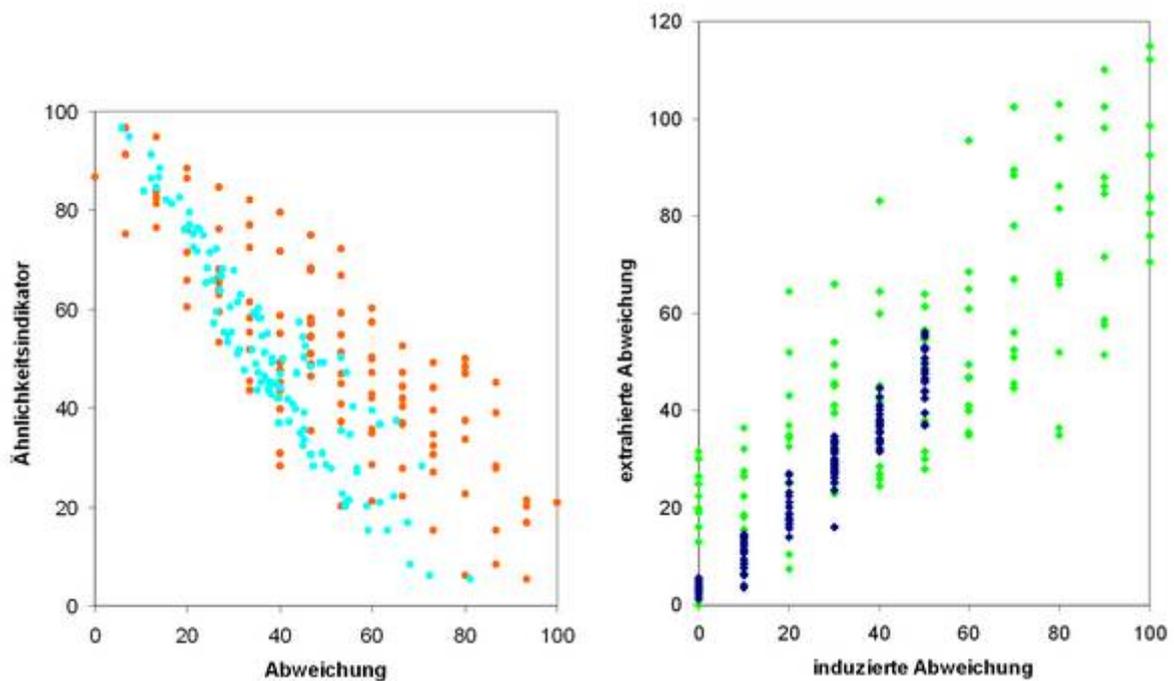


Abbildung 38 Testergebnisse gekrümmter Episoden.
Links: Ähnlichkeitsindikator in Abhängigkeit von induzierter (orange) und extrahierter (hellblau) Abweichung.
Rechts: extrahierte Abweichung in Abhängigkeit von induzierter Abweichung (grün: Episodenlänge; dunkelblau: Episodenhöhe)

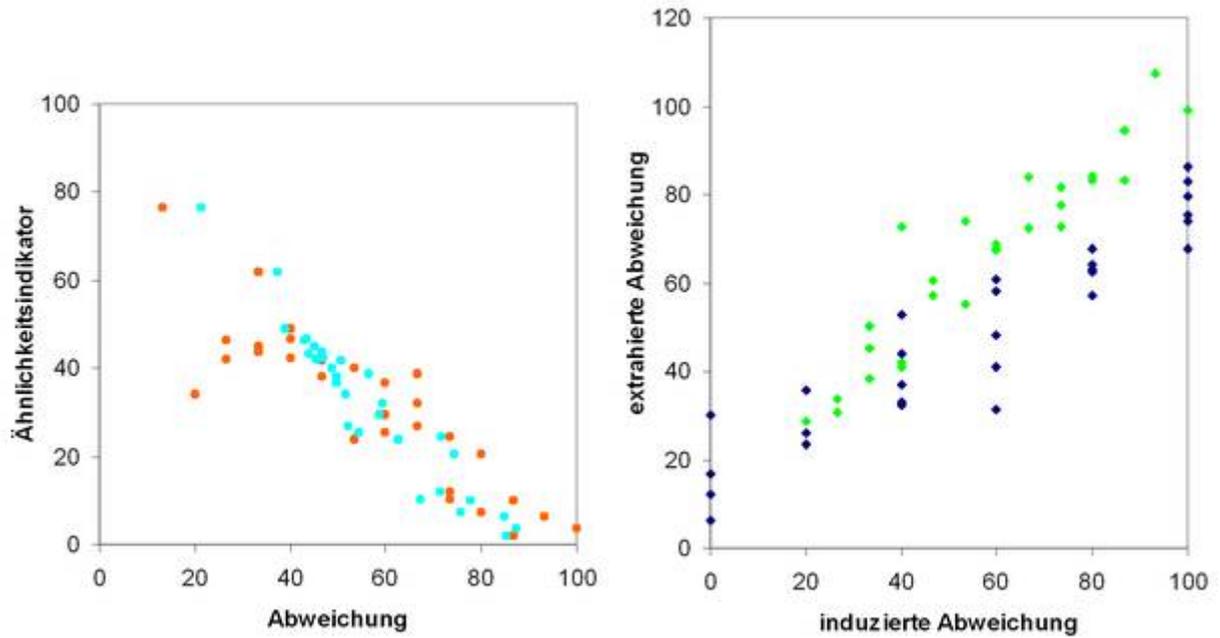


Abbildung 39 Testergebnisse linearer Episoden.
 Links: Ähnlichkeitsindikator in Abhängigkeit von induzierter (orange) und extrahierter (hellblau) Abweichung.
 Rechts: extrahierte Abweichung in Abhängigkeit von induzierter Abweichung (grün: Episodenlänge; dunkelblau: Episodenhöhe)

III.2 Nutzerdialog

Die Mensch-Technik-Schnittstelle besteht in modernen Leitwarten zunehmend aus Computermonitoren und Projektionswänden. Vor diesem Hintergrund gibt die Richtlinie VDI/VDE 3699-4 (1997) Empfehlungen für die Gestaltung und Verwendung der Bildschirmdarstellung zur Prozessführung. Die Richtlinie behandelt in mehreren Blättern weit verbreitete Darstellungen wie Fließbilder, Kurven, Meldungen sowie Darstellungen von Bedienverfahren und Bediengeräten. Blatt 4 der Richtlinie gibt Regeln und Empfehlungen zur Darstellung und Bedienung von Kurven auf Bildschirmen. Die Anwendung der Richtlinie soll zu Darstellungen führen, die kompatibel mit den übrigen Darstellungsformen und konform mit geltenden Normen sind.

III.2.1 Entwurfsprozess

Wie bereits in Tabelle 1 aus Kapitel I deutlich wird, waren zwei neuartige Dialoge zwischen Nutzer und Suchmaschine zu entwickeln: Trendbeschreibung und Ergebnispräsentation. Als Kreativtechnik für Entwürfe komplexer Systeme, bietet sich die morphologische Analyse an (Pahl et al., 2005). Dazu werden alle relevanten Merkmale des Zielsystems in und deren mögliche Ausprägungen in sogenannte Morphologische Kästen (n-dimensionale Matrizen) aufgenommen. Liegen beispielsweise drei Merkmale vor, stellen diese die Achsen eines dreidimensionalen Kastens dar. Auf jeder Achse sind nun die jeweiligen Ausprägungen des Merkmals abgetragen. Eine Kombination von drei Ausprägungen liefert einen konkreten Punkt innerhalb des Kastens.

Um die Morphologischen Kästen als Entscheidungsmatrizen für die oben genannten Dialoge zu erstellen, wurde zu Beginn des Entwurfsprozesses nach folgender Prozedur vorgegangen:

1. Die bestimmenden Merkmale werden aufgelistet, wobei deren wechselseitige Unabhängigkeit zu gewährleisten ist.
2. Dokumentation aller möglichen Ausprägungen des jeweiligen Merkmals in der entsprechenden Zeile.

3. Wahl einer Ausprägung je Merkmal in Abhängigkeit von Faktoren wie Umsetzbarkeit, Implementierungskosten, Mehrwert oder Kombinationsfähigkeit.

Tabelle 16 Morphologischer Kasten für Dialogsystem „Ergebnispräsentation“.
Kursiv: Gewählte Systemlösung

Merkmal	Ausprägungen			
	A	B	C	D
Trendbeschreibung	<i>Segmentieren & Kategorisieren anhand von Trendlets</i>	Mensch-Technik Kooperation zur Kurven-approximation	Freihandzeichnen	Automatisierte Kategorisierung inklusive Vetorecht für Nutzer
Bezugsobjekte für Toleranzen	Anfangs- und Endzeitpunkte einzelner Referenz-Episoden	Anfangs- und Endzeitpunkte des gesamten Referenzrends	<i>Längen und Höhen einzelner Referenz-Episoden</i>	Längen und Höhen des gesamten Referenzrends
Darstellungsart für Toleranzen	<i>numerische Toleranzangabe für Wertebereich</i>	graphische Toleranzangabe für Wertebereich		
Gewichtung von Ähnlichkeitskriterien	Für einzelne Referenz-Episoden	<i>für gesamten Referenztrend</i>		

Tabelle 17

Morphologischer Kasten für Dialogsystem „Ergebnispräsentation“.
Kursiv: Gewählte Systemlösung

Merkmal	Ausprägungen			
	A	B	C	D
Häufigkeit vergleichbarer Trends	absolut	prozentual im Vergleich mit gesamten zeitlichen Suchintervall		
Darstellung aller Ergebnisse	<i>Sortiert nach integriertem Ähnlichkeits-indikator</i>	Sortiert nach einzelnen, auswählbaren Ähnlichkeitsindikatoren		
Darstellung von Einzelergebnissen	Referenzkurve + Ergebniskurve	Referenzkurve + Referenztrend + Ergebniskurve	<i>Referenzkurve + Ergebniskurve + Ergebnistrend</i>	Referenzkurve + Referenztrend + Referenzkurve + Ergebnistrend
Ergebniscluster	Referenzkurve + aggregierte Ergebniskurve	Referenztrend + aggregierter Ergebnistrend		
Graphische Visualisierung der Ähnlichkeit	farblich markiertes Integral zwischen Referenz und Ergebnis	<i>Möglichkeiten der Überlagerung von Referenz und Ergebnis</i>		
numerische Visualisierung der Ähnlichkeit	für Ergebniskurve	<i>für Ergebnistrend</i>	für Ergebnis-Episoden	

Um die Wahl aus Punkt drei zu ermöglichen, wurden zum einen die Vor- und Nachteile einzelner Design-Entscheidungen und deren Probleme für die softwaretechnische Implementierung zusammen getragen (siehe Anhang B.1.1). Zum anderen wurden schematische Prototypen mit Microsoft Powerpoint entworfen (siehe Anhang B.1.2). Derartige Prototypen (sog. Mockups) werden oft in

frühen Entwicklungsphasen eingesetzt, um Anforderungen an die Benutzeroberfläche in Zusammenarbeit mit Auftraggeber und Anwendern besser ermitteln zu können. Es handelt sich meist um ein reines Grundgerüst der Bedienelemente ohne weitere Funktionalität (Gebhardt, 2003). Im hier verfolgten Entwicklungsprozess hatten die erstellten Mockups zusätzlich die Aufgabe, als Implementierungsspezifikation für den Programmierer zu dienen.

III.2.2 Prototyp

III.2.2.1 Dialogprinzipien

Die Bildschirmaufteilung erfolgte in Anlehnung an die Richtlinien VDI/VDE 3699-4 (1997) und dem Layout des PE/SEE Frameworks (vgl. Kapitel III.3). Abbildung 40 zeigt die Oberfläche des Prototypen mit Markierungen der relevanten Interaktionsareale. Ein vollständiger Überblick über den interaktiven Nutzerdialog findet sich in Anhang B.1.3. Diese werden im Folgenden weiter erläutert.

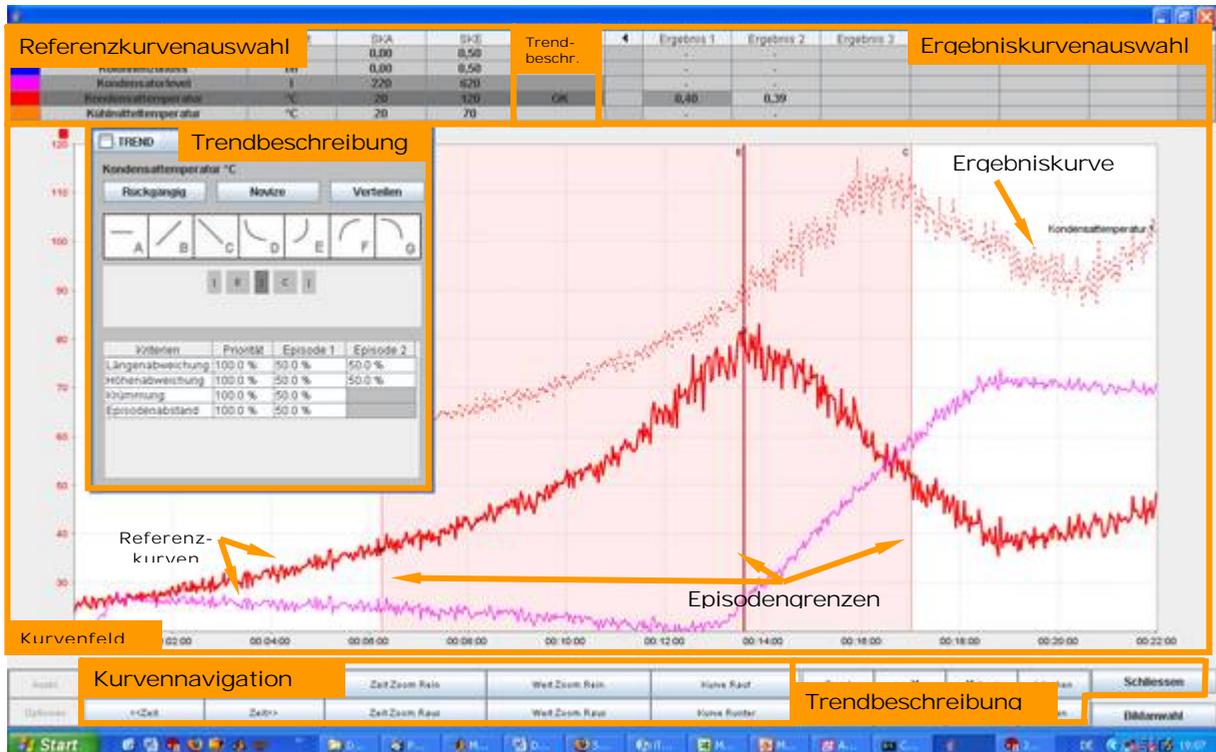


Abbildung 40 Dialog-Oberfläche des Nutzerdialogs (orange gefärbte Rahmen und Pfeile kennzeichnen die Hauptinteraktionsfelder)

III.2.2.2 Referenzkurvenauswahl

Zur Beschreibung eines Trends wählt der Nutzer eine Referenzkurve aus. Eine Referenzkurve ist die Kurve einer beliebigen Variablen aus der Referenz-Kurvenschar. Diese Referenzkurve dient als visueller bzw. inhaltlicher Ankerpunkt. Der Variablenname ist dabei die einzige Information, welche direkt für die Suchanfrage übernommen wird.

Die Kurven aller Variablen einer Referenz-Kurvenschar sind zeitlich voneinander abhängig. Die Skalierung und Positionierung der Werteachse kann jedoch unabhängig voneinander erfolgen. Um dies zu ermöglichen, müssen sich die Kurven einzeln „anfassen“ lassen. Darüber hinaus soll das Kurvenfeld Raum für Referenz- und Ergebniskurven bieten. Die Richtlinie VDI/VDE 3699-4 (1997) empfiehlt konform zu den Arbeiten von Miller (1956) maximal sieben verschiedene Kurven pro Kurvenfeld. Arbeitet der Nutzer beispielsweise mit einer Kurvenschar von 5 Kurven und erhält er drei Ergebnis-Kurvenscharen mit je fünf Kurven, liegen insgesamt 20 anzeig- und vergleichbare Kurven vor. Um ein effektives Arbeiten für den Nutzer zu ermöglichen, muss ihm die Möglichkeit gegeben werden, einzelne Kurven auszublenden.

Um die genannten Interaktionskriterien zu erfüllen, wurden drei verschiedene Zustandsmodi der Referenzkurven definiert, welche sich alle per Mausklick aktivieren lassen (siehe Tabelle 18). Der Wechsel zwischen den Modi wird auf direktem Wege durch einmaliges oder zweimaliges Betätigen der linken Maustaste vorgenommen. Abbildung 41 zeigt die entsprechende Zustandsübergangsmatrix.

Ein Modus-Wechsel einer Variablen erfolgt jedoch auch indirekt über den Modus-Wechsel einer anderen Variablen. Dies resultiert aus den folgenden zwei Bedingungen:

1. Es kann sich immer nur eine Referenzkurve im Navigationsmodus befinden.
2. Es ist immer mindestens eine Referenzkurve im Navigationsmodus.

Die zweite Bedingung garantiert, dass immer eine Werteachse für das Kurvenfeld vorhanden ist. Zudem bleibt der Mausklick-Dialog konsistent.

Tabelle 18 Zustandsmodi für Kurvenauswahl

Modus I	ausgeblendet	die Zeile der Variablen im Bereich „Referenzkurvenauswahl“ erscheint hellgrau keinerlei Daten im Kurvenfeld angezeigt
Modus II	Ansicht	die Zeile der Variablen im Bereich „Referenzkurvenauswahl“ erscheint mittelgrau Kurven werden mit gemeinsamer Zeitachse dargestellt und lassen sich entlang dieser gemeinsam zoomen und verschieben
Modus III	Navigation	die Zeile der Variablen im Bereich „Referenzkurvenauswahl“ erscheint dunkelgrau Werteachse der Kurve wird angezeigt Zoomen und Verschieben entlang der Werteachse möglich



Abbildung 41 Zustandsübergangsmatrix für die Modi-Wechsel der Referenzkurven.
Durchgezogene Linie = Einmalklick der linken Maustaste
Gestrichelte Linie = Doppelklick der linken Maustaste

Abbildung 42 und Abbildung 43 veranschaulichen die Konsequenz für den Nutzerdialog. Wechselt eine Variable A vom Navigationsmodus in den Ansichtsmodus (obere Graphik der Abbildung 42), so wird automatisch diejenige Variable in den Navigationsmodus geschaltet, welche

- in der Auswahlliste über der Variablen A steht und
- sich im Ansichtsmodus befindet.

Steht in der Auswahlliste über der Variablen A keine Variable im Ansichtsmodus, wird diejenige Variable gewählt, welche an unterster Position in der Auswahlliste steht und sich zusätzlich im Auswahlmodus befindet (untere Graphik der Abbildung 42).

Ist keine weitere Variable im Ansichtsmodus vorhanden (obere Graphik der Abbildung 43), wird automatisch die darüber stehende Variable in den Navigationsmodus geschaltet (untere Graphik der Abbildung 43). Steht die aktuelle Variable an höchster Stelle der Auswahlliste, wird automatisch die unterste Variable gewählt.

	Variable	Einheit	SKA	SKE	Trend?
	Kühlwasserdurchfluss	l/h	0,00	0,50	
	Kolonnenzufluss	l/h	0,00	0,50	
	Kondensatorlevel	l	220	620	
	Kondensattemperatur	°C	20	120	OK
	Kühlmitteltemperatur	°C	20	70	

	Variable	Einheit	SKA	SKE	Trend?
	Kühlwasserdurchfluss	l/h	0,00	0,50	
	Kolonnenzufluss	l/h	0,00	0,50	
	Kondensatorlevel	l	220	620	
	Kondensattemperatur	°C	20	120	OK
	Kühlmitteltemperatur	°C	20	70	

Abbildung 42 Modi-Wechsel zwischen den unteren drei Variablen durch einmaliges Anklicken der sich im Navigationsmodus befindenden Variablen.
SKA = Skalenanfang, SKE = Skalenende

	Variable	Einheit	SKA	SKE	Trend?
	Kühlwasserdurchfluss	l/h	0,00	0,50	
	Kolonnenzufluss	l/h	0,00	0,50	
	Kondensatorlevel	l	220	620	
	Kondensattemperatur	°C	20	120	OK
	Kühlmitteltemperatur	°C	20	70	

	Variable	Einheit	SKA	SKE	Trend?
	Kühlwasserdurchfluss	l/h	0,00	0,50	
	Kolonnenzufluss	l/h	0,00	0,50	
	Kondensatorlevel	l	220	620	
	Kondensattemperatur	°C	20	120	OK
	Kühlmitteltemperatur	°C	20	70	

Abbildung 43 Modi-Wechsel der Variable „Kondensat-Temperatur“ von „Deaktiviert“ in „Navigation“ durch einmaliges Anklicken der Variablen „Kühlmitteltemperatur“

III.2.2.3 Kurvenfeld

Im Kurvenfeld werden alle aktivierten Referenz- und Ergebniskurven angezeigt. Unterschiedliche Variablen werden über unterschiedliche Farbcodes identifiziert. Im Navigationsmodus wird die Linie der Kurve durch eine verbreiterte Linie hervorgehoben. Zusätzlich wird die Werteachse über den Farb-Code definiert.

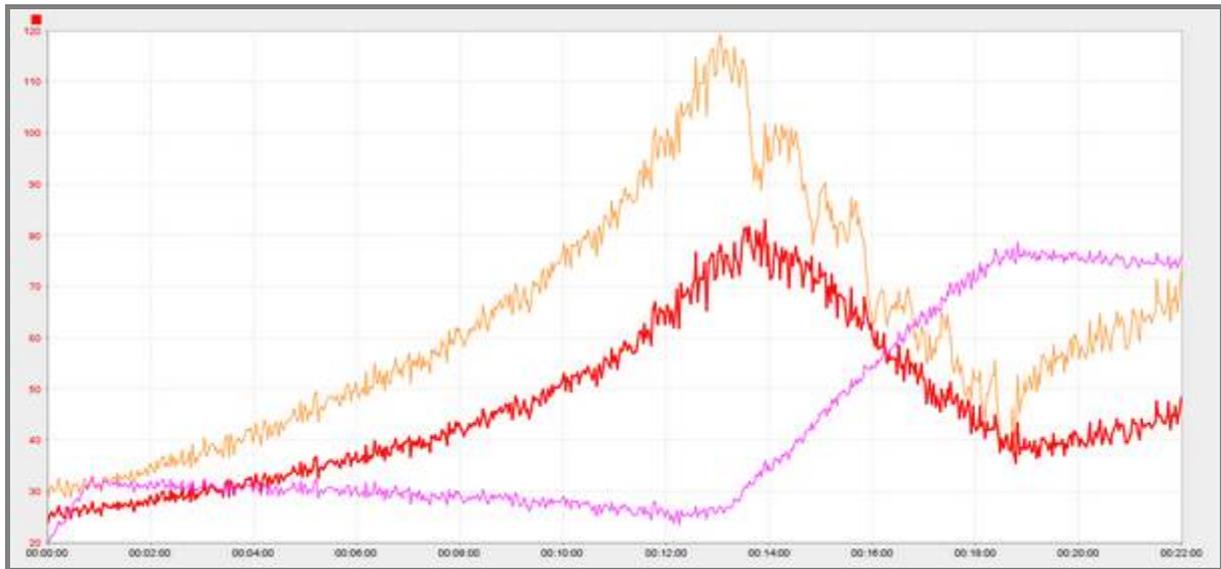


Abbildung 44 Kurvenfeld (Variable der rot gefärbten Kurve ist im Navigationsmodus)

III.2.2.4 Kurvennavigation

Der Nutzer hat entsprechend der Richtlinie VDI/VDE 3699-4 (1997) die Möglichkeit, den Zeitabschnitt und die Zoomstufen entlang der Zeit- und Werteachse zu wählen. Alle Kurven der Referenz-Kurvenschar können gemeinsam entlang der Zeitachse verschoben werden. Der im Kurvenfeld angezeigte Zeitausschnitt lässt sich ebenfalls für alle Kurven einer Referenz-Kurvenschar gemeinsam vergrößern bzw. verkleinern. Alle Variablen einer Referenz-Kurvenschar werden unabhängig voneinander entlang der Werteachse Y verschoben. Die Auflösung der Werteskala wird ebenso individuell gewählt. Die Zoomfunktionen orientieren sich an der horizontalen bzw. vertikalen Mitte des Kurvenfeldes.

III.2.2.5 Trendbeschreibung

Bei der Trendbeschreibung wird dem Nutzer freigestellt, wie stark er sich von den Rohwerten der Referenzkurve entfernt. Mit Aktivierung des Trendfensters (Abbildung 45) wird das Kurvenfeld sensitiv gegenüber Mausklicks, welche das Setzen von Episodengrenzen in Form von vertikalen Linien ermöglicht. Aus den Schnittpunkten der Episodengrenzen mit der ausgewählten Referenzkurve werden automatisch die Amplituden der Episoden abgeleitet. Die Selektion von bereits angelegten Episoden sowie deren Grenzen erfolgt im Trendfenster. Dort lassen sich den Episoden zudem die Trendlets zuordnen. Die Koordinaten der ausgewählten Episodengrenzen lassen sich im Trendbeschreibungs-Panel verändern.

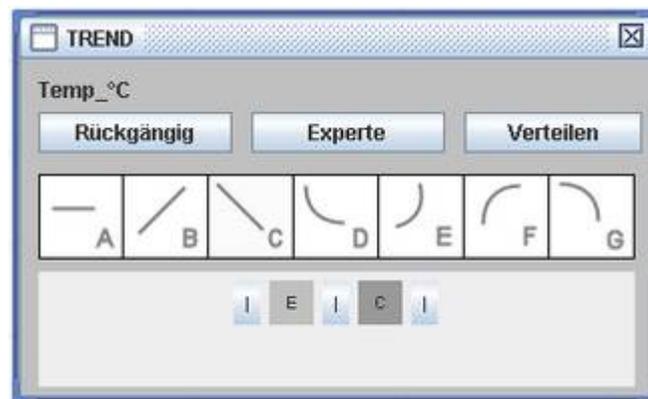


Abbildung 45 Fenster für Trendbeschreibung. Für zwei Episoden wurden die Trendlets „E“ und „C“ gewählt. Der dunkelgraue Ton des Interaktionsfelds der Episode 2 signalisiert dessen Aktivierung. Das Trendlet kann per Mausklick geändert werden.

III.2.2.6 Expertenmodus

Alle weiteren, der in Kapitel III.1.2.3 vorgestellten Suchparameter werden automatisch als Default gesetzt. Einige lassen sich im Expertenmodus (Abbildung 47) verändern. Für jede Episode lassen sich die maximal erlaubten Längen- und Höhenabweichungen prozentual bezogen auf die jeweilige Episodenlänge bzw. –höhe fest legen. Die „Länge“ einer Episode bezieht sich auf die zeitliche Ausdehnung, die „Höhe“ auf die Ausprägung des Werteunterschieds zwischen Start- und Endzeit-

punkt. Weiterhin kann für alle gekrümmten Episoden die gesuchte Krümmungsstärke spezifiziert werden. Eine Krümmung von 0% entspricht dabei einer Geraden, eine Krümmung von 100% entspricht der Approximation durch ein Dreieck (Abbildung 46).

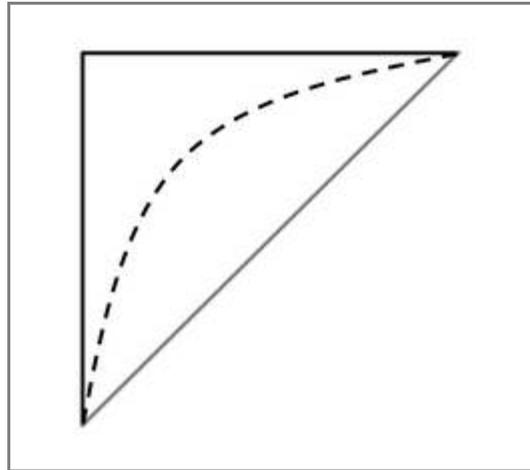


Abbildung 46 Krümmungsstärken. Grau: 0% Gestrichelt: \approx 40% Schwarz: 100%

Obwohl ein Trend in Form von direkt aufeinander folgenden Episoden definiert ist, kann für jedes Episodenpaar ein maximaler Abstand zwischen dem Endzeitpunkt der vorherigen und dem Startzeitpunkt der folgenden Episode von 0-100% zugelassen werden. Die Prozentangabe bezieht sich dabei auf die Länge der kürzeren von beiden Episoden. Für die Berechnung der Ähnlichkeitsindikatoren der Ergebnisse können alle vier Kriterien (Längenabweichung, etc.) unterschiedlich gegeneinander gewichtet werden (vgl. Abbildung 47).



Abbildung 47 Expertenmodus für die Trendbeschreibung

III.2.2.7 Ergebniskurvenauswahl

Es werden die besten sechs Suchergebnisse in einer Tabelle in der rechten oberen Ecke des Prototypen aufgelistet. Die Zeile der Variable, für welche ein Trend beschrieben wurde, zeigt die errechneten Ähnlichkeitsindikatoren (vgl. Kapitel III.1.2.3). Die Auswahl der Ergebniskurven erfolgt analog zur Auswahl der Referenzkurven. Es lassen sich die Kurven beliebig vieler Variablen der Ergebnisse anzeigen, aber nur eine Kurve kann in den Navigationsmodus gelangen. Der Wechsel zwischen den Modi weicht insofern vom Modi-Wechsel der Referenzkurven ab, als das

- ein einfacher Mausklick auf eine Zelle im Navigationsmodus die dort hinterlegte Kurve komplett ausblendet (Abbildung 48), und
- keine indirekten Modi-Wechsel erfolgen.



Abbildung 48 Zustandsübergangsmatrix für die Modi-Wechsel der Ergebniskurven.
 Durchgezogene Linie = Einmalklick der linken Maustaste
 Gestrichelte Linie = Doppelklick der linken Maustaste

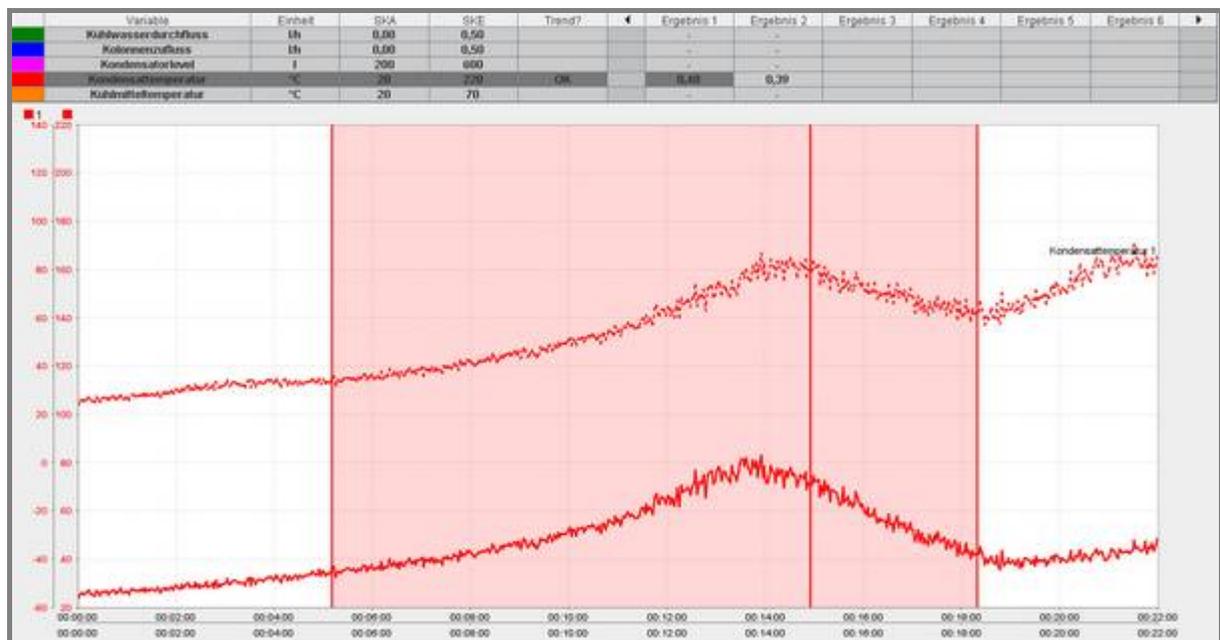


Abbildung 49 Ansicht einer Ergebniskurve im Navigationsmodus (farbliche Unterlegung des Ergebnistrends inklusive der Episodengrenzen)

III.2.2.8 Ergebniskurvennavigation

In Ergänzung des Kapitels III.2.2.4 sei an dieser Stelle der vollständige Regelsatz zur Kurvennavigation aufgeführt:

- Alle Referenz- und Ergebniskurven werden entlang der Zeitachse gemeinsam vergrößert und verkleinert.
- Alle Kurven der Referenz-Kurvenschar und eines beliebigen Ergebniskurvensets werden entlang der Zeitachse gemeinsam verschoben.

- Die Referenzkurve und alle Ergebniskurven einer Variablen werden entlang der Werteachse gemeinsam vergrößert und verkleinert.
- Alle Referenz- und Ergebniskurven können entlang der Werteachse unabhängig voneinander verschoben werden.

Um den Vergleich von Referenz- und Ergebniskurven zu vereinfachen, wird im Fenster „Trend“ (Abbildung 45 & Abbildung 47) eine Verteilungsfunktion angeboten. Alle sich im Ansichtsmodus befindenden Referenz- und Ergebniskurven werden damit entlang der Ordinate verteilt (Abbildung 49).

111.2.3 Expertenevaluation

Zur Evaluation von Nutzerdialogen kann auf eine erschöpfende Menge an quantitativen und qualitativen Methoden zurück gegriffen werden. Die Güte der quantitativen Methoden variiert entlang der Dimensionen *Kontrollierbarkeit* und *Generalisierbarkeit*. Obwohl etwaige Störvariablen in experimentellen Laboruntersuchungen oft sehr gut kontrollierbar sind, muss hier allzu oft auf artifizielle Aufgabenszenarien und auf nicht repräsentative Stichproben^a zurückgegriffen werden. Eine andere Möglichkeit bietet die qualitative Evaluation durch Usability-Experten. Meist sind solche Experten tätig als Interaktionsdesigner, Softwareentwickler, Software-Ergonom oder Human-Factors Spezialist, und damit besonders sensibilisiert für Probleme der Gebrauchstauglichkeit. Mit einem geringen Schulungsaufwand können sie deshalb mit verschiedenen qualitativen Methoden der Usability-Evaluation hilfreiche Rückmeldungen für den Entwurfsprozess geben. Allerdings sind Usability-Experten keine Nutzer und verfügen darum selten über Domänen- und Anwendungsexpertise. Deshalb liefern Expertenevaluationen hauptsächlich regelbasierte statt erfahrungsbasierte Hinweise auf Usability-Probleme. Letztere haben jedoch oft schwerwiegendere Auswirkungen für die Nutzer. Die Expertenevaluation wird deshalb nicht als Substitut zu Usability-Tests mit realen Nutzern (siehe Kapitel IV), sondern als Ergänzung im Vorfeld verstanden. Im Folgenden seien kurz einige Methoden der Expertenevaluation erläutert.

^a Da beispielsweise echte Anlagenfahrer nur selten zu einem 2-3 Stunden andauernden Laborversuch zu gewinnen sind, werden oftmals Studenten und/oder Arbeitslose als Versuchspersonen gewonnen.

Beim *Cognitive Walkthrough* (Lewis & Wharton, 1997; Wharton et al., 1994) bearbeitet der Experte konkret vorgegebene Handlungsabläufe und hinterfragt dabei, ob ein potentieller Nutzer diese in Interaktion mit dem vorliegenden Interface folgen würde. Dabei wird sich insbesondere auf die mentalen Prozesse eines hypothetischen Nutzers konzentriert. Zu Beginn der Evaluation muss deshalb das Vorwissen dieser potentiellen Nutzer definiert werden. Beim Cognitive Walkthrough erkannte Probleme werden begründet und Problemlösungen vorgeschlagen.

Ein weiter ausgearbeitetes Verfahren zur Bewertung der Software-ergonomischen Qualität von Benutzungsschnittstellen stellt EVADIS II dar (Oppermann et al., 1992). Kernstück bildet eine Checkliste von Software-ergonomischen Kriterien, die ein benutzerfreundliches Interface aufweisen sollte. Begleitend kann ein Fragebogen eingesetzt werden, um die Benutzereigenschaften in die Analyse einbeziehen zu können. Außerdem existiert ein Leitfaden zur Erstellung der Prüfaufgaben, der die Auswahl standardisierter Prüfaufgaben unterstützen soll.

Jakob Nielsen (1994) hat mit der Heuristischen Evaluation eine weitere formative^a Bewertungsmethode vorgestellt. Eine kleine Gruppe von ca. fünf Experten versucht dabei, möglichst viele Probleme der Gebrauchstauglichkeit zu identifizieren. Dafür nehmen sie teilweise die Perspektive potentieller Systembenutzer ein und werden durch eine Liste an Heuristiken unterstützt. Ein Beispiel einer derartigen Heuristik lautet: *„The system should continuously inform the user about what it is doing and how it is interpreting the user's input“* (Nielsen, 1993, S. 134).

Die Zuordnung der einzelnen gesammelten Probleme zu den kategorialen Heuristiken erlaubt zudem verallgemeinernde Aussagen anhand von deskriptiver Statistik. Diese Form der Quantifizierung wird verstärkt durch eine Vergabe an Prioritäten, mit welchen die einzelnen Probleme gelöst werden sollten (siehe Mehrere Autoren haben Vorschläge zur Kategorisierung von Problemen der Gebrauchstauglichkeit publiziert. Für die vorliegende Studie wurden verschiedene Kataloge (EN ISO 9241-10, 1996; Nielsen, 1993; Oppermann et al., 1992) zusammengefasst und um Kriterien reduziert, welche nicht in den hier adressierten Kontext passten (vgl. Anhang B.1.4.4). Diese

^a Mit „formativ“ wird eine Methode bezeichnet, deren Ergebnisse in den laufenden Entwicklungsprozess mit aufgenommen werden können. Eine abschließende Evaluation eines fertig entwickelten System bezeichnet man als „summativ“.

Vorgehensweise war nötig, um den Experten die Aufgabe zu erleichtern, die erarbeiteten Mängel möglichst vollständig und trennscharf zu kategorisieren. Tabelle 20 zeigt die integrierte und reduzierte Liste an Evaluationskriterien. Für die Beantwortung der zweiten Fragestellung wurde den Experten das an Nielsen (1994) angelehnte Bewertungssystem der Fehler! Ungültiger Eigenverweis auf Textmarke. zur Verfügung gestellt.

Tabelle 21 in Kapitel III.2.3.1).

Während bei der heuristischen Evaluation die Experten erst zu einem späten Zeitpunkt miteinander in den Diskurs gehen, ist die Durchführung von Fokus-Gruppen von Beginn an auf die gemeinsame Reflektion des Nutzerdialogs ausgelegt (Greenbaum, 1993). Sechs bis neun Benutzern werden von einem Moderator während des Austausches von zuvor erlebten Interfacelösungen begleitet.

Für die Evaluation des entworfenen Nutzerdialogs wurden in einem hybriden Ansatz die hier vorgestellten Methoden integriert. Als wesentliche methodische Aspekte wurden integriert:

- die aufgabenorientierte Bearbeitung des Nutzerdialogs
- die Ableitung von Problemen der Gebrauchstauglichkeit anhand von Heuristiken bzw. Checklisten
- die Gruppen-Reflektion von Erfahrungen mit dem Nutzerdialog
- die Zuordnung von Einzelproblemen zu übergeordneten Kategorien
- die Bewertung einzelner Probleme der Gebrauchstauglichkeit hinsichtlich ihrer Priorität für den weiteren Entwicklungsprozess

Im Laufe der folgenden zwei Kapiteln soll die Durchführung der Evaluationsstudie sowie deren Ergebnisse dargestellt werden.

III.2.3.1 Evaluationsstudie

Auf der Frühjahrsschule 2007 des Zentrum Mensch-Maschine-Systeme wurde eine Expertenevaluation im Rahmen eines Workshops durchgeführt. Ziel war die effiziente Aufdeckung und Einschätzung von Problemen der Gebrauchstauglichkeit beim Nutzerdialog und dessen Interface-Implementierung. Die Funktionalität des Suchalgorithmus war dabei kein Evaluationsobjekt. Die

Ergebnisse dienten als Grundlage für ein Re-Design des Prototypen im Vorfeld einer ganzheitlichen Evaluation inklusive Suchalgorithmus (siehe Kapitel IV).

Vier Gruppen a drei Usability-Experten arbeiteten 30 Minuten mit dem Prototypen (siehe Zeitplan in Anhang B.1.4.1). Sie erhielten dafür keinerlei Einführung und keine Bedienungsanleitung. Angeleitet wurden sie lediglich durch eine genaue Auflistung der auszuführenden Interaktionsschritte (siehe Anhang B.1.4.3). Die Aufgaben lauteten zum Beispiel:

- „Beschreibe für die Temperaturkurve zwischen 00:08:00 und 00:18:00 Uhr einen Trend bestehend aus zwei Episoden mit den Trendlets E & C.“
- „Wechsel in den Expertenmodus und setze den zulässigen Episodenabstand zwischen beiden Episoden auf 80%.“

Tabelle 19 Rollen und Aufgaben der Expertengruppen

Rolle	Aufgabe
Benutzer	Interaktion mit dem Prototypen Bericht aller Probleme der Gebrauchstauglichkeit
Protokollant	Auflistung und vorläufige Kategorisierung aller berichteten Probleme der Gebrauchstauglichkeit
Beobachter	Unterstützung des Benutzers bei der vollständigen und erfolgreichen Bearbeitung der Aufgabenliste sowie beim Erkennen von Problemen der Gebrauchstauglichkeit. Unterstützung des Protokollanten bei der effizienten und vollständigen Dokumentation aller Probleme der Gebrauchstauglichkeit

Die Mitglieder jeder Gruppe wurden einzelnen Rollen zugeteilt. Jede Rolle hatte verschiedene Aufgaben zu erfüllen (Tabelle 19). Die Arbeit in Kleingruppen mit verteilten Rollen beschleunigte den Evaluationsprozess aufgrund folgender Vorteile:

- parallele Bearbeitung der Interaktions- und Dokumentationsaufgabe
- kooperatives Problemlösen in Sackgassen explorativer Interaktion
- frühzeitige Verständigung über Problembeschreibung und –kategorisierung

Aufgrund der unerwartet hohen Anzahl an protokollierten Usability-Problemen wurden den Kleingruppen weitere zehn Minuten Zeit gegeben, sich intern auf maximal fünf wesentlichen Probleme zu einigen. Danach erhielten alle Kleingruppen die Möglichkeit, diese Probleme zu benennen. Der Moderator fasste diese in einer Liste exklusiver, eindeutiger und für alle Experten verständlicher Usability-Probleme zusammen. Alle Items der so entstandenen Liste wurden anschließend von allen Experten in Einzelarbeit bewertet. Zwei Fragen wurden dabei beantwortet:

- Welchen Kriterien der Gebrauchstauglichkeit lassen sich die einzelnen Probleme zuordnen?
- Wie schwer wirken die einzelnen Probleme auf die Gebrauchstauglichkeit?

Tabelle 20 Kriterien der Gebrauchstauglichkeit

Kategorie	Erläuterung
externe Konsistenz	Systemkonformität mit Normen und Konventionen eines durchschnittlichen Nutzerinterface (Sprache, Anordnung, Dialogkonzepte)
interne Konsistenz	einheitliche interne Verwendung von Sprache, Zuständen und Aktionen
Adaptierbarkeit	Anpassungsmöglichkeiten in Abhängigkeit von Nutzerfertigkeiten, z. B. in Form von Tastenkombinationen, Parametrisierung, Oberflächenanpassungen, etc.
Dialogkomfort	Effizienz der Dialogschritte
Informationsdarstellung	Effizienz der Informationsdarstellung (irrelevante oder extrem selten zu gebrauchende Informationen sollten den Nutzerdialog nicht stören)
Gedächtnisstützen	Erkennen statt Erinnern von Objekten, Aktionen und Optionen
Übersichtlichkeit	Übersichtlichkeit der Darstellung (Effizienz bei der Suche nach Eingabe- und Ausgabefeldern)
Selbstbeschreibungsfähigkeit	Explorationseffizienz
Fehlerprävention	Vorbeugen von Nutzerfehlern z.B. in Form von Warnhinweisen oder Eingabebeschränkungen

Fehlertoleranz	Systemstabilität und Rückführbarkeit (z.B. Funktion „Rückgängig“)
Fehlermeldung	Vollständigkeit, Verständlichkeit und Konstruktivität von Fehlermeldung
Erkennbarkeit des Systemzustandes	Status, Ursächlichkeit und Prognose aktueller Rechenprozesse.
Verfügbarkeit	funktionale Korrektheit, Reaktionszeiten des technischen Systems

Mehrere Autoren haben Vorschläge zur Kategorisierung von Problemen der Gebrauchstauglichkeit publiziert. Für die vorliegende Studie wurden verschiedene Kataloge (EN ISO 9241-10, 1996; Nielsen, 1993; Oppermann et al., 1992) zusammengefasst und um Kriterien reduziert, welche nicht in den hier adressierten Kontext passten (vgl. Anhang B.1.4.4). Diese Vorgehensweise war nötig, um den Experten die Aufgabe zu erleichtern, die erarbeiteten Mängel möglichst vollständig und trennscharf zu kategorisieren. Tabelle 20 zeigt die integrierte und reduzierte Liste an Evaluationskriterien. Für die Beantwortung der zweiten Fragestellung wurde den Experten das an Nielsen (1994) angelehnte Bewertungssystem der Fehler! Ungültiger Eigenverweis auf Textmarke. zur Verfügung gestellt.

Tabelle 21 Bewertungssystem zur Evaluation von Problemen der Gebrauchstauglichkeit

Note	Beschreibung
1	kein Problem für die Gebrauchstauglichkeit
2	kosmetisches Problem für die Gebrauchstauglichkeit
3	kleines Problem für die Gebrauchstauglichkeit
4	großes Problem für die Gebrauchstauglichkeit
5	Katastrophe für die Gebrauchstauglichkeit

III.2.3.2 Ergebnisse

Die einzelnen Aufgaben, insgesamt 24 an der Zahl (siehe Anhang B.1.4.3), können zu insgesamt sieben Aufgabenabschnitte zusammengefasst werden. Während der Bearbeitung der einzelnen Aufgabenabschnitte wurden insgesamt 44 verschiedene Probleme der Gebrauchstauglichkeit festgestellt. Für den Aufgabenkomplex „Trendbeschreibung“ wurden die meisten Probleme identifiziert.

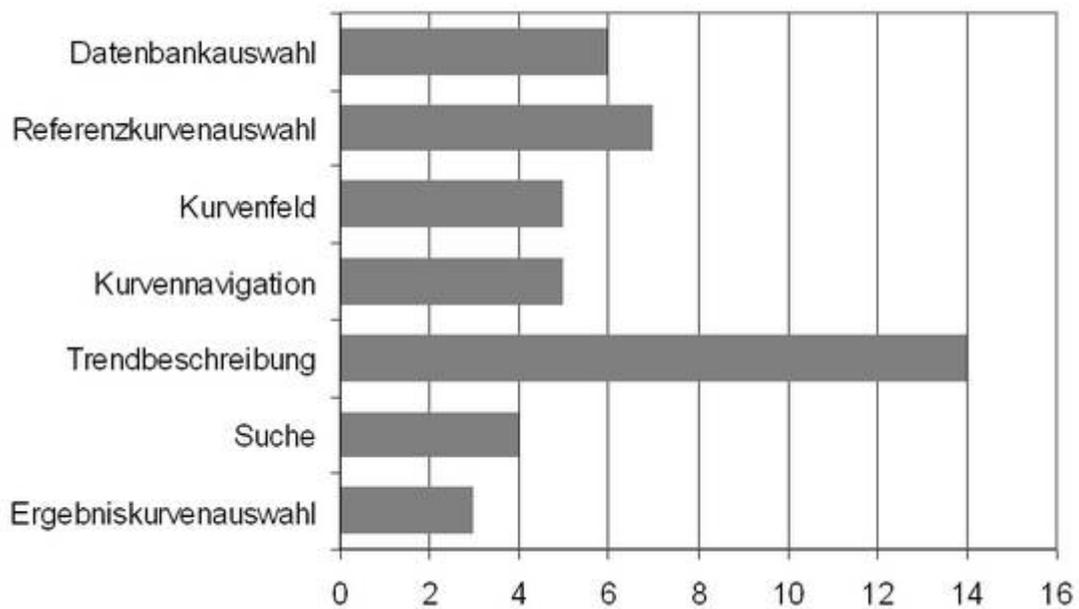


Abbildung 50 Anzahl identifizierter Probleme der Gebrauchstauglichkeit je Aufgabenabschnitt

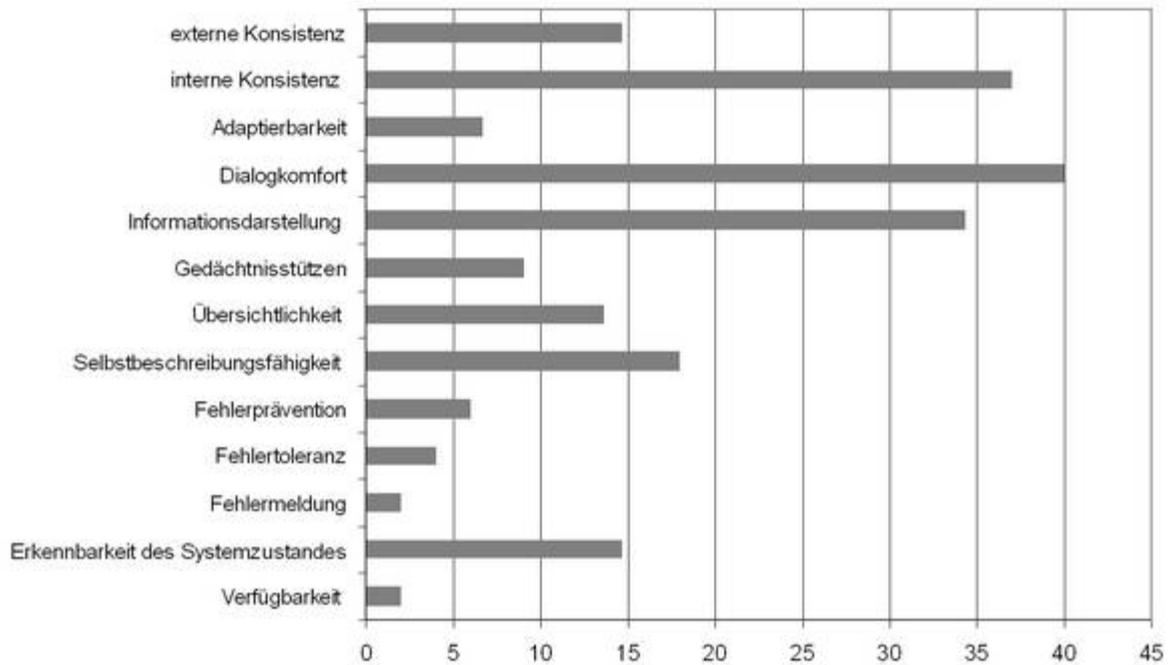


Abbildung 51 Anzahl der Zuordnung identifizierter Probleme der Gebrauchstauglichkeit zu den einzelnen Kriterien aus Tabelle 20

Sechzehn der 44 Probleme wurden von den Experten besonders hervorgehoben und in einem weiteren Schritt (siehe Kapitel III.2.3.1) kategorisiert und bewertet. Wie Abbildung 51 zeigt, betrafen die ausgewählten Probleme der Gebrauchstauglichkeit vor allem den Dialogkomfort, die interne Konsistenz und die Informationsdarstellung. Es wurden dagegen wenige Probleme aufgelistet, die sich auf Benutzerfehler und Systemverfügbarkeit beziehen.

Die Bewertung der Probleme fiel im Mittel positiv aus. Die Experten waren also der Meinung, dass der evaluierte Nutzerdialog überwiegend kosmetische und kleine Probleme aufweist. Einzig der Aufgabenabschnitt „Datenbankauswahl“ lieferte Probleme, deren Bandbreite (bzw. Standardabweichung) von „kosmetisch“ bis „groß“ verlief.

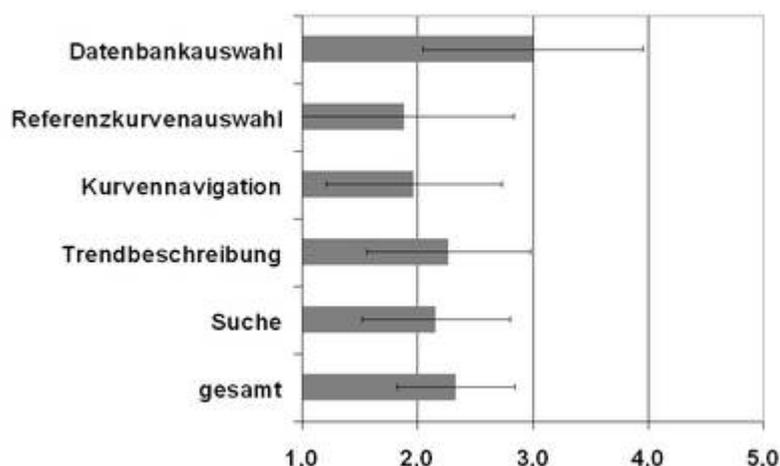


Abbildung 52 Durchschnittliche Bewertung (inklusive Standardabweichung) ausgewählter Probleme der Gebrauchstauglichkeit, sortiert nach Aufgabenabschnitten.

III.2.4 Re-Design

Im Rahmen eines Re-Designs konnte die Hälfte der insgesamt 44 aufgezeigten Probleme der Gebrauchstauglichkeit behoben werden (Abbildung 53). Ein Drittel war durch Instruktionen vermeidbar, welche den Experten vor ihrer Aufgabenbearbeitung nicht bereitgestellt wurden. Sechs Vorschläge für Usability-Probleme wurden nach weiteren Analysen abgewiesen. Nur drei Probleme erfordern Softwareänderungen in einem Umfang, welcher im Rahmen des hier bearbeiteten Entwicklungszyklus nicht mehr zu leisten war. Tabelle 22 zeigt einige Beispiele für die entsprechenden Problemkategorien.

Tabelle 22 Beispiele für verschiedene Kategorien von Problemen der Gebrauchstauglichkeit während des Re-Designs

Kategorie	Beispielproblem
behoben	Objektbezug der Löschfunktion unklar.
erlernbar	Verschiedene Aktivierungsmodi der Variablen unklar.

diskutabel	Funktion der Buttons "Kurve rauf" und "Kurve runter" nicht erkennbar.
zu aufwändig	Maus ermöglicht Setzen von Episodengrenzen, nicht aber das Verschieben der Grenzen entlang der Zeitachse bzw. deren Werte entlang der Ordinate.

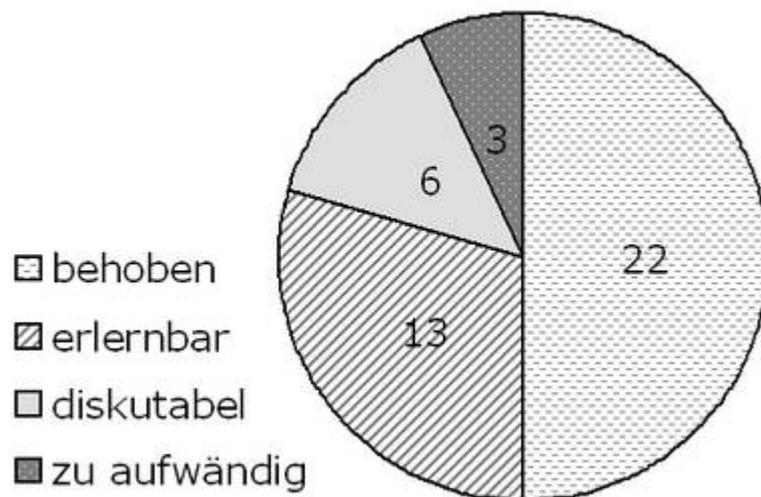


Abbildung 53 Verteilung der Probleme der Gebrauchstauglichkeit im Re-Design

III.3 Implementierung

Für die prototypische Implementierung der Nutzeroberfläche wurde die Simulationsumgebung PE/SSE (ProperEduct/SmallSystemsEdition; Urbas, 1999) gewählt. Das Java basierte Framework stellt eine Emulation eines industriellen Prozessleitsystems dar. Aufgrund seiner skalierbaren Architektur wird die effiziente Erstellung verschiedener experimenteller Mikrowelt-Szenarien ermöglicht. Mikrowelten stellen einen viel versprechenden Kompromiss dar. Zum einen wird die unkontrollierte Komplexität der realen Welt systematisch reduziert und abstrahiert. Zum anderen wird im Vergleich zu artifiziellen Laborexperimenten eine erhöhte ökologische Validität gewährleistet. Weitere Vorteile von PE/SSE sind die Online-Fähigkeit, die Schnittstellen zu Prozessmodellen und die Möglichkeit zur automatischen Generierung von Interaktionsprotokollen.

Für die Implementierung des Prototypen im Rahmen des PE/SSE konnte auf eine umfangreiche Klassenbibliothek (Buttons, Panels, etc.) zurückgegriffen werden. Zugleich wurde diese Bibliothek durch Erstellung neuer Klassen (z.B. `HistorianGraph.java`) erweitert. Zudem ist eine Verknüpfung der augenblicklichen Standalone-Anwendung mit weiteren Experimentalszenarien (z.B. Monitoring einer Destillationskolonne) möglich. Damit wird der Prototyp für eine nachhaltige Anwendung in weiteren wissenschaftlichen Studien zur Verfügung gestellt.

Die Softwarearchitektur basiert auf dem MVC-Modell (Model-View-Controller), welches aufgrund der modularen Wiederverwendbarkeit einzelner Komponenten den iterativen Entwurfsprozess erleichtert (Lahres & Rayman, 2006). Das Modell enthält Daten und Methoden, ohne Annahmen über die Herkunft, die Darstellung, den Zusammenhang oder die Steuerung der Daten zu machen.

Die *View* übernimmt die Darstellung relevanter Daten aus dem Modell und aktualisiert diese bei Änderungen innerhalb des Modells. Weiterhin leitet sie Benutzeraktionen an die Steuerung weiter.

Die Weiterverarbeitung der vom Benutzer übergebenen Daten regelt der *Controller*, wie auch alle weiteren Kommunikationsschritte zwischen Modell und View.

Abbildung 54 zeigt die Kommunikation der Komponenten im MVC-Modell. Die View registriert Anfragen oder Aktionen des Nutzers, und leitet diese an den Controller weiter. Der Controller sorgt

für die entsprechenden Methodenaufrufe im Model, welches die gewünschten Änderungen auf dessen Daten ausführt. Eine Änderung dieser Daten wird entweder vom Model selbst oder vom Controller an die View gemeldet, welche sich daraufhin aktualisiert.

Abbildung 54 zeigt weiterhin die Interaktion mit der Datenbank, deren Inhalte deckungsgleich in den externen Programmen MATLAB® und Microsoft® Excel abgelegt sind. Bei dem Start einer Suchanfrage generiert der Controller eine temporäre Datei wxout.m. In dieser stehen alle Parameter der Suchanfrage sowie ein Aufruf der Datei wxaux.m. Diese wiederum führt die Suche in einer Datenbank aus MATLAB-Datenblättern (*.mat). Abschließend wird von ihr eine temporäre Datei wxout.xml geschrieben (vgl. Quellcode 3). In ihr steht die Spezifikation der Suchergebnisse in Form der entsprechenden Excel-Tabelle (Zeile 12), Tabellenblatt (Zeile 13), Variablenname (Zeile 13), Start- und Endzeitpunkt des Ergebnistrends (Zeile 14), Ähnlichkeitsindikator (Zeile 14) sowie die Start- und Endzeitpunkte der einzelnen Ergebnis-Episoden (Zeile 15 & 16).

Der Controller greift auf Basis dieser Informationen über eine JDBC-Schnittstelle auf die entsprechenden Daten zu und übergibt diese an die Modelkomponente *Kurvenbild*. In der Modelkomponente *Trendbeschreibung* werden alle Informationen der Suchanfrage abgelegt. Die View gliedert sich analog dazu in die Hauptkomponenten *Kurvenbild* und *Trendbeschreibung*, wobei erstere entsprechend der Aufteilung der Nutzeroberfläche in die Subkomponenten *Kurvenfeld*, *Kurvenauswahl* und *Kurvennavigation* unterteilt werden.

Quellcode 3: Inhalt eines von MATLAB R2006b generierten Suchoutputs

```
10 <search>
11   <database name="database">
12     <table name="[page16$]">
13       <column name="signal">
14         <trend start="332" end="505" score="0.955">
15           <episode start="332" end="415"/>
16           <episode start="416" end="505"/>
17         </trend>
18       </column>
19     </table>
20   </database>
21 </search>
```

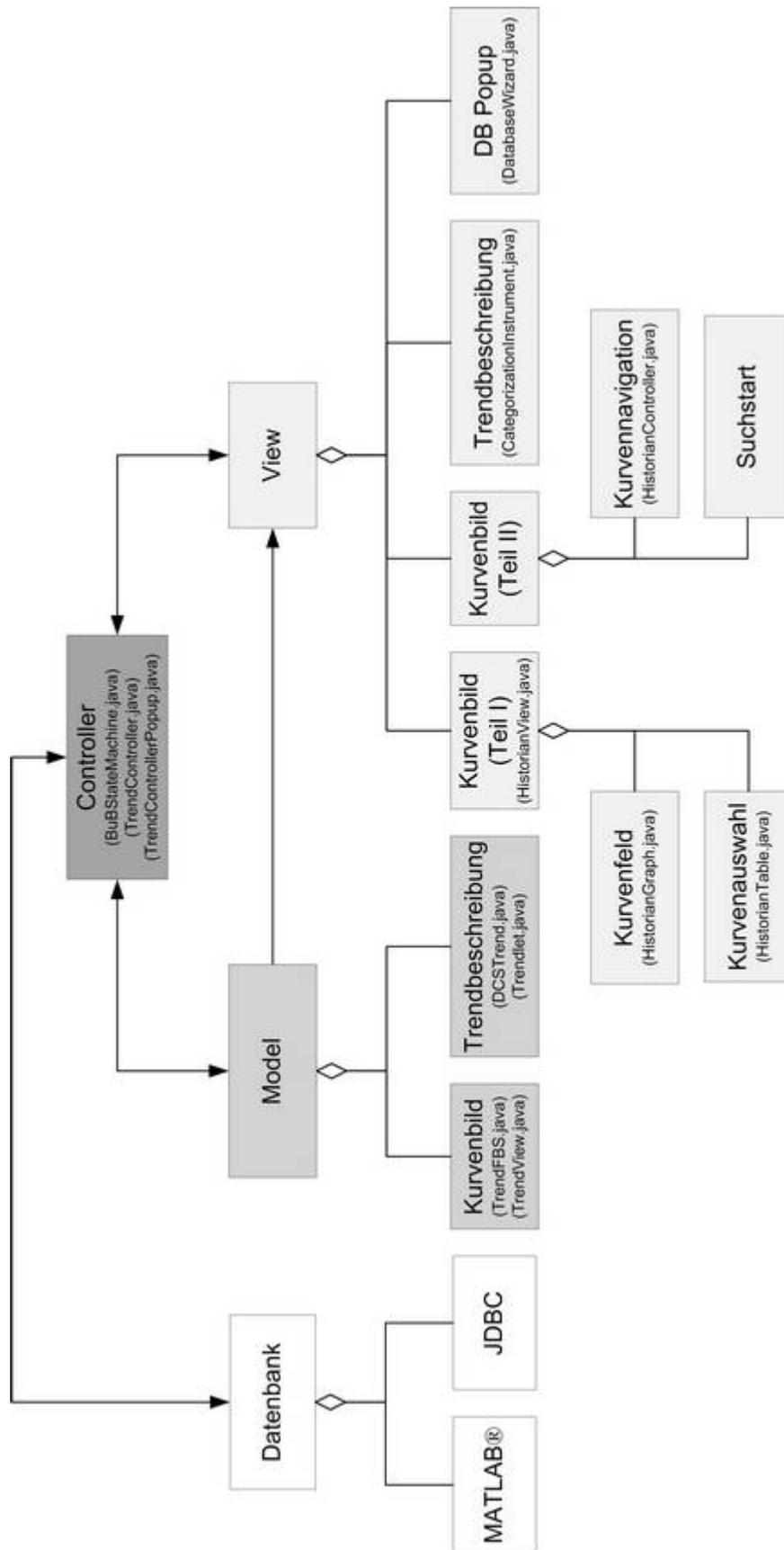


Abbildung 54 Aktionsdiagramm inklusive Auswahl an zugehöriger Java-Klassen

IV Systembewertung

Nach Abschluss des parallel iterativen Systemdesigns lagen ein technisch überprüfbares Subsystem (Suchalgorithmus) sowie ein heuristisch optimierter Nutzerdialog vor. Damit konnte mit der Evaluation des Gesamtsystems anhand von experimentellen Untersuchungen begonnen werden. Notwendig waren darüber hinaus eine klare Aufgabenbeschreibung, eine Auswahl an skalierbaren Testszenarien sowie eine Spezifikation des Nutzerkreises.

Die Ergebnisse sollten Aufschluss geben, inwieweit die effektive und effiziente Arbeitsteilung zwischen Nutzer und Algorithmus mit dem vorliegenden Prototypen möglich ist. Konkret sollten die Nutzertests die Beantwortung folgender Fragestellungen ermöglichen:

1. Wie stark und auf welche Art weichen die Trendbeschreibungen der Versuchspersonen von den entsprechenden normativen Trendbeschreibungen ab? (Ergebnisse in Kapitel IV.3.1)
2. Wie stark wird die Ergebnislösungsgüte von qualitativen und quantitativen Abweichungen von den entsprechenden normativen Trendbeschreibungen beeinflusst? (Ergebnisse in Kapitel IV.3.2)
3. Welchen Einfluss hat die Anzahl an Episoden pro Trendbeschreibung auf die objektive gemessene und subjektiv wahrgenommene Ergebnislösungsgüte? (Ergebnisse in Kapitel IV.3.3)
4. Wie beurteilen die Versuchspersonen die Ähnlichkeitsschätzung der Ergebnisse durch den Suchalgorithmus? (Ergebnisse in Kapitel IV.3.4)

IV.1 Untersuchungsmaterial

Das Untersuchungsmaterial bestand zum einen aus vier verschiedenen Testsets, alle bestehend aus verschiedenen Testszenarien (Kapitel IV.1.1). Zum anderen wurde die Untersuchung mit dem Prototypen durchgeführt, welcher eine bestimmte Parametrisierung erhielt (Kapitel IV.1.2).

IV.1.1 Testsets und Testszenarien

Die Testsets wurden wie auch bereits bei der explorativen Untersuchung des Suchalgorithmus (Kapitel III.1.2.5) anhand artifizierlicher dynamischer Episoden erstellt. Anfängliche Überlegungen, die Testszenarien anhand empirischer Daten einer Laboranlage zu generieren (Barz et al. 2005), wurden aus Gründen der Kontrollierbarkeit verworfen. Die Komplexität der Grunddynamiken aller vier Testsets steigt stetig an. Sie beschreiben typische Verläufe verfahrenstechnischer Größen (Abbildung 55). Die Dynamik aus Testset 2 ähnelt der Antwort eines positiven PT2-Gliedes. Testset 2 bildet zwei aufeinander folgende Sprungantworten von PT1-Gliedern mit Sprunghöhen unterschiedlich starker Amplituden und Vorzeichen ab. Die zwei- und dreifache Kombination entgegengesetzter Antwortverhalten von PT2-Gliedern werden in den Testsets drei und vier nachempfunden.

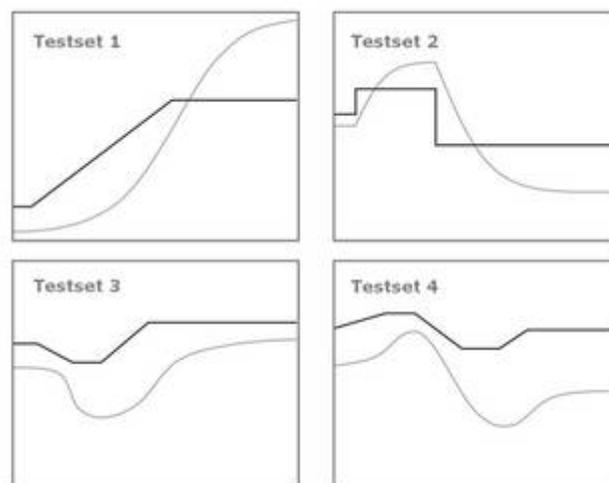


Abbildung 55 Komplexe Testdynamiken
(schwarz: Eingangsgroße; grau: Ausgangsgroße)

Tabelle 23 gibt einen vollständigen Überblick über die Variationen der Testszenarien aller Testsets. Auszüge des genauen Erstellungsquellcodes können im Anhang C.1.1 eingesehen werden. Mit Ausnahme des Testsets 4 wurden alle Testszenarien mit einem konstanten weißen Rauschen der Stärke $X = 0.1$ erstellt (vgl. Kapitel III.1.2.5.1).

Alle Testszenarien eines Testsets weichen in ihren Ausprägungen der Variationskriterien von dem Referenzszenario ab. Auf Basis der Höhe dieser Abweichungen lassen sie sich in einer Rangreihe abbilden, welche die Ähnlichkeit der Testszenarien mit dem Referenzszenario repräsentiert.

Tabelle 23 Überblick über Testsets

Testset	Trendlet-Abfolge	Szenarien-anzahl	Episoden	Kriterium	Wertebereich	
1	Referenz	AEFA	1	EF	(Trendlet, Länge Höhe)	(E, 1.2, 1.4), (F, 1.4, 1.2) ^a
	Test- szenarien	AEFA	24	EF	Länge	(0.6 : 0.2 : 1.4) ^b
				EF	Höhe	(0.6 : 0.2 : 1.4)
		ABA	16	B	Höhe	(1.0 : 0.4 : 2.2)
				B	Länge	(1.0 : 0.4 : 2.2)
2	Referenz	AFDA	1	FD	(Trendlet, Länge Höhe)	(E, 1.0, 0.5), (F, 1.0, 1.0)
	Test- szenarien	AFDA	29	FD	Länge	(0.5 : 0.5 : 1.5)
				FD	Höhe	(0.5 : 0.5 : 1.5)
				FD	Anfangssteigung	(0.2 : 0.2 : 0.8)
				FD	Endsteigung	(0.2 : 0.2 : 0.8)
		AFBCDA	9	BC	Länge	(0.1 : 0.2 : 0.5)
				BC	Höhe	(0.1 : 0.2 : 0.5)
		AEFDGDA	9	E	Länge & Höhe	(0.3 : 0.3 : 0.9)
				GD	Länge & Höhe	(0.3 : 0.3 : 0.9)

^a Für jede der Episoden wird eine Folge aus Trendlet, Episodenlänge und Episodenhöhe definiert.

^b Variation des entsprechenden Episodenkriteriums, wobei die erste Zahl den Startwert darstellt, die zweite die Schrittweite und die dritte den Endwert.

Testset	Trendlet-Abfolge	Szenarien-anzahl	Episoden	Kriterium	Wertebereich	
3	Referenz	AGDEFA	1	GDEF	(Trendlet, Länge Höhe)	(G, 0.5, 1.0), (D, 0.5, 1.0), (E, 1.0, 1.5), (F, 1.0, 1.5)
	Test- szenarien	AGDEFA	8	GDEF	Länge	(0.5 : 0.5 : 1.5)
				GDEF	Höhe	(0.5 : 0.5 : 1.5)
		AGDAEFA	9	A (Mitte)	Länge	(0.5 : 0.5 : 1.0)
		AGBCDBCEBCFA	48	BC (alle)	Länge	(0.2)
					Höhe	(0.1 : 0.1 : 0.9)
		ADEA	AEFGDAEFGDA	DE	Länge & Höhe	(0.1 : 0.1 : 1.5)
	GD (alle)	Länge		(1.0)		
Höhe		(0.5 : 0.5 : 1.5)				
4	Referenz	AFGDEFA	1	FGDEF	(Trendlet, Länge Höhe)	(F, 0.5, 0.5), (G, 0.5, 1.0), (D, 0.5, 1.0), (E, 0.5, 0.5), (F, 0.5, 0.5)
	Test- szenarien	AFGDEFA	23	GD	Höhe	(0.2 : 0.2 : 1.6)
				EF	Höhe	(0.1 : 0.1 : 0.8)
				alle	Rauschfaktor	(0.02:0.08:0.18)

Fehler! Ungültiger Eigenverweis auf Textmarke. zeigt die Herleitung einer solchen Rangreihe exemplarisch für den Testset 1. Je geringer der Rang eines Testszenarios, desto ähnlicher ist es dem Referenzszenario. Das Maß der Ergebnislänge entspricht dem invertierten Rangplatz.

Tabelle 24 normatives Ranking aller Testsznarien am Beispiel von Testset 1

Episode	E	E	F	F	E	F	E&F	Rang	Ergebnis- güte
Abweichung Szenariennr.	Länge	Höhe	Länge	Höhe	gesamt	gesamt	gesamt		
1	60	80	80	60	140	140	280	24	16
2	60	60	60	60	120	120	240	22,5	17,5
3	60	40	40	60	100	100	200	19,5	20,5
4	60	20	20	60	80	80	160	15	25
5	60	0	0	60	60	60	120	10	30
6	40	80	80	40	120	120	240	22,5	17,5
7	40	60	60	40	100	100	200	19,5	20,5
8	40	40	40	40	80	80	160	15	25
9	40	20	20	40	60	60	120	10	30
10	40	0	0	40	40	40	80	5,5	34,5
11	20	80	80	20	100	100	200	19,5	20,5
12	20	60	60	20	80	80	160	15	25
13	20	40	40	20	60	60	120	10	30
14	20	20	20	20	40	40	80	5,5	34,5
15	20	0	0	20	20	20	40	2	38
16	0	80	80	0	80	80	160	15	25
17	0	60	60	0	60	60	120	10	30
18	0	40	40	0	40	40	80	5,5	34,5
19	0	20	20	0	20	20	40	2	38
20	20	80	80	20	100	100	200	19,5	20,5
21	20	60	60	20	80	80	160	15	25
22	20	40	40	20	60	60	120	10	30
23	20	20	20	20	40	40	80	5,5	34,5
24	20	0	0	20	20	20	40	2	38
25...40	linear	-	-	-	-	-	-	32	8

IV.1.2 Parametereinstellungen

Zur Vorverarbeitung der Testsets wurde zugunsten kurzer Antwortzeiten während der Benutzung des Prototypen mit $\zeta = \{5,10,20\}$ und $\Lambda = \{5,10\}$ ein vergleichsweise kleiner Suchraum definiert. Die Parameter der Suchanfrage konnten bzw. sollten von den Versuchspersonen nicht verändert werden. Die Voreinstellung war also für die gesamte Untersuchung bindend.

Dem Linearitätskoeffizient λ_{Ω} (vgl. Kapitel III.1.2.2.2) konnte aufgrund des moderaten Rauschfaktors der Testszenarien ein eher kleiner Wert von fünf gegeben werden. Die Längen- und Höhentoleranz sowie die drei Toleranzen des Episodenübergangs wurden mit Werten von 0.9 sehr großzügig gewählt. Damit wurde sicher gestellt, dass auch im Falle von stark einschränkenden Suchanfragen (sehr kurze oder niedrige Episoden) der Versuchsperson immer ein Angebot an Suchergebnissen vorliegt. Von den Ähnlichkeitskriterien gingen lediglich die Episodenlänge und -höhe in die Berechnung des Ähnlichkeitsindikators ein, und zwar zu gleichen Anteilen. Auf die Verwendung der anderen Ähnlichkeitsindikatoren wurde in dieser Untersuchung verzichtet, da es einem Nutzer mit dem vorliegenden Nutzerdialog nicht möglich ist, die Effekte dieser Indikatoren visuell nachzuvollziehen. Entsprechend des Prinzips von Vereinfachung und Generalisierung erhielten alle Sensitivitätsparameter einen Wert von 0.3 (vgl. Abbildung 27 auf Seite 72). Da pro Testszenario nur das jeweils beste Ergebnis in die Menge der Gesamtergebnisse aufgenommen werden sollten, konnte die Redundanztoleranz r_{tol} mit dem Wert Null ausgewiesen werden.

Tabelle 25 Parametereinstellung des Suchalgorithmus für Nutzertests

Parameter	λ	θw_e	ϕw_e	θh_e	G_w	G_h	$G_{\theta w}$	$G_{\phi w}$	$G_{\theta h}$	G_v
Wert	5	0.9	0.9	0.9	1	1	0	0	0	0
Parameter		w_{ε_e}	h_{ε_e}	r_{tol}	S_w	S_h	$S_{\theta w}$	$S_{\phi w}$	$S_{\theta h}$	S_v
Wert		0.9	0.9	0	0.3	0.3	0.3	0.3	0.3	0.3

IV.2 Durchführung

Die Stichprobe von 21 Versuchspersonen bestand zu etwa je einer Hälfte aus Studenten bzw. Absolventen der Informatik sowie Ingenieur- und Naturwissenschaften. Diese Auswahl wurde gezielt angestrebt, da die Population der potentiellen Endnutzer zum überwiegenden Teil aus diesen Disziplinen besteht. In diesem Sinne wurde auch darauf geachtet, dass die Stichprobe nur männliche Versuchspersonen beinhaltet. Obwohl der Prozentsatz an Frauen in der Prozessindustrie stetig steigt, sind weibliche Betriebsleiter, Schichtmeister bzw. Verfahrenstechniker immer noch eine Seltenheit. Die Vorkenntnisse bezüglich der Arbeit mit Kurvendiagrammen traten in der Stichprobe annähernd normal verteilt auf.

Die Testsets wurden in Einzelsitzungen von je eineinhalb Stunden Bearbeitungszeit durchgeführt. Zu Beginn erhielten die Versuchspersonen eine allgemeine Erläuterung zu den Zielen der experimentellen Untersuchung. Danach wurden ihnen die Zielstellung des Prototypen sowie dessen grobe Arbeitsweise anhand der „Google Analogie“ erläutert (vgl. Anhang C.2.1). In diesem Zusammenhang wurde zudem der Begriff „Trend“ definiert und mit Hilfe von Graphiken dargestellt (vgl. Anhang C.2.2). Dabei wurden alle relevanten Beschreibungskriterien (Länge, Höhe, Dynamik, Toleranzen) erklärt, sowie eine grobe Erläuterung der Suchindexe gegeben (vgl. Anhang C.2.3).

In einem zweiten Schritt wurde die Arbeitsweise mit dem Prototypen erläutert. Die Versuchspersonen wurden dafür aufgefordert, eine Übungssequenz unter Anleitung durchzuführen. Inhalte der Übung war die Kurvennavigation, die Trendbeschreibung sowie die Auswahl von Ergebniskurven (vgl. Anhang C.2.4). Nach erfolgreicher Interaktion der Versuchsperson mit dem Prototypen wurden weitere Fragen der Versuchsperson beantwortet.

Die Aufgabeninstruktion erhielten die Versuchspersonen in Textform (vgl. Anhang C.2.5). Darin wurden sie aufgefordert, die vier verschiedenen in Kapitel IV.1.1 vorgestellten Testsets zu bearbeiten. Bei jedem Testset starteten sie zunächst damit, einen Trend ihrer Wahl für die Referenzkurve zu beschreiben, welcher nur eine Episode enthielt. Danach steigerte sich die Episodenanzahl pro Trend sukzessive um eine Episode. Die Maximalanzahl variierte von zwei bis

vier Episoden. Die Versuchspersonen wurden weiterhin aufgefordert, die Rangfolge der Suchergebnisse nach jedem Durchgang zu überprüfen. Ab dem zweiten Durchgang jedes Testsets sollten sie zudem einschätzen, ob die neuen Suchergebnisse besser, gleich gut oder schlechter als die Ergebnisse der vorangegangenen ausfallen.

Es gab keinerlei Zeitbegrenzung für die Bearbeitung der einzelnen Testsets. Nach Ablauf der eineinhalb Stunden Versuchsdauer wurde die Untersuchungssitzung jedoch mit dem augenblicklich bearbeiteten Testset beendet. Da die Instruktion ca. 30 Minuten in Anspruch nahm, verblieben ca. 60 Minuten für die Bearbeitung der Testsets.

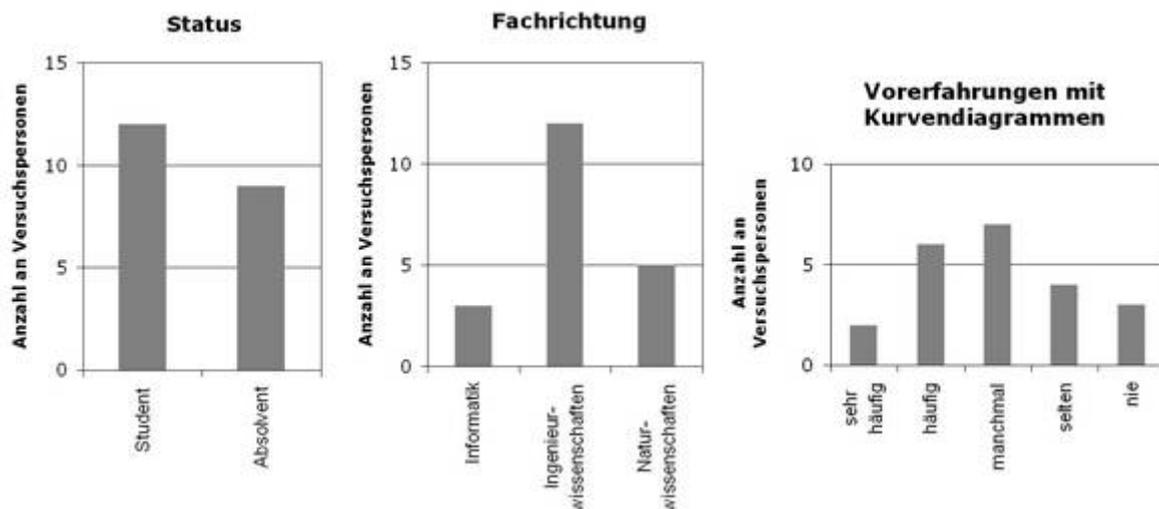


Abbildung 56 Stichprobenbeschreibung

IV.3 Ergebnisse

Während die verfügbare Zeit zur Bearbeitung aller Einzeltests durch das Durchführungsdesign begrenzt war, wurden den Versuchspersonen zur Bearbeitung der einzelnen Testsets keinerlei zeitlichen Vorgaben gemacht. Dies führte dazu, dass nicht alle Versuchspersonen den kompletten Satz an Testsets bearbeiteten. Tabelle 26 zeigt die Anzahlen an Versuchspersonen, welche die Bearbeitung der vier Einzeltests mit wenigstens einer Trendbeschreibung erfolgreich absolviert haben. Da einige Versuchspersonen nicht alle Trendbeschreibungen eines Testsets erfolgreich

absolviert haben, verringerte sich die tatsächliche Anzahl an Trendbeschreibungen je Testset nochmals um bis zu 15%.

Tabelle 26 Durchführungstatistik

Testset	Anzahl VP	maximale Anzahl an Trendbeschreibungen	tatsächliche Anzahl an Trendbeschreibungen
1	21	42	41
2	19	38	38
3	16	64	56
4	12	48	41

Auf Basis dieser Datenlage werden nun die Ergebnisse zur Beantwortung der eingangs dieses Kapitels gestellten Evaluationsfragen präsentiert.

IV.3.1 Abweichungen von normativer Trendbeschreibung

Wie bereits in den Kapiteln III.1.2.5 und IV.1.1 erläutert, liegt für jedes Referenzszenario der einzelnen Testsets ein normatives Modell der Trendbeschreibung vor. Dieses basiert auf der Erstellung der entsprechenden Zeitreihen anhand von parametrisierbaren Episodenfolgen. Abbildung 57 zeigt die Häufigkeiten normkonformer Trendbeschreibungen durch die Versuchspersonen. Da die absolute Anzahl der Trendbeschreibungen für jeden Versuch eines jeden Testsets variiert, wurden diese Häufigkeiten in ein prozentuales Verhältnis gesetzt. Eine Trendbeschreibung ist normkonform, wenn die einzelnen Episoden anhand der normativ korrekten Trendlets spezifiziert wurden.

Hatten die Versuchspersonen zur Trendbeschreibung aus Testset 1 nur eine Episode zur Verfügung, nutzten über die Hälfte dafür das lineare Trendlet \mathcal{L} . Erst bei der Trendbeschreibung anhand von zwei Episoden wurde von dem überwiegenden Anteil der Versuchspersonen die normkonforme Trendlet-Abfolge \mathcal{EF} verwendet. Bei Testset 2 startete die Hälfte aller Versuchspersonen mit der

normkonformen Trendbeschreibung anhand des Trendlets \mathcal{D} . Den normkonformen Trend mit der Trendlet-Abfolge \mathcal{FD} nutzten im folgenden Schritt jedoch nur zwei dieser zehn Versuchspersonen. Die meisten nicht normkonformen Trendbeschreibungen enthielten die Abfolgen \mathcal{DA} oder \mathcal{GD} . Während erstere Abfolge ein mangelhaftes Verständnis der Aufgabenstellung nahelegt^a, lässt sich die zweite Abfolge durch ein fehlendes Konzept von gebrochenen Übergängen zwischen Episoden erklären.

Die sehr niedrige Anzahl an normkonformen Trendbeschreibungen mit einer Episode aus Testset 4 lässt sich analog zum Testset 1 erklären. Der überwiegende Anteil der Versuchspersonen nutzte an dieser Stelle das lineare Trendlet \mathcal{C} . In beiden Fällen beschreiben die Versuchspersonen den stark linear anmutenden glatten Übergang zwischen zwei gekrümmten Episoden.

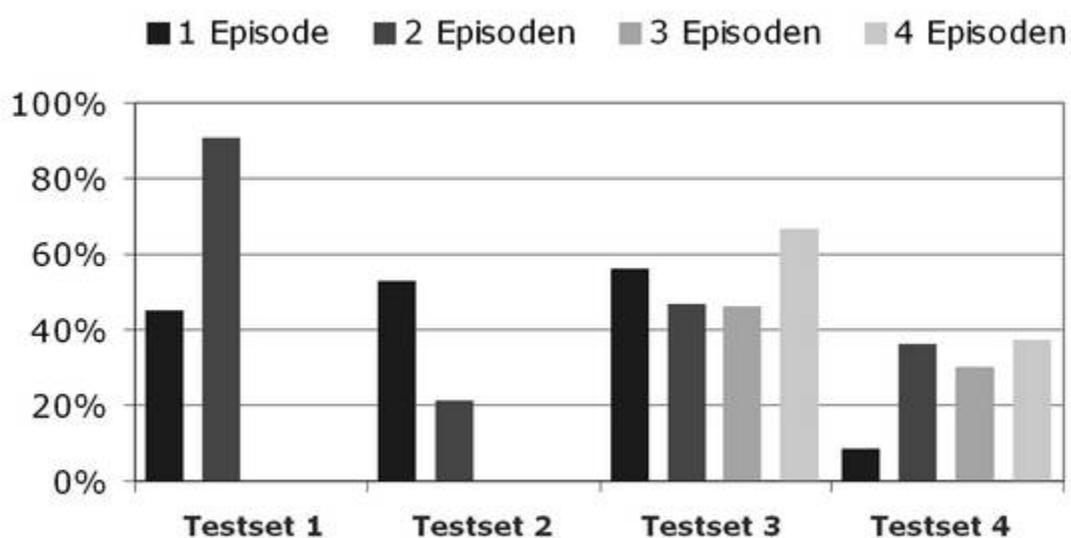


Abbildung 57 Prozentuale Häufigkeitsanteile normkonformer Trendbeschreibungen

Darüber hinaus zeigen sich keine Auffälligkeiten in den Trendbeschreibungen der Testsets drei und vier. Etwa die Hälfte aller Versuchspersonen nutzen im Testset drei die normativen Trendlet-Abfolgen, unabhängig davon wie viele Episoden der Trend enthält. Das dies im Testset 4 nur noch

^a Der bedeutsame Anstieg zu Beginn eines Testszenarios wird zugunsten einer stationären Episode vernachlässigt, welche besser als Abschluss des korrekt erkannten Abfalls interpretiert werden könnte. Ein analoger Fehler bei der Benutzung einer textbasierten Suchmaschine wäre die Verwendung der Schlagworte „Haus“ & „Hauseingang“ anstatt der Schlagworte „Hauseingang“ und „Treppe“.

ca. ein Drittel vermögen, liegt nicht zuletzt an der gesteigerten Komplexität des zugrunde liegenden Testsets^a.

Zur anschließenden Betrachtung quantitativer Normabweichungen wird der Datensatz aus Tabelle 26 auf die qualitativ normkonformen Trendbeschreibungen reduziert (Testset 1, N=28; Testset 2, N=14; Testset 3, N=30; Testset 4, N=11). Die Episoden all dieser Trendbeschreibungen weichen in ihren Längen und Höhen vom normativen Modell ab. Abbildung 58 bildet das durchschnittliche Ausmaß dieser Abweichungen ab. Dabei werden die jeweiligen absoluten Abweichungen an den absoluten Längen und Höhen des normativen Referenzrends relativiert.

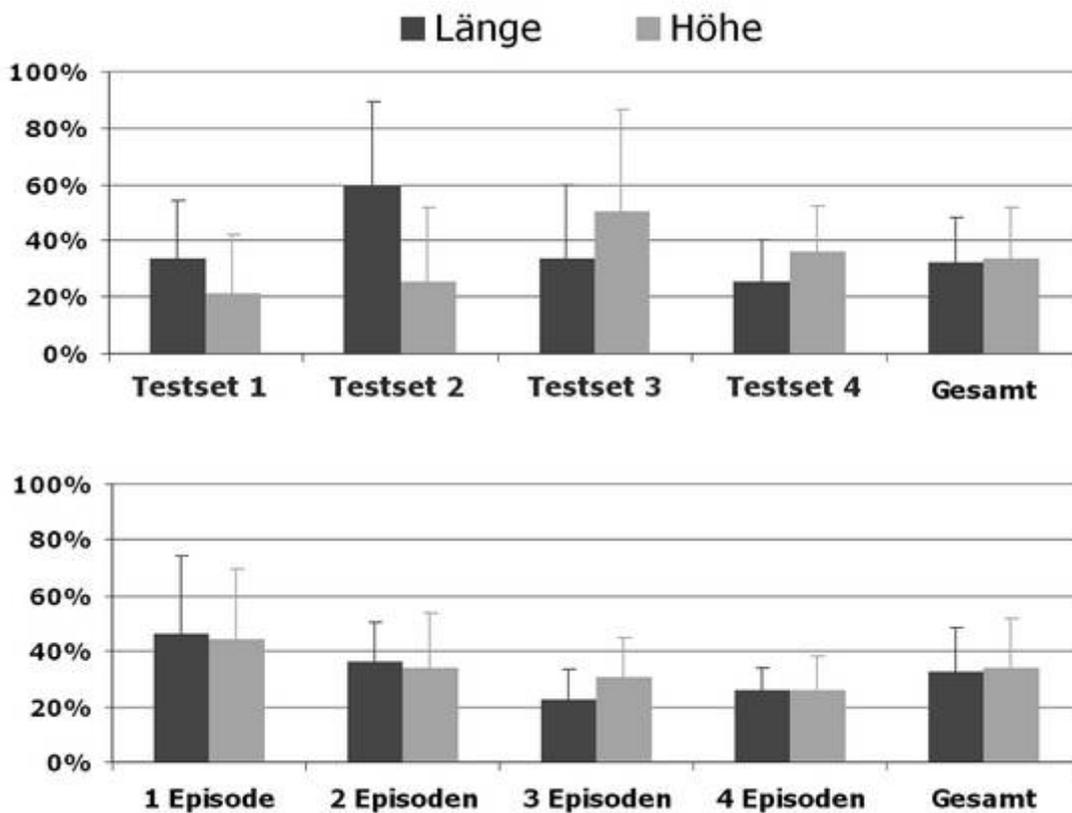


Abbildung 58 Prozentuale, gemittelte Abweichungen (Episodenlänge und –höhe) der Trendbeschreibungen aller Versuchspersonen von der normativen Trendbeschreibung

^a Dies besteht im Testset 4 aus 5+2 normativen Episoden, im Gegensatz zu den 4+2 Episoden aus Testset 3.

Über alle Testsets lag die mittlere prozentuale Abweichung bei 32% (Länge) und 34% (Höhe). Im Testset zwei wurden jedoch Längenabweichungen um die 60% beobachtet. Dies erklärt sich aus der Verwendung des Trendlets D. Ein Großteil der Versuchspersonen macht hier einen ähnlichen Fehler wie diejenigen, welche die Trendlet-Abfolge \mathcal{DA} genutzt hat. Sie weiten die Trendbeschreibung in den stationären Bereich hinter dem normativen \mathcal{D} aus und integrieren diesen Zeitbereich in das von ihnen verwandte \mathcal{D} . Die verstärkten Höhenabweichungen aus Testset 3 erklären sich aus der Tatsache, dass für viele der mit den Trendlets D und E spezifizierten Episoden von den Versuchspersonen ein gebrochener Übergang zu einem stationären Vor- bzw. Nachlauf angenommen wurde. Die Höhenabweichungen verringerten sich bei der Ausweitung des Trends auf mehr als zwei Episoden^a.

Betrachtet man die Entwicklung der Längen- und Höhenabweichungen in Abhängigkeit von der Episodenanzahl, lässt sich ein leicht negativer Zusammenhang ausmachen. Dies ist direkt auf den Effekt zurückzuführen, dass die Längen- und Höhenabweichungen zwischen den einzelnen Episoden immer geringer ausfallen, als bei den Start- und Endepisoden eines Trends. Je mehr Zwischenepisoden ein Trend aufweist, desto kleiner wird die Gesamtabweichung.

IV.3.2 Ergebnsgüte in Abhängigkeit von der Trendbeschreibung

Im Kapitel IV.1.1 wurde die Berechnung der Ergebnsgüte für alle Testszenarien eines Testsets beschrieben. Zur Berechnung der Güte eines Suchergebnisses werden die Werte der besten sechs Testszenarien aufsummiert. Die maximal erreichbare Ergebnsgüte wird durch die Suchanfrage mit einer normativen Trendbeschreibung definiert. Die Ergebnisse aller empirischen Suchanfragen werden an diesen Absolutwerten relativiert. Abbildung 59 zeigt die Mittelwerte der relativierten bzw. prozentualen Ergebnsgüten aller Suchanfragen. Über alle Testsets wird eine Ergebnsgüte registriert, welche bei den Suchanfragen mit normkonformen Trendbeschreibungen ca. 15% höher liegt als bei Suchanfragen mit normkonträren Trendbeschreibungen. Dieser Unterschied lässt sich auch in fast allen Einzelbetrachtungen wieder finden. Besonders stark scheint der Effekt bei Trends

^a Entspricht einem erweiterten Trend mit der Trendlet-Abfolge ADEA. Diese wird später zugunsten der Abfolge GDEF verworfen. Aufgrund des dabei entstehenden glatten Übergangs halbieren sich die Episodenhöhen.

mit mehr als zwei Episoden ausgeprägt zu sein. Betrachtet man die einzelnen Testsets isoliert, fällt lediglich Testset 2 aus diesem Muster heraus.

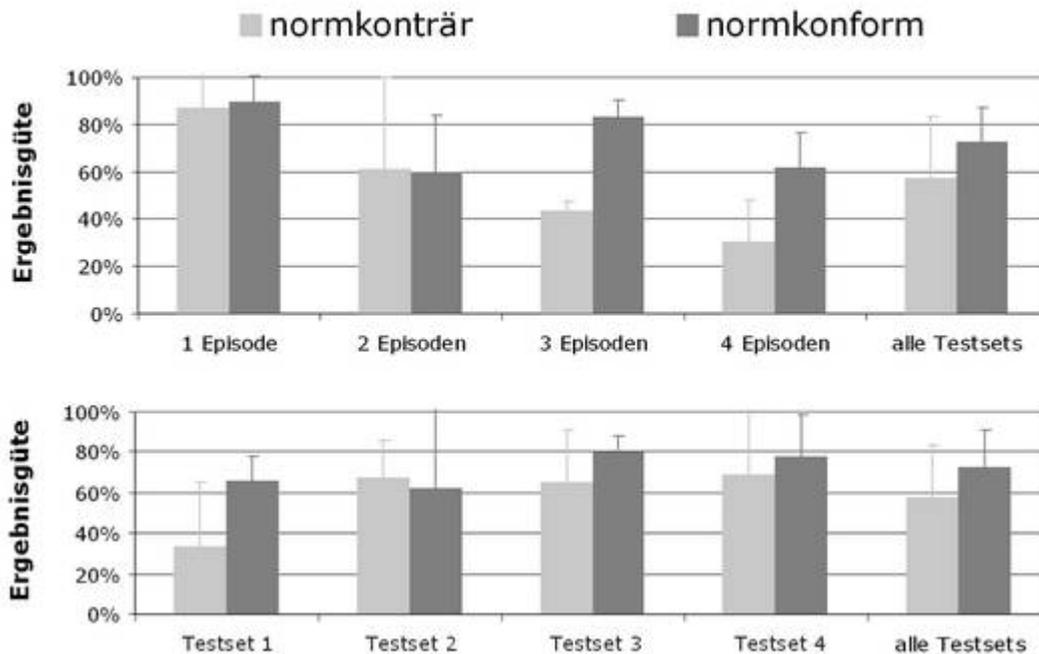


Abbildung 59 Abhängigkeit der Ergebnisgüte von der Qualität der Trendbeschreibung (normkonträr vs. normkonform) über alle Testsets und Episodenanzahlen

Neben der qualitativen Dimension der Trendlet-Auswahl hat auch die quantitative Ausprägung der Episodenlängen und –höhen Einfluss auf die Ergebnisgüte. Abbildung 60 zeigt ausschließlich negative Zusammenhänge zwischen der Höhe der quantitativen Abweichungen dieser Ähnlichkeitskriterien und der Ergebnisgüte. Der Einfluss des Höhenkriteriums scheint dabei mit einem über alle Testsets gemittelten Korrelationskoeffizienten von $r = -0.43$ fast doppelt so stark wie der des Längenkriteriums ($r = -0.24$) zu sein.

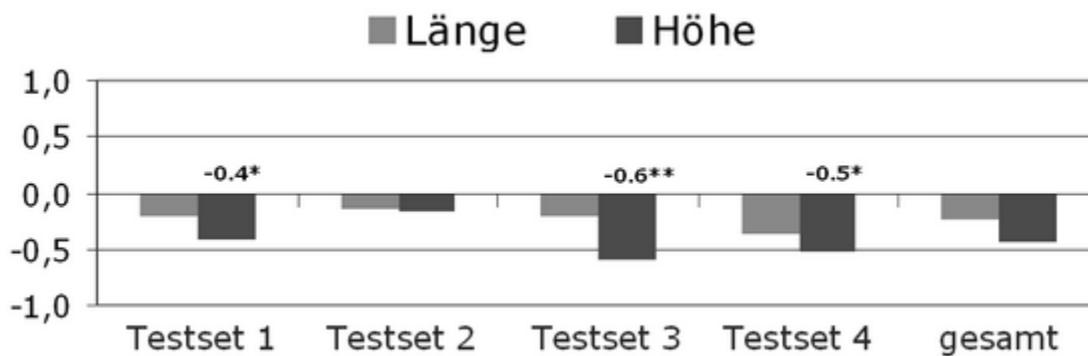


Abbildung 60 Korrelativer Zusammenhang (Pearson) zwischen Ergebnislänge und Höhe der quantitativen Normabweichungen

IV.3.3 Einfluss der Episodenanzahl auf die Ergebnisgüte

Die Versuchspersonen wurden pro Testdurchlauf nach jedem Ergebnis gefragt, ob sie dieses besser (Wert = 1), gleich gut (Wert = 0) oder schlechter (Wert = -1) als die vorangegangenen Ergebnisse einschätzen^a. Die Urteile aller Testpersonen aus allen vier Testsets lassen sich nun zu gemeinsamen Mittelwerten zusammen führen. Abbildung 61 macht deutlich, dass sich aus Sicht der Versuchspersonen die Ergebnisse mit zunehmender Episodenanzahl der Trendbeschreibung stetig verbessern. Dies steht jedoch im eindeutigen Widerspruch zu den objektiven Daten der Ergebnisgüte (Abbildung 62). Wie sich auch schon in Abbildung 59 andeutet, sinkt die Ergebnisgüte aller Trendbeschreibungen (normkonform & normkonträr) mit der steigenden Anzahl an genutzten Episoden.

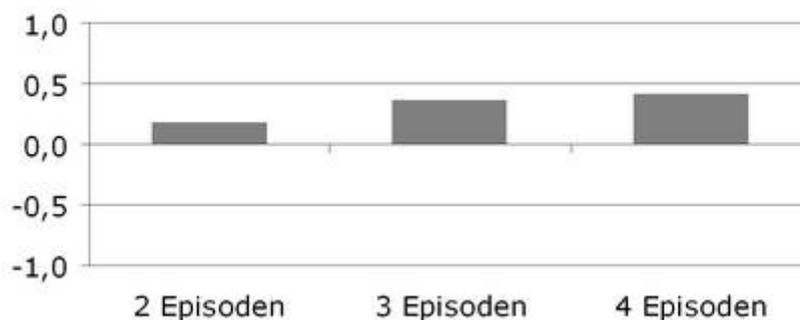


Abbildung 61 Evolution der subjektiven Ergebnisgüte mit steigender Anzahl an Episoden pro Trendbeschreibung

^a Für die Testsets 1 & 2 erhält man auf diese Weise ein Urteil pro Versuchsperson und Test. Für die Testsets 3 und 4 waren dies aufgrund der erhöhten Episodenanzahl jeweils sechs Urteile.

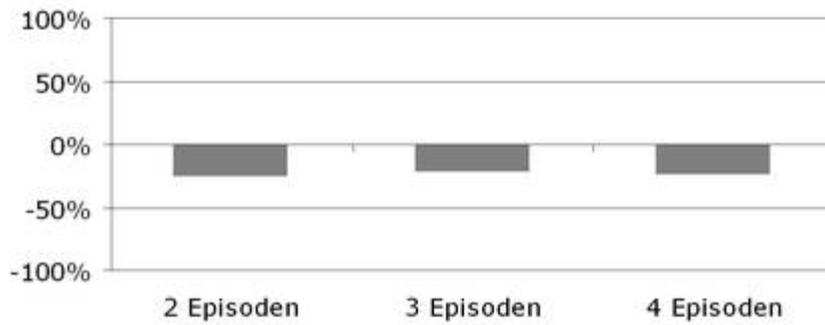


Abbildung 62 Evolution der objektiven Ergebnisgüte mit steigender Anzahl an Episoden pro Trendbeschreibung

IV.3.4 Nutzerevaluation der Ähnlichkeitsschätzung

Die besten sechs Ergebnisse einer Suchanfrage wurden den Versuchspersonen von der Suchmaschine in einer Rangreihe präsentiert. Diese wurde nach Ermessen der Versuchspersonen korrigiert. Die Übereinstimmung einer empirischen^a und korrigierten Rangreihe mit $N \leq 6$ kann anhand des Korrelationskoeffizienten τ von Kendall (nach Bortz & Lienert, 2003) bestimmt werden. Der mittlere Korrelationskoeffizient aller Ergebnisrangreihen (alle Versuchspersonen & alle Testsets) fällt mit $\tau = 0.32$ eher klein aus. Abbildung 63 verdeutlicht zudem, dass sich die Übereinstimmung der empirischen und korrigierten Rangreihe zwar in allen Testsets ähneln, die Übereinstimmung innerhalb der Testsets jedoch stark variiert.

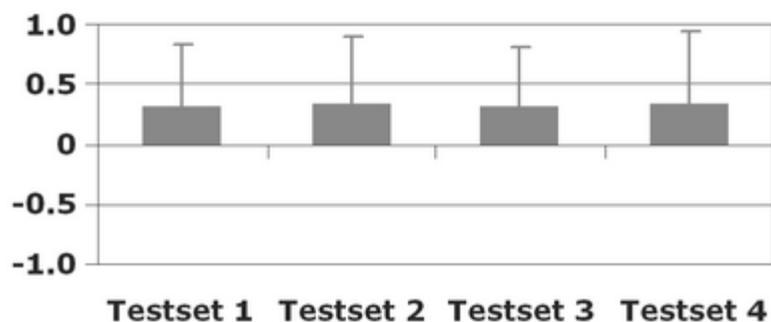


Abbildung 63 Mittelwerte aller Rangkorrelationen zwischen empirischer und korrigierter Rangreihe über alle Testsets

^a „Empirisch“ meint hier „von der Suchmaschine vorgeschlagen“.

IV.4 Zusammenfassung und Diskussion

Annähernd die Hälfte (46%) der Versuchspersonen nutzen für ihre Trendbeschreibung normkonforme Trendlets, obwohl sie keinerlei Übung mit dieser Art von Suchmaschine hatten. Dies lässt auf einen geringen Trainings- und Übungsaufwand bei der Einführung einer derartigen Suchmaschine schließen. Allerdings haben nur 70% der Versuchspersonen alle Testsets innerhalb der verfügbaren Zeit erfolgreich bearbeitet. Je nach Dauer der Instruktionsphase hatten die Versuchspersonen 50 bis 70 Minuten Zeit für die komplette Aufgabenbearbeitung. Bei insgesamt zwölf geforderten Trendbeschreibungen waren also ca. fünf Minuten für die Trendbeschreibung und die Ergebnisevaluation verfügbar. Auf eine detaillierte Dokumentation der benötigten Zeiteinheiten für die einzelnen Aufgabenschritte wurde verzichtet. Beobachtungen des Versuchsleiters haben jedoch ergeben, dass ein Großteil der Zeit auf die Evaluation der Ergebnisse verwendet wurde. Bei allen Aufgaben wurde von den Versuchspersonen zudem die Genauigkeit der Geschwindigkeit vorgezogen.

Die quantitativen Abweichungen der qualitativ normkonformen Trendbeschreibungen liegen bei ca. einem Drittel der normativen Episodenlängen und -höhen. Diese Abweichungen erklären sich nicht allein aus dem Rauschanteil der Referenzzeitreihe. Größtenteils sind sie auf systematische Fehler in dem Umgang mit dem Trendlet-Alphabet zurück zu führen. Insbesondere in der Start- und Endphase eines Trends kommt es zu Überschätzungen der Abstände bedeutsamer Extrem- und Wendestellen gekrümmter Episoden. Hier ist ein eindeutiger Bedarf an Nutzertraining identifizierbar.

Niedrige Werte in der Ergebnislänge lassen sich direkt auf qualitative und quantitative Abweichungen von der normativen Trendbeschreibung zurückführen. Das bestätigt die erhoffte, wenn auch nicht selbstverständliche Heuristik: „Je trainierter die Nutzer, desto besser die Ergebnisse“. Allerdings fallen die Effekte relativ gering aus. Die Nutzung qualitativ normkonformer Trendbeschreibungen bewirkt lediglich eine Verbesserung der Ergebnislänge um 15%. Die Zusammenhänge

zwischen den Normabweichungen der Episodenlängen und –höhen und der Ergebnsgüte werden nur in drei von vier Testsets für das Ähnlichkeitskriterium „Episodenhöhe“ statistisch signifikant.

Die subjektiv wahrgenommene Ergebnsgüte steigt mit dem Detaillierungsgrad der Trendbeschreibung im Sinne der Episodenanzahl. Gleichzeitig ist jedoch eine Verschlechterung der objektiven bzw. berechneten Ergebnsgüte zu verzeichnen. Dies erklärt sich zum Großteil aus der Tatsache, dass die Anzahl der Ergebnisse mit zunehmender Spezifizierung eines Trends^a häufig unter die maximal betrachtete Anzahl von sechs fällt. In diesen Fällen verringert sich automatisch die Ergebnsgüte des Suchergebnisses.

Tabelle 27 Zusammenfassung der Ergebnisse

Fragestellung	Ergebnis
1	46% aller Trendbeschreibungen sind normkonform. Die Abweichungen der Episodenlängen und –höhen aller normkonformen Trendbeschreibungen liegen im Durchschnitt bei ca. einem Drittel der normativen Episodenlängen und –höhen.
2	Aus den normkonformen Trendbeschreibungen resultieren zu 15% bessere Ergebnisse als aus den normkonträren Trendbeschreibungen. Es lässt sich ein moderater Zusammenhang zwischen der Abweichung von Episodenlängen und -höhen normkonformer Trends und der Ergebnsgüte nachweisen.
3	Während die objektive Ergebnsgüte mit zunehmender Anzahl an Episoden pro Trendbeschreibung abnimmt, steigt die subjektiv wahrgenommene Ergebnsgüte stetig.
4	Die Ähnlichkeitsschätzungen von Suchalgorithmus und Versuchspersonen stimmen moderat miteinander überein.

^a Im Sinne einer erhöhten Anzahl an Episoden.

Die subjektiv wahrgenommene Güte der automatischen Ähnlichkeitsschätzungen liegt bei 30%. Dieses Ergebnis ist nicht befriedigend. Allerdings muss man beachten, dass die Versuchspersonen im Novizenmodus mit voreingestellten Gewichten der Ähnlichkeitstoleranzen gearbeitet haben. Experimente im Expertenmodus lassen bessere Ergebnisse erwarten. Weiterhin steht zur Diskussion, nach welchen Kriterien die Versuchspersonen die Rangreihenkorrektur vorgenommen haben. Weichen diese Kriterien von den Kriterien des Suchalgorithmus (Episodenlänge & Episodenhöhe), wäre dies eine mögliche Erklärung.

V Systemkorrektur - Ausblick

Für die Entwicklung einer Trend basierten Datenbanksuchmaschine wurden Konzepte zur Datenindexierung, für einen Such- und Vergleichsalgorithmus und für einen Nutzerdialog spezifiziert und prototypisch implementiert. Diese Implementierung ermöglichte die Durchführung von experimentellen Nutzertests. Zugleich eignet sich der Prototyp zu unmittelbaren Präsentationszwecken, wie bereits auf international besuchten Messen und auf einigen Tagungen feststellbar war. Darüber hinaus ist der Prototyp mit geringem Einrichtungsaufwand als Standalone-Anwendung nutzbar. Die Gebrauchstauglichkeit der Trend basierten Suchmaschine konnte auf Basis des Prototypen bereits zu einer frühen Phase der Produktentwicklung überprüft und optimiert werden. Im Folgenden sollen einige Empfehlungen zur Verbesserung, Optimierung bzw. Erweiterung von Suchalgorithmus und Nutzerdialog gegeben werden.

Eine Optimierung des vorgestellten Suchalgorithmus (Entwurf II) ist primär zum Zwecke der Recheneffizienz zur Verkürzung der Antwortzeiten zu verfolgen. Mit dem vorliegenden Prototyp wurde die grundsätzliche Funktionsweise des Suchalgorithmus entworfen sowie dessen Kompatibilität mit dem Nutzerdialog erreicht. Eine Steigerung der Recheneffizienz ist aufgrund des modularen Aufbaus des Suchalgorithmus möglich und stellt den nächsten Schritt für die Entwicklung vom Prototypen hin zur Anwendung dar. Beispielsweise ist die Effizienz bei einer hohen Anzahl an Faltungsskalen steigerungsfähig. Je feiner die Auflösung im Skalenraum, desto redundanter die dort abgelegte Information. Bisher erfolgt eine Identifikation potentieller Ergebnistrends unabhängig voneinander. Eine Reduktion der Redundanz erfolgt erst zum Ende der Vergleichsprozedur. Vergleichbare Fragestellungen der Optimierung können für Diplom- und Doktorarbeiten mit einem Schwerpunkt in den Disziplinen der Informatik bzw. Mathematik von wissenschaftlichem Interesse sein.

Beim Nutzerdialog birgt eine Verbesserung der graphischen Visualisierung zur Trendbeschreibung am meisten Optimierungspotential. Hierbei ist vor allem die Maus gesteuerte Manipulation der

Episodencharakteristika Krümmung, Länge und Höhe zu nennen. Da sich damit allerdings weiter von den Richtlinien der VDI/VDE 3699-4 (1997) entfernt wird, ist eine solche Veränderung des Nutzerdialogs nur unter Berücksichtigung des Nutzungskontextes zu realisieren. Des Weiteren ist die Kombination unterschiedlicher Trendbeschreibungen verschiedener Variablen ein oft genannter Wunsch von potentiellen Endnutzern. Um eine Suche von Trends unabhängig vom Variablentyp zu ermöglichen, sollten relative Episodenhöhen eingeführt werden. Die genannten Vorschläge zur Erweiterung und Verbesserung des Nutzerdialogs erscheinen vor allem für Diplomarbeiten der Studienfächer Interaktionsdesign, Psychologie oder Arbeitswissenschaften interessant.

Darüber hinaus ist eine zudem Integration des bestehenden Prototypen in reale Experimental- und/oder Prozessleitsystem denkbar. Zum einen ist dabei die inhaltliche Einbindung in ein Experimentalleitsystem zu nennen, welches vom Herrn Prof. Urbas und seinen Mitarbeitern in den letzten Jahren entwickelt wurde, und welches nun an seinem Lehrstuhl der Technischen Universität Dresden die Beantwortung weiterer Grundlagenwissenschaftlicher Fragen ermöglicht. Zum anderen soll die Nutzung der Suchmaschine als Standalone-Version durch vereinzelt Anwender, wie z.B. die Ingenieure der BASF in Ludwigshafen unterstützt werden. Eine dritte Zielstellung ist die Integration des Konzepts in eine bestehende Prozessüberwachungs- und analysesoftware, wie sie von den Herstellern der Prozessleitsysteme angeboten werden (vgl. Tabelle 5 auf Seite 44).

Allgemeiner betrachtet lässt sich die potentielle Anwendung der Trend basierten Datenbanksuchmaschine durch folgende Szenarien prognostizieren. Erhält man Kurvenverläufe aus der Vergangenheit, lassen sich diese nicht nur hinsichtlich deren Ursachen analysieren (vgl. Abbildung 16 und Abbildung 17 auf Seite 47 u. f.), sondern auch hinsichtlich des weiteren Verlaufs, der projiziert auf das augenblickliche Referenzszenario dessen Zukunft abbildet. Zudem lassen sich auf Basis der Ähnlichkeitsschätzungen Cluster an Ereignissen bilden, welche dann mit weiteren statistischen Verfahren in Relation zu anderen Anlagenparametern wie Produktqualitäten, Equipment-Ausfällen u. ä. setzen lassen (vgl. Kapitel II.1.2). Von den interviewten Prozessexperten (Kapitel II.2.5) wurde auch häufig der Wunsch genannt, anhand derartiger Trendbeschreibungen automatische Fehlererkennungssysteme zu programmieren. Von diesen wurde auch die Anwendung der Suchma-

schine bei der Analyse von Daten aus Prozesssimulationen und bei dem Vergleich von Batchprozessen vorgeschlagen.

Neben der Anwendung im in dieser Arbeit fokussierten Domäne der Prozessüberwachung und –analyse ist auch eine Anwendung in anderen wissenschaftlichen und wirtschaftlichen Feldern denkbar. So nimmt im Rahmen der Klimaforschung die Analyse und Prognose von Verlaufsmustern immer größere Bedeutung ein. Auch im medizinischen Anwendungsbereich finden sich viele Beispielszenarien, in denen die Interpretation und Analyse von Kurvenmustern von Bedeutung sind (Rittner & Döring, 2007), und eine nachhaltige und breit zugängliche Nutzung von diesen Daten verfolgt wird.

Zum Abschluss des Ausblicks soll die Generalisierbarkeit der Entwicklungsmethodik diskutiert werden. Die Umsetzung des Parallel Iterativen Entwicklungsprozesses (Timpe & Kolrep, 2000) hat sich in der prospektiven Gestaltung einer Trend basierten Datenbanksuchmaschine bewährt. Das Resultat ist ein ausgewogener Prototyp, der sowohl die technischen Konzepte des Suchalgorithmus als auch die Gebrauchstauglichkeit des entworfenen Nutzerdialogs überprüfbar gemacht hat. Beide Entwicklungspfade haben von der Durchführung in Form von häufigen und regelmäßigen Iterationen profitiert. Die parallele Bearbeitung ermöglichte zudem eine kontinuierliche Anpassung der aktuellen Konzepte. Damit konnte auf ein aufwendiges und ineffizientes Re-Design verzichtet werden. Innerhalb der Systemevaluation ist die Durchführung der Expertenevaluation in Form von Kleingruppenarbeit hervorzuheben. Darüber hinaus konnte anhand der experimentellen Nutzertests eine ganzheitliche Evaluation des Gesamtsystems erreicht werden. Somit kann die in dieser Arbeit angewandte Vorgehensweise für die erfolgreiche Konzeption und Implementierung vergleichbarer Mensch-Technik-Interaktion empfohlen werden.

Literaturverzeichnis

- Abdi, H. (2007). Signal detection theory (SDT). In: N. Salkind (Ed.), Encyclopedia of measurement and statistics (pp. 886-889). Thousand Oaks: Sage.
- Agrawal, R., Faloutsos, C. & Swami, A. (1993). Efficient similarity search in sequence databases. In D. B. Lomet (Ed.), Proceedings of the 4th Int'l Conference on Foundations of Data Organization and Algorithms in Chicago, 1993 (pp. 69-84). New York: Springer.
- Alsmeyer, F. (2005, July). Trend-based treatment of process data: Application to practical problems. Paper presented at the 7th World Congress of Chemical Engineering, Glasgow, Scotland.
- Bakshi, B. R., Locher, G., Stephanopoulos, G. & Stephanopoulos, G. (1994). Analysis of operating data for evaluation, diagnosis and control of batch operations. Journal of Process Control, 4 (4), 175-194.
- Barz, T., Frey O., Huss J. & Urbas L. (2006, August). Mining process data using user defined curve patterns. Paper presented at the 17th International Congress of Chemical and Process Engineering CHISA, Prague.
- Barz, T., Huss, J. & Frey, O (2005). Erstellung und Nutzung praxisrelevanter Szenarien für die Prototypenentwicklung in der Prozessführung. In L. Urbas & Ch. Steffens (Hrsg.), Zustandserkennung und Systemgestaltung (S. 227-230). Düsseldorf: VDI Verlag.
- Bortz, J. (1999). Statistik für Sozialwissenschaftler. Berlin: Springer.
- Bortz, J. & Döring, N. (2006). Forschungsmethoden und Evaluation für Human- und Sozialwissenschaftler. Berlin: Springer.
- Bortz, J. & Lienert, G. A. (2003). Kurzgefasste Statistik für die klinische Forschung. Berlin: Springer.
- Buono, P., Aris, A., Plaisant, C., Khella, A. & Shneiderman, B. (2005). Interactive pattern search in time series. In R. F. Erbacher, J. C. Roberts, M. T. Gröhn & K. Borner (Eds.), Proceedings of Conference on Visualization and Data Analysis, Washington DC, 2005 (pp. 175-186). Bellingham WA. SPIE.

- Carroll, J. M. (2000). Making use: Scenario-based design of human-computer interactions. Cambridge, MA: MIT Press.
- Cheung, J. T. Y. & Stephanopoulos, G. (1990). Representation of process trends, Part I. Computers and Chemical Engineering, 14, 495-510.
- Cockburn, A. (2005). *Writing effective use cases*. Boston, MA: Addison-Wesley.
- DIN EN ISO 9241-10. (1996). Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten. Teil 10: Grundsätze der Dialoggestaltung. Berlin: Beuth.
- DIN EN ISO 9241-11. (1998). Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten. Teil 11: Anforderungen an die Gebrauchstauglichkeit - Leitsätze. Berlin: Beuth.
- Fitts, P. M. (1951). Human engineering for an effective air navigation and traffic control system (Report prepared for the Air Navigation Development Board). Columbus: Ohio State University Research Foundation.
- Flehmig, F., von Watzdorf, R. & Marquardt, W. (1998). Identification of trends in process measurements using the wavelet transform. Computers and Chemical Engineering, 22, 491-496.
- Frey, O. (2006). Entwicklung eines effizienten Algorithmus für die Suche nach nutzerdefinierten Prozesszuständen in der chemischen Industrie. Unveröffentlichte Diplomarbeit, Technische Universität Berlin.
- Gebhardt, A. (2003). Rapid prototyping. München: Hanser.
- Green, D. M. & Swets, J. A. (1988). Signal detection theory and psychophysics. New York: Wiley.
- Greenbaum, T. L. (1993). The handbook for focus group research. Toronto: Lexington.
- Haimowitz, I. J. & Kohane, I. S. (1993). Automated trend detection with alternate temporal hypotheses. In Ruzena Bajcsy (Ed.), Proceedings of the 13th International Joint Conferences on Artificial Intelligence in Chambéry, 1993 (Vol. 1, pp. 146-151). San Mateo, CA: Morgan Kaufmann.

- Hochheiser, H. (2003). Interactive graphical querying of time series and linear sequence data sets. Unpublished doctoral dissertation, University of Maryland, Department of Computer Science, College Park, MD, USA.
- Hochheiser, H. & Shneiderman, B. (2002). Visual queries for finding patterns in time series data. Technical Report ISR TR 2005-27. University of Maryland, Computer Science Department. Available: <https://drum.umd.edu/dspace/handle/1903/6490>
- Hochheiser, H. & Shneiderman, B. (2004). Dynamic query tools for time series data sets: timebox widgets for interactive exploration. Information Visualization 3, 1-18.
- Honeywell Inc. (2005). PlantScape® Vista™: Specification and technical data. Release 400 Manual. Available: <http://hpsweb.honeywell.com/Cultures/en-US/Products/default.htm>
- Horst, J. H. & Beichl, I. (1996, June). Efficient piecewise linear approximation of space curves using chord and arc length. Paper presented at the SME Applied Machine Vision '96 Conference, Cincinnati, Ohio.
- Huber, A. S. (2001). Der Siemens-Bereich A&D auf dem Weg zu einem der führenden Anbieter in der Prozessautomatisierung und für MES-Lösungen. Pressegespräch vom 06. Februar 2001. Verfügbar unter: http://www.automation.siemens.com/wwwdocs/nc_folien/v_huber1.pdf [28.12.2007]
- Huss, J., Steffens, Ch. & Urbas, L. (2006). Gebrauchstauglichkeit bei der Gestaltung von Mensch-Technik-Systemen in der Prozessführung. Automatisierungstechnische Praxis, 48 (3), 36-41.
- Huss, J. & Urbas L. (2006). Expert Oriented Design of an Engineering Tool for Data Base Investigation. Paper presented at the 16th World Congress on Ergonomics, Maastricht, Netherlands.
- Huss, J. & Barz, T. (2005, April). Trendbasierte Suche in Prozessdatenbanken. Poster auf dem 2. Berliner Symposium für Informationstechnologien für Entwicklung und Produktion in der Verfahrenstechnik, Berlin.
- Jacobson, I. (1993). *Object-oriented software engineering: A use case driven approach*. New York: ACM Press.

- Janusz M. & Venkatasubramanian V. (1991). Automatic generation of qualitative descriptions of process trends for fault detection and diagnosis. Engineering Applications of Artificial Intelligence Journal, 4 (5), 329-339.
- Karrer, K., Baumgarten, T. & Seifert, K. (2005). Parallel-iterative Entwicklung eines Assistenzsystems zur Fahrerzustandserkennung. In K. Karrer, B. Gauss, & Ch. Steffens (Hrsg.), Beiträge zur Mensch-Maschine-Systemtechnik aus Forschung und Praxis (S. 229-246). Düsseldorf: Symposion.
- Kaynak, O. (2005). How Intelligent are Intelligent Systems? In M. Mohammadian (Ed.), Book of Abstracts and Conference Program (pp. 14-15), International Conference on Computational Intelligence for Modelling, Control & Automation, Wien.
- Keogh, E., Chakrabarti, K., Pazzani, M. & Mehrotra, S. (2001, May). Locally adaptive dimensionality reduction for indexing large time series databases. Paper presented at the ACM SIGMOD Conference on Management of Data, Santa Barbara, CA.
- Keogh, E., Hochheiser, H. & Shneiderman, B. (2002, October). An augmented visual query mechanism for finding patterns in time series data. Paper presented at the 5th International Conference for Flexible Query Answering Systems, Copenhagen, Denmark.
- Keogh, E. & Kasetty, S. (2003). On the need for time series data mining benchmarks: A survey and empirical demonstration. Data Mining and Knowledge Discovery, 7 (4), 349-371.
- Kindsmüller, M. C. (2006). Trend-Literacy - zur Interpretation von Kurvendarstellungen in der Prozessführung. Aachen: Shaker.
- Kivikunnas, S. (1999, September). Overview of process trend analysis methods and application. Paper presented at the Workshop on Applications in Chemical and Biochemical Industry, Aachen, Germany.
- Komischke, T. & Burmester, M. (2000). User-centered standardization of industrial process control user interfaces. International Journal of Human-Computer Interaction, 12 (3&4), 375-386.
- Krick, V. (2003). Die Relevanz von Kurvenbildern bei der computergesteuerten Bedienung prozesstechnischer Anlagen - Eine psychologische Annäherung mit multiplen mediengestützten Methoden. Unveröffentlichte Diplomarbeit, Freie Universität Berlin.

- Lahres, B. & Rayman, G. (2006). Praxisbuch Objektorientierung. Bonn: Galileo Press.
- Lewis, C. & Wharton, C. (1997). Cognitive walkthroughs. In M. Helander, T. K. Landauer & P. Prabhu (Eds.). Handbook of Human-Computer Interaction (pp. 717-732). New York: Elsevier.
- Mallat, S. & Zhong, S. (1992). Characterization of signals from multiscale edges. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14 (7), 710-732.
- Mallat, S. (1999). A wavelet tour of signal processing. San Diego: Academic Press.
- Miller, G. A. (1956). The magical number seven, plus or minus two. Psychological Review, 63, 81 - 97.
- Misra, S., Kumar, V. & Kumar, U. (2005, May). Goal-oriented or scenario-based requirements engineering technique – what should practitioner select? Paper presented at the 18th IEEE Canadian Conference on Electrical and Computer Engineering, Saskatoon.
- Maurya, M. R., Rengaswamy, R. & Venkatasubramanian, V. (2007). Fault diagnosis using dynamic trend analysis: A review and recent developments. Engineering Applications of Artificial Intelligence, 20, 133-146.
- Nielsen, J. (1993). Usability Engineering. San Diego: Kaufmann.
- Nielsen, J. (1994). Heuristic Evaluation. In J. Nielsen & R. Mack (Eds.). Usability inspection methods (pp. 25-62). New York: Wiley & Sons.
- Oppermann, R., Murchner, B., Reiterer, H. & Koch, M (1992). Software-Ergonomische Evaluation. EVADIS II. Berlin: Gruyter.
- OSIsoft Inc. (2007). *PI Batch View*. Product Information. Available: <http://www.osisoft.com/Products/PI+BatchView.htm> [03.01.2008]
- Pahl, G., Beitz, W., Feldhusen, J. & Grote, K. H. (2005). Pahl/Beitz Konstruktionslehre: Grundlagen erfolgreicher Produktentwicklung Methoden und Anwendung. Berlin: Springer.
- Pieters, H. A. & Krätzner, M. (2006). ERP-MES-Kopplung am Beispiel von SAP R/3 und easyworks. Unveröffentlichte Diplomarbeit, Fachhochschule Oldenburg, Ostfriesland/Wilhelmshaven.
- Pomberger, G. & Blaschek, G. (1996). Software-Engineering - Prototyping und objektorientierte Programmierung. München: Carl Hanser Verlag.

- Rengaswamy, R. & Venkatasubramanian, V. (1995). A syntactic pattern-recognition approach for process monitoring and fault diagnosis. Engineering Applications of Artificial Intelligence, 8 (1), 35-51.
- Rittner, F. & Döring, M. (2007). Kurven und Loops in der Beatmung. Drägerwerk AG. Verfügbar unter: http://www.draeger-medical.de/MT/internet/pdf/lib/de/ventilation/loops_book.pdf [11.01.2008]
- Schuler, H. (2006). Automation in Chemical Industry. Automatisierungstechnik, 54 (8), 363-371.
- Shahabi, C., Tian, X. & Zhao, W. (2000, July). TSA-tree: a wavelet based approach to improve the efficiency of multi-level surprise and trend queries. Paper presented at the 12th Int'l Conference on Scientific and Statistical Database Management, Berlin, Germany.
- Sheridan, T. B. (1997). Task analysis, task allocation and supervisory control. In M. Helander, T. K. Landauer & V. B. Prabhu (Eds.), Handbook of human-computer interaction (pp. 87-106). Amsterdam: Elsevier.
- Sheridan, T. B. (2000). Function allocation: algorithm, alchemy or apostasy? International Journal of Human-Computer Studies, 52, 203 – 216.
- Struzik, Z. R. & Siebes, A. (1998). Wavelet transform in similarity paradigm II. Scientific Report. Stichting Mathematisch Centrum. Amsterdam.
- Sutcliffe, A. (2002). User centred requirements engineering. London: Springer.
- Sutcliffe, A. (2003, September). Scenario-based requirements engineering. Paper presented at the 11th IEEE International Requirements Engineering Conference, Monterey Bay, USA.
- Timpe, K. P. & Kolrep, H. (2000). Das Mensch-Maschine-System als interdisziplinärer Gegenstand. In K. P. Timpe, T. Jürgensohn & H. Kolrep (Hrsg.), Mensch-Maschine-Systemtechnik (S. 9-40). Düsseldorf: Symposion.
- Treier, C., Schildheuer, R. & Weich, A. (2006). Veränderung im Arbeitssystem Warte durch den Einzug digitaler Leittechnik. In Gesellschaft für Arbeitswissenschaften e.V. (Hrsg.), Innovation für Arbeit und Organisation (S. 69-73). Dortmund: GfA Press.
- Urbas L. (1999). Entwicklung und Realisierung einer Trainings- und Ausbildungsumgebung zur Schulung der Prozessdynamik und des Anlagenbetriebs im Internet. Düsseldorf: VDI Verlag.

- Urbas, L. & Timpe, K. P. (2004). Leitwartengestaltung für die Prozessführung. In R. Bruder & M. Schütte (Hrsg.), Ergonomie und Design (S. 145-154). Stuttgart: Ergonomia.
- Urbas, L., Steffens, Ch., Beu, A. & Jakob, F. (2006). Der Nutzer steht im Mittelpunkt - Usability in der Industrie. In A. Grütz (Hrsg.), Jahrbuch Elektrotechnik 2007 (S. 39-52). Berlin: VDE-Verlag.
- VDI/VDE 3699 Blatt 4. (1997). Prozessführung mit Bildschirmen: Kurven. Berlin: Beuth.
- Vedam, H. & Venkatasubramanian, V. (1997, June). A wavelet theory-based adaptive trend analysis system for process monitoring and diagnosis. Paper presented at the American Control Conference, Albuquerque, NM.
- Venkatasubramanian, V., Rengaswamy, R., Yin, K. & Kavuri, S. N. (2003). A review of process fault detection and diagnosis Part I: Quantitative model-based methods. Computers and Chemical Engineering, *27*, 293-311.
- Wandke, H. (1997). Digitale Anzeigen. In H. Luczak & W. Volpert (Hrsg.), Handbuch der Arbeitswissenschaft (S. 530-534). Stuttgart: Schäffer-Poeschel.
- Wang, C. & Wang, X. S. (2000, July). Supporting content-based searches on time series via approximation. Paper presented at the 12th Int'l Conference on Scientific and Statistical Database Management, Berlin, Germany.
- Wattenberg, M. (2001, April). Sketching a graph to query a time series database. Extended Abstracts of the 2001 Conference Human Factors in Computing Systems, Seattle.
- Wharton, C., Rieman, J., Lewis, C. & Polson, P. (1994). The cognitive walkthrough method: A practitioner's guide. In J. Nielsen & R. L. Mack (Eds.), Usability inspection methods (pp. 105-141). New York: Willy & Sons.
- Wickens, C., Lee, J., Liu, Y. & Becker, S. (2004). An introduction to human factors engineering. New Jersey: Prentice Hall.
- Witkin, A. P. (1983). Scale-space filtering: A new approach to multi-scale description. In S. Ullman & W. Richards (Eds.), Image Understanding (pp. 79-95). Norwood: Ablex.
- Wittenberg, C. (2004). Benutzeranforderungen für den Einsatz von mobilen Endgeräten in der Industrieautomatisierung. Automatisierungstechnik, *52* (3), 136-145.

Begriffsglossar

Batch	<i>Im Gegensatz zur kontinuierlichen Betriebsweise werden in einem Batch (eng. Stapel) mehrere Einzeloperationen sequentiell abgearbeitet.</i>
Datenbank	<i>Gesamtmenge aller per Zeitstempel gespeicherten Daten</i>
dyadisch	<i>dem Zweiersystem zugehörend</i>
E-Kette	<i>Extremstellen-Kette. Folge an Punktepaaren, welche die Koordinaten zueinander gehörender Extremstellen auf den Wavelet-Transformierten abbilden. Die Ketten werden über alle Skalen der Wavelet-Transformation gebildet.</i>
Episode	<i>Zeitlich begrenztes Segment eines Trends</i>
Ergebniskurve	<i>Kurve, welche (von einem Algorithmus) als ähnlich der Referenzkurve beurteilt wird.</i>
Ergebnistrend	<i>Trend für Ergebniskurve (durch Datenbankindexierung erzeugt)</i>
explorativ	<i>forschend, prüfend</i>
Faltung	<i>Bezeichnet einen mathematischen Operator, der für zwei Funktionen f und g eine dritte Funktion in Form einer "Überlappung" liefert.</i>
Faltungsskala	<i>Entspricht dem Skalierungsparameter (synonym zu Frequenzauflösung) der Funktion g, mit welchem die Faltung der Originalfunktion durchgeführt wird.</i>
High-Fidelity-Prototype	<i>Prototyp mit ausgeprägter Funktionalität.</i>
Human-Factors	<i>Sammelbegriff für psychische, kognitive und soziale Einflussfaktoren in Mensch-Maschine-Systemen</i>
Kurve	<i>Darstellung des endlichen Verlaufs der Werte einer Größe als Linie in Abhängigkeit von der Zeit (Start- und Endwert sind bekannt)</i>

Kurvenbild	<i>Ein Kurvenbild besteht nach VDI/VDE 3699-4 (1997) aus einem Kurvenfeld, einem Kurvenbeschriftungsfeld und einem Titelfeld.</i>
Kurven- darstellung	<i>Kartesische Koordinatendarstellung, die eine Zeitdimension auf der Abszisse und eine Prozessgröße (Mess- oder Stellgröße) auf der Ordinate aufweist.</i>
Kurvenfeld	<i>Das Kurvenfeld enthält ein oder mehrere Kurvendarstellungen (Kurve versus Kurvenschar) und nimmt als Hauptinformationsträger des Kurvenbilds die zentrale Bildschirmposition und den größten Teil der Anzeigefläche ein.</i>
Kurvenhistorie	<i>Kurve, bei der sowohl zeitlicher Start- als auch Endwert in der Vergangenheit liegt</i>
Kurvenlesen	<i>(engl. Trend-Literacy) Wahrnehmung und Interpretation von Kurven in Form bedeutungshaltigen Trends.</i>
Kurvenschar	<i>Menge aller im Kurvenfeld angezeigten Kurven.</i>
Kurven- vorgeschichte	<i>Dem aktuellen Wert einer Prozessgröße vorangehender Verlauf.</i>
Mexican-Hat- Wavelet	<i>Typ eines Mother-Wavelets, welches sowohl der zweiten Ableitung der Gaußfunktion als auch der graphischen Darstellung eines mexikanischen Hutes ähnelt.</i>
Mother-Wavelet	<i>Typischer Grundtyp einer Funktion, mit Hilfe derer eine Originalfunktion in eine Wavelet-Transformierte überführt wird.</i>
Prozess- monitoring	<i>Überwachen und Beobachten eines vorrangig automatisch ablaufenden verfahrenstechnischen Prozesses</i>
Referenz-Episode	<i>Zeitlich begrenztes Segment eines vom Nutzer beschriebenen Referenz-trends</i>
Referenzkurve	<i>Kurve, für welche ähnliche Kurven aus der Kurvenhistorie gefunden werden sollen</i>

Referenz- Kurvenschar	<i>Menge aller im Kurvenfeld angezeigten Referenzkurven</i>
Referenztrend	<i>Vom Nutzer beschriebener Trend einer ausgewählten Referenzkurve</i>
Referenzszenario	<i>Gesamtheit aller Kurven, welche über den Dialog „Daten“ in den Prototypen geladen werden. In dieser Arbeit entspricht ein Referenzszenario einem Subszenario, welches als Teil des künstlich erstellten Testsets erstellt wurde.</i>
summative Evaluation	<i>Form einer Produkt bewertenden Evaluation, bei welcher die abschließende Qualitätsbewertung im Vordergrund steht.</i>
Testscenario	<i>Charakteristischer, in dieser Arbeit künstlich erzeugter Kurvenverlauf. Ein Szenario enthält eine Menge an Testszenarien, welche alle einen ähnlichen typischen Verlauf mit jedoch unterschiedlicher quantitativer Ausprägung enthalten.</i>
Testset	<i>Menge aller zusammen gehörigen Testszenarien inklusive Referenzszenario</i>
Transkript	<i>Verschriftung einer Audio-Aufzeichnung</i>
Trend	<i>Daten reduzierende, abstrahierende Interpretation einer Kurve / Beschreibung einer Entwicklungscharakteristik</i>
Trendlet	<i>„Buchstaben“ oder „Primitiva“, mit welchen eine Episode charakteristisch beschrieben werden kann</i>
Trendlet Alphabet	<i>Alphabet aus einer Reihe unterschiedlicher geometrischer Primitiva</i>
Trend-Literacy	<i>Wahrnehmung und Interpretation von Kurven in Form bedeutungshaltigen Trends.</i>
Usability	<i>Gebrauchstauglichkeit eines Produktes in einem konkreten Nutzungskontext</i>

Wavelet-Skalen	<i>Entspricht dem Skalierungsparameter (synonym zu Frequenzauflösung) des Mother-Wavelet, mit welchem die Transformation der Originalfunktion durchgeführt wird.</i>
Wavelet-Transformierte	<i>Resultat einer Wavelet-Transformation</i>
Zeitreihe	<i>Reihe von Werten einer beliebigen Variablen über einen Zeitbereich äquidistanter Schrittweiten.</i>

Zeichenglossar

II.1 Funktionsanalyse – Stand der Technik

II.1.1 Transformation von Zeitreihen

Ψ	Mother-Wavelet
Θ	Mexican-Hat-Wavelet
σ	Skalierungsparameter
τ	Translationsparameter
$f_{\sigma}(t)$	Faltungsfunktion

III.1.1 Entwurf I

III.1.1.2 Vorverarbeitung

$\hat{f}_{\sigma}(t)$	Normierte Faltungsfunktion
$m_{\sigma,t}$	Extremstelle zum Zeitpunkt t auf der Faltungsskala σ
$t_{m_{\sigma}}$	Zeitpunkt einer Extremstelle m auf der Faltungsskala σ
$\hat{f}_{\sigma}(t_m)$	Normierte Amplitude einer Extremstelle zum Zeitpunkt t auf der Faltungsskala σ
M	Kette an Extremstellen gleichen Vorzeichens über eine Menge an Faltungsskalen
m_{linTol}	Schwellwert zur Identifikation von Start- und Endzeitpunkten linearer Episoden
t_{lin}	Start- bzw. Endzeitpunkte linearer Episoden

III.1.1.3 Suchprozedur

t_{st} Vom Nutzer angegebener Startzeitpunkt

t_{end} Vom Nutzer angegebener Endzeitpunkt

Lineare Episoden

$\emptyset M_{st}$ Extremstellenkette, welche den Start einer linearen Referenz-Episode kennzeichnet

$\emptyset M_{end}$ Extremstellenkette, welche das Ende einer linearen Referenz-Episode kennzeichnet

\widehat{m}_{linTol} Spezifischer Schwellwert zur Identifikation von Start- und Endzeitpunkten linearer Episoden

Δt Vom Nutzer angegebene Toleranz für die Länge linearer Episoden

Δh Vom Nutzer angegebene Toleranz für die Höhe linearer Episoden

Gekrümmte Episoden

\overline{t}_{m_σ} Zeitpunkte der Extremstellen $m_{\sigma,t}$ einer potentiellen Ergebnis-Episode

$m_{gekr,t}$ Schwellwert für die Abweichungen der Extremstellen zweier Ketten auf der Zeitachse

$m_{gekr,f}$ Schwellwert für die Abweichungen der Extremstellen zweier Ketten auf der Zeitachse

III.1.2 Entwurf II

III.1.2.2 Vorverarbeitung

Transformation

$\hat{f}_\sigma(t)$ Standardisierte Faltungsfunktion

Extraktion

Z_c Index gekrümmter Episoden

T_c Index aller Trendlets (bzw. entsprechenden Vorzeichenkombinationen) gekrümmter Episoden

W_c Index aller Start- und Endzeitpunkte gekrümmter Episoden

H_c Index aller Amplituden der Start- und Endpunkte gekrümmter Episoden

${}_\Lambda L_c$ Index linearer Episoden

s Bogenlänge einer linearen Episode (Originallänge)

c Sehnenlänge einer linearen Episode (approximierte Länge)

${}_\Lambda W_c$ Index aller Start- und Endzeitpunkte linearer Episoden

${}_\Lambda H_c$ Index aller Amplituden der Start- und Endpunkte linearer Episoden

Γ Suchraum

Parameter

$\varsigma = \{\sigma_1, \dots, \sigma_p\}$ Skalenraum, welcher eine Menge an verschiedenen Faltungsskalen enthält

$\Lambda = \{\lambda_1, \dots, \lambda_q\}$ Linearitätsindex, welcher eine Menge an verschiedenen Linearitätskoeffizienten enthält

III.1.2.3 Suchprozedur

Eingangsparameter

Ω	Referenztrend
Ω_e	Referenz-Episode
T_e	Trendlet einer Referenz-Episode
w_e	Länge einer Referenz-Episode
w_{ε_e}	Längentoleranz einer Referenz-Episode
h_e	Höhe einer Referenz-Episode
h_{ε_e}	Höhentoleranz einer Referenz-Episode
υ_e	Krümmungsfaktor einer Referenz-Episode
λ_{Ω}	Linearitätsparameter für Referenztrend Ω
Φ	Menge aller Toleranzen für die Verknüpfung aufeinanderfolgender Episoden eines Trends
θw_e	Relative Toleranz für die Überlappung zweier aufeinanderfolgender Episoden Ω_e und Ω_{e-1} entlang der Zeitachse
$\widehat{\theta w_e}$	Absolute Toleranz für die Überlappung zweier aufeinanderfolgender Episoden Ω_e und Ω_{e-1} entlang der Zeitachse
ϕw_e	Relative Toleranz für den Abstand zweier aufeinanderfolgender Episoden Ω_e und Ω_{e-1} entlang der Zeitachse
$\widehat{\phi w_e}$	Absolute Toleranz für den Abstand zweier aufeinanderfolgender Episoden Ω_e und Ω_{e-1} entlang der Zeitachse

θh_e	Relative Toleranz für die Verschiebung zweier aufeinanderfolgender Episoden Ω_e und Ω_{e-1} entlang der Werteachse
$\widehat{\theta h_e}$	Absolute Toleranz für die Verschiebung zweier aufeinanderfolgender Episoden Ω_e und Ω_{e-1} entlang der Werteachse
$\xi = \begin{cases} G_w, G_h, G_{\theta w} \\ G_{\phi w}, G_{\theta h}, G_v \end{cases}$	Menge an Gewichtungsfaktoren verschiedener Ähnlichkeitskriterien (Länge, Höhe, Episodenüberlappung, Episodenabstand, vertikale Episodenverschiebung, Krümmung)
$\zeta = \begin{cases} S_w, S_h, S_{\theta w} \\ S_{\phi w}, S_{\theta h}, S_v \end{cases}$	Menge an Sensitivitätsparameter verschiedener Ähnlichkeitskriterien (Länge, Höhe, Episodenüberlappung, Episodenabstand, vertikale Episodenverschiebung, Krümmung)
r_{tol}	Redundanztoleranz für die Überlappung zweier Ergebnistrends aus der Menge \bar{P} .
Suchraumeinschränkung	
Λ_e	Zur Suche der Referenz-Episode Ω_e eingeschränkter Linearitätsindex
ζ_e	Zur Suche der Referenz-Episode Ω_e eingeschränkter Skalenraum
Γ_e	Zur Suche der Referenz-Episode Ω_e eingeschränkter Suchraum
Trendsuche	
$\left(\overline{\overline{Y_e}}, \overline{\overline{Y_e}}, \overline{\overline{Y_e}}, \overline{\overline{Y_e}} \right)$	Zwischenergebnisse
Y_e	Ergebnis-Episode
T_e	Trendlet der Ergebnis-Episode Y_e des Ergebnistrends ${}_i Y$

a_e	Startzeitpunkt einer Ergebnis-Episode Y_e
b_e	Endzeitpunkt einer Ergebnis-Episode Y_e
$\hat{f}_\sigma(a_e)$	Amplitude zum Startzeitpunkt einer Ergebnis-Episode Y_e
$\hat{f}_\sigma(b_e)$	Amplitude zum Endzeitpunkt einer Ergebnis-Episode Y_e
Y	Ergebnistrend
\bar{P}	Menge aller potentiellen Ergebnistrends ${}_i Y$
P	Finale Menge aller Ergebnistrends ${}_i Y$
E	Menge aller Ergebnistrends und deren Ähnlichkeitsindikatoren

Ähnlichkeitsschätzung

g_w	Ähnlichkeitsindikator Länge
g_h	Ähnlichkeitsindikator Höhe
g_v	Ähnlichkeitsindikator Krümmung
$g_{\theta w}$	Ähnlichkeitsindikator Episodenüberlappung
$g_{\phi w}$	Ähnlichkeitsindikator Episodenabstand
$g_{\theta h}$	Ähnlichkeitsindikator Episodenverschiebung
g_e	Ähnlichkeitsindikator der Ergebnis-Episode Y_e
g	Ähnlichkeitsindikator des Ergebnistrends Y

Anhang

A	SYSTEMANALYSE	164
A.1	Aufgabenanalyse – Stand des Wissens	164
A.1.1	Das Kurvenbild als Werkzeug der Prozessüberwachung und -analyse	164
A.1.1.1	Interviewleitfaden	164
A.1.1.2	Fragebogen	164
A.1.2	Aufgabenanalyse anhand einer Prototypen-Evaluation.....	166
A.1.2.1	Prototyp - Version 0.7	166
A.1.2.2	Sitzungsprotokoll	167
B	SYSTEMDESIGN	168
B.1	Nutzerdialog	168
B.1.1	Entwurfsprozess	168
B.1.1.1	Kalkulationstabellen zur Dialogkomponente „Trendbeschreibung“	168
B.1.1.2	Mockups	171
B.1.2	Prototyp	174
B.1.2.1	Zustandsübergangdiagramm für Nutzerdialog	174
B.1.3	Expertenevaluation	175
B.1.3.1	Zeitplan	175
B.1.3.2	Allgemeine Instruktionen	176
B.1.3.3	Aufgabenstellungen für Bediener	177
B.1.3.4	Integration verschiedener Kategoriensysteme für Kriterien der Gebrauchstauglichkeit	178
C	SYSTEMBEWERTUNG	180
C.1	Untersuchungsmaterial	180
C.1.1	Testsets und Testszenarien	180
C.1.1.1	Testset 1.....	180
C.1.1.2	Testset 2.....	180
C.1.1.3	Testset 3.....	182
C.1.1.4	Testset 4.....	184
C.2	Durchführung	185
C.2.1	„Google-Analogie“	185
C.2.2	Veranschaulichung der Trendbeschreibung	185
C.2.3	Erläuterung der Suchindexe	186
C.2.4	Allgemeine Erläuterungen des Prototypen.....	187
C.2.5	Aufgabeninstruktionen	188

A Systemanalyse

A.1 Aufgabenanalyse – Stand des Wissens

A.1.1 Das Kurvenbild als Werkzeug der Prozessüberwachung und -analyse

A.1.1.1 Interviewleitfaden

Definition Kurven

- Darstellung einer Prozessvariable (Steuer- oder Regelgröße, Messgrößen oder berechnete Kennwerte) über die Zeit

Anwendungsfelder in der Prozessführung (Brainstorming)

- Persönlicher Erfahrungsbereich
- Anwendungsfelder außerhalb des persönlichen Erfahrungsbereichs (Ratingskala)

Funktion der Kurven (pro Anwendungsfeld)

- persönliches F&E-Feld -> Critical Incidents (Wann?, Wobei?, Wo?, Wofür?)
- Operatives Anlagenfahren (Fragebogen -> Kommentare!)

A.1.1.2 Fragebogen

Um die allgemeine Verwendungs-Häufigkeit (im Sinne von Zeitdauer) von Trend- bzw. Kurvenbilder durch Anlagenfahrer im Schichtdienst abzuschätzen, vergleichen Sie diese bitte paarweise mit anderen gängigen Bildern im PLS.

	häufiger		
Schnee		X	Regen
Trend- bzw. Kurvenbilder			Systembilder
Meldeliste			Grafikbilder (Übersichtsbilder)
Trend- bzw. Kurvenbilder			Meldeliste
Systembilder			Meldeliste
Grafikbilder (Übersichtsbilder)			Trend- bzw. Kurvenbilder
Meldeliste			Gruppenbilder
Gruppenbilder			Systembilder
Gruppenbilder			Trend- bzw. Kurvenbilder
Grafikbilder (Übersichtsbilder)			Gruppenbilder
Systembilder			Grafikbilder (Übersichtsbilder)

Geben Sie an bzw. schätzen Sie die Häufigkeit (im Sinne von Zeitdauer), mit der Anlagenfahrer im Schichtdienst Kurven aus dem PLS für Ihre Tätigkeit nutzen.

Differenzieren Sie dafür zwischen einer Auswahl aus verschiedenen Aufgaben, die mit der Nutzung von Kurven verfolgt werden.

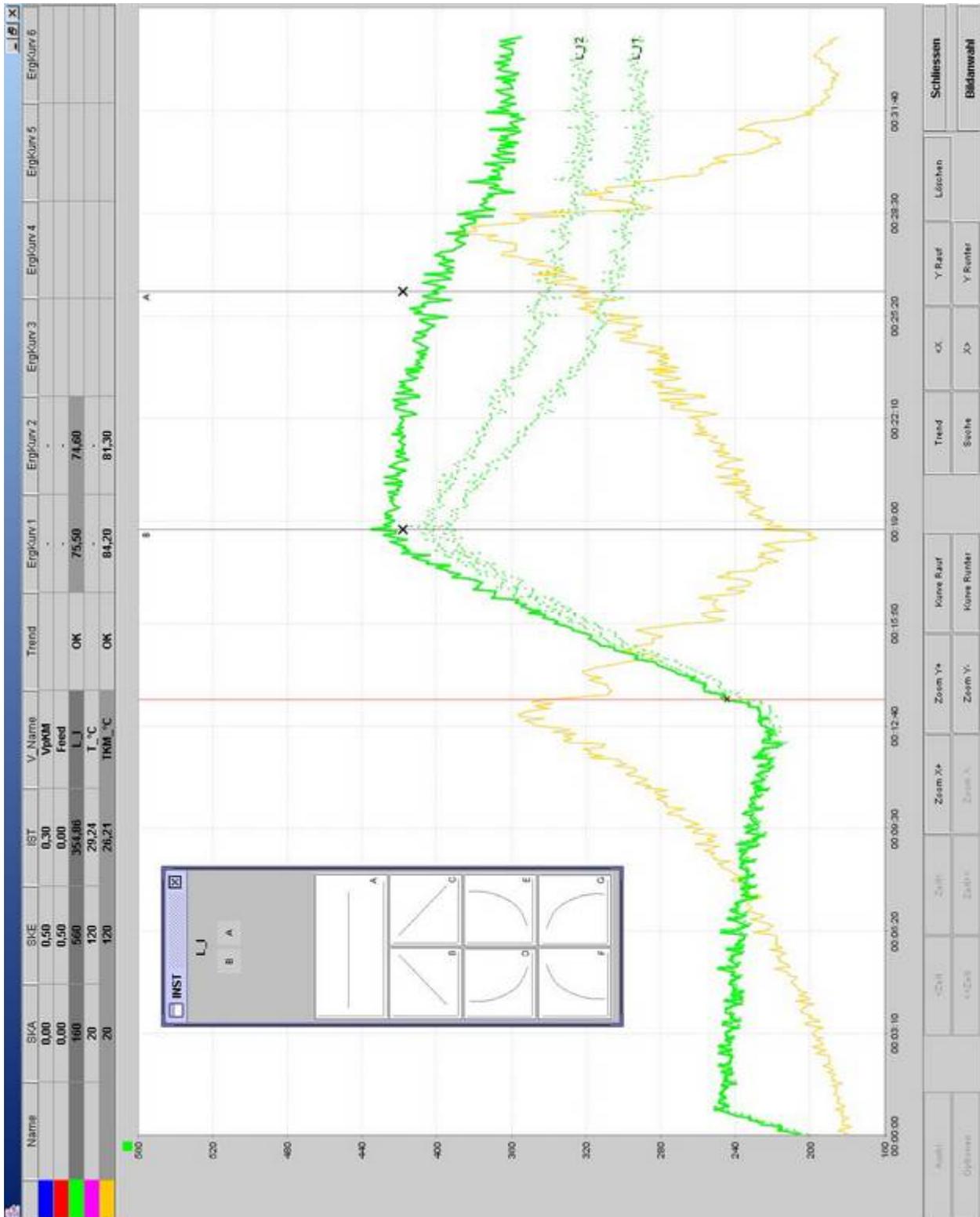
Gesamteindruck	Wie hat sich der Prozess über den Zeitraum der letzten Schicht verhalten?
Messwerte-Ablesen	Was ist/war der konkrete Messwert zu einem konkreten Zeitpunkt?
Fehler/Störungsdiagnose	Was ist ohne direkten Eingriff an Auffälligkeiten geschehen & warum?
Wirkkontrolle	Wie hat sich ein manueller Eingriff ausgewirkt?
Qualitätskontrolle	Erfüllt mein Produkt die geforderten Spezifikationen?
Prädiktion	Vorhersage vom weiteren Verlauf aktueller Prozessereignisse
Recherche-Prozesshistorie	Recherche von Kurvenverläufen von abgeschlossenen Ereignissen in der Prozesshistorie

Um Ihnen die Angabe der relativen Häufigkeit zu erleichtern, nutzen Sie bitte die Möglichkeit von Paarvergleichen in der Art von: „Bei uns in Mitteleuropa fällt häufiger bzw. mehr Regen als Schnee“

	häufiger		
Schnee		X	Regen
Gesamteindruck			Qualitätskontrolle
Gesamteindruck			Messwerte-Ablesen
Wirkkontrolle			Gesamteindruck
Wirkkontrolle			Prädiktion
Wirkkontrolle			Recherche-Prozesshistorie
Prädiktion			Gesamteindruck
Wirkkontrolle			Qualitätskontrolle
Messwerte-Ablesen			Fehler/Störungsdiagnose
Qualitätskontrolle			Fehler/Störungsdiagnose
Prädiktion			Recherche-Prozesshistorie
Messwerte-Ablesen			Prädiktion
Gesamteindruck			Recherche-Prozesshistorie
Fehler/Störungsdiagnose			Recherche-Prozesshistorie
Fehler/Störungsdiagnose			Prädiktion
Gesamteindruck			Fehler/Störungsdiagnose
Qualitätskontrolle			Recherche-Prozesshistorie
Messwerte-Ablesen			Wirkkontrolle
Messwerte-Ablesen			Recherche-Prozesshistorie
Fehler/Störungsdiagnose			Wirkkontrolle
Prädiktion			Qualitätskontrolle
Qualitätskontrolle			Messwerte-Ablesen

A.1.2 Aufgabenanalyse anhand einer Prototypen-Evaluation

A.1.2.1 Prototyp - Version 0.7



A.1.2.2 Sitzungsprotokoll

1. Vorstellung des Prototypen

Ziel des Prototypen

Nutzerdefinierte kurvenbasierte Suche nach ähnlichen Prozessverläufen in der Prozessdatenbank – Google-Analogie

Ziel der Diskussion

- Anwendungsmöglichkeiten und dafür erforderliche Änderungen bzw. Ergänzungen des Interaktionskonzepts
- Aus Zeitgründen Verzicht auf Thematisierung von Anwenderfreundlichkeit und Darstellungsfehler (lässt sich auch gut mit Domän-fremden Usability-Experten gut machen)
- noch stets Entwicklung und Optimierung des Prototypen
- parallele Entwicklung der Suchalgorithmen
- Verknüpfung in den nächsten Wochen
- Aktuell: reiner Interaktionsprototyp – keine echte Suchfunktionalität implementiert

Erläuterung des exemplarischen Anwendungsbeispiels

- Batchprozess (angelehnt an Polymerisationsprozess)
- 2 aufeinander folgende exotherme Flüssigkeitsreaktionen
- B gewünschtes Produkt
- Einflussgrößen
 - Reaktortemp. - geregelt über Kühlmittelzufluss
 - Feedzufuhr nach Rezept
- Start bei Raumtemperatur per Initiator/Katalysator bei Teilvorlage
- Kein Produktabzug

Erläuterung der Oberflächenfunktionalität

- Achsen des Kurvenbilds
- Kurvenauswahl-Panel
 - Zeile = Variable
 - Linker Tabellenteil = Referenzkurvenset
 - Rechter Tabellenteil = Ergebniskurvensets (spaltenweise)
 - Aktivationsmodi
- Bedienpanel
 - X- & Y Zoom & X- & Y Panning für alle Variablen unabhängig voneinander
 - X- & Y Zoom gemeinsam für Referenzkurve und Ergebniskurven einer Variablen
 - X- & Y Panning für Referenzkurve und Ergebniskurven einer Variablen voneinander
- Trendbeschreibung
 - Segmentierung
 - Kategorisierung
 - Spezifikation
- Ergebnispräsentation
 - Ergebniskurvensets
 - Variablensets

Fragen des Experten zur Funktionalität

2. Anwendungsmöglichkeiten des Prototypen

Anwendungsfälle & Nutzergruppen (Brainstorming)

3. Änderungsbedarf für Prototypen

- Trendletalphabet (siehe Bsp.)
- Grammatik und andere Automatisierungen
- Kombination von Kurvenbeschreibungen
- Definition von zusätzlichen Suchraumeinschränkungen (Betriebspunkt etc.)

B Systemdesign

B.1 Nutzerdialog

B.1.1 Entwurfsprozess

B.1.1.1 Kalkulationstabellen zur Dialogkomponente „Trendbeschreibung“

Hauptvarianten der Suchanfrage Spezifikationen der Suchanfrage	Priorität	Anwendungsziel	Vorteile hinsichtlich Gebrauchstauglichkeit	Nachteile hinsichtlich Gebrauchstauglichkeit	generelle (technische) Anmerkungen	Implementierungsprobleme hinsichtlich der Suchalgorithmen
Segmentierung & Kategorisierung mit Trendlets	1	Transformieren eines kardinalskalierten in ein nominales Signal -> Reduktion der Datenmenge & Komplexität Fokus auf relevanten Kurvenmustern Definition der Wertenstufen, in welcher gesucht werden soll	geringer Zeitaufwand für Nutzer Rückgriff auf Abstraktionsfähigkeit von Prozesslogiken für Kurveninterpretation	Erhöhung von zeitlichem und logischem Aufwand für Suchanfrage (allerdings nur, wenn Option nicht salutabiv)	Art und Komplexität des Trendlet-Alphanets ist Prozessabhängig geeignet für schnelle und skizzenhafte Suche in operativen Anwendungskontext	mögliche Lösung: Minkowski Edge Detection method, angewendet von BalaSubrahmanya Mitt die Erhebung von Wertenlet-Koeffizienten Messuns über die Stufen, um Wertenpunkte zu identifizieren (Die Wertenlets gehören zur orthogonalen Familie)
A. Toleranzgabe für Zeit- und Wertebereich (für jeden Anhangs- und Endpunkt einer Episode)	3	Definition der Suchtiefe	Anpassbarkeit der Suchanfrage (höherer Detaillierungsgrad als bei Option B) Beschränkung der Suchtiefe	Erhöhung von zeitlichem und logischem Aufwand für Suchanfrage (allerdings nur, wenn Option nicht salutabiv)	Variablen x & y sind konstant -> eine Beschränkung auf nur ein Koordinatenintervall für eine Variable sollte ausreichen und würde den Nutzer weniger verwirren	DTW ist nicht kompatibel mit der distanzten Wertenlet-Analyse -> Flexibilität bezüglich der Zeitachse ist nicht gegeben Der von Bleich & Stephanou gebrauchte vorgeschlagene Algorithmus benötigt eine rigorose Heuristik, um Episoden einer bestimmten Länge anzupassen ...
B. Toleranzgabe für Zeit- und Wertebereich (für Anhangs- und Endpunkt des gesamten Referenztrend)	2	Definition der Suchtiefe	Anpassbarkeit der Suchanfrage (niedriger Detaillierungsgrad als bei Option A) Beschränkung der Suchtiefe	Erhöhung von zeitlichem und logischem Aufwand für Suchanfrage (allerdings nur, wenn Option nicht salutabiv)		
C. Priorisierung der Ähnlichkeiten in Zeit- und Wertebereich für jede einzelne Episode	2	Spezifikation der Vergleichskriterien	Anpassbarkeit der Suchanfrage (höherer Detaillierungsgrad als bei Option D) Beschränkung der Suchtiefe	Erhöhung von zeitlichem und logischem Aufwand für Suchanfrage (allerdings nur, wenn Option nicht salutabiv)		
D. Priorisierung der Ähnlichkeiten in Zeit- und Wertebereich für den gesamten Referenztrend	1	Spezifikation der Vergleichskriterien	Anpassbarkeit der Suchanfrage (niedriger Detaillierungsgrad als bei Option C) Beschränkung der Suchtiefe	Erhöhung von zeitlichem und logischem Aufwand für Suchanfrage (allerdings nur, wenn Option nicht salutabiv)		
E. automatische Kategorisierung (verb right for user)	2	Beschleunigung der manuellen Suchanfrage	Verringerung von zeitlichem und logischem Aufwand für Suchanfrage	starke Beeinträchtigung des Nutzers bei seiner Interpretationsaufgabe		
F. Spezifikation der episodischen Trendlets	2	Spezifikation der episodischen Trendlets auf Interaktionsniveau -> erste Ableitung der Anhangs- und Endwerte	Anpassbarkeit der Suchanfrage Beschränkung der Suchtiefe	Erhöhung von zeitlichem und logischem Aufwand für Suchanfrage (allerdings nur, wenn Option nicht salutabiv)	inverletzte Version, die nur implementiert werden sollte, wenn die einfache Version keine betrieblichen Suchergebnisse liefert	

Hauptvarianten der Suchanfrage		Priorität	Anwendungsziel	Vorteile hinsichtlich Gebrauchstauglichkeit	Nachteile hinsichtlich Gebrauchstauglichkeit	generelle (technische) Anmerkungen	Implementierungsprobleme hinsichtlich der Suchalgorithmen
Spezifikationen der Suchanfrage							
II Mensch-Maschine Kooperation zur Kurvenapproximation		1	Komplexitätsreduktion Fokussierung auf relevante Kurveninformationen Definition der Such-relevanten Wavelet-Skala	Einheitliches Format für Referenztrend und Referenzkurve	geringeres Ausmaß an Datenreduktion als bei Methode I	geeignet für analytische Datenbankanfragen durch Betriebsingenieure	
A	numerische Toleranzangabe für Wertebereich	2	Definition der Suchtiere	Anpassbarkeit der Suchanfrage Beschränkung der Suchtiere	Erhöhung von zeitlichem und kognitivem Aufwand für Suchanfrage (allerdings nur, wenn Option nicht fakultativ)	Zahlereingabe in Text- oder Tabellenfelder für ein kontinuierliches Toleranzband oder für jeden einzelnen Trendanker kombiniert mit einer automatisierten Extrapolierung des Toleranzbandes	
B	graphische Toleranzangabe für Wertebereich	2	Definition der Suchtiere	Anpassbarkeit der Suchanfrage Beschränkung der Suchtiere	Erhöhung von zeitlichem und kognitivem Aufwand für Suchanfrage (allerdings nur, wenn Option nicht fakultativ)	graphische Manipulation des Toleranzbandes durch Mausclick	
C	Kombination von A&B	1	Definition der Suchtiere	Anpassbarkeit der Suchanfrage Beschränkung der Suchtiere	Erhöhung von zeitlichem und kognitivem Aufwand für Suchanfrage (allerdings nur, wenn Option nicht fakultativ)	numerische Eingabe/Manipulation und graphische Visualisierung des Toleranzbandes	
D	Definition der Such-relevanten Wavelet-Skala	3	Beschleunigung der Datenbank-Suche		Schwierigkeiten bei Relationierung zw. numerischen Werten (Frequenzbänder) und graphischen Bildern (Kurven). Eine Hilfe könnte die automatisierte Vorgabe von Frequenzbändern sein, die als Anker für den Nutzer dienen.		

Hauptvarianten der Suchanfrage		Priorität	Anwendungsziel	Vorteile hinsichtlich Gebrauchstauglichkeit	Nachteile hinsichtlich Gebrauchstauglichkeit	generelle (technische) Anmerkungen	Implementierungsprobleme hinsichtlich der Suchalgorithmen
Spezifikationen der Suchanfrage							
III Freihandzeichnen		3	Komplexitätsreduktion Fokussierung auf relevante Kurveninformationen		hoher Zeitaufwand und Zeichnerische Expertise notwendig	touch-screen notwendig	
A	numerische Toleranzangabe für Wertebereich	2	Definition der Suchfiliere	Anpassbarkeit der Suchanfrage Beschränkung der Suchfiliere	Erhöhung von zeitlichem und kognitivem Aufwand für Suchanfrage (allerdings nur, wenn Option nicht fakultativ)		
B	graphische Toleranzangabe für Wertebereich	3	Definition der Suchfiliere	Anpassbarkeit der Suchanfrage Beschränkung der Suchfiliere	Erhöhung von zeitlichem und kognitivem Aufwand für Suchanfrage (allerdings nur, wenn Option nicht fakultativ)		
C	Kombination von A&B	1	Definition der Suchfiliere	Anpassbarkeit der Suchanfrage Beschränkung der Suchfiliere	Erhöhung von zeitlichem und kognitivem Aufwand für Suchanfrage (allerdings nur, wenn Option nicht fakultativ)	numerische Eingabe/Manipulation und graphische Visualisierung des Toleranzbandes	
D	Definition der Such-relevanten Wavelet-Skala	3	Beschleunigung der Datenbank-Suche		Schwierigkeiten bei Relationierung zw. numerischen Werten (Frequenzbänder) und graphischen Bildern (Kurven). Eine Hilfe könnte die automatisierte Vorgabe von Frequenzbändern sein, die als Anker für den Nutzer dienen.		

B.1.1.2 Mockups

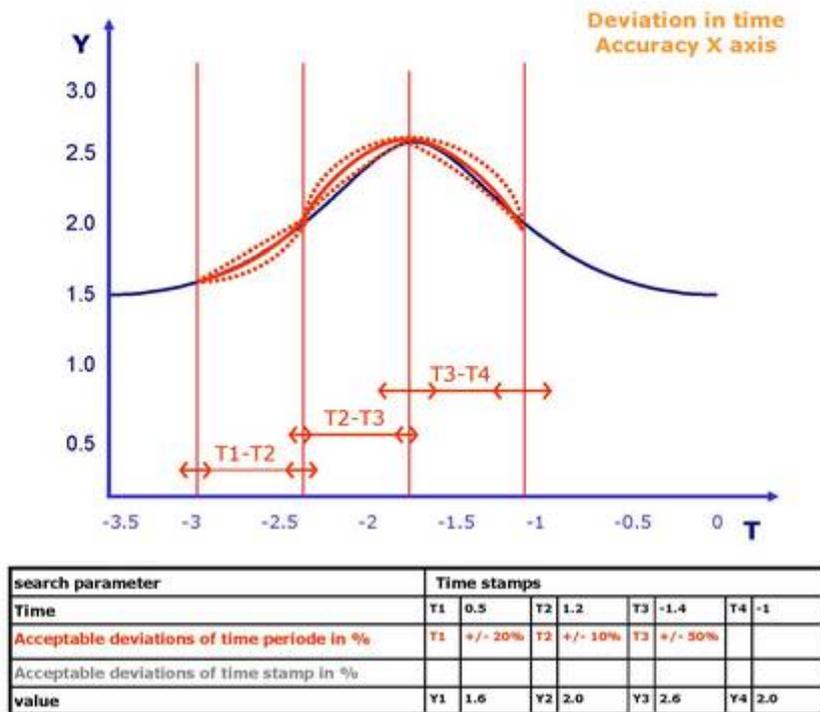


Abbildung 1 Mockup zur Spezifikation von relativen Toleranzen der Episodenlängen

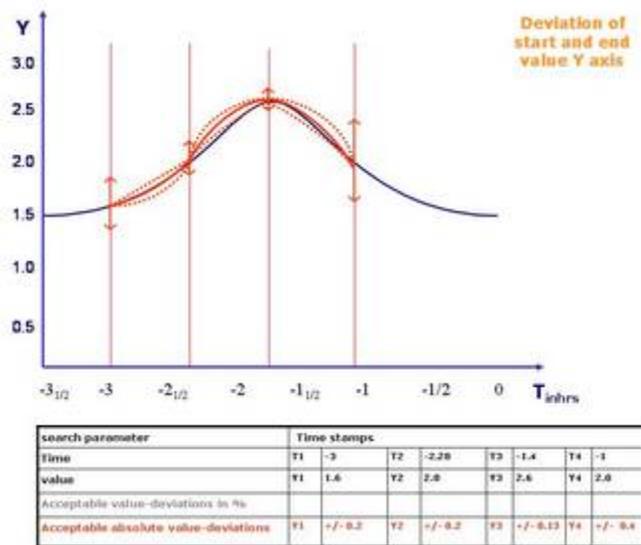


Abbildung 2 Mockup zur Spezifikation von absoluten Toleranzen der Wertekoordinaten von Episodengrenzen

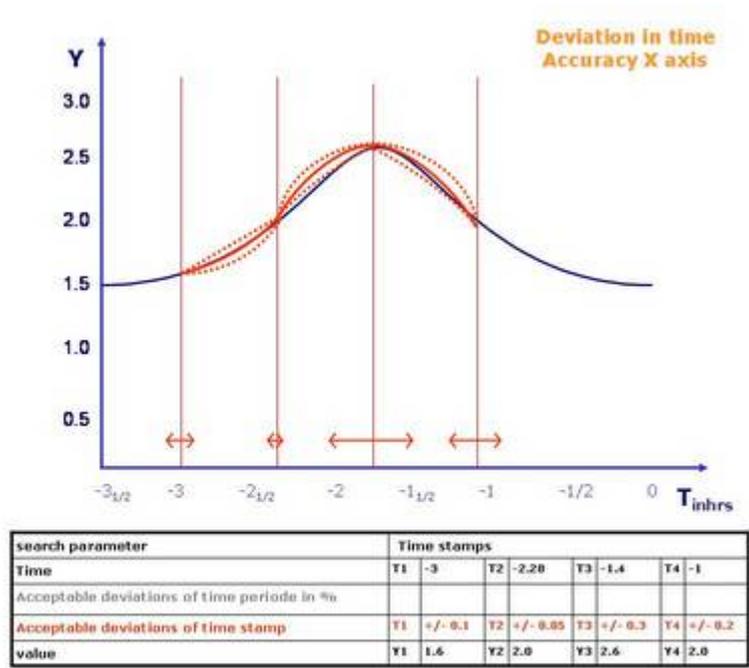


Abbildung 3 Mockup zur Spezifikation von absoluten Toleranzen der Zeitkoordinaten von Episodengrenzen

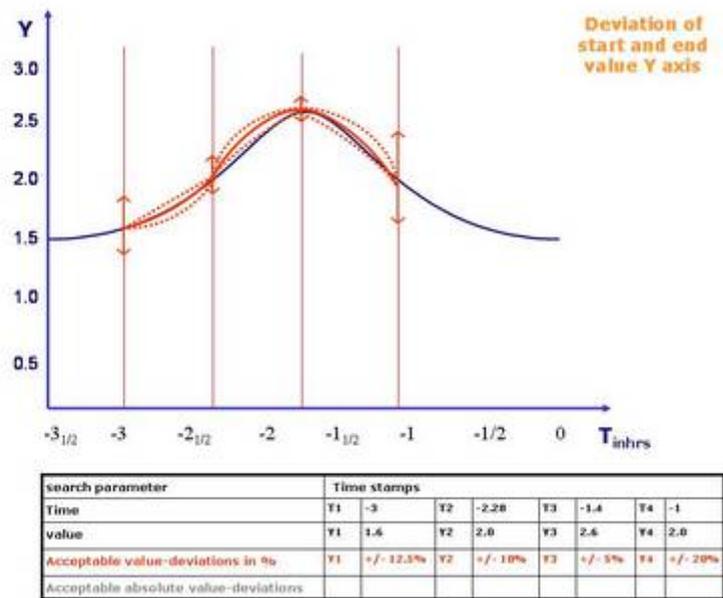


Abbildung 4 Mockup zur Spezifikation von relativen Toleranzen der Wertekoordinaten von Episodengrenzen

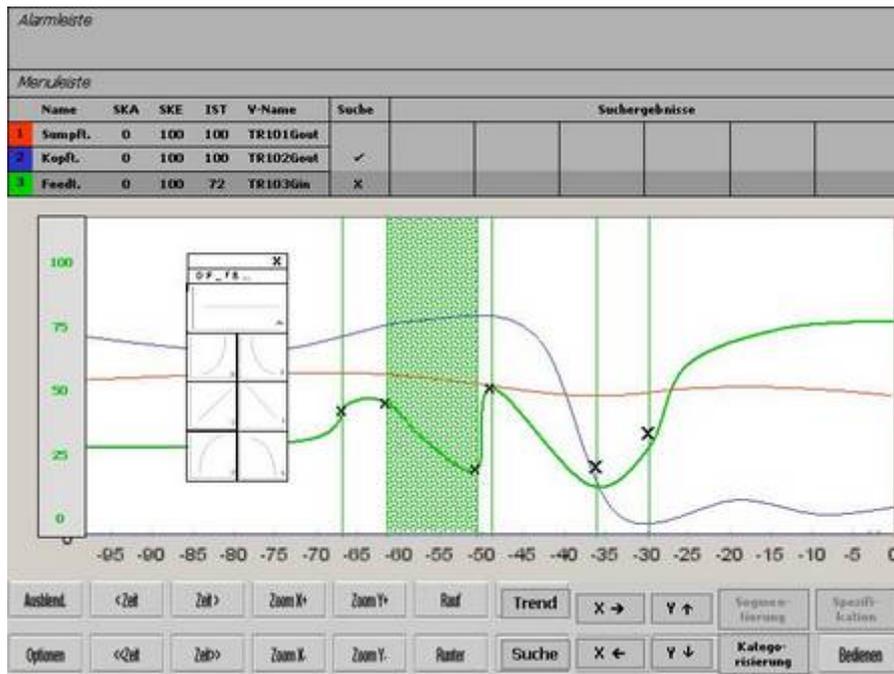


Abbildung 5 Interface Mockup für Trendbeschreibung



Abbildung 6 Interface Mockup für Ergebnispräsentation

B.1.3 Expertenevaluation

B.1.3.1 Zeitplan

	Zeitdauer	Material	Bemerkungen
Instruktion der Gruppenteilnehmer	10	Erläuterung von Ziel, Vorgehensweise, Arbeitsteilung, Aufgabenstellung, Usabilitykriterien und Mängelprotokoll	3 Teilnehmer pro Gruppe Arbeitsteilung: Benutzer, Beobachter, Protokollant Instruktionen liegen schriftlich vor. Jemand aus der Gruppe liest den allgemeinen Teil (Ziel, Vorgehensweise und Arbeitsteilung) laut vor. Danach überfliegen alle Teilnehmer Ihre individuellen Unterlagen leise. Benutzer: Aufgabenstellung und Usabilitykriterien Beobachter: Aufgabenstellung und Usabilitykriterien Protokollant: Usabilitykriterien und Mängelprotokoll
Bearbeitung der prototypischen Aufgaben & erstellen einer Mängelliste	30	Prototyp, Aufgabenstellung, Mängelprotokoll	
Problemdiskussion	25	Prototyp, Mängelprotokolle, Screenshots des Prototypen	alle Mitglieder aller Untergruppen
Bewertung der Mängelliste	15	Mängelprotokoll	in Einzelarbeit (es wird ein Drucker gebraucht, um die Evaluationsprotokolle rasch zu vervielfältigen)

B.1.3.2 Allgemeine Instruktionen

Die aufgeteilten Gruppen finden sich an den Rechnerarbeitsplätzen zusammen. Ein beliebiges Gruppenmitglied liest die folgenden Instruktionen vor.

Ziel der Heuristischen Evaluation ist eine Liste von Problemen der Gebrauchstauglichkeit. Diese wird in Kleingruppenarbeit erstellt. Für jedes Gruppenmitglied gibt es eine bestimmte Rolle.

Der Benutzer

- arbeitet eine Liste von Aufgaben ab. Diese dienen der exemplarischen und zielgerichteten Exploration des Prototypen.
- berichtet alle ihm wesentlich erscheinenden Probleme der Gebrauchstauglichkeit an die Gruppe.

Der Protokollant

- notiert und kategorisiert Probleme der Gebrauchstauglichkeit, welche bei der Bearbeitung durch den Benutzer auffallen.

Der Beobachter

- unterstützt den Benutzer bei der vollständigen und erfolgreichen Abarbeitung der Aufgabenliste sowie beim Erkennen von Problemen der Gebrauchstauglichkeit.
- hilft dem Protokollanten bei der effizienten und vollständigen Dokumentation der Usability-Probleme.
- beschleunigt somit den Evaluationsprozesses, da er die parallele Arbeit von Benutzer und Protokollant ermöglicht.

Bitte teilt nun die Rollen für Eure Gruppe auf!

Danach überfliegt bitte jeder kurz die Materialien, welche der jeweiligen Rolle für ihre Aufgabenbearbeitung benötigt.

Benutzer: Interface-Screenshot, Aufgabenstellung und Usabilitykriterien

Beobachter: Interface-Screenshot, Aufgabenstellung und Usabilitykriterien

Protokollant: Interface-Screenshot, Usabilitykriterien und Mängelprotokoll

Beginnt sodann mit der Bearbeitung der Aufgabenliste.

Bitte geht dabei zügig voran. Wichtig ist eine möglichst prägnante Beschreibung der Usabilityprobleme. Die Qualität der protokollierten Mängel ist wichtiger als deren Quantität.

Ihr werdet auf schwere und leichte Aufgaben treffen. Falls ihr eine Aufgabe nicht erfolgreich bearbeiten könnt, meldet euch bitte bei mir.

Bitte verwendet nicht zu viel Zeit in die Diskussion der Usability-Kriterien. Die endgültige Kategorisierung wird zu einem späteren Zeitpunkt in individueller Arbeit erstellt.

B.1.3.3 Aufgabenstellungen für Bediener

Aufgabentyp	Nr.	Aufgabe	Wenn Aufgabe nicht vollständig erfüllbar, ...
Start	1	Bitte starte den Prototyp über die Datei "demonstrator" auf dem Desktop. Lade aus der Datei "datenbank" die Datenpunkte 1001-2000 der Tabelle 3.	kontaktiere Moderator!
Auswahl und Navigation	2	Verändere das Kurvenbild derart, dass nur noch die Kurven der Temperatur "Temp_°C" und des Level "Level_l" zu sehen sind.	kontaktiere Moderator!
	3	Verschiebe die Kurven entlang der Zeitachse, so dass der Zeitbereich zwischen 00:05:00 bis 00:22:25 zu sehen ist.	gehe zu Nr. 4 !
	4	Zoom und verschiebe die Levelkurve entlang der Werteachse (Ordinate), so dass der Wertebereich zwischen 140 und 540 Litern zu sehen ist.	gehe zu Nr. 5 !
Trendbeschreibung	5	Gehe in den "Trend"-Modus und setze für die Levelkurve drei Dir sinnvoll erscheinende Episodengrenzen zwischen 00:08:00 und 00:21:00.	kontaktiere Moderator!
	6	Wähle für die einzelnen Episoden drei Dir sinnvoll erscheinende Trendlets.	gehe zu Nr. 7 !
	7	Verschiebe den Schnittpunkt letzten Episodengrenze mit der Levelkurve nach oben, sodass der neue Wert dieses Punktes ca. 60 Einheiten höher liegt als der bisherige.	gehe zu Nr. 8 !
	8	Verschiebe die Episodengrenze zwischen der ersten und zweiten Episode auf den Zeitpunkt 00:14:00.	gehe zu Nr. 9 !
	9	Lösche die letzte Episode.	gehe zu Nr. 10 !
	10	Mache den Löschvorgang rückgängig.	gehe zu Nr. 11 !
	11	Lösche die erste Episode.	gehe zu Nr. 12 !
	12	Beschreibe für die Temperaturkurve zwischen 00:08:00 und 00:18:00 Uhr einen Trend bestehend aus zwei Episoden mit den Trendlets E & C.	kontaktiere Moderator!
	13	Wechsel in den Expertenmodus und setze den zulässigen Episodenabstand zwischen beiden Episoden auf 80%.	gehe zu Nr. 14 !
	14	Setze die zulässigen Längen- und Höhenabweichungen beider Episoden auf 80% und die Krümmung der gekrümmten Episoden auf 20%.	gehe zu Nr. 15 !
	15	Erhöhe die Priorität für das Kriterium "Längenabweichung" auf 150%	gehe zu Nr. 16 !
Ergebnispräsentation	16	Starte die Suche und warte maximal 10 Sekunden auf das Erscheinen von Werten in dem Bereich der "Ergebniskurvenauswahl"	kontaktiere Moderator!
	17	Blende die Kurven aller Variablen aus Ergebnis 2 ein.	gehe zu Nr. 18 !
	18	Blende die Kurven aus Ergebniskurvenset 2 aus und die Kurven für Temperatur und Level aus dem Ergebnis 1 ein.	gehe zu Nr. 19 !
	19	Blende alle Kurven außer der Temperatur-Referenzkurve aus.	gehe zu Nr. 20 !
	20	Blende zusätzlich die beiden Temperaturkurven der Ergebnisse 1 & 2 ein.	gehe zu Nr. 21 !
	21	Lasse dir den Bereich der Ergebniskurve 2 anzeigen, für den die Suchmaschine einen Trend gefunden hat, der mit dem von Dir beschriebenen Trend vergleichbar ist.	kontaktiere Moderator!
	22	Vergleiche dieses Ergebnis mit dem von Dir beschriebenen Trend.	kontaktiere Moderator!
	23	Verteile alle drei Kurven über vertikal über das gesamte Kurvenfeld	gehe zu Nr. 24 !
freies Explorieren	24	Sind die 30 Minuten noch nicht verstrichen, exploriere bitte frei, versuche die wesentlichen Merkmale des Nutzerdialogs zu verstehen und notiere weitere Probleme der Gebrauchstauglichkeit.	kontaktiere Moderator!

B.1.3.4 Integration verschiedener Kategoriensysteme für Kriterien der Gebrauchstauglichkeit

Nr.	Kategorie	Erläuterung	Quelle
1	Sichtbarkeit des Systemstatus	Das System soll die Benutzer ständig darüber informieren, was geschieht, und zwar durch eine angemessene Rückmeldung in einem vernünftigen zeitlichen Rahmen.	Nielsen
	<i>Erkennbarkeit des Systemzustandes</i>	<i>Status, Ursächlichkeit und Prognose aktueller Rechenprozesse.</i>	<i>Huss</i>
2	Übereinstimmung zwischen dem System und der realen Welt	Das System sollte die Sprache des Benutzers sprechen, und zwar nicht mit systemorientierter Terminologie, sondern mit Worten, Phrasen und Konzepten, die den Benutzern vertraut sind. Dabei soll die natürliche und logische Reihenfolge eingehalten werden.	Nielsen
	Erwartungskonformität	Ein Dialog ist erwartungskonform, wenn er konsistent ist und den Merkmalen des Benutzers entspricht, z. B. seinen Kenntnissen aus dem Arbeitsgebiet, seiner Ausbildung und seiner Erfahrungen sowie den allgemein anerkannten Konventionen.	ISO 9241-10
	<i>externe Konsistenz</i>	<i>Systemkonformität mit Normen und Konventionen eines durchschnittlichen Nutzerinterface (Sprache, Anordnung, Dialogkonzepte)</i>	<i>Huss</i>
3	Benutzerkontrolle und -freiheit	Benutzer wählen Systemfunktionen oft fälschlicherweise aus und benötigen einen „Notausgang“, um den unerwünschten Zustand wieder zu verlassen. Dazu dienen Undo- und Redo-Funktionen.	Nielsen
	Fehlertoleranz	Ein Dialog ist fehlertolerant, wenn das beabsichtigte Arbeitsergebn trotz erkennbar fehlerhafter Eingaben entweder mit keinem oder mit minimalem Korrekturaufwand seitens des Benutzers erreicht werden kann.	ISO 9241-10
	<i>Fehlertoleranz</i>	<i>Systemstabilität und Rückführbarkeit (z.B. Funktion „Rückgängig“)</i>	<i>Huss</i>
4	Konsistenz und Standards	Benutzer sollten sich nicht fragen müssen, ob verschiedene Begriffe oder Aktionen dasselbe bedeuten. Deshalb sind Konventionen einzuhalten.	Nielsen
	<i>interne Konsistenz</i>	<i>einheitliche interne Verwendung von Sprache, Zuständen und Aktionen</i>	<i>Huss</i>
5	Fehlerverhütung	Noch besser als gute Fehlermeldungen ist ein sorgfältiges Design, das Fehler verhütet.	Nielsen
	<i>Fehlerprävention</i>	<i>Vorbeugen von Nutzerfehlern z.B. in Form von Warnhinweisen oder Eingabebeschränkungen</i>	<i>Huss</i>
6	Wiedererkennen, statt sich erinnern	Objekte, Optionen und Aktionen sollten sichtbar sein. Die Benutzer sollten sich nicht an Informationen aus einem früheren Teil des Dialogs mit dem System erinnern müssen. Instruktionen sollen sichtbar oder leicht auffindbar sein.	Nielsen
	<i>Gedächtnisstützen</i>	<i>Erkennen statt Erinnern von Objekten, Aktionen und Optionen</i>	<i>Huss</i>
7	Flexibilität & Effizienz	Häufig auftretende Aktionen sollten vom Benutzer angepasst werden können, um Fortgeschrittenen eine schnellere Bedienung zu erlauben.	Nielsen
	Individualisierbarkeit	Ein Dialog ist individualisierbar, wenn das Dialogsystem Anpassungen an die Erfordernisse der Arbeitsaufgabe sowie an die individuellen Fähigkeiten und Vorlieben des Benutzers zulässt.	ISO 9241-10
	<i>Adaptierbarkeit</i>	<i>Anpassungsmöglichkeiten in Abhängigkeit von Nutzerfertigkeiten, z.B. in Form von Tastenkombinationen, Parametrisierung, Oberflächenanpassungen, etc.</i>	<i>Huss</i>

8	Ästhetik und minimalistisches Design	Dialoge sollten keine irrelevanten Informationen enthalten, da die Informationen um die Aufmerksamkeit des Benutzers konkurrieren.	Nielsen
	<i>Informationsdarstellung</i>	<i>Effizienz der Informationsdarstellung (irrelevante oder extrem selten zu gebrauchende Informationen sollten den Nutzerdialog nicht stören)</i>	<i>Huss</i>
9	Hilfe beim Erkennen, Diagnostizieren und Beheben von Fehlern	Fehlermeldungen sollten in natürlicher Sprache ausgedrückt werden (keine Fehlercodes), präzise das Problem beschreiben und konstruktiv eine Lösung vorschlagen.	Nielsen
	<i>Fehlermeldung</i>	<i>Vollständigkeit, Verständlichkeit und Konstruktivität von Fehlermeldung</i>	<i>Huss</i>
10	Hilfe und Dokumentation	Jeder Information der Hilfe oder Dokumentation sollte leicht zu finden sein, auf die Aufgabe abgestimmt sein und die konkreten Schritte zur Lösung auflisten. Außerdem sollte sie nicht zu lang sein.	Nielsen
	Für hier vorgelegten Prototypen nicht relevant.		
11	Verfügbarkeit	Störungsfreiheit, konsistente Antwortzeiten proportional zur Aufgabenkomplexität	Evadis
	<i>Verfügbarkeit</i>	<i>funktionale Korrektheit, Antwortzeiten</i>	<i>Huss</i>
12	Nützlichkeit	Angemessene Funktionalität der Software für Aufgabenerfüllung	Evadis
	Für hier vorgelegten Prototypen nicht relevant.		
13	Komfort	Unterstützungsangebote ohne unnötige Zusatzbelastung	Evadis
	<i>Dialogkomfort</i>	<i>Effizienz der Dialogschritte</i>	<i>Huss</i>
14	Übersichtlichkeit	Darstellung und Anordnung von Bildschirminformationen	Evadis
	<i>Übersichtlichkeit</i>	<i>Übersichtlichkeit der Darstellung (Effizienz bei der Suche nach Eingabe- und Ausgabefeldern)</i>	<i>Huss</i>
15	Selbstbeschreibungsfähigkeit	Ein Dialog ist selbstbeschreibungsfähig, wenn jeder einzelne Dialogschritt durch Rückmeldungen des Dialogsystems unmittelbar verständlich ist oder dem Benutzer auf Anfrage erklärt wird.	ISO 9241-10
	<i>Selbstbeschreibungsfähigkeit</i>	<i>Explorationseffizienz</i>	<i>Huss</i>
16	Lernförderlichkeit	Ein Dialog ist lernförderlich, wenn er den Benutzer beim Erlernen des Dialogsystems unterstützt und anleitet.	ISO 9241-10
	Für hier vorgelegten Prototypen nicht relevant.		
17	Steuerbarkeit	Ein Dialog ist steuerbar, wenn der Benutzer in der Lage ist, den Dialogablauf zu starten sowie seine Richtung und Geschwindigkeit zu beeinflussen, bis das Ziel erreicht ist.	ISO 9241-10
	Für hier vorgelegten Prototypen nicht relevant.		

C Systembewertung

C.1 Untersuchungsmaterial

C.1.1 Testsets und Testscenarien

C.1.1.1 Testset 1

1. AEFA

```
range_1 = (0.6 : 0.2 : 1.4);
range_2 = (0.6 : 0.2 : 1.4);
for index_1 = range_1
    for index_2 = range_2
        T = wxalphabet( ...
            wxnoise(0.05, ...
                L (5,0), ...
                E (index_1,index_2,0.8,0.8,0.6,0.6), ...
                F (index_2,index_1,0.8,0.8,0.6,0.6), ...
                L (5,0) ...
            ));
    end
end
```

2. ABA

```
range_1 = (1.0 : 0.4 : 2.2);
range_2 = (1.0 : 0.4 : 2.2);
for index_1 = range_1
    for index_2 = range_2
        T = wxalphabet( ...
            wxnoise(0.05, ...
                L (5,0), ...
                L (index_1,index_2), ...
                L (5,0) ...
            ));
    end
end
```

C.1.1.2 Testset 2

1. AFDA

```
range_1 = (0.5 : 0.5 : 1.5);
range_2 = (0.2 : 0.2 : 0.8);
for index_1 = range_1
    for index_2 = range_2
        T = wxalphabet( ...
            wxnoise(0.1, ...
                L (3,0), ...
                F (index_1,(2-index_1),index_2,(1-index_2),0.6,0.6), ...
                D ((2-index_1),index_1,(1-index_2),index_2,0.6,0.6), ...
                L (3,0) ...
            ));
    end
end
```

2. AFDA

```
range_1 = (0.5 : 0.5 : 1.5);
range_2 = (0.5 : 0.5 : 1.5);
for index_1 = range_1
    for index_2 = range_2
        T = wxalphabet( ...
            wxnoise(0.1, ...
                L (3,0), ...
                F (index_1,index_2,0.8,0.8,0.6,0.6), ...
                D (index_1,index_2,0.8,0.8,0.6,0.6), ...
                L (3,0) ...
            ));
    end
end
```

3. AFDA

```
range_1 = (0.5 : 0.5 : 1.5);
range_2 = (0.5 : 0.5 : 1.5);
for index_1 = range_1
    for index_2 = range_2
        T = wxalphabet( ...
            wxnoise(0.1, ...
                L (3,0), ...
                F (1,index_1,0.8,0.8,0.6,0.6), ...
                D (1,index_2,0.8,0.8,0.6,0.6), ...
                L (3,0) ...
            ));
    end
end
```

4. AFBCDA

```
range_1 = (0.1 : 0.2 : 0.5);
range_2 = (0.1 : 0.2 : 0.5);
for index_1 = range_1
    for index_2 = range_2
        T = wxalphabet( ...
            wxnoise(0.1, ...
                L (3,0), ...
                F (1,1,0.8,0.8,0.6,0.6), ...
                L (index_1,index_2), ...
                L (index_1,-index_2), ...
                D (1,1,0.8,0.8,0.6,0.6), ...
                L (3,0) ...
            ));
    end
end
```

5. AEFDGDA

```
range_1 = (0.3: 0.3 : 0.9);
range_2 = (0.3: 0.3 : 0.9);
for index_1 = range_1
    for index_2 = range_2
        T = wxalphabet( ...
            wxnoise(0.1, ...
                L (3,0), ...
                E (index_1,index_1,0.8,0.8,0.6,0.6), ...
                F (1,1,NaN,0.8,0.6,0.6), ...
                D (1,1,0.8,0.8,0.6,0.6), ...
                G (index_2,index_2,NaN,0.8,0.6,0.6), ...
                D (index_2,index_2,NaN,0.8,0.6,0.6), ...
                L (3,0) ...
            ));
    end
end
```

```

        ));
    end
end

```

C.1.1.3 Testset 3

1. AGDEFA

```

range_1 = (0.5 : 0.5 : 1.5);
range_2 = (0.5 : 0.5 : 1.5);
for index_1 = range_1
    for index_2 = range_2
        T = wxalphabet( ...
            wxnoise(0.1, ...
                L (3,0), ...
                G (index_1,1,1,0.8,1,1), ...
                D (index_1,1,NaN,1,0.6,0.6), ...
                E (1,index_2,NaN,0.8,0.6,0.6), ...
                F (1,index_2,NaN,1,1,1), ...
                L (3,0) ...
            ));
    end
end

```

2. AGDAEFA

```

range_1 = (0.5 : 0.5 : 1.5);
range_2 = (0.0 : 0.5 : 1.0);
for index_1 = range_1
    for index_2 = range_2
        T = wxalphabet( ...
            wxnoise(0.1, ...
                L (3,0), ...
                G ((2-index_1),1,1,0.8,1,1), ...
                D (index_1,1,NaN,1,0.6,0.6), ...
                L (index_2,0), ...
                E (index_1,1,NaN,0.8,0.6,0.6), ...
                F ((2-index_1),1,NaN,1,1,1), ...
                L (3,0) ...
            ));
    end
end

```

3. AGBCDEBCFA

```

range_1 = (0.5 : 0.5 : 1.5);
range_2 = (0.2 : 0.2 : 0.8);
for index_1 = range_1
    for index_2 = range_2
        T = wxalphabet( ...
            wxnoise(0.1, ...
                L (3,0), ...
                G (index_1,1,1,0.8,1,1), ...
                L (0.1,index_2),...
                L (0.1,(0.2-index_2-0.4)), ...
                D (index_1,1,NaN,1,0.6,0.6), ...
                E (index_1,1,NaN,0.8,0.6,0.6), ...
                L (0.1,(index_2-0.2)), ...
                L (0.1,(-index_2)),...
                F (index_1,1,NaN,1,1,1), ...
                L (3,0) ...
            ));
    end
end

```

4. AEFGBDCBEBFCFGDA

```

range_1 = (0.5 : 0.5 : 1.5);
range_2 = (0.2 : 0.2 : 0.8);
for index_1 = range_1
    for index_2 = range_2
        T = wxalphabet( ...
            wxnoise(0.1, ...
                L (3,0), ...
                E (1,0.3,1,0.8,0.6,0.6), ...
                F (1,0.3,NaN,0.8,0.6,0.6), ...
                G (1,index_1,NaN,0.8,1,1), ...
                L (0.1,(-index_2)),...
                L (0.1,(0.2+index_2)), ...
                D (1,1,NaN,1,0.6,0.6), ...
                L (0.1,(-1.5*index_2)),...
                L (0.1,(1.5*index_2)), ...
                E (1,1,NaN,0.8,0.6,0.6), ...
                L (0.1,(index_2-0.2)), ...
                L (0.1,(-index_2)),...
                F (1,index_1,NaN,1,1,1), ...
                G (1,0.3,1,0.8,0.6,0.6), ...
                D (1,0.3,NaN,0.8,0.6,0.6), ...
                L (3,0) ...
            ));
    end
end

```

5. ADCBEBCA

```

range_1 = (0.5 : 0.5 : 1.5);
range_2 = (0.0 : 0.1 : 0.4);
for index_1 = range_1
    for index_2 = range_2
        T = wxalphabet( ...
            wxnoise(0.1, ...
                L (3,0), ...
                D (index_1,(2-index_1),NaN,1,0.6,0.6), ...
                L (0.1,(-1.5*index_2)),...
                L (0.1,(1.5*index_2)), ...
                E ((index_1-index_2),(2-index_1+index_2),NaN,0.8,0.6,0.6), ...
                L (0.1,(index_2-0.2)), ...
                L (0.1,(-index_2)),...
                L (3,0) ...
            ));
    end
end

```

6. AEFGBDCBGBBCDEFDA

```

range_1 = (0.5 : 0.5 : 1.5);
range_2 = (0.3 : 0.3 : 0.9);
for index_1 = range_1
    for index_2 = range_2
        T = wxalphabet( ...
            wxnoise(0.1, ...
                L (3,0), ...
                E (1,0.3,1,0.8,0.6,0.6), ...
                F (1,0.3,NaN,0.8,0.6,0.6), ...
                G (1,index_1,NaN,0.8,1,1), ...
                L (0.1,(-index_2)),...
                L (0.1,(0.2+index_2)), ...
                D (1,1,NaN,1,0.6,0.6), ...
                L (0.1,(-1.5*index_2)),...
            ));
    end
end

```

```

        L (0.1,(1.5*index_2)), ...
        G (1,index_1,NaN,0.8,0.6,0.6), ...
        L (0.1,(index_2-0.2)), ...
        L (0.1,(-index_2)),...
        D (1,index_1,NaN,1,1,1), ...
        E (1,0.3,1,0.8,0.6,0.6), ...
        F (1,0.3,NaN,0.8,0.6,0.6), ...
        L (3,0) ...
    );
end
end

```

C.1.1.4 Testset 4

AFGDEFA

```

range_1 = (0.2: 0.2 : 1.6);
range_2 = (0.02: 0.08 : 0.18);
for index_1 = range_1
    for index_2 = range_2
        T = wxalphabet( ...
            wxnoise(index_2, ...
                L (3,0), ...
                F (0.5,0.5,NaN,0.6,0.6,0.6), ...
                G (0.5,index_1,NaN,0.6,0.6,0.6), ...
                D (0.5,index_1,NaN,0.6,0.6,0.6), ...
                E (0.5,(index_1/2),NaN,0.6,0.6,0.6), ...
                F (0.5,(index_1/2),NaN,0.6,0.6,0.6), ...
                L (3,0) ...
            );
    end
end
end

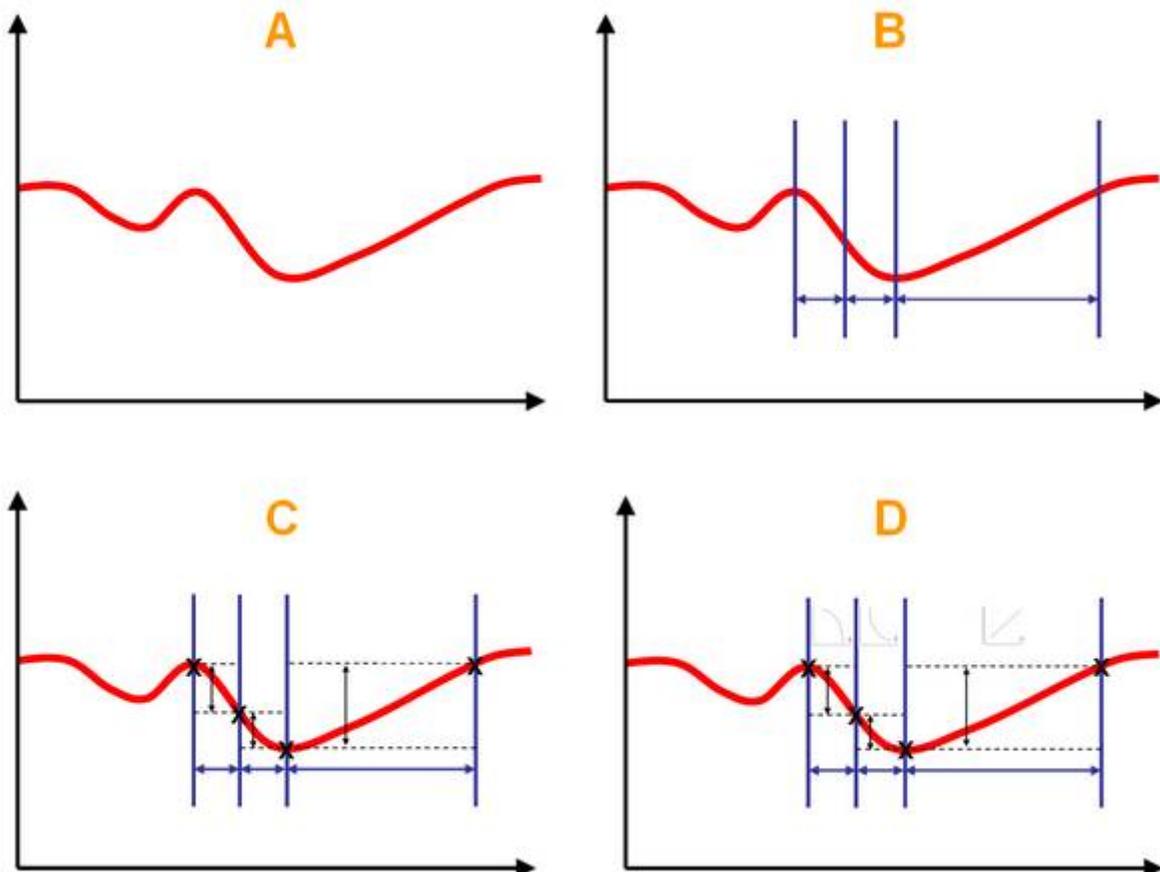
```

C.2 Durchführung

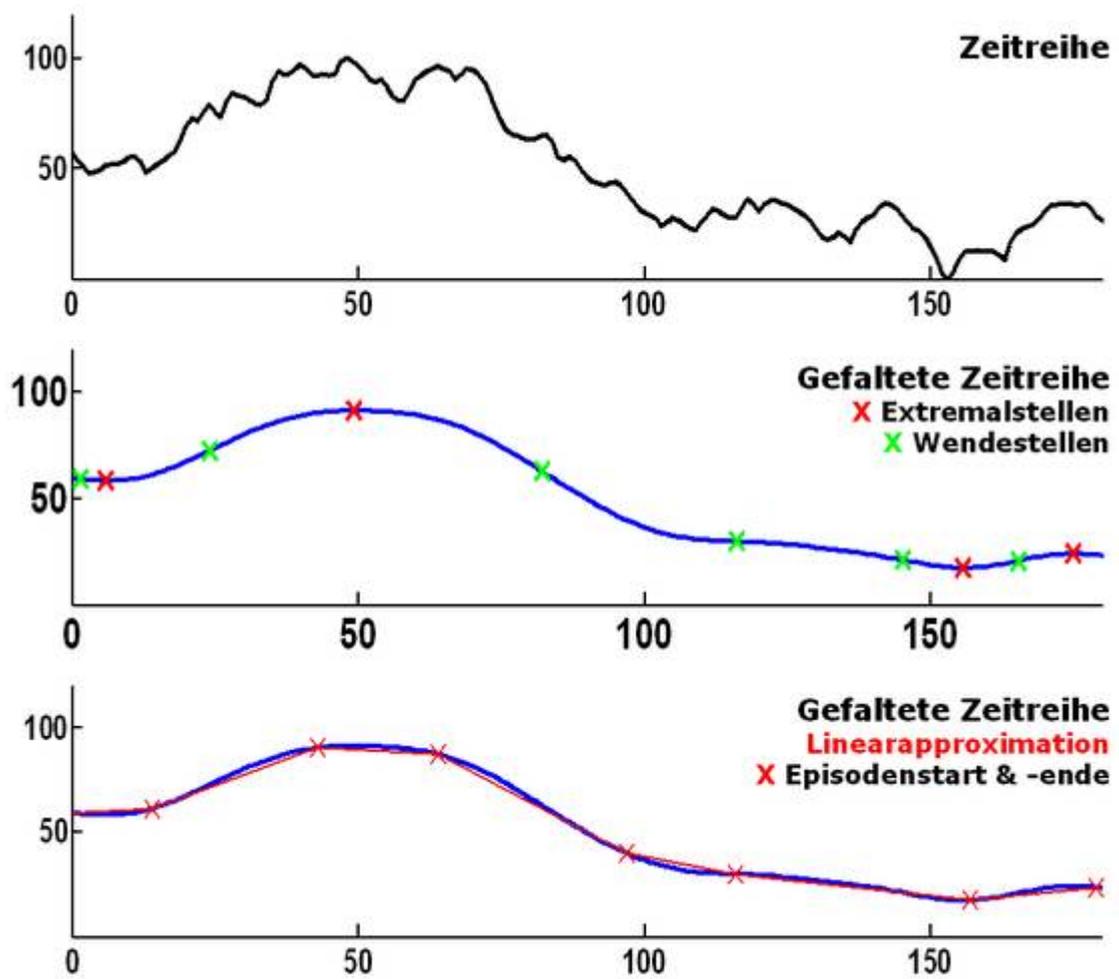
C.2.1 „Google-Analogie“

Akteur	Trend basierte Suche	Text basierte Suche
Mensch	Kurvenlesen & Mustererkennung	Textlesen & Inhaltsanalyse
Mensch	Trendbeschreibung	Schlagwortdefinition
Technik	Trendsuche	Schlagwortsuche
Technik	Ausgabe von Ergebniskurven	Ausgabe von Ergebnistexten
Mensch	Recherche in Kurvenausschnitten	Recherche in Textauswahl

C.2.2 Veranschaulichung der Trendbeschreibung



C.2.3 Erläuterung der Suchindexe



C.2.4 Allgemeine Erläuterungen des Prototypen

Prototyp: Kurvennavigation

- Verschieben entlang der Zeitachse.
- Zoomen entlang der Zeitachse.
- Verschieben entlang der Werteachse.
- Zoomen entlang der Werteachse.
- Orientierung der Zoomfunktionen an Zentralachse des Kurvenbildes

Prototyp: Trendbeschreibung

- Fenster für die Trendbeschreibung
 - über den Button „Trend“ aufgerufen
 - lässt sich komplett über den Bildschirm verschieben.
- Kurvenbild wird damit sensitiv gegenüber Mausklicks
 - d.h. jeder Klick mit der linken Maustaste fügt eine vertikale Linie an der Stelle des Mausursors ein.
 - Linie kennzeichnet eine zeitliche Grenze (Start oder Ende) einer zum Trend gehörenden Episode.
 - Funktion „Rückgängig“ und „Löschen“
- Default-Automatik für Start- oder Endzeitpunkt
 - Schnittpunkt der Grenze mit der Referenzkurve
- interaktive Symbole im „Trend“-Fenster
 - Grenze
 - „X Rauf“ und „X Runter“
 - „<X“ und „X>“
 - Episode
 - Trendlet-Zuordnung
- Vollständiger Trend -> Start mit „Suche“
- Toleranzen (Länge & Höhe) ansagen!

Ergebniskurvenauswahl

- Rangliste der Suchergebnisse
- Werte in den Zellen
 - errechnete Ähnlichkeitsindikatoren
 - liegen zwischen Null und Eins
- berechnet auf Basis der Längenabweichung, Höhenabweichung und dem Episodenabstand
- Auswahl der Ergebniskurven durch Einfach- und Zweifach-Mausklicks
- Ergebniskurvennavigation
 - Die Referenzkurve und alle Ergebniskurven werden gemeinsam vergrößert und verkleinert.
 - Die Referenzkurve und alle Ergebniskurven werden unabhängig voneinander verschoben.

Fragen?

C.2.5 Aufgabeninstruktionen

Die Instruktionen waren für alle Tests identisch. Exemplarisch wird die Instruktion für Test 1 dargestellt. Die Instruktion erfolgt anhand des folgenden Hand-outs.

Datenbank-Auswahl für Test 1

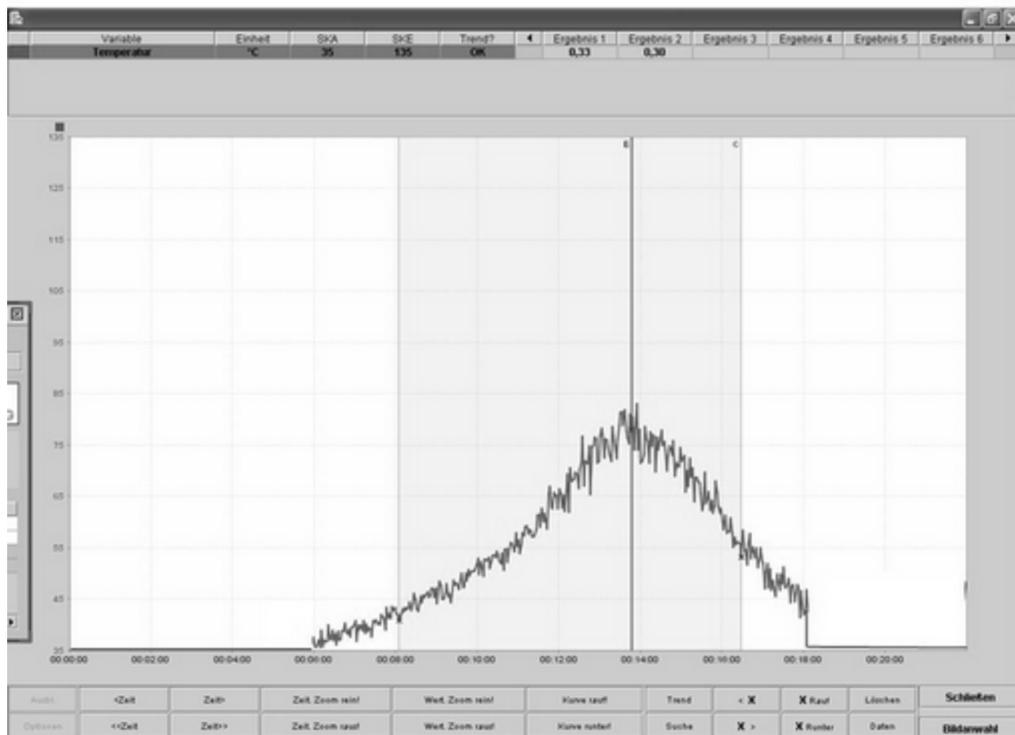
Datenbank	Szenario	Datenausschnitt
sz6	Page20	1-1000

1. Beschreibung eines Trends mit einer Episode

- Beliebiger Länge
- Beliebiger Höhe
- Beliebiger Dynamik

Einzige Einschränkung:

- maximaler Zeitbereich (Vergleichsbereich) für Trendbeschreibung: 03:30:00 bis 06:30:00



Beispiel für Vergleichsbereich von 00:06:00 bis 00:18:00 (inklusive Trendbeschreibung)

2. Starten der Suche!

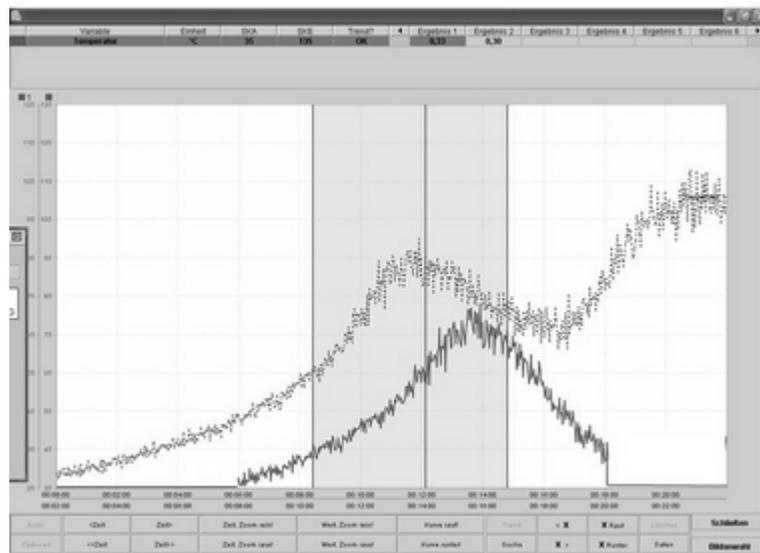
- Wenn keine Ergebnisse, dann screen shot erstellen, speichern und Trendbeschreibung ändern!
- Heuristiken:
 - Episoden zu lang oder zu kurz
 - Dynamiken der Episoden nicht adäquat
- Wenn nach zwei weiteren Versuchen keine Ergebnisse, dann bitte beim Versuchsleiter melden.

3. Die Ergebnisse

- werden in Rangfolge dargeboten.

Ergebnis 1	Ergebnis 2	Ergebnis 3	Ergebnis 4	Ergebnis 5	Ergebnis 6
0,98	0,96	0,95	0,94	0,93	0,90

- zeigen ganze Zeitreihen (Kurven)
- Im Navigationsmodus (Doppelklick auf Ähnlichkeitsindikator des entsprechenden Ergebnis) wird der Bereich markiert (Ergebnistrend), welcher laut Suchmaschine dem beschriebenen Trend ähnelt.



Beispiel für Präsentation von Ergebniskurve inklusive Ergebnistrend.

4. Ergebnisbewertung

- Überprüfung der Ähnlichkeit zwischen den Ergebniskurven und der Referenzkurve über den gesamten Vergleichsbereich.
 - Ergebniskurve darf vor dem Vergleich maximal eine Rasterlinie nach links oder rechts verschoben werden.
- Überprüfung der Rangfolge:
 - Erscheint die Rangfolge der Ergebnisse nach Überprüfung der Ergebnisse über den Vergleichsbereich plausibel?
 - Wenn nein, welche andere Rangfolge müssten die Ergebnisse erhalten?
 - neue Rangfolge in Tabelle eintragen!

Ergebnisreihe 1

Suchergebnisse	1	2	3	4	5	6
Ähnlichkeitskorrektur						

5. Dokumentation der Ergebnisse
 - screen shot von allen Ergebnissen inklusive Trendbeschreibung mit „Alt & Druck“
 - Einfügen in Bildverarbeitungsprogramm auf rechtem Bildschirm
 - Speichern des Bildes in dem Ergebnisordner der Versuchsperson

6. Erweiterung des Trends um eine weitere Episode:
 - beliebiger Länge
 - beliebiger Höhe
 - beliebiger Dynamik
 - Vorhandene Episoden kann bei Bedarf verändert werden, hinsichtlich:
 - des Ortes
 - der Länge
 - der Höhe
 - der Dynamik
 - Einzige Einschränkung:
 - maximaler Zeitbereich (Vergleichsbereich) für Trendbeschreibung: 03:30:00 bis 06:30:00

7. Starten der Suche!
 - Wenn keine Ergebnisse, dann screen shot erstellen, speichern und Trendbeschreibung ändern!
 - Heuristiken:
 - Episoden zu lang oder zu kurz
 - Dynamiken der Episoden nicht adäquat
 - Wenn nach zwei weiteren Versuchen keine Ergebnisse, dann bitte beim Versuchsleiter melden.

8. Wiederholung der Schritte 4-5

9. Ergebnisbewertung

Ergebnisreihe 2

Suchergebnisse	1	2	3	4	5	6
Ähnlichkeitskorrektur						

10. Dokumentation (screen shot)

11. Bewertung des Progress

- Bewerten Sie, ob sich die Suchergebnisse im Allgemeinen verbessert haben:
 - Unabhängig von der Plausibilität der Rangfolge.
 - Unabhängig von der absoluten Zahl an Ergebnissen
 - Vergleichen Sie nur so viele Ergebnisse, wie in beiden Rangfolgen vorhanden sind.
 - Nehmen Sie den screen shot der Ergebnisreihe 1 zu Hilfe.

Die Ergebnisse der neuen Ergebnisreihe ähneln der Referenzkurve im Vergleichsbereich mehr als ...	stimmt	im Gegenteil	weis nicht
... die Ergebniskurven aus der Ergebnisreihe 1?			