

Institut für Visualisierung und Interaktive Systeme

Universität Stuttgart
Universitätsstraße 38
D - 70569 Stuttgart

Bachelorarbeit Nr. 188

Query Visualization for Time-Based Graph Data

Dirk Väth

Studiengang:	Informatik
Prüfer:	Prof. Dr. Thomas Ertl
Betreuer:	Dipl.-Inf. Florian Haag
begonnen am:	24. November 2014
beendet am:	26. Mai 2015
CR-Klassifikation:	H.1.2, H.2.3, H.3.3, H.5.2, I.2.4

Kurzfassung

Für die visuelle Filterung graphbasierter Daten existieren schon verschiedene Konzepte, die aber meist die Filterung nach temporalen Daten vernachlässigen und diese wie andere Datentypen behandeln. In dieser Arbeit soll ein Konzept vorgestellt werden, das sich darauf konzentriert, graphbasierte Daten nicht nur nach absoluten Zeitpunkten, sondern auch nach Anordnung der temporalen Daten zu ermöglichen, mit dem Ziel einer guten Lesbarkeit und Verständlichkeit der temporalen Filter. Um dies feststellen zu können, wurde ein Prototyp implementiert und in einer Nutzerstudie getestet. Die Ergebnisse dieser Studie und deren Bedeutung für die Ziele des Konzepts sollen ebenfalls in dieser Arbeit vorgestellt werden.

Abstract

Today, many different concepts for the visual filtering of graph-based data exist. Most of the time however they neglect temporal data by handling it like any other data type. This paper presents a new visual concept with focus on filtering graph-based data not only by absolute points in time but also by the order of temporal data with the goal of easy readability as well as comprehensibility. To be able to judge the concept by this goal, a prototype was implemented and a user study was conducted. The results of this study and their meaning concerning the goals of this concept are also presented in this paper.

Inhaltsverzeichnis

1. Einleitung	9
1.1. Motivation	9
1.2. Zielsetzung	9
1.3. Kapitelübersicht	9
2. Hintergrund	11
2.1. Zeit	11
2.2. Anfragen	14
3. Konzept	21
3.1. Zeit	21
3.2. Anfragenstruktur	22
3.3. Filter	23
3.4. Anfragen	28
3.5. Kombination mit existierenden Konzepten	33
4. Implementierung	35
4.1. Eingesetzte Technologien	35
4.2. Klassenhierarchie	36
4.3. Verfügbare Filter	38
4.4. Graphzeichnung	39
4.5. Anfragegenerierung	41
5. Nutzerstudie	45
5.1. Vorbereitung und Durchführung	45
5.2. Ergebnisse	47
5.3. Auswertung	53
6. Zusammenfassung und Ausblick	57
A. Materialien der Nutzerstudie	59
A.1. Interview mit Probanden	60
A.2. Aufgaben für Probanden	61
A.3. Anleitung für Probanden	63
A.4. Fragebogen für Probanden	71
Literaturverzeichnis	73

Abbildungsverzeichnis

2.1. Intervalldarstellung von Zeitrelationen	12
2.2. Physikalische Metaphern für Intervalle und deren Relationen	13
2.3. Spuren-Metapher für die Anordnung von Aktivitäten in <i>AsbruView</i>	14
2.4. Visualisierung Bool'scher Relationen mit dem Filter/Flow-Konzept	15
2.5. Aussagenlogische Filterverknüpfung durch Positionierung	15
2.6. Visuelle Anfrage für verknüpfte Daten mittels ER-Diagramm	16
2.7. Anfrage in GQL	16
2.8. Temporale Anfragenformulierung in <i>TempViz</i>	17
2.9. Aussagenlogische Verknüpfung von Intervallen und temporalen Abstraktionen	18
2.10. Anfrageeditor und ein Anfragebeispiel in <i>Time Automaton</i>	19
3.1. Visuelle Repräsentation eines Faktenknotens und seiner Verknüpfung mit aussagenlogischen Ausdrücken	25
3.2. Popover zur Literaldefinition.	25
3.3. Visuelle Repräsentation eines Ereignisknotens und seiner temporalen Verknüpfungen	27
3.4. Subjektknoten für das Subjekt Person	29
3.5. Unverbundene und verbundene Subjektgraphen	29
3.6. Logische Verknüpfung von Filtern	31
3.7. Visualisierung möglicher relativer Ereignisverknüpfungen	32
4.1. Klassendiagramm des Prototyps	37
4.2. DBPedia-Auszug	38
4.3. XSD-Schema der Anfragesubjekt-Definitionsdatei.	40
4.4. Anfragebeispiel im Prototyp	42
4.5. Ergebnisse der Sparql-Anfrage im Prototyp	43
5.1. Subjektive Schwierigkeit aller Probanden je Prototypfeature.	48
5.2. Gefühlt benötigte Zeit für das Erstellen einer Anfrage.	50
5.3. Benötigte Zeit je Verständnisaufgabe.	51
5.4. Mittlere subjektive Schwierigkeit aller Probanden je Verständnisaufgabe.	52
5.5. Benötigte Zeit je Kompositionsaufgabe.	54
5.6. Subjektive Schwierigkeit aller Probanden je Kompositionsaufgabe.	54

Tabellenverzeichnis

2.1. Relationen temporaler Intervalle nach Allen und deren Punktrepräsentation. . .	12
---	----

Verzeichnis der Listings

4.1. Erste generierte SPARQL-Anfrage	42
4.2. Zweite generierte SPARQL-Anfrage	43

1. Einleitung

1.1. Motivation

Visuelle Anfragesprachen zur Filterung graphbasierter Daten existieren zwar bereits, doch während sich die meisten dieser Konzepte zum Beispiel auf die Darstellung von Verbindungen zwischen Ressourcen oder der aussagenlogischen Kombination der Filterkriterien fokussieren, wird meist die Visualisierung des temporalen Aspekts vernachlässigt. Dabei sind für viele Szenarien temporale Beziehungen wichtig, zum Beispiel könnte ein Knowledge Graph, der Daten zu Politikern enthält, nach Länge der Amtsperioden, Reihenfolge der verschiedenen Ämter eines Politikers oder nach Reihenfolge der Amtsinhaber gefiltert werden.

1.2. Zielsetzung

Als Ziel dieser Arbeit soll ein neues Konzept einer visuellen Anfragesprache für graphbasierte Daten entwickelt werden, das sowohl deren absolute als auch relative temporale Filterung ermöglicht. Mittels einer prototypischen Implementierung soll das Konzept in einer Nutzerstudie evaluiert werden.

1.3. Kapitelübersicht

Die Arbeit ist in folgende Kapitel unterteilt:

Kapitel 1, Einleitung Dieses Kapitel erläutert Motivation und Zielsetzung der vorliegenden Arbeit sowie die Kapiteleinteilung.

Kapitel 2, Hintergrund In diesem Kapitel wird eine Übersicht über verwandte Forschungsarbeiten gegeben.

Kapitel 3, Konzept In diesem Kapitel wird das für diese Arbeit entwickelte, neue Konzept einer visuellen Anfragesprache zur temporalen Filterung graphbasierter Daten vorgestellt. Weiterhin werden mögliche Verknüpfungen mit vorhandenen Konzepten diskutiert.

Kapitel 4, Implementierung Hier werden Erläuterungen zu eingesetzten Technologien, Struktur und Funktionsweise für die Entwicklung eines Prototyps für das in Kapitel 3 vorgestellte Konzept gegeben.

1. Einleitung

Kapitel 5, Nutzerstudie Dieses Kapitel beschreibt Planung, Durchführung und Ergebnisse einer Nutzerstudie zum Test des in Kapitel 4 vorgestellten Prototypen.

Kapitel 6, Zusammenfassung und Ausblick Das abschließende Kapitel fasst die Ergebnisse der Arbeit zusammen und stellt Verbesserungsvorschläge bezüglich des vorgestellten Konzepts vor.

2. Hintergrund

In diesem Kapitel werden existierende Konzepte vorgestellt, die für Aspekte des in dieser Arbeit vorgestellten Konzepts wichtig oder diesen ähnlich sind.

2.1. Zeit

2.1.1. Abstraktionen

Für den Entwurf einer möglichst intuitiven Anfragevisualisierung für temporale Daten muss zunächst das alltägliche Verständnis von Zeit betrachtet werden. Im Duden werden als mögliche Definitionen unter anderem

„(umgangssprachlich) Zeitpunkt; eng begrenzter Zeitraum (in Bezug auf seine Stelle im Zeitablauf), Augenblick“

und

„Zeitraum; Zeitabschnitt, Zeitspanne“

angegeben [Dud14]. Dieses Zeitverständnis unterscheidet zwischen den Abstraktionen Punkt und Intervall. Diese wiederum können eingeteilt werden in absolute oder relative Zeitpunkte bzw. Intervalle. Unter absoluten Zeitpunkten oder Intervallen wird dabei eine Zeitangabe wie etwa ein Datum verstanden. Relative Zeitpunkte oder Intervalle definieren sich nicht über eine Zeitangabe, sondern über ihre Relation zu anderen Zeitpunkten oder Intervallen, zum Beispiel, dass ein Zeitintervall vor einem anderen beginnen soll.

Zwischen Intervallen und Zeitpunkten kann konvertiert werden, indem ein Intervall in Start- und Endpunkt zerlegt bzw. zwei Punkte als Start und Ende eines Intervalls festgelegt werden. Ebenso können auch die möglichen Relationen zwischen Intervallen mit Punkten nachgestellt werden. Eine vollständige Übersicht über die 13 Intervallrelationen nach Allen [AA83] und deren Entsprechung in Punktform gibt Tabelle 2.1.

2. Hintergrund

Intervallrelation	Punktrepräsentation
I_1 vor I_2	$I_1^+ < I_2^-$
I_1 trifft I_2	$I_1^+ = I_2^-$
I_1 überlappt I_2	$(I_1^- < I_2^-) \wedge (I_1^+ > I_2^-) \wedge (I_1^+ < I_2^+)$
I_1 startet I_2	$(I_1^- = I_2^-) \wedge (I_1^+ < I_2^+)$
I_1 beendet I_2	$(I_1^- > I_2^-) \wedge (I_1^+ = I_2^+)$
I_1 während I_2	$(I_1^- > I_2^-) \wedge (I_1^+ < I_2^+)$
I_1 gleich I_2	$(I_1^- = I_2^-) \wedge (I_1^+ = I_2^+)$
I_1 enthält I_2	$(I_1^- < I_2^-) \wedge (I_1^+ > I_2^+)$
I_1 beendet von I_2	$(I_1^- < I_2^-) \wedge (I_1^+ = I_2^+)$
I_1 gestartet von I_2	$(I_1^- = I_2^-) \wedge (I_1^+ > I_2^+)$
I_1 überlappt von I_2	$(I_1^- > I_2^-) \wedge (I_1^- < I_2^+) \wedge (I_1^+ > I_2^+)$
I_1 getroffen von I_2	$(I_1^- = I_2^+)$
I_1 nach I_2	$(I_1^- > I_2^+)$

Tabelle 2.1.: Intervallrelationen nach Allen [AA83] und deren Punktrepräsentation [QVDMW14]. I_1 und I_2 bezeichnen Intervalle, ein $-$ kennzeichnet den Startpunkt eines Intervalls, ein $+$ den Endpunkt.



Abbildung 2.1.: Verschiedene Relationen zwischen zwei Ereignissen A und B in Intervalldarstellung [CC01].

2.1.2. Visualisierung von Zeit und temporalen Relationen

Tufte erwähnt die Möglichkeit, Zeit in Intervallform durch Balken zu repräsentieren. Die Länge eines Balkens gibt die Länge des Intervalls an, während seine Position bezüglich einer Achse im Raum seine Position in der Zeit angibt [TGM83]. Allen verwendet bei seinen beispielhaften Visualisierungen von Relationen der Zeitintervalle ein ähnliches Konzept [AA83]. Ein Beispiel für die Darstellung von Zeitintervallen in verschiedenen Relationen durch Balken ist in Abbildung 2.1 gegeben.

Chittaro und Combi schlagen vor, nicht auf abstrakte Visualisierungen wie Balken zu setzen, sondern Zeitintervalle und deren Relationen durch Abbildungen realer Objekte zu ersetzen, um vorhandenes Wissen der Nutzer einbringen zu können [CC01]. Dabei schlagen sie drei verschiedene Visualisierungsmöglichkeiten vor: Elastische Bänder, Federn und Farbstreifen (siehe Abbildung 2.2). Elastische Bänder und Federn geben ihre Position in der Zeit durch eine Fixierung mit Schrauben an der entsprechenden Stelle bezüglich einer horizontalen Achse an, die aber nicht sichtbar

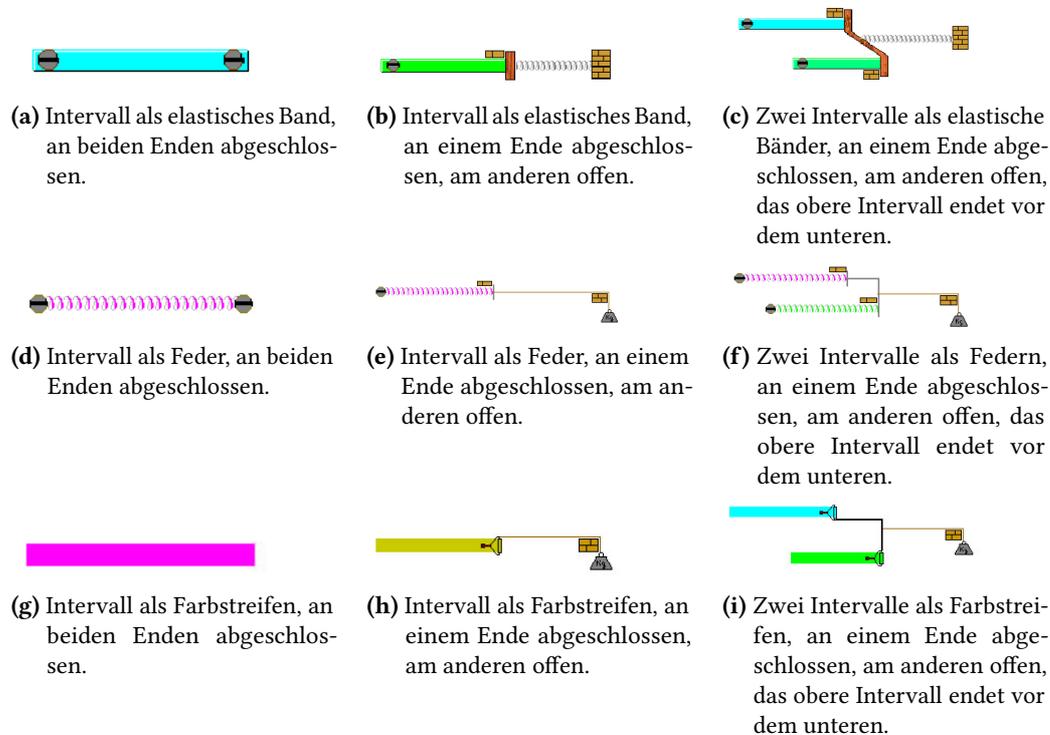


Abbildung 2.2.: Physikalische Metaphern für Intervalle und deren Relationen [CC01].

sein muss, Farbstreifen durch Beginn und Ende ihrer Streifen, ähnlich der Balkenvisualisierung. Relationen zwischen Intervallen werden durch die Position der Schrauben bzw. Beginn und Ende der Farbstreifen und deren Anbindung an bewegliche Massensysteme dargestellt. Mit einem Massensystem können mehrere Intervalle verbunden werden, wobei dann die Position der Anbindung an das Massensystem über die Relation zwischen den beiden Intervallen entscheidet.

Eine visuelle Metapher für die Visualisierung temporaler Abläufe wird im *AsbruView*-Konzept vorgestellt [MKSJ98]. Es visualisiert Ablaufpläne, die von Personen ausgeführt werden sollen, im dreidimensionalen Raum durch Spuren und Verkehrsschilder. Aktivitäten in diesem Plan können nicht nur sequentiell oder parallel, sondern auch zyklisch angeordnet werden. Jede Aktivität entspricht einer beschrifteten Spur. Die x -Achse definiert dabei eine Zeitlinie, in der zwei hintereinander angeordnete Spuren eine sequentielle Anordnung beschreiben. Auf dieser können auch temporale Abstände beziehungsweise die Dauer von Aktivitäten definiert werden. Gleichzeitig auszuführende Aktivitäten werden als in der y -Ebene parallele Spuren dargestellt. Teilpläne können auch zusammengefasst werden und in einer anderen z -Ebene als Abstraktion der Teilpläne dargestellt werden. Verschiedene Verkehrsschilder, die Bedingungen repräsentieren, steuern die möglichen Übergänge von einer Aktivität in eine oder mehrere andere Aktivitäten durch ihre Art und Positionierung.

2. Hintergrund

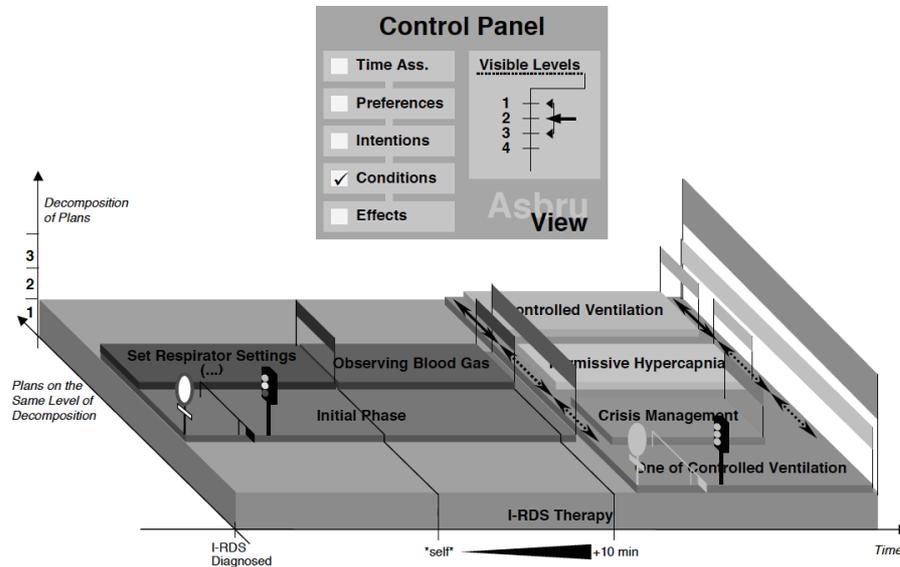


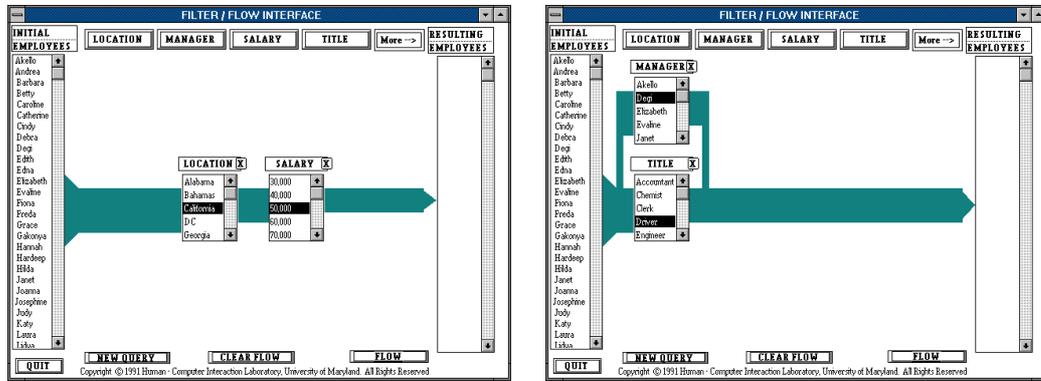
Abbildung 2.3.: Spuren-Metapher für die Anordnung von Aktivitäten in AsbruView [MKSJ98].

2.2. Anfragen

2.2.1. Aussagenlogische Anfragen

Komplexe Suchanfragen erfordern eine Möglichkeit zur Kombination der verschiedenen Suchparameter. Viele Konzepte nutzen dazu aussagenlogische Junktoren wie *und*, *oder* und *nicht*, die auch im natürlichen Sprachgebrauch Verwendung finden. Doch Studien, wie von Hildreth zusammengefasst, stellen erhebliche Probleme bei der Formulierung aussagenlogischer Anfragen ohne Expertenwissen auf diesem Gebiet fest [Hil89].

Shneidermans Filter/Flow-Konzept [Shn98] und dessen Implementierung und Nutzerstudien-durchführung durch Shneiderman und Young [YS98] legen jedoch nahe, dass eine metaphorische visuelle Repräsentation aussagenlogischer Verknüpfungen das Verständnis dieser erleichtert, indem es die Bedeutung der Anfragen intuitiv erkennbar macht, auch ohne Wissen im Bereich der Aussagenlogik. Die verwendete Metapher in diesem Konzept ist ein Fluss, der von links nach rechts die Filter der Anfrage durchfließt. Er beginnt am linken Bildschirmrand an einer Liste mit allen verfügbaren Daten und endet am rechten Bildschirmrand in einer Liste, in der die Ergebnisse der Suche angezeigt werden. Nach jedem durchflossenen Filter wird der Fluss dünner, um anzuzeigen, dass Daten herausgefiltert wurden. Filter auf einem gemeinsamen Flussarm sind dabei konjunktiv verbunden (siehe Abbildung 2.4a). Wird der Fluss aufgespalten in mehrere Flussarme, so repräsentiert jeder Flussarm eine Disjunktion (siehe Abbildung 2.4b). Klammern und Operatoren werden durch dieses Konzept überflüssig. Die Nutzerstudie von Young kommt zum Ergebnis, dass das Filter/Flow-Konzept zum Verständnis der Anfragen beiträgt und dass die



(a) Und-Beziehung.

(b) Oder-Beziehung.

Abbildung 2.4.: Visualisierung Bool'scher Relationen mit Shneiderman's Filter/Flow-Konzept [Shn98].

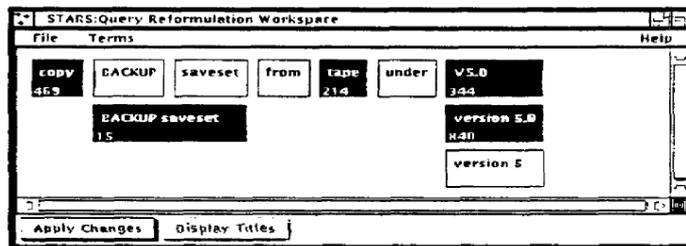


Abbildung 2.5.: Für die Anfrage aktiv sind die Filterwörter *copy*, *BACKUP saveset*, *tape*, *v5.0* und *version 5.0*. Die Anfrage ergibt sich zu $copy \wedge BACKUP \wedge saveset \wedge tape \wedge (v5.0 \vee version5.0)$.

meisten uninformierten Nutzer auch komplexere Anfragen mit zweifacher Klammerungstiefe korrekt erstellen konnten [YS98].

Eine alternative Darstellung aussagenlogischer Anfragen kommt ganz ohne die Darstellung von Kanten aus [ABF⁺89]. Abbildung 2.5 zeigt eine Beispielanfrage in dieser Darstellung, die die Verknüpfung der verwendeten Filter alleine durch deren Positionierung definiert. Filter werden durch Boxen repräsentiert, die für eine Anfrage aktiviert oder deaktiviert sind. Ein schwarzer Hintergrund einer Filterbox zeigt an, dass der Filter in die aktuelle Anfrage einbezogen wird, ein weißer, dass er nicht einbezogen wird. Übereinander gruppierte aktive Filterboxen entsprechen einer Disjunktion der darunter beziehungsweise darüber befindlichen aktiven Filterboxen, horizontal nebeneinander positionierte aktive Filterboxen sind über ein und in die Anfrage einbezogen.

2. Hintergrund

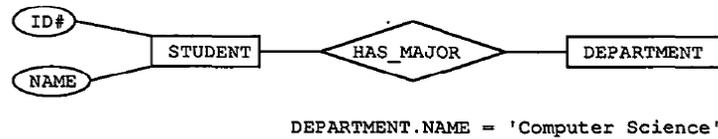


Abbildung 2.6.: Visuelle Anfrage für verknüpfte Daten mittels ER-Diagramm, die die Namen und ID's aller Studenten zurückgibt, die Informatik als Hauptfach studieren. [CEC87].

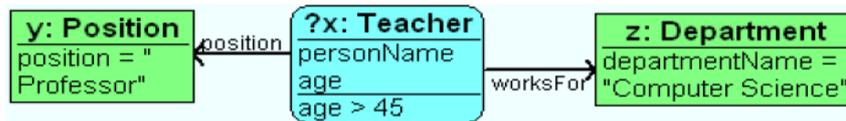


Abbildung 2.7.: Anfrage in GQL (Graphical Query Language), die alle Lehrkräfte in einer RDF-Datenbank findet, die als Professor in der Informatik-Abteilung arbeiten und älter als 45 Jahre sind [BRZ09].

2.2.2. Relationssuche und -visualisierung

Für die visuelle Suche verknüpfter Daten dienen vielen Konzepten *Entity-Relationship*-Diagramme als Grundlage. Eine Beispielanfrage eines Entity-Relationship-Anfragekonzepts ist in Abbildung 2.6 gegeben [CEC87]. Dort werden die Beziehung *HAS_MAJOR* zwischen den zwei Entitäten *STUDENT* und *DEPARTMENT* sowie der Wert ‚Computer Science‘ für deren Eigenschaft *NAME* als Filterkriterien visualisiert. Interessant bei diesem Konzept ist auch die subtraktive Entstehung einer Anfrage, d.h. Eigenschaften, Entitäten und Beziehungen zwischen diesen werden nicht vom Nutzer hinzugefügt, sondern entfernt. Dementsprechend startet eine neue Anfrage mit einem Diagramm der gesamten zugrundeliegenden Datenstruktur, aus der dann die nicht benötigten Elemente entfernt werden. Dadurch müssen sich Nutzer nicht Namen bestimmter Elemente merken bzw. Vorkenntnisse über die Datenstruktur haben.

GQL ist ein visueller Filtermechanismus für RDF-Datenbanken, der SPARQL-Anfragen erzeugt [BRZ09]. Die Visualisierung basiert auf der graphischen UML-Notation. Variablen einer SPARQL-Anfrage werden typisiert, zum Beispiel ist die Variable *x* in Abbildung 2.7 vom Typ *Teacher*, die Variable *y* vom Typ *Position* und die Variable *z* vom Typ *Department*. Ergebnisspalten und Filterkriterien zu einer Variablen werden wie Attribute in UML aufgeführt und können eine Bedingung zugewiesen bekommen, den Ergebnisse erfüllen müssen. So müssen Ergebnisse für die Variable *x* aus Abbildung 2.7 ein Alter größer 45 Jahre aufweisen. Relationen zwischen Variablen, die in RDF-Datenbanken Kanten zwischen Ressourcenknoten entsprechen, werden über beschriftete Kanten zwischen den Variablen der Visualisierung festgelegt, wobei die Beschriftung angibt, über welche IRI die Ressourcen verbunden sein müssen. Abbildung 2.7 zeigt zum Beispiel die Definition einer Beziehung *worksFor* zwischen einem *Teacher* und einem *Department*, die Ergebnisse ebenfalls besitzen müssen. Das Konzept enthält noch weitere Möglichkeiten, etwa die Kennzeichnung optionaler oder verbotener Verknüpfungen.

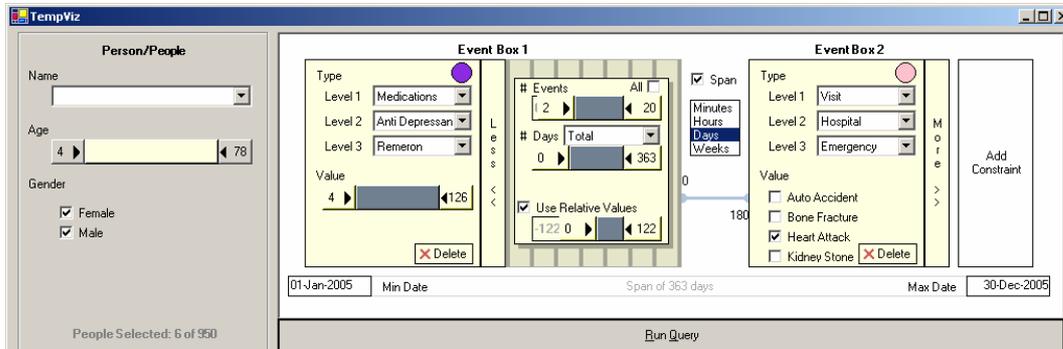


Abbildung 2.8.: Temporale Anfragenformulierung in TempViz [FKSS06].

2.2.3. Temporale Anfragen

Ein Anfragekonzept, das den Schwerpunkt auf temporale Filterung legt, wird von Fails, Karlson, Shahamat und Shneiderman beschrieben [FKSS06]. Nicht zeitbezogene Daten werden ganz links in vorgegebenen Auswahlmöglichkeiten angegeben, Ereignisse werden daneben von links nach rechts angeordnet (siehe Abbildung 2.8). Ein weiter rechts liegendes Ereignis findet später statt als ein weiter links liegendes, auch wenn kein Abstand oder Datum angegeben werden. Durch die Aneinanderreihung von Ereignissen ergibt sich ein Pfad von Ereignissen, der als Muster bezeichnet wird. Allerdings ist nur ein solcher Pfad möglich, verzweigte Zeitabfragen mit alternativen Pfaden können also nicht verwendet werden.

Auf der in Abbildung 2.2 gezeigten visuellen Metapher der Farbstreifen von Chittaro und Combi basiert eine temporale Anfragevisualisierung für klinische Daten von Combi und Oliboni [CO12]. Dieses Konzept erlaubt die aussagenlogische Kombination temporaler Filter. Dazu ist die Arbeitsfläche in zwei Bereiche unterteilt, wie in Abbildung 2.9 zu sehen ist. Der obere Bereich dient dabei der Definition der Zeitintervalle und deren Relationen. Jedes Intervall erhält dabei einen Namen bzw. eine Nummerierung und ist mit einer eigenen Farbe gefüllt. Mehrere Intervalle können zu einer ebenfalls benannten temporalen Abstraktion zusammengefasst werden, die ganz oben über einen Reiter mit ihrem Namen zur Ansicht bzw. Bearbeitung ausgewählt werden kann. Für jede zusammengesetzte Relation können die enthaltenen Intervalle über verschiedene Reiter in unterschiedlichen temporalen Granularitäten dargestellt und bearbeitet werden, in Abbildung 2.9 ist zum Beispiel für die zusammengefasste Abstraktion *pneumonia* die Granularitätsebene *Day* verfügbar. Im unteren Bereich der Arbeitsfläche können die Abstraktionen aussagenlogisch verknüpft werden. Sie werden dort durch mit ihrer Nummerierung beschriftete Kreise und einer eigenen Farbe je Abstraktion repräsentiert. Die Visualisierung von Konjunktionen erfolgt durch die Gruppierung von Kreisen innerhalb eines Rechtecks mit geschlossener Kontur, die Visualisierung von Disjunktionen wird über eine Gruppierung von Kreisen innerhalb eines Rechtecks mit gestrichelter Kontur dargestellt und eine Verneinung durch Durchstreichen eines dieser Rechtecke. Außerdem wird der entstehende aussagenlogische Ausdruck auch noch in Textform über dem aussagenlogischen Kompositionsbereich angezeigt.

2. Hintergrund

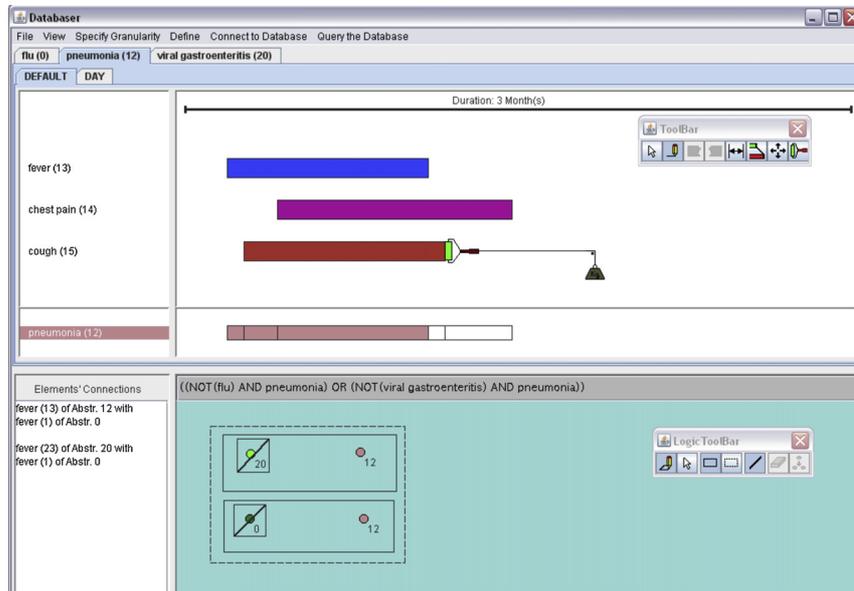
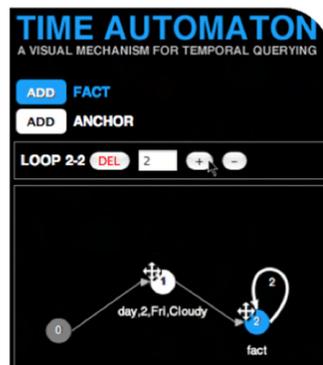
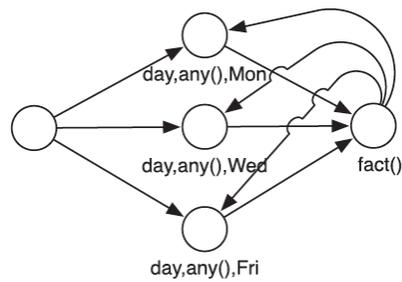


Abbildung 2.9.: Aussagenlogische Verknüpfung von Intervallen und temporalen Abstraktionen [CO12].

Nach dem Vorbild eines endlichen Automaten wurde das *Time Automaton*-Konzept entwickelt [CGB13]. Es benötigt allerdings ein spezielles Datenformat, das in der Konzeptbeschreibung *Temporal String* genannt wird. Der endliche Automat, der vom Nutzer erstellt wird, durchsucht dann diesen *Temporal String* wie ein Parser. Er unterscheidet zwischen zwei verschiedenen Knotentypen, den Ankerknoten, die temporale Informationen enthalten, und den Faktenknoten, die Daten enthalten, die an bestimmte Zeitpunkte gekoppelt sind. Das Interessante an dieser Visualisierungsmethode einer temporalen Anfrage ist die Möglichkeit, auch wiederkehrende Fakten mit komplizierten zeitlichen Abstandsmustern zu finden. Außerdem sind verzweigte Zeitanfragen möglich, da ein Knoten des endlichen Automaten verschiedene Übergänge besitzen kann. Verschiedene Granularitäten können über die Aneinanderreihung von temporalen Knoten auf einem Pfad erreicht werden, von denen jeder die Granularität weiter verfeinert, zum Beispiel wählt ein erster Knoten ein bestimmtes Jahr, sein Nachfolgeknoten dann einen bestimmten Monat, und dessen Nachfolgeknoten dann einen bestimmten Tag.



(a) Anfrageeditor des *Time Automaton*-Konzepts.



(b) Eine Beispielanfrage im *Time Automaton*-Konzept. Ergebnisse sind alle Fakten, die an einem Montag, Mittwoch oder Freitag auftraten.

Abbildung 2.10.: Anfrageeditor und ein Anfragebeispiel in *Time Automaton* [CGB13].

3. Konzept

Das in diesem Kapitel vorgestellte, neu entwickelte Konzept einer visuellen Anfragesprache für temporale Filterung graphbasierter Daten soll ermöglichen, sowohl einfache Anfragen, bestehend aus nur wenigen konjunktiv verbundenen Filterkriterien, als auch komplexe Anfragen mit vielen Filtern, konjunktiv, disjunktiv und temporal verknüpft, zu erstellen. Dabei sollen relative Bezüge der temporalen Filter untereinander ebenso ermöglicht werden wie Filterung mittels absoluter Zeitangaben. Eine wichtige Eigenschaft dieses Konzepts ist die visuelle Anordnung der temporalen Filter, die deren Reihenfolge anzeigt und so zu einer intuitiven Übersicht der Anfrage verhelfen soll. Trotz dieser festen Anordnung ermöglicht das Konzept beliebige aussagenlogische Verknüpfungen der Anfragenfilter.

3.1. Zeit

Im Folgenden wird zwischen zwei temporalen Anordnungsformen unterschieden. Die *relative Ordnung* temporaler Filter umfasst die Anordnung dieser durch die in Kapitel 2.1.1 vorgestellten Relationen. *Absolute Ordnung* temporaler Filter liegt vor, wenn deren Position auf der Zeitachse durch Angabe eines Datums festgelegt ist. Beide Ordnungen können gemischt werden.

Im hier vorgestellten Konzept wird Zeit als Objekt für das Modellieren von Zeitpunkten und Abständen temporaler Filter benötigt. Die Klassifizierung des genutzten Zeitmodells erfolgt nach den von Frank vorgeschlagenen orthogonalen Eigenschaften Skala, Definitionsbereich, Anordnung und Sichtweise [Fra98].

Aufgrund der absoluten Ordnung reicht eine ordinale Skala nicht aus. Stattdessen wird eine diskrete Skala verwendet. Eine kontinuierliche Skala wurde nicht für das Konzept nicht eingesetzt, da die diskrete Skala zur Vereinfachung für den Nutzer sowie als Vereinfachung einer Implementierung dienen soll, da diskrete Zeitangaben in Computersystemen üblich sind, beispielsweise für die Systemzeit in Unix-Systemen. Die Skala unterstützt verschiedene Granularitäten, mit der kleinsten Einheit Sekunde. Je nach Anwendungsfall könnte diese natürlich entsprechend kleiner gewählt werden.

Als Definitionsbereich wird eine punktförmige Abstraktion gewählt. Zwar argumentiert Allen, dass Ereignisse in der realen Welt nur als Zeitspannen auftreten, da sich auch auf den ersten Blick instantan scheinende Ereignisse weiter zerlegen lassen [AA83], jedoch können Zeitpunkte hier aufgrund der begrenzten Granularität von Zeitangaben als Intervalle aufgefasst werden. Bezüglich einer Auflösung, die als kleinste Einheit Sekunden verwendet, ist ein Zeitpunkt auf einer kontinuierlichen Skala ein Intervall der Länge einer Sekunde. Die Auffassung als Zeitpunkt

3. Konzept

hat den Vorteil, dass sie unter anderem der umgangssprachlichen Definition entspricht (vgl. 2.1.1) und damit eventuell für den Nutzer intuitiver ist. Weiterhin sind Zeitpunkte bequemer zu definieren, da oft nicht beide Intervallenden eines Ereignisses bekannt sein müssen oder sollen, was hier nur das Hinzufügen eines Zeitpunkts anstatt der Definition zweier Zeitpunkte oder einer Dauer, wie im Fall eines Intervalls, erfordert. Für die Visualisierung zeitlich verzweigter Ereignisse muss so außerdem keine Methode der Darstellung von aufgespaltenen Intervallen gefunden werden. Ein Nachteil des Punktkonzepts könnte allerdings sein, dass nicht sofort erkennbar ist, welche Start- und Endpunkte zum selben Intervall gehören.

Weil das Konzept eine Sortierung nach absoluter und relativer Zeitordnung vorsieht, wird keine zyklische, sondern eine lineare Anordnung der Ereignisse gewählt. Durch die lineare Sortierung soll schnell erkennbar sein, in welcher Reihenfolge Ereignisse auftreten, unter anderem auch durch die weiter Verbreitung der zeitstrahlartigen lineare Anordnung temporaler Daten.

Um auch die Modellierung komplexer Anfragen zu ermöglichen, wird auf eine verzweigende Zeitsicht gesetzt. Verzweigend bedeutet in diesem Kontext, dass die Struktur der Zeitachse nicht nur einen Zeitverlauf angeben kann, sondern dass auch alternative Verläufe von Abschnitten zulässig sind. Diese Zeitsicht ermöglicht die Formulierung temporaler Disjunktionen für das Konzept, also die Möglichkeit, dass Anfrageergebnisse nicht nur eine bestimmte temporale Konstellation erfüllen müssen, sondern dass mehrere alternative Verläufe definierbar werden.

3.2. Anfragenstruktur

Eine Anfrage besteht aus *Subjekt(en)*, *Prädikat(en)* und *Objekt(en)*. Subjekte entsprechen Strukturen in der objektorientierten Programmierung. Sie bündeln thematisch zusammengehörende, für das Subjekt zulässige Prädikate. Prädikate bezeichnen die zu filternde Eigenschaft eines Subjekts und müssen in einer Anfrage immer auf ein bestehendes Subjekt bezogen sein. Jedem Prädikat muss außerdem ein Objekt zugeordnet sein, welches den Wert definiert, den Ergebnisse für die entsprechende Eigenschaft besitzen müssen.

Ein Beispiel wäre ein Subjekt *Person*, das Prädikate, also für dieses Subjekt zulässige Filtereigenschaften, wie *Vorname*, *Name* und *Geburtsdatum* bündelt. *Vorname*, *Name* und *Geburtsdatum* müssten bei Nutzung in einer Anfrage dabei um Objekte, also Filterwerte, ergänzt werden. Für diese Beispielprädikate wären Text- beziehungsweise Datumsliteral gültige Objekte. Eine vollständig definierte Anfrage hätte zum Beispiel das Subjekt *Person*, das Prädikat *Vorname* und dessen Textliteral-Objekt 'Max'.

Anfragen werden als Graphen modelliert, mit Subjekten als Startknoten, sowie Kanten, die die Zugehörigkeit von Prädikaten zu Subjekten und aussagenlogische Bedeutung der Anfrage festlegen.

3.3. Filter

Da jedem Prädikat ein Objekt zugeordnet werden muss, werden diese beiden zu einem gemeinsamen visuellen Konzept, einem Filter, zusammengefasst. Einerseits soll dies die Zusammengehörigkeit von Objekt und Prädikat hervorheben, andererseits beim Hinzufügen eines Prädikats automatisch das obligatorische Objekt erzeugen.

Das Prädikat eines Filters gibt an, welche Eigenschaft eines Subjekts Ergebnisse einschränken soll. Da Filter immer einem Subjekt zugeordnet werden müssen, kann die Auswahl der möglichen Prädikate durch den Subjekttyp festgelegt werden, damit zum Beispiel für ein Subjekt vom Typ *Person* kein unsinniges Prädikat, etwa *Anzahl Räder*, angegeben werden kann.

Das Objekt definiert den Wert, den die Ergebnisse für die durch das Prädikat definierte Eigenschaft besitzen müssen. Es kann sich sowohl um ein Literal als auch, bei bestimmten Filtertypen, um Verbindungen zu anderen Prädikaten handeln. Im Falle eines Literals sind zusätzlich verschiedene Relationen zum Objektwert, abhängig vom Objekttyp, erlaubt. Zum Beispiel kann für ein numerisches Literal die *größer-als*-Relation zum gewählten Zahlenwert definiert werden. Auch leere Objekte sind erlaubt. Die Semantik eines leeren Objekts ist, dass die Ergebnisse zwar eine Eigenschaft vom festgelegten zugehörigen Filterprädikat besitzen müssen, aber beliebige Werte für dieses annehmen dürfen.

Anhand der erlaubten Werttypen für die Objekte eines Filters werden Filter in drei verschiedene Typen unterteilt:

Fakt Als Fakten werden alle Filter klassifiziert, deren Objekt Literale vom Typ Zeichenkette oder Zahl akzeptiert.

Ereignis Als Ereignisse werden alle Filter klassifiziert, deren Objekt Literale vom Typ Datum akzeptiert.

Subjektverknüpfung Als Subjektverknüpfungen werden alle Filter klassifiziert, deren Objekt kein Literal, sondern nur Referenzen zu anderen Subjektverknüpfungsfilttern akzeptiert.

3.3.1. Darstellung

Alle Filter werden von einem oder zwei Labels und einem Kreis repräsentiert. Textfelder und Kreis sind mit einem kleinen vertikalen Abstand untereinander angeordnet und deren Mittelpunkte horizontal zentriert. Über dem Kreis steht als Beschriftung immer das Prädikat, bei Fakten und Ereignissen auch noch das Objekt, wie am Beispiel eines Faktenknotens in Abbildung 3.1a gezeigt. Der Kreis eines Filters stellt die Verbindungsfläche zu den Kanten eines Anfragegraphen dar, entspricht also einem Knoten in üblichen Graphdiagrammen. Die gewählte Anordnung und Nähe der Filterkomponenten soll die Zusammengehörigkeit von Prädikat, Objekt und Knoten intuitiv begreifbar machen, da nah beieinander liegende Objekte als Gruppierung wahrgenommen werden [War12, S. 189–190].

3. Konzept

Mit steigender Anzahl verbundener Kanten wächst die Knotengröße eines Filters, damit der Abstand zwischen den Verbindungen des Knotens gleich bleiben kann (siehe Abbildung 3.1c). Bei immer kleiner werdenden Abständen könnte nicht mehr gut zwischen den Pfaden unterschieden werden, ebenso wäre die Auswahl eines bestimmten Pfades für Interaktion nicht mehr leicht möglich. Weiterhin zeigt die Knotengröße die Signifikanz eines Knotens in einem Anfragegraphen an, da direkt sichtbar ist, ob ein Knoten für viele oder nur wenige Pfade von Bedeutung ist. Im aktuellen Stadium des Konzepts ist die Größe eines Knotens nicht beschränkt, was bei sehr vielen mit dem selben Filterknoten verbundenen Kanten, besonders bei kleiner Displaygröße, zu unübersichtlichen Anfragen führen kann, da ein Knoten beliebig groß werden kann. Vermieden werden können zu große Filterknoten aber leicht durch das Hinzufügen weiterer, identischer Filterknoten, auf die dann einige Pfade verteilt werden. Dass dies semantisch äquivalent ist, wird in Abschnitt 3.4.1 erläutert. Weitere Abhilfe könnte eine Zoom-Funktion schaffen.

Ware führt an, dass Farbe sehr effektiv zur Klassifizierung von Objekten dient und präattentiv wahrgenommen wird [War12, S. 123–127]. Um den Überblick für eine vorhandene Anfrage möglichst effizient zu gestalten, sind die unterschiedlichen Filtertypen in diesem Konzept deshalb unterschiedlich eingefärbt. Zum einen soll der Nutzer so schnell erkennen können, welche Informationen eine Anfrage filtert, also ob sie zum Beispiel temporale Filter besitzt. Zum anderen soll die gezielte Erfassung einer Filterkategorie gewährleistet werden, indem sich der Nutzer auf eine Farbe konzentrieren kann, die gut von den anderen in der Anfrage präsenten Farben unterscheidbar ist. Die für die Einfärbung der Filter verwendeten Farben sind dabei den von Ware empfohlenen 12 Klassifizierungsfarben entliehen [War12, S.125–127]: Fakten sind grün, Ereignisse blau und Subjektverknüpfungen türkis eingefärbt. In den gewählten Blau- und Türkistönen sind nur sehr geringe Rotanteile enthalten, um die Unterscheidung der Filter für Menschen mit Rot-Grün-Schwächen zu erleichtern. Der Grünton enthält einen großen Rotanteil, im Gegensatz zu den beiden anderen Farbtönen aber fast keinen Blauanteil, und sollte für die genannte Gruppe gut von den anderen Farben zu unterscheiden sein. Da die Rot-Grün-Schwäche die häufigste Farbsinnstörung ist, wird durch deren Berücksichtigung damit der größte Teil farbsinngestörter Personen einbezogen [SZ12, S. 268]. Andere Farbwahrnehmungsstörungen wurden nicht beachtet. Das Farbschema ist nicht für Szenarien wie Schwarz-Weiß-Druck geeignet, da die Helligkeitsunterschiede in den Farben gering ausfallen.

Visuelle Unterschiede zwischen den verschiedenen Filtertypen wurden, abgesehen von der Klassifizierungsfarbe, bewusst gering gehalten, um dem Nutzer zu verdeutlichen, dass diese einem einheitlichen Semantik- und Interaktionsprinzip folgen.

3.3.2. Interaktion

Durch Interaktion mit dem Prädikatstext öffnet sich eine Liste, die eine Auswahl aus allen durch das mit dem Filter verbundene Subjekt erlaubten Prädikaten erlaubt. Per Interaktion mit dem Objekttext öffnet sich, sofern ein Prädikat ausgewählt wurde, ein Popover-Menü, das die Eingabe eines für das Objekt erlaubten Wertes ermöglicht. Ein Beispiel für ein Popover eines numerischen Literals ist in Abbildung 3.2 gegeben. Sowohl numerischer Wert als auch Relation zu diesem Wert können gewählt werden. Bei Subjektverknüpfungsfilttern wird das Objekt, also eine Kante zu

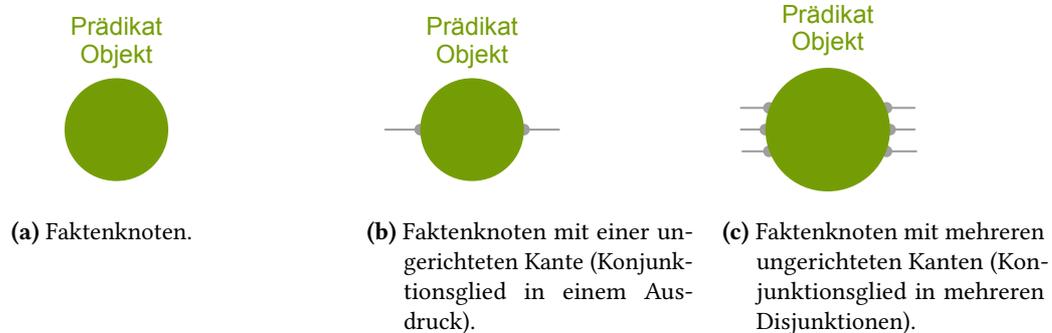


Abbildung 3.1.: Visuelle Repräsentation eines Faktenknotens und seiner Verknüpfung mit aussagenlogischen Ausdrücken.

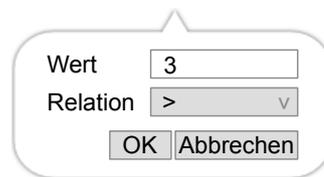


Abbildung 3.2.: Popover, das die Auswahl des numerischen Literals 3 mit Relation *größer als* zeigt. Ergebnisse für einen numerischen Filter mit diesem Literal müssen für das zugehörige Prädikat Werte größer drei aufweisen.

einem anderen Subjektverknüpfungsfiler, ausgewählt, indem deren Knoten per Drag'n'Drop verbunden werden. Eine Drag'n'Drop-Aktion schien hierfür intuitiver, da sie die Verbindung zwischen zwei Subjekten direkter und nachvollziehbarer gestaltet, als die Auswahl aus einer Liste oder per anderen Interface-Elementen.

Interaktion mit einer aussagenlogischen Kante führt zu einer farblichen Hervorhebung der Kante und aller Knoten, die durch diese Hyperkante verbunden sind.

3.3.3. Fakten

Fakten besitzen sowohl Prädikats- als auch Objektbeschriftung. Abbildung 3.1a zeigt die visuelle Repräsentation eines Fakts. Sie filtern Prädikate, für deren Objekt Daten vom Typ Zeichenkette oder Zahl akzeptiert werden.

Als Erweiterung der möglichen Objekttypen wären auch Referenzen denkbar. Um die Unterscheidung von Subjektverknüpfungsfilern mit diesem erweiterten Objektdefinitionsbereich deutlich zu machen, würde ein Filter genau dann als Fakt klassifiziert, wenn für sein Objekt Daten vom Typ Zeichenkette oder Zahl sowie optional Referenzen zu einem anderen Fakt gleichen Datentyps zulässig sind. Diese Erweiterung wäre sinnvoll, wenn für die Referenz zwischen zwei Fakten eine Relation gewählt werden muss. Beispielsweise könnten für Zeichenketten Relationen *gleich* und

3. Konzept

ungleich sowie für Zahlen zusätzlich Relationen *kleiner-gleich*, *kleiner*, *größer-gleich* und *größer* erlaubt werden. Damit könnten etwa Anfragen erstellt werden, die Paare von Musikern und Politikern ungleichen Nachnamens ausgeben.

3.3.4. Ereignisse

Die Angabe eines Literals ist bei Ereignissen optional. Ist ein Literal angegeben, handelt es sich um ein *absolutes* Ereignis, ansonsten um ein *relatives* Ereignis, das sich zu einem beliebigen Zeitpunkt zutragen darf.

Das Literal eines Ereignisses definiert seine absolute Position oder seine relative Position zu einem absoluten Punkt auf dem Zeitstrahl. Ersteres wird durch eine Angabe eines Zeitobjekts erreicht, letzteres durch Angabe eines Zeitobjekts und einer der folgenden Relationen: *vor*, *vor-oder-gleichzeitig*, *gleichzeitig*, *gleichzeitig-oder-nach* oder *nach*. Zum einen erlaubt dies weitere Filteroptionen, etwa die Ausgabe aller Menschen, die vor einem gewissen Datum geboren wurden, zum anderen hilft dies, mit einer gewissen Unsicherheit umzugehen. Ist zum Beispiel nur bekannt, dass ein Mensch zwischen 1800 und 1850 geboren wurde, können einfach zwei Geburtsdatums-Filter konjunktiv mit dem Subjekt Mensch verknüpft werden, wobei der eine die Einschränkung *gleichzeitig-oder-nach 1800* und der andere die Bedingung *vor-oder-gleichzeitig 1850* definiert.

Optisch hebt sich ein Ereignisfilter nicht nur durch seine Farbe, sondern auch durch den ihm eigenen äußeren Kreisring ab, den Fakten und Subjektverknüpfungsfiler nicht besitzen, wie in Abbildung 3.3a zu sehen ist. Der innere, gefüllte Kreis eines Ereignisses dient genau wie bei Fakten- und Subjektverknüpfungsfilern zur aussagenlogischen Verknüpfung und Subjektzugehörigkeitsanzeige, bekommt aber im Zusammenspiel mit dem umgebenden Kreisring noch eine zusätzliche Bedeutung für die Darstellung temporaler Relationen.

Ereignisse können per Drag'n'Drop relativ zueinander angeordnet werden. Dabei wird eine neue violettfarbene Kante mit zwei Endpunkten zwischen den zu verbindenden Ereignissen erzeugt, die anzeigen soll, dass diese Ereignisse in Zusammenhang stehen (siehe Abbildung 3.3b). Die Art des Zusammenhangs ist die relative temporale Anordnung der Ereignisse untereinander, also die Information, welches Ereignis vor einem anderen Ereignis stattfindet, beziehungsweise welche Ereignisse gleichzeitig stattfinden. Für ein Ereignis sind beliebig viele dieser temporalen Kanten zulässig, die dann gleichmäßig um den vertikalen Mittelpunkt des Kreisrings verteilt werden, um leichte Unterscheidbarkeit zwischen den einzelnen Kanten und Interaktion mit diesen zu gewährleisten.

Der Abstand eines Endpunkts zum inneren Kreis eines Ereignisses veranschaulicht dabei die Nähe der über diese Kante temporal angeordneten Ereignisse. Je näher ein solcher Punkt zum inneren Kreis liegt, desto näher liegen die verbundenen Ereignisse im Sinne der Relationen *vor/nach* bis *gleichzeitig* zusammen. Für diesen Abstand sind drei diskrete Positionen möglich:

1. **Endpunkte liegen direkt auf dem inneren Kreis:** Werden die Endpunkte direkt mit den inneren Kreisen zweier Ereignisse verbunden, so gibt es keinen Abstand zwischen ihnen. Zwei dergestalt verbundene Zeitpunkte finden also gleichzeitig statt (siehe Abbildung 3.7c).

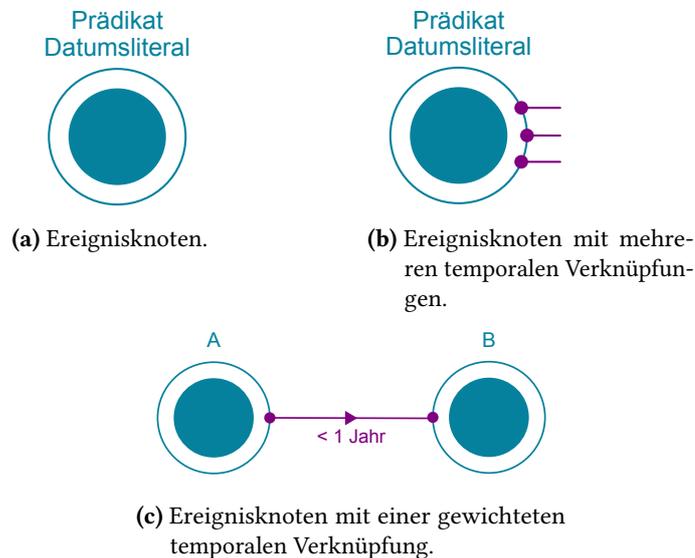


Abbildung 3.3.: Visuelle Repräsentation eines Ereignisknotens und seiner temporalen Verknüpfungen.

2. **Endpunkte liegen direkt auf dem äußeren Kreisring:** Bei der Verbindung mit dem äußeren Ring ist der Abstand zwischen den verbundenen Ereignissen am größten und zeigt an, dass zwei so verbundene Ereignisse echt auseinander liegen, wie es bei einer *vor*- beziehungsweise *nach*-Relation der Fall ist (siehe Abbildung 3.7a).
3. **Endpunkte liegen zwischen innerem Kreis und äußerem Kreisring:** Der Abstand der verbundenen Ereignisse liegt bei dieser Art der Verbindung zwischen den genannten Extremen, liegt also zwischen *gleichzeitig* und *vor* beziehungsweise *nach*. Diese Verknüpfung modelliert die *vor-oder-gleichzeitig* und *gleichzeitig-oder-nach*-Relationen (siehe Abbildung 3.7b).

Analog zur Objektfestlegung eines absoluten Ereignisses kann auch für diese temporalen Relationen eine absolute Zeitangabe zur Definition eines genauen Abstands oder eine Relation zu einer absoluten Zeitangabe für einen Mindest- oder Höchstabstand notiert werden. Dies geschieht per Interaktion mit einer temporalen Verbindung, bei der sich ein Popover-Menü öffnet, über welches der Abstand definiert werden kann. Der angegebene Abstand wird dann unterhalb der Verbindung angezeigt, wie in Abbildung 3.3c zu sehen ist.

3.3.5. Subjektverknüpfungen

Subjektverknüpfungsfiler unterscheiden sich optisch von Fakten und Ereignissen durch das Fehlen eines Literaltexts. Stattdessen werden als Objekt eines Subjektverknüpfungsfilters Kanten zu anderen Subjektverknüpfungsfiltern festgelegt, wobei die Filter nicht mit dem selben Subjekt verbunden sein dürfen. Die Semantik zweier dergestalt verbundenen Filter ist die Filterung von Subjekten, die sich gegenseitig referenzieren, das heißt, die Subjekte besitzen innerhalb

3. Konzept

der Graphdatenbank eine Kante, über die ihre den Prädikaten der Subjektverknüpfungsfiler entsprechenden Knoten verbunden sind.

Zur einfacheren Anwendung durch den Nutzer müssen vor dem Verknüpfen zweier Subjektverknüpfungsfiler nicht beide Prädikate festgelegt werden, denn durch Festlegung eines Prädikates und der anschließenden Verknüpfung kann das andere Prädikat bei bekanntem Schema der Graphdatenbank automatisch ausgewählt werden. Die Verknüpfung von Subjektverknüpfungsfilern erfolgt analog zur temporalen Verknüpfung von Ereignissen per Drag'n'Drop.

3.4. Anfragen

Modelliert wird eine Anfrage durch einen oder mehrere Hypergraphen $G = (V, E)$, in dem Subjekte und Filter durch Knoten repräsentiert werden. Eine Anfrage $A = (V', E')$ ist damit definiert als

$$(3.1) A := \bigcup G$$

mit der Menge der Subjektknoten

$$(3.2) SK := \{v \in V' : v \text{ ist Subjektknoten}\}.$$

Graue Kanten kennzeichnen die Zugehörigkeit von Filtern, und damit die deren Prädikate, zu Subjektknoten, und definieren die aussagenlogische Kombination der Filter. Bezeichnet man die Menge dieser aussagenlogischen Kanten der Hypergraphen als B , die Menge der temporalen Kanten als T und die Menge der Subjektverknüpfungskanten als L , kann die Kantenmenge von A als Zerlegung

$$(3.3) E' = B \cup T \cup L$$

geschrieben werden.

Subjekte sind die Startknoten eines Anfragegraphen. Für die Verwendung eines Subjektknotens sprechen unter anderem die Ergebnisse von Zviedris und Liepiņš, die beim iterativen Design einer visuellen Anfragesprache zu dem Ergebnis kamen, dass ein Startelement als Ausgangspunkt einer Anfrage kritisch für das Verständnis der Anfrage ist [ZL14]. In einer Anfrage sind mehrere Subjekte des gleichen Typs erlaubt, um Beziehungen wie zum Beispiel Verwandtschaft zwischen zwei Personen modellieren zu können. Damit in einem solchen Fall unterscheiden werden kann, welches Subjekt zu welchem Ergebnis gehört, sind Subjektknoten mit einer eindeutigen Instanznummer gekennzeichnet. Zum Beispiel hat das Subjekt in Abbildung 3.4 die Instanznummer $S1$, die es von allen anderen eventuell in der Anfrage vorhandenen Subjekten unterscheidet.

Bei der Erzeugung eines neuen Subjektknotens wird auch ein neuer, zunächst von eventuell bereits innerhalb der Anfrage vorhandener Graphen disjunkter Graph erzeugt, dessen einziger Knoten der neue Subjektknoten ist. Er enthält auch eine Kante, die im Subjektknoten beginnt, aber in keinem anderen Knoten endet. Eine solche Kante ist in jedem Graphen einer Anfrage



Abbildung 3.4.: Subjektknoten für das Subjekt *Person* mit Instanznummer *P1*

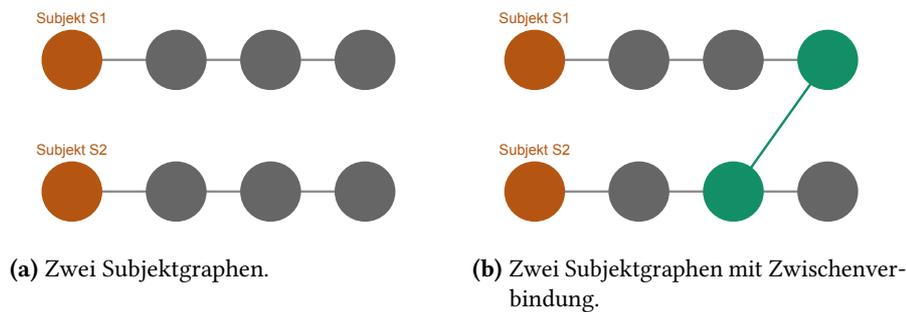


Abbildung 3.5.: Unverbundene und verbundene Subjektgraphen der Subjekte S1 und S2. Graue Kanten repräsentieren Verbindungen innerhalb eines Subjektgraphen, graue Knoten beliebige Filter.

vorhanden und erlaubt das Hinzufügen von Disjunktionen. Wird dieser Kante ein neuer Knoten hinzugefügt oder wird sie mit einem vorhandenen Knoten verbunden, also endet die Kante dann auch in einem Knoten, entsteht automatisch eine neue Kante, die nur mit dem Subjektknoten verbunden ist. Für die Aussage der Anfrage selbst ist diese Kante ohne jede Bedeutung.

Disjunkte Graphen innerhalb einer Anfrage, wie die in Abbildung 3.5a gezeigten, können über Kanten zwischen Subjektverknüpfungsknoten oder temporale Kanten zwischen Ereignissen unterschiedlicher Subjekte verbunden werden (siehe Abbildung 3.5b), wobei die Graphen, die ohne diese Zwischenverbindungen disjunkt wären, als *Subjektgraphen* bezeichnet werden. Für einen Subjektknoten $s \in SK$ ergibt sich als formale Definition des zugehörigen Subjektgraphen S also

$$(3.4) \quad \begin{aligned} S &:= (V'', E'') \\ \text{mit } E'' &:= \{e \in E' : s \in e \wedge e \in B\} \\ \text{und } V'' &:= \{v \in e : e \in E''\}. \end{aligned}$$

Für alle weiteren zu einem Subjektgraphen hinzugefügten Filter ist der Startknoten immer automatisch das Subjekt, so dass dieses nicht manuell festgelegt werden muss, sondern implizit durch die Auswahl des Subjektgraphen, an welchen der Filter angefügt wird, gegeben ist. Wird der Subjektknoten eines Subjektgraphen gelöscht, wird der gesamte Subjektgraph und alle seine Zwischenverbindungen zu anderen Subjektgraphen gelöscht.

3.4.1. Logische Verknüpfung der Filter

Wie bereits ausgeführt, werden die Filterknoten durch graue Kanten der Kantenmenge B zu logischen Ausdrücken verknüpft. Alle Filter, deren Knoten auf einem gemeinsamen grauen Pfad liegen, sind per Konjunktion verbunden. Verschiedene Pfade, die mit dem Subjektknoten verbunden sind, bilden Disjunktionen, so dass für jedes Subjekt eine Disjunktion von Konjunktionen, also eine Disjunktive Normalform, entsteht. Die Anfrageformel F_{snc} für die logischen Pfade eines Subjekts $s \in SK$, die keine Knoten mit Verknüpfungen zu anderen Subjektgraphen enthalten, ergibt sich damit zu

$$(3.5) \quad F_{snc} := \bigvee_{b \in B \text{ mit } s \in b \wedge \forall v \in b: (\forall t \in T: v \notin t \wedge \forall l \in L: v \notin l)} \left(\bigwedge_{v \in b} v \right).$$

Temporale Verknüpfungen oder Subjektverknüpfungen zwischen zwei Subjektgraphen $s \neq s'$, wie in Abschnitt 3.4 beschrieben, bilden alle möglichen Konjunktionen der logischen Pfade der verschiedenen Endknoten, zwischen denen die Verknüpfung zwischen den Subjektgraphen besteht. Dies lässt sich durch

$$(3.6) \quad F_{sc} := \bigvee_{b, b' \in B \text{ mit } s \in b, s' \in b' \wedge \exists e = \{u, v\} \in E': ((e \in T \vee e \in L) \wedge u \in b \wedge v \in b')} \left(\bigwedge_{v \in (b \cup b')} v \right)$$

beschreiben.

Die Anfrageformel eines Subjekts ergibt sich dann als Disjunktion aus den mit dem Subjekt verbundenen Pfaden, auf denen nur Knoten ohne Verknüpfungen zu anderen Subjektgraphen liegen, mit den Pfaden, auf denen Knoten mit anderen Subjektgraphen verknüpft sind. Pro Subjekt $s \in SK$ ist also die von den Ergebnissen zu erfüllende Formel

$$(3.7) \quad F_s := F_{snc} \vee F_{sc}$$

gegeben. Die gesamte Anfrage wird dann per Disjunktion aller für die einzelnen Subjekte entstandenen Formeln beschrieben:

$$(3.8) \quad F_A := \bigvee_{s \in SK} F_s.$$

Eine logische graue Kante zwischen einem Subjektknoten und einem Filterknoten ordnet außerdem das Prädikat des Filters diesem Subjektknoten zu. Ein Filterknoten kann immer nur mit einem Subjekt verbunden sein. Allerdings können Filterknoten mit mehreren Kanten verbunden werden, so dass sie als Konjunktionsglied in mehreren Disjunktionen nutzbar sind, was die Übersicht durch die Einsparung identischer Knoten verbessert. Außerdem kann Zeit bei der Anfragenerstellung eingespart werden, da die mehrfache Definition eines Filters durch eine Drag'n'Drop-Aktion zwischen Knoten und zu verbindenden Pfaden ersetzt wird.

Im Gegensatz zu Konzepten zur Visualisierung aussagenlogischer Anfragen wie sie in Abschnitt 2.2.1 beschrieben wurden, werden Pfade für die Darstellung einer Disjunktion nicht aufgespalten.

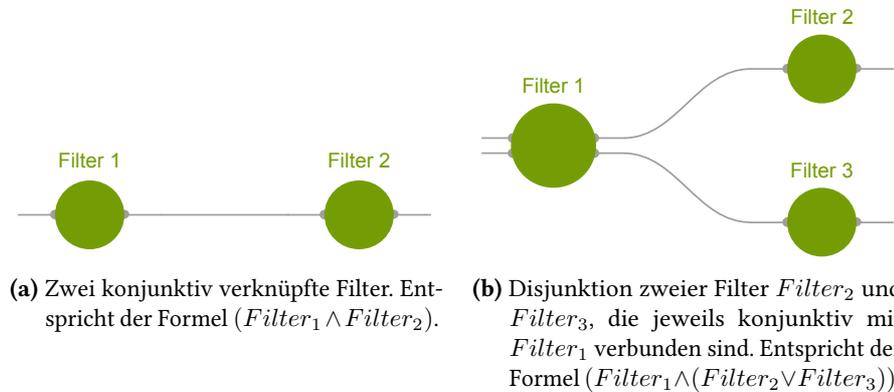


Abbildung 3.6.: Logische Verknüpfung von Filtern durch graue Hyperkanten.

Stattdessen wird für jede Disjunktion ein eigener Pfad angezeigt. Diese Methode erlaubt die Bildung beliebiger aussagenlogischer Ausdrücke trotz fester Anordnung der temporalen Filter.

Negationen können pro Objekt festgelegt werden. Die Idee, die Verbindungspunkte zwischen Filter und Kante für die Darstellung von Negationen zu nutzen, würde Mehrfachdefinitionen eines Filters einsparen, für den Fall, dass ein Filter in manchen Konjunktionen als Negation gewertet werden soll und in anderen ohne diese. Gegen diese Idee sprach schließlich, dass besondere Hervorhebungen für diese Verbindungspunkte aber allesamt sehr klein und damit schnell zu übersehen waren. Eine Vergrößerung dieser Verbindungspunkte kam auch nicht in Frage, da ansonsten auch die Kreise der Filter vergrößert werden müssten, da zwischen den Kanten ein gewisser Abstand zur besseren Unterscheidung herrschen sollte, was die Filter sehr schnell sehr groß und die Anfrage damit unübersichtlich erscheinen ließe. Außerdem sieht das Konzept schon die Nutzung von Relationen für die Definition von Objektliteralen vor, so dass deren Auswahl einfach um eine entsprechende Option ergänzt wird. Ein numerisches Literal erhält zum Beispiel als Negation der Gleichheit die zusätzliche Relation *ungleich*. Die Darstellung der Negation erfolgt dann integriert in den Text für das Objektliteral.

3.4.2. Anordnung der Filter

Ganz links werden immer die Subjektknoten der Graphen angezeigt, um sie als Startknoten hervorzuheben. Deren Nutzen als Startknoten wurde schon in Abschnitt 3.2 begründet. Sind mehrere Subjektknoten in einer Anfrage vorhanden, werden diese vertikal zueinander angeordnet. Neben diesen werden dann die zugehörigen Fakten- und Subjektverknüpfungsfiler von links nach rechts baumartig auf den entsprechenden Pfaden aufgereiht, aufsteigend sortiert nach Erzeugungszeitpunkt innerhalb eines Pfades. Rechts von den Faktenknoten werden die Ereignisse hierarchisch von links nach rechts angeordnet. Die horizontale Ebene eines Ereignisses wird zunächst nach seiner absoluten und anschließend nach seiner relativen temporalen Informationen bestimmt, zeitstrahltypisch von früher nach später. Alle Ereignisse, die keine sortierungsrelevanten Zeitinformatoren liefern, also keine absolute Zeitangabe definiert haben und mit keinem anderen

3. Konzept

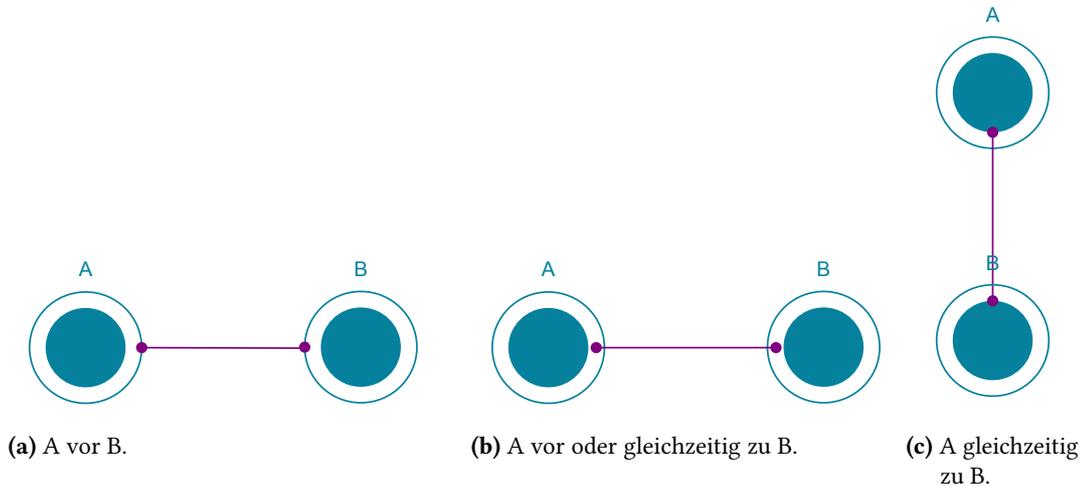


Abbildung 3.7.: Die drei möglichen relativen Verknüpfungen zweier Ereignisse A und B in Punktform.

Ereignis über eine Zeitkante relativ verbunden sind, werden in der ersten Ereignisknotenebene übereinander angeordnet. Die Anordnung der Ereignisse ist global, d.h. für alle Subjektgraphen einer Anfrage gleich.

Die feste, horizontal zeitstrahlbasierte Anordnung der Ereignisknoten soll die zeitliche Sortierung für den Nutzer leicht erkennbar darstellen, indem er schon durch die Anordnung der Ereignisse erkennen kann, in welcher Reihenfolge sie stattfinden sollen. Trotz der festen Anordnung dieser werden beliebige aussagenlogische Ausdrücke unterstützt, da ein Knoten mit jedem beliebigen Disjunktionsglied der Anfrage konjunktiv verbunden werden kann. Anhand der vertikalen Positionierung der Ereignisse, die dem Positionierungsprinzip der Fakten und Subjektverknüpfungsfiler folgt, soll die Zugehörigkeit zu einem bestimmten Subjekt gut erkennbar sein.

Horizontale Abstände zwischen Ereignissen zeigen ihre relative Position an. Wenn zwei Ereignisse A und B relativ zueinander angeordnet werden, also per A vor B , A vor oder gleichzeitig zu B , A gleichzeitig zu B bzw. mit den Inversen dieser Relationen, werden zwei Visualisierungsoptionen genutzt, um die Anordnung darzustellen:

1. Die horizontale Positionierung der Ereignisse soll einen schnellen, groben Überblick über die Reihenfolge der Ereignisse geben. Zwei voreinander positionierte Elemente bedeuten dabei, dass diese echt voreinander (siehe Abbildung 3.7a) bzw. voreinander oder gleichzeitig (siehe Abbildung 3.7b) angeordnet sind. Zwischen zwei gleichzeitig stattfindenden Ereignissen gibt es keinen horizontalen Abstand. Sie werden übereinander angeordnet (siehe Abbildung 3.7c).
2. Für den Fall, dass nicht nur ein Überblick über die Reihenfolge, sondern exaktes Wissen über die temporalen Beziehungen notwendig ist, wird die Visualisierung mit den Endpunkten der Visualisierung der temporalen Verbindungen genutzt, wie in Abschnitt 3.3.4 erläutert.

3.5. Kombination mit existierenden Konzepten

In diesem Abschnitt sollen Möglichkeiten zur Integration existierender Konzepte in das hier vorgestellte Konzept vorgestellt werden.

Das Konzept der dynamischen Anfragen [Shn94] könnte auf mehrere Arten verwendet werden. Zum einen zur direkten Anzeige der aktuellen Ergebnisse während der Manipulation der Anfrage, zum anderen könnte ein Auswahlmodus geschaffen werden, der die Selektion von Subjekt- oder Filterknoten beziehungsweise die Selektion von Kanten zulässt und die aktuelle Auswahl entsprechend hervorhebt und als neue Anfrage ausführt. Je nach Auswahl würden dann die Ergebnisse, die sich durch die Teilanfrage aus den ausgewählten Anfrageelementen ergeben, in der Ergebnisanzeige aller Ergebnisse hervorgehoben, oder die, die nicht dazu gehören, ausgeblendet. Beispielsweise würde die Auswahl einer aussagenlogischen Kante eine Anfrage unter Einbezug aller mit dieser Hyperkante verbundenen Knoten ausführen und die Ergebnisse dieser Anfrage unter den Ergebnissen der gesamten Anfrage in der Ergebnisanzeige suchen und hervorheben.

Die Idee der Visualisierung des Datenflusses zwischen verschiedenen Filtern mittels der Breite der Verbindung, also dass die Breite der eingehenden und ausgehenden Filterverbindung anzeigt, wie viele Daten vor und wie viele nach der Filterung durch den Filter noch in den Ergebnissen enthalten sind, wie sie im Filter/Flow-Konzept beschrieben wird [YS98], würde auch in diesem Konzept eine direkte optische Rückmeldung zu den Auswirkungen einer Filterkonfiguration bieten. Allerdings dürften die Breiten nicht zu groß gewählt werden, damit die Knoten, die ja mit der Anzahl und Breite der Kanten wachsen, nicht unnötig groß werden. Gerade zu der wachsenden Größe der Knoten wäre dies aber eine passende Ergänzung, da die Knotengröße so auch ausdrücken würde, ob ein Filter große oder kleine Auswirkungen auf die Größe der Ergebnismenge hat.

Eine für die temporalen Filter wichtige Erweiterung könnte die Definierbarkeit wiederkehrender Ereignisse sein, wie sie im *Time Automaton*-Konzept [CGB13] durch eine Multiplizitätsangabe auf einer Kante angegeben werden kann, die einen Zyklus verursacht. Die Idee, Schleifen mit einer Multiplizitätsangabe für Ereignisse zuzulassen, die angibt, wie oft das Ereignis auftreten muss, kann direkt übernommen werden. Komplexere Muster, wie sie im genannten Konzept möglich sind, könnten auch übernommen werden, indem die temporalen Literale von Ereignissen durch Verknüpfungen zu einem wie dort beschriebenen Automaten ersetzt werden. Unter anderem müsste dann aber auch eine Möglichkeit geschaffen werden, relative temporale Verknüpfungen von Ereignissen auf eine gewisse Anzahl wiederkehrender Ereignisse einschränkbar zu machen. Das heißt, gegeben ein Ereignis A , welches mehrfach auftreten soll, und ein Ereignis B sowie eine relative Ordnung A vor B , dass festlegbar sein müsste, ob A vor B nur für das Auftreten des ersten oder einer beliebigen Anzahl erster Auftreten von A gelten muss, oder ob die Beziehung von allen A erfüllt werden muss.

Diese Überlegung führt auch gleich zu der Integration eines allgemeineren Konzepts, der Möglichkeit der Quantifizierung beziehungsweise Multiplizitätsangabe für alle Filter, da eine Ressource in Graphdatenbanken mit mehreren verschiedenen Ressourcen über dieselbe Beziehung verknüpft sein kann. Sowohl Literale als auch Verknüpfungen könnten mit einer geeigneten Darstellung eines Existenz- oder Allquantors versehen werden, oder aber mit der Eingabe eines Zahlenbereichs,

3. Konzept

über den sich definieren lässt, wie viele Objekte mit dem angegebenen Literal beziehungsweise wie viele Verknüpfungen zu einer anderen Ressource gültige Ergebnisse wenigstens, genau oder höchstens für ein bestimmtes Prädikat besitzen müssen. Eine Darstellung einer solchen Multiplizitätsangabe kann zum Beispiel Kantenbeschriftungen in UML-Diagrammen entnommen werden [om210, S. 65–70], die untere und obere Schranke von zwei Punkten getrennt angeben.

4. Implementierung

4.1. Eingesetzte Technologien

4.1.1. Graphdatenbank und Anfragen

Aufgrund bereits vorhandener Ressourcen und der Verbreitung des *Semantic Web* [BLHL01] wurde ein *RDF-Triplestore* als Graphdatenbank genutzt. Mit *SPARQL* [PS08] existiert eine standardisierte Anfragesprache für diese Datenbanksysteme, so dass die Implementierung unabhängig von der tatsächlich eingesetzten SPARQL-fähigen Datenbank umgesetzt werden konnte. Der Prototyp wurde sowohl mit Datenquellen im Internet wie *DBPedia* [DAb], als auch lokal mit dem *OpenLink Virtuoso Server* [OS] getestet.

Da die vorhandenen Datenquellen im Internet während der Implementierungsphase des öfteren Ausfälle erlitten, fiel die Entscheidung, zur Durchführung der Nutzerstudie einen kleinen, für die Aufgaben der Studie relevanten Teildatensatz von *DBPedia* zu kopieren und mit einem lokalen *OpenLink Virtuoso Server* auf dem Testgerät bereitzustellen.

4.1.2. Programmiersprache, Grafik-Framework und Bibliotheken

Zur Erstellung des Prototyps wurde das *.NET*-Framework von Microsoft eingesetzt, mit der *Windows Presentation Foundation* als Grafik-Framework in Kombination mit der Programmiersprache *C#*.

Die Wahl eines objektorientierten Grafik-Frameworks bot sich an, da durch die Ähnlichkeit der verschiedenen Filterknoten und Kantentypen sowie identischer Interaktionskonzepte für diese klar war, dass viel doppelter Code durch Vererbung eingespart werden könnte. Abschnitt 4.2 erläutert diese Vererbungshierarchie genauer. Weiterhin sollte das Grafik-Framework gute Unterstützung für benutzerdefinierte Oberflächenelemente und selbstgezeichnete Grafiken bieten. Die *Windows Presentation Foundation* ist dafür gut geeignet, da sie eine einfache Erweiterung ihrer vorhandenen Oberflächenelemente, die schon Grundfunktionalitäten bereitstellen, erlaubt. Zum Beispiel wird die Ereignisbehandlung für Nutzereingaben vereinfacht, indem alles, was im eigenen Zeichencode gezeichnet wird, automatisch Nutzereingaben empfangen kann, ohne dass eine eigene Treffertest-Berechnung nötig wird.

Neben der Unterstützung der *Windows Presentation Foundation* sprach für die Nutzung des *.NET*-Frameworks auch das Vorhandensein geeigneter Bibliotheken und Erfahrung mit der *.NET*-Sprache

4. Implementierung

C#. SPARQL-Anfragen werden mit der institutseigenen Bibliothek *SparqlHelper* generiert und über die Open-Source Bibliothek *dotNetRDF* [VZA⁺] an einen SPARQL-Endpunkt weitergeleitet.

Nicht zuletzt sind für die Erstellung von Programmen mit dem .NET-Framework und der Windows Presentation Foundation komfortable Werkzeuge verfügbar, wie etwa ein User Interface Designer.

4.2. Klassenhierarchie

In Abbildung 4.1 wird ein Ausschnitt der Klassenhierarchie des Prototyps gezeigt, in dem die für das Konzept wichtigsten Klassen mit ihren Beziehungen enthalten sind.

Die abstrakte Klasse *AQueryItem* ist die Basis für alle Filterknoten, also für Fakten (*Fact*), Ereignisse (*Event*) und Subjektverknüpfungen (*ObjectInterconnectionAnchor*). Sie verwaltet Prädikat und Objekt eines Filters, repräsentiert durch *Predicate* und *Literal*. Weiterhin zeichnet sie die grundlegende Darstellung eines Filters, also einen Kreis für die aussagenlogischen Verknüpfungen und eine Zeichenkette für den Prädikatnamen sowie eine Zeichenkette für die Objektrepräsentation. Damit verbunden werden auch die Ereignisse wie Mausklick und -hover auf den Kreis und Prädikat- und Objekttext behandelt. Die *Event*-Klasse erweitert diese Darstellung um das Zeichnen des äußeren Kreises und die *ObjectInterconnectionAnchor*-Klasse blendet den Objekttext aus. Verbindungen zu einem aussagenlogischen Pfad werden durch die Zugehörigkeit zu einem oder mehreren *QueryPath*-Instanzen ermöglicht. *QueryPath* ist die Klasse, die eine Liste von konjunktiv verbundenen Filtern enthält, also die in Abschnitt 3.4.1 beschriebenen Hyperkanten bildet, und sich auch um Zeichnung der Kante und Interaktion mit ihr kümmert.

Predicate enthält den Anzeigenamen, der von *AQueryItem* gezeichnet wird. Der zugelassene Datentyp für das Prädikat, auf dessen Basis beim Bearbeiten eines Objekts dann ein entsprechender Editor angezeigt wird, ist auch dort definiert. Zur Generierung der SPARQL-Anfragen nötige Informationen wie *IRI* und Sprachtags sind auch enthalten.

Literal ist eine Oberklasse für die verschiedenen Wertetypen, die ein Literal annehmen kann, darunter Zeichenketten (*StringLiteral*), Zahlen (*NumberLiteral*) und Datumswerte (*DateLiteral*).

Geladen werden Subjektdefinitionen *QuerySubject* aus einer XML-Datei, wie in Abschnitt 4.3 beschrieben. Sie werden als Liste von *QuerySubject*-Instanzen gespeichert und enthalten einen *Copy*-Konstruktor für die weitere Verwendung im Programm, da ja auch mehrere gleiche Anfragesubjekte in einer Anfrage möglich sein sollen. *QuerySubject*-Objekte enthalten eine Liste erlaubter Prädikate und erlaubter Verbindungen zu anderen Subjekten, so dass in der Ausführung beim Hinzufügen eines Filters zu einem Subjektgraphen nur die Prädikate angezeigt werden, die auch für das zugehörige Subjekt zulässig sein sollen.

Zwischenverbindungen zwischen zwei Ereignissen, zwei Subjektverknüpfungsknoten oder zweier Fakten werden durch die Klassen *TimeConnector*, *TimeInterConnector*, *ObjectInterconnectionAnchor* und *FactInterconnector* modelliert, die Gemeinsamkeiten wie die Definition eines Beginn- und eines Zielfakts von der Schnittstelle *IConnector* erben. Die Schnittstelle *IInterconnector* erweitert *IConnector* dabei um die Angabe von Subjektgraph des Beginn- und Zielfakts und repräsentiert

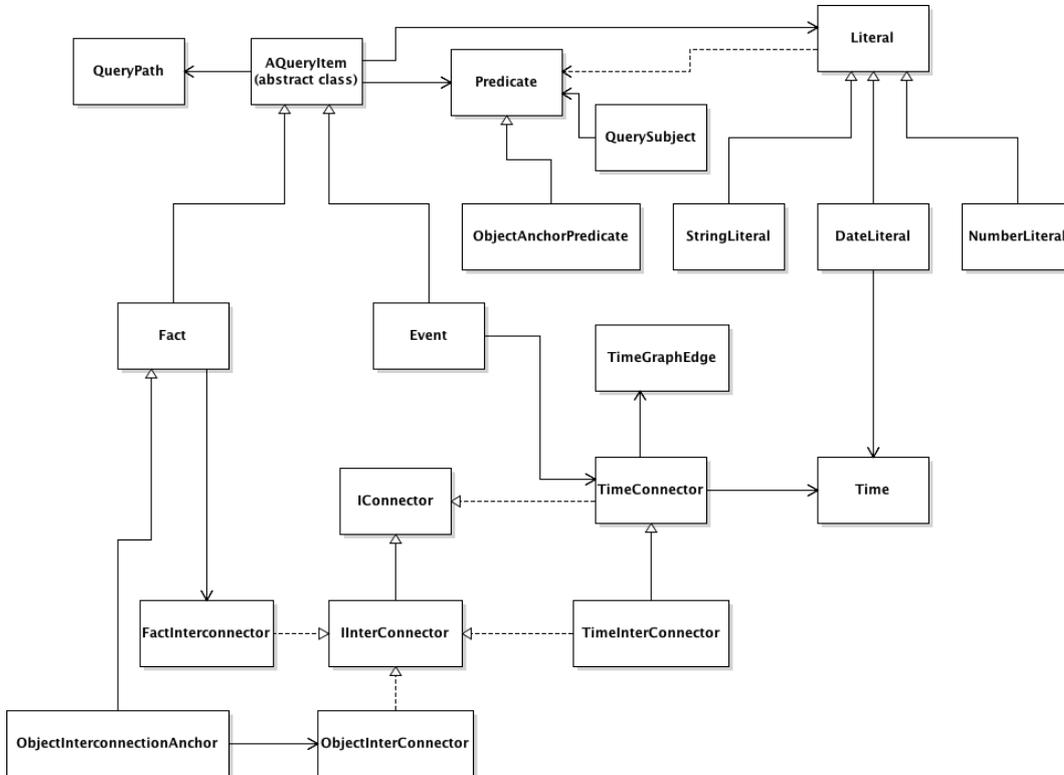


Abbildung 4.1.: Klassendiagramm für Anfragesubjekt, Prädikat und Objekte.

somit die Zwischenverbindungen zwischen verschiedenen Subjektgraphen, wie in Abschnitt 3.2 beschrieben.

Instanzen von *TimeGraphEdge* werden immer zusätzlich zu einer *TimeConnector*- oder *TimeInterConnector*-Instanz angelegt. Während letztere die visuelle Darstellung der Kanten übernehmen, wird aus den Instanzen von *TimeGraphEdge* ein gerichteter Graph aufgebaut, der zur Sortierung der Ereignisse dient. Er enthält nur Vorwärtskanten, also die temporalen Relationen *vor*, *vor-oder-gleichzeitig* und *gleichzeitig*. Vom Nutzer definierte temporale Beziehungen werden für diesen Sortiergraphen immer in Vorwärtskanten umgewandelt. Das Durchlaufen des Graphen ermöglicht so die relative Sortierung der Ereignisse, deren Funktionsweise in Abschnitt 4.4 noch genauer erläutert wird.

Absolute Zeitwerte und relative temporale Abstände sind Instanzen der Klasse *Time*, die verschiedene Granularitäten zulässt, Rechenmethoden für Addition und Subtraktion sowie eine Umwandlung in Zeichenketten bereitstellt.

4. Implementierung

dbpedia-owl:birthDate	▪ 1770-12-16 (xsd:date)
dbpedia-owl:birthPlace	▪ dbpedia:Bonn ▪ dbpedia:Electorate_of_Cologne
dbpedia-owl:birthYear	▪ 1770-01-01 (xsd:date)
dbpedia-owl:deathDate	▪ 1827-03-26 (xsd:date)
dbpedia-owl:deathPlace	▪ dbpedia:Vienna
dbpedia-owl:deathYear	▪ 1827-01-01 (xsd:date)
...	
dbpprop:dateOfBirth	▪ 1770-12-16 (xsd:date)
dbpprop:dateOfDeath	▪ 1827-03-26 (xsd:date)
...	
dbpprop:placeOfBirth	▪ Bonn, Electorate of Cologne
dbpprop:placeOfDeath	▪ Vienna, Austrian Empire

dbpprop:pnd	▪ 118508288 (xsd:integer)
dbpprop:pos	▪ left

(a) Mehrfach über unterschiedliche IRIs definierte Daten.

(b) IRIs, die mit unklaren Labels beschriftet sind.

Abbildung 4.2.: Auszüge des englischsprachigen DBPedia-Eintrags für Ludwig van Beethoven [DAa].

4.3. Verfügbare Filter

Die gesamte Ontologie einer Datenquelle automatisch für die Suche verfügbar zu machen, würde bei manchen Ontologien wie zum Beispiel der von DBPedia für Verwirrung der Nutzer sorgen, da viele identische Daten mehrfach über unterschiedliche IRIs bereitgestellt werden. Zu sehen ist dies anhand Abbildung 4.2a, in der derselben Ressource zum Beispiel das Geburtsdatum über zwei verschiedene IRIs, *dbpedia-owl:birthDate* und *dbpprop:dateOfBirth*, zugeteilt wird. Weiterhin liegen manche dieser doppelten Daten in unterschiedlichen Datentypen vor, was das Anfragesystem weiter erschweren würde. Manche der IRI-Labels sind auch derart beschriftet, dass nicht jedem Nutzer ersichtlich sein dürfte, was diese Label wirklich bedeuten. Für die in Abbildung 4.2b gezeigten IRIs *dbpprop:pnd* und *dbpprop:pos* etwa dürften auch deren Labels *PND* und *pos* für die meisten Nutzer ohne Nachschlagen in einer Dokumentation wohl unklar sein. Solche inkonsistenten Datenquellen erfordern die Möglichkeit, die dem Nutzer präsentierten Filtermöglichkeiten auf eine sinnvolle Auswahl einzuschränken.

Um dies zu erreichen, werden die verfügbaren Subjekte, Prädikate und deren zulässige Objekttypen in einer XML-Datei definiert. Das XSD-Schema zu dieser XML-Datei ist in Abbildung 4.3 zu sehen. Eine Instanz dieses Schemas enthält eine Liste von *QuerySubjectType*-Elementen, von denen jedes ein Anfragesubjekt mit Anzeigenamen für den Nutzer und *rdf:type*-URI repräsentiert.

Jedes *QuerySubjectType*-Element enthält außerdem eine Liste mit für dieses Subjekt zulässigen Prädikaten, die per Typ *PredicateBaseType* definiert werden. Das *type*-Attribut eines *Predicate*-Elements gibt an, ob das Prädikat und sein Objekt einen Fakten-, Ereignis- oder Subjektverknüpfungsfiler bilden. Die erlaubten Werte sind in der Enumeration *PredicateTypeEnum* angegeben. Das *id*-Attribut ist für die Implementierung der Prädikatsauswahl von Belang, eine durchlaufende Nummerierung, für jeden Filtertyp beim Wert Null beginnend, ist zwingend notwendig. Der Anzeigename *DisplayName* legt die Beschriftung des Prädikats für den Nutzer fest. Instanzen für Predicate-Elemente müssen vom Typ *PredicateType* oder *PredicateInterconnectionType* sein, da *PredicateBaseType* abstrakt ist.

PredicateType wird für Filter und Ereignisse eingesetzt und benötigt die Angabe der IRI für die SPARQL-Anfragen. *Type* definiert den Datentyp, den Objekte zu diesem Prädikat annehmen dürfen

und bestimmt damit auch, welches Popover dem Nutzer bei der Bearbeitung des Objekts angezeigt wird. *Lang* erlaubt schließlich noch die Definition der gewünschten Sprache für Ergebnisse zu Fakten oder Ereignissen.

Für Subjektverknüpfungen muss der *PredicateInterconnectionType* verwendet werden. Er erweitert *PredicateBaseType* um die Elemente *Source* und *Sink*, die jeweils Quell- und Zielsubjekttyp enthalten und über das Attribut *id* festlegen, welches Prädikat von Quelle und Ziel verbunden werden kann.

Durch die Definition von Subjekten, Prädikaten und deren erlaubten Objekttypen können unnötige oder ungewollte Filtermöglichkeiten vor dem Nutzer verborgen werden, was nicht nur die oben genannten Probleme löst, sondern auch erlaubt, spezialisierte Anfrageanwendungen innerhalb einer großen Ontologie bereitstellen zu können, ohne die Anwendung selbst modifizieren zu müssen. Falls doch die gesamte Ontologie einer Datenquelle verwendet werden soll, ist es aber leicht möglich, diese XML-Datei automatisiert aus der Ontologie zu generieren. Modifikationen am Programmcode sind aber nötig, falls Objekte von einem Datentyp sein dürfen, der von den unterstützten Grunddatentypen Zeichenkette, Zahl oder Datum abweicht. Ein solcher Datentyp kann durch Erweiterung der Klasse *Literal*, Implementierung des Interfaces *IQueryEditor* und Erweiterung des XSD-Schemas (siehe Abbildung 4.3) bereitgestellt werden.

4.4. Graphzeichnung

Der erste Schritt des Neuzeichnens einer Anfrage besteht in der Sortierung der Knoten. Faktenknoten und Subjektverknüpfungsknoten müssen nicht sortiert werden, da diese in einer Liste verwaltet werden und so automatisch nach Erstellungszeitpunkt sortiert sind. Jedes Vorkommen dieser beiden Filtertypen wird in für jedes Subjekt lokale horizontale Ebenen eingefügt. Bestimmt wird jede Ebene durch die Position eines Knotens innerhalb der nach Erstellungszeitpunkt sortierten Knoten einer Hyperkante.

Die globale Ordnung der Ereignisse durch einen Graphen wird mithilfe der in Abschnitt 4.2 beschriebenen Vorwärtskanten vom Typ *TimeGraphEdge* festgelegt. Der ersten Zeichenebene der Ereignisse werden alle Ereignisse zugeordnet, die weder über relative noch über absolute temporale Informationen verfügen. Anschließend werden die nächsten Ebenen gezeichnet. Dazu werden vorbereitend alle absoluten Ereignisse aufsteigend nach ihrem angegebenen absoluten Datumsliteral sortiert. Für das früheste beziehungsweise alle frühesten Ereignisse, falls innerhalb einer Anfrage mehrere mit gleichem Datumsliteral existieren, wird dann überprüft, ob Ereignisse existieren, diese mit relativen Kanten vor den absoluten frühesten Ereignissen angeordnet wurden. Existieren keine absoluten Ereignisse, werden anstelle dieser die Anfangsknoten des gerichteten Sortiergraphen verwendet. Die nächste Ebene ergibt sich dann durch die relativ verknüpften Nachfolgerknoten der Knoten der aktuellen Ebene. Falls solche nicht existieren, werden die nächsten frühesten absoluten Ereignisse eingefügt. Anschließend werden den restlichen Ereignissen weitere Ebenen nach dem selben Muster zugewiesen.

Nach der Sortierung sind die Maße jeder Ebene bekannt. Entsprechend können die Subjektknoten vertikal verteilt werden. Sie werden zu den zu ihnen gehörenden Filtern vertikal zentriert. Filter

4. Implementierung

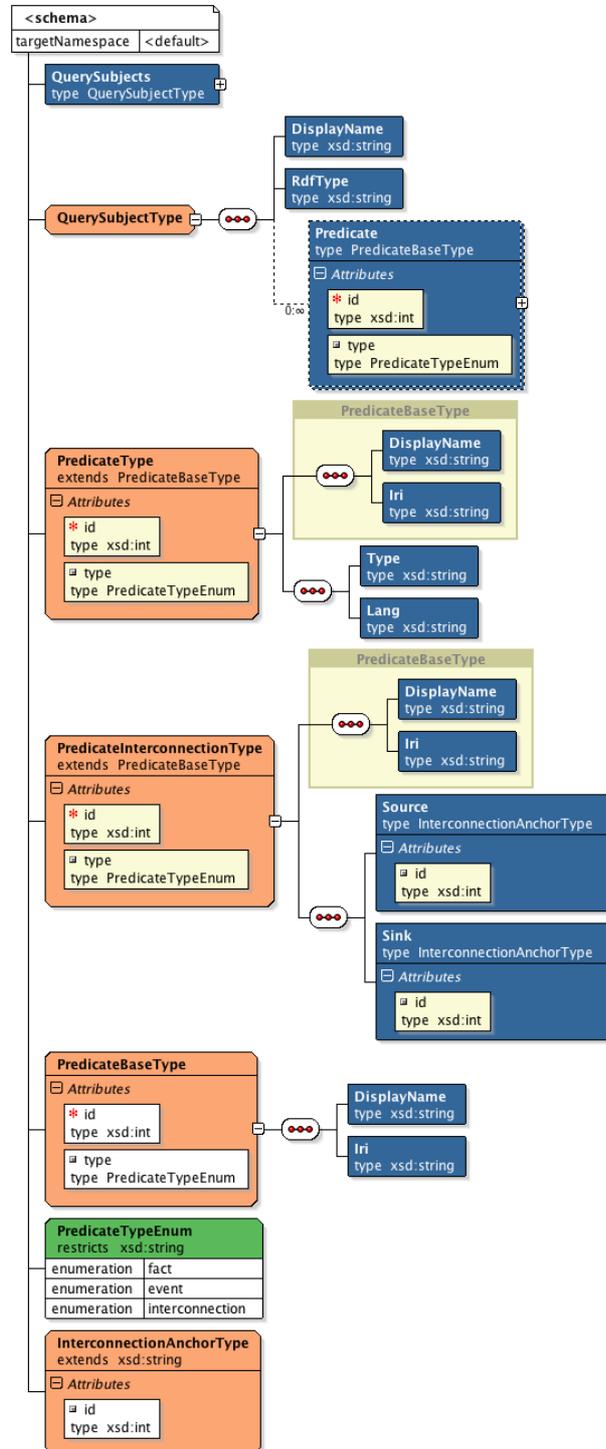


Abbildung 4.3.: XSD-Schema der Anfragesubjekt-Definitionsdatei.

werden dann baumartig nach ihrer aussagenlogischen Pfadzugehörigkeit verteilt, die horizontale Position ihrer lokalen oder globalen Ebene entsprechend.

Zum Abschluss werden die Hyperkanten durch die bereits positionierten Kanten gezeichnet.

4.5. Anfragegenerierung

Für jeden möglichen Pfad in den Anfragegraphen einer Anfrage wird eine eigene *SPARQL*-Anfrage generiert. Alle diese Anfragen werden asynchron ausgeführt, die Resultate zusammengeführt und dann nach Duplikaten gefiltert. Bei sehr großen Anfragen kann dies durch begrenzten Arbeitsspeicher auf der Client-Seite problematisch werden. Eine Erweiterungsmöglichkeit wäre hier der Einsatz einer lokalen Datenbank, in der die Ergebnisse der Teilanfragen zwischengespeichert und Duplikate effizient ausgefiltert werden könnten.

Diese Aufteilung in eine Anfrage pro Pfad war nötig, da in der Implementierungsphase getestete zusammenhängende Anfragen, vor allem solche, die mehrere *UNION*-Blöcke enthielten, oft zu Timeouts der verfügbaren Datenbank-Backends führten. Auch für die aufgespaltenen Anfragen war die Antwortzeit relativ hoch und ließ mit den getesteten Datenbanksystemen keine Umsetzung eines dynamischen Anfragesystems, dessen mögliche Verwendung in Abschnitt 3.5 diskutiert wird, zu.

Die Entscheidung fiel gegen ein dynamisches Anfragesystem, wie von Shneiderman beschrieben [Shn94], da in der Konzeptionsphase komplexere Anfragen an existierende graphbasierte Datenbanken mit mehreren Sekunden Antwortzeit entweder zu langsam für Echtzeitvisualisierung waren oder die Anfragen in den kompliziertesten Fällen sogar zu Timeouts führten.

Für jeden Subjektgraph werden alle Pfade durchlaufen und jeweils einer Pfadfolge hinzugefügt. Existieren auf einem Pfad Knoten mit Verbindungen zu anderen Subjektgraphen, so werden die Pfade, die mit dem Zielknoten der Verbindung verbunden sind, zur entsprechenden Pfadfolge hinzugefügt. Da so auch bis auf Reihenfolge identische Pfadfolgen mehrfach auftreten können, werden fertig erstellte Pfadfolgen auf Duplikate überprüft und diese entfernt.

Jede dieser Pfadfolgen ist anschließend eine Liste von Faktenknoten und Subjektknoten, die ihrerseits wiederum Kanteninformationen für Ereignis- und Subjektverknüpfungen innerhalb der Pfadfolge enthalten. Für jeden Subjektknoten einer Pfadfolge wird ein *rdfs:type*-Tripel mit einer Variable und dem Subjekttyp erzeugt. Zu allen zu diesem Subjekt gehörenden Knoten einer Pfadfolge werden dann Tripel mit der Variablen des Subjekts, der definierten *IRI* und dem angegebenen Wert bzw. Verweis auf die Variable eines anderen Filters generiert. Die Ereigniskanten werden dann als Filter mit den angegebenen Bedingungen erstellt und die Subjektverknüpfungskanten als Tripel mit den Variablen von Quell- und Zielsubjekt sowie deren Verknüpfungs-*IRI*.

4. Implementierung

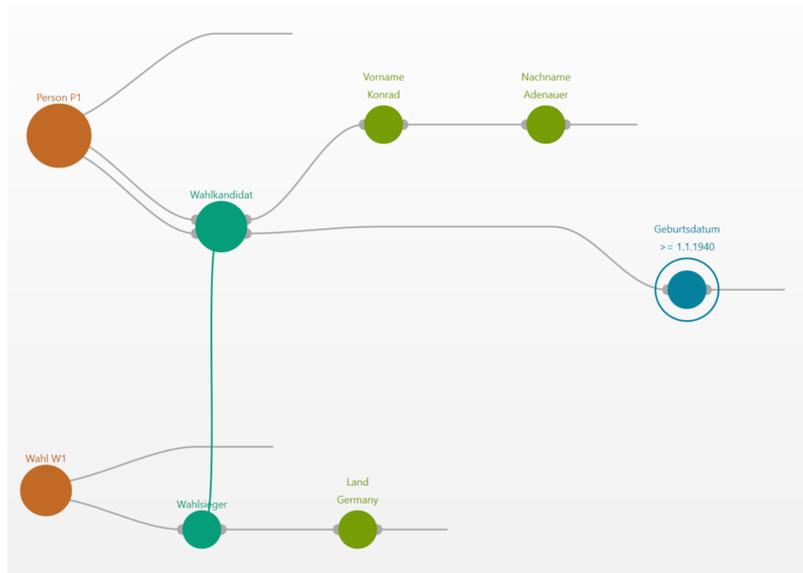


Abbildung 4.4.: Anfragebeispiel im Prototyp: Ergebnisse der Anfrage sind alle Personen, die Wahlsieger in einer deutschen Wahl waren und entweder nach dem 01.01.1940 geboren wurden oder mit Vorname *Konrad* und mit Nachname *Adenauer* heißen oder Personen, auf die beides zutrifft.

```
SELECT DISTINCT ?v6 ?v7 ?v8 ?v9 ?v10
WHERE {
  ?v1 a <http://dbpedia.org/ontology/Election> .
  ?v2 a <http://xmlns.com/foaf/0.1/Person> .
  ?v1 <http://dbpedia.org/property/afterElection> ?v2 .
  ?v2 <http://xmlns.com/foaf/0.1/givenName> ?v3 .
  ?v2 <http://xmlns.com/foaf/0.1/surname> ?v4 .
  ?v1 <http://dbpedia.org/property/country> ?v5 .
  ?v2 <http://xmlns.com/foaf/0.1/givenName> ?v6 .
  ?v2 <http://xmlns.com/foaf/0.1/surname> ?v7 .
  ?v2 <http://dbpedia.org/ontology/birthDate> ?v8 .
  ?v1 <http://dbpedia.org/property/afterParty> ?v9 .
  ?v1 <http://dbpedia.org/property/electionName> ?v10 .
  FILTER(regex(?v3, "Konrad", "") && regex(?v4, "Adenauer", "") && regex(?v5, "Germany", "")
    && langMatches(lang(?v6), "en") && langMatches(lang(?v7), "en") &&
    langMatches(lang(?v9), "en") && langMatches(lang(?v10), "en")).
}
```

Listing 4.1: Erste generierte SPARQL-Anfrage (4 Resultate).

4.5. Anfragegenerierung

P1.Vorname	P1.Nachname	P1.Geburtsdatum	W1.Siegerpartei	W1.Wahlname
Angela	Merkel	1954-07-17+02:00	CDU/CSU	German federal election, 2005
Angela	Merkel	1954-07-17+02:00	CDU/CSU	German federal election, 2009
Angela	Merkel	1954-07-17+02:00	CDU/CSU	German federal election, 2013
Gerhard	Schröder	1944-04-07+02:00	Social Democratic Party of Germany	German federal election, 2002
Gerhard	Schröder	1944-04-07+02:00	Social Democratic Party of Germany	German federal election, 1998
Konrad	Adenauer	1876-01-05+02:00	CDU/CSU	Election for the 2nd German Federal Diet
Konrad	Adenauer	1876-01-05+02:00	CDU/CSU	Election for the 4th German Federal Diet
Konrad	Adenauer	1876-01-05+02:00	CDU/CSU	Election for the 1st German Federal Diet
Konrad	Adenauer	1876-01-05+02:00	CDU/CSU	Election for the 3rd German Federal Diet

Abbildung 4.5.: Ergebnisse der SPARQL-Anfrage im Prototyp

```

SELECT DISTINCT ?v6 ?v7 ?v8 ?v9 ?v10
WHERE {
  ?v11 a <http://dbpedia.org/ontology/Election> .
  ?v12 a <http://xmlns.com/foaf/0.1/Person> .
  ?v11 <http://dbpedia.org/property/afterElection> ?v12 .
  ?v11 <http://dbpedia.org/property/country> ?v5 .
  ?v12 <http://dbpedia.org/ontology/birthDate> ?v13 .
  ?v12 <http://xmlns.com/foaf/0.1/givenName> ?v6 .
  ?v12 <http://xmlns.com/foaf/0.1/surname> ?v7 .
  ?v12 <http://dbpedia.org/ontology/birthDate> ?v8 .
  ?v11 <http://dbpedia.org/property/afterParty> ?v9 .
  ?v11 <http://dbpedia.org/property/electionName> ?v10 .
  FILTER(regex(?v5, "Germany", "") && ?v13 >=
    "1940-01-01"^^<http://www.w3.org/2001/XMLSchema#date> && LangMatches(lang(?v6),
    "en") && LangMatches(lang(?v7), "en") && LangMatches(lang(?v9), "en") &&
    LangMatches(lang(?v10), "en")).
}

```

Listing 4.2: Zweite generierte SPARQL-Anfrage (5 Resultate).

Zur Veranschaulichung sollen hier ein Anfragebeispiel im Prototypen, die generierten SPARQL-Anfragen sowie die Ergebnisse der Anfrage gezeigt werden. Abbildung 4.4 zeigt die Anfrage, deren Ergebnisse alle Personen sind, die eine deutsche Wahl in der Politik gewonnen haben und die entweder Konrad Adenauer heißen oder nach dem 01.01.1940 geboren sind. Die Anfrage enthält zwei Subjektgraphen, *Person P1* und *Wahl W1*. *P1* verfügt über zwei Pfade, *W1* über einen. Zur Anfrageerstellung sollen zuerst die beiden Pfade von *P1* betrachtet werden. Der oberste Pfad enthält die zwei Faktenknoten *Vorname* und *Nachname* sowie den Subjektknoten *Wahlkandidat* mit einer ausgehenden Kante zum Subjektgraph von *W1*. Deshalb wird auch der Pfad von *W1*, der das Ziel *Wahlsieger* des Subjektverknüpfungsknotens *Wahlkandidat* enthält, zur ersten Pfadfolge hinzugefügt. Für den zweiten Pfad von *P1* wird genauso verfahren. Schließlich wird noch der Pfad von *W1* betrachtet. Dieser könnte einmal mit dem oberen und einmal mit dem unteren Pfad von *P1* verknüpft werden, doch diese Pfadfolgen sind bereits vorhanden, nur in anderer Richtung,

4. Implementierung

weshalb sie verworfen werden. Dann werden alle Knoten einer Pfadfolge in SPARQL-Tripel bzw. -Filter umgewandelt.

Jede Pfadfolge wird als einzelne Anfrage betrachtet, weshalb in diesem Beispiel zwei Anfragen erstellt werden, die in den Listings 4.1 und 4.2 angegeben sind. Listing 4.1 korrespondiert mit dem oberen Pfad von *P1* und dem Pfad von *W1*, Listing 4.2 zeigt die SPARQL-Anfrage für den unteren Pfad von *P2* und den Pfad von *W1*. Die Ergebnisse dieser SPARQL-Anfrage mit *DBpedia* als Datenquelle sind in Abbildung 4.5 zu sehen. Dabei sind die ersten fünf Zeilen Ergebnisse der Anfrage aus Listing 4.1 und die letzten vier Zeilen Ergebnisse der Anfrage aus Listing 4.2.

5. Nutzerstudie

In der durchgeführten Studie sollte festgestellt werden, ob das vorgestellte Konzept gut und allgemein verständlich ist, d.h. ob es in kurzer Zeit und mit wenigen Fehlern interpretiert und selbst angewandt werden kann, auch von Personen unterschiedlichster fachlicher Herkunft. Der Fokus der Verständlichkeit lag auf Erkennbarkeit und Formulierbarkeit der temporalen Anfragenteile, also auf Ereignissen und deren Anordnung. Dies sollte für einfache Anfragen mit wenigen, nur absolut oder relativ angeordneten Ereignissen, bis hin zu komplexen Anfragen mit mehreren relativ und absolut angeordneten Ereignissen geprüft werden.

5.1. Vorbereitung und Durchführung

5.1.1. Teilnehmer

Die Studie wurde mit acht Teilnehmern im Alter von 22 bis 26 Jahren durchgeführt, davon fünf männlich und drei weiblich. Deren geringe Altersunterschiede sollten ähnliche kognitive und körperliche Voraussetzungen sicherstellen, damit Auswirkungen durch solche Unterschiede minimiert werden würden.

Alle Teilnehmer waren zum Zeitpunkt der Studie entweder eingeschriebene Studenten oder besaßen bereits einen universitären Abschluss. Eine Mischung verschiedener Studienfächer bei den Probanden war wünschenswert, um beurteilen zu können, ob das Konzept allgemein und unabhängig vom Fachbereich verständlich ist bzw. wie die jeweiligen Repräsentanten der Fachbereiche an die Lösung der Aufgaben herangehen. Die Studienfächer der Probanden umfassten unter anderem die Gebiete Betriebswirtschaftslehre, Chemie, Ernährungswissenschaften, Informatik und Maschinenbau. Für alle dieser Fächer existieren Szenarien, in denen eine temporale Filterung von Daten sinnvoll sein könnte (zeitlicher Verlauf von finanziellen Transaktionen, chemischen Reaktionen, Diäten, Netzwerkereignissen, Maschinenqualität, usw.).

Drei Studienteilnehmer hatten hilfreiche Kenntnisse aus der Aussagenlogik und bezüglich SQL-Datenbankanfragen. Zwei dieser drei hatten auch schon mit anderen visuellen Anfragetools gearbeitet. Einer dieser drei Probanden gab an, auch geringe Kenntnisse über das Semantic Web und SPARQL zu haben. Die anderen besaßen diesbezüglich keine Kenntnisse bzw. hatten noch nicht davon gehört. Alle Probanden waren Computerbesitzer, davon sechs Microsoft Windows-Nutzer, einer Mac OS X-Nutzer und einer Microsoft Windows- und Canonical Ubuntu-Nutzer. Sie gaben alle an, täglich Computer zu nutzen und erfüllten somit das für die Studienteilnahme erforderliche Kriterium der grundlegenden Computerbedienungskennntnisse, da ein Fehlen dieser auch die Resultate der Studie beeinflussen könnte. Hätte ein Teilnehmer zum Beispiel Probleme bei

5. Nutzerstudie

der Bedienung einer Computermaus gehabt, hätte sich dies negativ auf die benötigte Zeit für die Bearbeitung der Aufgaben ausgewirkt. Ebenso denkbar wäre eine Beeinflussung der empfundenen Schwierigkeit der Aufgaben.

5.1.2. Materialien und Durchführung

Aufgrund der begrenzten Anzahl an verfügbaren Testern und der kognitiv fordernden Natur der Aufgaben, die Verstehen und Problemlösen beinhalteten, wurde die Studie als *within-group*-Studie konzipiert, um potenzielle Unterschiede in den Fähigkeiten einzelner Probanden isolieren zu können [LFH10, S. 45–52]. Jeder Proband erhielt einen eigenen Termin für die Studiendurchführung, so dass Aktionen, Anmerkungen, benötigte Zeit und Probleme jedes Probanden zuverlässig festgehalten werden konnten.

Zu Beginn der Studie mussten alle Teilnehmer ein Interview ausfüllen, in welchem sie zu persönlichen Daten, Bildungsabschluss sowie Kenntnissen im Bereich Computernutzung und Suchanfragen befragt wurden (siehe Anhang A.1).

Anschließend sollten sie eine Anleitung des Programm-Prototypen lesen (siehe Anhang A.3), wofür die Teilnehmer ungefähr zehn Minuten benötigten. Beim Lesen der Anleitung auftauchende Fragen wurden direkt beantwortet. Es folgte eine ungefähr drei- bis fünf-minütige mündliche Einführung in das Programm, in der das Hinzufügen von Subjekten, Filtern und Verknüpfungen zwischen Filtern kurz exemplarisch demonstriert wurde.

Die Reihenfolge der Bearbeitung der darauffolgenden Aufgaben (siehe Anhang A.2) wurde zufällig in zwei verschiedene, gleich große Gruppen aufgeteilt um den Einfluss von eventuell auftretenden Lerneffekten einzelner Probanden auf die einzelnen Aufgabenteile auszugleichen. Eine Gruppe begann mit der Beantwortung der Verständnisaufgaben, die andere mit der Beantwortung der Kompositionsaufgaben. Die Studienteilnehmer wurden ermutigt, während der Bearbeitung der Aufgaben laut zu denken und auch ihre Eindrücke und Probleme zu äußern. Objektiv gemessen wurden unter anderem die benötigte Zeit per Aufgabe sowie Fehler, wobei den Probanden erklärt wurde, dass nicht ihre Leistung, sondern die Nutzbarkeit des Programms getestet würde, um sie nicht unter Druck zu setzen. Auffälligkeiten und Schwierigkeiten bei der Bedienung des Prototyps wurden auch aufgezeichnet.

Als Abschluss sollte von den Probanden ein Fragebogen ausgefüllt werden (siehe Anhang A.4), der sowohl geschlossene als auch offene Fragen bezüglich des Verständnisses und der Nutzbarkeit des Anfragekonzepts und besonders bezüglich dessen temporaler Aspekte enthielt.

Um Ausfällen der Datenbanken während der Studie vorzubeugen, wie sie während der Implementierung des Prototyps zum Beispiel mehrfach bei DBPedia auftraten, wurden relevante Daten in eine lokale Datenbank übertragen. Zur Bearbeitung der Aufgaben stand ein MacBook mit integrierter deutscher Tastatur und externer Maus zur Verfügung, auf dem der Prototyp in einer virtuellen Maschine mit Windows-Betriebssystem ausgeführt wurde. Das integrierte Retina-Display hatte eine Größe von 15 Zoll. Die verwendete Anleitung, Aufgaben, Hardware und Software wurde für keinen Probanden geändert, so dass für alle die gleichen Voraussetzungen galten.

Die gesamte Studie wurde mit jedem Probanden ohne Unterbrechung durchgeführt, da durch die kurz gehaltenen Teilaufgaben und einer geplanten Durchführungszeit von unter einer Stunde Pausen gegen Ermüdungserscheinungen nicht notwendig schienen.

5.1.3. Konzeption der Aufgaben

Für die Probanden wurden zwei verschiedene Aufgabentypen erstellt. Der erste Aufgabentyp enthielt Verständnisaufgaben für die Probanden, die aus Bildern von fertigen Anfragen bestanden (in Anhang A.2 als „Interpretationsaufgaben“ gekennzeichnet). Von den Studienteilnehmern sollte dann die Bedeutung der abgebildeten Anfragen aufgeschrieben werden. Mit diesem Aufgabentyp sollte vor allem die Lesbarkeit absoluter und relativ verknüpfter temporaler Filter getestet werden, aber im zweiten Aufgabenteil auch die Unterscheidung zwischen Konjunktion und Disjunktion.

Der zweite Aufgabentyp bestand aus Kompositionsaufgaben, in denen die gewünschten Einschränkungen für auszuführende Suchen angegeben wurden (in Anhang A.2 als „Textaufgaben“ gekennzeichnet). Diese sollten von den Probanden im Programmprototyp umgesetzt und ausgeführt werden. Beim Ausführen der Anfragen sollten die Probanden dann die Ergebnisse kontrollieren um festzustellen, ob ihre Anfragen korrekt gestellt waren. So sollte zum Beispiel eine Anfrage erstellt werden, die alle Personen mit dem Nachnamen *Mozart* zum Ergebnis hat, die weniger als 40 Jahre nach ihrer Geburt starben.

Die jeweils sechs Teilaufgaben der beiden Kompositionsaufgaben bauten aufeinander auf, so dass zwischen den Teilaufgaben die bisherigen Anfragen nicht gelöscht werden mussten, sondern erweitert werden konnten. Dies ermöglichte es, einzelne Aspekte des Konzepts gezielt auf Zeit, Fehler und subjektives Schwierigkeitsempfinden auszuwerten, indem die Änderungen von einem Aufgabenteil zum nächsten möglichst nur die auszuwertenden Konzeptaspekte beinhalteten. Auch der Schwierigkeitsgrad sollte zwischen den Aufgabenteilen steigen. Vor allem im zweiten Aufgabenteil wurden aber einige Konzeptpunkte in einer Teilaufgabe zusammengefasst, da die meisten schon in der ersten Aufgabe vorgestellt wurden und in der zweiten Aufgabe in Kombination getestet werden sollten, um in der Auswertung deren kombinierte Umsetzbarkeit mit weniger textueller Hilfe bewerten zu können.

Da der Schwerpunkt des Konzepts auf der temporalen Seite der Anfragen lag, bestanden die meisten Aufgaben aus Einfügen, Bearbeiten und Verknüpfen von Ereignissen. Weniger wichtige Konzeptpunkte wie Fakten und Subjektverknüpfungen wurden auch getestet und dienten bei mehrfachem Auftreten in den Aufgaben hauptsächlich der Sinngebung, damit keine zu abstrakten Anfragen gestellt werden mussten.

5.2. Ergebnisse

Für die Bewertung der Schwierigkeit der Konzeptkernpunkte und der bearbeiteten Aufgaben wurden von den Studienteilnehmern Punkte auf einer diskreten fünfstufigen Skala vergeben, wobei die Werte eins für *sehr schwierig*, zwei für *schwierig*, drei für *normale* Schwierigkeit, vier für *einfach* und fünf für *sehr einfach* standen.

5. Nutzerstudie

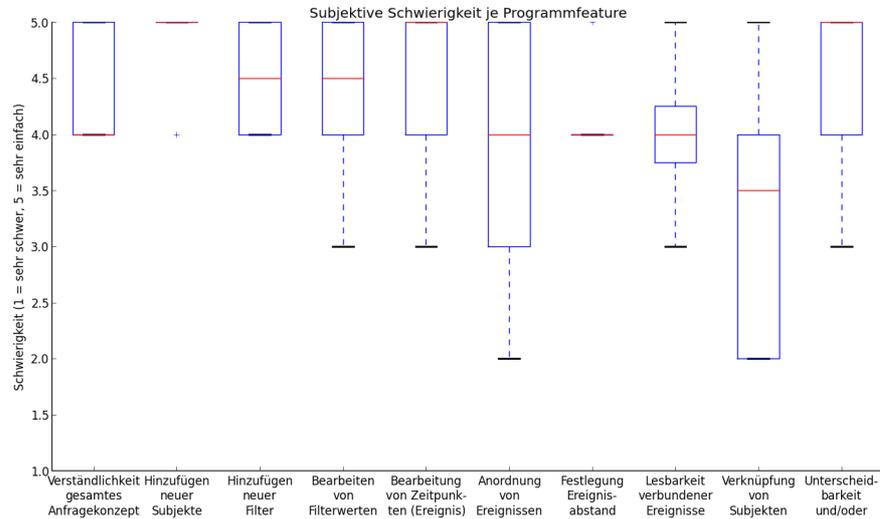


Abbildung 5.1.: Subjektive Schwierigkeit aller Probanden je Prototypfeature.

5.2.1. Konzeptbewertung

Abbildung 5.1 veranschaulicht die Bewertung der Verständlichkeit und Handhabung der Implementierung der Konzeptkernpunkte.

Für die Verständlichkeit des gesamten Anfragekonzepts, also die Unterteilung in Subjekte und verschiedene Filter, bestehend aus Prädikat und Objekt, sowie deren Zusammenwirken, gaben die Probanden ausschließlich Bewertungen mit *einfach* und *sehr einfach* an. Die Mehrzahl der Probanden gab als Schwierigkeit *einfach* an.

Das Hinzufügen neuer Subjekte wurde, bis auf einen Ausreißer in der Bewertung, als *sehr einfach* empfunden. Nur ein Proband gab an, dass es *einfach* sei. Neue Filter hinzufügen wurde auch nur als *einfach* und *sehr einfach* empfunden, jeweils genau zu 50%.

Das Bearbeiten von Filterwerten, also den Objekten, wurde mit einer Schwierigkeit von *normal* bis *sehr einfach* beurteilt. Der Median und die meisten Angaben liegen zwischen *einfach* und *sehr einfach*. Lediglich ein Studienteilnehmer gab dabei die Schwierigkeit *normal* an. Zeitpunkte der Ereignisse zu bearbeiten wurde ebenfalls mit Schwierigkeiten zwischen *normal* und *sehr einfach* bewertet. Hier gab aber die Mehrzahl der Probanden *sehr einfach* an. Dementsprechend liegt der Median bei *sehr einfach*. Die Wertung *normal* wurde auch hier wieder nur von einem Probanden gewählt.

Wertungen für die relative Anordnung von Ereignissen liegen weiter auseinander. Ein Proband befand dies für *schwierig*, zwei für *normal*, zwei für *einfach* und drei für *sehr einfach*. Der Median liegt aber immer noch bei *einfach*. Das Festlegen von Ereignisabständen der relativen Anordnungen wurde wieder nur als *einfach* und *sehr einfach* eingestuft, letzteres allerdings nur von zwei

Probanden. Bei der Lesbarkeit verbundener Ereignisse, also der Lesbarkeit deren Anordnung und Abstände, wurde der Median *einfach* erreicht. Meinungen reichen hier von normaler Schwierigkeit bis *sehr einfach*, wobei der Anteil der Bewertungen für *normal* und *sehr einfach* gleich groß ist.

Das einzige Ergebnis, bei dem der Median unter einfacher Schwierigkeit liegt, zwischen *normal* und *einfach*, ist die Verknüpfung von Subjekten. Damit ist sowohl Erstellung als auch Verbindung zweier Subjektverknüpfungsfiler gemeint. Die Wertungen verteilen sich zwischen *schwierig* und *sehr einfach*. *Schwierig* und *einfach* wurden jeweils von der gleichen Anzahl Probanden angegeben. *Normal* und *sehr einfach* wurden jeweils von einem Probanden gewählt.

Als Wertungen für die Unterscheidbarkeit zwischen konjunktiver und disjunktiver Verknüpfung wurden Stimmen von *normal* bis *sehr einfach* abgegeben. Der Median und alle Wertungen, mit Ausnahme dreier Probanden, liegen bei *sehr einfach*.

Als Bemerkungen in den offenen Fragen des Fragebogens gab es jeweils folgende Einträge:

Probleme beim Verständnis des Visualisierungskonzepts der Anfragen:

- „Nein, optisch gut dargestellt“
- „Bei Subjektverknüpfung Prädikat unklar“
- „Subjektverknüpfung“

Probleme mit der Bedienung des Programms

- „war eigentlich logisch, Zeitabstände müssen erst kurz überdacht und in das Denkschema des Systems gebracht werden“
- „links- u. rechts Maustaste vertauscht, Subjektverknüpfung“
- „War ein Macbook“

Vorteile/Nachteile in der Nutzung der Anwendung, Bevorzugung anderer Methoden/Werkzeuge

- „eher Vorteile → optisch gut nachzuvollziehen, andere Methoden bevorzugen: nein“
- „Andere: bei Anfragen ohne Filter (alle Daten o.E.)“
- „+ schnelle, unkomplizierte Aufklärung von Verwandtschaftsbeziehungen, zeitlichen Abfolgen, etc.“
- „+ zeitliche Ereignisse gut erkennbar, + übersichtliche Strukturierung“
- „Finde ich gut, weils Spaß macht“
- „- u.u. recht langsam bei komplexen Anfragen, + Angenehm für Touchscreen-Nutzung“
- „Nein, passt so“

Sonstige Anmerkungen

- „Bedienung hat Spaß gemacht“

5. Nutzerstudie

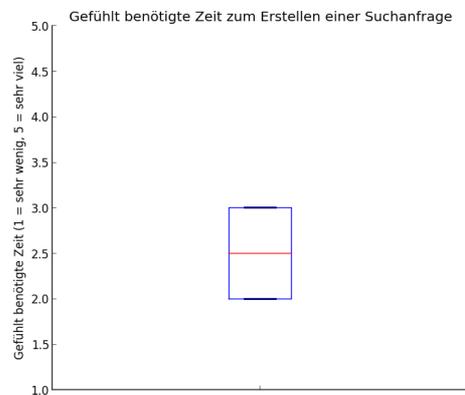


Abbildung 5.2.: Gefühlt benötigte Zeit für das Erstellen einer Anfrage.

- „+ farbliche Abhebung der Prädikate, + zeitliche Anordnung der Ereignisse und Beschriftung, + und/oder wie üblich (semantisch sinnvoll), + Verknüpfung von Subjekten (Möglichkeit als auch Umsetzung)“
- „am Anfang ist die Zeit der Suchanfrage länger, im weiteren Verlauf wird diese kürzer, Durchführung der Studie bzw. Programmbenutzung hat Spaß gemacht/sehr benutzerfreundlich“

Zusammenfassend heben diese Antworten Probleme im Zusammenhang mit den Subjektverknüpfungsfiltern und der Maustastenzuordnung, ein gut nachvollziehbares Konzept, übersichtliche Anordnung von Ereignissen sowie Spaß bei der Bedienung des Programmprototypen hervor.

Die für das Erstellen einer Anfrage gefühlt benötigte Zeit ist in Abbildung 5.2 ebenfalls auf einer Skala mit fünf Werten dargestellt, wobei die Werte eins für *sehr wenig*, zwei für *wenig*, drei für *normal*, vier für *viel* und fünf für *sehr viel* Zeit stehen.

5.2.2. Verständnisaufgaben

Die Bearbeitungszeit der Verständnisaufgaben wurde von Ansicht der Aufgabe bis zum Aufschreiben der Antwort gemessen. Wie in Abbildung 5.3 zu sehen ist, waren die Zeiten für die Bearbeitung der ersten Verständnisaufgabe, die nur eine Konjunktion zweier relativ angeordneter Ereignisse enthielt, sehr ähnlich mit einem Median von ungefähr 30 Sekunden. Die Bearbeitungszeiten der zweiten Aufgabe, die die erste Aufgabe um eine Disjunktion mit einem Ereignis zu einem festgelegten Zeitpunkt erweiterte, sind etwas unterschiedlicher mit einem Median von ungefähr 40 Sekunden. Ein paar Probanden benötigten auch mehr Zeit als andere, da sie die gesamte Antwort ausschrieben, während andere den Bezug zur ersten Aufgabe erkannten und dann nur eine kurze Referenz zur ersten Aufgabe angaben, gefolgt von den Unterschieden zu dieser.

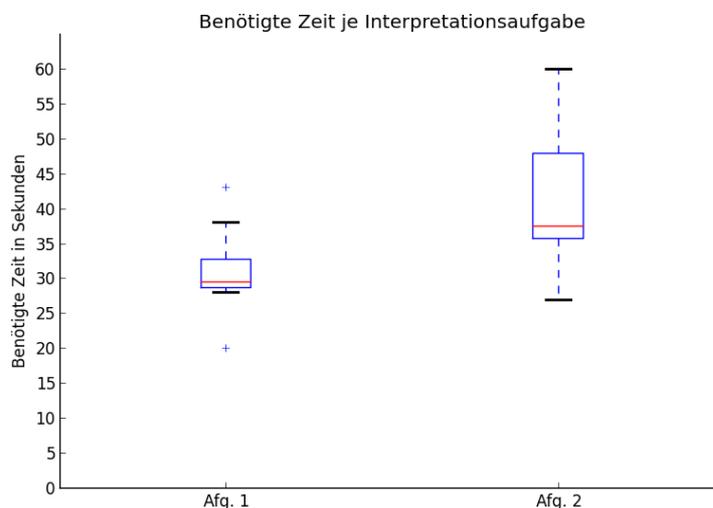


Abbildung 5.3.: Benötigte Zeit je Verständnisaufgabe.

Die subjektive Schwierigkeit der ersten Aufgabe wurde von allen Probanden bis auf zwei, die mit *einfach* bewerteten, als *sehr einfach* eingestuft. Der Median dieser gefühlten Schwierigkeit liegt bei *sehr einfach*. Für die zweite Aufgabe liegt der Median noch zwischen *einfach* und *sehr einfach*. Diese wurde aber auch von zwei Personen mit normaler Schwierigkeit bewertet. Zwei stimmten für *einfach* und vier für *sehr einfach*.

Bei den Verständnisaufgaben trat nur ein einziger Fehler auf, und zwar nicht bei der Anordnung der Ereignisse, sondern bei der Unterscheidung zwischen *und* sowie *oder*. Eine Probandin schrieb bei der zweiten Verständnisaufgabe statt *oder* ein *aber*, erkannte anstelle einer Disjunktion also eine weitere konjunktive Einschränkung, obwohl dieselbe Probandin mit der Bearbeitung der Kompositionsaufgaben begann und dort die Disjunktion aufgabe korrekt löste.

5.2.3. Kompositionsaufgaben

Die benötigte Zeit zum Lösen einer Kompositionsaufgabe wurde von der ersten Mausbewegung des Probanden nach dem ersten Durchlesen einer Aufgabe bis zum Klick auf die Schaltfläche *Anfrage ausführen* gemessen. Falls die Anfrage noch Fehler enthielt und dies den Probanden auffiel, wurde die Zeit weiter bis zur Behebung des Fehlers gemessen.

Der Median der gemessenen Zeiten bewegt sich je Aufgabe zwischen 27,5 Sekunden und 93 Sekunden, wie in Abbildung 5.5 gezeigt. Niedrigster Median wurde bei Aufgabe 1.2 festgestellt, höchster bei Aufgabe 2.2.

5. Nutzerstudie

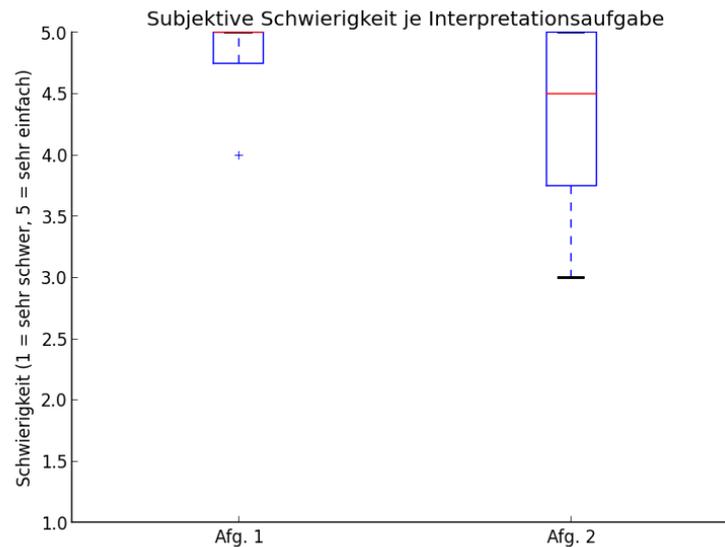


Abbildung 5.4.: Mittlere subjektive Schwierigkeit aller Probanden je Verständnisaufgabe.

Subjektive Schwierigkeiten für die Kompositionsaufgaben, in Abbildung 5.6 aufgetragen, reichen von *schwer* bis *sehr einfach*. Die Mediane der empfundenen Schwierigkeiten liegen im Bereich *normal* bis *sehr einfach*.

Bei der ersten Kompositionsaufgabe ging ein Proband davon aus, dass das Hinzufügen von Filtern auch automatisch Ergebnisspalten hinzufügen würde. Nachdem er deshalb zunächst keine auswählte und keine Ergebnisse angezeigt bekam, erinnerte er sich an die Einführung und bemerkte den Grund für das Fehlen von Ergebnissen. Er konnte seine Lösung dann durch Hinzufügen der entsprechenden Ergebnisspalten korrigieren.

Auch ein anderer Proband ging von einem Zusammenhang zwischen Filtern und Ergebnisspalten aus und fügte zusätzlich zur Ergebnisspalte *Geburtsdatum* in Kompositionsaufgabe 1.2 auch noch den gleichnamigen Filter hinzu, allerdings ohne Objekt, obwohl dies unnötig war. Dies fiel ihm aber nicht auf.

Eine Studienteilnehmerin wählte in Aufgabe 1.3 als Abstand zwischen Geburts- und Todesdatum zunächst die *gleich*-Relation anstelle der *kleiner-gleich*-Relation. Als sie beim Ausführen der Anfrage eine leere Ergebnismenge erhielt, begutachtete sie noch einmal ihre Anfragen, bemerkte den Fehler und korrigierte ihn anschließend. Bei derselben Aufgabe wählte ein Kandidat für die Anordnung von Geburts- und Todesdatum die *echt-kleiner*-Relation. Er löschte und korrigierte sie aber direkt und erklärte, die gewünschte Schaltfläche verfehlt zu haben.

Für die nächste Aufgabe, Aufgabe 1.4, ordnete eine Probandin das Todesdatum zwischen den zwei in der Anfrage enthaltenen Subjekten falsch herum an. Sie bemerkte diesen Fehler nicht. Nach

Beenden der Aufgabe wurde sie vom Studienleiter darauf hingewiesen, dass ihre Anordnung fehlerhaft sei. Ihrer Aussage nach hatte sie die Aufgabe falsch gelesen.

Aufgabe 1.5 und auch Aufgabe 1.6 wurden von allen Studienteilnehmern korrekt gelöst. Ein Proband erstellte zur Lösung von Aufgabe 1.6 zwar einen zweiten Ereignisfilter für das Todesdatum eines Subjekts, was aber ebenfalls ein legitimer Lösungsweg war.

In Aufgabe 2.1 fügten zwei Probanden zuerst fälschlicherweise ein neues Subjekt vom Typ *Person* anstelle vom Typ *Amtsperiode* ein. Einer dieser Probanden bemerkte dies direkt, ein anderer beim Auswählen des Prädikats eines Filters, den er diesem Subjekt hinzufügte. Von einer Studienteilnehmerin wurde ein für diese Aufgabe hinzugefügtes Ereignis versehentlich direkt wieder gelöscht. Diese Teilnehmerin hatte aber auch schon in der ersten Kompositionsaufgabe immer Probleme mit der Festlegung der Funktionen von linker und rechter Maustaste. Sie klickte zum Ändern eines Prädikats oder Objekts immer zuerst mit der rechten statt mit der linken Maustaste auf die zugehörigen Textfelder. Von ihr stammt auch, wie in Abschnitt 5.2.1 aufgeführt, die Anmerkung im offenen Teil des Fragebogens zur Vertauschung der linken und rechten Maustaste.

Beim Lösen der Aufgabe 2.4 bemerkte ein Proband, dass dieser Aufgabenteil auf zwei verschiedene Arten lösbar war, nämlich durch Anordnung und Festlegung eines Zeitabstandes sowie durch explizite Zeitangaben. Er löste ihn dann aber auch wie alle anderen Probanden durch Festlegung eines temporalen Abstandes. Eine Studienteilnehmerin wählte zunächst eine *echt-nach* anstelle einer *echt-vor*-Relation zwischen zwei Ereignissen aus, was ihr aber durch die automatische Umordnung der Ereignisse direkt auffiel und korrigiert wurde.

Aufgabe 1.5 wurde von allen Probanden korrekt gelöst. Allerdings gab ein Proband an, kurz überlegen zu müssen, wie die Einschränkung *spätestens* aus dem Aufgabentext im Programmprototyp umgesetzt werden könne.

In Aufgabe 2.6 wurde von einer Probandin zunächst ein Filter zum falschen Subjekt hinzugefügt. Sie bemerkte es jedoch beim Auswählen des Prädikats und verbesserte ihre Anfrage entsprechend. Alle Studienteilnehmer bis auf einen lösten die Disjunktion in dieser Aufgabe durch Hinzufügen eines neuen Pfades zum bestehenden Subjekt, einer jedoch erstellte und verknüpfte ein neues Subjekt, eine ebenso korrekte Lösung der Aufgabe.

5.3. Auswertung

Eine Anfrage in Subjekte, Prädikate und Objekte aufzuteilen scheint ein vielversprechendes Konzept zu sein, da die Verständlichkeit des gesamten Anfragekonzepts ausschließlich mit *einfach* und *sehr einfach* bewertet wurde. Ergänzt wird dies durch die Aussagen in den Textfeldern im Fragebogen, bei denen als einziges Problem des Visualisierungskonzepts mehrfach die Subjektverknüpfung auftaucht. Neben dieser Anmerkung gibt es aber überwiegend positive Aussagen, zum Beispiel dass es keine Probleme gab oder dass das Konzept übersichtliche Strukturierung und Anordnung böte. Objektiv spricht dafür, dass kein Proband bei der Unterscheidung zwischen Subjekt, Prädikat und Objekt Probleme zeigte oder Unklarheiten äußerte, weder in den Verständnisaufgaben noch in

5. Nutzerstudie

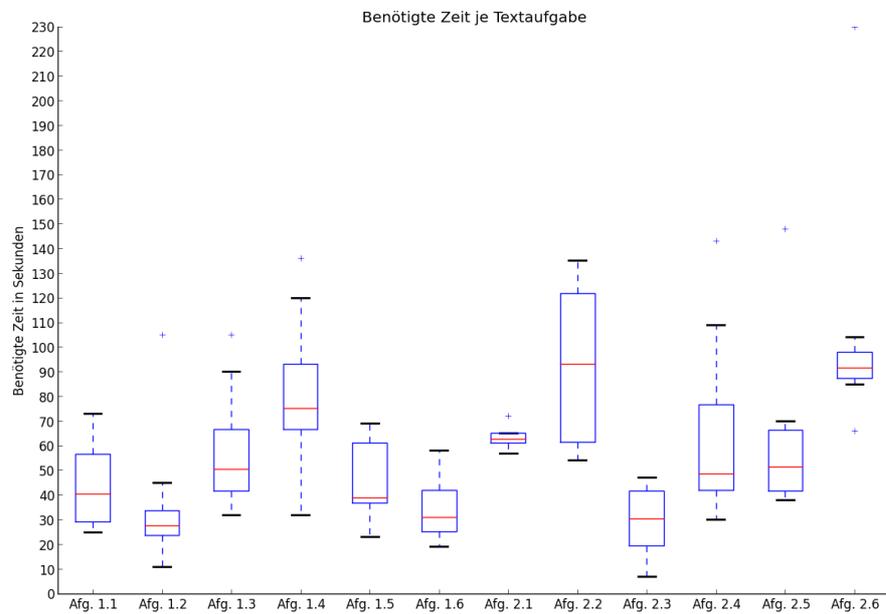


Abbildung 5.5.: Benötigte Zeit je Kompositionsaufgabe.

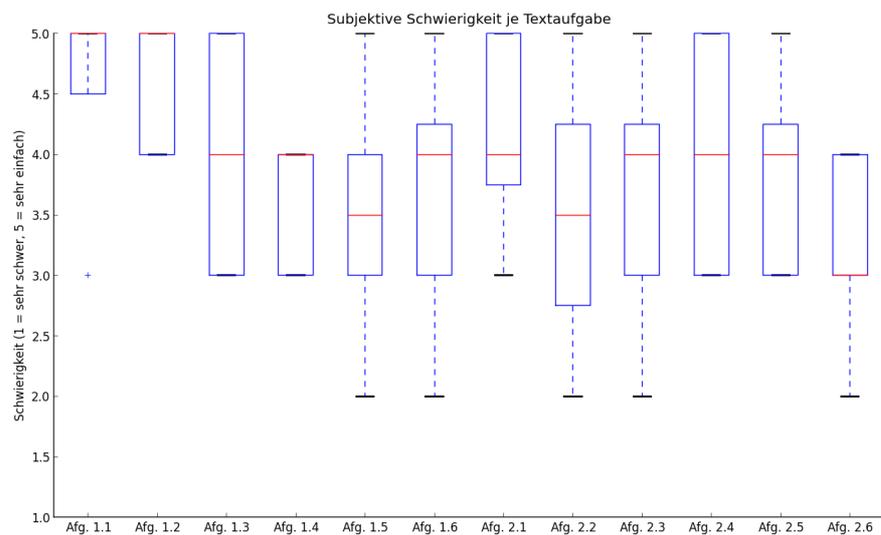


Abbildung 5.6.: Subjektive Schwierigkeit aller Probanden je Kompositionsaufgabe.

eigenen Anfragekonstruktionen für die Kompositionsaufgaben. Alle Studienteilnehmer wussten, wann sie ein Subjekt, wann ein Prädikat und wann ein Objekt benötigten und konnten alle Aufgaben lösen. Rückfragen der Probanden nach Lesen der Bedienungsanleitung für den Prototypen ließen ebenfalls erkennen, dass der Zweck dieser Anfragenaufteilung verstanden wurde und auch die entsprechenden Bezeichnungen wurden bei der Diskussion korrekt verwendet.

Die Lesbarkeit der relativen und absoluten Anordnung von Ereignissen scheint gut und schnell verständlich. Dies zeigt sich objektiv dadurch, dass für die Verständnisaufgaben insgesamt nur ein einziger Fehler gemacht wurde, aber nicht bei den temporalen Angaben, sondern durch das Nichterkennen einer Disjunktion. Außerdem wurde inklusive der benötigten Zeit zum Niederschreiben der Antworten ein Median von circa 30 und 40 Sekunden erzielt. Ein subjektiver Beleg findet sich in der Bewertung der Schwierigkeit der Lesbarkeit verbundener Ereignisse mit einem Median von *einfach*, und dadurch, dass keine Wertung mit *schwer* oder *sehr schwer* vorliegt. Ein weiterer zeigt sich in den subjektiven Schwierigkeiten der Verständnisaufgaben, die ja genau zum Testen der Lesbarkeit von angeordneten Ereignissen dienen sollten. Der Median dieser Schwierigkeiten für die erste Verständnisaufgabe liegt bei *sehr einfach* und für die zweite zwischen *einfach* und *sehr einfach*. Im Fragebogen wurde die Lesbarkeit verbundener Ereignisse explizit mit Wertungen zwischen *normal* und *sehr einfach* bewertet, mit Median bei *einfach*. Hinzu kommen die Aussagen der Probanden in den Textfeldern des Fragebogens, in denen keine Probleme bezüglich der Lesbarkeit der Ereignisanordnung geäußert wurden. Im Gegenteil, es wurde von einer schnellen, unkomplizierten Aufklärung von zeitlichen Abfolgen geschrieben und davon, dass die Anordnung und Beschriftung von Ereignissen gut erkennbar wäre. Außerdem konnten durch die automatische Anordnung der Ereignisfilter von den Probanden einige Fehler noch vor dem Ausführen der zugehörigen Anfrage erkannt und behoben werden und das Anordnen selbst wurde mit einem Median von *einfach* bewertet, war also nicht nur einfach zu verstehen, sondern auch einfach zu benutzen.

Im zweiten Verständnisteil wurde etwas mehr Zeit benötigt, da eine größere und komplexere Anfrage abgebildet war, die im Gegensatz zum ersten Teil nicht nur eine relative, sondern auch eine absolute temporale Relation beinhaltete und zusätzlich eine Disjunktion. Die größeren Abweichungen in den Zeiten für den zweiten Aufgabenteil ergaben sich unter anderem durch die unterschiedlichen Arten des Aufschreibens der Antwort, zum Beispiel referenzierten manche Probanden den ersten Aufgabenteil für einen Teil der Disjunktion, andere schrieben die Antwort aus dem ersten Aufgabenteil noch einmal nieder.

Subjektverknüpfungen, die in Aufgabe 2.2 getestet werden sollten, stellten für die meisten Probanden ein Problem dar. Zunächst dachte fast keiner daran, Filter für die Verknüpfung einzusetzen. Einige versuchten stattdessen, die Subjektknoten direkt per Drag'n'Drop zu verknüpfen. Die meisten inspizierten nach einigen Fehlversuchen dann auch das *Filter-Hinzufügen-Menü* und erinnerten sich beim Einsehen der Einträge an die schriftliche Anleitung und mündliche Einführung. Drei Probanden konsultierten nach ihren Versuchen das entsprechende Kapitel in der Anleitung. Trotz der Komplikationen konnte die Aufgabe von allen Studienteilnehmern korrekt gelöst werden. Die Notwendigkeit einer Anleitung sowie die im Vergleich zu den anderen Aufgaben relativ hohen Zeiten für das Hinzufügen und Verbinden zweier Filter fast aller Probanden sowie einige Schwierigkeitsbewertungen im Bereich *schwierig* bis *einfach* legen nahe, dass das Konzept in diesem Bereich noch verbessert werden sollte.

5. Nutzerstudie

Ebenso sollte die Erstellung eines Intervallabstands zwischen zwei Ereignissen vereinfacht werden. Während einfache Abstände zwischen Ereignissen leicht umgesetzt werden konnten, wurde für Aufgabe 2.6, die zwei temporale Kanten für die Definition eines Intervallabstands erforderlich machte, von allen Probanden nicht nur viel Zeit im Vergleich zu den anderen Aufgaben benötigt, sondern auch von den meisten Bewertungen als *normal*-schwierig eingestuft, von einer Probandin sogar als *schwierig*. Keine Bewertung liegt über *einfach*.

Das Hinzufügen neuer Subjekte und Prädikate wurde nur als *einfach* und *sehr einfach* empfunden, mit Medianen *sehr einfach* beziehungsweise *einfach*, ebenso das Bearbeiten von Objekten bis auf eine Einzelwertung *normal*. Obwohl diese Mechanismen nicht Kernbestandteil des Konzepts waren, scheinen sie sich also gut mit den temporalen Mechanismen und dem Anfragekonzept integrieren zu lassen. Auch die Erstellung von Disjunktion scheint leicht umsetzbar, da sie von allen Probanden auf Anhieb richtig erstellt wurde.

Eine Probandin schien gegen Ende der Studie Ermüdungserscheinungen zu zeigen: obwohl sie bei den ersten Kompositionsaufgaben noch zeitlich zu anderen Probanden vergleichbar blieb, benötigte sie ab Kompositionsaufgabe 2.3 die höchste Zeit aller Probanden für die Beantwortung von Kompositionsaufgaben. Für die letzten beiden Kompositionsaufgaben, 2.5 und 2.6, benötigte sie sogar mehr als doppelt so lange wie jeder andere Kandidat. In Kompositionsaufgaben 2.6 fügte sie außerdem zunächst auch noch ein falsches Subjekt hinzu und beurteilte diesen Aufgabenteil als einzige als *schwierig*.

Zur allgemeinen Zufriedenheit mit dem Konzept lässt sich anführen, dass in den offenen Fragen des Fragebogens außer der Geschwindigkeit des Prototypen bei komplexen Anfragen und Problemen bezüglich der Subjektverknüpfungsfiler nur positive Meldungen gelistet wurden. Drei Antworten enthalten sogar die Aussage, dass die Bedienung des Prototypen Spaß machte. Dass die Erstellung von Anfragen nicht als zu langwierig betrachtet wurde, wird auch durch die Bewertungen für die gefühlte benötigte Zeit für das Erstellen einer Anfrage belegt, die nur Aussagen von *normal* bis *schnell* erhielt und deren Median genau zwischen diesen beiden Antworten liegt.

6. Zusammenfassung und Ausblick

In dieser Bachelorarbeit wurde ein neues Konzept zur visuellen temporalen Filterung für graphbasierte Daten vorgestellt. Es erlaubt neben relativer und absoluter Ordnung temporaler Filter auch deren aussagenlogische Verknüpfung und ebenso die Einbindung weiterer Filtertypen, den Faktenfiltern und den Subjektverknüpfungsfilttern. Das Konzept wurde in einer prototypischen Implementierung umgesetzt, die Filter auf SPARQL-Konstrukte abbilden und die entstehenden SPARQL-Anfragen an eine SPARQL-fähige Datenquelle senden kann. Es wurde ferner ein Mechanismus zur Aufspaltung komplexer Anfragen in Teilanfragen integriert, der Berechnungstimeouts internetbasierter Datenquellen vermeidet. Zur Evaluation des Konzepts wurde der Prototyp in einer Nutzerstudie durch acht Probanden getestet. Wie die Auswertung der Nutzerstudie ergab, war das Konzept gut verständlich, die temporalen Filteraspekte des Konzepts gut lesbar und alle Aufgaben der Nutzerstudie mit wenigen Fehlern seitens der Probanden durch eine Anleitung und kurze mündliche Einführung in den Prototypen lösbar. Weder ein Konzeptpunkt noch die Lösung einer Aufgabe wurde von den Probanden als sehr schwierig empfunden und mehrere Probanden gaben an, bei der Bedienung des Prototypen Spaß empfunden zu haben. Andere Bereiche des Konzepts, etwa die Subjektverknüpfungsfiltter oder die Definition von Intervallabständen zwischen Ereignissen, sollten dem Evaluationsergebnis nach noch verbessert werden.

Ausblick

In diesem Abschnitt werden noch zukünftige Erweiterungs- sowie Verbesserungsmöglichkeiten des Konzepts und des Prototypen diskutiert.

Wie bereits festgestellt, wäre eine Änderung der Subjektverknüpfungsfiltter ratsam. Neben der in Kapitel 3.5 diskutierten möglichen Integration vorhandener Konzepte wäre auch eine Erweiterung der Abstandsdefinition der temporalen Kanten praktisch, denn im aktuellen Stadium sieht das Konzept nur die Einschränkung eines temporalen Abstandes mit einem Datumswert vor. Wird als Abstand zwischen zwei Ereignissen ein Intervall mit Mindest- und maximaler Dauer gewünscht, müssen zwei Kanten mit jeweils einer Abstandsangabe erstellt werden. Hier würde sich anbieten, das Konzept so zu erweitern, dass auch ein Intervallabstand zwischen zwei Ereignissen pro temporaler Kante definiert werden kann.

Erweitertes Feedback für den Nutzer, etwa das Hervorheben inkonsistenter temporaler Pfade, wie sie bei der Anordnung im aktuellen Prototypen noch möglich sind, wären zur Vermeidung fehlerbehafteter Anfragen hilfreich. Oder sogar ein Mechanismus, der das Hinzufügen temporaler Kanten oder die Änderung von Datumswerten, die solche Inkonsistenzen erzeugen würden,

6. Zusammenfassung und Ausblick

verhindern und eine entsprechende Warnmeldung anzeigen könnte, beziehungsweise auch gleich den Anfragenteil, in dem durch die Änderung Konflikte entstehen würden, hervorheben könnte.

Ebenfalls eine mögliche Erweiterung wäre eine dynamischere horizontale Positionierung der Ereignisse, deren räumliche Abstände momentan noch statisch sind. Die räumlichen Abstände könnten dabei aus den zeitlichen Abständen abgeleitet werden, so dass durch die Positionierung erkennbar ist, ob zwei Ereignisse temporal nahe beieinander oder weit auseinander liegen.

Da sich während der Evaluation des Prototypen zwei Probanden kurz erstaunt zeigten, dass Ereignisse automatisch umsortiert werden, könnte für die bessere Nachvollziehbarkeit und um das mentale Bild der Anfrage eines Nutzers nicht zu zerstören, eine Animation für den Übergang des Graphen von einer Repräsentation in eine geänderte hilfreich sein.

Für den Prototypen wäre noch eine Erweiterung des XML-Schemas für die XML-Dateien, die die verfügbaren Subjekte, Prädikate und Objekttypen für die Oberfläche bereitstellen, nützlich. Auch die Option für die automatische Erzeugung einer solchen Konfigurationsdatei bei Auswahl einer unbekanntes Datenquelle wäre denkbar. So könnten neue Datenquellen direkt genutzt werden. Eventuell nötige Anpassungen zur Spezialisierung oder zur Übersichtlichkeitssteigerung könnten dann anschließend noch in der generierten Datei vorgenommen werden.

Auch eine angepasste Version von Konzept und Prototyp für Mobilgeräte wäre denkbar, da diese heutzutage bei vielen Privatanwendern wie auch Unternehmen Verwendung finden. Für diese würden sich als Konzepterweiterungen unter anderem erweiterte Navigationsmöglichkeiten, wie zum Beispiel eine Übersichtskarte, eine kompaktere Darstellung, zum Beispiel mit der Option Subjektgraphen auf- und zuklappen zu können, anbieten.

Ein Nutzer schlug im offenen Teil des Fragebogens vor, dass der Prototyp beziehungsweise das Konzept angenehm für Touchscreen-Nutzung wäre. Sowohl für Desktop-Computer als auch Mobilgeräte wäre interessant, welche Verbesserungen in der Nutzung des Prototyps durch Touchscreen-optimierte Interaktion entstehen könnten.

A. Materialien der Nutzerstudie

A.1. Interview mit Probanden

Persönliche Angaben

1. Alter
2. Geschlecht
3. Beruf:
4. Höchster bisheriger Abschluss:
5. Muttersprache(n):
6. Sind Sie Computerbesitzer(in)? Falls ja, welches Betriebssystem nutzen Sie?
.....
7. Wie häufig benutzen Sie einen Computer? (nie/täglich/wöchentlich/
monatlich/...
.....
8. Wie Stufen Sie Ihre Kenntnisse über das Semantic Web und SPARQL ein?
(keine/wenig/durchschnittlich/Experte?)
.....
9. Haben Sie schon mit anderen visuellen Anfragetools gearbeitet? Falls ja,
welche?
.....

A.2. Aufgaben für Probanden

Aufgaben

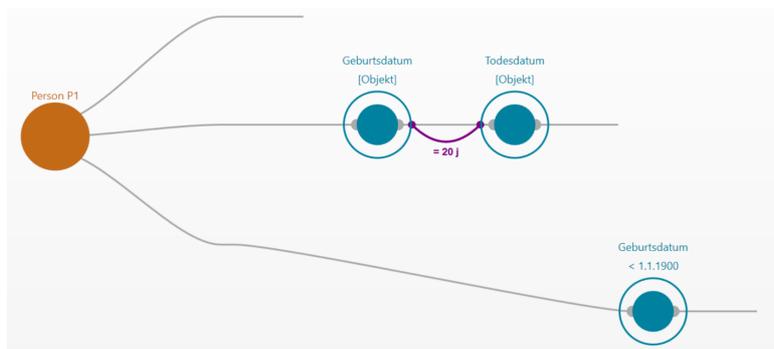
Interpretation

Geben Sie die Aussagen der folgenden Anfragen in eigenen Worten wieder. Beschreiben Sie die zeitlichen Beziehungen zwischen den Ereignissen bitte präzise.

Bild A)



Bild B)



Textaufgaben

Löschen Sie in den folgenden Aufgabenblöcken bitte keine Anfragen, da die Teilaufgaben aufeinander aufbauen.
Führen Sie nach jeder Teilaufgabe bitte die Anfrage aus und kontrollieren Sie die Ergebnisse.

1. Mozart's Familie

1. Finden Sie Vornamen und Nachnamen aller Personen (ohne Einschränkungen) im Datenbestand.
2. Finden Sie alle Personen, die mit Nachnamen ‚Mozart‘ heißen.
Lassen Sie sich die Ergebnisse mit den Ergebnisspalten Vorname, Nachname und Geburtsdatum anzeigen.
3. Grenzen Sie die Anfrage weiter ein: lassen Sie sich nur noch Personen mit dem Nachnamen ‚Mozart‘ anzeigen, bei denen der Tod weniger als 40 Jahre nach der Geburt eintrat.
Welches Ergebnis bleibt übrig?
4. Suchen Sie mögliche Vorfahren von Wolfgang Amadeus Mozart: Fügen Sie dazu ein weiteres Subjekt des Typs Person mit Filter für Nachnamen ‚Mozart‘ hinzu. Stellen Sie sicher, dass diese vor Wolfgang Amadeus geboren wurde und wählen Sie als Ergebnisspalten für das neue Subjekt Vorname, Nachname, Geburtsdatum und Todesdatum aus.
5. Erweitern Sie die Anfrage so, dass nur noch Personen mit dem Namen Mozart angezeigt werden, die Wolfgang Amadeus Mozart gekannt haben könnte, indem Sie die Einschränkung einbauen, dass die anderen Personen erst nach seiner Geburt starben
6. Erweitern Sie die Anfrage so, dass für die Mozarts, die nach Wolfgang Amadeus Mozart's Geburt starben nur noch Personen angezeigt werden, deren Sterbedatum noch vor Beginn des nächsten Jahrhunderts, also vor dem 01.01.1800, eintrat.

Löschen Sie nun bitte Ihre gesamte Anfrage.

2. Politik

1. Finden Sie alle Amtsperioden, deren Amt ‚Bundeskanzler‘ entspricht und die am oder nach dem 01.01.1949 begannen. Legen Sie als Ergebnisspalten Amt und Amtsbeginn fest.
2. Finden Sie die Personen, die mit diesen Amtsperioden verknüpft sind. Wählen Sie als Ergebnisspalten der Amtsinhaber-Person Vorname, Name und Geburtsdatum.
3. Wie Sie bemerken können, werden auch noch Vize-Bundeskanzler angezeigt. Wie können Sie diese durch bearbeiten des Amts-Filters der Amtsperioden verhindern?
4. Erweitern Sie die Anfrage so, dass nur noch Ergebnisse angezeigt werden, bei denen das Amtende mindestens 5 Jahre nach Amtsbeginn eintrat.
5. ... und bei denen die Amtsinhaber spätestens 5 Jahre nach Amtende starben.
6. Erweitern Sie die Anfrage jetzt so, dass alle Ergebnisse angezeigt werden, die die bisherigen Filter erfüllen oder Amtsperioden, die länger als 10 Jahre aber kürzer als 22 Jahre dauerten.

A.3. Anleitung für Probanden

Programmbedienung

A. Programmoberfläche

Abbildung 1 gibt einen Überblick über die Programmoberfläche:

1. **Subjekt hinzufügen:** Mit dieser Schaltfläche können neue Subjekte zur Anfrage hinzugefügt werden
2. **Alles löschen:** Mit dieser Schaltfläche wird die gesamte Anfrage im Anfragefenster gelöscht
3. **Anfrage ausführen:** Mit dieser Schaltfläche werden die Ergebnisse der Anfrage angezeigt
4. **Anfragefenster:** Bereich zum Bearbeiten der aktuellen Anfrage

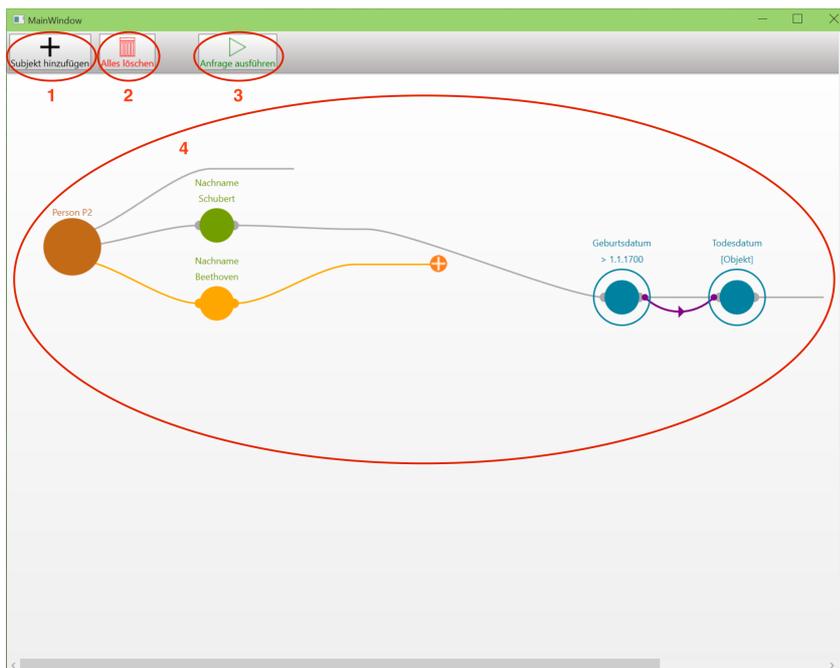


Abb. 1: Programmfenster

B. Anfragesubjekte

Eine Anfrage besteht aus **Subjekt(en)**, Prädikat(en) und Objekt(en).

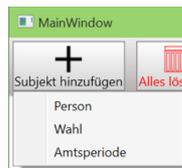
Für eine Anfrage werden ein oder mehrere **Subjekte** benötigt. Sie repräsentieren Dinge/Wesen der realen Welt und bündeln deren Eigenschaften (*Prädikate*).

Beispiel: Das Subjekt Person besitzt die Prädikate *Name, Vorname, Geburtsdatum* usw.

Beispiel: Das Subjekt Amtsperiode besitzt die Prädikate *Kandidat, Amt, Amtsantritt* usw.

Um ein neues **Subjekt** zu einer Anfrage hinzuzufügen, klicken Sie auf die Schaltfläche ‚Subjekt hinzufügen‘ (In Abbildung 1 mit der Nummer 1 gekennzeichnet).

Daraufhin erscheint ein Auswahlménü, in dem der gewünschte Subjekttyp ausgewählt werden kann:



Nach Auswahl eines Subjekttyps erscheint anschließend ein neues, nummeriertes **Subjekt** im Anfragebereich des Programms (In Abbildung 1 mit der Nummer 4 gekennzeichnet):



Wenn die Anfrage jetzt ausgeführt werden würde, würden in den Ergebnissen *alle* Personen des Datenbestands angezeigt, da die Person noch keine einschränkenden Filter (*Prädikate* und *Objekte*) besitzt.

Um die angezeigten Ergebnisse zu verfeinern, müssen zu einem **Subjekt** einschränkende Filter hinzugefügt werden.

C. Filter

Filter schränken die Suchergebnisse eines **Subjekts** ein. Ein Filter besteht aus *Prädikat* und *Objekt*. Dabei bezeichnet das Prädikat die Eigenschaft des Subjekts, die eingeschränkt werden soll, und das Objekt den Wert, den die Eigenschaft in den Ergebnissen besitzen muss.

Beispiel: Die Ergebnisse für ein **Subjekt** vom Typ Person sollen so eingeschränkt werden, dass nur noch Personen mit dem Nachnamen ‚Schubert‘ angezeigt werden:
Prädikat: Nachname, Objekt für Nachname: ‚Schubert‘.

Es gibt drei verschiedene Arten von Filtern:

1. **Fakten:** Ein **Fakt** erlaubt die Filterung nach *Wörtern und Zahlen*, z.B. ist der Vorname einer Person ein Fakt



2. **Ereignisse:** Ein **Ereignis** erlaubt die Filterung nach *zeitbasierten Daten*, z.B. ist das Geburtsdatum einer Person ein Ereignis



3. **Subjektverknüpfung:** Eine **Subjektverknüpfung** erlaubt die *Verknüpfung zweier Subjekte*, z.B. kann eine Person mit einer Amtsperiode über die Relation ‚Amtsinhaber‘ verknüpft werden. Das bedeutet, dass die Amtsinhaber des Amtes auch die Filter der Person erfüllen müssen bzw. dass die Personen das vorgegebene Amt mit den definierten Filtern besitzen.



Um einen Filter für ein **Subjekt** zu definieren, muss mit der Maus über einen vorhandenen Pfad gefahren werden, der mit dem gewünschten **Subjekt** verbunden ist. Dabei erscheint dann eine Schaltfläche ‚+‘ am rechten Ende des Pfades:



Das Klicken auf diese Schaltfläche öffnet ein Auswahlménü, das die Art des gewünschten Filters wählen lässt:



Nach dem Klick auf den gewünschten Filtertyp wird dieser dann zum orange markierten Pfad hinzugefügt:



Um Prädikat und Objekt (Wert) eines Filters zu bearbeiten, muss auf den entsprechenden Text über dem Kreis des Filters geklickt werden, was ein Bearbeitungsmenü öffnet:

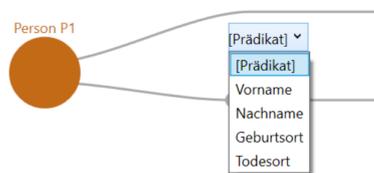


Abb.: Prädikatswahl (Klick auf [Prädikat])



Abb.: Filter (Klick auf [Filter])

Pfade

Alle Filter, die auf dem selben grauen Pfad liegen, sind mit dem verbundenen **Subjekt** durch ein **und** verknüpft, d.h. alle Ergebnisse zu diesem **Subjekt** müssen **alle Filter, die auf diesem Pfad liegen, erfüllen**.

Beispiel: Hier ist eine **und**-Verknüpfung einer Person und zweier Fakten abgebildet. Alle diese liegen auf einem gemeinsamen grauen Pfad, d.h. die Ergebnisse der verbundenen Person müssen sowohl den Vornamen ‚Franz‘ als auch den Nachnamen Schubert besitzen:

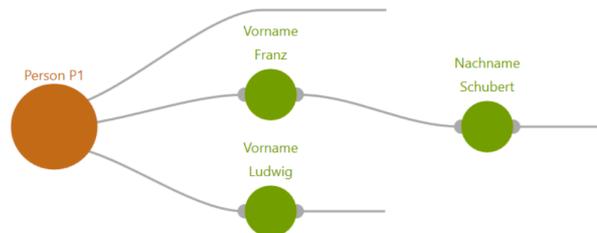


Mehrere graue Pfade an einem gemeinsamen **Subjekt** verhalten sich wie ein **oder**, d.h. die Ergebnisse müssen **alle Filter mindestens eines Pfades erfüllen**, nicht aber notwendigerweise die aller Pfade.

Beispiel: Hier ist eine **oder**-Verknüpfung einer Person und zweier Pfade abgebildet. Ergebnisse müssen entweder alle Filter des ersten **oder** alle Filter des zweiten **oder** alle Filter beider Pfade erfüllen.

D.h. Ergebnisse müssen entweder

- den Vornamen Franz und den Nachnamen Schubert
- oder den Vornamen Ludwig
- oder zwei Vornamen Franz und Ludwig und den Nachnamen Schubert besitzen:



Ein Filter kann auch mit mehreren grauen Pfaden verbunden werden, um den gleichen Filter nicht mehrmals erstellen zu müssen. Dazu muss einfach der gewünschte Pfad, mit dem der Filter verbunden werden soll, mit gedrückter Maustaste auf den gewünschten Filter gezogen werden. Nach dem Loslassen sind Pfad und Filter verbunden (Filter ist wie gehabt mit jedem verbundenen Pfad per *und* verknüpft).



Abb.: Mit drei verschiedenen Pfaden verbundener Filter

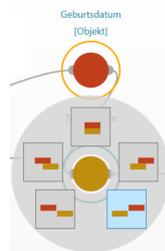
Ereignisse

Ereignisse können, müssen aber kein explizites Datum als Objekt angegeben haben. Die explizite Datumsauswahl erscheint wie bei Fakten beim Klick auf den Text [Objekt] über dem Kreis des Ereignisses. Dabei kann zusätzlich zu einem Datum auch angegeben werden, ob das Ereignis exakt am angegebenen Zeitpunkt oder davor/danach eintreten soll.

Alle **Ereignisse**, die auf dem selben grauen Pfad liegen, oder zu *unterschiedlichen Subjekten* gehören, können auch relativ zueinander angeordnet werden (bei unterschiedlichen *Subjekten* sind die Pfade der verbundenen **Ereignisse** danach *implizit per und verbunden*).

Um zwei **Ereignisse** zu verbinden, muss der Kreis eines dieser **Ereignisse** angeklickt werden und dann die Maus mit gedrückter Maustaste zum andern **Ereignis** bewegt und über dessen Kreis losgelassen werden.

Es erscheint ein kreisförmiges Auswahlmengü, dass die Art der Anordnung wählbar macht:



A. Materialien der Nutzerstudie

Dazu sind auf jedem Knopf des Auswahlmenüs die Anordnung der beiden **Ereignisse** angezeigt, in den Farben, die Sie gerade im Anfrageeditor besitzen.



Abb.: rot und gelb finden gleichzeitig statt



Abb.: rot findet vor gelb statt

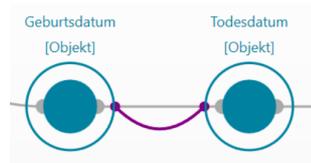


Abb.: rot findet vor oder gleichzeitig zu gelb statt

Nach Auswahl der gewünschten Anordnung wird dann eine neue lilafarbene **Verbindung** zwischen den **Ereignissen** angezeigt.

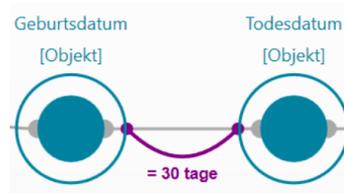
Die horizontale Positionierung der **Ereignisse** gibt Ihre Anordnung an, d.h. von zwei **Ereignissen** findet das weiter links liegende *vor* dem weiter rechts liegenden statt.

Beispiel: Die Geburt einer Person soll *vor* ihrem Tod stattfinden:



Der Abstand zwischen den so angeordneten **Ereignissen** ist *beliebig*, kann aber auch durch Klicken auf die lilafarbene **Zeitverbindung** angegeben werden.

Beispiel: Die Geburt einer Person soll *exakt 30 Tage vor* ihrem Tod stattfinden:



Die Art der Anordnung zweier **Ereignisse**, die durch eine **Zeitverbindung** verbunden sind, kann anhand der Position der **Endpunkte** der **Zeitverbindung** abgelesen werden.

Dabei gibt der Abstand eines **Punkts** einer **Verbindung** zwischen innerem und äußerem Kreis eines **Ereignisses** den Abstand zum **Ereignis** an.

Der Kreis eines **Ereignisses** ist wie ein Planet und der Ring um das **Ereignis** wie das Ende seiner Atmosphäre.

Je näher die Verbindungspunkte an den Planeten liegt, *desto enger liegen die beiden verbundenen Ereignisse beieinander*.

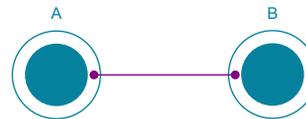
Je weiter die Verbindungspunkte vom Planeten weg, also in der Atmosphäre des Planeten, liegen, *desto weiter auseinander liegen die verbundenen Ereignisse*.

Es wird zwischen drei unterschiedlichen Positionen der Verbindungspunkte unterschieden:

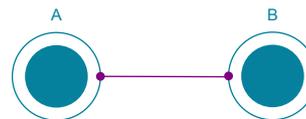
- Treffen die Punkte den inneren Kreis (Planeten), finden die beiden verbundenen Ereignisse gleichzeitig statt



- Liegen die Punkte zwischen innerem und äußerem Kreis (in der Atmosphäre, also weder direkt auf dem Planeten noch direkt am Ende der Atmosphäre), finden die beiden Ereignisse voneinander oder gleichzeitig statt



- Treffen die Punkte den äußeren Kreis (Atmosphärenende), finden die beiden Ereignisse echt voneinander statt.



Subjektverknüpfungen

Subjektverknüpfungen erlauben die Verknüpfung zweier Subjekte.

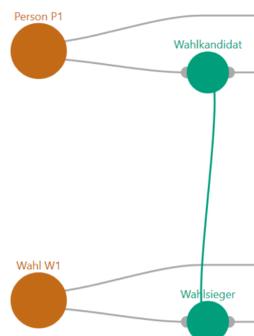
Dazu muss ein Subjektverknüpfungs-Filter zu beiden Subjekten hinzugefügt werden.

In einem Subjektverknüpfungs-Filter muss das gewünschte Prädikat, über das die Subjekte verbunden werden sollen (z.B. Wahlsieger in Subjekt Wahl) angegeben werden.

Subjektverknüpfungen werden verbunden wie Zeitverbindungen bei Ereignissen: durch Klick auf den Kreis einer Subjektverknüpfung, mit gedrückter Maustaste über den Kreis der anderen Subjektverknüpfung zeigen und dann loslassen.

Dabei wird das Prädikat der anderen Subjektverknüpfung automatisch ausgewählt.

Beispiel: Eine Person wird als Wahlsieger einer Wahl definiert :



A. Materialien der Nutzerstudie

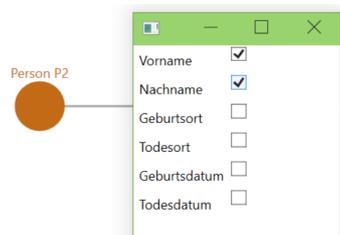
Ergebnisse müssen nun *die Filter beider Subjekte erfüllen*, die mit den gleichen grauen Pfaden wie die *Subjektverknüpfung* verbunden sind, d.h. im obigen Beispiel muss eine Person Wahlsieger einer Wahl sein, und alle Filter, die auf dem grauen Pfad der *Subjektverknüpfung* ‚Wahlkandidat‘ liegen, müssen von der Person erfüllt werden. Ebenso müssen von der Wahl, in der die Person Wahlsieger wurde, alle Filter erfüllt werden, die auf dem grauen Pfad der *Subjektverknüpfung* ‚Wahlsieger‘ liegt.

Ergebnisse

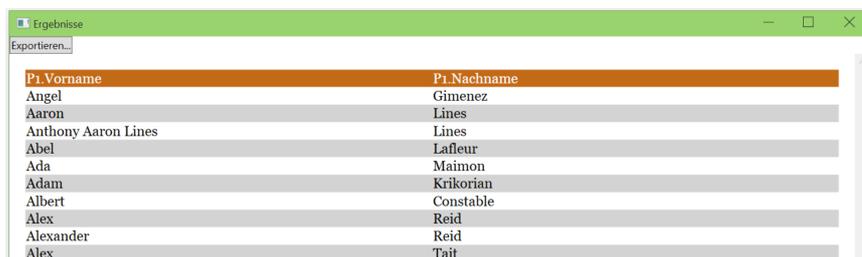
Die angezeigten Eigenschaften eines *Subjekts* in den Ergebnissen können per Rechtsklick auf das *Subjekt* ausgewählt werden:



Nach Klick auf die Schaltfläche ‚Ergebnisspalten bearbeiten‘ erscheint ein Auswahldialog, in dem die Auswahl der anzuzeigenden Eigenschaften ermöglicht wird:



Um eine Anfrage auszuführen und die Ergebnisse zu sehen, muss auf die Schaltfläche ‚Anfrage ausführen‘ (In Abbildung 1 mit Nummer 3 markiert) geklickt werden. Nach kurzem Warten öffnet sich ein Fenster mit den Ergebnissen:



P1.Vorname	P1.Nachname
Angel	Gimenez
Aaron	Lines
Anthony Aaron Lines	Lines
Abel	Lafleur
Ada	Maimon
Adam	Krikorian
Albert	Constable
Alex	Reid
Alexander	Reid
Alex	Tait

Die Ergebnisspalten geben dabei die Nummer des *Anfragesubjekts* an, zu dem die Ergebnisse im Anfrageditor gehören.

A.4. Fragebogen für Probanden

Evaluationsfragebogen					
<small>Bitte Kreuzen Sie die zutreffenden Antworten an.</small>					
	<small>sehr wenig</small>	<small>wenig</small>	<small>normal</small>	<small>viel</small>	<small>sehr viel</small>
Wie viel Zeit haben Sie gefühlt benötigt, um eine Suchanfrage zu erstellen?					
Wie schwer fanden Sie...					
	<small>sehr schwer</small>	<small>schwer</small>	<small>normal</small>	<small>einfach</small>	<small>sehr einfach</small>
Die Verständlichkeit des gesamten Anfragekonzepts					
Das Hinzufügen neuer Subjekte					
Das Hinzufügen neuer Filter					
Das Bearbeiten der Filterwerte (Objekte)					
... und speziell die Festlegung des Zeitpunkts für ein Ereignis					
Das Anordnen von Ereignissen (voreinander, nacheinander, gleichzeitig)					
Die Festlegung des Abstandes zwischen Ereignissen					
Die Lesbarkeit verbundener Ereignisse (visuelle Unterscheidung voreinander/ nacheinander/gleichzeitig + Abstand)					
Das Verknüpfen von Subjekten					
Die Unterscheidbarkeit zwischen <i>und/oder</i>					
Wie schwer fiel Ihnen das Lösen der					
	<small>sehr schwer</small>	<small>schwer</small>	<small>normal</small>	<small>einfach</small>	<small>sehr einfach</small>
Interpretationsaufgabe 1					
Interpretationsaufgabe 2					
Textaufgabe ‚Mozarts Familie‘					
1.1					
1.2					
1.3					
1.4					
1.5					
1.6					
Textaufgabe ‚Politik‘					
2.1					
2.2					
2.3					
2.4					
2.5					
2.6					
Hatten Sie Probleme beim Verständnis des Visualisierungskonzepts der Anfragen? Falls ja, welche?					
Hatten Sie Probleme mit der Bedienung des Programms? Falls ja, welche?					
Sehen Sie Vorteile/Nachteile in der Nutzung der Anwendung? Würden Sie andere Ihnen bekannte Methoden/Werkzeuge bevorzugen (falls ja, in welchen Fällen)?					
Sonstige Anmerkungen:					

Literaturverzeichnis

- [AA83] J. F. Allen, L. F. Allen. Maintaining knowledge about temporal intervals. *Communication of ACM*, S. 832–843, 1983. (Zitiert auf den Seiten 11, 12 und 21)
- [ABF⁺89] P. G. Anick, J. D. Brennan, R. A. Flynn, D. R. Hanssen, B. Alvey, J. M. Robbins. A direct manipulation interface for boolean information retrieval via natural language query. In *Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval*, S. 135–150. ACM, 1989. (Zitiert auf Seite 15)
- [BLHL01] T. Berners-Lee, J. Hendler, O. Lassila. A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, 284(5), 2001. (Zitiert auf Seite 35)
- [BRZ09] G. Barzdins, S. Rikacovs, M. Zviedris. Graphical query language as SPARQL frontend. In *Local Proceedings of 13th East-European Conference (ADBIS 2009)*, S. 93–107. 2009. (Zitiert auf Seite 16)
- [CC01] L. Chittaro, C. Combi. Representing of Temporal Intervals and Relations: Information Visualization Aspects and Their Evaluation. In *Temporal Representation and Reasoning, International Symposium on*, S. 13–13. IEEE Computer Society, 2001. (Zitiert auf den Seiten 12 und 13)
- [CEC87] D. M. Campbell, D. W. Embley, B. Czejdo. Graphical query formulation for an entity-relationship model. *Data & Knowledge Engineering*, 2(2):89–121, 1987. (Zitiert auf Seite 16)
- [CGB13] L. Certo, T. Galvão, J. Borges. Time Automaton: A visual mechanism for temporal querying. *Journal of Visual Languages & Computing*, 24(1):24–36, 2013. (Zitiert auf den Seiten 18, 19 und 33)
- [CO12] C. Combi, B. Oliboni. Visually defining and querying consistent multi-granular clinical temporal abstractions. *Artificial intelligence in medicine*, 54(2):75–101, 2012. (Zitiert auf den Seiten 17 und 18)
- [DAa] DBPedia Association. DBPedia Ressource zu Ludwig van Beethoven. http://dbpedia.org/page/Ludwig_van_Beethoven. (Zitiert auf Seite 38)
- [DAb] DBPedia Associaton. DBPedia. <http://wiki.dbpedia.org/about>. (Zitiert auf Seite 35)

- [Dud14] Duden online. Zeit, die. <http://www.duden.de/node/642492/revisions/1349887/view>, 2014. (Zitiert auf Seite 11)
- [FKSS06] J. A. Fails, A. Karlson, L. Shahamat, B. Shneiderman. A visual interface for multivariate temporal data: Finding patterns of events across multiple histories. In *Visual Analytics Science And Technology, 2006 IEEE Symposium On*, S. 167–174. IEEE, 2006. (Zitiert auf Seite 17)
- [Fra98] A. U. Frank. Different Types of Times in GIS. *Spatial and temporal reasoning in geographic information systems*, S. 40, 1998. (Zitiert auf Seite 21)
- [Hil89] C. R. Hildreth. Intelligent Interfaces and Retrieval Methods for Subject Searching in Bibliographic Retrieval Systems. S. 39–42. ERIC, 1989. (Zitiert auf Seite 14)
- [LFH10] J. Lazar, J. H. Feng, H. Hochheiser. *Research methods in human-computer interaction*. John Wiley & Sons, 2010. (Zitiert auf Seite 46)
- [MKSJ98] S. Miksch, R. Kosara, Y. Shahar, P. D. Johnson. AsbruView: Visualization of Time-Oriented, Skeletal Plans. In *AIPS*, S. 11–18. 1998. (Zitiert auf den Seiten 13 und 14)
- [om210] OMG Unified Modeling Language (omg uml), Infrastructure. Technischer Bericht, Object Management Group, 2010. Version 2.4.1. (Zitiert auf Seite 34)
- [OS] OpenLink Software. OpenLink Virtuoso Server. <http://virtuoso.openlinksw.com>. (Zitiert auf Seite 35)
- [PS08] E. Prud’hommeaux, A. Seaborne. SPARQL Query Language for RDF. <http://www.w3.org/TR/rdf-sparql-query/>, 2008. Änderungen für SPARQL 1.1 unter <http://www.w3.org/TR/sparql11-overview>. (Zitiert auf Seite 35)
- [QVDMW14] Y. Qiang, M. Valcke, P. De Maeyer, N. Van de Weghe. Representing time intervals in a two-dimensional space: an empirical study. *Journal of Visual Languages & Computing*, 25(4):466–480, 2014. (Zitiert auf Seite 12)
- [Shn94] B. Shneiderman. Dynamic queries for visual information seeking. *Software, IEEE*, 11(6):70–77, 1994. (Zitiert auf den Seiten 33 und 41)
- [Shn98] B. Shneiderman. Visual user interfaces for information exploration. 1998. (Zitiert auf den Seiten 14 und 15)
- [SZ12] C. P. Schaaf, J. Zschocke. *Basiswissen Humangenetik*. Springer-Verlag, 2012. (Zitiert auf Seite 24)
- [TGM83] E. R. Tufte, P. Graves-Morris. *The visual display of quantitative information*, Band 2. Graphics press Cheshire, CT, 1983. (Zitiert auf Seite 12)
- [VZA⁺] R. Vesse, R. M. Zettlemoyer, K. Ahmed, G. Moore, T. Pluskiewicz. dotNetRdf-Bibliothek. <http://dotnetrdf.org>. Version 1.0.7.3471. (Zitiert auf Seite 36)
- [War12] C. Ware. *Information visualization: perception for design*. Elsevier, 2012. (Zitiert auf den Seiten 23 und 24)

- [YS98] D. Young, B. Shneiderman. A graphical filter/flow representation of Boolean queries: a prototype implementation and evaluation. 1998. (Zitiert auf den Seiten 14, 15 und 33)
- [ZL14] M. Zviedris, R. Liepiņš. Readability of a diagrammatic query language. In *Visual Languages and Human-Centric Computing (VL/HCC), 2014 IEEE Symposium on*, S. 227–228. IEEE, 2014. (Zitiert auf Seite 28)

Alle URLs wurden zuletzt am 18.05.2015 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.

Unterschrift:

Stuttgart, den 26. Mai 2015

Declaration

I hereby declare that the work presented in this thesis is entirely my own.

I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations.

Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before.

The electronic copy is consistent with all submitted copies.

Signature:

Stuttgart, May 26th, 2015