Institut für Visualisierung und Interaktive Systeme

Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Bachelorarbeit Nr. 294

Weiterentwicklung einer Anfragevisualisierung für Ereignissequenzen

Tina Tremel

Studiengang:	Informatik
Prüfer/in:	Prof. Dr. Thomas Ertl
Betreuer/in:	M.Sc. Robert Krüger, Dipl.-Inf. Florian Haag
Beginn am:	28. Januar 2016
Beendet am:	29. Juli 2016
CR-Nummer:	H.1.2, H.2.3, H.3.3, H.5.2, I.5.3

Kurzfassung

Bewegungsdaten werden heutzutage durch die weite Verbreitung von mobilen Geräten mit GPS-Technik und anderen Positionsbestimmungssystemen immer häufiger und in größeren Datenmengen und feinerer Auflösung erzeugt. In diesem Bereich können visuelle Anfragesprachen verwendet werden, die eine übersichtliche und einfache Methode darstellen Anfragen an die Datensammlungen zu stellen und diese Daten so untersuchbar machen. Die meisten visuellen Anfragesprachen für Ereignissequenzen, wie VESPa, werden dabei als Graph repräsentiert. In dieser Arbeit wird eine Weiterentwicklung von VESPa vorgestellt, die visuell mehr Bezug zu der zeitlichen Komponente von Ereignissen vorweist und mehr Funktionalität bietet. In Rahmen der Arbeit wurde desweiteren ein einfaches Analysesystem entwickelt, welches in Kombination mit der visuellen Anfrage einen einfachen Analyse-Loop realisiert. Der Prototyp der Anfragesprache und des Systems wurden in einer Benutzerstudie und Anwendungsfällen evaluiert und konnte dort einen positiven Eindruck bei den Nutzern hinterlassen.

Abstract

Movement data, made possible by todays mobile devices with GPS-technology and other location tracking systems, is gathered more and more often in increasingly large quantities and detail. In this area, visual query languages can be used to provide an easy and clear method of accessing these data collections and to make them explorable. Most visual query languages, like VESPa, are represented as graphs. This thesis will present an improved version of VESPa, providing more visual connection to the time aspect of events, as well as increased functionality. Additionally, a simple analysis system has been developed that is able to form a simple analysis loop in combination with the visual queries. The prototype for the query language and system has been evaluated in a user study and case studies, resulting in a positive impression on the user's side.

Inhaltsverzeichnis

Abbildungsverzeichnis

Verzeichnis der Listings

Verzeichnis der Algorithmen

1. Einleitung

In vielen Bereichen werden heutzutage Bewegungsdaten in großen Mengen aufgezeichnet, die für verschiedene Zwecke analysiert werden müssen. Dabei kann es sich zum Beispiel um Geodaten von Personen oder Tieren, die sich in Gebäuden, Städten oder im freien Gelände bewegen, handeln. Verwendungen finden sich somit zum Beispiel im Bereich der Vehrkehrsüberwachung und -planung oder im Geomarketing, wo das Verhalten der Kunden anhand von Bewegungsdaten untersucht wird um Angebotsplatzierungen und das Marketing zu verbessern. Die rohen Bewegungsdaten werden zu diesem Zweck oft noch zusätzlich semantisch mit weiteren relevanten Informationen angereichert die helfen Rückschlüsse über das Bewegungsverhalten zu ziehen. Bei der Analyse der Daten komm es dabei darauf an aus diesen großen Datenmengen Strukturen und Informationen zu gewinnen. Das manuelle Untersuchen der rohen Datensammlungen ist dafür allerdings nicht geeignet und somit werden Methoden benötigt die helfen diese Daten zu verarbeiten.

Bei gezielten Suchanfragen an solche Datensammlungen werden häufig nicht visuelle Notationen wie SQL verwendet die zwar den meisten Fachleuten bekannt sind, aber bereits bei kleineren Suchanfragen in komplexen und unübersichtlichen Anweisungen enden. Die Lesbarkeit ist besonders bei langen und stark verschachtelten Anweisungen nur bedingt gegeben und selbst Experten in diesem Bereich sind nicht in der Lage die Bedeutung hinter einer Anfrage auf den ersten Blick zu erfassen. Fehler können sich so leicht einschleichen und nur sehr zeitaufwendig erkannt und beseitigt werden. Speziell im Bereich von Visual Analysics wo es darum geht Daten interaktiv mit optischen Hilfsmitteln zu untersuchen, eigenen sich viele textuelle Anfragesprachen nicht, da das Erstellen einer Anfrage sehr viel Zeit beansprucht und Wissen über die interne Speicherung der Daten voraussetzt, das nicht unbedingt gegeben und auch nicht für die Analyse erforderlich sein sollte. Für Experten im Bereich der Bewegungsdatenanalyse wird somit ein tiefes Verständnis für die Datenhaltung vorausgesetzt, das nur für die Formalisierung der Anfrage aber nicht für die Analyse selbst nötig gewesen wäre.

Ein Lösungsansatz für die Analyse von großen Datensammlungen kann der Einsatz einer geeigneten Visualisierung sein. Bei der Untersuchung von großen Datenmengen kann diese helfen, Muster und Zusammenhänge in den Daten sichtbar zu machen und so dabei unterstützen, Schlüsse und Informationen aus den Daten zu gewinnen. Ohne eine Visualisierung sind rohe Datenmengen sonst bereits ab einer relativ keinen Menge kaum noch zu überschauen und können ohne Hilfsmittel nicht oder nur schwerlich untersucht werden. Im Bereich der Informationsvisualisierung gibt es aus diesen Gründen zahlreiche Möglichkeiten wie Daten visualisiert werden können um die enthalten Informationen in den Vordergrund zu stellen.

Für unterschiedliche Datentypen sind unterschiedliche Visualisierungen verfügbar die speziell auf die enthaltenen Informationen angepasst sind. Insgesamt soll dies die Arbeit mit den Daten verständlicher, einfacher und übersichtlicher gestalten und ist teilweise sogar zwingend notwendig um überhaupt mit den Daten arbeiten zu können.

Diese Vorteile können auch bei Anfragen an Datenmengen genutzt werden. Visuelle Anfragesprachen nutzen im Gegensatz zu SQL und anderen textuellen Anfragesprachen visuelle Komponenten um Anfragen zu modellieren. Diese sind im Allgemeinen leichter zu verstehen, setzen weniger Vorwissen voraus und können somit auch von Laien verwendet werden. bei der Gestaltung der einzelnen grafischen Elemente nicht zu viel Vorwissen vorausgesetzt werden. Bei der Symbolik sollte somit auf bereits bekannte und allgemeingültige Abstraktionen zurückgegriffen werden. Dabei kann man annehmen, dass eine Nutzer einer solchen Anfragesprache zumindestens eine Grundverständnis für die Problemstellung mitbringt, dies aber nicht in eine formale textuelle Anfrage umsetzen kann. Die Aufgabe besteht somit darin dem Nutzer dabei zu helfen diese Gedanken in ein formale Anfrage an das System umzusetzen. Ein Beispiel für eine solche Anfragenvisualisierung an Bewegungsdatensätze ist VESPa [HKE16]. Haag et al. visualisieren hierbei die Anfrage in Form eines einfachen Graph der ein Muster bestehend aus Ereignissen als Knoten darstellt. Dabei liegt der Fokus auf Suche nach Mengen von Sequenzen die sich in bestimmten Punkten räumlich begegnen sollen. Dabei kann die Notation nicht nur dazu dienen Anfragen die Datensammlung zu stellen, sondern stellt auch durch die Visualisierung der Anfragekomponenten eine allgemeine Beschreibung der Daten dar die durch dieses Muster gefunden werden können. Allerdings gibt es in diesem Konzept noch Raum für Verbesserungen.

Ziel dieser Arbeit ist die Weiterentwicklung der Anfragesprache VESPa, die sich speziell mit Anfragen von Bewegungsdaten beschäftigt. Dabei soll die Funktionalität erweitert werden und einige der Verständnisproblem die in der Benutzerstudie zum Vorschein kamen verbessert werden. Die Sprache soll dabei gezielt an Nutzer richten, die sich mit der Bewegungsdatenanalyse beschäftigen aber nur wenige Grundlagen im IT-Bereich vorweisen können. Insgesamt muss die Sprach somit intuitiv sein und wenig technisches Vorwissen voraussetzen. Desweiteren wird um die Anfragesprache eine System gebaut, welches sowohl in der Lage ist die Ergebnisse der Anfrage zu visualisieren als auch eine Möglichkeit bietet direkt aus der Ergebnisdarstellung Anfragen zu generieren. Das neue Konzept soll im Rahmen dieser Arbeit in einer Benutzerstudie evaluiert werden.

Gliederung

Die Arbeit ist in folgender Weise gegliedert:

Kapitel 2 – Verwandte Arbeiten: In den verwandten Arbeiten werden andere wissenschaftliche Arbeiten vorgestellt, die sich ebenfalls mit visuellen Anfragesprachen und der

Visualisierung von Ereignissen beschäftigen und teilweise in dieser Arbeit Verwendung gefunden haben.

Kapitel 3 – Grundlagen: Grundlegende Voraussetzungen für das Verständnis der Problemstellung sowie eine Einführung in die Daten, die Definition von Ereignissen und Ereignissequenzen, sowie Grundlagen der Informationsvisualisierung und Visual Analytics werden hier eingeführt.

Kapitel 4 – Visuelle Anfragesprache: In diesem Kapitel liegt der Fokus auf der neuen visuellen Anfragesprache. Die verwendete Notation, sowie die Semantik der Sprache wird beschrieben.

Kapitel 5 – Einbettung im Analysesystem: Als Ergänzung zur Anfragesprache wird hier ein System vorgestellt, mit dem es möglich ist die Ergebnisse der Anfrage zu visualisieren und auf Basis dieser Visualisierung neue Anfragen interaktiv zu stellen.

Kapitel 6 – Implementierung: In der Implementierung werden technische Details der Umsetzung, sowie die Datenmodelle welche hinter der visuellen Anfragesprache stehen vorgestellt.

Kapitel 7 – Evaluation: Um die neue visuelle Anfragesprache auf im Hinblick auf Verständlichkeit und Benutzbarkeit zu untersuchen, wurde eine Benutzerstudie durchgeführt, die in diesem Kapitel vorgestellt wird. Des Weiteren werden hier anhand von Anwendungsfällen das Konzept des Analysesystems erklärt.

Kapitel 8 – Diskussion: In einer Diskussion wird das System kritisch beurteilt, Vor- und Nachteile herausgearbeitet und die Grenzen der Anfragesprache und des Systems aufgezeigt.

Kapitel 9 – Zusammenfassung und Ausblick Das letzte Kapitel enthält eine Zusammenfassung der Arbeit und einen Ausblick auf das mögliche weitere Vorgehen, sowie ein Fazit zur neuen Anfragesprache.

2. Verwandte Arbeiten

In den verwandten Arbeiten wird zunächst ein Überblick über grundlegende Konzepte von Visualisierungen zeit-orientierter Daten, visuelle Anfragesprachen im Allgemeinen und für Ereignissequenzen im Speziellen gegeben.

2.1. Visualisierung von Zeitdaten und Ereignissen

Visualisierungen bestimmter Datentypen versuchen, bestimmte Merkmale der Daten herauszuarbeiten. Ereignisse, wie sie in dieser Arbeit verwendet werden, haben immer einen zeitlichen Bezug, der für die Darstellung der Ereignisse verwendet werden kann. Ereignisse können dabei sowohl auf einen einzelnen Zeitpunkt oder über einen ganzen Zeitintervall definiert werden. Bei der Visualisierung kann diese zeitliche Komponente zum Beispiel auf einer Zeitachse an der entsprechenden Position zusammen in einer ereignisspezifischen Darstellung eingetragen werden [CK91]. Die zeitlichen Beziehungen einzelner Ereignisse werden dadurch sichtbar gemacht.

Um einen Überblick über den zeitlichen Verlauf von Dokumenten oder anderen kategorischen Daten zu bekommen, können ThemeRiver nach Harve et al. [HHN00] verwendet werden. Hierbei werden die Daten thematisch zusammengefasst und im zeitlichen Verlauf dargestellt. Dieser Ansatz kann allerdings auch dazu verwendet werden, Vorkommen von Ereignissen im zeitlichen Verlauf zu repräsentieren. In dieser Form wird das ThemeRiver Konzept auch in dieser Arbeit verwendet.

PlanningLines nach Aigner et al. [AMTB05] arbeiten im Gegensatz zu den anderen vorgestellten Arbeiten mit Unsicherheiten bei der Einordnung in eine Zeitachse. PlanningLines sind glyph-basierte Visualisierungen von Aufgaben in Zeit- oder Projektplänen, die Planungsunsicherheiten enthalten. Aigner et al. erweitert für diesen Zweck die Timeline um weitere Komponenten, sodass sowohl die minimale und maximale Zeitdauer wie auch der frühste und späteste Zeitpunkt für das Beginnen und Beenden repräsentiert werden. Paint Strips [CC03] beschäftigen sich ebenfalls mit der Visualisierung von unbestimmten Zeitintervallen und deren zeitlichen Beziehungen. In dieser Arbeit werden diese Konzepte als Vorbild für die Modellierung von Unsicherheiten bei der Visualisierung der zeitlichen Ereignisaspekte verwendet. Im Gegensatz zu den vorgestellten Konzepten wird in dieser Arbeit die Dauer einer Timeline allerdings nicht in der Darstellung kodiert, sondern nur die zeitlichen Beziehungungen mehrerer Zeitspannen von Ereignissen sind von Bedeutung.

Andere Visualisierungen, die speziell für das Darstellen zeitlich-räumlicher Daten existieren, stellen die Zeitangabe als zusätzliche Dimension zu den zweidimensionalen räumlichen Koordinaten in einem dreidimensionalen Raum, dem Space-Time-Cube [Kra03], dar. Ein Space-Time-Path stellt in dieser Umgebung eine dreidimensionale Trajektorie dar, mit der sich darstellen lässt wie sich ein Objekt über einen Zeitraum bewegt hat. Zeitliche und räumliche Nähe zwischen Ereignissen sind direkt in dieser Darstellung sichtbar.

2.2. Visuelle Anfragesprachen

Visuelle Anfragesprachen für Ereignissequenzen basieren meist auf graphischen Notationen. Eine Auswahl von verschiedenen Verfahren ist im Folgenden zu sehen.

Das FilterFlow Konzept repräsentieren logische Aussagen durch einen gerichteten Graphen [Shn98]. Die Knoten des Graphen stellen dabei verschiedene Filter dar, durch die gedanklich ein Fluss von Daten geleitet wird. Die Filter lassen dabei immer nur den gefilterten Teil der Daten passieren. Sequentielle Anordnungen von Knoten entsprecht somit logischen UND-Verknüpfungen der Filter-Operationen und Verzweigungen können als ODER-Verknüpfungen gesehen werden. Dieses Konzept unterscheidet sich von dem Konzept dieser Arbeit, dadurch, dass die Knoten nicht für Ereignisse sondern für Filteroperationen stehen. Die Visualisierung repräsentiert somit die Operationen, die verwendet wurden um zu einem bestimmten Ergebnis zu kommen. Im Gegensatz dazu ist die Anfragesprache dieser Arbeit eine Repräsentation der eigentlichen Daten.

Ähnlich dazu basieren (s|q)ueries [ZDFD15] auf der Idee, visuelle Anfragen nach Vorbild von regulären Ausdrücken aufzubauen. Hier geht es dabei speziell um Ereignissequenzen, die in einem Graph durch die Positionierung der Ereignisse in einer Anordnung von links nach rechts gebracht werden. Operatoren regulärer Ausdrücke wie Alternativen, Verkettungen und Kleenesche Sterne werden visuell durch die Ordnung und Verbindungen der Graphkomponenten ausgedrückt. Die Ergebnisse werden dabei direkt in die visuelle Repräsentation der Anfrage integriert und können dadurch untersucht werden. (s|q)ueries konzentriert sich dabei auf Sequenzen aus Punktereignissen, die keine zeitliche Ausdehnung besitzen und somit nur vor- oder nacheinander stattfinden können. In dieser Arbeit sollen allerdings auch Ereignisse mit einer gewissen zeitlichen Ausdehnung berücksichtigt werden können.

VESPa [HKE16] ist ebenfalls eine visuelle Anfragesprache, die sich mit dem Suchen von Ereignisabfolgen beschäftigt. Der Fokus dieses Konzepts liegt dabei bei der Suche nach Ereignissequenzen mehrerer unabhängiger Objekte in Bewegungsdaten. Hierzu wird ein Muster durch einen Graph ausgedrückt und Überschneidungen individueller Ereignisse unterschiedlicher Objekte werden zu einem Knoten in der Darstellung zusammengefasst. Als Erweiterung zu diesem Konzept wird in dieser Arbeit eine ähnliche Visualisierung von Anfragen ausgearbeitet, die die zeitliche Komponente von Ereignissen stärker repräsentiert.

3. Grundlagen

In diesem Kapitel werden zunächst einige Grundlagen, die für das Verständnis dieser Arbeit wichtig sind, eingeführt. Dazu gehören an erster Stelle die Daten, anhand derer in den folgenden Kapiteln die Struktur, die Bedeutung der Anfragesprache und des Systems erklärt werden. Dazu werden im Folgenden die verwendeten Datensammlungen, sowie die genaue Struktur und die Bestandteile der Daten selbst beschrieben. Des Weiteren wird ein Überblick über Visual Analystics und Informationsvisualisierung gegeben.

3.1. Informationsvisualisierung und Visual Analytics

Die Informationsvisualisierung umfasst viele Konzepte und Methoden mit deren Hilfe Informationen auf eine angemessene Art dargestellt werden können, um daraus Einsichten und Erkenntnisse zu ziehen. In diesem Zusammenhang gibt es viele unterschiedliche Visualisierungen, die jeweils unterschiedliche Aspekte der Daten beleuchten. Visualisierungen sind vor allem bei großen Datenmengen, wie sie auch in Bewegungsdaten anfallen, wichtig, um überhaupt Informationen daraus gewinnen zu können. Der Weg von den rohen Daten bis zur eigentlichen Visualisierung wird dabei oft durch die Visualisierungspipeline nach Card et al. [CMS99] beschrieben. Dabei werden die rohen Daten zuerst durch Filter und Aggregationen strukturiert, diese können danach auf visuelle Attribute abgebildet werden, die dann in der Visualisierung dargestellt werden. Auf alle Schritte der Pipeline kann der Nutzer durch Interaktion zugreifen und die angewendeten Tranformationen den Bedürfnissen anpassen. Shneiderman's Visual Information Seeking Mantra „Overview first, zoom & filter, then details-on-demand" kann dabei als Richtline für eine gute Visualisierung gesehen werden [Shn96].

Visual Analytics befasst sich dabei konkret mit der Idee mit visuellen Repräsentationen und durch Interaktion Lösungen für analytische Probleme zu finden. Der Visual Analytics Process [KKEM10] beschreibt hier das prinzipielle Vorgehen, wie durch Visualisierungen und Modelle Wissen aus Daten generiert werden kann. In einem iterativen Prozess können somit neue Erkenntnisse über die Daten gewonnen werden, die wieder in neuen Visualisierungen und Anpassungen des Modells enden können. Als Ziel der Analyse kann dabei sowohl das Explorieren der Daten, als auch das Überprüfen einer Hypothese angesetzt werden.

In dieser Arbeit werden Methoden der Informationsvisualisierung und Visual Analytics eingesetzt, um die Anfragesprache darzustellen und die Ergebnisse in einer angemessenen Weise

zu visualisieren. Speziell im Analysesystem werden verschiedene Konzepte wie Brushing-And-Linking, Selektion von Daten, Filter und Sortierungen angewandt, die es dem Nutzer ermöglichen, interessante Daten zu markieren, einzuschränken und auf verschiedenen miteinander verknüpften Ansichten zu betrachten.

3.2. Daten

Möchte man eine Anfrage an eine Datensammlung stellen, ist es zunächst wichtig zu wissen, um welche Daten es sich dabei konkret handelt. In dieser Arbeit geht es um Anfragen und Visualisierungen von Ereignissequenzen, und im Speziellen um Bewegungsdaten. Diese können heutzutage durch die weite Verbreitung von mobilen Geräten mit GPS-Technik auf eine einfache und präzise Art als Bewegungsprofile für viele verschiedene Objekte erstellt werden. Die rohe Datenmenge besteht hierbei aus einzelnen Messwerten für eine Reihe von Zeitpunkten, die aneinandergereiht eine Trajektorie und somit einen Pfad bilden. Um aus diesen einfachen Bewegungsmustern Ereignissequenzen zu generieren, müssen geographisch Orte festgelegt werden, die einen bestimmten Kontext besitzen und den Ort eines potenziellen Ereignisses repräsentieren. Diese Orte werden auch Points of Interest (POI) genannt. Wird nun bestimmt, zu welchen Zeiten sich ein Objekt innerhalb verschiedener POIs befindet, kann dies als Aufenthalt und somit im Kontext dieser Arbeit als Ereignis betrachtet werden. Weitere semantische Anreicherung wird meistens ebenfalls betrieben, um die erzeugten Daten weiter zu spezifizieren. Eine Sequenz in einem Bewegungsdatensatz kann dann zum Beispiel eine Person sein, für die bekannt ist wann und wie lange sie sich an bestimmten Orten aufgehalten hat.

In Abbildung 3.1 ist somit ein Beispiel für eine solche Ereignissequenz, besehend aus 5 Ereignissen, die jeweils eine gewisse zeitliche Ausdehnung besitzen, abgebildet. Das Objekt, welches diese Bewegung durchführt, ist der Besitzer der Sequenz. Die genaue Definition von Ereignissen für diese Arbeit wird im Folgenden gegeben.

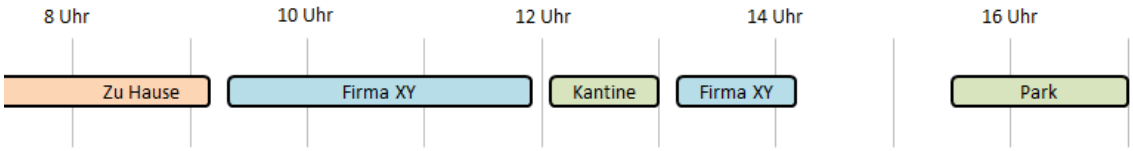


Abbildung 3.1.: Ein Ausschnitt einer einzelnen Sequenz einer Person aus einem Bewegungsdatensatz

3.2.1. Ereignisse

Nachdem im vorherigen Abschnitt eine allgemeine Beschreibung der Daten und konkreten Datensammlungen einen ersten Überblick über die Art der Daten gegeben hat, werden nun die

Bestandteile genauer definiert. Dabei fällt die Definition von einem Ereignis je nach Kontext unterschiedlich aus. Im Alltag versteht man unter einem Ereignis oder Event eine Veranstaltung, bei der sich mehrere Personen zu einer bestimmten Zeit an einem bestimmten Ort gleichzeitig aufhalten. Meistens ist damit ein Sinn verknüpft, wie zum Beispiel eine Feier oder ein Meeting. Eine andere Definition ist das Eintreffen eines Zustandes zu einem bestimmten Zeitpunkt.

Haag et al. definieren ein Ereignis im Kontext von VESPa als einen Ort zu einem bestimmten Zeitpunkt [HKE16]. Diese Definition wird in dieser Arbeit verallgemeinert, sodass nicht nur der Ort sondern auch der Kontext entscheidend ist. Für die Definition eines Ereignisses kann somit ein allgemeiner Identifikator gewählt werden, der bestimmt was ein Ereignis neben dem Ort und der Zeit zusätzlich definiert. In der re:publica-Datensammlung werden so zum Beispiel neben dem Raum auch die Vorträge zur Bestimmung und Abgrenzung von Ereignissen verwendet. Besucher die mehrere Vorträge am Stück hören, haben dadurch mehrere Einzelereignisse und nicht nur ein großes Ereignis. Der Kontext, der zusätzlich hinzugezogen wird, hängt dabei von den Daten ab. Auch VESPa reichert die Sequenzen mit weiterer Semantik an, sodass bei der Analyse der Daten mehr Informationen zu den Objekten bekannt sind und diese auf verschiedene Aspekte hin untersucht werden können.

Da Ereignisse mit einer zeitlichen Komponente versehen sind, kann man zeitliche Beziehungen zwischen Ereignissen bestimmen. Sie können sich unter Anderem zeitlich überschneiden oder vor- oder nacheinander stattfinden [All83]. In dieser Arbeit werden für einzelne Objekte ausschließlich nicht überlappende Ereignisse betrachtet. In Bewegungsdaten ist dies der Fall, wenn sich die Gebiete, die die Orte bestimmen, nicht überlappen und jedes Ereignis nur einem Kontext angehören kann. Ereignisse unterschiedlicher Objekte können allerdings zur gleichen Zeit und im gleichen Kontext an einem Ort stattfinden. In Bewegungsdaten entspricht dies einer potentiellen Begegnung der beteiligten Objekte. Formal entspricht das Ereignis somit einem Tupel bestehend aus der Zeit, dem Ort und dem Kontext. Ort und Kontext können dabei gemeinsam repräsentiert werden.

Ereignissequenzen sind somit Aneinanderreihungen von mehreren Ereignissen. Die Zeitkomponente sorgt dafür, dass eine Ordnung innerhalb der Ereignisse existiert, die eine sequentielle Anordnung ermöglicht. Betrachtet man zum Beispiel Bewegungsdaten von Personen, so ist die Anordnung einfacher Ereignisse klar durch die Zeitkomponente gegeben. Für jede Person kann eine einfache Ereignissequenz definiert werden, die aus einfachen Zeitabschnitten in Kombination mit der Position der Person entsteht.

3.2.2. re:publica

Die re:publica ist eine Konferenz, die unter anderem Themen wie digitale Gesellschaft, soziale Medien, Netzpolitik und Kultur umfasst [rep13]. Im Jahr 2013 wurde die Position der Konferenzbesucher von den Veranstaltern während der Dauer der gesamten Konferenz über die WLAN-HotSpots bestimmt. Dazu wurden die MAC-Adressen der Netzwerkkarten der mobilen Geräte der Besucher aufgezeichnet, die sich jeweils mit einem der WLAN-HotSpots verbunden

hatten. Diese Daten wurden anschließend anonymisiert über OpenDataCity [Ope13] online verfügbar gemacht.

Für den Zweck der Analyse wurden diese Rohdaten von Krüger et al. aufbereitet, in ein Datenmodell überführt und semantisch mit dem Konferenzplan, Lageplan und weiteren Informationen der Redner angereichert [KHH+15]. In diesem Zusammenhang entstanden mehrere Repräsentationen von Ereignissequenzen, die die Besucherströme, den Verlauf und die Reihenfolge der besuchten Veranstaltungen der Konferenz wiederspiegeln. Eine dieser Sequenzmengen ist der erste der beiden Datensätze, die in dieser Arbeit als Beispiel verwendet werden. Beim dem ersten Datensatz handelt es sich somit um reelle Bewegungsdaten innerhalb eines Gebäudekomplexes mit relativ grober räumlicher Auflösung. Mit diesen Daten können die Besucherströme der Konferenz untersucht werden.

3.2.3. VAST Challenge 2014

Der zweite Datensatz besteht erneut aus Bewegungsdaten von Menschen, allerdings handelt es sich dabei um Daten von Personen, die sich innerhalb einer Stadt bewegen. Im Gegensatz zum vorherigen Datensatz sind diese künstlich im Rahmen der VAST Challenge 2014 [Vis14] entstanden. In diesem fiktiven Datensatz ging es darum, ein Verbrechen, welches in einer fiktiven Stadt begangen wurde, aufzudecken. Konkret handelt es sich dabei um den Datensatz der Mini-Challenge 2, bei dem die Fahrzeuge der Mitarbeiter einer fiktiven ortsansässigen Firma während des Betriebs getrackt wurden und so Informationen über die Bewegungen der Mitarbeiter in der Stadt verfügbar sind. Des Weiteren sind die Kreditkarteninformationen sowie die Kreditkartenabrechnungen der Mitarbeiter verfügbar, sodass bekannt ist, welche Geldtransaktionen in der Stadt stattfinden. Auf Basis dieser Daten sollte das fiktive Verbrechen an mehreren Mitarbeiter im Rahmen der VAST Challenge aufgeklärt werden.

Krüger et al. [KHHE15] haben diese Daten aufbereitet um durch verschiedene visuelle Analyse Tools die Daten zu untersuchen und die Challenge zu lösen. Dabei ist auch die erste Version von VESPa [HKE16] entstanden. Somit lagen die Daten des zweiten Datensatzes bereits in der richtigen Form vor.

3.3. VESPa

VESPa [HKE16] wurde vom Institut für Visualisierung und interaktive Systeme der Universität Stuttgart entwickelt, um auf eine einfache intuitive Art Anfragen an Bewegungsdatensammlungen zu stellen. VESPa verwendet als Anfrage eine einfache visuelle Notation, mit der ein Muster von Ereignissequenzen definiert werden kann. In der Datensammlung sind dann Ereignissequenzen gesucht, die diesem Muster entsprechen. Dabei liegt der Fokus nicht auf der Suche nach einzelnen Sequenzen mit bestimmten Mustern, sondern auf einer Gruppe von Personen, deren Sequenzen sich zeitlich überlappende Ereignisse an bestimmten Orten haben

oder explizit unterschiedlich sein müssen. Ein einzelnes Ergebnis einer Anfrage besteht somit aus mehreren Personen und deren Ereignissequenzen.

Als Notation wird ein einfacher Graphen verwendet, der primär aus Knoten für Ereignisse und Transitionen, die die Bewegungen zwischen den Ereignissen darstellen, besteht. Sogenannte Akteure stellen eine Zugehörigkeit der Kanten mit Ereignissen dar. Alle Komponenten können dabei mit den Restriktionen genauer spezifiziert werden. In Abbildung 3.2 ist dieser Aufbau exemplarisch an einem Beispiel zu sehen.

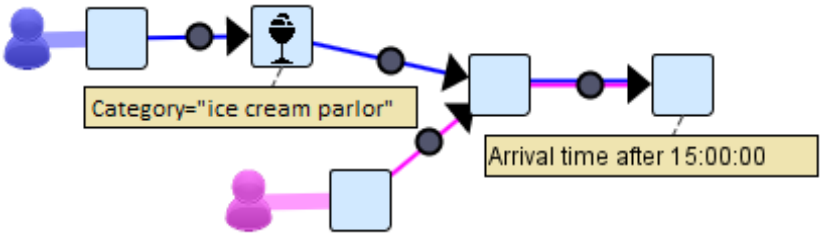


Abbildung 3.2.: Beispiel für 2 Personen, die sich nach individuellen Ereignissen treffen [HKE16].

Haag et al. haben dieses Konzept in einer Benutzerstudie mit mehreren Personen getestet und dort Schwierigkeiten bei der Interpretation der genauen Bedeutung hinter den Ereignissen beobachtet. Des Weiteren sind einige Fälle mit der Anfragesprache noch nicht abgedeckt, die durchaus bei den zugrundeliegenden Datensammlungen von Interesse sein können. In Ausblick stellten Haag et al. zum Beispiel Erweiterungen, die es ermöglichen, wiederholte Sequenzen oder Alternativen zu definieren. Auch im Hinblick auf die visuelle Darstellung der Komponenten wurden weitere Verbesserungsmöglichkeiten in Aussicht gestellt.

In dieser Arbeit wird eine Weiterentwicklung von VESPa vorgestellt, die die angesprochenen Probleme und Uneindeutigkeiten in der Visualisierung der Anfragesprache zu verbessern und das Konzept zu erweitern sucht. Insbesondere soll mehr Wert auf eine visuelle Unterstützung der Bedeutung der Komponenten gelegt werden, sodass die Nutzer schnelleren Zugang zu der Bedeutung der Komponenten bekommen. Dabei wird vor Allem der zeitliche Aspekt der Datensammlungen mehr in den Vordergrund gestellt und die Notation und somit auch die Funktionalität um ein paar Komponenten erweitert.

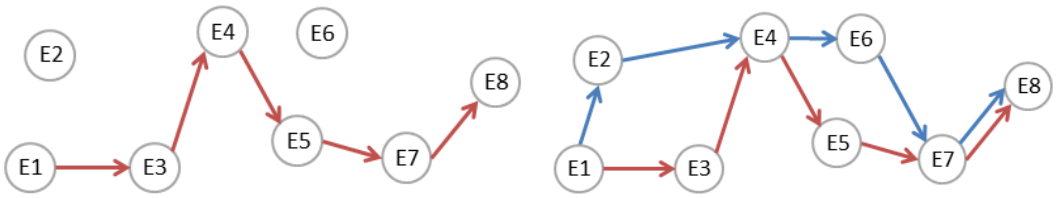
4. Visuelle Anfragesprache

Mit VESPa lassen sich Muster aus Ereignissequenzen definieren die dazu dienen können Anfragen an eine Datensammlung zu stellen. Wie bereits in der Einleitung angesprochen, verwendet VESPa für die Visualisierung der Sequenzen eine Darstellung als Graph aus Knoten, die Ereignisse repräsentieren, und Kanten, die die Sequenzen festlegen. Die neue Anfragesprache, die in diesem Kapitel vorgestellt wird, wird auch in der grundlegenden Struktur als Graph repräsentiert. Allerdings wird dabei die Darstellungsform des Graphen verändert, sodass die sequentiellen Eigenschaften der zugrundeliegenden Datensammlung besser repräsentiert und so dem Nutzer mehr Hilfestellungen bei der Interpretation gegeben werden.

Insgesamt soll die Anfragesprache, wie auch VESPa, intuitiv und leicht verständlich bleiben und von Personen verstanden werden können die nur grundlegende Kenntnisse im EDV-Bereich vorweisen können, sich aber trotzdem mit der Analyse von Bewegungsdaten beschäftigen wollen. In diesem Rahmen finden sowohl Weiterentwicklungen im Bereich der Symbolik als auch in der Semantik der Sprache statt, um diese der Intuition des Nutzers anzupassen. Dazu wird im Folgenden das Konzept und die Idee hinter der Anfragesprache vorgestellt und anhand von konkreten Beispielen aus der re:publica-Datensammlung erklärt.

4.1. Anfragegraph

Bei der Suche nach Ereignissequenzen in Datensammlungen ist es naheliegend, Anfragen als Graph zu modellieren. Ereignisse können als Knoten interpretiert werden, die durch gerichtete



(a) (b)

Abbildung 4.1.: Darstellung von Ereignissequenzen in Form eines Graphen. (a) zeigt eine einzelne Ereignissequenz. In (b) werden 2 Sequenzen dargestellt, die gleiche Ereignisse besitzen.

Kanten in eine bestimmte Reihenfolge gebracht werden können und somit eine Ereignissequenz bilden. Eine einzelne Sequenz kann als Abfolge von miteinander verbundenen Knoten repräsentiert werden und bildet somit einen zyklenfreien, gerichteten Graphen, der für jede eingehende Kante eines Knotens eine ausgehende Kante besitzt. In Abbildung 4.1a ist eine solche einfache Sequenz in einer Menge von Ereignissen eingezeichnet. Der jeweils erste und letzte Knoten markieren hierbei den Start beziehungsweise das Ende der Sequenz und haben entsprechend nur ein Kante. Dieses Konzept lässt sich problemlos durch den Einsatz von unterschiedlichen Kanten auch auf mehrere Sequenzen erweitern. Auf diese Weise lassen sich mehrere Ereignissequenzen gemeinsam in einen Graph einzeichnen wie in Abbildung 4.1b zu sehen ist. Hierbei können einzelne Ereignisse auch Teil mehrerer Sequenzen sein.

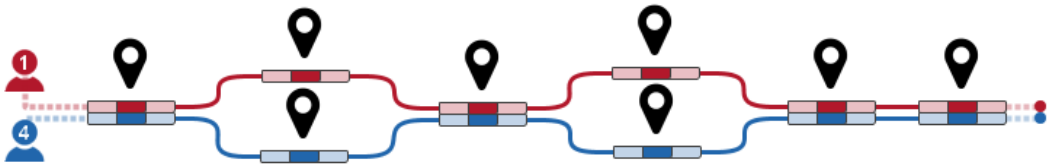


Abbildung 4.2.: Visualisierung der gleichen Ereignissequenzen wie in Abbildung 4.1b in sequentieller Form

Um die sequentiellen Eigenschaften der Daten auch visuell zu unterstützen verwendet die neue Darstellung, im Gegensatz zu VESPa, keinen normalen Graphen bei dem Kanten an beliebigen Stellen ein- und ausgehen können, sondern orientiert sich an der Darstellung von Syntaxdiagrammen, da diese ebenfalls sequentielle Abfolgen repräsentieren. Das besondere daran ist die Positionierung der Kanten, die sequentiell angeordnet sind. Dies bringt die Vorteile mit sich, dass das Lesen der Graphen intuitiver wird und mehr Bezug zu den visualisierten Daten vorweist. In Abbildung 4.2 sind die gleichen Sequenzen wie auch in Abbildung 4.1b zu sehen, diese sind hier allerdings geordnet und entsprechen optisch einem Fluss von links nach rechts.

Die komplette Anfrage setzt sich somit aus mehreren Komponenten zusammen. Die Basis bilden die Pfadkomponenten, die im vorherigen Beispiel zu sehen sind. Sie definieren die eigentliche Ereignissequenz und bilden das Grundgerüst für weitere Komponenten. Dazu gehören der Akteur, die Pfadkanten und die Ereignisse selbst.

4.2. Pfadkomponenten

Das Grundgerüst der Anfragevisualisierung bilden die Komponenten die die Pfade definieren und somit die Sequenzen festlegen. Ein einfacher Pfad besteht aus genau einem Akteur und mehreren Ereignissen und Pfadkanten (siehe Abbildung 4.3). Um einen Überblick über die

visuellen Komponenten zu bekommen werden diese im Folgenden zuerst kurz beschrieben und erst anschließend genauer betrachtet.

Abbildung 4.3.: Die Pfadkomponenten im Überblick.

Akteure Der Akteur markiert den Startpunkt einer Ereignissequenz im Graphen von dem aus die Kanten die den Pfad definieren ausgehen. In der Anfrage dient der Akteur als Platzhalter für jeden möglichen Besitzer oder eine konkrete Person für die die definierte Ereignissequenz in der Datensammlung enthalten ist. Alle Pfade und Ereignisse, die zu einem Akteur gehören, werden in der Anfrage genau auf eine Person abgebildet. Pfade mit unterschiedlichen Akteuren können in der Anfrage nicht auf die selbe Person abgebildet werden.

Ereignis Die Ereignis-Komponente ist das Kernelement der Anfragesprache und vereinigt mehrere unterschiedliche Aspekte miteinander. Ein Ereignis wird als ein Ort in einem bestimmten Kontext in einer festgelegten Zeitspanne definiert. Der räumliche Aspekt wird durch einen PlaceMarker markiert und symbolisiert den Ort des Ereignisses. Der zweite Bestandteil ist der Zeitraum. Dieser wird in der Darstellung durch eine Zeitleiste, die Timeline, visualisiert, die einem Aufenthalt eines Akteurs in einem Ereignis entspricht.

Timeline Die Timeline-Komponente ist ein Bestandteil eines Ereignisses. Die Farbe der Timeline gibt dabei an, auf welchen Akteur sich die Timeline jeweils bezieht. Da es auch möglich ist, dass mehrere Akteure sich zur gleichen Zeit an einem Ereignis befinden, können auch mehrere Timelines für die unterschiedlichen Akteure unter dem Ereignissymbol angehängt werden.

Pfadkanten Pfadkanten verbinden Akteure mit Ereignissen oder Ereignisse untereinander. Eine Pfadkante gehört immer genau zu einem Akteur und repräsentiert die eigentliche Ereignissequenz des Akteurs. Die Länge und Krümmung der Kanten haben dabei keine Bedeutung. Zudem repräsentieren Kanten zwischen zwei Ereignissen nicht zwangsläufig auch eine Veränderung des Ortes. Finden 2 Ereignisse nacheinander am selben Ort statt, so markiert die Pfadkante lediglich semantischen Übergang zwischen den beiden Ereignissen.

An einem konkreten Beispiel aus dem re:publica-Datensatz wird nun das Zusammenspiel der Pfadkomponenten genauer betrachtet. Dieses Beispiel ist in Abbildung 4.4 zu sehen. Es gibt 4 Akteure und somit wird nach einer Personengruppe von insgesamt 4 Leuten gesucht. Umgangssprachlich formuliert ergibt sich aus dem Bild folgende Anfrage an das System:

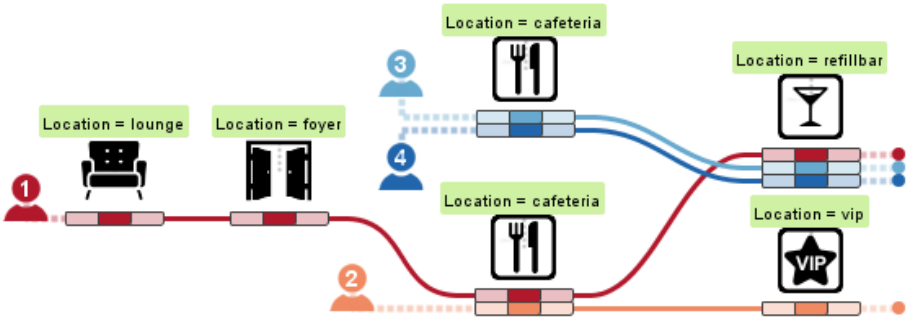


Abbildung 4.4.: Beispiel für eine Anfrage an den re:publica-Datensatz.

„ Gibt es 4 Personen, wovon einer (rot) zuerst in der Lounge und im Foyer war und sich anschließend mit einem anderen (orange) der 4 Personen in der Cafeteria getroffen hat und am Ende mit den beiden anderen (hellblau und dunkelblau) in der Refillbar war, während der 4. (orange) in den VIP-Bereich gegangen ist?“

Der Akteur übernimmt somit jeweils die Rolle einer Person in der Anfrage. Es sollte dabei klar sein, dass ein Akteur in einem Ergebnis der Anfrage somit mit genau einem Besitzer einer Ereignissequenz in Verbindung gebracht werden kann. Ereignisse entsprechen hierbei den Aufenthalten der Personen an einem bestimmten Ort zu einer bestimmten Zeit. Wenn sich die Pfade von mehreren Akteuren schneiden, wie es zum Beispiel bei der Refillbar oder der Cafeteria der Fall ist, bedeutet dies für die Anfrage dass Ereignissequenzen, die diesem Muster entsprechen, zu bestimmten Zeitpunkten Ereignisse besitzen, welche am gleichen Ort und zu überlappenden Zeiten stattfinden. Treffen sich die Pfade an einem Ereignis nicht, so bedeutet dies im Umkehrschluss, dass kein Treffen zwischen den Akteuren stattfindet.

4.2.1. Die Bedeutung der Timeline

Durch die Einführung der Timeline als Ergänzung zum Ereignissymbol wird eines der Probleme von VESPa direkt adressiert. In der Benutzerstudie wurde beobachet, dass manche Teilnehmer Ereignisse teilweise nur mit Orten in Verbindung gebracht haben und ihnen der zeitliche Kontext der Ereignisse nicht bewusst war. Aus diesem Grund wurde die neue Timeline-Komponente eingeführt um dafür zu sorgen, dass das Ereignis nun auch visuell mit einer Zeitkomponente versehen ist und den Benutzern somit eine Hilfestellung bei der Interpretation gegeben wird. Dieser Effekt soll auch von der sequentiellen Anordnung der Ereignisse unterstützt werden, da die Pfadkanten direkt mit den Timeline-Komponenten verbunden sind und so die Idee eines zeitlichen Verlaufs und somit einer definierten Reihenfolge besser repräsentiert wird.

Allerdings soll die Timeline nicht nur ein einfaches Symbol in der Anfragesprache darstellen sondern kann auch dazu verwendet werden um zeitliche Informationen zu den Ereignissen

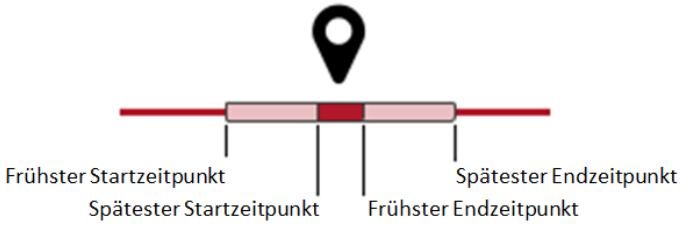


Abbildung 4.5.: Übersicht über den Aufbaue einer Timeline.

kompakt und in einer angemessen Form darzustellen. Entsprechend wurde für die Visualisierung der Timeline ein Konzept ausgearbeitet, welches in einer stark vereinfachten Form auf den Planning Lines nach [AMTB05] basiert und somit eine bestimmte Zeitspanne repräsentiert.

Genau wie beim Planen von Aufgaben möchte man auch beim Anfragen an Datensammlungen mit Unsicherheiten, was Start- und Endzeitpunkte betrifft, arbeiten können. Aus diesem Grund ist am Anfang und Ende der Timeline ein Bereich definiert in dem der tatsächliche Start- oder Endpunkt der Zeitspanne liegen kann (siehe Abbildung 4.5). Allerdings soll bei den Timelines in dieser Arbeit im Gegensatz zu der PlanningLine Definition nach Aigner et al. der Länge der einzelnen Abschnitte keine Bedeutung zugemessen werden. Entscheidend bei diesem Ansatz ist nur, ob es jeweils einen Bereich der Unsicherheit einer anderen Timeline des Ereignisses gibt, der vor oder nach dem unsicheren Bereich der eigenen Timeline endet oder anfängt. So können mit der Timeline-Repräsentation lokale Beziehungen zwischen den Aufenthalten einzelner Akteure definiert und mit verschiedenen Positionierungen der Balkenbegrenzungen unterschiedliche Aussagen modelliert werden. Diese Repräsentation bildet die Basis einiger Restriktionen, die in einem späteren Abschnitt dieses Kapitels weiter definiert werden.

Die Bedeutung der Ereignisse

Als Kernelement der Anfrage spielt das Ereignis in ihr eine große Rolle. Haag et al. definieren ein Ereignis als einen Ort an einem bestimmten Zeitpunkt. Auch in der neuen Anfragesprache wird diese grundlegende Festlegung beibehalten. Allerdings kann dabei noch der Kontext miteinbezogen werden. So können mehrere aufeinander folgende Ereignisse einer Sequenz auch am selben Ort stattfinden, wenn sich an diesem Ort der Kontext geändert hat. Ein Beispiel dafür wäre ein Konferenzsaal bei dem eine Vortragsreihe stattfindet. Hier kann jeder Vortrag als eigenes Ereignis betrachtet werden, obwohl keine räumliche Veränderung stattgefunden hat.

Eine wichtige Änderung im Vergleich zu VESPa ist der andere Umgang mit teilweisen Überschneidungen von Ereignissen mehrerer Akteure. Denkt man an ein Treffen einer Personengruppe bei dem Leute vielleicht schon gehen, bevor die letzten eingetroffen sind oder zwischendurch nicht anwesend sind, aber später wieder zur Gruppe stoßen, ist es nachvollziehbar, dass dies als ein Ereignis gewertet werden kann. Ein solches Beispiel ist auch in

Abbildungen 5.2 zu sehen. Betrachtet man diese Daten allerdings ohne die Semantik, geht dies so nicht hervor. Hier sind nur teilweise Überschneidungen von Teilmengen der Personen aufspürbar und nicht die Personengruppe als ganzes. In den neuen Anfragesprache möchte man dennoch diesen Fall abfragen können. Dazu wird das Ereignis allerdings in mehrere einzelne Teilereignisse zerlegt die für die Treffen von Teilmengen der Personengruppen stehen.

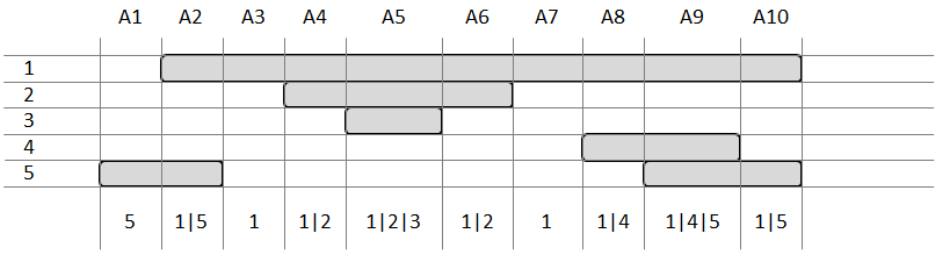


Abbildung 4.6.: Die grauen Balken im Diagramm zeigen jeweils an, in welchen Zeiträumen die 5 Personen anwesend waren. Den Zeitstrahl auf dem diese Balken liegen kann man in einzelne Abschnitte (A1 – A10) unterteilen, sodass jeder Abschnitt einem Zeitraum entspricht in dem eine bestimmte Teilmenge der Personen anwesend ist.

Aufgrund der Definition der Ereignisse, wäre es in VESPa nicht möglich gewesen die Muster mit dieser Bedeutung zu modellieren, da ein Ereignis eines Akteurs immer nur exakt auf ein Ereignis in der Anfrage abgebildet werden kann. Somit könnte die erste Person in diesem Beispiel in VESPa nur entweder mit Person 5, mit Person 2 und 3 oder mit 4 und 5 ein gemeinsames Ereignis haben. Ein Muster, das dieser Sequenz nahe kommt, kann somit nicht definiert werden. In der neuen Anfragesprache wird dies durch die neue Definition allerdings möglich.

4.3. Alternativpfade und Wiederholungsschleifen

Eine erste Erweiterung im Verleich zu VESPa ist die Möglichkeit Alternativpfade und Wiederholungsschleifen in der Anfrage formulieren zu können. Auf diese Art können mehrere alternative Szenarien zusammen in einer Anfrage formuliert und gemeinsam abgefragt werden. Zudem sind Wiederholungsschleifen nützlich um wiederholte Sequenzen kompakt und einfach modellieren zu können.

4.3.1. Verzweigungen

Wie bereits erwähnt können Verzweigungen eingesetzt werden, um alternative Pfade in der Anfrage zu definieren. Dazu wird eine neue visuelle Komponente, die Verzweigung, eingeführt die, wie in Abbildung 4.14 zu sehen, durch einen Kreis mit dem Schriftzug „OR“ dargestellt wird.

Alle Pfade im Muster, die in einen solchen Verzweigungsknoten eingehen, können ab dieser Stelle auf mehrere unterschiedliche Arten und Weisen fortgesetzt werden. Die Anzahl der Alternativen, die durch eine Verzweigung entstehen, wird dabei durch die Anzahl der Ausgänge angezeigt. In diesem Prototyp der Anfragesprache sind jeweils bis zu vier Alternativen je Verzweigungsknoten möglich. In Abbildung 4.7a wird so zum Beispiel ein Muster definiert, welches sowohl Teilsequenzen findet, die bei den beiden Akteure gemeinsam in der Refillbar als auch in der Cafeteria enden. Da immer alle Akteure einer Alternative folgen, ist somit nicht möglich, dass ein Akteur in die Refillbar geht und ein Akteur in die Cafeteria.

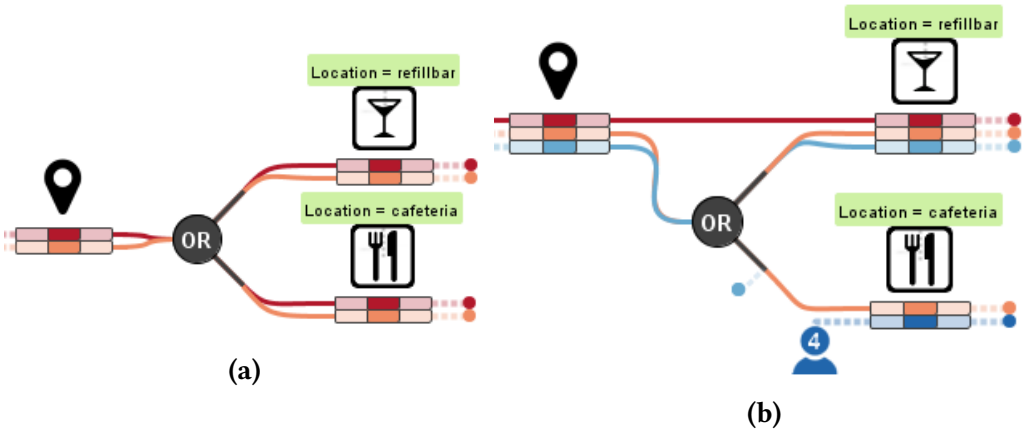


Abbildung 4.7.: Beispiele für die Verzweigungskomponente. In (a) ist ein kleines Beispiel für den Einsatz der Verzweigung zu sehen. In (b) ist kein gültiges Muster zu sehen. Die rote Pfadkante, die nicht über die Verzweigung geht, aber dennoch auch in der Refillbar endet ist falsch.

Dabei muss nicht für jeden Akteur der Pfad bei jeder Alternative weiterhin definiert sein. In Abbildung 4.7b ist ein Beispiel zu sehen, bei dem der hellblaue Pfad nur für die erste Alternative weiterhin definiert ist. Bei der Zweiten endet der Pfad von hellblau direkt nach der Verzweigung. In der gleichen Abbildung ist allerdings auch ein Pfad eingezeichnet, der fehlerhaft ist. So kann der rote Akteur nicht direkt einen Übergang zur Refillbar haben, ohne vorher über die Verzweigung gegangen zu sein. Dies Widerspricht sich mit der Verzweigung, da im Fall der 2. Alternative orange und hellblau sich gemeinsam in der Cafeteria aufhalten sollen, aber rot an dieser Stelle der Sequenz auch ein gemeinsames Ereignis mit den beiden Akteuren in der Refillbar aufweisen muss. Mit der gleichen Argumentation kann man somit allgemein begründen, warum alle Akteure, die sowohl vor als auch nach der Verzweigung im Muster beteiligt sind, zwangsläufig mit der Verzweigung verbunden sein müssen. Der Pfad über die Verzweigung stellt sicher, dass für jeden Akteur jede Alternative definiert ist und es keine widersprüchlichen Angaben im Graphen gibt.

4.3.2. Wiederholungsschleifen

Wiederholungsschleifen dienen, wie die Verzweigungen, dazu, komplexere Pfade der Akteure modellieren zu können. Mit Wiederholungsschleifen kann angegeben werden, ob ein Teilpfad der Anfragesprache mehrfach durchlaufen werden soll. Die dazugehörige visuelle Komponente setzt sich aus zwei Knoten, die dem Teilpfad abgrenzen, und einer Rückkante zusammen. Die Angaben *min* und *max* auf dieser Kante zeigen an, wie viele Male dieser Pfad minimal und maximal wiederholt werden kann. Das Muster in Abbildung 4.8 findet somit Teilsequenzen die mindestens ein und maximal drei Ereignisse in *Workshop1* nacheinander beinhalten, bevor die definierten Teilsequenzen mit einem Ereignis in der Cafeteria enden.

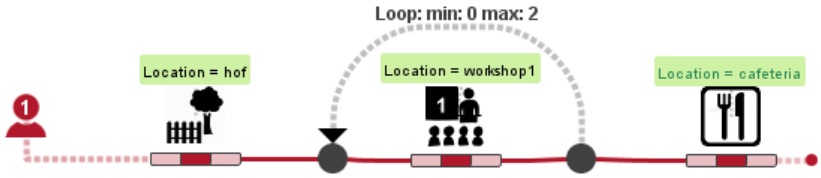


Abbildung 4.8.: Anfragegraph mit einer Wiederholungsschleife. Teilsequenzen, die damit gefunden werden, beinhalten mindestens ein und maximal drei Ereignisse in *workshop1* nacheinander.

Damit man auch Wiederholungen über Teilpfade mehrerer Akteure definieren kann, muss ein Teilpfad bestimmte Eigenschaften erfüllen. Alle Akteure, die sich in einer wiederholenden Sequenz befinden, müssen bei allen Schleifendurchläufen definiert sein und müssen somit mit dem End- und Startknoten der Schleife verbunden sein. Dadurch ist klar definiert welche Bedeutung eine Wiederholung für einen Akteur hat.

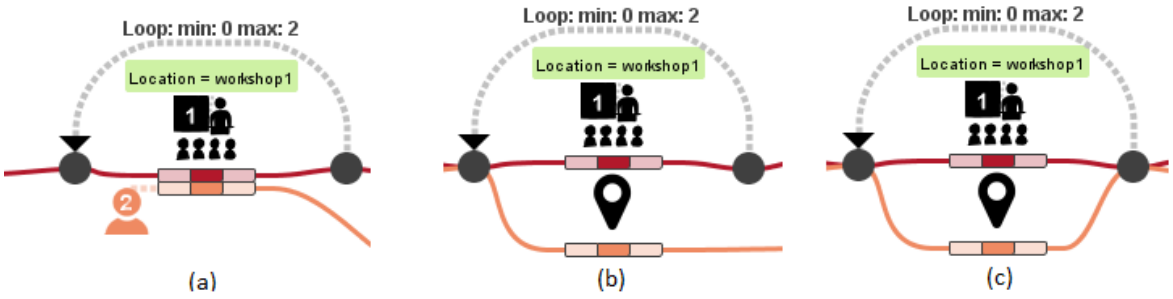


Abbildung 4.9.: Verschiedene Möglichkeiten, Kanten mit den Schleifenknoten zu verbinden. Muster (a) und (b) sind nicht korrekt, Muster (c) ist korrekt.

In Abbildung 4.9 ist dazu eine Übersicht verschiedener Fälle zu sehen, die sowohl falsche als auch richtige Muster zeigt. In (a) ist ein Akteur in dem Teilpfad innerhalb der Schleife beteiligt, über den nicht mit der Wiederholung iteriert wird. Somit ist dieses Muster ungültig. Ebenfalls ungültig ist das Muster in (b), da hier der orange Akteur nicht mit dem Endknoten der Schleife

verbunden ist. Das letzte Beispiel hingegen ist korrekt, da alle beteiligten Akteure mit beiden Knoten verbunden sind.

4.4. Attribute und Restriktionen

Nachdem durch die Pfadkomponenten das Zusammenspiel der gesuchten Ereignissequenzen modelliert ist, kann mit Attributen und weiteren Restriktionen genauer definiert werden, um welche Art von Ereignissen es sich jeweils handelt und ob diese zeitlich in irgendeiner Form eingeschränkt sein sollen. Bereits in den vorherigen Beispielen wurden einfache Attribute bereits ohne Einführung eingesetzt, um die Orte der Ereignisse einzuschränken. In dem folgenden Kapitel werden die unterschiedlichen Restriktionen, die in der neuen Anfragesprache zur Auswahl stehen, beschrieben.

4.4.1. Attribute

Mit Attributen können sowohl die Ereignisse als auch die Akteure und bei Bedarf auch die Pfadkanten eingeschränkt werden. Attribute bilden die einfachste Art der Restriktion von Ereignissen und können im Normalfall ohne viel Erklärung verstanden werden. Attribute sind somit Eigenschaften von Ereignissen, Akteuren oder Pfadkanten, mit deren Hilfe man durch explizite Angabe Einschränkungen treffen kann. Welche Attribute verfügbar sind, hängt dabei von der jeweiligen Datensammlung ab. Für den Republica-Datensatz sind zum Beispiel *location*, *topic* und *event ID* mögliche Attribute, die zur Spezifizierung der Ereignisse verwendet werden können. Für den Akteur könnte in diesem Datensatz die *Device-ID* ein mögliches Attribut sein. Visuell werden Attribute in einer Box, die sich über der dazugehörigen Komponente befindet und mit dieser verbunden ist, dargestellt. Dabei kann je nach Datentyp angegeben werden, ob ein Attribut gleich, unterschiedlich oder bei numerischen Attributen größer oder kleiner als ein bestimmter Wert sein soll. Beispiele hierzu sind in Abbildung 4.10 zusehen.

Abbildung 4.10.: Beispiele für Attribute

4.4.2. Vergleiche

Wie die Attribute sind auch Vergleiche in VESPa bereits definiert gewesen und konnten ebenfalls dazu verwendet werden, Ereignisse oder Akteure miteinander zu vergleichen. Dabei beziehen sich die Vergleiche auf die Werte der Attribute, die miteinander verglichen werden sollen. Die Visualisierung von Vergleichen wird durch einen zusätzlichen Knoten im Graph repräsentiert, der den Vergleich beinhaltet. Beispiel sind in Abbildung 4.11 zu sehen.

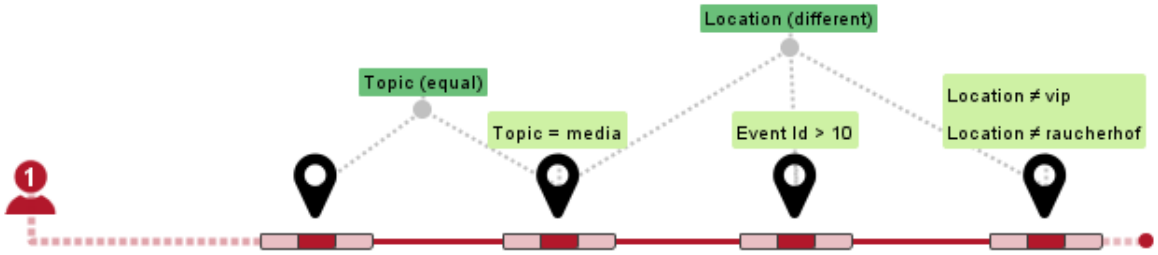


Abbildung 4.11.: Beispiele für Vergleiche

Für Knoten kann bestimmt werden, ob sie sich bestimmte Eigenschaften teilen oder sich in einem oder mehreren Punkten unterscheiden sollen. Alle Komponenten, die in einen Vergleich einbezogen werden und somit bestimmte Eigenschaften teilen oder sich in einem oder mehreren Punkten unterscheiden sollen, können mit einer Kante zum jeweiligen Vergleichsknoten verbunden werden. Dort können dann die gewünschten Vergleiche definiert werden. Welche Eigenschaften hier auswählbar sind ist wieder von den Daten abhängig und steht im direkten Zusammenhang mit den definierten Attributen. Für jedes Attribut besteht die Möglichkeit, darüber einen Vergleich zu ziehen.

4.4.3. Zeitliche Einschränkungen

Zu den größten Erweiterungen der Anfragesprache gehört der Umgang mit zeitlichen Einschränkungen. Bei VESPa werden zeitliche Einschränkungen, wie Ankunftszeit, Abreisezeit oder die Dauer wie jedes andere Attribute zu den Ereignissen hinzugefügt. Ob zeitliche Restriktionen vorhanden waren, war somit auf den ersten Blick nicht erkennbar. Da die Zeit bei Ereignissequenzen allerdings eine große Rolle spielt, bietet die neue Anfragesprache zusätzliche Restriktionen an, die sich nur mit der zeitlichen Komponente beschäftigen. Diese unterscheiden sich nicht nur optisch von den normalen Attributen, sondern stellen die Werte auch in einer jeweils passenden symbolischen Repräsentation dar. Für alle zeitlichen Restriktionen wird dabei die bereits eingeführte neue Timeline-Komponente verwendet, mit deren Hilfe detaillierte Einstellungen getroffen werden können und die den besonderen Stellenwert von zeitlichen Einschränkungen auch visuell hervorheben soll.

Timeline Vorher-Nachher Beziehungen

Die ersten Restriktionen, die durch die Timelines beschrieben werden kann, sind die Vorher-Nachher-Beziehungen (temperal relations [All83]) zwischen den einzelnen Akteuren eines Ereignisses. Möchte ein Nutzer beispielsweise im re:publica-Datensatz explizit angeben, ob eine Person vor oder nach einer anderen Person an einem Ort eintrifft, so kann dies explizit durch die Timeline angegeben werden. Dies wird durch das Verschieben der einzelnen Bereiche der Timeline realisiert. Die relative Positionierung der Balken ist dabei entscheidend. Passt man die unsicheren Bereiche jeweils für alle Akteure eines Ereignisses an, können diese Beziehungen bewusst modelliert werden. Da diese Restriktion normalerweise erst Anwendung findet, wenn der Nutzer eine klare Vorstellung hat, nach was er sucht, sind die Balken der Timeline in der Standardeinstellung so positioniert, dass keine Vorher-Nachher-Beziehungen daraus resultieren (siehe Abbildung 5.2). Möchte man allerdings Einstellungen in diesem Bereich vornehmen, lassen sich 4 verschiedene unterschiedliche paarweise Positionierungen einstellen, die in Abbildung 6.5 dargestellt sind.

Abbildung 4.12.: Verschiedene Möglichkeiten der zeitlichen Beziehung des unsicheren Bereichs der Timeline [All83].

Die Darstellung und die Einstellungen der Timeline sind dabei immer relativ zu sehen. Die Länge der Balken hat somit keine Bedeutung. Entscheidend ist allein, welche Bereiche vor oder nach anderen beginnen.

Timeline Zeit- und Datumsangaben

Der Start- und Endzeitpunkt, sowie beliebige Zeitpunkte die innerhalb eines Ereignisses liegen sollen, können direkt in die Timeline eingezeichnet werden. Damit sind in der neuen Anfragesprache mehr Möglichkeiten verfügbar, wie Zeitpunkte in Ereignissen definiert werden können. Der größte Unterschied besteht allerdings in der Darstellung dieser Angaben. Alle Zeitangaben werden direkt in die Timeline an der entsprechenden Position eingezeichnet. Entscheidend für die Bedeutung ist die Position der Markierung. Liegt die Linie zum Beispiel am Anfang des unsicheren Bereichs, wie in Abbildung (b) so wird damit ausgesagt, dass das dazugehörige Ereignis dieses Akteurs zu diesem Zeitpunkt beginnt. Weitere mögliche Positionierungen und deren Bedeutung sind in Abbildung 4.13 zu sehen.

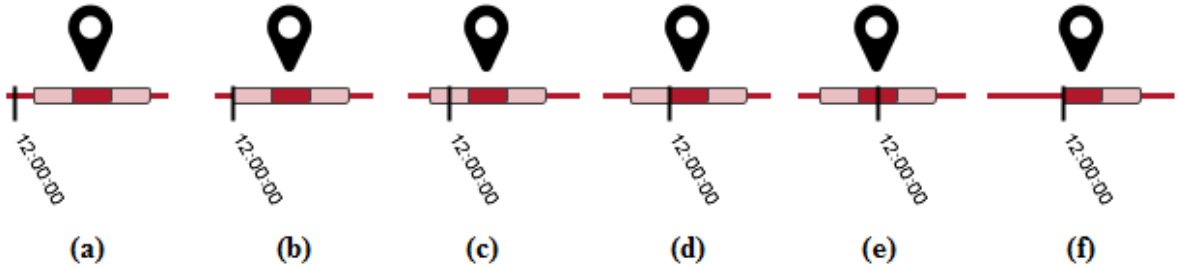


Abbildung 4.13.: Übersicht über die verschiedenen Möglichkeiten, eine Zeitangabe zu positionieren. Das abgebildete Ereignis startet: (a) nach 12 Uhr ; (b) frühestens um 12 Uhr ; (c) vor oder nach 12 Uhr (somit ist keine Restriktion vorhanden) ; (d) spätestens um 12 Uhr ; (e) vor 12 Uhr ; (f) um 12 Uhr

Timeline gemeinsame Aufenthaltsdauer

Die letzte zeitliche Restriktion, die an der Timeline eingegeben werden kann und dort auch dargestellt wird, sind Angaben zur maximalen bzw. minimalen gemeinsamen Aufenthaltsdauer. Diese Angabe bezieht sich auf alle Timelines eines Ereignisses und bezieht sich auf die längste Zeitspanne in der alle Akteure anwesend sind. Die maximale und minimale Dauer dieser Überschneidung kann explizit im Timeline Menü angegeben werden. Für die Darstellung der Zeitangaben wird die der Vorher-Nachher-Beziehungen der Timelines genutzt. Dort kann die gemeinsame Aufenthaltsdauer durch eine Zeitspanne einfach eingetragen werden. Wo diese Zeitspannen liegen, bestimmt dabei nicht der Nutzer, sondern kann anhand der Angaben der Vorher-Nachherbeziehungen bestimmt werden. Die minimale Dauer wird durch den maximalen Bereich bestimmt, in dem alle Akteure sicher da sein müssen. Für die maximale Dauer ist der maximale Bereich entscheidend, in dem noch alle Akteure in den Vorher-Nachher-Beziehungen definiert sind. Entsprechend sieht die Visualisierung wie in Abbildung 4.14b aus.

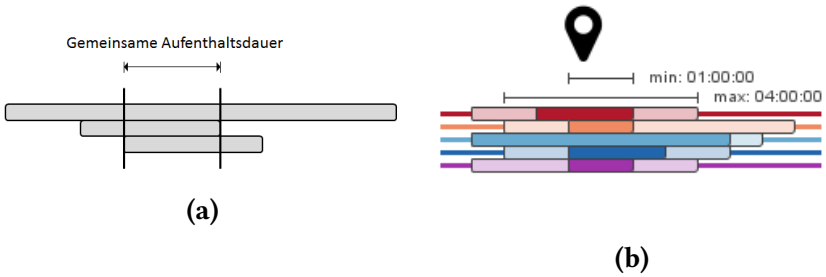


Abbildung 4.14.: Die gemeinsame Aufenthaltsdauer bezieht sich auf die längste Zeitspanne in der alle Akteure anwesend sind. In (a) ist diese Zeitspanne bei eine Visualisierung konkreter Ereignisse eingezeichnet. In (b) ist die Visualisierung der Restriktion in der Anfragesprache zu sehen.

5. Einbettung im Analysesystem

Ist man sich beim Untersuchen einer Datensammlung darüber im Klaren welche Daten einen interessieren, kann mit der visuellen Anfragesprache eine Anfrage an die Datensammlung gestellt werden und die entsprechenden Daten werden gefunden. In vielen Fälle ist allerdings von den gesuchten Daten noch kein klares Bild vorhanden und eine Anfrage lässt sich somit auch nicht formulieren. Dann ist es nötig, einen groben Überblick über die Daten zu bekommen, indem man diese visualisiert und dadurch untersuchbar macht. Die Grundlage für das Explorieren von Daten ist dabei eine geeignete Darstellung und die Möglichkeit durch Filter Teilmengen der Daten zu untersuchen und dabei unwichtige Details ausblenden zu können. Der Weg von einer groben Idee, was gesucht ist, bis zur konkreten Formulierung der Anfrage die die gewünschten Ergebnisse liefert kann dann durch einen iterativen Prozess beschrieben werden. Dabei kann die Anfragesprache in diesen Prozess aktiv eingebunden werden und sowohl als Formulierungshilfe als auch als Abstraktion der vorhandenen Daten dienen.

In dem nachfolgend beschrieben Analysesystem soll der komplette Prozess von der ersten Exploration der Daten, über das Filtern bis zur Formulierung einer konkreten Anfrage möglich sein. Neben der Visualisierung der Daten und der Anfragesprache selbst, werden im System Möglichkeiten angeboten diese durch eine passende Sortierung und Filterung zu strukturieren und so auf bestimmte Bereiche der Daten fokussieren zu können.

5.1. Die Komponenten des Systems

Das komplette System (siehe Abbildung 5.1) setzt sich aus vielen einzelnen Komponenten zusammen, die in enger Verbindung miteinander stehen. Zunächst wird eine Übersicht über alle Komponenten gegeben:

① **Anfragepanel** In diesem Bereich des Systems kann die Anfrage modelliert werden. Alle visuellen Komponenten der Anfragesprache die zuvor beschrieben wurden können im Menü der Oberfläche ausgewählt und miteinander verbunden werden.

② **Overview** Die Overview Komponente soll einen groben Überblick über die Daten geben. Als Darstellung wurde ein ThemeRiver verwendet, der angibt zu welchen Zeitpunkten wie viele Ereignisse in welchen Bereichen stattfinden. Zudem kann auf dieser Oberfläche der Sichtbereich der Ergebnisdarstellung eingestellt werden.

③ **Ereignisdarstellung** In diesem Bereich der Anwendung werden die Ergebnisse der Anfrage visualisiert. Es stehen mehrere Ansichten zur Verfügung die unterschiedliche Teilmengen der Ereignissequenzen darstellen und bestimmte Bereiche können je nach Bedarf ein- oder ausgeblendet werden.

④ **Analyse-Bereich** Im Analyse-Bereich sind die Einstellungsmöglichkeiten für den umgekehrten Weg von den Daten zur Anfrage verfügbar. Hier können die Attribute bestimmt werden die in die neue Anfrage einfließen sollen.

⑤ **Einstellungsmöglichkeiten** Weitere Einstellungsmöglichkeiten, wie die Wahl einer Sortierung für die Ereignissequenzen oder die Wahl des Datensatzes — VAST Challenge oder re:publica — können hier vorgenommen werden.

⑥ **Legende und Color Mapping** Die Legende dient nicht nur als Übersicht über die verwendeten Symbole und Farben der Attribute sondern bietet auch die Möglichkeit, die Visualisierung anzupassen. Mit der Legende können bestimmte Kategorien ein- oder ausgeblendet werden.

Alle Komponenten sind im System eng miteinander verbunden und tauschen Informationen über die aktuell eingestellten Werte der Komponenten aus. So verwenden zum Beispiel alle Komponenten, die Ereignissequenzen visualisieren, das Farbschema welches in der Legende momentan definiert ist. Bei einer Aktion werden entsprechend alle Komponenten die sich

Abbildung 5.1.: Alle Komponenten des Analysesystems im Überblick: ① Anfragepanel; ② Überblick über die Daten; ③ Visualisierung der Ereignissequenzen; ④ Selektionsbereich und Panel für die Analyse; ⑤ Einstellungen; ⑥ Legende

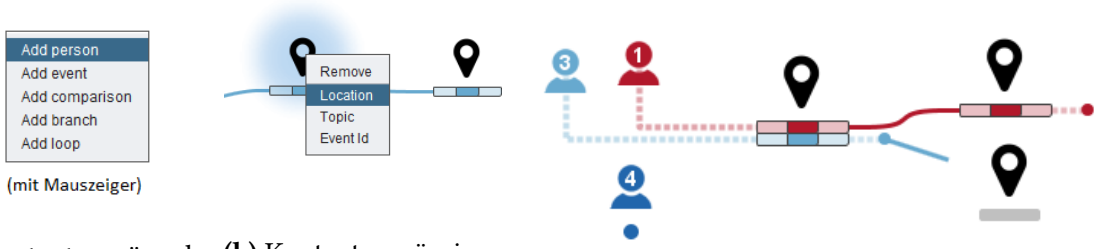
darauf beziehen informiert und gegebenenfalls aktualisiert. Welche Funktionen diese einzelnen Komponenten genau enthalten wird nun separat für jede Komponente beschrieben.

5.1.1. Anfragepanel

Das Anfragepanel ist das Herzstück des Analysesystems. Hier können Anfragen mit der neuen visuellen Anfragesprache modelliert und ausgeführt werden. Da ein Teil der Zielsetzung war, eine einfache und intuitive Anfragesprache zu entwickeln, wurde bei der Anwendung großer Wert auf eine einfache, übersichtliche und reduzierte Gestaltung der Oberfläche und der Einstellungsmöglichkeiten gelegt.

Anfrage erstellen

So können die Nutzer der Anwendung die Komponenten weitestgehend mit der Maus bedienen. Alle Pfadkomponenten, außer die Kanten, sind im Kontextmenü der Oberfläche (Abbildung 5.2a) verfügbar und können an jeder Stelle einfach hinzugefügt werden und von dort gegebenenfalls an die gewünschte Position verschoben werden. Die Kanten können dann direkt, wie bei (s|q)ueries [ZDFD15], von einer Komponente zur nächsten gezogen werden. Dazu wird unter jeder Komponente von der aus es möglich ist Kanten zu ziehen ein Kreis unter der Komponente angezeigt, von dem aus die Kante mit gedrückter Maustaste gezogen werden kann (siehe Abbildung 5.2c). Berührt man mit dem Mauszeiger eine Komponente, mit der eine Kante möglich ist, so wird diese Kante sofort erstellt. So lassen sich mehrere Kanten, und somit ein ganzer Teilpfad, am Stück definieren und erst wenn die Maustaste wieder losgelassen wird, wird der Pfad an der letzten verbundenen Komponente beendet. Dadurch können die Komponenten auf eine schnelle und spielerische Art miteinander verbunden werden.



(a) Kontextmenü der Oberfläche **(b)** Kontextmenü einer Komponente **(c)** Erstellen von Pfaden

Abbildung 5.2.: Bedienung des Anfragepanels

Restriktionen und Einstellungen bei den Komponenten werden über ein Kontextmenü der Komponente (5.2b)realisiert. Hier befinden sich die verfügbaren Restriktionen für die Komponente und verschiedene Aktionen wie das Entfernen der Komponente. Das Einstellen funktioniert dann für alle Komponenten gleich. Durch Anklicken einer einstellbaren Komponente wird ein

Popup aufgerufen welches die jeweiligen Einstellungsmöglichkeiten anzeigt. Eine Übersicht einiger Einstellungsoberflächen ist in Abbildung 5.3 zu sehen.

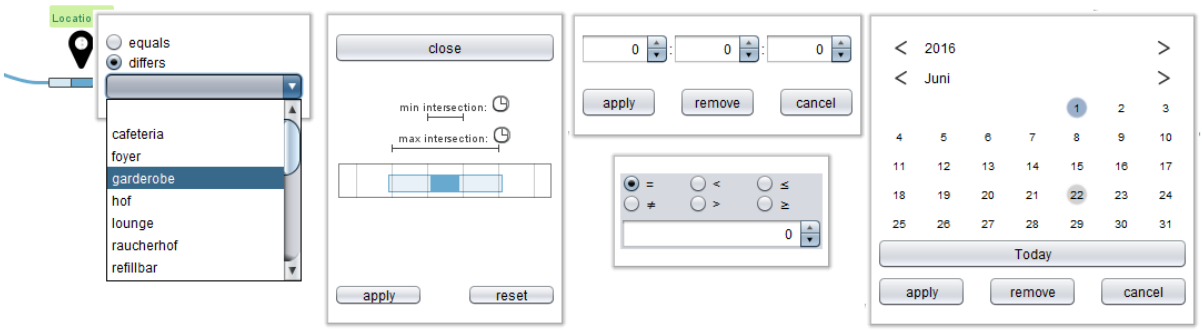


Abbildung 5.3.: Übersicht einiger Einstellungsoberflächen.

5.1.2. Overview

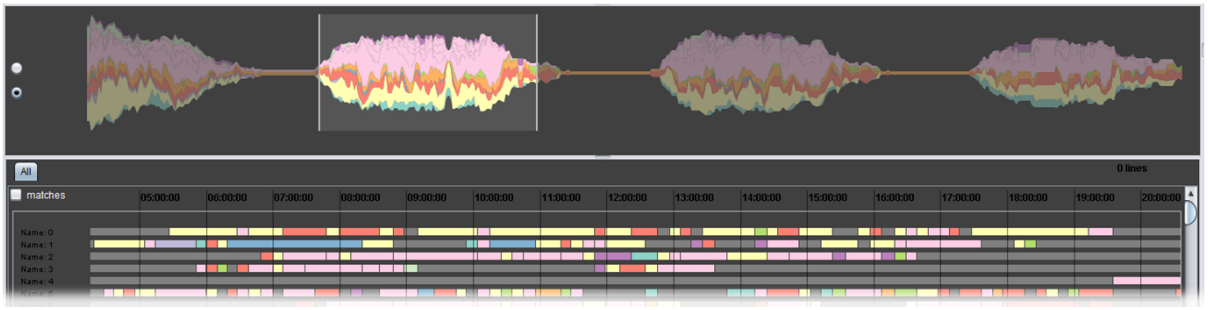


Abbildung 5.4.: Der Overview bietet die Möglichkeit den Sichtbereich einzuschränken. Hier wurde zum Beispiel ein Tag im re:publica-Datensatz ausgewählt.

Der Overview erfüllt im Analysesystem zwei Hauptaufgaben. Die erste Aufgabe dieser Komponente ist es einen Überblick über die Ereignissequenzen zu geben. Dieser wird durch einen ThemeRiver [HHN00] realisiert, der die Anzahl der Ereignisse, die zu einer Kategorie gehören, im zeitlichen Verlauf darstellt. Dadurch entsteht eine Visualisierung, wie sie in Abbildung 5.4 zu sehen ist, die anzeigt wann welche Orte oder Ereignisse mit einem bestimmten Kontext vermehrt vorkommen. Am Anfang der Analyse ist dieses Clustern der Ereignisse besonders hilfreich, da die Daten erst exploriert werden müssen und noch nicht klar ist wie die Zusammensetzung der Daten aussieht. Neben der Darstellung der Ereignissequenzen selbst, wird in der Übersicht zusätzlich angezeigt wo sich Matches und selektierte Daten befinden. So ist sofort erkennbar in welchen Bereichen sich bestimmte Ergebnisse häufen. Dieser Bereich kann dann genauer betrachtet werden.

Die zweite Aufgabe knüpft an dieser Stelle an und ermöglicht es direkt den Sichtbereich auf einen Intervall der Übersicht einzugrenzen der bei der Exploration interessante Muster

aufgewiesen hat. Somit dient die Übersicht zusätzlich als zeitlicher Filter für das Result-Panel. Im Gegensatz zu einer normalen Scrollbar, oder der Einstellung des Sichtbereichs durch Angabe eines Zeitintervalls, wird durch diesen Ansatz sofort klar welche Daten in diesem Bereich vorkommen. Finden Nutzer zum Beispiel interessante Muster in einem bestimmten Zeitfenster, so kann direkt in der Übersicht auf diesen Bereich eingegrenzt werden. Zudem kann der definierte Intervall wieder als kleine Übersicht über die Daten im Sichtfenster verstanden werden.

5.1.3. Result-Panel

Für die Visualisierung der Ereignisse wurde eine einfache Darstellung auf einer Zeitleiste verwendet. Im Zusammenspiel mit den Filter-Möglichkeiten werden hier die Daten einfach als Balken mit entsprechender zeitlicher Ausdehnung eingetragen. Dabei sind mehrere Ansichten verfügbar, die verschiedene Teilmengen, wie zum Beispiel nur die Ergebnisse der Anfrage, anzeigen. Fährt man mit der Maus über die einzelnen Ereignisse der Sequenzen werden hier noch zusätzliche Informationen angezeigt.

Abbildung 5.5.: Ergebnisdarstellung im Result-Panel

5.1.4. Legende und Color Mapping

Die Legende dient nicht nur als Übersicht über verwendete Symbole und Farben der Attribute sondern bietet auch die Möglichkeit die Visualisierung anzupassen. Mit der Legende können bestimmte Werte in den anderen Ansichten aus- oder eingeblendet werden. Dies ermöglicht es Kategorien die nicht im Interesse der aktuellen Suchanfrage liegen in der Ergebnisdarstellung visuell in den Hintergrund zu stellen und so den Fokus auf die relevanten Daten zu lenken. Um ein geeignetes Color Mapping der Daten zu erreichen ist das im System enthalte Standardfarbschema [1] allerdings nicht immer geeignet. So macht es teilweise Sinn verwandte Kategorien mit

[1] ColorBrewer, 12-class Set3, http://colorbrewer2.org/#

ähnlichen Farben zu visualisieren und so einen Zusammenhang zu erschaffen oder Daten mit denen bereits ein anderes bekanntes Farbschema verbunden wird auch so in der Visualisierung darzustellen. Das Farbschema kann aus diesen Gründen an die Daten abgepasst werden.

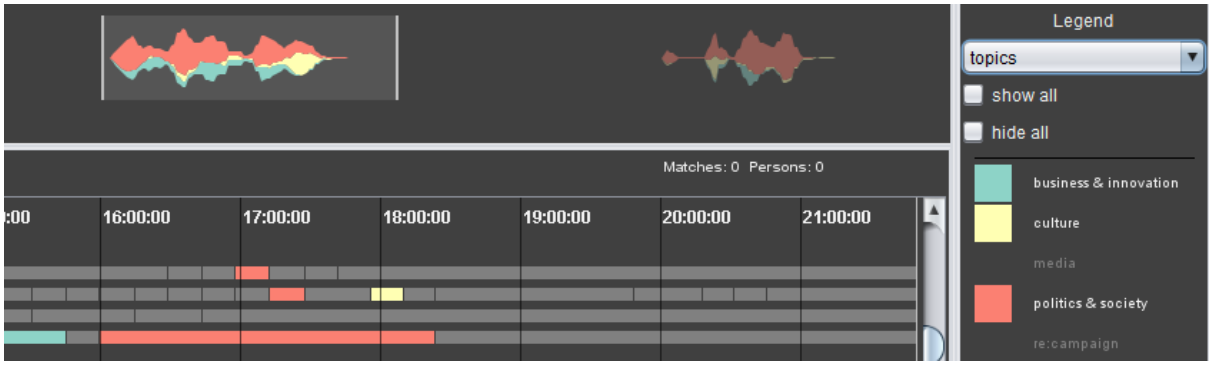


Abbildung 5.6.: Manche Kategorien wurden in diesem Beispiel teilweise ausgeblendet um den Fokus auf die verbleibenden Kategorien zu lenken.

5.1.5. Sortierung

Zu den unterstützten Einstellungsmöglichkeiten gehört auch die Wahl einer passenden Sortierung um einen besseren Überblick über die Daten zu bekommen. Da im System beliebige Ereignissequenz-Daten untersucht werden können, gibt es keine Sortierung die für alle Daten bestens geeignet ist. Aus diesem Grund existiert hier auch die Möglichkeit benutzerdefinierte Sortier-Algorithmen in das System zu integrieren. Ein paar einfache Algorithmen, die sich nur auf die Eigenschaften beziehen, die alle Ereignissequenzen teilen sind allerdings bereits integriert. In Abbildung 5.6 wird eine dieser Sortierungen dargestellt, die die Ereignisse im Vergleich lokalen Ähnlichkeit der selektierten Teilsequenz sortiert.

5.2. Analyse im System

Hat man in der Visualisierung der Daten Muster erkannt die man genauer untersuchen will, ist es für die Analyse hilfreich diese Muster direkt aus den Daten extrahieren zu können und damit zum Beispiel neue Anfragen zu generieren oder die bestehende Anfrage zu erweitern. So können Anfragen iterativ verfeinert werden, bis ein ausreichende Eingrenzung der interessanten Daten gefunden wurde. Dieser Prozess des iterativen Verfeinerns macht die Analyse nicht nur intuitiver sondern auch schneller. Das Analysesystem bietet für diesen Zweck die Möglichkeit direkt aus den Daten Anfragen zu generieren. Dabei handelt sich um ein halbautomatisches Verfahren, dass aus den Ereignissen die der Nutzer selektiert hat automatisch eine Anfrage generiert, die anschließend noch vom Benutzer angepasst werden kann.

5.2.1. Halbautomatische Anfragegenerierung

Das Verfahren zur Anfragegenerierung baut auf die Darstellung Dabei ist die Wahl auf ein halbautomatisches Verfahren gefallen, da so die einzelnen Arbeitsschritte die zur Anfrage führen entsprechend der Kompetenzbereiche von Nutzer und System aufgeteilt werden können. Zusätzlich behält der Nutzer so die Kontrolle und das Verfahren übernimmt nur die Arbeit die Intention des Nutzer möglichst genau mit der Anfrage zu formalisieren.

Der erste Schritt bei der Anfragegenerierung ist die Wahl der Ereignisse die miteinbezogen werden sollen. Diese Aufgabe wird vom Nutzer durchgeführt, da dieser Im Result-Panel können für diesen Fall einzelne Ereignisse selektiert werden, die für die Anfrage Oberstes Ziel bei der Erstellung der Anfrage soll dabei immer sein, dass diese Ereignisse im Anschluss auch Teil des Abfrageergebnisses sind. Nachdem die relevanten Ereignisse selektiert sind, können weitere Einstellungen getroffen werden, die bestimmen welche Aspekte, wie zum Beispiel Restriktionen in der Anfrage miteinbezogen werden. Zusätzlich wird auch die ausgewählten Legendeeinträge bei der Bestimmung der Restriktionen einbezogen. Hierbei wird davon ausgegangen, dass Kategorien die momentan in der Legende nicht eingeblendet sind auch nicht relevant sind, werden Restriktionen zu diesen Kategorien nicht hinzugefügt. Abschließend ist dann

Abbildung 5.7.: Halbautomatische Anfragegenerierung an einem Beispiel. Hierbei können die gemeinsame Ergebnisse nachträglich oder bereits automatisch miteinander verknüpft werden. Das Ergebnis dieser Anfrage entspricht in diesem Fall dann genau dem selektierten Bereich.

wieder wird der Nutzer wieder aktiv dazu aufgefordert dieses Muster nochmal gegebenenfalls anzupassen und so auch die gewünschte Abfrage zu generieren.

6. Implementierung

Die Umsetzung der Konzepte der vorherigen Kapitel erfolgte in einer Java-Anwendung die auf Seiten der Anfragesprache direkt auf die originale Implementierung von VESPa aufbaut. Das dazugehörige Analysesystem ist komplett eigenständig in befindet sich in einem separaten Projekt. Die Anfragesprache kann somit allein oder im Zusammenhang mit dem System verwendet werden. Die einzelnen Komponenten des Systems sind ebenfalls weitestgehend voneinander unabhängig, wodurch das System auch in einer reduzierten Form lauffähig ist. Die Datensammlungen können als serialisierte Java-Klassen in das System geladen. Damit jede beliebige Datensammlung im System geladen werden kann, wurde durch den Einsatz von Interfaces ein allgemeines Datenmodell definiert, das zwischen der Anwendung und den Daten steht. Das System ist somit nicht direkt mit den Daten verbunden sondern kommuniziert ausschließlich durch die Methoden der Interfaces und ist somit allgemein gehalten und nicht nur auf die in dieser Arbeit verwendeten Datensätze zugeschnitten. Dieses universelle Datenmodell wird im folgenden beschrieben.

6.1. Datenmodell

Das Datenmodell der Anfragesprache stellt die Grundlage für eine einheitliche Behandlung aller Daten dar. Alle Datensammlungen die mit der Anfragesprache und dem dazugehörigen Analysesystem untersucht werden sollen müssen die Interfaces dieses Modells implementieren und können so angesprochen werden. Das Modell besteht neben der Hauptklasse selbst, aus mehreren kleinen Klassen die im späteren Verlauf dazu dienen werden die Komponenten der Anfragesprache auf die eigentlichen Daten abzubilden. Diese Klassen enthalten somit die Daten der Datensammlung. ActualPathOwner beinhaltet die Daten der Akteure, ActualState enthält die Daten der eigentlichen Ereignisse und ActualTransfer wird für die Kanten verwendet. Die Klasse ActualPathOwnerSource dient dazu die benötigten Daten vom Modell zu holen. In

Abbildung 6.1.: Klassendiagramme des Datenmodells

Abbildung 6.1 ist ein Überblick über alle Klassen des Modells zu sehen. Dieser Aufbau wurden direkt von VESPa übernommen, wurde aber an einigen Stellen an die neuen Anforderungen angepasst werden.

Wenn die Daten diesem Schema entsprechen, kann daraus eine einheitliche relationale Datenbank erstellt werden auf die mit der visuellen Anfragesprache zugegriffen werden kann. Um dies zu möglichen muss zum einen die Umsetzung der visuellen Anfragesprache in eine SQL-Anweisung stattfinden und zum anderen die eigentliche Datenbank anhand des vorgestellten Datenmodells erstellt werden. Die Umsetzung beider Teile wird in den folgenden Anschnitten beschrieben. Begonnen wird mit der eigentlichen Erstellung der Datenbank.

6.2. Datenbank

Über Interfaces des allgemeinen Datenmodell kann einheitlich auf die Daten zugegriffen werden und eine relationale Datenbank daraus erstellt werden. Dazu werden zunächst alle PathOwner, die zugehörigen Besitzer der Ereignissequenzen, über die ActualPathOwnerSource Klasse geholt. Über die PathOwner kann dann auf die einzelnen Ereignisse zugegriffen und die Daten in die entsprechende Tabelle gespeichert werden. Das Schema das dazu auch schon in VESPa verwendet wurde ist in Abbildung 6.2 zu sehen.

Neben den Ereignis spezifizierenden Daten können hier auch individuelle Attribute, die aus dem jeweils verwendeten Datenmodell hervorgehen, angehängt werden mit denen in Anfragen weiter eingeschränkt und gefiltert werden kann. Im Schema 6.2 können sowohl für die Akteure also auch für den Stays, die den Ereignissen entsprechen, Filter-Attribute angehängt werden. Ein Attribut kann zum Beispiel der Name des Akteurs, der Name der Veranstaltung oder das Thema der Veranstaltung sein. Diese Attribute werden dann für die meisten Restriktionen verwendet.

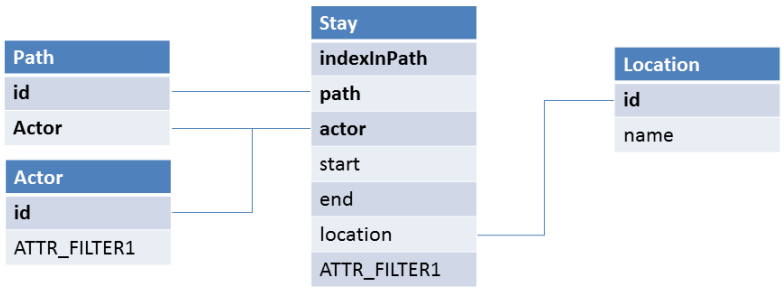


Abbildung 6.2.: Datenbankschema

6.3. Übersetzung der Anfragesprache in eine SQL-Anweisung

Um mit der graphischen Darstellung Anfragen an das System zu stellen, muss eine Übersetzung zu einer herkömmlichen Datenbank-Anfragesprache realisiert werden. Wie auch bei VESPa werden die Komponenten auf eine SQL-Anweisung abgebildet. Als Eingabe wird eine MapQuery-Klasse verwendet, die alle visuellen Anfragekomponenten und deren Beziehungen kennt. Diese Klasse wird dann ausgewertet und die entsprechenden SQL-Statements hinzugefügt. Der Aufbau der SQL-Anweisung wird anhand des Beispiels in Abbildung 6.3 und der dazugehörigen SQL-Anweisung in 6.1 erklärt.

Listing 6.1 SQL-Anweisung des Anfragegraphs aus 6.3

```
1  SELECT A1.id, S1_1.path AS "Path1", S1_1.indexInPath, S2_1.indexInPath,
       S3_1.indexInPath, A2.id, S1_2.path AS "Path2", S1_2.indexInPath, S2_2.indexInPath
2  FROM  Actor A1, Actor A2, Stay S1_1, Stay S1_2, Stay S2_1, Stay S2_2, Stay S3_1
3  WHERE  (A1.id <> A2.id)
4        AND (S1_1.path = S2_1.path)
5        AND (S2_1.indexInPath - S1_1.indexInPath BETWEEN 0 AND 1)
6        AND (S1_2.path = S2_2.path)
7        AND (S2_2.indexInPath - S1_2.indexInPath BETWEEN 0 AND 1)
8        AND (S2_1.path = S3_1.path)
9        AND (S3_1.indexInPath - S2_1.indexInPath BETWEEN 0 AND 1)
10
11       AND (S1_1.actor = A1.id)
12       AND (S1_2.actor = A2.id)
13       AND (S1_1.location = S1_2.location)
14       AND (S1_1.start < S1_2.end AND S1_2.start < S1_1.end)
15       AND (NOT ((S1_1.location = S2_1.location) AND (S1_1.start < S2_1.end) AND
             (S2_1.start < S1_1.end) AND (S1_1.start < S2_2.end) AND (S2_2.start <
             S1_1.end) AND (S1_2.start < S2_1.end) AND (S2_1.start < S1_2.end) AND
             (S1_2.start < S2_2.end) AND (S2_2.start < S1_2.end) ))
16       AND (NOT ((S1_1.location = S3_1.location) AND (S1_1.start < S3_1.end) AND
             (S3_1.start < S1_1.end) AND (S1_2.start < S3_1.end) AND (S3_1.start <
             S1_2.end) ))
17
18       AND (S2_1.actor = A1.id)
19       AND (S2_2.actor = A2.id)
20       AND (S2_1.location = S2_2.location)
21       AND (S2_1.start < S2_2.end AND S2_2.start < S2_1.end)
22       AND (NOT ((S2_1.location = S3_1.location) AND (S2_1.start < S3_1.end) AND
             (S3_1.start < S2_1.end) AND (S2_2.start < S3_1.end) AND (S3_1.start <
             S2_2.end) ))
23
24       AND (S3_1.actor = A1.id)
```

Im FROM-Bereich werden zunächst die Tabellen für die Akteure, Pfade und alle Aufenthalte ausgewählt. Damit alle Aufenthalte eines Akteurs auf genau einen zusammenhängenden Pfad abgebildet werden, muss für die Aufenthalte gelten, dass alle dem selben Pfad angehören und direkt aufeinanderfolge Indizes im Pfad besitzen. Dies wird in den Zeilen 4 – 9, 11 – 12, 18 – 19 und in Zeile 24 realisiert. In Zeile 13 – 14 und 20 – 21 wird dann sichergestellt, dass sich die beiden Ereignisse, die in der Anfrage zwei beteiligte Akteure besitzen, auch tatsächlich zu überlappenden Zeiträumen und am selben Ort stattfinden. Zuletzt wird in den Zeilen 15 – 16 und in der Zeile 22 die Einschränkung getroffen, dass sich unterschiedliche Ereignisse nicht zeitlich schneiden können.

Nach diesem Prinzip werden alle visuellen Pfadkomponenten in eine SQL-Anweisung übersetzt. Im weiteren Verlauf wird nun erklärt wie die erweitern Pfadkomponenten realisiert und Restriktionen sowie Vergleiche zur Anweisung hinzugefügt werden.

6.3.1. Schleifen und Verzweigungen

Um Schleifen und Verzweigungen zu unterstützen, muss die Anfrage in mehrere Teilabfragen unterteilt werden, die erst später zu der gesamten Ergebnismenge zusammengefasst werden. Diese Teilabfragen sind dann einfache Anfragegraphen, die keine Schleifen und Verzweigungen enthalten und somit genau ein Muster repräsentieren. Bei einer Verzweigung entspricht ein Teilgraph einer der möglichen Alternativen, die aus dem Graphen hervorgehen. Bei Schleifen muss jede mögliche Anzahl von Schleifendurchgängen separat als Anfrage betrachtet werden. Dazu muss die Schleife bis zu einer bestimmten Anzahl von Durchläufen ausgerollt werden, sodass wieder ein einfacher Graph entsteht. Ein Beispiel ist in Abbildung 6.4 zu sehen. Die Menge an Teilanfragen wird dann nacheinander abgearbeitet und anschließend zu einer gesamten Ergebnismenge zusammengefasst. Die komplette Abbildung der Ergebnisse wird später nochmal separat thematisiert.

6.3.2. Restriktionen

Nachdem alle Pfadkomponenten in die SQL-Anweisung eingebunden wurden, können im WHERE-Bereich der Anweisung alle expliziten Restriktionen angehängt werden. Je nachdem welche Restriktion gegeben ist, sehen die entsprechenden Bedingungen unterschiedlich aus.

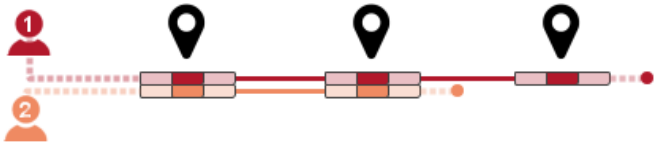


Abbildung 6.3.: Beispiel Anfragegraph der in 6.1 in eine SQL-Anweisung umgesetzt wurde.

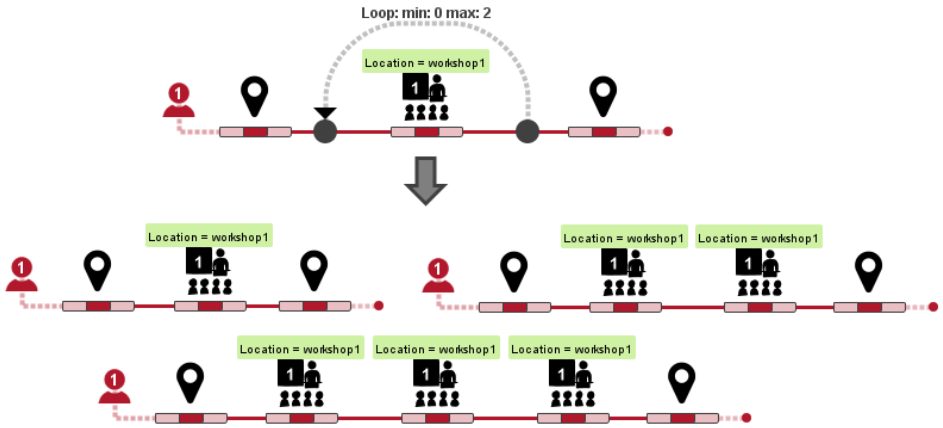


Abbildung 6.4.

Welche Bedingungen aus den verschiedenen Arten von Restriktionen folgen, wird in den kommenden Abschnitten beschrieben.

AttributeRestriction

Die einfachste Restriktion ist die AttributeRestriction und existierte in dieser Form auch bereits in VESPa. Damit werden die Attribute, die in der Anfrage für die Ereignisse definiert werden können, auf die jeweiligen Stays in der Datenbank abgebildet. Die Umsetzung erfolgt durch das Hinzufügen von einem einfachen Vergleichsoperator für alle beteiligten Stays:

Listing 6.2 Ergänzungen für AttributeRestriction

```
1  WHERE  [...]
2         AND (S2_1.ATTR_FILTER = XY)
3         AND (S4_2.ATTR_FILTER = XY)
4         AND (S3_3.ATTR_FILTER = XY)
```

Von der allgemeinen AttributeRestriction können spezielle Klassen für Attribute eines bestimmten Datentyps abgeleitet werden, sodass für jeden Datentyp eine passende Vergleichsoperation in der SQL-Anweisung gewählt werden kann. Momentan sind folgende Klassen verfügbar:

- StringBasedRestriction
- StringListBasedRestriction
- LongBasedRestriction
- DateBasedRestriction
- TimeBasedRestriction

Vergleiche

Vergleiche zwischen Ereignissen werden über die AttributeRestrictions gezogen. Dort kann angegeben werden, ob sich bestimmte Attribute gleichen oder unterschieden sollen. Die Einbindung in die SQL-Anweisung erfolgt durch das Hinzufügen von der entsprechenden Vergleichsoperation. Diese ist wiederum abhängig vom Datentyp. Für die LongBasedRestriction würde bei Gleichheit die SQL-Anweisung folgendermaßen ergänzt werden:

Listing 6.3 Ergänzungen für Vergleiche

```
1 WHERE  [...]
2        AND (S2_1.ATTR_FILTER = S2_1.ATTR_FILTER)
```

TimelineRelationRestriction

Für Restriktionen, die sich mit dem zeitlichen Konzept der Ereignissequenzen beschäftigen, wurden verschiedene Restriktionen erstellt. Die TimelineRelationRestriction gehört zu dieser Art von Restriktionen und modelliert die relativen Vorher-Nachher-Beziehungen der einzelnen Akteure.

(a) **(b)** **(c)** **(d)**

Abbildung 6.5.: Fallunterscheidung der paarweisen Überschneidungen der Timelines. Es lassen sich insgesamt 4 unterschiedliche Fälle für die Beudeutung unterscheiden [All83].

In der visuellen Timeline-Komponente können diese Beziehungen durch Einstellen eines sicheren und eines unsicheren Bereichs angegeben werden. Um die Beziehungen zwischen den Akteuren definieren zu können, müssen die Timelines der Akteure paarweise miteinander verglichen werden. Die Überlappung dieser Bereiche legt die Vorher-Nachher-Beziehung zwischen den Timelines fest und bestimmt somit die Restriktionen, die im WHERE-Bereich hinzugefügt werden müssen. Man unterscheidet prinzipiell zwischen den in Abbildung 6.5 gezeigten Fällen. Diese beziehen sind jeweils auf das Ende der Timeline, der Start ist dazu symmetrisch. Im Einzelnen ergeben sich daraus folgende Bedingungen für die SQL-Anweisung:

Für den letzten Fall wird keine Bedingung erzeugt, da keine Aussage darüber getroffen werden kann, ob Timeline A vor oder nach Timeline B endet. Für ein Ereignis muss für jeden der Stays paarweise auf diese Art miteinander verglichen und die entsprechenden Bedingungen hinzugefügt werden um die Vorher-Nachher-Beziehungen in der Anfrage einzubauen.

Listing 6.4 Fallunterscheidung für die TimelineRelationRestriction

```
1  WHERE [...]
2        --Fall a:
3        AND (A.end <= B.end)
4        --Fall b:
5        AND (A.end = B.end)
6        --Fall c:
7        AND (A.end < B.end )
8        [...]
```

TimelineTimeRestriction

Die TimelineTimeRestriction bindet Zeitpunkte wie eine Uhrzeit oder das Datum in die Anfrage ein. Hierzu müssen wieder die Balken der Timeline mit dem unsicheren Bereich und dem sicheren Bereich einbezogen werden, da, abhängig von der Position des Zeitpunktes in dieser Linie, unterschiedliche Bedeutungen entstehen. Hier werden wieder mehrere Fälle unterschieden, die in Abbildung 6.6 zu sehen sind. Wieder ist nur ein Teil der Fälle aufgezeigt, der andere Teil der Fälle ist symmetrisch dazu.

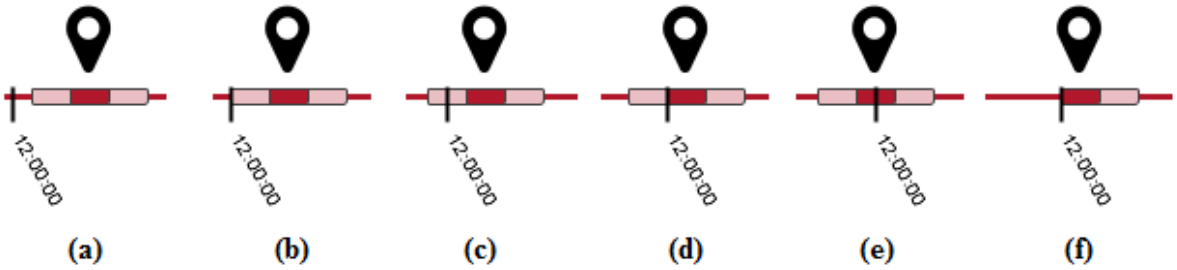


Abbildung 6.6.

Die Umsetzung der Fälle in eine SQL-Anweisung sind in Listing 6.5 zu sehen. Hier ist, wie auch bei den TimelineRelationRestrictions, ein Fall dabei, der keine einschränkende Bedingung erzeugen kann. Dabei handelt es sich um Fall c.

TimelineIntersectionRestriction

Um bestimmen zu können ob sich alle individuellen Aufenthalte der beteiligten Personen eines Ereignisses überschneiden reicht es, die Aufenthalte paarweise miteinander auf Überschneidungen zu testen, wie es auch bei der Übersetzung der Pfadkomponenten in die SQL-Anweisung gemacht wird. Für die TimelineIntersectionRestriction muss allerdings die Zeitspanne dieses Schnittes betrachtet werden. Die analytische Berechnung der tatsächlich daraus resultierenden Aufenthaltsdauer, in der alle Akteure gleichzeitig anwesend sind, ist dabei innerhalb einer SQL-Anweisung nicht möglich. Stattdessen kann eine logische Bedingung definiert werden, die

Listing 6.5 Fallunterscheidung der TimelineTimeRestriction

```
1
2  WHERE [...]
3        --Fall a:
4        AND ((((((S13_1.end - (S13_1.end % (1000 * 60 * 60 * 24)))/(1000*60*60*24)) =
             ((S13_1.start - (S13_1.start % (1000 * 60 * 60 * 24)))/(1000*60*60*24)))
             AND ((S13_1.start) % (1000*60*60*24)) > X)))
5
6        --Fall b:
7        AND ((((((S13_1.end - (S13_1.end % (1000 * 60 * 60 * 24)))/(1000*60*60*24)) =
             ((S13_1.start - (S13_1.start % (1000 * 60 * 60 * 24)))/(1000*60*60*24)))
             AND ((S13_1.start) % (1000*60*60*24)) >= X)))
8
9        --Fall d:
10       AND ((((((S13_1.end - (S13_1.end % (1000 * 60 * 60 * 24)))/(1000*60*60*24)) =
             ((S13_1.start - (S13_1.start % (1000 * 60 * 60 * 24)))/(1000*60*60*24)))
             AND ((S13_1.start) % (1000*60*60*24)) <= X)))
11
12       --Fall e:
13       AND (((S13_1.start % (1000*60*60*24) < X)
14       AND (S13_1.end % (1000*60*60*24) > X)))
15
16       --Fall f:
17        AND ((((((S13_1.end - (S13_1.end % (1000 * 60 * 60 * 24)))/(1000*60*60*24)) =
              ((S13_1.start - (S13_1.start % (1000 * 60 * 60 * 24)))/(1000*60*60*24)))
              AND ((S13_1.start) % (1000*60*60*24)) = X)))
18       [...]
```

für genau alle Paare der Aufenthalte $stays$ eines Ereignisses gilt, die eine minimale gemeinsame Aufenthaltsdauer von X besitzen:

$$\begin{aligned}\bigwedge_{s1,s2 \in stays} & ((s1.start < s2.start) \wedge (s1.end < s2.end) \wedge (s1.end - s2.start) >= X) \\ & \vee((s1.start < s2.start) \wedge (s1.end >= s2.end) \wedge (s2.end - s2.start) >= X) \\ & \vee((s1.start >= s2.start) \wedge (s1.end < s2.end) \wedge (s1.end - s1.start) >= X) \\ & \vee((s1.start >= s2.start) \wedge (s1.end >= s2.end) \wedge (s2.end - s1.start) >= X))\end{aligned} \tag{6.1}$$

Konkret im Fall von zwei Akteuren, die einen gemeinsamen Aufenthalt von mindestens 15 Minuten haben sollen, entsteht folgende SQL-Anweisung:

Listing 6.6 SQL-Anweisung für eine minimale Aufenthaltsdauer von 15 Minuten

```
1  WHERE [...]
2  AND   (((s1.start<s2.start) AND (s1.end<s2.end) AND (s1.end-s2.start)>=900000)
3         OR ((s1.start<s2.start) AND (s1.end>=s2.end) AND (s2.end-s2.start)>=900000)
4         OR ((s1.start>=s2.start) AND (s1.end<s2.end) AND (s1.end-s1.start)>=900000)
5         OR ((s1.start>=s2.start) AND (s1.end>=s2.end) AND (s2.end-s1.start)>=900000))
6         [...]
```

Die maximale Aufenthaltsdauer kann analog dazu formuliert werden. Diese wird allerdings nach oben durch den kürzesten gemeinsamen Aufenthalt aller paarweisen Aufenthalte begrenzt. Es ergibt sich folgende Bedingung:

$$\begin{aligned}\bigwedge_{s1,s2 \in stays} & ((s1.start < s2.start) \vee (s1.end < s2.end) \vee (s1.end - s2.start) <= X) \\ & \vee((s1.start < s2.start) \vee (s1.end >= s2.end) \vee (s2.end - s2.start) <= X) \\ & \vee((s1.start >= s2.start) \vee (s1.end < s2.end) \vee (s1.end - s1.start) <= X) \\ & \vee((s1.start >= s2.start) \vee (s1.end >= s2.end) \vee (s2.end - s1.start) <= X))\end{aligned} \tag{6.2}$$

Möchte man bei der Anfrage eine exakte Dauer des Schnitts einstellen, so erreicht man dies indem man die minimale und maximale gemeinsame Aufenthaltsdauer auf den gleichen Wert setzt. Dadurch fällt die untere mit der oberen Schranke zusammen und nur Ereignisse mit exakt dieser Dauer werden gefunden.

6.3.3. ResultMapping

Das Ergebnis der SQL-Anweisung, die im vorherigen Unterkapitel beschrieben wurde, liegt direkt nach der Ausführung in Form einer Tabelle vor. Damit die Daten allerdings weiterverarbeitet und im System dargestellt werden können, müssen diese wieder mit dem Datenmodell aus Kapitel 6.1 in Verbindung gebracht werden. In der Klasse *QueryResultMapping* wird diese Abbildung direkt nach Durchführung der Anweisung vorgenommen. Falls Schleifen oder Verzweigungen in der Anfrage enthalten sind, werden erst alle einzelnen Anfragen nacheinander durchgeführt und anschließend zusammengefasst in einem Ergebnisarray gespeichert.

6.4. System

Um die im Kapitel 4 beschriebene Funktionalität zu erreichen, müssen die Komponenten untereinander kommunizieren und sich gegenseitig über Änderungen informieren. Um dieses Verhalten zu implementieren wurde in dieser Arbeit das Beobachter-Entwurfsmuster (Observer) verwendet. Dabei werden teilweise Push Notifications — die beobachtete Komponente

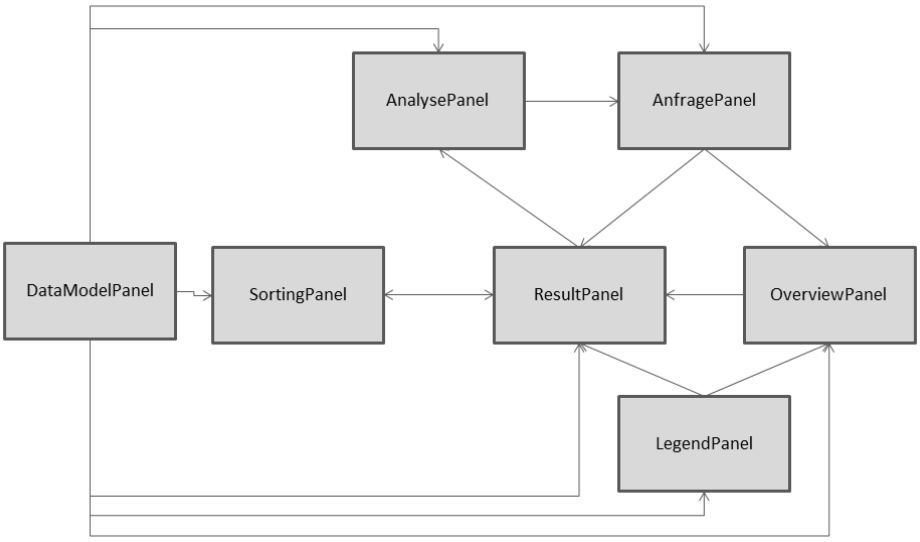


Abbildung 6.7.: Übersicht der Listener. Kanten in Richtung einer Komponente zeigen an, dass diese Notifications von der anderen Komponente erhält.

informiert alle Beobachter — und teilweise Push-Update Notifications — die beobachte Komponente informiert und übermittelt zusätzlich die geänderten Attribute — eingesetzt. Die Umsetzung erfolgt dabei mit Listener-Interfaces, die die Beobachter implementieren müssen. Anschließend können sie dann bei der Komponente registriert werden. Zu den Komponenten, die beobachtet werden, gehören:

Anfrage-Panel Informiert alle Listener wenn ein Anfrage durchgeführt wurde und übermittelt die Ergebnisse (Push-Update Notification).

Legende-Panel Informiert alle Listener wenn sich das ColorMapping geändert hat (Push Notification).

Modell-Panel Informiert alle Listener wenn sich das Datenmodell geändert hat (Push-Update Notification).

Overview-Panel Informiert alle Listener wenn sich der Sichtbereich geändert hat (Push-Update Notification).

Sorting-Panel Informiert alle Listener wenn sich der Sortieralgorithmus geändert hat (Push-Update Notification).

Result-Panel Informiert alle Listener über Selektionen (Push-Update Notification).

Da die Komponenten sehr eng miteinander verbunden sind, entsteht so ein Netz welches in Abbildung 6.7 zu sehen ist.

6.4.1. Ergebnisdarstellung

Die Ereignisdarstellung ist wie bereits erwähnt in mehrere Ansichten (Views) gegliedert, die jeweils eine Teilmenge der Ereignissequenzen darstellen. Die PathOwner und somit die Ereignissequenzen werden dabei direkt über die PathOwnerSource geholt und im Result-Panel für die einzelnen Ansichten verfügbar gemacht. Das Result-Panel kümmert sich zentral darum, nach jeder durchgeführten Anfrage die Teilmengen der PathOwner zu bestimmen, die für die einzelnen Ansichten benötigt werden.

Für die Visualisierung muss eine Abbildung der Ereignissequenzen auf visuelle Komponenten erfolgen, die sich um die Darstellung in der Ansicht kümmern.

Für die Darstellung der Sequenzen existiert für jede Ansicht genau eine Instanz der Klasse View, die eine Teilmenge der gefundenen PathOwner anzeigt.

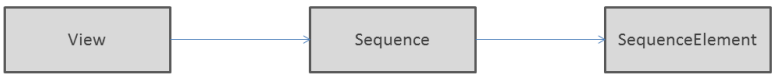


Abbildung 6.8.: Datenmodell hinter der Visualisierung der Ereignissequenzen.

Selektion

Mit diesen Klassen wird auch die Selektion der ActualStates im Panel realisiert. Bei einem Mausklick auf ein SequenceElement der Oberfläche wird im Element und im View vermerkt, dass dieses Element selektiert wurde. Da jedes SequenceElement den ActualState kennt, zu dem es gehört, kann bei Anfrage die Menge der selektierten ActualStates zurückgegeben werden. Das Result-Panel kümmert sich bei jeder Änderung darum, dass alle Listener informiert werden.

6.4.2. Overview

Im Overview wird ein ThemeRiver [HHN00] implementiert, der einen groben Überblick über die Sequenzdaten darstellt. Bei der Implementierung muss dazu die gesamte Zeitspanne des Overviews in diskrete, gleichmäßige Abschnitte unterteilt werden. Anschließend muss für jeden Abschnitt bestimmt werden, bei wie vielen Sequenzen zu diesem Zeitpunkt ein Ereignis stattfindet, welches in eine der Kategorien fällt. Zu diesem Zweck werden in einem zweidimensionalen Array die Ereignisse gezählt, die in die einzelnen Abschnitte fallen. Dazu muss zunächst berechnet werden, über welche Abschnitte die Ereignisse reichen. In Abbildung 6.9 sind hier beispielhaft die passenden Einträge für die unten dargestellte Sequenz aufgezeigt. Die einzelnen Sequenzen erstrecken sich dabei jeweils über mehrere Abschnitte hinweg und

können auch in der Mitte eines Abschnittes enden. Für jeden Abschnitt der komplett in einem Ereignis enthalten ist, wird der Zähler im entsprechenden Eintrag erhöht. Endet ein Ereignis mitten in einem Abschnitt, wird dieser nur um einen Bruchteil erhöht.

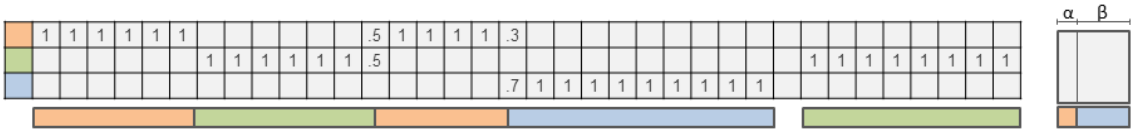


Abbildung 6.9.: Beispiel für die Einträge eines Arrays mit 3 Kategorien (orange, grün, blau) nach einer verarbeiteten Sequenz.

Gezeichnet wird der ThemeRiver, indem aus den Einträgen pro Kategorie ein Polygon berechnet wird. Um die Polygone übereinander und um die Mitte zentriert anzuordnen, muss der Y-Wert der Polygone noch entsprechend verschoben werden. Das Ergebnis sieht dann in etwa wie in Abbildung 6.10 aus.

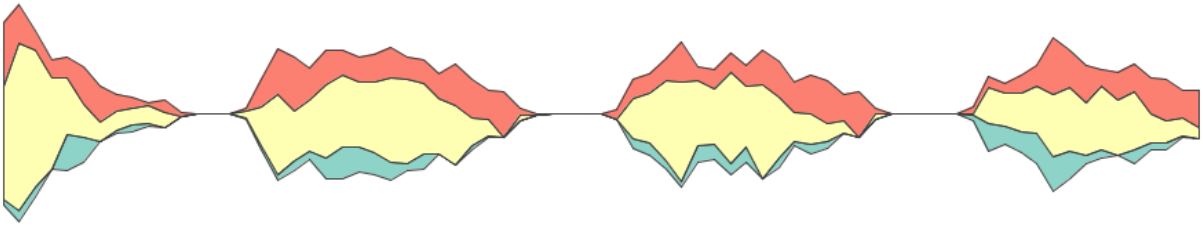


Abbildung 6.10.: Beispiel eines ThemeRivers mit 3 Kategorien und einer groben Auflösung der Zeitschritte. Um die Grenzen der Polygone sichtbar zu machen sind in diesem Beispiel die Kanten eingezeichnet.

Zusätzlich wird in dieser Oberflächenkomponente auch angezeigt, in welchen Bereichen sich Matches und selektierte Ereignisse befinden. Die Implementierung dieser Anzeigen ist ähnlich zum ThemeRiver, mit dem Unterschied, dass hier nur ein eindimensionales Array verwendet wird um die Anzahl der Matches beziehungsweise Selektionen zu zählen, die in die diskreten Zeitschritte fallen. Für die Darstellung der Werte werden diese zuvor auf den Wertebereich zwischen 0 und 1 beziehungsweise zwischen 0 und 255 im RGB-Farbraum skaliert.

6.4.3. Sortierung

Neben der Möglichkeit eigene Sortieralgorithmen direkt mit der Implementierung der Datenmodell Interfaces einzubinden, werden im System bereits zwei einfache Sortieralgorithmen angeboten, mit denen die Ereignissequenzen in eine bestimmte Reihenfolge gebracht werden können. Diese beschränken sich auf Eigenschaften des allgemeinen Datenmodells. So sortiert zum Beispiel der einfachste Algorithmus die Sequenzen anhand des ActualPathOwner-Namens

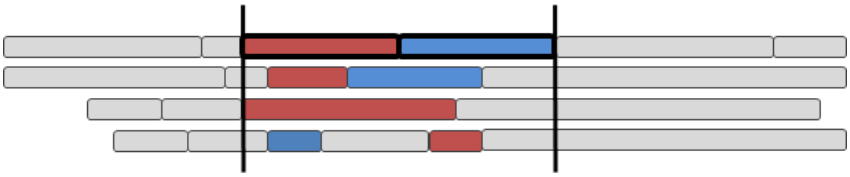


Abbildung 6.11.: Verteilungsmetrik

lexographisch. Der zweite Sortieralgorithmus versucht, die Sequenzen lokal um den Punkt des momentanen Interesses zu sortieren. Er wird im Folgenden vorgestellt.

Kosinus-Ähnlichkeit der lokalen Verteilung

Bei diesem Sortieralgorithmus werden die Ereignissequenzen lokal in einer bestimmten Zeitspanne miteinander verglichen. Die Festlegung der Zeitspanne geschieht durch das Selektieren einer Teilsequenz der Daten. In diesem Bereich werden die Sequenzen nach Ähnlichkeit zur selektierten Sequenz sortiert. Unter der Annahme, dass der selektierte Bereich auch im Fokus des Interesses steht, sortiert der Sortieralgorithmus somit die Sequenzen absteigend nach aktueller Relevanz.

Als Distanzmetrik zur Bestimmung der Ähnlichkeit wird hierbei die Kosinus-Ähnlichkeit der Feature-Vektoren der Sequenzen verwendet. Die Feature-Vektoren basieren dabei auf der prozentualen zeitlichen Verteilung der Kategorien, die in Betracht gezogen werden und im ausgewählten Bereich liegen. Für die Teilsequenz aus Abbildung 6.11 würde der Feature-Vektor folgendermaßen aussehen:

$$v = (0.5, 0.5, 0.0)^T \tag{6.3}$$

Im Beispiel wird so durch die rot und blau markierten Ereignisse der lokale Bereich des Interesses festgelegt. Die Farben in der Skizze stehen dabei für drei unterschiedliche Kategorien. Mit der folgenden Metrik kann danach die Distanz zwischen der selektierten Sequenz, dem Referenzvektor r, und jeder anderen Sequenz in Vektorform v in einem N-dimensionalen Vektorraum für N Kategorien bestimmt werden durch:

$$d_i = \frac{\sum_{i=1}^{N}(v_i \cdot r_i)}{\sqrt{\sum_{j=1}^{N}(v_i)^2} \cdot \sqrt{\sum_{j=1}^{N}(r_1)^2}} \tag{6.4}$$

Da alle Vektoren per Definition die Länge 1 haben, vereinfacht sich die Formel zu:

$$d_i = \sum_{i=1}^{N}(v_i \cdot r_i) \tag{6.5}$$

6.4.4. Anfragegenerierung aus den selektierten Daten

Bei der Implementierung des Analyse-Panels steht man vor zwei großen Aufgaben. Zum Einen muss der Nutzer in der Lage sein, eine Teilmenge der Daten zu selektieren und zum Anderen muss ein Verfahren dafür sorgen, dass aus diesen Daten auch eine Anfrage generiert wird. Das erste Problem der Selektion wird dabei bereits von dem Result-Panel übernommen, welches die selektierten ActualStates und somit die Daten der Ereignisse an das Analyse-Panel übermittelt.

Für die Verarbeitung der Daten müssen dann die PathOwner der States bestimmt und, basierend auf der Reihenfolge in der Sequenz, berechnet werden, ob die ausgewählten Ereignisse miteinander verbunden werden müssen. Das Ergebnis des ersten Schrittes sind somit mehrere einzelne Ereignissequenzen, die keine gemeinsamen Ereignisse besitzen. Anschließend werden die Sequenzen auf Überschneidungen mit anderen Ereignissen getestet. Findet eine Überschneidung statt, werden diese Ereignisse zu einem Ereignis zusammengefasst, indem die Restriktionen des einen Ereignisses auf das andere übertragen und die Kanten entsprechend verschoben werden.

7. Evaluation

Das wichtigste bei einer Anfragesprache oder einem kompletten Analysesystem ist die Benutzbarkeit. Die Konzepte müssen für den Nutzer sowohl verständlich als auch in einem gewissen Maß universell sein und dürfen nicht nur mit ganz speziellen Daten oder für sehr definierte sein. und Lösungen für Um verschiedene Aspekte der Anfragesprache zu evaluieren wird in diesem Kapitel zum einen die Ergebnisse einer Benutzerstudie zur Anfragesprache vorgestellt und zum anderen anhand von Anwendungsfällen die Verwendung des Analysesystems erklärt.

7.1. Anwendungsfälle

In den Anwendungsfällen geht es zunächst um das komplette System und seine Anpassungsfähigkeit an das jeweilige Datenmodell. Hier wird beschrieben welche Anpassungen vorgenommen werden müssen um die Daten in einer geeigneten Form zu integrieren und wie anschließend damit Daten analysiert werden können und Anfragen erstellt werden können. Das Vorgehen wird dabei an zwei konkreten Szenarien beschrieben. Bei den Szenarien handelt es sich um die bereits in der Einführung beschriebenen Datensammlungen.

7.1.1. re:publica Anwendungsfall

Das Analyseziel in diesem Szenario soll sein einen Überblick über die Daten zu bekommen und auffällige Muster zu extrahieren und zu untersuchen ob Zusammenhänge zwischen bestimmten besuchten Veranstaltungen bestehen. Es wird somit gezeigt wie Datensammlungen mit dem System exploriert werden können.

Bevor die Analyse starten kann, muss zunächst die Integration der Daten und die Überführung in das allgemeine Datenmodell stattfinden. Hierzu müssen wie in der Implementierung beschrieben die Interfaces der Datenmodelle im re:publica-Datenmodell implementiert werden. Im Datenmodell entsprechen die WLAN-Geräte den ActualPathOwner und die für die einzelnen Geräte gemessenen, zusammengefassten und angereicherten Zeitspannen ergeben die ActualStates uns sind somit die Ereignisse. ActualTransfer wird in diesem Fall nicht benötigt. Als Filter-Attribute wurde für die WLAN-Geräte *Device ID* ausgewählt und für die Ereignisse werden *Topic*, *Room*, *Event ID*, *Event Title* und *Event Speakers* verwendet. Zusätzlich wird für

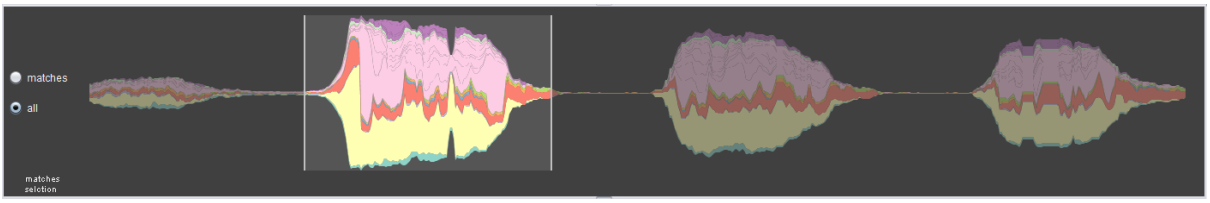


Abbildung 7.1.: Überblick über die Räume

room und *culture* ein Color Mapping verfügbar gemacht. Die in diesem Datensatz verwendeten Icons stammen aus Flaticon, einer Datenbank für Symbole und Icons, und sind optisch abgepasst worden[1].

Analyse

Lädt man die so vorbereitete Datensammlung in das Analysesystem kann man zunächst mit einem Blick auf das Overview-Panel einen Überblick über die Daten bekommen. In diesem Fall kann man erkennen, dass die Besuchermenge, wie zu erwartend ist, morgens stark ansteigt, über den Tag relativ konstant bleibt abends langsam zu sinken beginnt. Gut zu sehen ist auch, dass am ersten Tag der Datensammlung die Konferenz noch nicht begonnen hat und die Anzahl der Personen, die sich auf dem Gelände befinden, entsprechend gering ausfällt. In den 3 Tagen der Konferenz zeigen sich jeweils ähnliche Verteilungen an den einzelnen Orten.

Ein Zeitpunkt sticht dabei beim ersten Tag besonders deutlich hervor. Dort scheinen sich viele Personen fast zur gleichen Zeit auf den Weg zur *stage1* zu machen. Die Vermutung liegt nahe, dass es sich dabei um eine Eröffnungsveranstaltung handelt. Mit einem Wechsel des Color Mappings, sodass nun die Themengebiete angezeigt werden kann gezielt angezeigt werden zu welchen Zeiten welche Themen am häufigsten besucht wurden. Hier wird ebenfalls angezeigt, dass in diesem Zeitraum viele Ereignisse mit dem Thema *republica* stattfinden. Betrachtet man davon einzelne Sequenzen im Result-Panel sieht man dass es sich dabei auch um ein Ereignis mit dem Title *Eröffnung* handelt.

Nun könnte man auf Basis dieser Beobachtung beispielsweise daran interessiert sein welche Personen zu der Eröffnungsveranstaltung gegangen sind. Dazu wird ein Ereignis der Eröffnungsveranstaltung ausgewählt und anschließend im Analyse-Panel die Attribute ausgewählt welche bei der Anfrage von Interesse sind. In diesem Fall möchte man die genaue Veranstaltung in der Anfrage abbilden und wählt entsprechend *Event Title* aus. Als Ergebnis werden dann alle Personen angezeigt die in dieser Veranstaltung waren (siehe Abbildung **??**). Wechselt man dann zum Beispiel auf den anderen Modus der Übersicht, sodass nur eine Übersicht über alle Ereignissequenzen gegeben wird in denen ein Ergebnis für die vorherige Anfrage gefunden

[1]Die Icons wurden von http://www.freepik.com/ und http://www.flaticon.com/authors/picol erstellt, stammen von www.flaticon.com und sind lizenziert von CC 3.0 BY https://creativecommons.org/licenses/by/3.0/

Abbildung 7.2.: Ergebnis der Analyse

wurde, sieht man, dass viele Personen nach der Einführung die nächste Veranstaltung in *stage1* ebenfalls besuchen. In der Ansicht ist auch zu sehen, dass am gleiche Tag von vielen zusätzlich auch eine Veranstaltung mit dem Thema *culture* besucht wird. Durch Auswählen der entsprechenden Ereignisse in einer Sequenz kann dieses Muster wieder in eine Suchanfrage umgesetzt werden. Hier muss lediglich vorher noch das *topic* ausgewählt werden, damit das dritte Ereignis auch die Restriktion *culture* bekommt. Da uns in diesem Fall nur die Ereignisse am Eröffnungstag interessieren, wird das dritte Ereignis noch manuell mit einer Restriktion versehen. Als Ergebnis dieser Anfrage werden alle Muster angezeigt, die auch zuvor in den Daten zu erkennen waren.

7.1.2. VAST Challenge 2014

Bei der VAST Challenge 2014 ging es darum anhand der Daten ein Verbrechen aufzudecken. In einem Anwendungsfall von VESPa wurde gezeigt, dass es eine typische Alltagsroutine gibt, die von vielen Personen durchlaufen wird. In diesem Szenario wird gezeigt, dass die beschriebene Alltagsroutine sehr schnell entdeckt und modelliert werden kann.

Integration und Anpassung der Daten

Für die Integration des Datenmodells muss wieder zunächst die Interfaces des Datenmodelle implementiert werden. Im Datenmodell entsprechen die Fahrzeugbesitzer den ActualPathOwnern im Datenmodell und die Aufenthalte an einem Ort zwischen den Fahrzeiten ergeben die

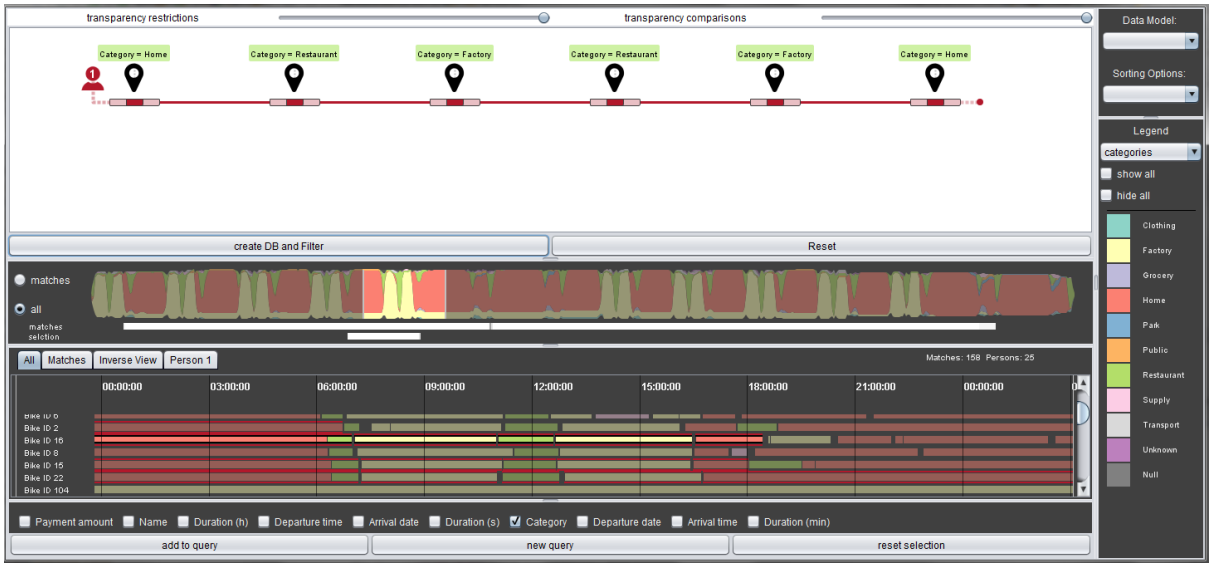


Abbildung 7.3.: Ansicht des kompletten Systems nach dem Analyse-Schritt.

ActualStates. Diese sind wieder mit weiteren Informationen angereichert. In diesem Fall sind es die Ausgaben die an einem Ort getätigt wurden. Als Filter-Attribute wurde diesen Mal für die Fahrzeugbesitzer *Bike ID* ausgewählt und für die Ereignisse werden *Payment Amount*, *Name* und *Category* verwendet. Für das Color Mapping wird dieses Mal die category verwendet.

Analyse

Im Overview-Panel ist wieder eine Übersicht der Ereignisse zu sehen(siehe Abbildung 7.3). Aus den ersten Blick ist dadurch erkennbar, dass es hier zwei Wochen dargestellt werden, weil der typische Tagesablauf zu sehen ist. Diese Alltagsroutine kann einfach aus den Daten selektiert werden und mit einer kleinen Einstellungsänderung im Analyse-Panel als Anfrage formuliert werden. Die komplette Extraktion der Alltagsroutine ist somit sehr schnell realisierbar.

7.2. Benutzerstudie

Wie auch bei VESPa soll die neue visuelle Anfragesprache mit einer Benutzerstudie evaluiert werden. Dabei soll getestet werden, ob die Teilnehmer der Studie nach einer kurzen Einführung in die Sprache und kleinen Aufwärmaufgaben bereits in der Lage sind, Anfragegraphen zu lesen, zu verstehen und, auf Basis einer umgangssprachlichen Anfragebeschreibung, Graphen eigenständig zu erstellen.

Dabei soll im Speziellen auch untersucht werden, ob die Verständnisprobleme, die in der Benutzerstudie von VESPa entdeckt wurden, in der weiterentwickelten Anfragesprache weniger

geworden oder sogar komplett verschwunden sind. Da die Sprache durch Erweiterungen allerdings auch umfangreicher geworden ist und sich die Darstellung mancher Konzepte und Komponenten verändert hat, kann es sein, dass neue Verständnisprobleme hinzugekommen sind. Dennoch ist die Erwartungshaltung für die Studie, dass die Anfragesprache weiterhin intuitiv ist und die sequentielle Darstellung sowie die neuen Timeline-Komponenten dazu beitragen, dass die Teilnehmer weniger Probleme mit der Interpretation der Ereignisse haben als zuvor.

7.2.1. Studienaufbau

Die gesamte Studie wurde in einem kleinen Büro mit einem Desktop-PC als Arbeitsplatz durchgeführt. Alle Dokumente, wie die Einführung, die Fragebögen und die Aufgabenstellung, wurden in ausgedruckter Form zu Beginn der Studie vorgelegt. Als Datensatz wurde bei allen Beispielen der Republica-Datensatz verwendet, da das Konferenzszenario einfach zu verstehen ist.

Die gesamte Studie setzt sich aus vier Teilen zusammen: Einer Einführung in das Thema und die Anfragesprache, einem Teil mit Verständnisaufgaben, einem anderen Teil aus Kompositionsaufgaben und einer abschließenden Bewertung. Dabei wurden allen Teilnehmern die gleichen Aufgaben in der gleichen Reihenfolge gestellt. Wie gut die Teilnehmer die Komponenten verstanden haben wurde anhand der Fehler, die sie während der Bearbeitung der Aufgaben gemacht haben, gemessen. Die genaue Aufgabenstellung sowie alle Dokumente können im Anhang A eingesehen werden.

Einführung

Die Einführung zu Beginn der Studie wurde in schriftlicher Form gegeben, sodass alle Teilnehmer die gleichen Informationen und Hilfsmittel zur Verfügung hatten. Es war erlaubt, diese Dokumente während der gesamten Studie als Nachschlagewerk zu verwenden. Nachfragen waren in den meisten Fällen nur zur Aufgabenstellung und dem Verlauf der Studie erlaubt. In der eigentlichen Einführung wurde dann das Thema sowie das Hintergrundwissen für die Studie eingeführt und die Anfragesprache anhand von konkreten Beispielen aus der verwendeten Datensammlung erklärt. Ebenfalls enthalten war eine schriftliche Kurzbeschreibung aller visuellen Komponenten, die während der Studie als Orientierungshilfe dienen sollte. Die Komponenten wurden dabei bewusst nicht vollständig definiert um zu sehen wie die Teilnehmer bestimmte Komponenten interpretieren.

Verständnisaufgaben

Bei den Verständnisaufgaben wurden die Nutzer aktiv dazu aufgefordert, die vorliegenden Anfragegraphen verbal zu beschreiben. Damit die Teilnehmer ein Gefühl dafür bekommen, wie

solche Beschreibungen formuliert werden können, wurden direkt vor diesem Aufgabenblock zwei Aufwärmaufgaben gestellt. Dort wurde den Teilnehmern jeweils ein Anfragegraph gezeigt und zwei mögliche Beschreibungen vorgelegt, wovon eine korrekt und eine falsch war. Die Aufgabe der Teilnehmer war es zu bestimmen, welche der beiden Beschreibungen zu dem Anfragegraph passt.

Der Großteil der Verständnisaufgaben bestand darin, Anfragegraphen frei zu beschreiben. Um eine gewisse Vergleichbarkeit der Antworten zu erreichen gab es für jeden Graphen eine Liste mit Punkten, die mit den Beschreibungen der Teilnehmer abgedeckt werden sollten. Diese Liste war nur für den Studienleiter sichtbar und konnte somit nicht von den Teilnehmern als Hilfestellung verwendet werden. Waren am Ende der Beschreibung noch nicht alle Punkte der Liste genannt worden, wurden zu diesen Aspekten noch Fragen gestellt. Der Studienleiter hat dabei jede gegebene Antwort akzeptiert und nur Nachfragen gestellt, falls der Teilnehmer keine eindeutige Antwort auf die gestellte Frage gegeben hat. Den Teilnehmern wurde dieses Vorgehen zuvor mitgeteilt, sodass sie von den Nachfragen möglichst nicht irritiert oder verunsichert wurden. Am Ende der Verständnisaufgaben gab es noch eine Reihe von Multiple-Choice-Aufgaben, mit denen Details der Anfragesprache abgefragt wurden.

Kompositionsaufgaben

Bei dem zweiten Teil der Studie ging es dann darum, eigenständig mit der Anfragesprache zu arbeiten. Auch hier gab es wieder eine Aufwärmaufgabe, um mit der Anwendung vertraut zu werden. Die Teilnehmer wurden dieses Mal dazu aufgefordert, einen umfangreichen Graphen nachzubauen, der alle visuellen Komponenten enthält, sodass die komplette Benutzeroberfläche und die Bedienung der Anwendung eigenständig erkundet werden konnte. Fragen zur Anwendung waren dabei zwar erlaubt, sollten aber erst gestellt werden, wenn die Teilnehmer selbst nicht weiter kamen. Anschließend wurden dann die Beschreibungen gegeben, die es nachzubauen galt. Es gab jeweils eine leichte, mittlere und schwere Aufgabe. Die Teilnehmer wurden dieses Mal aktiv dazu aufgefordert Fragen zur Bedienung zu stellen, da es hierbei ausschließlich um die Bearbeitung der Aufgaben und nicht um die Bedienung der Anwendung ging.

Fragebogen

Zum Schluss der Studie wurden die Teilnehmer mit einem Fragebogen noch um eine Bewertung der Anfragesprache gebeten. Damit die Teilnehmer ihre Leistung und ihr Verständnis besser einschätzen konnten, wurden davor kurz die Fehler, die sie in der Studie gemacht hatten besprochen. Die Teilnehmer konnten so besser beurteilen, wie gut sie die Konzepte während des Studienverlaufs interpretiert hatten. Insgesamt wurden Fragen zur subjektiven Wahrnehmung der Teilnehmer, zur empfundenen Komplexität und dem persönlichen Verständnis des Konzepts gestellt.

Zielgruppe

Die Anfragesprache soll sowohl für erfahrene Nutzer, die sich bereits im Datenbankbereich auskennen, als auch für Domänenexperten der Bewegungsanalyse geeignet sein, die nur grundlegende EDV-Kenntnisse besitzen. Für diese Studie wurden somit Studenten im Informatikbereich ausgewählt, da bei diesem Personenkreis davon ausgegangen werden kann, dass die oben genannten Grundkenntnisse vorhanden sind und die Teilnehmer somit der potentiellen Zielgruppe entsprechen. Des Weiteren sind Studierende im Informatikbereich noch keine Experten im Datenbankgebiet, sodass sich die Ergebnisse auch auf andere Nutzer übertragen lassen. Konkret gab es in der Studie insgesamt 10 Teilnehmer im Alter zwischen 21 und 27, die entweder Softwaretechnik oder Informatik an der Universität Stuttgart studieren.

7.2.2. Ergebnisse

In diesem Unterkapitel werden nun die Ergebnisse der Studie vorgestellt. Die Bewertung des Verständnisses der Teilnehmer wurde dabei, wie bereits erwähnt, anhand der Fehler, die sie während der Aufgaben gemacht haben, gemessen. Dabei war nicht die Häufigkeit der Fehler entscheidend, sondern in welchen Bereichen diese gemacht wurden. Zudem kann eine inkonsequente Interpretation einzelner Komponenten darauf hinweisen, dass das Konzept entweder fehleranfällig oder sehr komplex ist.

Verständnisaufgaben

Durch die Verständnisaufgaben wurde zunächst untersucht, ob die Teilnehmer in der Lage waren, die Graphen zu lesen und zu interpretieren. Obwohl die Aufwärmaufgaben nicht in die Bewertung des Verständnisses miteinbezogen wurden, ist erwähnenswert, dass diese von fast allen Teilnehmern komplett richtig beantwortet werden konnten. Nur ein Teilnehmer hatte hier einen Graphen falsch interpretiert. Für die Auswertung der Beschreibungen wurde für jeden Teilnehmer bestimmt, in welchen Bereichen er Fehler begangen hatte und um welchen konkreten Fehler es sich dabei handelte. Die Ergebnisse dieser Auswertung sind in den beiden Diagrammen 7.4 und 7.5 zu sehen. Die Hauptfehlerquellen werden nun im Einzelnen besprochen.

Wie in 7.5 zu sehen ist, gab es die größten Probleme bei der Interpretation der Anzahl der Schleifendurchläufe. Nur drei Teilnehmer entschieden sich hier für die vorgesehene Definition. Alle anderen Teilnehmer zählten die Schleifen genau um einen Durchgang versetzt und dachten somit die Angaben auf der Schleife bezögen sich auf die Anzahl der gesamten Durchgänge. Abgesehen davon wurde das Konzept der Schleifen allerdings sehr gut verstanden. Alle Teilnehmer erkannten, dass Ereignisse in einer Schleife nur Platzhalter mit den angegebenen Einschränkungen sind und in jedem Schleifendurchlauf auf unterschiedliche Ereignisse in den Daten abgebildet werden können. Zudem räumten die meisten Teilnehmer im Nachhinein

ein, dass die eigentliche Definition der Schleifendurchgänge besser zur Visualisierung der Komponente passt, als ihre eigene Interpretation.

Obwohl es weniger Teilnehmer gab, die mit den Verzweigungen Probleme hatten, sind die Fehler die dort gemacht wurden schwerwiegender. Zwei Teilnehmer hatten den Sinn hinter der Verzweigung nicht verstanden und hatten entsprechend alle Aufgaben diesbezüglich falsch interpretiert, zwei weitere Teilnehmer hatten teilweise Probleme damit gehabt. Hierbei wurde überwiegend angenommen, dass Akteure sich ab der Verzweigung trennen und unabhängig voneinander in unterschiedliche Zweige gehen können.

Eine weitere große Fehlerquelle waren die Timeline-Komponenten, die eigentlich zu einem besseren Verständnis der Ereignisse beitragen sollten. Im Speziellen wurde dabei die maximale und minimale gemeinsame Aufenthaltsdauer von vielen Teilnehmern nicht immer richtig interpretiert. Zudem konnten nur wenige Teilnehmer auf Basis dieser Angaben Schlüsse zu den individuellen Aufenthalten einzelner Personen ziehen. Weitere Fehlerquellen in diesem Bereich gab es bei den Vorher-Nachher-Beziehungen zwischen den Akteuren. Diese wurden nicht immer sofort richtig erkannt und die Teilnehmer brauchten überdurchschnittlich viel Zeit für die Beschreibungen. Bei den kleineren Beispielen, in denen nur wenige Restriktionen verwendet wurden, kamen die Antworten jedoch schneller und häufiger richtig.

Bei den Ereignissen, die bereits auch in VESPa falsch interpretiert wurden, hatten viele Probleme bei der Definition, dass Personen die kein gemeinsames Ereignis haben sich auch nicht treffen können. Allerdings verstand jeder Teilnehmer dieses Mal, dass Ereignisse, die zwar am selben Ort, aber zu unterschiedlichen Zeiten stattfinden, auf verschiedene Ereigniskonten abgebildet werden und auch den Positionen der Komponenten und der Länge der Kanten wurden überwiegend keine Bedeutung zugewiesen. Nur ein Teilnehmer nahm hierbei an, dass ein Ereigniss, das rechts von einem anderen Ereignis liegt, später als dieses stattfinden muss. Ein anderes Problem in diesem Zusammenhang war die Interpretation von Kanten als eine zwangsläufig räumliche Bewegung. So dachten zwei Teilnehmer, dass in einem Muster, bei dem

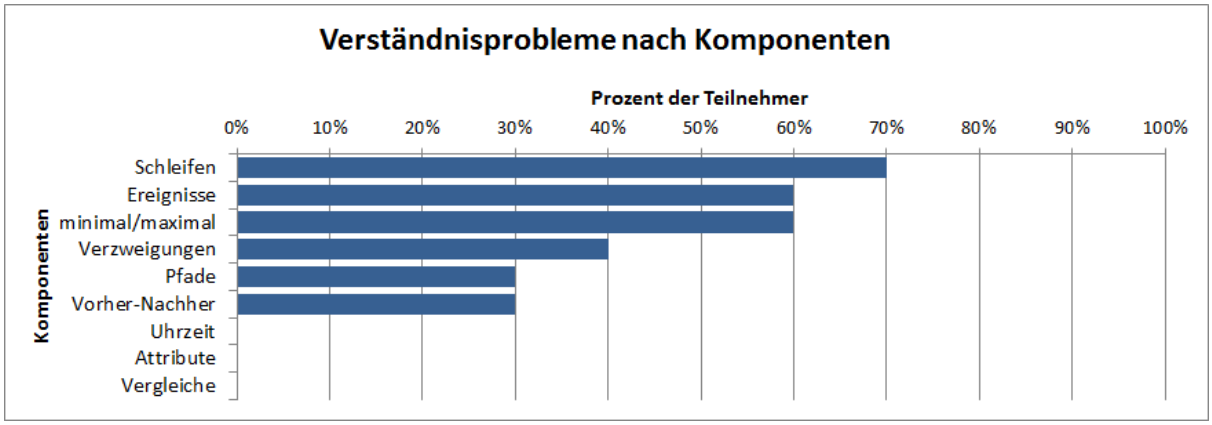


Abbildung 7.4.: Diagramm mit den Komponenten

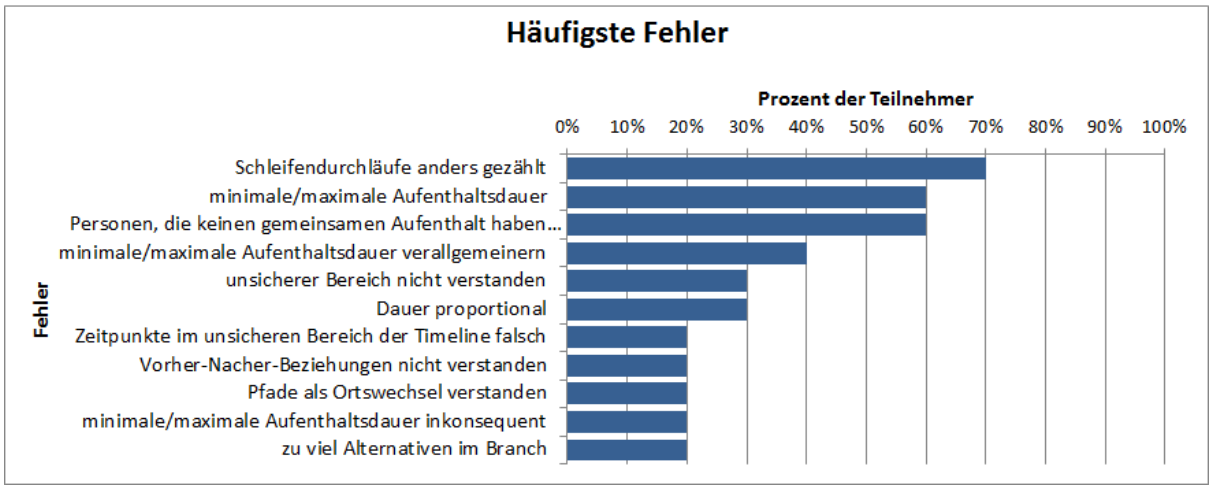


Abbildung 7.5.: Diagramm mit den häufigsten Fehlern, die bei den Verständnisaufgaben gemacht wurden. Hier ist die Prozentzahl der Teilnehmer angegeben, die einen der aufgelisteten Fehler mindestens einmal gemacht hatten.

zwei Ereignisse am selben Ort hintereinander stattfinden, der Akteur diesen Ort zwangsläufig immer erst verlassen muss, um dann wieder zurückzukommen.

Trotz der vielen Fehler, die im vorigen Teil beschrieben wurden, wurde die visuelle Anfragesprache weitestgehend gut verstanden. Schwerwiegende Fehler wie die falsche Interpretation der Verzweigungen wurden nur von wenigen Teilnehmern gemacht und die häufigsten Fehler traten in Bereichen auf, die ein tiefes Verständnis für die Anfragesprache voraussetzen. Im Bereich der Vergleiche, Attribute und der Uhrzeitangaben wurden zudem keine Fehler gemacht und fast jeder Teilnehmer konnte auch für jeden Graphen eine Beschreibung geben, die grundsätzlich korrekt war und nur im Detail nicht ganz den Definitionen entsprach.

Kompositionsaufgaben

Bei den Kompositionsaufgaben konnten alle Teilnehmer alle Aufgaben gut lösen und hatten selbst bei der schwersten Aufgabe einen Graphen erzeugt, der der Lösung sehr ähnlich war (siehe 7.6).

Bei der ersten Aufgabe wurden nur von drei Teilnehmern kleine Fehler gemacht, die vermutlich aus einem kleinen Interpreationsspielraum der Frage beruhen, da alle Teilnehmer die gleiche alternative Lösung hatten. Auch bei der zweiten Aufgabe haben alle Teilnehmer erkannt, dass Zeiteinstellungen gemacht werden mussten, konnten diese aber teilweise nicht korrekt umsetzen. Der restliche Anfragegraph war bei allen korrekt. Bei der schwersten Aufgabe gab es wie zu erwarten war am meisten Probleme. Dieser Graph war sehr umfangreich es gab viele Details die es zu beachten gab. Dennoch wurde auf diese Aufgabe von vier Teilnehmern

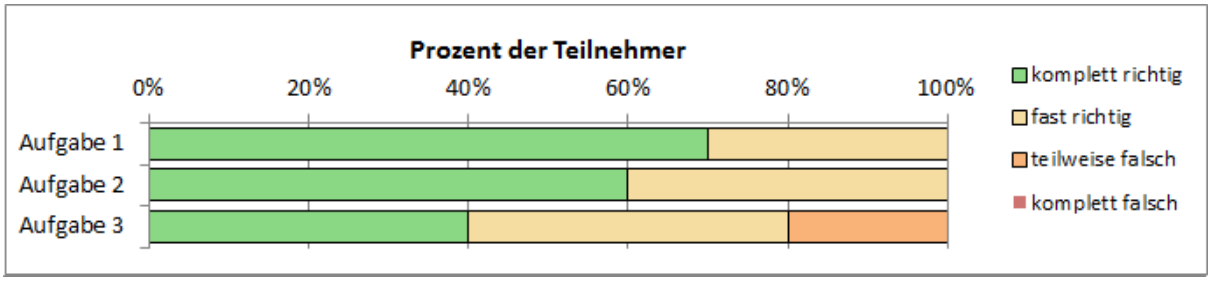


Abbildung 7.6.

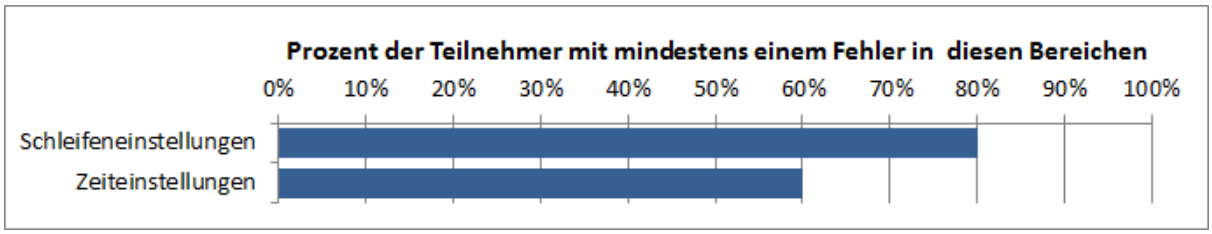


Abbildung 7.7.

komplett und von weiteren vier Teilnehmern fast richtig beantwortet. Nur zwei Teilnehmer hatten hier an mehreren Stellen Probleme.

Die häufigsten Fehler während der Kompositionsaufgaben wurden wieder bei der Anzahl der Schleifendurchgänge gemacht (siehe 7.7). Alle Teilnehmer die auch bereits im vorherigen Aufgabenteil die Schleife anders interpretiert haben, haben dies auch bei dieser Aufgabe gemacht. Darüber hinaus wurden allerdings auch die Zeiten manchmal falsch und manchmal richtig eingestellt. Vielleicht sind diese Fehler aus Unachtsamkeit oder aus einer fehlerhaften Bedienung der Anwendung entstanden.

Insgesamt hat jeder die Kernaussage der Anfragen erkannt und konnte diese auch in der Anfragesprache umsetzen. Lediglich bei den Details wie beim Einstellen der exakten Dauer oder den Schleifendurchläufen gab es Probleme.

Bewertung der Teilnehmer

Beim der anschließenden Bewertung wurde die Anfragesprache überwiegend positiv aufgenommen und als intuitiv und einfach empfunden. Ein großer Teil der Teilnehmer fand die visuelle Anfragesprache zudem ansprechend und würde sie auch wieder verwenden wollen. Auf die abschließende Frage der Studie „Wenn Sie Anfragen an eine Datensammlung von Ereignissequenzen stellen müssten, würden Sie diese Anfragesprache gegenüber herkömmlichen Methoden vorziehen? Begründen Sie Ihre Antwort kurz.“ wurden überwiegend bejahende Antworten gegeben, unter anderem: „Ja. Es ist ziemlich intuitiv und nicht schwer zu benutzen.“, „Ja, besserer Überblick, Graph lässt sich mit herkömmlichen textuellen Methoden nicht

übersichtlich bearbeiten.“, „Ja, weil textuelle Anfragesprachen zu komplizierten Anfrageketten führen, bei deren man zu leicht die Übersicht verliert“. Andere Teilnehmer waren eher neutral gestimmt: „Kommt auf die Komplexität und Länge der Anfrage an. Vermutlich herkömmliche Methode“, „vllt, wenn Anfragen komplex sind“. Und nur ein Teilnehmer würde textuelle Methoden bevorzugen: „Nein, ich würde lieber tippen (reiner Text, kompatibler, expliziter)“.

Positiv lässt sich zum Schluss noch anmerken, dass bei der Bewertung 9 der 10 Teilnehmer am Ende der Studie angaben die Konzepte nach der Endbesprechnung verstanden zu haben und die Definitionen — auch die der Schleifendurchgänge — für logisch halten.

Diskussion

Im direkten Vergleich zu VESPa ist eine Verbesserung bei der Interpretation der Ereignisse im Bezug auf die zeitliche Komponente der Ereignisse erkennbar. Dies lässt sich vermutlich sowohl auf die neue sequentielle Anordnung der Ereignisse als auch auf die neuen Timeline-Komponenten zurückführen, die nochmal visuell auf einen zeitlichen Aspekt hinweisen. Da mit der Einführung der Timeline-Komponente aber auch die Komplexität in diesem Bereich gestiegen ist, wurden die zeitlichen Restriktionen nicht immer richtig erkannt. Im realen Anwendungsfall gehören zeitliche Restriktionen allerdings eher zu Spezialfällen und sind vor allem nicht in einem solchen geballten Form wie in der Studie vorhanden. Besonders bei den kleinen Beispielen hat sich gezeigt, dass fast alle Teilnehmer die richtige Bedeutung der Komponente erkennen, wenn nur wenige Restriktionen vorhanden sind. In realen Anfragen sollte dies somit nicht zu Problemen führen.

Im Großen und Ganzen lässt sich die anfängliche Annahme der Studie bestätigen. Die Nutzer waren bereits nach einer kurzen Einarbeitungsphase in der Lage mit der visuellen Anfragesprache zu arbeiten und konnten zu mindestens einfache Aufgaben damit lösen. Somit wurden die Grundprinzipien von allen Teilnehmern weitestgehend verstanden und nur die sehr speziellen Komponenten wie die Einstellung von Zeitdauer und die Vorher-Nachher-Beziehungenen waren für manche Teilnehmer schwer zu verstehen.

8. Diskussion

Nachdem in der Evaluation die Benutzbarkeit der Anwendung untersucht wurde, werden in diesem Kapitel die Unterschiede zu VESPa herausgearbeitet. Da nach wie vor nicht alle Anfragen möglich sind und auch das Analysesystem nicht in der Lage ist alle Aspekte der Daten ausreichend zu visualisieren, werden anschließend noch die Grenzen beider Konzepte aufgezeigt und die Skalierbarkeit des Systems diskutiert.

8.1. Vergleich zu VESPa

Als Weiterentwicklung von VESPa adressiert die neue Anfragesprache die gleiche Problemstellung und lässt sich somit gut direkt mit diesem Konzept vergleichen. Da diese Arbeit zusätzlich auch direkt auf der Implementierung von VESPa aufbaut, sind viele Aspekte komplett übernommen worden und die Funktionalität von VESPa ist auch mit der neuen Anfragesprache möglich. Zu den Gemeinsamkeiten der Sprachen gehören gehört der graphbasierte Aufbau der Sprache, das Konzept der allgemeinen Restriktionen und die Vergleiche. Doch in wichtigen Bereich wie bei der Definition von Ereignissen in der Sprache und dem Umgang mit zeitlichen Restriktionen unterscheiden sich die beiden Konzepte.

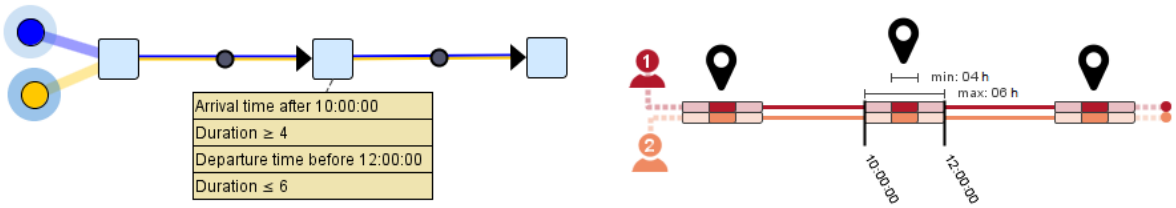


Abbildung 8.1.: Gegenüberstellung der beiden Anfragesprachen an einem Beispiel mit gleicher Semantik.

Im Vergleich zu VESPa zeichnet sich die neue Anfragesprache vor allem durch den stärkeren visuellen Bezug zu dem Aufbau der Daten und der zeitlichen Komponente der Ereignissequenzen aus. Dieses Konzept wird zum einen durch die Einführung der Timeline als Ergänzung für das Ereignissymbol, sowie durch die sequentielle Anordnung der Daten umgesetzt. Den Nutzern der Sprache wird somit eine visuelle Hilfestellung bei der Interpretation der Ereignisse gegeben. Im Gegensatz dazu findet in VESPa keine besondere Behandlung von zeitlichen

Komponenten statt. Zeitliche Restriktionen werden wie andere Attribute behandelt und auch so im Anfragegraphen dargestellt. Stellt man die beiden Entwürfe direkt gegenüber, wie in Abbildung 8.1 zu sehen, so ist bei VESPa nicht auf den ersten Blick erkennbar, dass es sich bei den Restriktionen des zweiten Ereignisses um zeitliche Restriktionen handelt. In der neuen Anfragesprache hingegen ist sofort sichtbar, dass zeitliche Restriktionen in der Anfrage enthalten sind. Diesbezüglich konnte auch die Benutzerstudie zeigen, dass mit der neuen Timeline-Komponente die Interpretation der Ereignisse leichter gefallen ist und diese nicht nur mit einem Ort verbunden wurden.

Neben der Erweiterung um eine visuelle zeitliche Komponente wurde die Definition der Ereignisse in der Anfrage zusätzlich noch überarbeitet. In VESPa wurde der Ereignisknoten (event node) als Ort zu einer bestimmten Zeit definiert. Zusätzlich wurde von Haag et al. festgelegt, dass zwei Ereignisknoten einer Anfrage sich nicht auf Ereignisse beziehen können, die am gleichen Ort zu überlappenden Zeiten stattfinden [HKE16]. Während die erste Definition auch für die neue Anfragesprache gilt — mit dem Zusatz, dass eine Kombination aus Kontext und Ort ein Ereignis bestimmt — wurde die zweite Definition verallgemeinert. In der neuen Anfragesprache ist es sehr wohl möglich und gewollt dass sich Ereignisse teilweise überlappen können. Zudem kann ein individuelles Ereignis einer Person auch auf mehrere gemeinsame Ereignisse in der Anfrage abgebildet werden. Die neue Bedingung sieht nur vor, dass keine Überlappung aller individuellen Ereignisse zweier Ereignisknoten erlaubt ist. Dies ist zum Beispiel der Fall wenn sich eine Person eine längere Zeit an einem Ort befindet und nacheinander verschiedene Personen trifft.

Darüber hinaus wurde die Funktionalität um die erweiterten Pfadkomponenten, wie die Verzweigung und die Wiederholungsschleife ergänzt, die nochmal mehr Möglichkeiten bei der Modellierung erlauben. Diese ganzen Erweiterungen haben allerdings auch dazu geführt, dass neue Konzept auch sehr viel umfangreicher geworden ist und die Nutzer mehr verschiedene Konzepte erlernen müssen. Die Darstellung der Timeline ist sehr komprimiert und somit kann auf einem relativ kleinen Raum sehr viel Information enthalten sein. Die komplette Bedeutung ist somit nicht immer einfach zu überblicken So zeigte sich auch in der Benutzerstudie, dass nicht alle Teilnehmer die verschiedenen Timeline-Aspekte auf Anhieb richtig interpretieren konnten und überdurchschnittlich viel Zeit benötigt haben, um die einzelnen zeitlichen Restriktionen herauszulesen. Auch bei den Verzweigungen und Wiederholungsschleifen wurden häufig Fehler gemacht, die sich größtenteils auf nicht eindeutige Visualisierungen der Komponenten zurückführen lassen. VESPa hingegen ist wesentlich reduzierter und überfordert dadurch die Nutzer der Sprache weniger.

8.2. Funktionalität und deren Grenzen

Da die Anfragesprache einfach gehalten sein sollte, wurden bezüglich der Funktionalität Grenzen gesetzt und nicht alle Sonderfälle berücksichtigt. In diesem Abschnitt werden diese Grenzen aufgezeigt.

Ein Problem der Anfragesprache sind zufällige Überschneidungen von Ereignissen. Ist einem Nutzer die zeitliche Beziehung der Akteure in manchen Bereichen des Graphen egal, so können Sequenzen mit sich zufällig überschneidenden Ereignissen nicht gefunden werden, obwohl sie für den Nutzer ebenfalls interessant sein können. Per Definition werden überlappende Ereignisse immer auf einen gemeinsamen Ereignisknoten abgebildet. Es gibt somit keine Möglichkeit die Beziehung zwischen Ereignisknoten nicht zu definieren.

Zudem gibt es Probleme, wenn die Daten nicht richtig vor verarbeitet wurden und die Ereignisse nicht nach den Kriterien und Definitionen erzeugt wurden, die für die Unterscheidung der Ereignisse herangezogen werden. Wird ein logisches Ereignis beispielsweise in mehrere Ereignisse der gleichen Art aufgeteilt, entspricht dies einem anderen Muster, und kann nicht unbedingt durch einen Ereignisknoten gefunden werden. Kleine Messfehler, die zu Ereignissen führen die nicht vorhanden sind, können zwar durch die Kantentoleranz ausgeglichen werden, dadurch werden allerdings auch mehr Ergebnisse gefunden, die nicht der eigentlichen Bedeutung entsprechen, aber noch im Rahmen der Toleranz liegen. Auch bei den neu eingeführten Komponenten mussten Einschränkungen getroffen werden. Zum Beispiel ist es zwar möglich mit Verzweigungen die Anfrage ab einem bestimmten Punkt in zwei alternative Anfragen zu teilen, allerdings ist das spätere Zusammenführen dieser Alternativen nicht möglich. In diesem Prototyp werden auch die Anzahl der Verzweigungen, der Wiederholungsschleifendurchgänge und der Akteure aus Verarbeitungsgründen begrenzt.

Im Analysesystem werden verschiedene Möglichkeiten geboten, die Datensammlung und die Ergebnisse zu visualisieren und sich auf bestimmte Bereiche zu fokusieren. Allerdings gehen bei der Darstellung die Zusammenhänge zwischen den Akteuren, die gemeinsam in einem Ergebnis der Anfrage stehen, weitestgehend verloren. Es ist zwar erkennbar auf welchen Akteur eine Teilsequenz im Muster zutrifft, allerdings ist nicht sofort erkennbar welche Sequenzen zusammen in einem Ergebnis vorkommen. Diese Verbindung wird erst in den zusätzlichen Detail-Informationen sichtbar.

8.3. Skalierbarkeit des Systems

In diesem Abschnitt wird das System auf die Skalierbarkeit im Hinblick auf die Darstellung und Verarbeitung der Datensammlungen untersucht.

8.3.1. Visual Scalability

Mit der ThemeRiver Übersicht und dem Ausblenden von bestimmten Kategorien der Legende kann man auch bei großen Datenmengen einen Überblick über die Daten gewinnen und darauf basierend die Analyse gezielt starten. Die visuelle Skalierbarkeit im Hinblick auf die Datenmenge ist somit gegeben. Sind die Ereignissequenzen allerdings über einen sehr großen Zeitraum definiert und die zeitliche Granularität der Ereignisse im Vergleich sehr klein, passt

die Auflösung des Überblicks nicht zu den Daten. In diesem Fall müsste die Visualisierung zum Beispiel durch hierarchische Ansätze oder Zoom-Funktionen angepasst werden. Das Result-Panel selbst skaliert nicht mit der Datenmenge. Hier kann nur eine sehr kleine Anzahl an Ereignissequenzen gemeinsam dargestellt werden und auch die Anzahl der Ereignisse, die pro Sequenz dargestellt werden, wird begrenzt durch die Auflösung des Displays. Sobald die Zeitspannen nur noch wenige Pixel breit auf dem Display dargestellt werden, sind diese fast nicht mehr erkennbar. In Kombination mit dem ThemeRiver und Sortierungsmethoden kann damit dennoch auch mit großen Datensammlungen gearbeitet werden.

Auch die Anfragesprache selbst ist hingegen nur bedingt visuell skalierbar, da ab einer gewissen Anzahl von Knoten und Kanten zum Einen der Graph größer als das Display wird und so nicht mehr dargestellt werden kann. Visual Clutter stellt ebenfalls ein Problem dar, wenn viele gemeinsame Ereignisse mit jeweils unterschiedlichen Teilmengen der Akteure eingezeichnet werden sollen. In diesem Fall ist die Wahrscheinlichkeit hoch, dass sich visuelle Objekte schneiden. Die Timeline-Komponente ändert die Größe mit den Einstellungen, die damit getroffen werden und Ereignisknoten können somit zusätzlichen Platz einnehmen. Restriktionen und Vergleiche verstärken dieses Problem weiter. Durch das Ausblenden von bestimmten Gruppen von Komponenten kann dem allerdings entgegen gewirkt werden, sodass beim Modellieren des Graphen bestimmte Bereiche ausgeblendet werden können.

Die Farbgestaltung stellt ein weiteres Problem dar. Ab einer bestimmten Anzahl an Kategorien lassen sich die Farben nicht mehr unterscheiden oder müssen im Standardfarbschema auf die gleichen Farben abgebildet werden.

8.3.2. Computational Scalability

Je größer eine Datensammlung ist, desto mehr Verarbeitungsaufwand entsteht bei der Darstellung und Verarbeitung der Daten. Um den Aufwand bei der Ergebnisdarstellung der Daten im Result-Panel gering zu halten, werden nur Primitiven verarbeitet, die im Sichtbereich liegen. Da die Ereignisse sowohl horizontal als auch vertikal sortiert sind, ist das clippen einfach und effizient möglich. Zudem ist die Anzahl der darstellbaren Elemente in diesem Bereich des Systems vergleichsweise gering, sodass auch bei großen Datensammlungen nur eine begrenzte Anzahl an Daten gleichzeitig visualisiert werden kann. Beim Overview-Panel hingegen werden alle Datenelemente der Datensammlung gleichzeitig visualisiert. Da diese Anzeige allerdings weitestgehend statisch ist, muss das Aggregieren der Ereignisse und die Berechnung der Polygone nur bei Änderungen der Auflösung und bei der Durchführung von Anfragen neu berechnet werden. Der Aufwand dieser Berechnung steigt dabei linear mit der Zahl der einzelnen Ereignisse. Die ebenfalls enthaltenden Anzeigen skalieren analog dazu linear mit der Anzahl der Ergebnisereignisse und der Anzahl der selektierten Ereignisse. Der Zeichenaufwand der Visualisierung der Anfrage ist abhängig von der Anzahl der Akteure und Ereignisse. Da die Anfrage aufgrund der schlechten visuellen Skalierbarkeit allerdings nicht groß werden kann, stellt dies bei dem Prototypen kein Problem dar.

Das größte Problem bei der automatischen Auswertung der Daten sind die Datenbankanfragen und die damit verbundene Aufbereitung der Ergebnisse. Die Ergebnistabelle beinhaltet für jeden Akteur zwei Spalten und für jeden Aufenthalt an einem Knoten eine Spalte. Da dafür das kartesische Produkt dieser Tabellen berechnet werden muss, hat die Ergebnistabelle entsprechend viele Zeilen. Die Dauer der Ausführung hängt stark von den Restriktionen in der Anfrage ab. Werden nur die Restriktionen des Anfragepfades verwendet, bleibt die Ergebnismenge groß und die Enumeration der Ergebnisse dauert entsprechend lang. Entscheidend für die Ausführungsdauer einer Anweisung ist somit die Größe des Graphen und die zu erwartende Ergebnismenge und Umsetzung von Anfragen mit dem jetzigen Datenbankschema skaliert somit weder mit der Größe des Anfragegraphen noch mit der Datenmenge gut. Entsprechend hat dies auch Auswirkungen auf die Interaktivität des Systems.

9. Zusammenfassung und Ausblick

In dieser Arbeit wird eine weiterentwickelte visuelle Anfragesprache für Bewegungsdaten auf Basis einer bereits vorhanden Anfragesprache namens VESPa [HKE16] vorgestellt und beschrieben. Im Fokus stand dabei die Weiterentwicklung der Darstellung und der Umgang mit der zeitlichen Komponente, die in Bewegungsdaten eine große Rolle spielt. In der Anfragesprache wurde die Zeit darum an mehreren Stellen durch einen visuellen zeitlichen Bezug und spezielle neu eingeführte zeitliche Restriktionen besser repräsentiert. Zudem wurde auch die Definition der Ereignisknoten in der Anfrage überarbeitet, sodass mit der neuen Anfragesprache auch Ereignissequenzen gefunden werden können bei denen Mengen von sich teilweise überschneidenden Ereignissen gemeinsam gefunden werden können. Aber auch in anderen Bereichen wurde die Funktionalität durch die Einführung von Schleifen und Verzweigungen erweitert.

In Kombination mit dem Analysesystem kann die Anfragesprache nicht nur dazu verwendet werden einfache Anfragen zu formulieren, sondern dient auch als Abstraktion der Daten. Die Anfragesprache wird hier auch auf dem umgekehrten Weg verwendet, sodass es möglich ist aus einer kleinen Auswahl von Ereignissen der Datensammlung eine Anfrage genieren zu lassen, die wieder dazu verwendet werden kann die Datensammlung anzufragen. Das Analysesystem, das für den Zweck der interaktiven Bewegungsdatenanalyse erstellt wurde, bietet einige Möglichkeiten um in Kombination mit dieser Funktion einen einfachen iterativen Analyseprozess zu ermöglichen.

In der Evaluation hat sich gezeigt, dass das Konzept der Anfragesprache von den Benutzern in den Grundzügen sehr gut verstanden wurde. Bei bestimmten Komponenten gab es allerdings noch Schwierigkeiten beim Verständnis der Bedeutung. Insgesamt haben die meisten Nutzer die Anfragesprache überwiegend positiv bewertet und würden diese auch in Zukunft für Anfragen an Bewegungsdaten verwenden und wenn möglich anderen textuellen Methoden vorziehen.

Ausblick

Wie aus der Diskussion und der Benutzerstudie hervorgeht, gibt es einige Bereiche an denen in der Zukunft weiterentwickelt werden kann. So hat sich in der Evaluation gezeigt, dass einige Komponenten nicht bei allen Teilnehmern der Studie gleich interpretiert wurden. Dieser Interpretationsspielraum lässt sich teilweise auch auf die Darstellung der Komponenten

zurückführen und könnte durch eine Überarbeitung der visuellen Komponente besser von den Nutzern angenommen werden.

Und auch im Bereich der Datenverwaltung und der Datenbankabfrage gibt es Verbesserungsmöglichkeiten, sodass auch umfangreiche und komplexe Anfragen in einer Zeit abgearbeitet werden können die noch eine interaktive Analyse der Daten ermöglicht. Dazu könnte in Betracht gezogen werden, das zugrundeliegende Datenbankschema durch ein anderes zu ersetzen und auch ein Wechsel zu einer Graphdatenbank könnte zu Vorteilen bei der Performanz führen.

A. Material der Benutzerstudie

Auf den folgenden Seiten sind die Dokumente der Benutzerstudie abgebildet.

Einleitung

Was für eine Anfragesprache?

In dieser Studie geht es um eine visuelle Anfragesprache für Ereignissequenzen wie zum Beispiel Bewegungsdaten. Mit dieser Anfragesprache sollen Muster definiert werden können nach denen dann in der Datensammlung gesucht werden kann. Der Fokus liegt dabei auf Personengruppen die bestimmte Ereignisse nacheinander besuchen und sich zwischendurch an verschiedenen Orten begegnen können.

Beispiele für Anfragen:

- Gibt es 2 Personen die sich im Park treffen und dann anschließend zusammen in einem Café sind?
- Gibt es eine männliche Person die erst nach 23 Uhr das Bürogebäude verlässt und anschließend nicht nach Hause geht?

Ziel dieser Anfragesprache soll es sein solche Anfragen in ein konkretes Muster in Form eines Graphen zu überführen um damit eine Datensammlung nach Personen abzusuchen die genau diesem Muster entsprechen.

Welcher Datensatz?

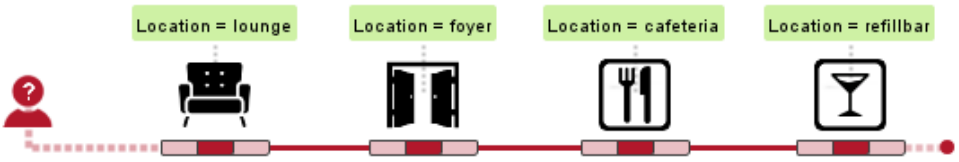
Die Datensammlung die für diese Studie verwendet wird beinhaltet echte Bewegungsdaten von Besuchern einer Konferenz namens *re:publica*. Die Besucher konnten sich frei auf dem Konferenzgelände bewegen und wurden dabei anonymisiert getrackt indem in regelmäßigen Abständen die Position der Besucher bestimmt wurde. So wurde für jeden Besucher eine Abfolge von verschieden Orten an denen er sich aufgehalten hat zusammen mit Informationen zu dort stattfindenden Ereignissen (Vorträge, Veranstaltungen, Workshops,…) abgespeichert.

In allen kommenden Aufgaben werden Muster aus diesem Datensatz dargestellt.

Die Anfragesprache

Eine Anfrage entspricht einem Graphen der als Muster für eine Ereignissequenz dient. Es gibt Knoten die Platzhalter für bestimmte Personen oder Ereignisse darstellen und Kanten die die Reihenfolge zwischen Ereignissen definieren. Für den einfachsten Fall mit einer gesuchten Person entsteht ein einfacher Pfad der alle Ereignisse verbindet. Der folgende Graph definiert somit folgende Anfrage:

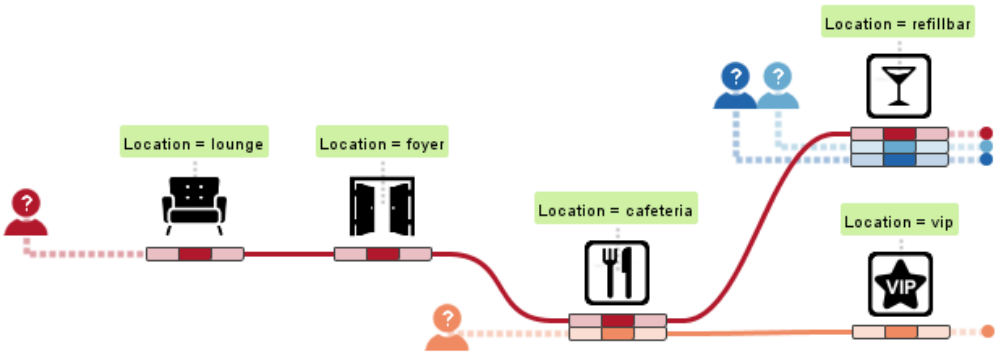
Gibt es eine Person die erst ein Ereignis in der Lounge hatte und anschließend 3 Ereignisse im Foyer, der Cafeteria und der Refillbar besucht hat?



Es entsteht ein einfacher Graph der, von links nach rechts gelesen, eine Ereignisreihenfolge definiert. Die Kanten dienen nur als Übergänge und bedeuten nicht, dass sich die Person von einem Ort zum anderen bewegt sondern zeigen nur an dass hier ein Übergang zwischen 2 Ereignissen stattfindet.

In Kombination mit mehreren Personen und Ereignissen die sich überschneiden können auch kompliziertere Anfragen gestellt werden. Immer wenn mehrere Personen sich zu einer Zeit gleichzeitig an einem Ereignis befinden, teilen sie sich ein gemeinsames Ereignis. Der nachfolgende Graph erweitert das Muster von oben um mehrere Treffen mit anderen Personen. Der Graph hat folgende Bedeutung:

Gibt es 4 Personen, wovon einer (rot) zuerst in der Lounge und im Foyer war und sich anschließend mit einem anderen (orange) der 4 Personen in der Cafeteria betroffen hat und am Ende mit den beiden anderen (hellblau und dunkelblau) in der Refillbar war, während der 4. (orange) in den VIP-Bereich gegangen ist?

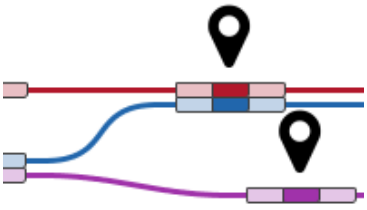


Zusätzliche Komponenten wie Schleifen, Verzweigungen, Einschränkungen, Zeiteinstellungen und Attribute erweitern dieses Konzept. Die einzelnen Komponenten sind auf den nächsten 2 Seiten aufgelistet. Diese Übersicht kann während der gesamten Studie als Nachschlagewerk verwendet werden.

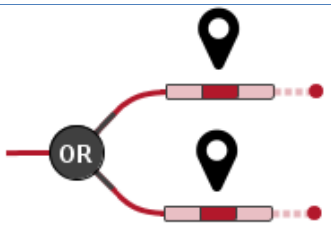
Komponenten

Akteure *(person)*
Der Akteur ist ein Platzhalter für eine bestimmte Person im Muster. Er markiert den Startpunkt eines Pfades in der Anfrage.
Von dort aus wird durch Verbinden mit verschiedenen Ereignissen festgelegt welche und in welcher Reihenfolge Ereignisse besucht werden sollen.

Ereignis *(event)*
Der Ereignis Knoten ist ein Platzhalter für ein bestimmtes Ereignis. Ein Ereignis ist eine Veranstaltung oder ein Treffen von einer oder mehreren Personen an einem bestimmten Ort zu einer bestimmten Uhrzeit. Wenn 2 Person zur gleichen Zeit an einem Ort sind teilen sie sich ein gemeinsames Ereignis.

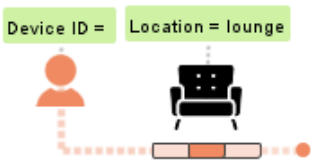


Übergänge
Übergänge zwischen Ereignissen legen die Reihenfolge der Ereignisse fest. Befinden sich die Ereignisse an unterschiedlichen Orten, können sie als Weg interpretiert werden. Der Pfad einer Person ist durchgehend, ohne Unterbrechungen und besitzt die gleiche Farbe wie die zugehörige Person. Die Länge des Pfades hat keine Bedeutung und stellt nicht immer eine räumliche Bewegung dar.

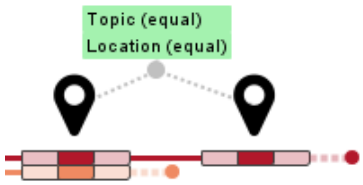


Verzweigungen *(branch)*
Verzweigungen können eingesetzt werden um Alternativen zu definieren. Ein oder mehrere Personen können entweder dem einen oder dem anderen Pfad folgen. Das Muster findet somit jede Ereignissequenz die einer der beiden Alternativen entspricht.

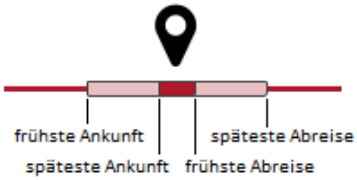
Wiederholungschleifen *(loop)*
Mit Wiederholungsschleifen kann ein Muster definiert werden in dem eine Reihe von Ereignissen auch mehrfach direkt nacheinander durchlaufen werden kann.



Attribute
Durch das Hinzufügen von Attributen können die Ereignisse oder Personen weiter spezifiziert werden. Welche Attribute verfügbar sind hängt von den Daten ab.



Vergleiche *(comparison)*
Für Knoten kann bestimmt werden ob sie sich bestimmte Eigenschaften teilen oder sich in einem oder mehreren Punkten unterscheiden sollen.
Alle Knoten auf die der Vergleich angewandt werden soll müssen mit einer Kante zum Vergleichsknoten verbunden sein.



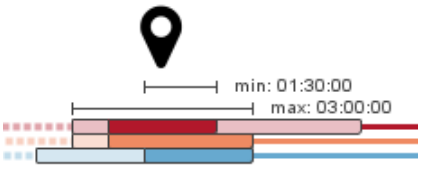
Timeline
Jedes Ereignis bekommt für jede Person die dort einen Aufenthalt hat eine Timeline zugewiesen, die den Aufenthalt zeitlich darstellt.
Die Timeline ist unterteilt in 3 Bereiche:

Ankunftsbereich: Hier trifft die Person ein.
sicherer Bereich: Hier ist die Person sicher anwesend.
Abreisebereich: Hier geht die Person.

Personen die innerhalb des ersten Bereichs ankommen, im 2. Bereich da sind und erst im 3. Bereich gehen passen zum Muster und werden gefunden.

Relative Vorher-Nachher Beziehungen
Bei Ereignissen mit mehreren Personen kann mit der Start und Endposition der Bereiche eine relative Reihenfolge von Ankunft und Abreise definiert werden.
Die Länge der Timelines hat keine Bedeutung.

Zeit- und Datumsangaben
Der Start- und Endzeitpunkt sowie beliebige Zeitpunkte die innerhalb eines Ereignisses liegen sollen können explizit angegeben werden. Entscheidend für die Bedeutung ist die Position der Markierung in den einzelnen Timelines der Personen.



Maximale und minimale Dauer der Überschneidung
Die maximale und minimale Dauer der Überschneidungen der Aufenthalte aller Personen.
Die Dauer der Überschneidung ist die Zeitspanne in der alle Personen anwesend sind.

Einwilligungserklärung für die Studie

„Weiterentwicklung einer Anfragevisualisierung für Ereignissequenzen“

Name des Studienteilnehmers:

Vorname des Studienteilnehmers:

Datum:

Hiermit willigen Sie ein, dass Sie an dieser Studie teilnehmen möchten, diese freiwillig ist und all Ihre Angaben korrekt sind.

Außerdem werden alle personenbezogenen Daten von uns lediglich zu statistischen Zwecken ausgewertet.

Alle Tonaufnahmen die während der Studie gemacht werden, werden ausschließlich zur Auswertung der Antworten verwendet und nach beenden der Studie wieder gelöscht.

Ihre datenschutzrechtlichen Belange werden ohne Einschränkung gewährleistet und es wird keine Übermittlung Ihrer Daten an Dritte erfolgen.

Unterschrift des Studienteilnehmers:

Unterschrift des Studienleiters:

Teilnehmer Fragebogen

Teilnehmer-Nr: _____

Informationen zum Studien-Teilnehmer

Alter: _____

Muttersprache: _____

höchste bisherige Ausbildung: _____

momentaner Studiengang: _____

1. Geschlecht?
   ◯ männlich ◯ weiblich
2. Ishihara Test bestanden?
   ◯ Ja ◯ Nein

Vorwissen

Schätzen Sie Ihr Vorwissen in den folgenden Bereichen ein.

1. Datenanalyse und Datenbanken allgemein:
   unbekannt ◯ ◯ ◯ ◯ ◯ vertraut
2. Graphen allgemein:
   (Was sind Knoten, Kanten,...)
   unbekannt ◯ ◯ ◯ ◯ ◯ vertraut
3. Suchen und Filtern:
   (Logische und reguläre Ausdrücke, Boolesche Suche, Abfragesprachen wie SQL, etc.)
   unbekannt ◯ ◯ ◯ ◯ ◯ vertraut
4. Visuelle Notation:
   (Präskriptive und deskriptive Modelle, Struktur- und Verhaltensdiagramme, UML, etc.)
   unbekannt ◯ ◯ ◯ ◯ ◯ vertraut
5. Informationsvisualisierung allgemein:
   unbekannt ◯ ◯ ◯ ◯ ◯ vertraut
6. Geografische Analyse und GIS:
   (raum-zeitliche Datenanalyse, Bewegungsdatenanalyse)
   unbekannt ◯ ◯ ◯ ◯ ◯ vertraut
7. Visuelle Such- und Filteranfragen:
   (Fokus und Kontext Techniken wie interaktive Linsen, Filter-Flow)
   unbekannt ◯ ◯ ◯ ◯ ◯ vertraut

1 Beschreibungen zuordnen

Im folgenden sind mehrere Ereignismuster abgebildet. Ordnen Sie dem Ereignismuster eine der beiden Beschreibungen zu.

1. Bild 1
   Beschreibungen:
   - ◯ 2 Personen (rot und orange) haben einen gemeinsamen Aufenthalt im Hof. Anschließend geht rot direkt in die Garderobe und orange geht irgendwo anders hin.
   - ◯ 2 Personen (rot und orange) haben einen gemeinsamen Aufenthalt im Hof. Anschließend geht rot direkt in die Garderobe und orange geht wieder in den Hof.

1. Bild 2
   Beschreibungen:
   - ◯ 3 Personen (rot, orange und hellblau) haben einen gemeinsamen Aufenthalt. Anschließend gehen rot und orange zusammen in die Stage7 und dann geht rot alleine in den Workshop1 und orange in den Workshop2. Hellblau geht währenddessen in die Stage6, kommt dort früher an und hat einen gemeinsamen Aufenthalt mit dunkelblau. Hellblau und Dunkelblau gehen zusammen in den Workshop2 und haben dort einen gemeinsamen Aufenthalt mit orange.
   - ◯ 3 Personen (rot, orange und hellblau) haben einen gemeinsamen Aufenthalt. Anschließend haben rot und orange einen gemeinsamen Aufenthalt in Stage7 und dann geht rot in den Workshop1 und orange in den Workshop2. Hellblau geht in die Stage6 und hat dort einen gemeinsamen Aufenthalt mit dunkelblau. Hellbau und dunkelblau haben anschließend einen gemeinsamen Aufenthalt mit orange im Workshop2.

2 Comprehension

Im folgenden sind mehrere Ereignismuster abgebildet. Versuchen Sie die Bedeutung der Muster kurz in eigene Worte zu fassen.

Im zweiten Teil sollen Sie alle Aussagen ankreuzen, die Ihrer Meinung nach für das jeweilige Muster zutreffen. Es können auch alle oder keine der Aussagen zutreffen. Manchmal schließen sich Aussage gegenseitig aus.

1. Bild 1
   (a) Allgemeine Beschreibung
1. Bild 2
   (a) Allgemeine Beschreibung
1. Bild 3
   (a) Allgemeine Beschreibung
1. Bild 4
   (a) Allgemeine Beschreibung
1. Bild 5
   (a) Beschreibung von relativer Ankunft, Abreise und Dauer
   z.B. Rot kommt gleichzeitig mit/vor/nach orange.
   z.B. Alle bleiben maximal 2 Stunden.
1. Bild 6
   (a) Beschreibung von Zeitpunkten
   z.B. Rot kommt um/vor/nach 18 Uhr.

1. Bild 7
   (a) Aussagen:
      ◯ Rot geht direkt von Stage1 zu Stage2.
      ◯ Orange kann bis zu 2 Ereignisse zwischen dem Ereignis in Stage1 und dem in Stage2 besuchen.
      ◯ Orange kann bis zu 3 Ereignisse zwischen dem Ereignis in Stage1 und dem in Stage2 besuchen.
      ◯ Orange kann direkt von dem Ereignis in Stage1 in das Ereignis in Stage2 gehen.

1. Bild 8
   (a) Aussagen:
      ◯ Rot und orange treffen sich.
      ◯ Rot und orange können das gleiche Ereignis besuchen, dürfen aber nicht gleichzeitig anwesend sein.

1. Bild 9
   (a) Aussagen:
      ◯ Die Person verlässt die Location jedes Mal um später wieder zurückzukommen. Dazwischen darf die Person keine anderen Ereignisse besuchen.
      ◯ Es ist nicht sicher ob die Person die Location verlässt, sie kann auch die ganze Zeit dort bleiben. Dazwischen darf die Person keine anderen Ereignisse besuchen.

1. Bild 10
   (a) Nennen Sie alle möglichen Alternativ-Pfade die aus diesem Ereignismuster hervorgehen. Es sind also alle möglichen Ereignisabfolgen der beteiligten Personen gesucht, die in diesem Muster vorkommen und damit gefunden werden könnten.

3 Zeichnen

Versuchen Sie eigenständig das Ereignismuster aus Bild 11 mit Hilfe der Anwendung nachzubauen. Versuchen Sie diese Aufgabe soweit wie möglich eigenständig zu lösen. Stellen sie möglichst wenig Fragen zur Bedingung der Anwendung.

4 Composition

Versuchen Sie eigenständig die folgenden textuellen Beschreibungen von Ereignismustern so genau wie möglich nachzubauen. Sie können Fragen zur Bedingung der Anwendung stellen.

1. 2 Personen verabreden sich in der Cafeteria und treffen sich direkt anschließend auch in der Lounge. Danach geht die eine Person in den Hof und die andere irgendwo anders hin.

2. 2 Personen verabreden sich in der Cafeteria um um 12 Uhr Mittag zu essen. Einer der beiden ist zu spät und kommt erst nach 12 Uhr, der andere ist auf die Minute genau pünktlich da. Die Person die pünktlich war geht bereits um 14 Uhr wieder alleine zu eine Veranstalung in Stage1, die andere Person bleibt noch etwas und geht später zu einer Veranstaltung in Stage2.

3. 3 Personen möchten gemeinsam einen Vortrag zum Thema Kultur in Stage3 hören. Alle Personen gehen anschließend auch in einen anderen Vortrag in Stage3. Anschließend bleiben 2 Personen in Stage3 und hören sich bis zu 5 weitere Vorträge an und gehen dann in die Refillbar, während die dritte Person Hunger bekommt, in die Cafeteria geht und sich dort mit einer anderen Person trifft. Die Personen in der Cafeteria bleiben mindestens 2 Stunden gemeinsam dort, gehen aber bereits vor 18 Uhr.

5 Bewertung

Bewerten Sie diese Studie und die visuelle Anfragesprache.

1. Hat die Einführung in die Anfragesprache ausgereicht um Ereignismuster zu verstehen?
   trifft zu ◯ ◯ ◯ ◯ ◯ trifft nicht zu
2. Hat die Einführung in die Anfragesprache ausgereicht um Ereignismuster selber zu modellieren?
   trifft zu ◯ ◯ ◯ ◯ ◯ trifft nicht zu
3. Wie viel Vorwissen war nötig um das Konzept zu verstehen?
   viel ◯ ◯ ◯ ◯ ◯ wenig
4. Hatten Sie das Gefühl die Aufgaben schnell bearbeiten zu können oder war das Erstellen der Muster langwierig?
   schnell ◯ ◯ ◯ ◯ ◯ langwierig
5. Denken Sie die Anfragesprache ist leicht zu verstehen?
   eher leicht ◯ ◯ ◯ ◯ ◯ eher schwer
6. Denken Sie die Anfragesprache ist eher umfangreich/komplex oder überschaubar/einfach gehalten?
   eher komplex ◯ ◯ ◯ ◯ ◯ eher einfach gehalten
7. Finden Sie die Anfragesprache intuitiv?
   mehr intuitiv ◯ ◯ ◯ ◯ ◯ weniger intuitiv
8. Wie gut haben Sie die einzelnen Konzepte verstanden.
   Ereignisse (event)
   verstanden ◯ ◯ ◯ ◯ ◯ nicht ganz verstanden

   Schleifen (loop)
   verstanden ◯ ◯ ◯ ◯ ◯ nicht ganz verstanden

   Verzweigungen (branch)
   verstanden ◯ ◯ ◯ ◯ ◯ nicht ganz verstanden

   Pfade/Übergänge
   verstanden ◯ ◯ ◯ ◯ ◯ nicht ganz verstanden

   Timeline
   verstanden ◯ ◯ ◯ ◯ ◯ nicht ganz verstanden

   Vorher-Nachher Beziehungen bei Ankunft und Abreise
   verstanden ◯ ◯ ◯ ◯ ◯ nicht ganz verstanden

   Uhrzeit und Datumsangaben
   verstanden ◯ ◯ ◯ ◯ ◯ nicht ganz verstanden

   minimale und maximale Ereignisdauer
   verstanden ◯ ◯ ◯ ◯ ◯ nicht ganz verstanden

   Attribute
   verstanden ◯ ◯ ◯ ◯ ◯ nicht ganz verstanden

   Vergleiche (comparison)
   verstanden ◯ ◯ ◯ ◯ ◯ nicht ganz verstanden

9. Wie schwer fanden Sie das Erstellen der Komponenten in der Anwendung.
   Ereignisse (event)
   schwer ◯ ◯ ◯ ◯ ◯ leicht

   Schleifen (loop)
   schwer ◯ ◯ ◯ ◯ ◯ leicht

   Verzweigungen (branch)
   schwer ◯ ◯ ◯ ◯ ◯ leicht

   Pfade/Übergänge
   schwer ◯ ◯ ◯ ◯ ◯ leicht

   Timeline
   verstanden ◯ ◯ ◯ ◯ ◯ nicht ganz verstanden

   Vorher-Nachher Beziehungen bei Ankunft und Abreise
   schwer ◯ ◯ ◯ ◯ ◯ leicht

   Uhrzeit und Datumsangaben
   schwer ◯ ◯ ◯ ◯ ◯ leicht

   minimale und maximale Ereignisdauer
   schwer ◯ ◯ ◯ ◯ ◯ leicht

   Attribute
   schwer ◯ ◯ ◯ ◯ ◯ leicht

   Verleiche (comparison)
   schwer ◯ ◯ ◯ ◯ ◯ leicht

10. Hatten sie konkrete Probleme beim Verständnis der Anfragesprache? Wenn ja, welche?

11. Hatten sie konkrete Probleme beim der Bedienung der Anwendung? Wenn ja, welche?

12. Wenn Sie Anfragen an eine Datensammlung von Ereignissequenzen stellen müssten, würden Sie diese Anfragesprache gegenüber herkömmlichen Methoden vorziehen? Begründen Sie ihre Antwort kurz.

Auswertungsbogen

Teilnehmer-Nr: _____

1. Bild 1
   (a) Allgemeine Beschreibung
   - ◯ Rot und orange haben einen gemeinsamen Aufenthalt im Hof.
   - ◯ Rot und orange haben einen gemeinsamen Aufenthalt in der Cafeteria.
   - ◯ Rot und orange haben davor einen Aufenthalt alleine.
   - ◯ Wo ist nicht klar.
   - ◯ Können sich dort nicht treffen.

2. Bild 2
   (a) Allgemeine Beschreibung
   - ◯ Rot, orange und hellblau haben zuerst ein gemeinsames Ereignis.nd in Stage7
   - ◯ Abreise ist nicht eingeschränkt.
   - ◯ Ankunft ist eingeschränkt.
   - ◯ Rot kommt zuerst.
   - ◯ Orange und blau kommen vielleicht später.
   - ◯ Nach dem ersten Ereignis gehen entweder rot und hellblau in die Cafeteria oder rot und orange gehen in die Refillbar.
   - ◯ orange oder hellblau sind nicht definiert.

3. Bild 3
   (a) Allgemeine Beschreibung
   - ◯ Das erste Ereignis von rot und orange findet an verschiedenen Orten statt.
   - ◯ Rot und orange besuchen mehrfach nacheinander abwechselnd ein Ereignis im Workshop1 und ein Ereignis in Stage6.
   - ◯ Rot und orange können in diesem Muster maximal 4 Ereignisse im Workshop1 besuchen.
   - ◯ Rot und orange können in diesem Muster minimal ein Ereignisse im Workshop1 besuchen.
   - ◯ Rot und Orange können nicht direkt vom ersten Ereignis in die Cafeteria gehen.
   - ◯ Die Ereignisse im Workshop1 uns Stage6 können unterschiedlich sein.

4. Bild 4
   (a) Allgemeine Beschreibung
   - ◯ gemeinsamen Aufenthalt rot, orange und hellblau.
   - ◯ rot hat maximal 3 Aufenthalte vor dem 3. Ereignis.
   - ◯ rot hat minimal 2 Aufenthalte vor dem 3. Ereignis.
   - ◯ orange hat genau einen Aufenthalt dazwischen.
   - ◯ orange hat maximal 6 Aufenthalt vor der Verzweigung.
   - ◯ orange hat minimal einen Aufenthalt vor der Verzweigung.
   - ◯ rot kommt vor orange im 3. Ereignis an.
   - ◯ entweder 3 gehen in die Refillbar.
   - ◯ oder 4 gehen in die stages
   - ◯ fängt um 22 und 23 Uhr an

5. Bild 5
   (a) Beschreibung von relativer Ankunft, Abreise und Dauer

   (b) Aussagen zum Ereignis in der Cafeteria:
      ◯ Rot.
      ◯ Orange.
      ◯ Hellblau.
      ◯ Dunkelblau.
      ◯ Orange ist maximal 4 Stunden in der Cafeteria.
   (c) Aussagen zum 2. Ereignis:
      ◯ minimal und maximal
      ◯ Rot ist minimal 3 Stunden am Ort des 2. Ereignisses.
      ◯ Orange ist maximal 3 Stunden am Ort des 2. Ereignisses.

6. Bild 6
   (a) Beschreibung von Zeitpunkten
   (b) Aussagen zum Ereignis in der Cafeteria:
      ◯ Rot.
      ◯ Orange.
      ◯ Hellblau.
      ◯ Dunkelblau.
      ◯ Alle kommen zwischen 10 und 13 Uhr zur Cafeteria.
      ◯ Orange geht um 15 Uhr.
      ◯ Hellblau und Dunkelblau und rot gehen nach 15 Uhr.
      ◯ Rot ist um 10 Uhr noch nicht da.
      ◯ Um 13 Uhr sind spätestens alle da.
   (c) Aussagen zum Treffen zwischen Rot und Orange:
      ◯ Rot und Orange kommen und gehen nicht gleichzeitig.
      ◯ Rot kommt nach 18 Uhr.
   (d) Aussagen Allgemein:
      ◯ Rot, Orange und Hellblau kommen genau um 23 Uhr in der Refillbar an.
      ◯ Die beiden Ereignisse von hellblau/dunkelblau findet später statt.

7. Zeichnung von Bild 11
   (a) Zeichnung richtig?
      ◯ Ja ◯ Nein
   (b) Anzahl der Fragen?
   (c) Anmerkungen:

Bilder

1 Beschreibungen zuordnen

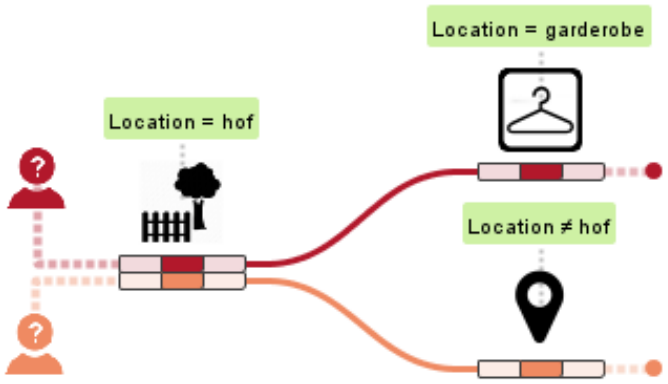


Figure 1.1:

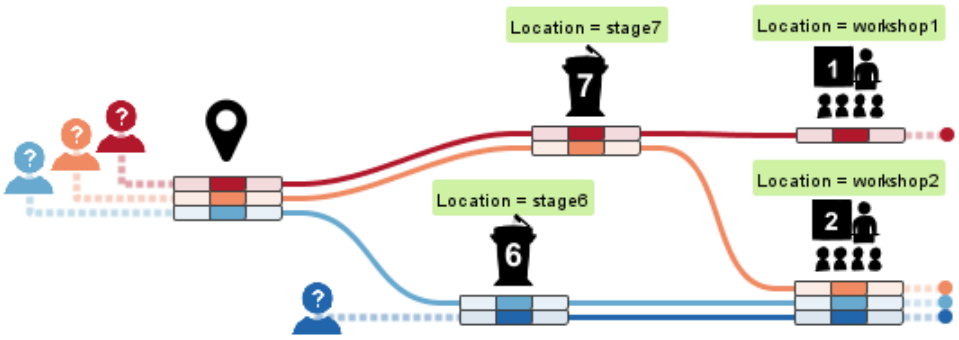


Figure 1.2:

2 Comprehension

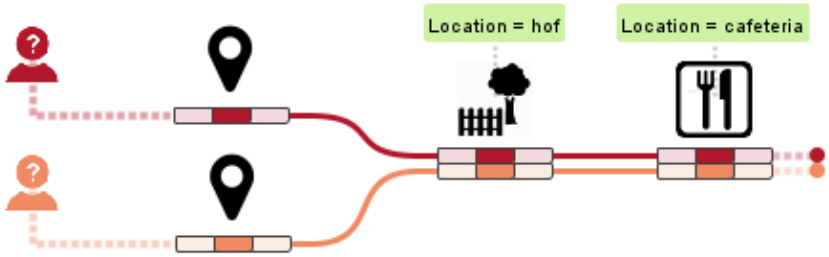


Figure 2.1:

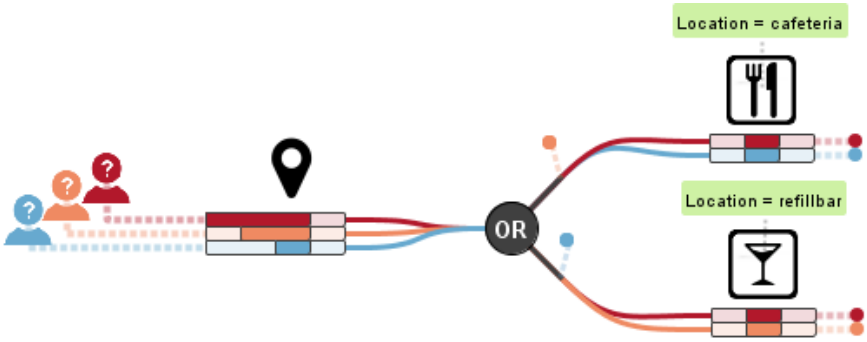


Figure 2.2:

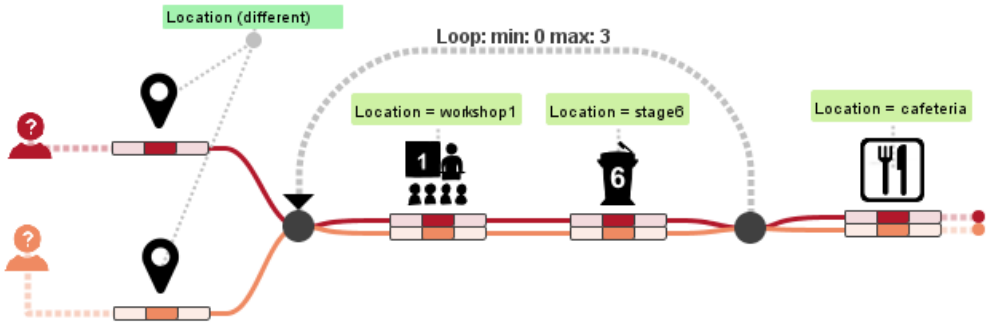


Figure 2.3:

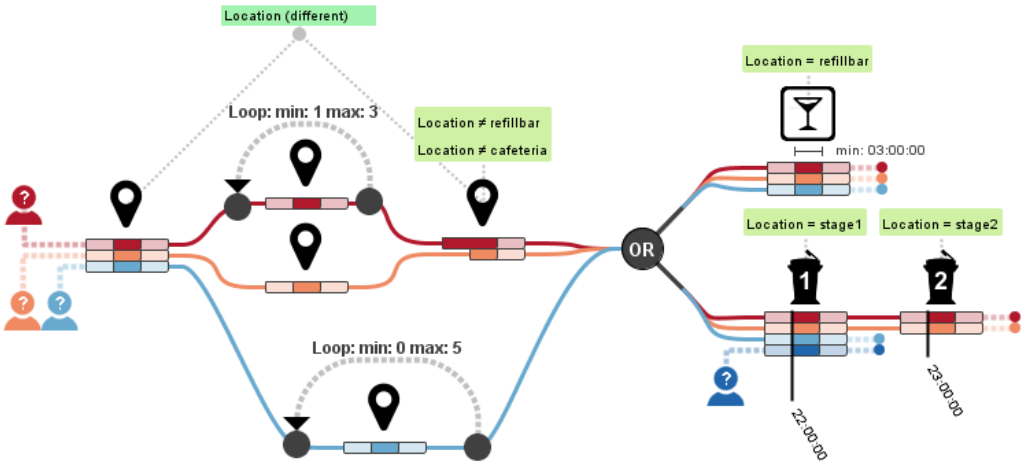


Figure 2.4:

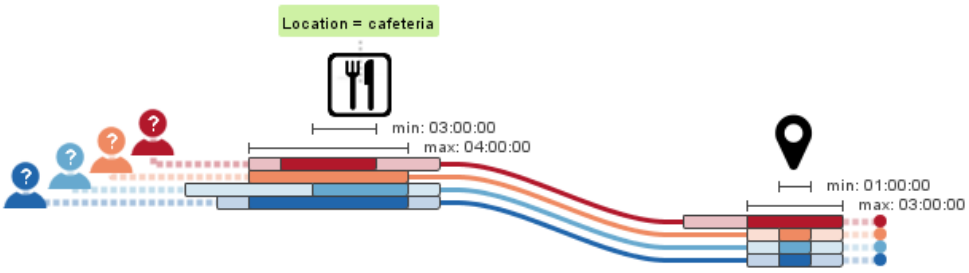


Figure 2.5:

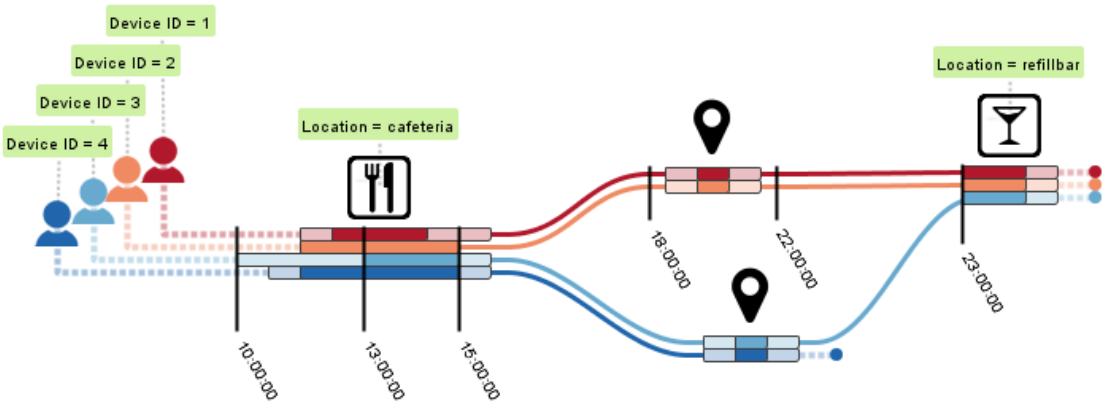


Figure 2.6:

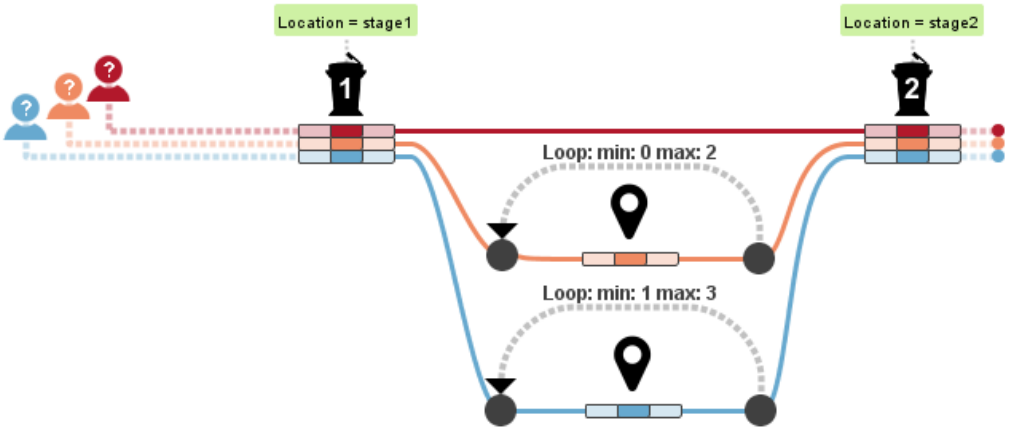


Figure 2.7:

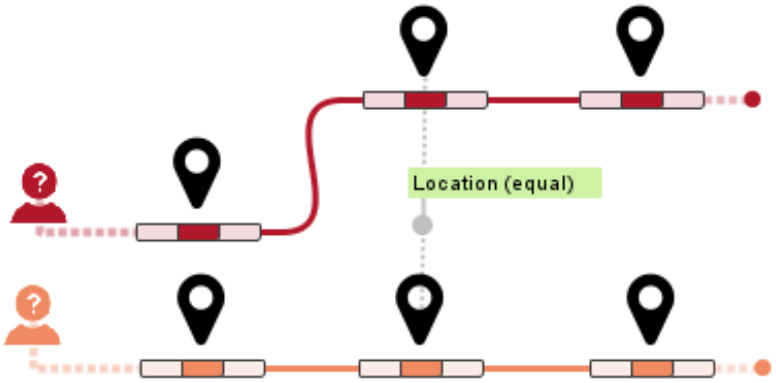


Figure 2.8:

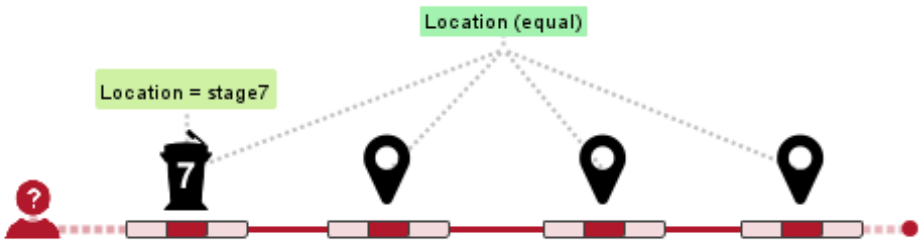


Figure 2.9:

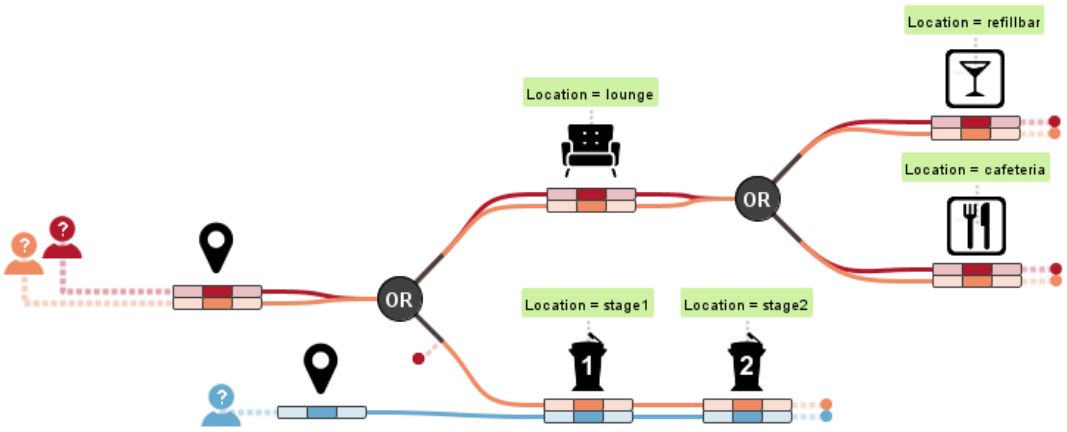


Figure 2.10:

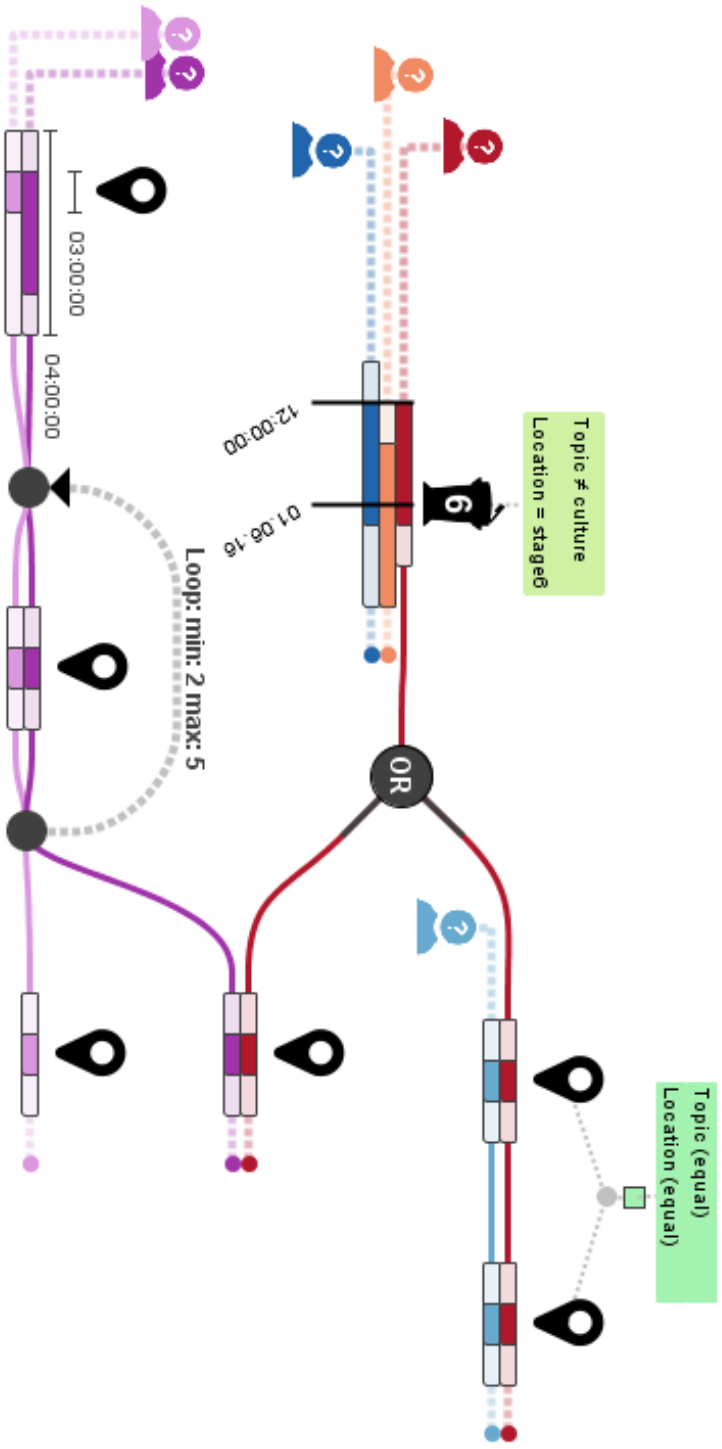
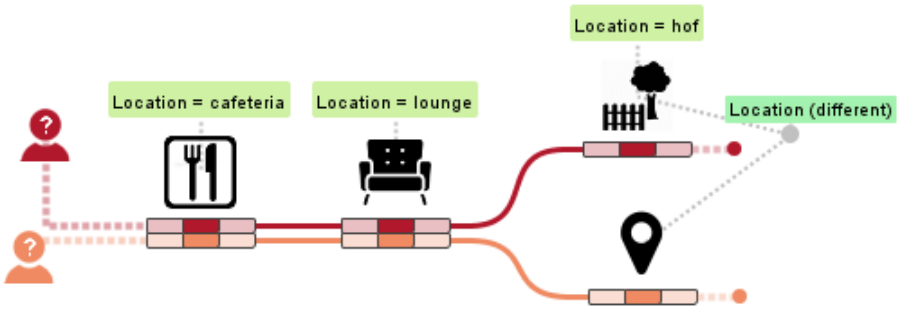
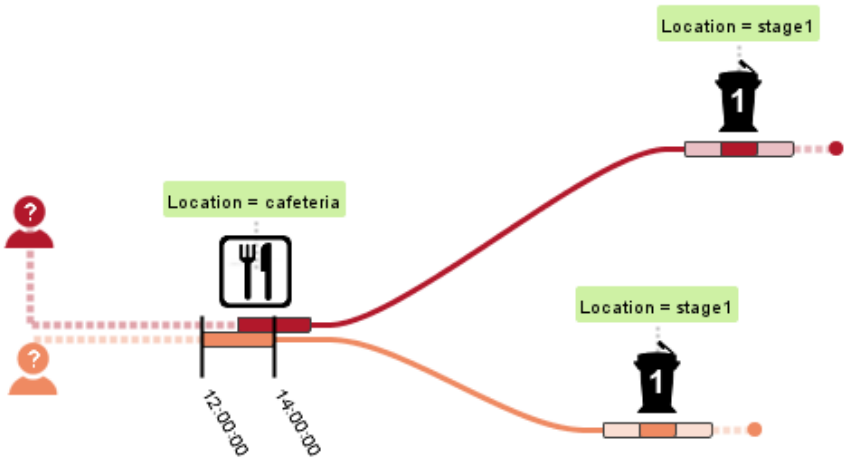
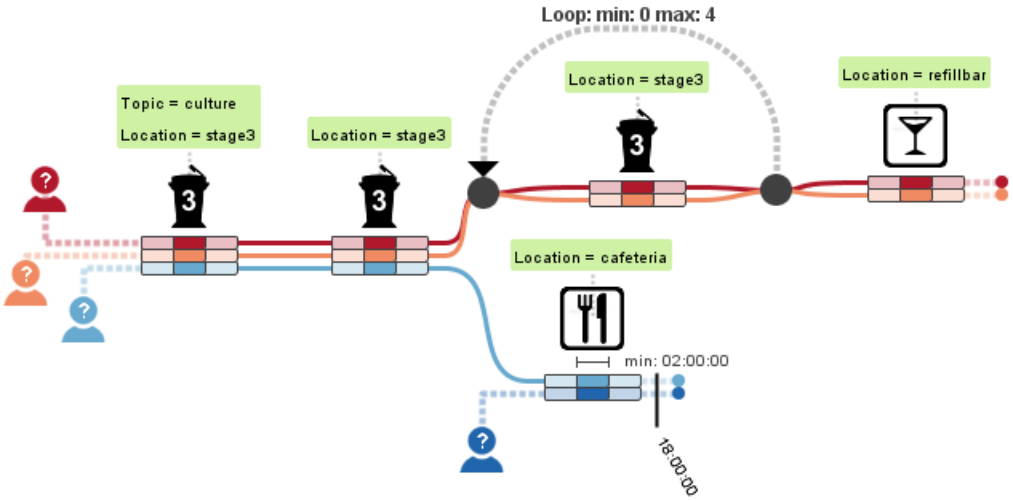


Figure 2.11: Zeichnen


Location = hof
Location = cafeteria
Location = lounge
Location (different)


Location = stage1
Location = cafeteria
Location = stage1
12:00:00
14:00:00


Loop: min: 0 max: 4
Location = stage3
Location = refillbar
Topic = culture
Location = stage3
Location = stage3
Location = cafeteria
min: 02:00:00
18:00:00

Literaturverzeichnis

[All83] J. F. Allen. „Maintaining Knowledge About Temporal Intervals“. In: *Commun. ACM* 26.11 (Nov. 1983), S. 832–843. URL: http://doi.acm.org/10.1145/182.358434 (zitiert auf S. 23, 37, 52).

[AMTB05] W. Aigner, S. Miksch, B. Thurnher, S. Biffl. „PlanningLines: novel glyphs for representing temporal uncertainties and their evaluation“. In: *Ninth International Conference on Information Visualisation (IV'05)*. Juli 2005, S. 457–463 (zitiert auf S. 19, 31).

[CC03] L. Chittaro, C. Combi. „Visualizing queries on databases of temporal histories: new metaphors and their evaluation“. In: *Data & Knowledge Engineering* 44.2 (2003), S. 239–264 (zitiert auf S. 19).

[CK91] S. B. Cousins, M. G. Kahn. „The Visual Display of Temporal Information“. In: *Artificial Intelligence in Medicine* 3.3 (1991), S. 341–357 (zitiert auf S. 19).

[CMS99] S. K. Card, J. D. Mackinlay, B. Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999 (zitiert auf S. 21).

[HHN00] S. Havre, B. Hetzler, L. Nowell. „ThemeRiver: visualizing theme changes over time“. In: *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on*. 2000, S. 115–123 (zitiert auf S. 19, 42, 57).

[HKE16] F. Haag, R. Krüger, T. Ertl. „VESPa: A Pattern-Based Visual Query Language for Event Sequences“. In: *Proceedings of the 7th International Conference on Information Visualization Theory and Applications (IVAPP 2016)*. Bd. 7. 2016 (zitiert auf S. 16, 20, 23–25, 74, 79).

[KHH+15] R. Krueger, F. Heimerl, Q. Han, K. Kurzhals, S. Koch, T. Ertl. „Visual Analysis of Visitor Behavior for Indoor Event Management“. In: *System Sciences (HICSS), 2015 48th Hawaii International Conference on*. Jan. 2015, S. 1148–1157 (zitiert auf S. 24).

[KHHE15] R. Krüger, D. Herr, F. Haag, T. Ertl. „Inspector-Gadget: Integrating Data Preprocessing and Orchestration in the Visual Analysis Loop“. In: *International Workshop on Visual Analytics EuroVA*. 2015 (zitiert auf S. 24).

[KKEM10] D. A. Keim, J. Kohlhammer, G. Ellis, F. Mansmann. *Mastering the information age-solving problems with visual analytics*. Florian Mansmann, 2010 (zitiert auf S. 21).

[Kra03] M.-J. Kraak. „The space-time cube revisited from a geovisualization perspective“. In: *Proc. 21st International Cartographic Conference*. 2003, S. 1988–1996 (zitiert auf S. 20).

[Ope13] OpenDataCity. *re:log - Besucherstromanalyse per re:publica W-LAN*. 2013. URL: http://apps.opendatacity.de/relog/ (zitiert auf S. 24).

[rep13] re:publica 13. *re:publica 13 - Die Konferenz. Das Ereignis.* 2013. URL: https://13.re-publica.de/ (zitiert auf S. 23).

[Shn96] B. Shneiderman. „The eyes have it: a task by data type taxonomy for information visualizations“. In: *Visual Languages, 1996. Proceedings., IEEE Symposium on*. Sep. 1996, S. 336–343 (zitiert auf S. 21).

[Shn98] B. Shneiderman. „Visual user interfaces for information exploration“. In: (1998) (zitiert auf S. 20).

[Vis14] Visual Analytics Community. *VAST Challenge 2014: Mini-Challenge 2*. 2014. URL: http://www.vacommunity.org/VAST+Challenge+2014:+Mini-Challenge+2 (zitiert auf S. 24).

[ZDFD15] *(s|qu)eries: Visual Regular Expressions for Querying and Exploring Event Sequences*. ACM – Association for Computing Machinery, Apr. 2015. URL: https://www.microsoft.com/en-us/research/publication/squeries-visual-regular-expressions-for-querying-and-exploring-event-sequences/ (zitiert auf S. 20, 41).

Alle URLs wurden zuletzt am 27. 07. 2016 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift