



Dissertation

eingereicht bei

Fakultät für Informatik und Mathematik
Universität Passau

Ein Ansatz zur semantik-basierten Anfragerelaxation für hierarchische Strukturen

Alexander Stenzer

20. August 2018

Betreuer: Prof. Dr. Burkhard Freitag

Dissertation für die Erlangung eines Doktorgrades in Naturwissenschaften an der Fakultät für Informatik und Mathematik der Universität Passau.

1. *Gutachter:* Prof. Dr. Burkhard Freitag
2. *Gutachter:* Prof. Dr. Ralf Schenkel
3. *Gutachter:* Prof. Dr. Dietmar Seipel

Abstract

For historic buildings as part of our cultural heritage in particular as well as for buildings in general, the MonArch project [Fre16; FS17] developed different methods for the digital storage of information about historic buildings. The resulting MonArch system is used for documenting historic buildings and storing their digital model of a building in a relational database. The digital model of the building is created by segmenting the building into several parts, which can be summarized in a structure hierarchy. In this context a structure hierarchy is a part-of hierarchy of building parts. The structure hierarchy allows to tag information e.g. documents with a spatial reference. In addition, a topic hierarchy allows to describe information by topics.

Looking at spatial and topical queries in networked MonArch systems, in which several building archives are connected with each other, information rigidly fixed to the unique structure of each building is an obstacle for a simple method for spatial query. As each building has its individual structural and spatial composition, a spatial query that is specific to a certain structure hierarchy does not provide any results for other buildings. For topical queries, incompatible topic hierarchies are an obstacle which prevent a general query. The main challenge is to provide an alignment of structural- and thematic hierarchies.

In order to solve the problem described above, in a networked information system, the solution is to transform the original query by a query relaxation to extend the focus of the query or to adapt the query to the conditions of other information system. The query transformation and -relaxation method presented in this thesis uses a generalization to automatically transform a query to a specific structure and topic hierarchy. In the case of topical hierarchies common topics are a starting point. For structural hierarchies, type information of building parts can be used as generalization relationship. The transformed and thereby relaxed query can then be sent to a network of MonArch systems without a manual selection of the building parts in other structural hierarchies or an appropriate selection of topics. The structural hierarchies of other buildings in the network of MonArch systems can be unknown. In this thesis different relaxation methods, e.g. an adapted spreading activation method [Aki+11], for automatic query transformation of spatial and topical queries is presented with the aim to avoid a complete alignment between the structural hierarchies of buildings and topic hierarchies. The objective is achieved by the extension of the MonArch data model and queries, which allow a query transformation at runtime.

Kurzfassung

Für Monumentalbauten als Teil unseres Kulturgutes im Speziellen als auch für Gebäude im Allgemeinen, wurden im Rahmen des MonArch-Projektes [Fre16; FS17] verschiedene Methoden zur digitalen Speicherung von Informationen über Monumentalbauten erforscht. Das daraus entstandene MonArch-System ist für die Dokumentation von Monumentalbauten verwendbar und speichert das digitale Modell des Bauwerks in einer relationalen Datenbank. Das digitale Modell des Bauwerks entsteht durch eine Segmentierung in Gebäudeteile, die dann in einer Strukturhierarchie zusammengefasst werden können. Als Strukturhierarchie versteht man in diesem Zusammenhang eine Hierarchie von Gebäudeteilen, die in einer Teil-von-Beziehung stehen. Die Strukturhierarchie erlaubt es Informationen z.B. Dokumente mit einem räumlichen Bezug auszuzeichnen. Zusätzlich wird eine Themenhierarchie unterstützt, die es erlaubt Informationen thematisch mit Begriffen zu beschreiben.

Betrachtet man räumliche und thematische Anfragen in vernetzten MonArch-Systemen, in denen sich mehrere Gebäudearchive zusammenschließen, ist diese starke Bindung der Information an die einzigartige Struktur jedes Gebäudes ein Hindernis für ein einfaches Verfahren zur räumlichen Suche. Da sich jedes Gebäude in seinem speziellen strukturellen und räumlichen Aufbau unterscheidet, liefert eine räumliche Anfrage, die speziell auf diese Eigenheiten eines Gebäudes ausgerichtet ist, für andere Gebäude keine Suchergebnisse. Für thematische Anfragen stellen nicht kompatible Themenhierarchien ein Hindernis dar, die eine übergreifende thematische Anfrage verhindern. Die größte Herausforderung ist es, Struktur- und Themenhierarchien aufeinander abzubilden.

Zur Lösung des geschilderten Problems wird in vernetzten Informationssystemen auf eine geeignete Transformation der ursprünglichen Anfrage zurückgegriffen, um den Anfragefokus zu erweitern (Relaxation) oder eine Anpassung an die Gegebenheiten des entfernten Informationssystems zu erreichen (Transformation). Das Anfragetransformations- und -relaxationsverfahren, das in dieser Arbeit vorgestellt wird, nutzt eine Generalisierungsbeziehung aus, um ausgehend von einer Anfrage an eine spezielle Struktur- und Themenhierarchie eine automatische Transformation der Anfrage durchzuführen. Bei Themenhierarchien sind gemeinsame Oberthemen ein Ansatzpunkt. Bei Strukturhierarchien können Typinformationen zu Gebäudeteilen die Generalisierungsbeziehung darstellen. Die transformierte und dadurch relaxierte Anfrage kann dann an ein Netzwerk von MonArch-Systemen gestellt werden, ohne dass eine manuelle Auswahl der Gebäudeteile in anderen Strukturhierarchien oder eine angepasste Themenauswahl erfolgen muss. Dazu muss die Strukturhierarchie der anderen Gebäude im Netzwerk von MonArch-Systemen nicht bekannt sein. Im Rahmen der vorliegenden Arbeit werden verschiedene Relaxationsverfahren, z.B. ein angepasstes Spreading-Activation-Verfahren [Aki+11], zur automatischen Anfragetransformation von

räumlichen und thematischen Anfragen vorgestellt, mit dem Ziel eine vollständige Abbildung zwischen den Strukturhierarchien von Gebäuden und Themenhierarchien zu vermeiden. Erreicht wird das Ziel durch eine Erweiterung des MonArch-Datenmodells und eine Verallgemeinerung der MonArch-Anfragen, die eine Anfragetransformation zum Anfragezeitpunkt erlauben.

Inhaltsverzeichnis

1.	Einleitung	1
1.1.	Motivation	1
1.2.	Fragestellung und Herangehensweise	4
1.3.	Wissenschaftlicher Beitrag	6
2.	Anwendungsfall MonArch	9
2.1.	MonArch-Methodik	10
2.2.	MonArch-Datenmodell	17
2.2.1.	Strukturhierarchie und grafische Repräsentation	17
2.2.2.	Themenhierarchie	19
2.2.3.	Dokumente	20
2.2.4.	Metaebenen des Datenmodells	21
2.3.	MonArch-Archive St. Lorenz - St. Sebald	22
2.4.	MonArch-Anfragen	26
2.4.1.	Räumliche Anfrage	26
2.4.1.1.	Räumliche Anfrage - Variante eins	27
2.4.1.2.	Räumliche Anfrage - Variante zwei	28
2.4.1.3.	Räumliche Anfrage - Variante drei	29
2.4.1.4.	Räumliche Anfrage - Geometrisch	30
2.4.2.	Thematische Anfrage	31
2.4.3.	Anfragekombination	33
2.5.	MonArch-Verbund	35
3.	Erweitertes Datenmodell	39
3.1.	Strukturgraph	40
3.2.	Themengraph	46
3.3.	Objektmodell	48
3.4.	Datenmodell in Beschreibungslogik	52
3.4.1.	Raum- und Typontologie	57
3.4.2.	Themaontologie	60
3.4.3.	Objektontologie	62
3.4.4.	Realisierung in OWL-2-DL	64
3.4.5.	Konsistenz und Zusammenspiel der Ontologien	68
3.5.	Fazit	70

4.	Anfragemodell	71
4.1.	Anforderungen an MonQL	73
4.2.	Grundstruktur von MonQL	74
4.3.	Räumliche Anfrage	78
4.4.	Thematische Anfrage	86
4.5.	Typbasierte Anfrage	91
4.6.	Anfragekombination	92
4.7.	Übersetzung in SPARQL	96
4.8.	Fazit	102
5.	Anfragerelaxation und -transformation	103
5.1.	Fragstellung	107
5.2.	Zieleigenschaften	111
5.3.	Anwendbarkeit existierender Verfahren	114
5.3.1.	Klassen von Verfahren des Ontology-Matchings	116
5.3.2.	Kombinierte Structural-level Verfahren	117
5.3.3.	Zusammenfassung	121
5.4.	Anfragetransformation	121
5.4.1.	Anfragetransformation im Quellsystem	123
5.4.2.	Anfragetransformation im Zielsystem	124
5.4.3.	Betrachtung Intentionserhaltung	124
5.4.4.	Spreading-Activation	125
5.5.	Transformation thematischer Anfragen	136
5.5.1.	Relaxation durch Verallgemeinerung	138
5.5.2.	Spreading-Activation für die thematische Anfragerelaxation	141
5.5.2.1.	Relaxation durch Verallgemeinerung	145
5.5.2.2.	Relaxation durch Verwandtschaft	147
5.6.	Transformation räumlicher Anfragen	153
5.6.1.	Transformation durch Typanfragen	156
5.6.2.	Spreading-Activation für die räumliche Anfragerelaxation	160
5.7.	Kombination der Anfragerelaxation	168
5.8.	Eigenschaften und Grenzen des Transformationsansatzes	171
6.	Eignung für den praktischen Einsatz	175
6.1.	Verwendete Gütekriterien	175
6.2.	Anpassung des Spreading-Activation-Verfahren	179
6.3.	Auswertung der Ergebnisse	181
6.4.	Täglicher Einsatz	184
7.	Zusammenfassung und Ausblick	187
A.	Themenhierarchie	189
B.	Ontologien	191

C.	Abbildung DL nach OWL-2-DL	201
D.	MonQL Grammatik	203
	Literatur	207
	Abbildungsverzeichnis	219
	Tabellenverzeichnis	221
	Liste der Listings	223
	Liste der Algorithmen	225

1. Einleitung

Im Bereich des Kulturguterhaltes spielt die Digitalisierung eine immer wichtigere Rolle. Dies bedeutet, dass Informationen über Kulturgüter zusätzlich zur analogen Form immer mehr in digitaler Form vorliegen. Verschiedene moderne Technologien versetzen uns in die Lage sehr große Bestände an Schriftdokumenten, Planzeichnungen, Fotografien, Büchern und andere Archivalien zu digitalisieren.

Die im Zuge der Digitalisierung stetig wachsende Menge und Komplexität der verfügbaren Informationen erfordert in gleichem Maße die Entwicklung komfortabler und in der Praxis einsetzbarer Lösungen für die Speicherung, Vernetzung, Suche und Analyse der Daten.

Ein aktuelles Beispiel zeigt wie schnell digitale Informationen entstehen. Das Projekt „Europeana.eu“¹ bietet zum Zeitpunkt dieser Arbeit Zugriff auf über 40 Millionen Bücher, Filme, Zeichnungen, museale Objekte und andere Archivalien [Fou17]. Die erste Version startete 2009 und in 2010 waren bereits über 10 Millionen digitale Objekte verfügbar [Com10].

1.1. Motivation

Für Monumentalbauten als Teil unseres Kulturgutes im Speziellen als auch für Gebäude im Allgemeinen, wurden im Rahmen des MonArch-Projektes [Fre16; FS17] verschiedene Methoden zur digitalen Speicherung von Informationen über Monumentalbauten erforscht.

Das in diesem Projekt entwickelte MonArch-System ist für die Dokumentation von Bauten wie Monumentalbauten, archäologische Denkmäler oder urbane Situationen verwendbar. Ein MonArch-System ist eine Client-Server-Anwendung. Die Client-Anwendung ermöglicht die strukturelle und grafische Visualisierung eines digitalen Modells des Bauwerks. Die Server-Anwendung koordiniert den Mehrbenutzerbetrieb und speichert in einer relationalen Datenbank das digitale Modell des Bauwerks. Das digitale Modell des Bauwerks entsteht durch eine Segmentierung in Gebäudeteile, die dann in einer Strukturhierarchie zusammengefasst werden können. Als Strukturhierarchie versteht man in diesem Zusammenhang eine Hierarchie von Gebäudeteilen die in einer Teil-von-Beziehung stehen.

Zusätzlich zu dieser digitalen Repräsentation können weitere Informationen über das Bauwerk in Form von digitalen Dokumenten in der Datenbank gespeichert werden.

¹<http://research.europeana.eu/>

Der Dokumentenbestand der MonArch-Systeme umfasst dabei folgende Arten von Dokumenten, ohne einen Anspruch auf Vollständigkeit:

- Scans von alten Bauplänen,
- Scans von alten Fotos und digitale Fotos, die den Zustand der Gebäude dokumentieren,
- Rechnungen von alten und neuen Restaurierungs- und Baumaßnahmen,
- CAD-Pläne für verschiedene Zwecke, von Bauplänen bis zu Schadenskartierungen,
- Rezepturen und Arbeitsanweisungen für die Restaurierung.

Die Strukturhierarchie erlaubt es, für jedes Dokument in der Datenbank einen räumlichen Bezug zu speichern. Ein Dokument wird dadurch sozusagen an ein Gebäudeteil angehängt. Mithilfe dieser Verortung von Dokumenten können Anfragen an das MonArch-System gestellt werden, die den räumlichen Kontext berücksichtigen, z.B.:

„Zeige alle Dokumente, die einen Bezug zu einem bestimmten Pfeiler eines speziellen Gebäudes aufweisen.“

Das MonArch-System ermöglicht neben der Verortung von Dokumenten entlang der Strukturhierarchie die Auszeichnung der Dokumente mit Begriffen aus Vokabularen. Diese Vokabulare beschreiben z.B. das verwendete Material, festgestellte Schadensarten oder eine zeitliche Einordnung. Diese thematische Auszeichnung der Dokumente bildet die Grundlage für weitere Anfragen an das MonArch-System, beispielsweise:

„Zeige alle Dokumente die Schäden an einer bestimmten Gesteinsart dokumentieren.“

Eine Instanziierung des MonArch-Systems für ein Gebäude wird als MonArch-Archiv bezeichnet und ermöglicht den Anwendern, z.B. Architekten, die mit der Restaurierung des Gebäudes betraut sind, einen Zugriff auf den Informationsbestand.

Ein MonArch-Archiv ist dabei nicht auf ein Gebäude beschränkt. Dies ermöglicht es z.B. einer Organisation, alle ihre Gebäude mit einem MonArch-Archiv zu dokumentieren. Es existieren Szenarien, in denen mehrere Gebäude zu unterschiedlichen Organisationen gehören, es aber trotzdem erwünscht ist, den gesamten Informationsbestand gemeinsam zu durchsuchen.

Das MonArch-Projekt stellte sich unter anderem der Frage, wie MonArch-Archive miteinander vernetzt werden können. Neben den technischen Herausforderungen, die es zu lösen galt, um die MonArch-Archive zu vernetzen, kam der Wunsch nach einem Anfragemechanismus auf, welcher die Gebäude in einem MonArch-Netzwerk nicht isoliert betrachtet.

Werden mehrere Gebäude in einem MonArch-Archiv oder in einem Netzwerk von MonArch-Archiven dokumentiert, wird jedes Gebäude durch eine Strukturhierarchie repräsentiert.

Betrachtet man folgende Anfrage:

„Zeige alle Dokumente die einen Bezug zu Pfeilern in einem Netzwerk von MonArch-Archiven aufweisen.“

Diese Art von Anfrage lässt sich, obwohl sie Bezug auf Gebäudeteile nimmt, nicht einfach beantworten. Die Strukturhierarchien sind für jedes Gebäude unterschiedlich. Daher müssen bei der hier skizzierten Anfrage für jede Strukturhierarchie - also für jedes Gebäude - Pfeiler identifiziert werden, um die Anfrage zu realisieren.

Jetzt könnte man die Identifizierung der Pfeiler dem Anwender überlassen. Diese Herangehensweise wäre zwar möglich und führt auch zum gewünschten Ergebnis, hätte aber folgende Nachteile:

- Je nach Anzahl der Gebäude im Netzwerk kann die Identifizierung der Gebäudeteile sehr aufwendig sein. Betrachtet man ein Netzwerk aus Monumentalbauten wie z.B. Sakralbauten kann die Anzahl der Pfeiler und der damit manuell auszuwählenden Gebäudeteile einen enormen Aufwand erzeugen.
- Je nach Gebäudeteil das in der Anfrage selektiert werden soll, kann bereits in einem einzelnen Gebäude ein enormer Aufwand entstehen. Möchte man z.B. alle Fenster eines Gebäudes für eine Anfrage selektieren, kann dies bei einem Einfamilienhaus praktisch noch möglich sein, bei einem Sakralbau oder einem Hochhaus wird dies in der Praxis schnell sehr mühsam.

Während die manuelle Auswahl für den Fall eines einzelnen Bauwerks und damit eines einzelnen MonArch-Archivs noch durch einen geeigneten Gruppierungsmechanismus unterstützt werden könnte, ist die Situation im Fall mehrerer Bauwerke und Archive ungleich schwieriger.

- Damit eine manuelle Selektion möglich ist muss dem Anwender Zugriff auf alle Strukturhierarchien im Detail gewährt werden. Dies kann im Allgemeinen aber nicht sichergestellt werden, da z.B. Sicherheitsbedenken oder Datenschutz dies nicht erlauben.
- Um über die Relevanz von Gebäudeteilen für eine Anfrage entscheiden zu können, müsste ein Anwender jede Strukturhierarchie jedes Gebäudes und der darin enthaltenen Gebäudeteile kennen. Diese Forderung ist in der Realität nicht haltbar, da z.B. ein Anwender nicht zwangsläufig ein Experte für alle Gebäude in einem Netzwerk aus MonArch-Archiven ist.

Diese Nachteile zeigen, dass eine manuelle Spezifizierung der Anfrage nicht zielführend ist und ein Automatismus diese Aufgabe übernehmen sollte.

1.2. Fragestellung und Herangehensweise

Diese Arbeit präsentiert ein Verfahren, das ausgehend von einer Anfrage an ein MonArch-Archiv Anfragen generiert, die an ein Netzwerk von MonArch-Archiven gestellt werden, ohne dass der Anwender dabei eingreifen muss.

Die größte Herausforderung bei der Realisierung eines Verfahrens ist es Strukturhierarchien aufeinander abzubilden. Es wird sich zeigen, dass bei genauerer Betrachtung des Aufbaus von Gebäuden und deren Gebäudeteilen nicht nur eine Strukturhierarchie sondern auch weitere Informationen abgeleitet werden können.

Obwohl die konkrete Strukturhierarchie eines Gebäudes einzigartig ist, lassen sich doch Gemeinsamkeiten von Gebäudetypen erarbeiten. Ein Sakralbau mag eine unterschiedliche Anzahl an Fenstern, Türmen oder Pfeilern aufweisen, die auch an unterschiedlichen Orten im Gebäude zu finden sind. Trotzdem sind die Bausteine der Gebäude oft identisch oder weichen nur marginal voneinander ab.

Die Idee ist es die Gebäudeteile zu typisieren und eine Gebäudeteiltyphierarchie zu modellieren, welche Gebäudeteile anhand ihres Typs vergleicht.

Hierzu führen wir Typinformationen ein, die in der Lage sind Gemeinsamkeiten unterschiedlicher Strukturhierarchien zu modellieren.

Das Anfragetransformationsverfahren, das in Kapitel 5 vorgestellt wird, nutzt diese Typinformation der Gebäudeteile aus, um eine automatische Transformation einer Anfrage durchzuführen. Die transformierte Anfrage kann dann an ein Netzwerk von MonArch-Archiven gestellt werden, ohne dass eine manuelle Auswahl der Gebäudeteile erfolgen muss. Es muss nicht einmal die Strukturhierarchie der anderen Gebäude im Netzwerk von MonArch-Archiven bekannt sein und die vorher beschriebenen Nachteile einer manuellen Selektion der Gebäudeteile werden dabei weitestgehend vermieden.

Ausgehend von der vorher beschriebenen Anfrage:

„Zeige alle Dokumente die einen Bezug zu einem bestimmten Pfeiler eines speziellen Gebäudes aufweisen.“

kann das Anfragetransformations- und -relaxationsverfahren die Anfrage analysieren, selektierte Gebäudeteile in der Anfrage identifizieren und über die Typinformation eine Anfrage generieren, die Gebäudeteile gleichen Typs in den anderen Strukturhierarchien der MonArch-Archive des Netzwerks selektiert. Das Ergebnis der Anfragetransformation ist eine Anfrage, die nach Gebäudeteiltypen selektiert:

„Zeige alle Dokumente die einen Bezug zu Pfeilern aufweisen.“

wobei hier Pfeiler der Gebäudeteiltyp für einen bestimmten Pfeiler ist.

Die Relevanz und Notwendigkeit für solch ein Verfahren illustriert folgendes Beispiel:

Beispiel 1.1 [Schadensanalyse Dombaumeisterin]

Stellen wir uns vor, eine Dombaumeisterin hat einen Schaden am Pfeiler mit der internen Nummer 1 im Westbau ihres Doms festgestellt.

Sie stellt Informationen zu dem beobachteten Schadensphänomen zusammen, indem sie das MonArch-System für ihre Recherche nutzt. Durch eine Anfrage nach Schäden am Pfeiler 1 konnte sie die Ursache des Schadens ermitteln, da dieses Schadensphänomen in der Vergangenheit schon einmal aufgetreten ist.

Die Dokumentationslage erlaubt es ihr zwar vom beobachteten Schaden am Gebäude und den Dokumentationen des Schadens in der Vergangenheit auf die Ursache zu schließen. Wie aber in der Vergangenheit dieser Schaden behoben wurde, ist leider nicht dokumentiert.

Ihr erster Ansatz besteht darin, dass derselbe Schaden an einem anderen Pfeiler im Dom ebenfalls aufgetreten ist und dort auch die Schadensbehebung dokumentiert wurde. Dazu weitet sie ihre Anfrage auf alle Pfeiler aus. Aber auch diese Recherche liefert keine weiteren Ergebnisse.

Ihr zweiter Ansatz besteht darin auf das gesammelte Wissen ihrer Dombaumeister-Kollegen zuzugreifen, die das gleiche Schadensphänomen an ihren jeweiligen Gebäuden erfolgreich repariert haben. Da sie aber nicht weiß bei welchem Gebäude und bei welchen Pfeiler die Information hinterlegt ist, stellt sie eine Anfrage an das Netzwerk von MonArch-Archiven.

Dies ist eine Situation in der die automatische Anfragetransformation in der Lage ist die ursprüngliche Anfrage, die sich speziell auf den Pfeiler 1 im Dom der Dombaumeisterin bezieht, so zu relaxieren, dass die transformierte Anfrage den speziellen Pfeiler 1 durch Pfeiler in anderen Sakralbauten ersetzt. Das Ergebnis der relaxierten Anfrage liefert passend zum Schadensphänomen Informationen zur Schadensbehebung. □

Das hier beschriebene Verfahren unterscheidet sich dabei von verteilten Anfragen in verteilten Datenbanken [ÖV11]:

- In diesem Szenario findet keine Fragmentierung der Daten im klassischen Sinne statt. D.h. jedes MonArch-Archiv verwaltet seine eigenen Daten und verteilt seine Daten nicht auf mehrere Archive.
- Das Schema der einzelnen MonArch-Archive unterscheidet sich nicht, weshalb Schema-Matching-Verfahren für die Transformation nicht relevant sind. Jedes MonArch-Archiv hat das gleiche interne Datenbankschema. Lediglich die Ausprägung, also die konkrete Strukturhierarchie, ein Teil der Vokabulare und die daran verorteten und ausgezeichneten Dokumente unterscheiden sich.
- Es sind keine verteilten Operationen, wie z.B. verteilte Joins für die verteilte Anfragebearbeitung nötig. Das Anfragetransformationsverfahren verändert die Anfrage in einer Art und Weise, dass in jedem MonArch-Archiv im Netzwerk die Anfrage unabhängig interpretiert werden kann, ohne auf Informationen von anderen Archiven zurückgreifen zu müssen.

- Das Anfragetransformationsverfahren ist nicht auf einen verteilten Index angewiesen. Jedes Archiv hält für seine Strukturhierarchie und seine Vokabulare lokale Indizes die bei der Anfrageverarbeitung genutzt werden können, unabhängig davon ob es sich um eine lokale Anfrage oder eine transformierte Anfrage handelt.

Die vorliegende Arbeit gliedert sich in sieben Kapitel. In Kapitel 2 wird der MonArch-Anwendungsfall eingeführt, der im weiteren Verlauf immer wieder als Motivation und Beispielgeber dient. Kapitel 3 führt ein konsolidiertes und erweitertes Datenmodell ein, das in der Lage ist den MonArch-Anwendungsfall abzubilden und darüber hinaus auch weitere ähnliche Anwendungsfälle. Kapitel 4 führt eine Anfragesprache ein, die als Grundlage für die spätere Anfragetransformation notwendig ist, zusätzlich aber auch die Anfragen des MonArch-Anwendungsfalls abdeckt. Die Anfragetransformation wird in Kapitel 5 beschrieben. In diesem Kapitel wird auch der Bezug zu bestehenden Arbeiten und Verfahren hergestellt. Kapitel 6 diskutiert die Eignung der Anfragetransformation für den praktischen Einsatz. Abgeschlossen wird diese Arbeit mit einer Zusammenfassung und einem Ausblick (siehe Kapitel 7).

1.3. Wissenschaftlicher Beitrag

Die Hauptbeiträge der vorliegenden Arbeit sind folgende:

- Konsolidierung und Formalisierung eines Datenmodells
 - zur Repräsentation von Gebäuden, insbesondere der Segmentierung in Gebäudeteile und des räumlichen Zusammenhangs der Gebäudeteile untereinander.
 - zur Repräsentation von Themenhierarchien.
- Erweiterung des Datenmodells
 - mit zusätzlichen Typinformationen für Gebäudeteile.
 - von einer Strukturhierarchie zum Strukturgraphen.
 - von einer Themenhierarchie zum Themengraphen.
- Reformulierung des erweiterten Datenmodells in Beschreibungslogik
 - zur Anreicherung um weitere Eigenschaften und Überprüfung der Eigenschaften auf Konsistenz.
 - als Basis für die Überprüfung von bestehenden Verfahren.
 - als Grundlage für den Datenaustausch in OWL-2-DL.
- Definition einer Anfragesprache für räumliche und thematische Anfragen
 - zur kompakten Formulierung der Anfragen im MonArch-Anwendungsfall.

- zur Reduktion der Komplexität für eine unabhängige Anfragetransformation.
- Definition, Realisierung und Bewertung eines Anfragetransformations- und -relaxationsverfahrens für räumliche und thematische Anfragen im Zusammenhang vernetzter MonArch-Archive mit dem Ziel:
 - Gebäudeübergreifende räumliche und thematische Verbundanfragen in den vernetzten Archiven zu ermöglichen.
 - Anfragen sowohl räumlich als auch thematisch zu relaxieren um neue Anfrageszenarien zu ermöglichen.
 - die Intention der zu transformierenden Anfrage zu erhalten.

2. Anwendungsfall MonArch

Im Rahmen des von der DFG geförderten Projektes MonArch [Fre16; FS09] wurde eine Methodik zur Aufarbeitung und digitalen Sicherung von konventionellen Archiven von Monumentalbauten erforscht. Das interdisziplinäre Projekt beleuchtete dabei diese Methodik zum einen aus der Sicht der Informatik, zum anderen aus der Sicht der Restaurierungswissenschaft und der praktischen Bauwerkspflege.

Das Ziel von MonArch war es, die Archive von Monumentalbauten digital zugänglich zu machen, sie zu katalogisieren, zu inventarisieren, digital zu sichern und so ihr kulturelles Erbe zu bewahren. Die daraus resultierende primäre Fragestellung aus Sicht der Informatik war die Entwicklung eines passenden Indizierungsschemas das in der Lage ist, die semantische Beschreibung der digitalisierten Inhalte der Archive zu indexieren, so dass Anfragen der Nutzer, wie z.B. Wissenschaftler als auch Architekten und Dombaumeister, beantwortet werden können.[FS09]

Die Ausgangssituation im Projekt stellte sich wie folgt dar. Es wurden zwei Großkirchen - St. Lorenz und St. Sebald in Nürnberg - mit ihren konventionellen Archiven für die Erschließung ausgewählt. Die konventionellen Archive beider Gebäude waren dabei nur teilweise erschlossen und inventarisiert. Abbildung 2.1 zeigt die grundlegenden Schritte, die während der Erschließung der Archive durchgeführt wurden.

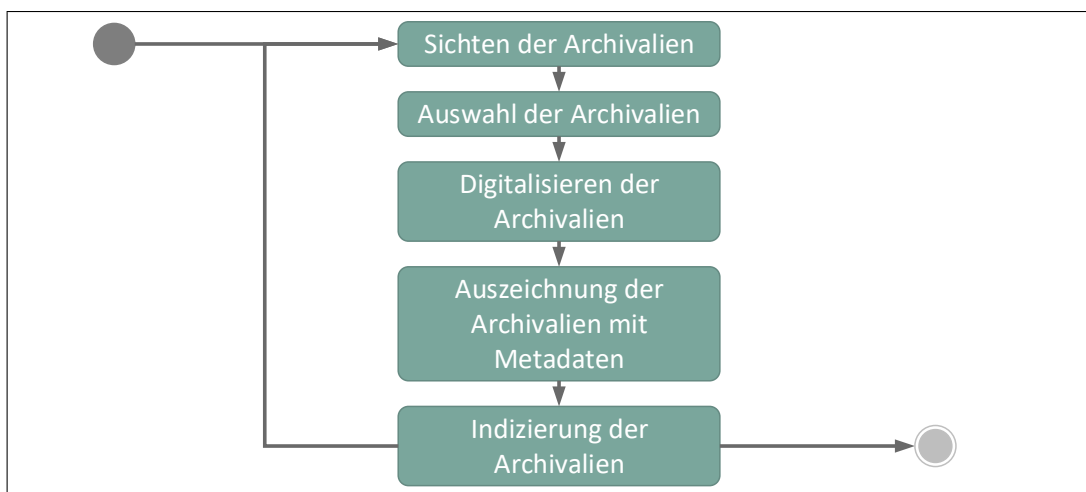


Abbildung 2.1.: MonArch-Arbeitsschritte bei der Erschließung von Archivalien

Ein Team aus verantwortlichen Architekten, Dombaumeistern und Wissenschaftlern aus dem Bereich der Restaurierungswissenschaft sichtete im ersten Schritt die Archivalien der Archive, wie z.B. Pläne, Bautagebücher, Bildaufnahmen auf verschiedenen Medien und Rechnungen.

Im zweiten Schritt wurde eine Auswahl getroffen welche Archivalien für eine Digitalisierung in Frage kommen.

Anschließend wurden die ausgewählten Archivalien im dritten Schritt digitalisiert.

Im vierten Schritt, welcher einer der aufwendigsten war, wurden die nun digitalisierten Archivalien, die als Dateien vorliegen, mit Metadaten ausgezeichnet. Die Auszeichnung mit Metadaten erfordert ein genaues Verständnis der in den Archivalien dargestellten oder beschriebenen Informationen, weshalb diese Aufgabe nur von Experten der jeweiligen Domäne durchgeführt werden kann.

Die Auszeichnung der digitalen Archivalien mit Metadaten bildete die Grundlage für die Indizierung der Archivalien, was den fünften und letzten Schritt darstellt. Zu diesem Zweck wurden im Projekt Indizierungsschemata konzipiert und Softwareprogramme entwickelt, um den verantwortlichen Architekten, Dombaumeistern und Wissenschaftlern ein Werkzeug an die Hand zu geben mit dem sie die beschriebenen Schritte effizient durchführen können.

Für die in den folgenden Kapiteln beschriebenen Verfahren zur Anfragetransformation und -relaxation stellt das MonArch-Projekt das primäre Anwendungsgebiet dar.

2.1. MonArch-Methodik

Das MonArch-Informationsmodell basiert auf einer grafischen Repräsentation des Gebäudes und einer hierarchischen Zerlegung des Gebäudes in Gebäudeteile [SWF11a]. Die hierarchische Zerlegung eines Gebäudes in Gebäudeteile wird in der MonArch-Methodik als Strukturhierarchie (siehe Definition 2.1) bezeichnet.

Definition 2.1 [Strukturhierarchie]

Als *Strukturhierarchie* bezeichnen wir die hierarchische Zerlegung eines Gebäudes in Gebäudeteile. Die Segmentierung ist dabei frei gewählt und basiert in der Regel auf einer etablierten Sicht auf ein Gebäude.

Der Wurzelknoten in der Strukturhierarchie entspricht dem Gebäude selbst. Die darunterliegenden Knoten der Hierarchie repräsentieren Gebäudeteile die zueinander in einer Teil-von-Beziehung stehen. □

Eine Strukturhierarchie lässt sich für verschiedene Gebäude erstellen. Sei es ein Einfamilienhaus das sich z.B. in *Stockwerke* und *Zimmer* oder eine Kirche die sich z.B. in *Türme*, *Schiffe* und den *Chor* aufteilen lässt. Zerlegt man ein Gebäude in seine Einzelteile so ergeben sich Gebäudeteile, die wiederum in kleinere Gebäudeteile zerlegt werden können. Eine *Wand* lässt sich z.B. in ihre *Steine* und dem darin vorkommenden *Fenster* zerlegen. Eine *Kirchenfassade* lässt sich beispielsweise in ihre *Fensterflächen* und *Pfeiler* zerlegen.

Zusätzlich zu dieser baumartigen Darstellung des räumlichen Aufbaus des Gebäudes können die einzelnen Gebäudeteile und Bereiche mit einer zweidimensionalen grafischen Repräsentation verknüpft werden [SSF10]. Die grafische Repräsentation, die z.B. als Vektorzeichnung vorliegt, kann ein Grundriss oder eine Seitenansicht des Gebäudes sein. Je nach Bedarf kann auch nur ein Ausschnitt eines Grundrisses mit höherem Detailgrad verwendet werden.

Das folgende Beispiel 2.2 verdeutlicht den Zusammenhang zwischen hierarchischer Zerlegung und grafischer Repräsentation.

Beispiel 2.2 [Hierarchische symbolische Teil-von-Zerlegung der Westfassade von St. Lorenz mit verknüpfter zweidimensionaler Seitenansicht]

Für die Westfassade von St. Lorenz könnte ein Ausschnitt der Zerlegung wie folgt aussehen:

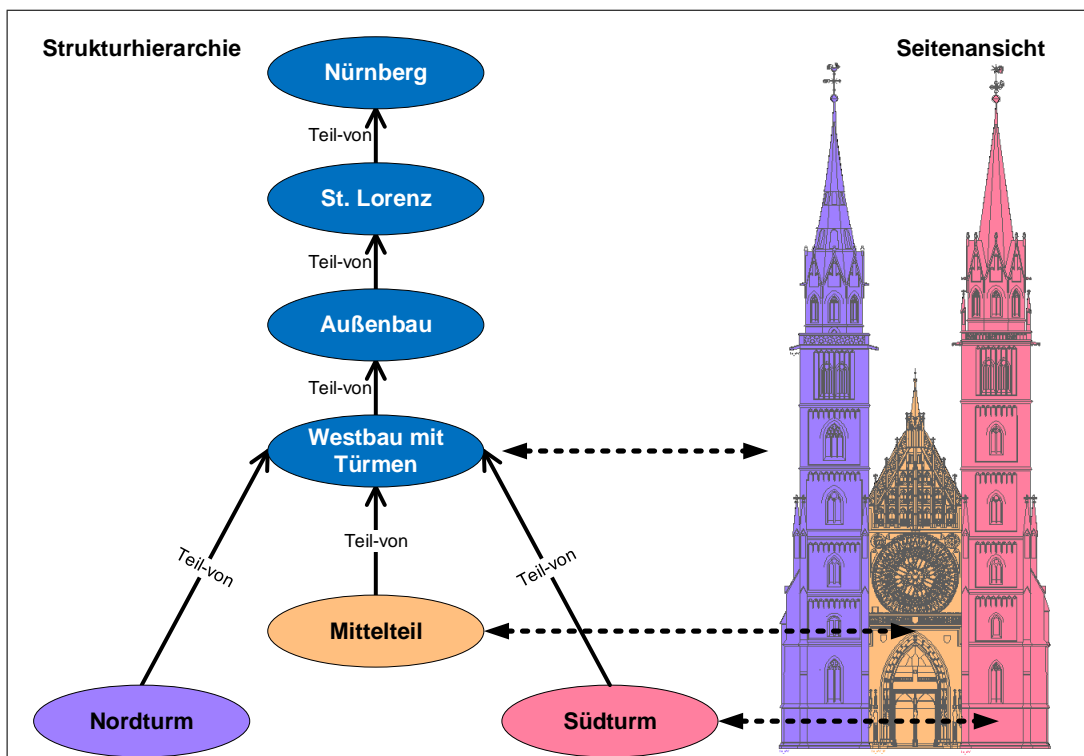


Abbildung 2.2.: Zusammenhang zwischen Zerlegung und Seitenansicht

Begonnen wird mit dem Ort **Nürnberg** in dem das Kirchengebäude **St. Lorenz** liegt. **St. Lorenz** kann in zwei Bereiche aufgeteilt werden: den **Außenbau** und den **Innenbau**. In diesem Beispiel wird der **Außenbau** betrachtet und nur dieser exemplarisch weiter verfeinert. Im **Außenbau** gibt es unter anderem den **Westbau mit Türmen**. Der **Westbau mit Türmen** kann in **Nordturm**, **Südturm** und den **Mittelteil** aufgeteilt werden. Der **Westbau mit Türmen** wird von einer Seitenansicht von **St. Lorenz** grafisch repräsentiert. In dieser Seitenansicht gibt es wiederum drei grafische Repräsentationen der Bauteile **Nordturm**, **Südturm** und **Mittelteil**.

Diese Verknüpfung beider Repräsentationen erlaubt es, z.B. das reale Objekt **Nordturm** als symbolisches Element in der hierarchischen Zerlegung oder als grafisches Element in der

Seitenansicht zu verstehen. □

Die Strukturhierarchie dient neben der Repräsentation des Gebäudes auch als räumlicher Anker für weitere Informationen in Form von Dokumenten (siehe Definition 2.3).

Definition 2.3 [Dokument]

Ein *Dokument* bezeichnet im MonArch-System eine Datei, wie z.B. eine Textdatei, eine Bilddatei oder eine CAD¹-Datei, die mit mindestens einem Gebäudeteil der Strukturhierarchie verknüpft bzw. an diesem verortet ist.

Dokumente sind die primären Objekte im MonArch-Informationsmodell und Ergebnisse von MonArch-Anfragen. □

Dokumente werden an Gebäudeteilen verortet, wobei der Inhalt des Dokuments etwas über das Gebäudeteil aussagt oder das Gebäudeteil im Dokument vorkommt. Die vorher beschriebenen Archivalien werden als Dokumente in einem MonArch-System gespeichert.

Neben dem räumlichen Bezug der durch die Verortung der Dokumente an Gebäudeteilen ermöglicht wird, sieht die MonArch-Methodik auch die Beschreibung des thematischen Inhalts der Dokumente vor. Dazu kann eine Begriffshierarchie modelliert werden. Die einzelnen Begriffe dieser Begriffshierarchie werden im Folgenden Themen (siehe Definition 2.4) genannt.

Definition 2.4 [Thema]

Unter einem *Thema* verstehen wir im MonArch-System einen Begriff aus einer Begriffshierarchie (Ontologie). Diese Begriffshierarchie stellt ein Fachvokabular dar das entweder aus Standards entnommen ist oder im Rahmen des MonArch-Projektes entwickelt wurde. Die einzelnen Begriffe stehen dabei in einer Ober- und Unterbegriffsbeziehung zueinander. Innerhalb dieser Beziehung formen die Themen eine Themenhierarchie. □

Beispiel 2.5 zeigt einen Ausschnitt aus der im MonArch-Projekt entstandenen Themenhierarchie.

¹computer-aided design

Beispiel 2.5 [Ausschnitt aus der Themenhierarchie für Material und Baugeschichte]
Für die Beschreibung von Materialien und Baugeschichte wurden im MonArch-Projekt unter anderem folgende Themen definiert (siehe Abbildung 2.3).

Quarzit ist ein Thema aus der Themenhierarchie für Materialien. Das Thema *Quarzit* ist dabei ein Unterbegriff des Themas *Naturstein* der wiederum ein Unterbegriff von *Stein* ist. Dies lässt sich weiter fortsetzen.

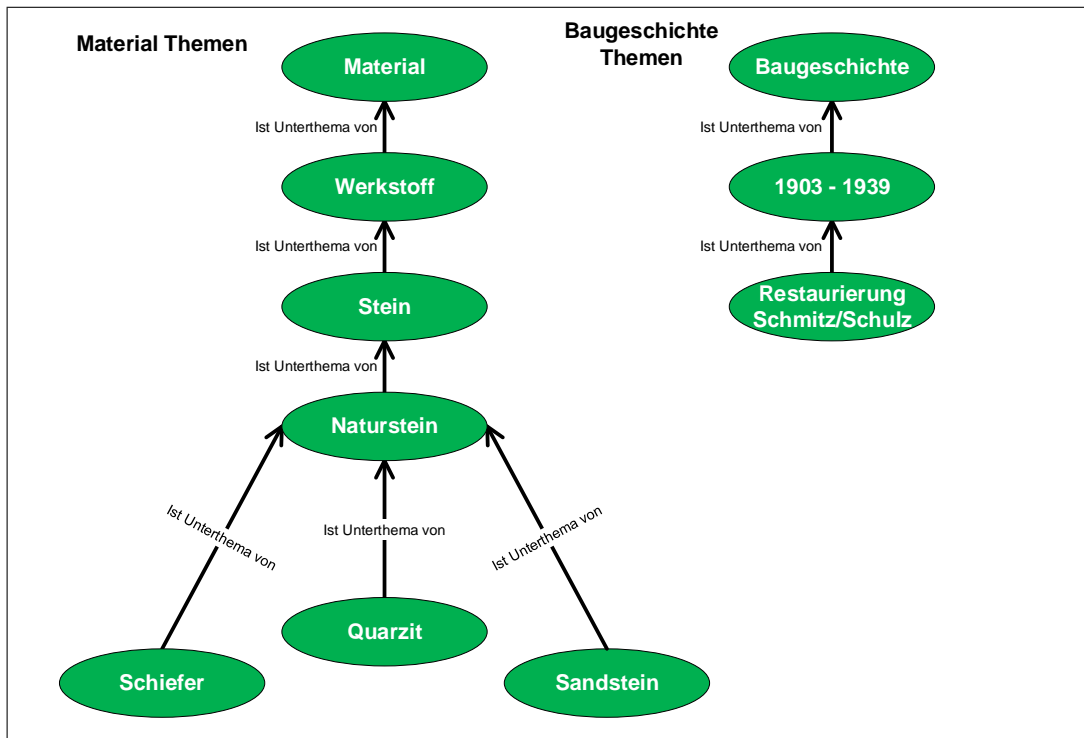


Abbildung 2.3.: Themenhierarchie für die thematische Auszeichnung

□

Betrachtet man die Metadaten der Dokumente, können diese in den meisten Fällen in drei Kategorien aufgeteilt werden:

- Erstens in Metadaten die den räumlichen Bezug zum jeweiligen Gebäude beschreiben.
- Zweitens in Metadaten die den thematischen Inhalt der Dokumente beschreiben.
- Und drittens in Systemmetadaten, die Eigenschaften der Dokumente wie Dateityp oder Auflösung bei Bildern beschreiben.

Folgendes Beispiel 2.6 illustriert die räumliche und thematische Auszeichnung von Dokumenten im MonArch-System.

Beispiel 2.6 [Kartierung der Westfassade]

Bei der vorliegenden Kartierung (Abbildung 2.4) von **St. Lorenz** wurden folgende Metadaten durch die Domänenexperten als relevant erachtet:

- **Nürnberg**
- **St. Lorenz**
- **Westfassade**
- **Auswechslung**
- **Quarzit**
- **1903 bis 1939**

Die ersten drei Daten beschreiben den räumlichen Bezug, da der Plan die **Westfassade** von **St. Lorenz** in **Nürnberg** zeigt. Die letzten drei Daten beschreiben den thematischen Inhalt, nämlich die **Auswechslung** von **Quarzit** im Jahre **1903 bis 1939**. In der Kartierung sind die Steine die ausgewechselt wurden rot gezeichnet.

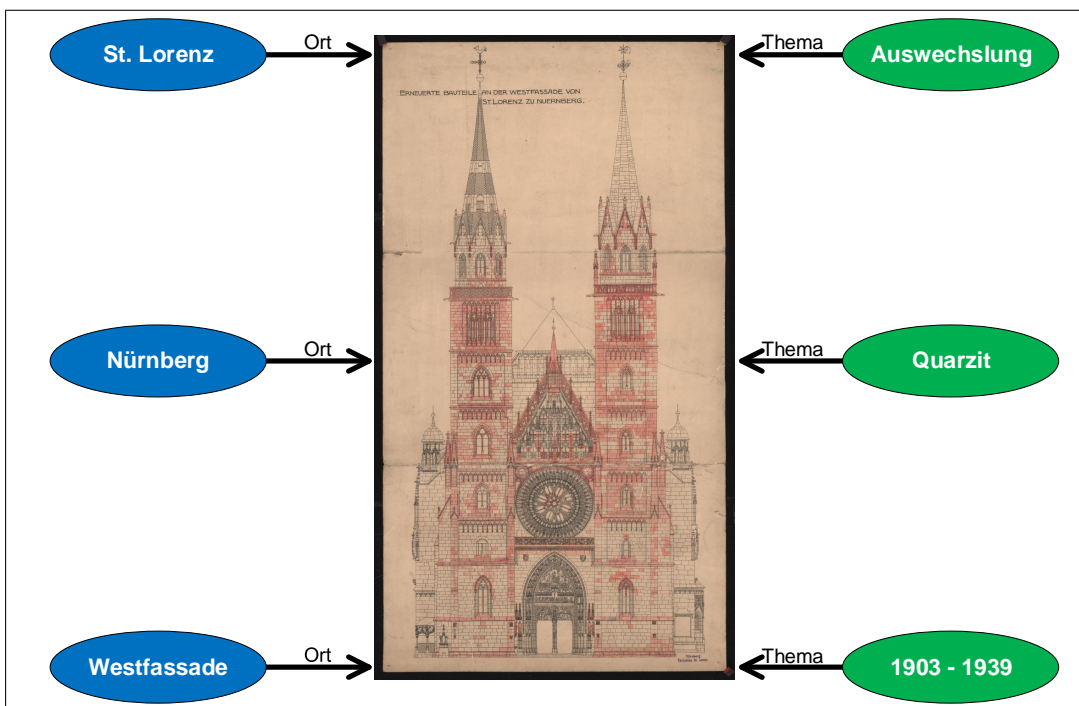


Abbildung 2.4.: Kartierung der Westfassade von St. Lorenz in Nürnberg mit räumlichen und thematischen Metadaten

□

Betrachtet man die Arbeitsweise des vorher beschriebenen Benutzerkreises, stellte sich im Rahmen des MonArch-Projektes heraus, dass sich passend zu den Metadaten der Archivalien die Fragestellungen der Anwender in vielen Fällen in räumliche oder thematische Fragestellungen aufteilen lassen.

Eine räumliche Fragestellung ist in diesem Zusammenhang eine Anfrage, deren Ziel es ist Dokumente zu finden, deren Inhalte einen räumlichen Bezug zu einem Gebäude oder Gebäudeteil aufweisen. Eine thematische Fragestellung hingegen hat das Ziel, Dokumente zu finden, deren Inhalte die angefragten Themen abdecken.

Die MonArch-Methodik stellt sowohl Mechanismen zur Speicherung von Dokumenten und deren Metadaten bereit als auch die Möglichkeit Anfragen zu beantworten.

Die Aufteilung des MonArch-Informationsmodells in drei Komponenten:

- eine räumliche Komponente zur Repräsentation des Gebäudes,
- eine thematische Komponente zur Modellierung von Themen,
- eine Komponente zur Speicherung von Dokumenten,

findet sich im MonArch-Datenmodell und den MonArch-Anfragen wieder, aber auch in der Oberfläche der MonArch-Anwendung (siehe Abbildung 2.5).

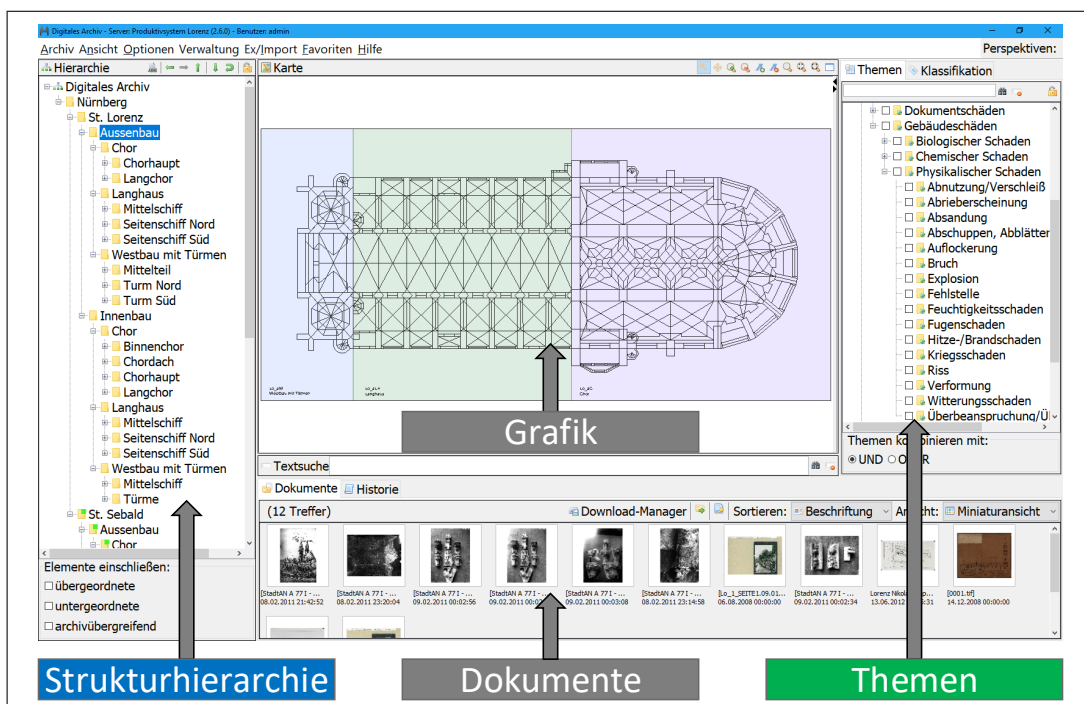


Abbildung 2.5.: Oberfläche der MonArch-Anwendung

Auf der linken Seite der Anwendung sieht man die Strukturhierarchie von St. Lorenz. Auf der rechten Seite der Anwendung ist ein Ausschnitt der Themen von St. Lorenz sichtbar. Im

unteren Bereich der Anwendung werden Dokumente angezeigt, die Ergebnis einer MonArch-Anfrage sind. Im Zentrum ist die grafische Repräsentation des Gebäudes zu sehen, hier der Außenbau von St. Lorenz.

Wählt ein Anwender ein Gebäudeteil in der Strukturhierarchie aus oder selektiert er ein Objekt in der 2D-Grafik wird passend dazu der räumliche Kontext auf das ausgewählte Gebäudeteil gesetzt. Als Resultat werden Dokumente im Ergebnisfenster angezeigt, die an diesem Gebäudeteil verortet wurden. Analog verhält es sich bei der Auswahl eines Themas, das auch wieder durch eine Einfach- oder Mehrfachselektion ausgewählt werden kann.

Sowohl der räumliche als auch der thematische Kontext wirken wie Filter auf die Dokumente die in einem MonArch-System gespeichert sind.

Zur Erstellung der Strukturhierarchie und der damit verknüpften grafischen Repräsentationen müssen eine Reihe von notwendigen Einzelschritten (siehe Abbildung 2.6) vom Anwender durchgeführt werden.

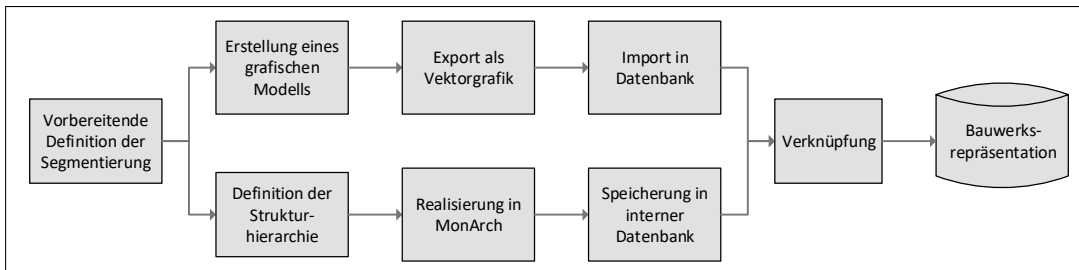


Abbildung 2.6.: MonArch-Arbeitsschritte bei der Erstellung der digitalen Repräsentation des Gebäudes (entnommen aus [SES17])

Als erster Schritt muss die Segmentierung des Gebäudes vorbereitet werden. Für Sakralbauten existieren oft schon Segmentierungen des Gebäudes die mit kleineren Anpassungen übernommen werden können. Die Segmentierung für andere Gebäude folgt teilweise aus deren Konstruktion. Die Erstellung der grafischen Repräsentation und der Strukturhierarchie können parallel ausgeführt werden.

Für die grafische Repräsentation muss zuerst ein grafisches Modell z.B. in einer CAD-Anwendung erstellt werden. Ist das grafische Modell erstellt, kann dieses als Vektorgrafik exportiert werden. Eine Vektorgrafik kann anschließend in das MonArch-System importiert werden.

Ausgehend von der Segmentierung kann die Strukturhierarchie definiert und im MonArch-System realisiert werden. Die Speicherung der Strukturhierarchie erfolgt genau wie die Speicherung der grafischen Repräsentation in der MonArch-Datenbank.

Als letzter Schritt müssen die Objekte der grafischen Repräsentation mit den Gebäudeteilen der Strukturhierarchie manuell verknüpft werden.

2.2. MonArch-Datenmodell

Wie auch das MonArch-Informationsmodell ist das MonArch-Datenmodell in drei Bereiche unterteilt [SWF11b]. Das MonArch-Datenmodell ist als relationales Modell realisiert und daher werden im Folgenden erweiterte Entity-Relationship-Diagramme zur Präsentation des Datenmodells verwendet.

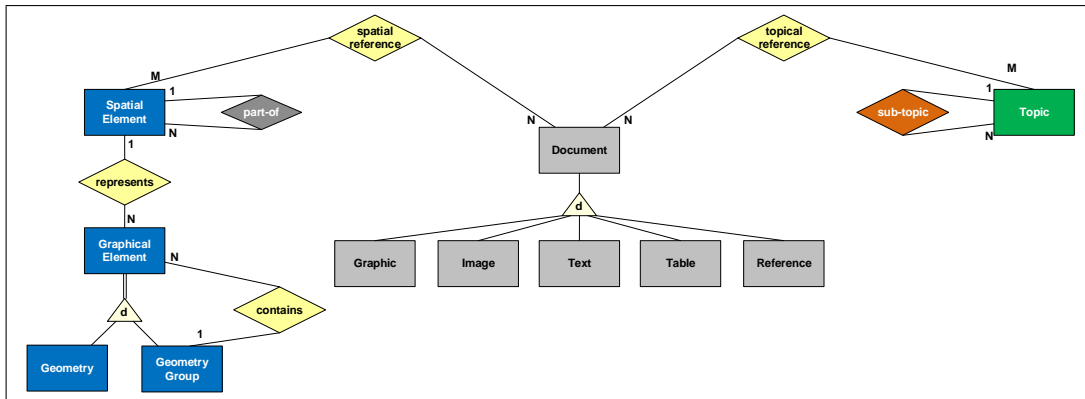


Abbildung 2.7.: ER-Modell des MonArch-Datenmodells

In Abbildung 2.7 ist zentral in der Mitte der Entity-Typ **Document** zu sehen der die gespeicherten Dokumente repräsentiert (siehe Unterabschnitt 2.2.3). Auf der linken Seite des ER-Diagramms sind die Entity-Typen zur Modellierung der Strukturhierarchie und der grafischen Repräsentation des Gebäudes zu sehen (siehe Unterabschnitt 2.2.1). Auf der rechten Seite des ER-Diagramms ist die Themenhierarchie modelliert (siehe Unterabschnitt 2.2.2).

Im weiteren Verlauf gehen wir auf die einzelnen Teilbereiche des MonArch-Datenmodells ein.

2.2.1. Strukturhierarchie und grafische Repräsentation

Das MonArch-Datenmodell ermöglicht eine hierarchische symbolische Zerlegung eines Gebäudes in seine Gebäudeteile und Bereiche. Die Gebäudeteile und Bereiche stehen zueinander in einer Teil-von (part-of) Beziehung. Die daraus resultierende Strukturhierarchie des Gebäudes kann als räumliche Referenz für ein Dokument genutzt werden. Zusätzlich können einem Gebäudeteil eine oder mehrere grafische Repräsentationen zugeordnet werden.

Aus diesen Überlegungen zum räumlichen Aufbau eines Gebäudes lässt sich ein Datenmodell definieren das als Ausschnitt in Abbildung 2.8 zu sehen ist. Dieser Ausschnitt des Datenmodells ist für die Speicherung des räumlichen Bezuges und für die Speicherung der digitalen Repräsentation des Gebäudes verantwortlich.

Im Detail haben die einzelnen Entity-Typen folgende Aufgaben:

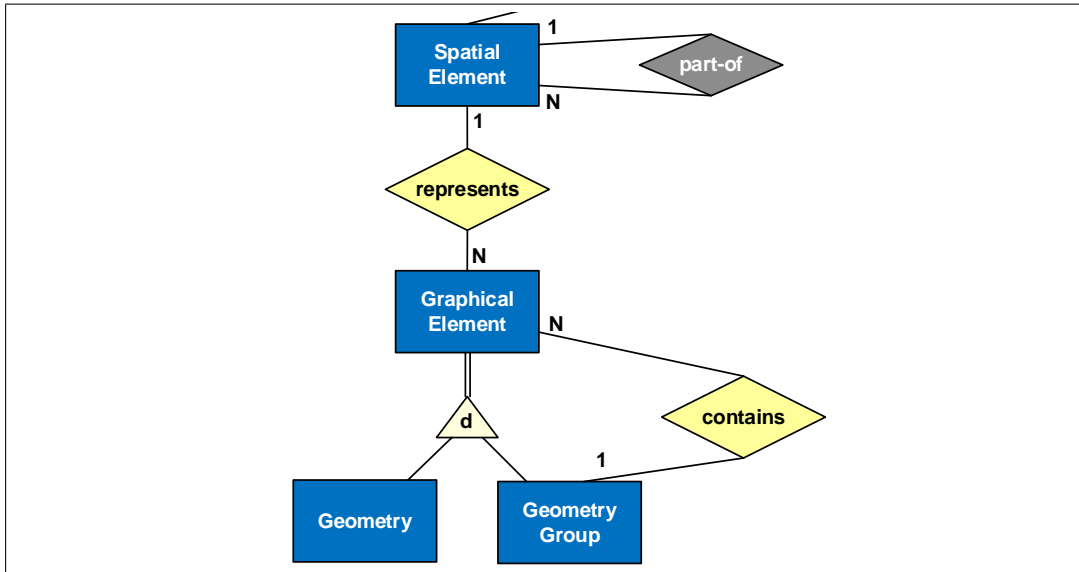


Abbildung 2.8.: ER-Modell des MonArch-Datenmodells (räumlicher Teil)

Spatial Element

Der Entity-Typ **Spatial Element** modelliert Gebäudeteile und Bereiche der Strukturhierarchie. Die einzelnen Gebäudeteile können dabei in einer **part-of** Beziehung stehen.

Das Gebäudeteil *Nordturm* ist z.B. ein Teil des übergeordneten Gebäudeteiles *Westbau mit Türmen*.

Geometry

Der Entity-Typ **Geometry** modelliert die Grundelemente, wie z.B. Punkte, Linien und Polygone, einer grafischen Repräsentation. Diese Elemente können aus CAD-Zeichnungen entnommen werden.

Geometry Group

Der Entity-Typ **Geometry Group** ist eine Zusammenfassung von **Geometry** oder **Graphical Element**. Dies erlaubt einerseits grafische Elemente zu Gruppen zusammenzufassen und somit komplexe grafische Elemente aufzubauen. Andererseits kann man diese Gruppen auch als Ebenen, wie sie in CAD-Zeichnungen Verwendung finden, verstehen.

Graphical Element

Der Entity-Typ **Graphical Element** dient als Komponente für die Umsetzung eines composite patterns im Datenmodell. **Geometry Group** ist dabei das Kompositum und **Geometry** das Blatt.

Der Entity-Typ **Graphical Element** stellt zusätzlich die Verknüpfung zwischen Strukturhierarchie und grafischer Repräsentation her. Ein **Graphical Element** kann dabei

nur ein **Spatial Element** repräsentieren. Diese Einschränkung gewährleistet die Eindeutigkeit der Verknüpfung zwischen grafischer Repräsentation zu Gebäudeteilen.

Ein **Graphical Element**, sei es nun ein Punkt, eine Linie, ein Polygon oder eine Gruppe von grafischen Elemente in einer Zeichnung, ist einem Gebäudeteil der Strukturhierarchie durch die **represents**-Beziehung eindeutig zuzuordnen. Diese Eigenschaft ist für räumliche grafische Anfragen, die im nächsten Abschnitt genauer beschrieben werden, wichtig.

Ein **Spatial Element** kann durch mehrere **Graphical Element** repräsentiert werden. Dies wiederum erlaubt es ein Gebäudeteil durch verschiedene grafische Elemente, beispielsweise in verschiedenen Detailstufen oder verschiedenen Zeichnungen, zu repräsentieren.

Das Gebäudeteil *Nordturm* kann auf einem Grundriss z.B. nur durch ein Rechteck repräsentiert werden. In einer Detail-Seitenansicht hingegen kann der *Nordturm* zusätzlich von einer ganzen Gruppe von grafischen Elementen repräsentiert werden. Das Gebäudeteil *Nordturm* hat in diesem Beispiel zwei grafische Repräsentation. Eine grafische Repräsentation z.B. ein Rechteck kann nicht gleichzeitig den *Nord-* und *Südturm* repräsentieren.

2.2.2. Themenhierarchie

Für die thematische Auszeichnung der Dokumente ermöglicht das MonArch-Datenmodell die Definition und Speicherung einer Themenhierarchie. Dabei repräsentieren die Knoten der Themenhierarchie die Themen mit denen die Dokumente ausgezeichnet werden können. Zwischen den Themen können im MonArch-Datenmodell Beziehungen definiert werden die eine semantische Verwandtschaft von Ober- und Unterbegriffen darstellen.

Die Modellierung der Themen als Themenhierarchie erlaubt einerseits die Festlegung eines Vokabulars für die Auszeichnung der Dokumente mit thematischen Metadaten, andererseits eine Verallgemeinerung bzw. Spezialisierung der Suche entlang der Ober- und Unterbegriffe der Themenhierarchie.

Die beschriebenen Überlegungen und Anforderungen an die Speicherung der Themenhierarchie werden im thematischen Teil des MonArch-Datenmodells umgesetzt. Abbildung 2.9 zeigt den Ausschnitt des Datenmodells, welcher die Speicherung ermöglicht.

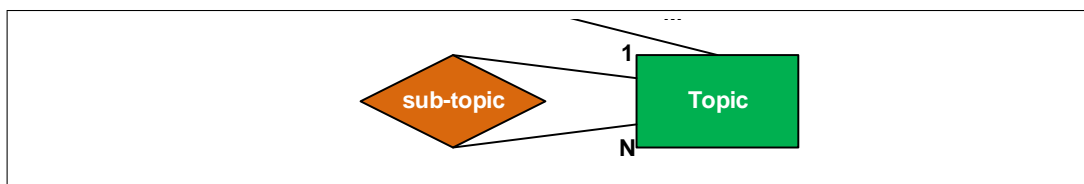


Abbildung 2.9.: ER-Modell des MonArch-Datenmodells (thematischer Teil)

Im Detail hat der Entity-Typ folgende Aufgaben:

Topic

Der Entity-Typ **Topic** modelliert die Themen der Themenhierarchie. Zwischen den Themen kann eine Beziehung modelliert werden auf deren Basis die Ober- und Unterbegriffshierarchie aufgebaut werden kann. Dabei nimmt ein Thema einmal die Rolle des Oberbegriffs und einmal die Rolle des Unterbegriff ein.

Naturstein ist beispielsweise breiter gefasst als *Quarzit*.

2.2.3. Dokumente

Im MonArch-Datenmodell spielt das Dokument bzw. die Datei eine zentrale Rolle. Dokumente können räumlich als auch thematisch ausgezeichnet werden. Das folgende ER-Diagramm (siehe Abbildung 2.10) zeigt die Speicherung der Dokumente und die Beziehungen zur Strukturhierarchie und Themenhierarchie.

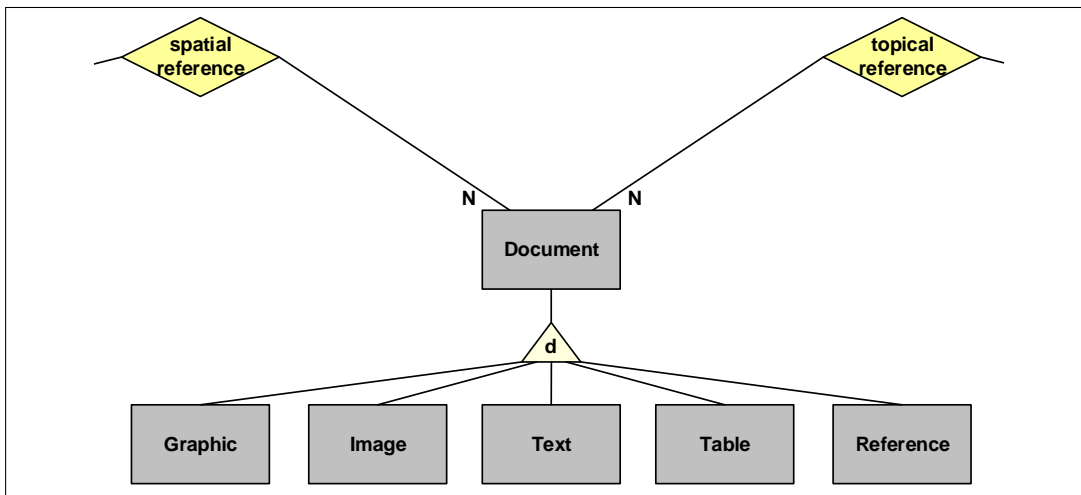


Abbildung 2.10.: ER-Modell des MonArch-Datenmodells (Dokumente)

Im Detail übernehmen die Entity-Typen folgende Aufgaben:

Document

Der Entity-Typ **Document** modelliert das Dokument im MonArch-Datenmodell. Ein Dokument kann an einem oder auch mehreren Gebäudeteilen verortet werden. Zusätzlich wird ein Dokument mit Themen aus der Themenhierarchie ausgezeichnet. Je nach Dokumenttyp, z.B. Bilddokument, CAD-Zeichnung oder Textdokument, wird der Entity-Typ **Document** verfeinert.

Graphic

Der Entity-Typ **Graphic** repräsentiert CAD-Zeichnungen oder andere Vektorzeichnungen. Hierbei ist zu unterscheiden, ob CAD-Zeichnungen als Dokument, d.h. als primäre Information, gespeichert werden oder als grafische Repräsentation des Gebäudes, d.h. als sekundäre Information. Im ersten Fall wird es als **Graphic** gespeichert. Im zweiten

Fall wird die CAD-Zeichnung in ihre Elemente **Geometry**, z.B. Polygone oder Punkte, und **Geometry Group**, z.B. Ebenen oder Gruppen, zerlegt und gespeichert. Somit sind CAD-Zeichnungen die einzigen Elemente im MonArch-Datenmodell die sowohl als primäre als auch als sekundäre Informationen auftreten können.

Image, Text, Table

Jede Art von Bildinformation wird als **Image** gespeichert. Textdokumente werden als **Text** gespeichert. Tabellenartige Daten werden als **Table** gespeichert.

Reference

Der Entity-Typ **Reference** repräsentiert Verlinkungen auf externe Dokumente oder allgemein auf Informationen, die nicht im MonArch-Datenmodell repräsentiert oder gespeichert werden. Ein Beispiel sind Webseiten oder Dokumente auf Netzlaufwerken fremder Server.

2.2.4. Metaebenen des Datenmodells

Für eine bessere Einordnung und in Hinblick auf eine Erweiterung dieser Art der Modellierung in den Folgenden Kapiteln wird in Abbildung 2.11 die Aufteilung zwischen Daten und Metadaten visualisiert.

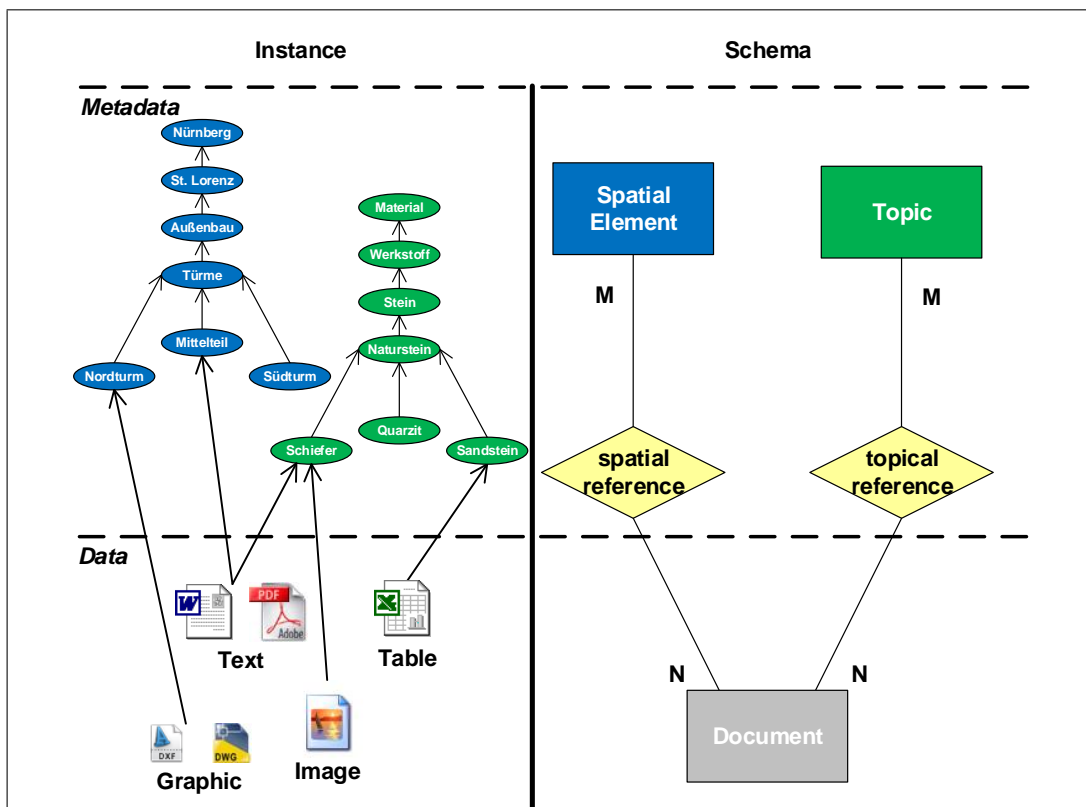


Abbildung 2.11.: Metaebenen im MonArch-Datenmodell

Die **Datenebene** enthält die Dokumente, die in der Datenbank gespeichert werden. Dokumente werden mit Elementen aus der Metadatenebene ausgezeichnet. In der **Metadatenebene** befinden sich die Strukturhierarchie und die Themenhierarchie.

Das MonArch-Datenmodell, welches hier vorgestellt wurde, erlaubt sowohl eine mehrdimensionale Auszeichnung von Dokumenten mit räumlichen als auch mit thematischen Metadaten. Es dient als Grundlage für die im MonArch-System relevanten Anfragen, die im Folgenden Abschnitt 2.4 vorgestellt werden. Bevor auf die Anfrageszenarien genauer eingegangen wird, stellt der nächste Abschnitt 2.3 zwei reale Datenbestände der digitalen MonArch-Archive St. Lorenz und St. Sebald vor. Diese beiden Archive werden im Weiteren immer wieder als Beispiele herangezogen und dienen als primäres Anwendungsszenario.

2.3. MonArch-Archive St. Lorenz - St. Sebald

Im Rahmen des MonArch-Projektes entstanden Instanzen des MonArch-Datenmodells, die im Folgenden als MonArch-Archive (siehe Definition 2.7) bezeichnet werden. Viele Kathedralen und Kirchen, Profanbauten und archäologische Denkmäler setzen das MonArch-System zur Dokumentation ihrer Bauwerke ein, unter ihnen bekannte UNESCO Weltkulturerbestätten. Beispiele dafür sind der Dom St. Stephan in Passau, die Großkirchen St. Lorenz und St. Sebald in Nürnberg [DFS11], die Altstadt von Buchara in Usbekistan[KDS17], die Kaiser- und Barbarathermen in Trier [DT14] sowie die Pfalzenforschung in Aachen[LW14].

Definition 2.7 [MonArch-Archiv]

Ein *MonArch-Archiv* besteht aus einem Server auf dem die Server-Software ausgeführt wird, einem relationalen Datenbanksystem, in dem die Metadaten der Dokumente, die Strukturhierarchie und die Themenhierarchie gespeichert werden und aus einem Dateiserver in dem die eigentlichen Dokumente, die Dateien, gespeichert werden. Ein MonArch-Archiv speichert die Daten für mindestens ein Gebäude. □

Die MonArch-Archive St. Lorenz und St. Sebald beinhalten die Strukturhierarchie der beiden Großkirchen in Nürnberg.

In Zusammenarbeit mit den DFG-Projektpartnern entstanden zwei Instanzen des MonArch-Datenmodells. Der räumliche Aufbau beider Großkirchen ist sehr komplex weshalb die im Folgenden in Ausschnitten vorgestellte Strukturhierarchie einen Kompromiss zwischen Detailgrad auf der einen und Wartbarkeit auf der anderen Seite darstellt. Dieser Kompromiss hat sich in der Praxis bewährt. Bei den Gebäudeteilen könnte man beispielsweise bis zum einzelnen Stein gehen, was den Detailgrad erheblich erhöhen würde. Jeder Stein müsste in der Strukturhierarchie verortet werden, was zu einem enormen Aufwand für die Erstellung einer solchen Hierarchie führt.

Damit der Aufwand für die Erstellung zum eigentlichen Nutzen in einem vernünftigen Verhältnis steht, entschied man sich im Rahmen des MonArch-Projektes nur für bestimmte Gebäudebereiche eine sehr detaillierte Modellierung der Strukturhierarchie vorzunehmen.

Gleichwohl ist dieser Detailgrad in anderen, hier nicht betrachteten Anwendungen des MonArch Systems, gewählt worden.

In der Regel sind die kleinsten Gebäudeteile in der Strukturhierarchie von St. Lorenz und St. Sebald Fenster, Pfeiler und Fassadenabschnitte. Nur bei besonderen Gebäudeteilen, z.B. speziellen Fenstern, wurde bis hin zu feingranularen Elementen, beispielsweise einzelnen Fenster-elementen, verfeinert.

Eine Besonderheit ist die Aufteilung des Gebäudes in ein Äußeres und ein Inneres. Die Hierarchie teilt sich also in der ersten Ebene in Außenbau und Innenbau auf. Diese Entscheidung ist fachlich begründet und folgt aus der Arbeitsweise der Architekten, Dombaumeister und Wissenschaftler, die meist das Äußere und das Innere eines Gebäudes unabhängig voneinander betrachten.

Das folgende Beispiel führt die hierarchische Gebäudeteilzerlegung von St. Lorenz und St. Sebald ein. Es dient als Grundlage für weitere Beispiele und Anfrageszenarien im weiteren Verlauf dieser Arbeit.

Beispiel 2.8 [Gebäudeteilzerlegung von St. Lorenz und St. Sebald]

In Abbildung 2.12 sind jeweils die ersten drei Ebenen der Strukturhierarchie aus den MonArch-Archiven St. Lorenz und St. Sebald abgebildet.

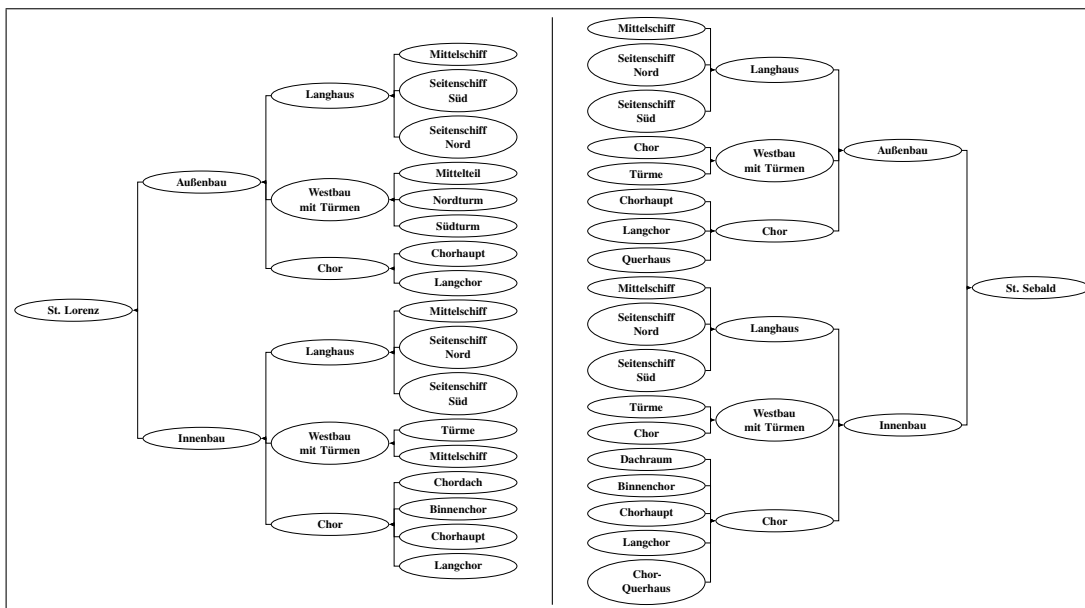


Abbildung 2.12.: Ausschnitt der ersten drei Ebenen der Strukturhierarchie von St. Lorenz und St. Sebald

Bei beiden Archiven ist wie bereits angesprochen in der ersten Ebene zwischen dem Außenbereich und dem Innenbereich des Gebäudes aufgeteilt. Anschließend folgt bei beiden Gebäuden die Aufteilung in Bereiche. Die letzte dargestellte Aufteilung verfeinert diese Bereiche.

2. Anwendungsfall MonArch

In St. Sebald z.B. wird der Innenbereich des **Chors** unter anderem in **Binnenchor** und **Chorquerhaus** aufgeteilt. In St. Lorenz hingegen existiert im Innenbereich des **Chors** kein **Chorquerhaus**.

Obwohl sich beide Großkirchen vom Aufbau sehr ähneln, gibt es Unterschiede in der hierarchischen Gebäudeteilzerlegung, die sich einerseits durch unterschiedliche räumliche Gegebenheiten erklären lassen, andererseits durch eingebürgerte, historisch bedingte Benennungsschemata oder durch die Sicht des Erstellers der Strukturhierarchie auf das Gebäude. □

Neben der Strukturhierarchie der Gebäude gehört zum MonArch-Datenmodell auch eine Themenhierarchie zur thematischen Auszeichnung der Dokumente. Hierbei wurde für beide Gebäude eine einheitliche Themenhierarchie entwickelt.

Im MonArch-Projekt war es das Ziel möglichst eine Themenhierarchie zu definieren, die Themen abdeckt, um Schäden und Reparaturen am Gebäude und an den analogen Dokumenten in den Archiven selbst zu beschreiben. Zeitgeschichtliche Informationen sowie Ersteller und Besitzer der Archivalien können ebenfalls durch Themen beschrieben werden. Abbildung 2.13 zeigt die oberste Hierarchieebene der Themenhierarchie und die zweite Hierarchieebene der Materialien die im Rahmen des MonArch-Projektes definiert wurde.

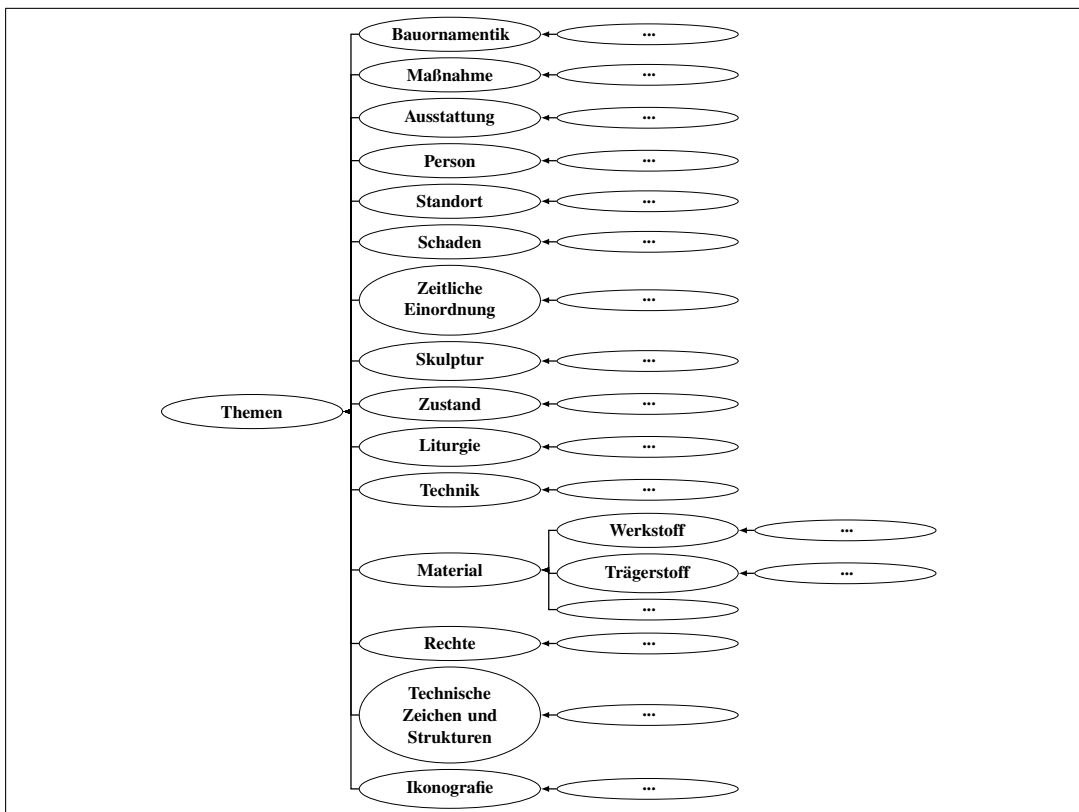


Abbildung 2.13.: Ausschnitt der ersten Themenebene von St. Lorenz und St. Sebald

Eine kurze Beschreibung welche Bedeutung die einzelnen Themen der obersten Hierarchieebene haben ist im Anhang A in Beispiel A.1 nachzulesen.

Die im Rahmen des MonArch-Projektes entstandene Themenhierarchie erhebt keinen Anspruch auf Vollständigkeit. Sie ist entstanden, um die digitalisierten Archivalien der Archive St. Lorenz und St. Sebald thematisch auszeichnen zu können. Besonderes Augenmerk wurde dabei zunächst auf ein sehr spezialisiertes Vokabular gelegt, das dem einzelnen Bauwerk gerecht wird. Durch die Einführung gemeinsamer allgemeinerer Oberthemen kann jedoch von dem spezifischen Gebäude St. Lorenz und St. Sebald abstrahiert werden. Somit teilen sich beide MonArch-Archive gemeinsame Oberthemen in der Themenhierarchie und spezialisieren, falls erforderlich, die Themen lokal.

Aktuelle Entwicklungen lassen neue Standards entstehen, wie z.B. die GND² - Gemeinsame Normdatei. Diese können die im Rahmen des MonArch-Projektes entwickelte Themenhierarchie ergänzen oder teilweise ersetzen. Dazu muss eine Abbildung zwischen der MonArch-Themenhierarchie und dem jeweiligen Standard gefunden werden, um in einem automatischen Migrationsschritt die Themen zu ersetzen. Zwar ist die Anbindung externer Vokabulare und Datenquellen ein aktueller Forschungsgegenstand, spielt jedoch für die vorliegende Arbeit keine Rolle.

²http://www.dnb.de/DE/Standardisierung/GND/gnd_node.html

2.4. MonArch-Anfragen

Ausgehend von dem in Abschnitt 2.2 vorgestellten Datenmodell lassen sich zwei Arten von MonArch-Anfragen ableiten: Erstens räumliche und zweitens thematische Anfragen. Das Ergebnis ist dabei immer eine Menge von Dokumenten.

2.4.1. Räumliche Anfrage

Im MonArch-System gibt es drei Varianten einer räumlichen Anfrage (siehe Definition 2.9).

Definition 2.9 [Räumliche MonArch-Anfrage]

Eine *räumliche Anfrage* ist eine Suchanfrage nach Dokumenten, die an dem in der Anfrage spezifizierten Gebäudeteil oder Bereich der Strukturhierarchie verortet sind.

Bei räumlichen Anfragen ist im MonArch-System die Selektion von nur einem Gebäudeteil pro Suchanfrage erlaubt. □

Zur Illustration der räumlichen Anfragevarianten dient folgendes Beispiel 2.10 einer Strukturhierarchie mit vier verorteten Dokumenten.

Beispiel 2.10 [Westfassade von St. Lorenz mit Dokumenten]

In diesem Beispiel sind vier Dokumente an den ersten drei Ebenen der Strukturhierarchie von St. Lorenz verortet. Die Zuordnung (siehe Abbildung 2.14) ist wie folgt.

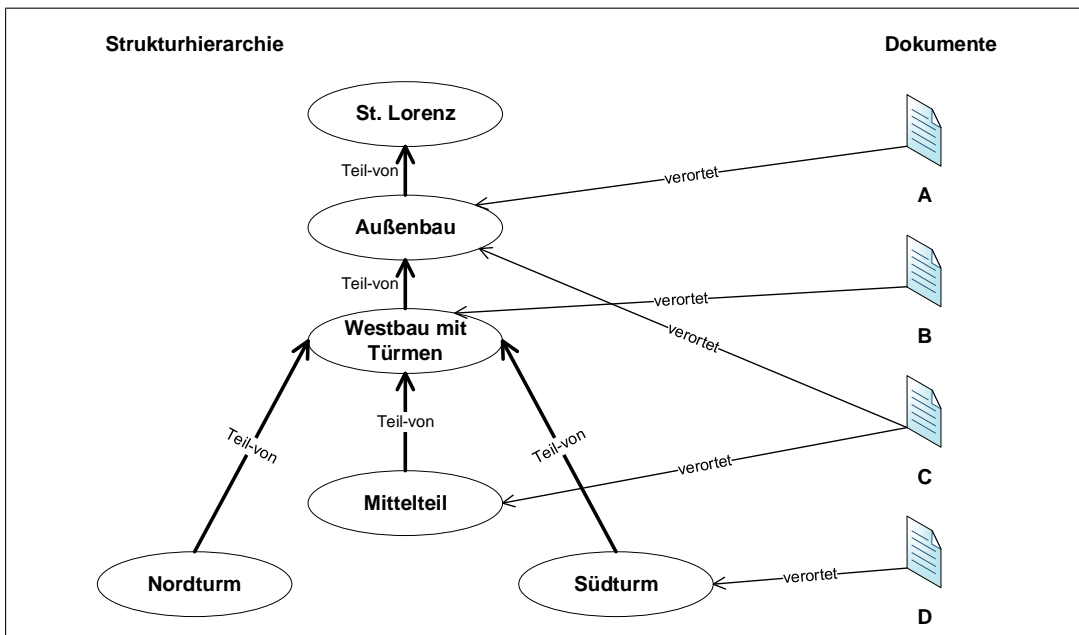


Abbildung 2.14.: Vier Dokumente verortet an der Westfassade von St. Lorenz

Dokument A ist am Außenbau, Dokument B am Westbau mit Türmen, Dokument C ist sowohl am Außenbau als auch am Mittelteil und Dokument D am Südturm verortet. □

2.4.1.1. Räumliche Anfrage - Variante eins

Die erste Variante betrachtet nur Dokumente, die direkt mit dem in der Anfrage spezifizierten Gebäudeteil oder Segment verknüpft sind.

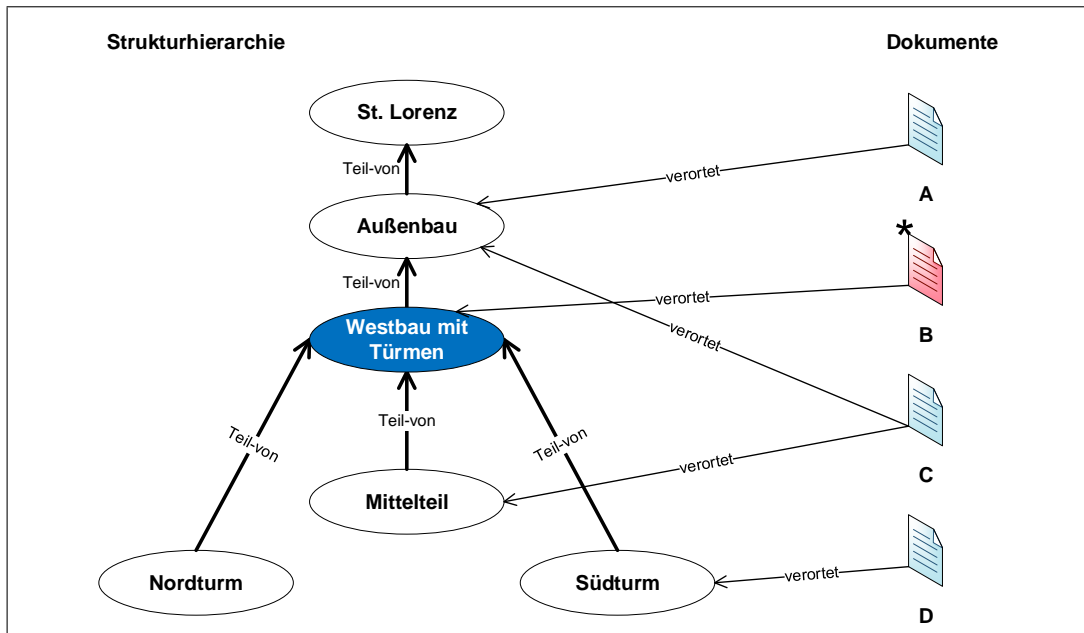


Abbildung 2.15.: Räumliche Anfrage Variante eins an der Westfassade von St. Lorenz

Das Ergebnis der Anfrage

„Suche nach Dokumenten, die direkt mit dem Segment „Westbau mit Türmen“ verknüpft sind!“

ist das Dokument B, da dieses direkt mit dem *Westbau mit Türmen* verknüpft ist (siehe Abbildung 2.15).

Sinnvoll ist eine derartige Anfrage, wenn man die räumliche Zugehörigkeit eines Dokuments zu einem Bereich oder Gebäudeteil so genau wie möglich spezifizieren möchte. Auch ist **Variante eins** ein guter Start für eine Anfrageserie, da hier die Dokumentenmenge maximal gefiltert wird und somit die Ergebnismenge meist möglichst klein ausfällt.

Variante eins liefert hier das Dokument B das direkt verortet wurde und unmittelbar Informationen enthält, die mit dem Bereich *Westbau mit Türmen* in Beziehung stehen. Eine Anfrage dieser Variante liefert keine Dokumente die z.B. am *Mittelteil* des Westbaus verortet wurden, obwohl der Mittelteil ein Teil des Segments *Westbau mit Türmen* ist.

2.4.1.2. Räumliche Anfrage - Variante zwei

Die zweite Variante betrachtet zusätzlich zur ersten Variante auch Dokumente die an Gebäudeteilen oder Bereichen verortet sind, die in der Hierarchie unterhalb des ausgewählten Segments liegen, d.h. die transitiv Teil-von dem ausgewählten Segment sind.

Sollen auch die Gebäudeteile, die Teil von *Westbau mit Türmen* sind, in die Suche miteinbezogen werden, ist **Variante zwei** besser geeignet. Abbildung 2.16 zeigt **Variante zwei** der räumlichen Anfrage.

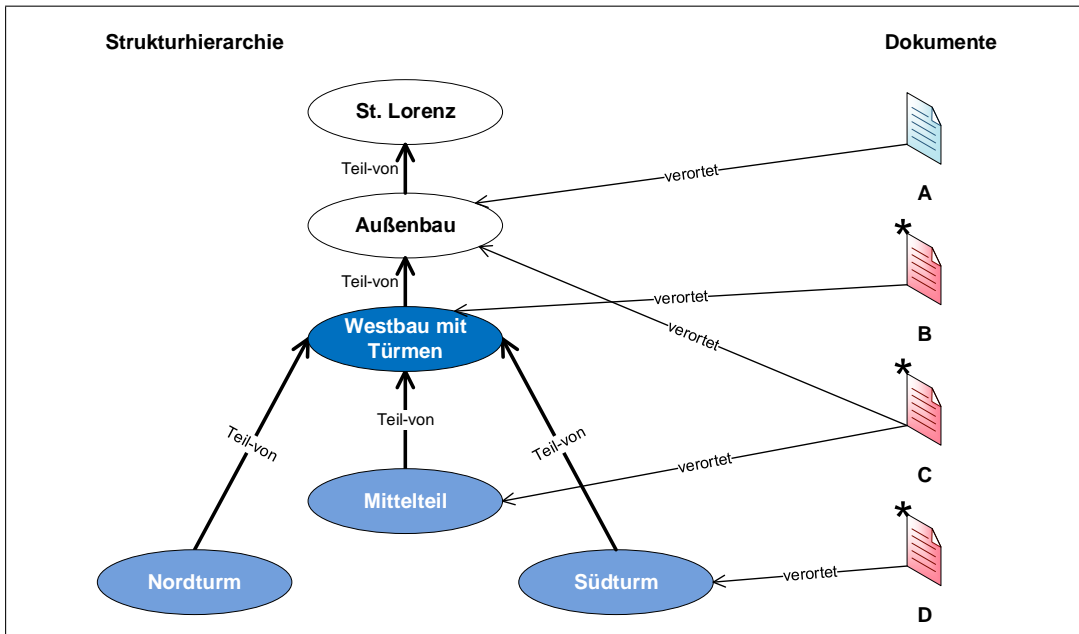


Abbildung 2.16.: Räumliche Anfrage Variante zwei an der Westfassade von St. Lorenz

Das Ergebnis der Anfrage

„Suche nach Dokumenten die direkt mit dem Segment „Westbau mit Türmen“ oder mit Gebäudeteilen aus denen der Westbau besteht verknüpft sind!“

ist wie in **Variante eins** das Dokument B, da dieses direkt mit dem Segment *Westbau mit Türmen* verknüpft ist, und die Dokumente C und D, da diese mit Gebäudeteilen verknüpft sind, die Teil von *Westbau mit Türmen* sind.

Variante zwei der räumlichen Anfrage liefert als Ergebnis alle Dokumente die direkt mit dem Westbau zu tun haben, erweitert aber den Suchraum auf Gebäudeteile die transitiv Teil des Westbaus sind. Somit werden auch Dokumente gefunden die ganz spezielle Informationen zum *Westbau mit Türmen* aufweisen da sie mit einem speziellen Teil des Westbaus verknüpft sind.

Mit dieser Variante wird somit der Suchraum der Anfrage erweitert, was in der Regel zu einer größeren Ergebnismenge führt und die Anfrage verallgemeinert.

2.4.1.3. Räumliche Anfrage - Variante drei

Die dritte Variante betrachtet zusätzlich zur ersten Variante auch Dokumente, die an Gebäudeteilen oder Bereichen verortet sind, die in der Hierarchie oberhalb des gewählten Segments auf einem Pfad zum Wurzelknoten liegen, d.h. von denen das ausgewählte Gebäudeteil oder der ausgewählte Bereich transitiv ein Teil-von ist.

Variante eins und **zwei** decken noch nicht alle Möglichkeiten ab. Betrachten wir das Dokument A, das im Beispiel am *Außenbau* verortet ist. Dieses Dokument A könnte ein Plan des kompletten Außenbaus sein. Da der *Westbau mit Türmen* ein Teil des Außenbaus ist könnte es für die Anfrage relevant sein.

Variante drei der räumlichen Anfrage ermöglicht es genau diesen Fall mit abzudecken. Abbildung 2.17 zeigt **Variante drei** mit dem selektierten Bauteil *Westbau mit Türmen*.

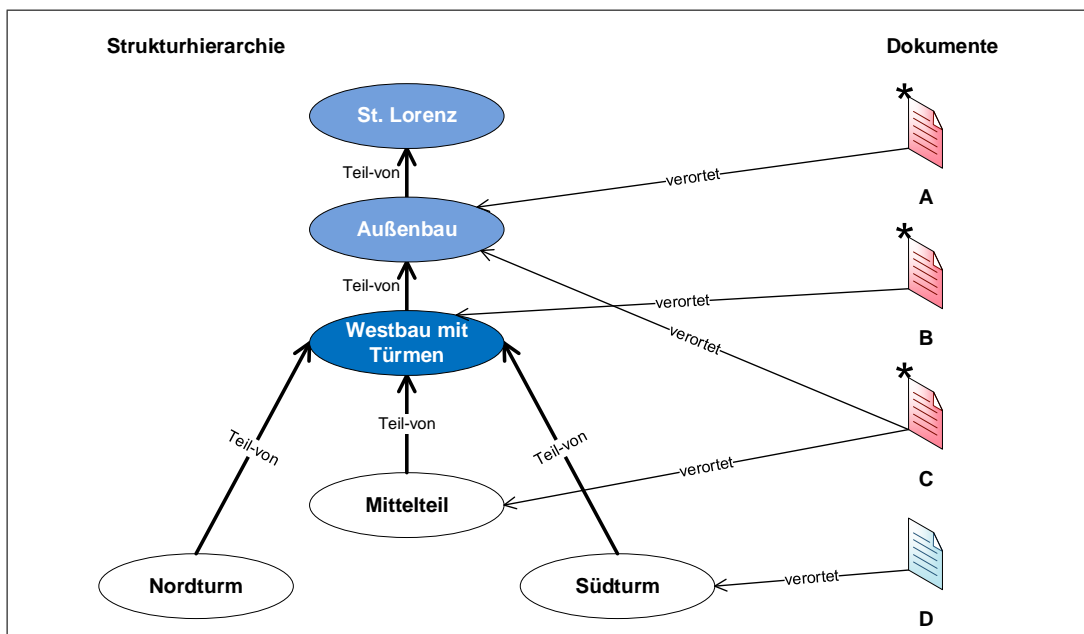


Abbildung 2.17.: Räumliche Anfrage Variante drei an der Westfassade von St. Lorenz

Das Ergebnis der Anfrage

„Suche nach Dokumenten die direkt mit dem Segment „Westbau mit Türmen“ oder Gebäudeteilen, in denen der Westbau als Gebäudeteil vorkommt, verknüpft sind!“

ist, wie in Variante eins, das Dokument B, da dieses direkt mit dem Segment *Westbau mit Türmen* verknüpft ist. Die Anfrage der **Variante drei** betrachtet zusätzlich den *Außenbau* und den Wurzelknoten *St. Lorenz*, mit dem Ergebnis, dass die Dokumente A und C in die Ergebnismenge aufgenommen werden.

Variante drei ist somit auch eine Verallgemeinerung von **Variante eins** lediglich mit einem anderen Fokus. Es werden nicht speziellere Gebäudeteile in die Anfrage miteinbezogen, sondern die Anfrage wird auf Gebäudeteile ausgeweitet, die allgemeiner sind und das selektierte Gebäudeteil transitiv enthalten.

Der MonArch-Anwendungsfall lässt eine Kombination der **Varianten zwei** und **drei** zu. Hierbei werden die Antwortmengen der Anfragen vereinigt.

2.4.1.4. Räumliche Anfrage - Geometrisch

Neben der räumlichen Anfrage über die Strukturhierarchie ermöglicht das MonArch-Datenmodell eine geometrie-basierte Anfrage, die mit den vorgestellten Varianten kombiniert werden kann.

Im MonArch-Datenmodell kann eine N:1 **represents**-Beziehung zwischen einem grafischen Element und einem Gebäudeteil aus der Strukturhierarchie bestehen. Besteht eine solche Beziehung und wird ein derartiges grafisches Element durch den Benutzer selektiert, kann diese Beziehung verwendet werden um den Einstiegspunkt für eine räumliche Anfrage auszuwählen. Ist das Gebäudeteil oder der Bereich ermittelt, mit dem das grafische Element verknüpft ist, kann eine der Varianten oder eine Kombination der beschriebenen räumlichen Anfrage durchgeführt werden.

Die grafische Selektion erfolgt durch die zentrale Kartenansicht des MonArch-Clients (vgl. Abbildung 2.5).

Um diese Art der Anfrage zu veranschaulichen, wird in Beispiel 2.11 eine grafische Anfrage beschrieben.

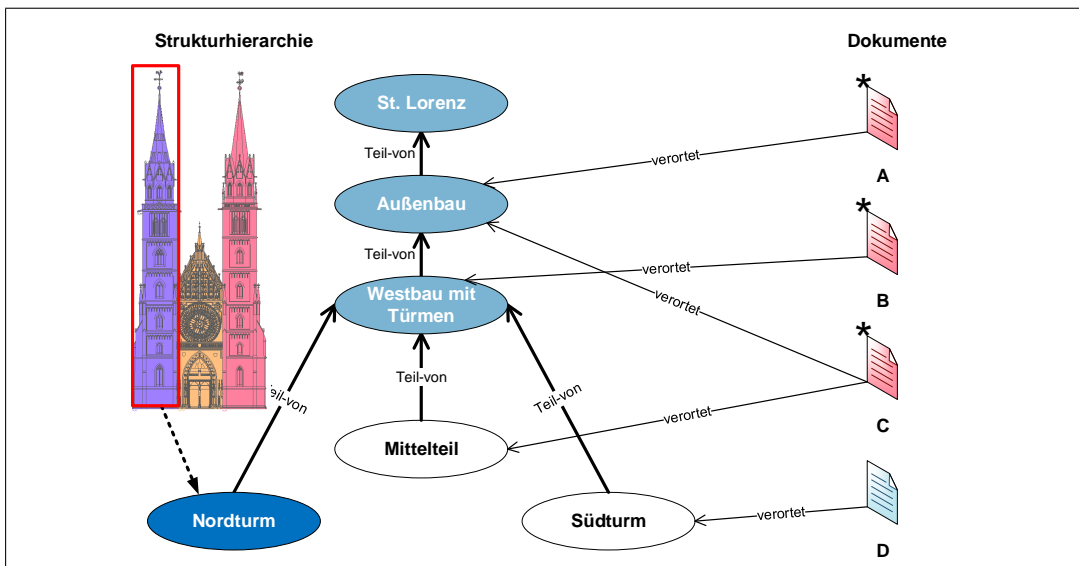


Abbildung 2.18.: Räumliche Anfrage am Nordturm von St. Lorenz durch grafische Selektion

Beispiel 2.11 [Grafische Anfrage an den Nordturm von St. Lorenz]

In Abbildung 2.18 ist die Westansicht von St. Lorenz zu sehen.

Passend dazu ist die Strukturhierarchie mit der grafischen Repräsentation der Westansicht exemplarisch über den Nordturm verknüpft.

Der Benutzer selektiert den Nordturm (roter Kasten) und daraufhin wird die Verknüpfung nachverfolgt und somit das Gebäudeteil **Nordturm** für die eigentliche räumliche Anfrage ausgewählt. □

2.4.2. Thematische Anfrage

Das MonArch-System erlaubt neben den räumlichen Anfragen zusätzlich thematische Anfragen (siehe Definition 2.12).

Definition 2.12 [Thematische MonArch-Anfrage]

Eine *thematische Anfrage* im Zusammenhang mit MonArch ist eine Suchanfrage nach Dokumenten, die mit einem oder mehreren Themen ausgezeichnet sind.

Bei thematischen Anfragen ist eine Mehrfachselektion von Themen möglich. □

In Beispiel 2.13 führen wir einen Ausschnitt aus einer Themenhierarchie für nachfolgende Beispiele ein.

Beispiel 2.13 [Thematische Anfragen in St. Lorenz]

In diesem Beispiel sind vier Dokumente mit Themen aus der Themenhierarchie von St. Lorenz ausgezeichnet.

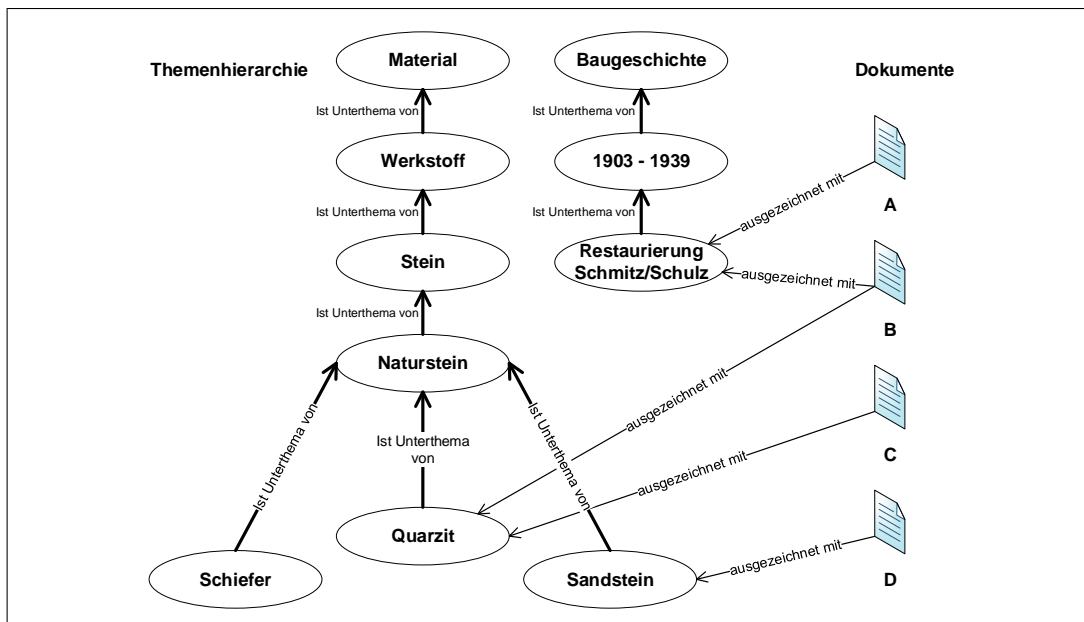


Abbildung 2.19.: Vier Dokumente ausgezeichnet mit Themen von St. Lorenz

Die Zuordnung (siehe Abbildung 2.19) ist wie folgt:

Dokument A ist mit **Restaurierung Schmitz/Schulz**, Dokument B mit **Restaurierung Schmitz/Schulz** und **Quarzit**, Dokument C mit **Quarzit** und Dokument D mit **Sandstein** ausgezeichnet.

Ausgehend von dieser Auszeichnung der vier Dokumente mit Themen werden im Folgenden beide möglichen Anfragevarianten betrachtet. Bei beiden Varianten wird jeweils ein Thema aus der abgebildeten Themenhierarchie ausgewählt. Die Auswahl bei den Materialthemen ist **Naturstein**, bei den Baugeschichtsthemen **Restaurierung Schmitz/Schulz**. □

Das MonArch-System lässt bei einer Mehrfachselektion von Themen eine UND- oder ODER-Verknüpfung der selektierten Themen zu.

Zusätzlich ist ein Dokument in der Ergebnismenge, wenn es mit den ausgewählten Themen oder deren Unterthemen ausgezeichnet ist. Da die Themenhierarchie als Ober- und Unterbegriffshierarchie modelliert ist, können bei der Auswahl eines Themas auch alle seine Unterthemen in der Suche berücksichtigt werden.

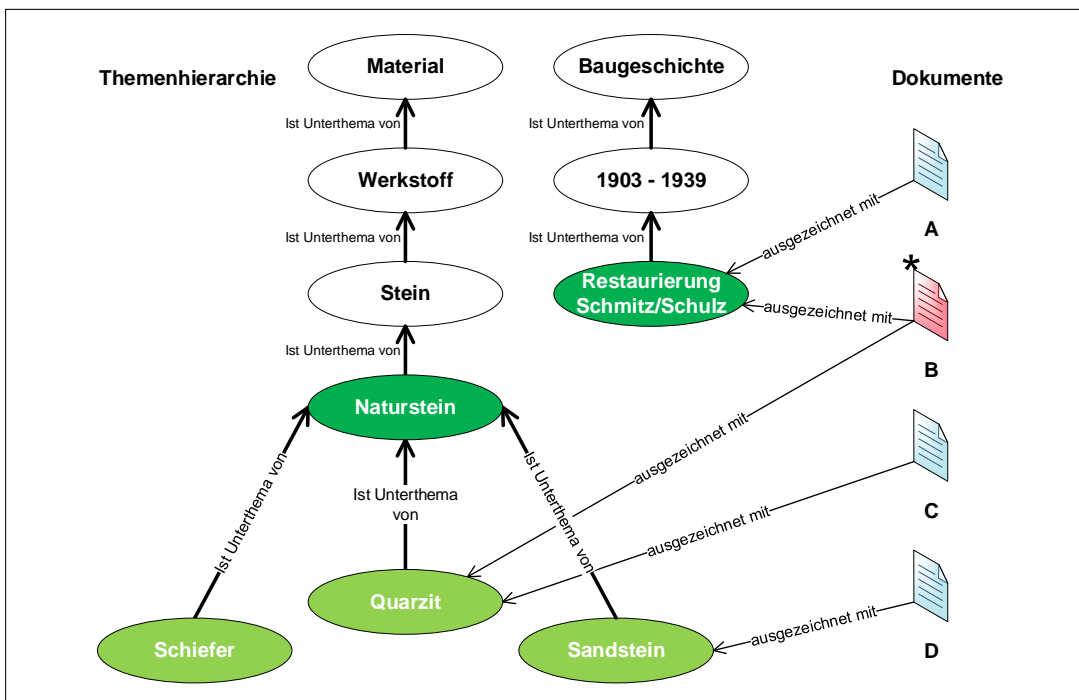


Abbildung 2.20.: Thematische Anfrage Variante eins an St. Lorenz

Die in Abbildung 2.20 gezeigte Anfrage einer UND-Verknüpfung von Themen, bei der die Themen **Naturstein** und **Restaurierung Schmitz/Schulz** selektiert sind, kann wie folgt formuliert werden:

„Suche nach Dokumenten, die Informationen über das Material Naturstein in Zusammenhang mit der Restaurierungsphase von Schmitz und Schulz aufweisen!“

Das Ergebnis der Anfrage ist das Dokument B, da dieses mit dem Thema *Quarzit*, welches ein Unterbegriff von *Naturstein* ist, und zusätzlich mit dem Thema *Restaurierung Schmitz/Schulz* ausgezeichnet ist.

Abbildung 2.21 zeigt dieselbe Selektion wie in Abbildung 2.20 mit dem Unterschied, dass beide selektierten Themen ODER-verknüpft sind.

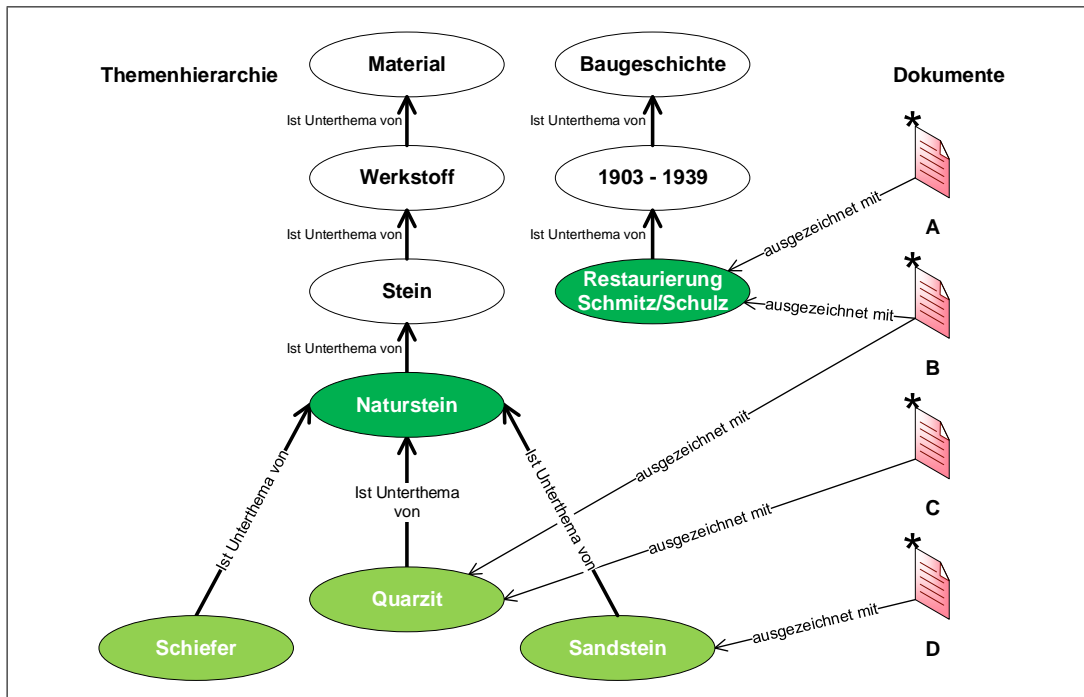


Abbildung 2.21.: Thematische Anfrage Variante zwei an St. Lorenz

Die Anfrage dazu lautet:

„Suche nach Dokumenten, die Informationen über das Material Naturstein oder zur Restaurierungsphase von Schmitz und Schulz aufweisen!“

Das Ergebnis sind alle vier Dokumente, da sie entweder direkt mit einem der Themen aus der Anfrageselektion verknüpft sind oder mit einem Unterbegriff der selektierten Themen ausgezeichnet sind

2.4.3. Anfragekombination

Nachdem die räumliche und thematische Anfrage im MonArch-System im Einzelnen betrachtet wurden, wird im Folgenden die Kombination beider Anfragen genauer beleuchtet.

Es können die Varianten der räumlichen Anfrage beliebig mit einer thematischen Anfrage kombiniert werden. Eine MonArch-Anfrage besteht aus einer räumlichen und einer thematischen Anfrage, wobei deren Ergebnis durch einen Mengenschnitt der Einzelergebnisse

berechnet wird. Durch die Auswahl des Mengenschnitts als Operator für die Anfragekombination der räumlichen und thematischen Anfrage ergibt die kombinierte Anfrage eine stärkere Fokussierung gegenüber den einzelnen Anfragen.

Abschließend wird Beispiel 2.14 für eine Anfragekombination basierend auf den vorangegangenen Beispielen eingeführt.

Beispiel 2.14 [MonArch-Anfragen in St. Lorenz]

Stellen wir uns vor, eine Dombaumeisterin hat einen Schaden im Westbau ihres Doms festgestellt. Sie stellt Informationen zu dem beobachteten Schadensphänomen zusammen, indem sie das MonArch-System für ihre Recherche nutzt.

Sie interessiert sich für alle Dokumente im MonArch-Archiv von St. Lorenz die am Westbau mit Türmen verortet sind. Da die räumliche Einschränkung der Anfrage zu viele Dokumente liefert, schränkt die Dombaumeisterin ihre Anfrage auf das hier relevante Material ein. Zusätzlich ist ihr auch noch bekannt, dass unter den Architekten Schmitz und Schulz Restaurierungen am Gebäude stattgefunden haben.

Daraus ergibt sich eine Anfrage die wie folgt formuliert werden kann:

„Suche nach Dokumenten, die direkt am Westbau mit Türmen oder an Gebäudeteilen aus denen der Westbau besteht verortet sind. Zusätzlich sollen nur Informationen zum Material Naturstein und die Restaurierungsphase von Schmitz und Schultz betrachtet werden!“

Die Anfrage ist eine Kombination der Anfragen in Abbildung 2.16 und in Abbildung 2.21. Das Ergebnis der Anfrage sind somit die Dokumente B, C und D. Das Dokument A ist nicht im Schnitt der beiden Anfragen. □

Die Dombaumeisterin kann die Anfrage z.B. durch Selektion der entsprechenden Elemente in den räumlichen und thematischen Hierarchien realisieren. In Abbildung 2.22 ist die MonArch-Oberfläche der Webbrowserversion zu sehen mit der entsprechenden Auswahl für die in Beispiel 2.14 beschriebene Anfrage.

Die räumliche Selektion des Gebäudeteils oder Bereiches und damit die Einschränkung der Anfrage auf diesen Bereich ist in der linken Baumdarstellung zu sehen.

Die thematische Selektion der Themen und damit die Einschränkung der Anfrage auf Dokumente, die mit den Themen ausgezeichnet sind, ist in der rechten Baumdarstellung zu sehen.

Die Mitte zeigt die zu der momentanen räumlichen Selektion passende zweidimensionale Darstellung des Gebäudes, hier St. Lorenz.

Im unteren Bereich der Anwendung werden die Dokumente angezeigt, die als Ergebnis der Anfrage gefunden wurden.

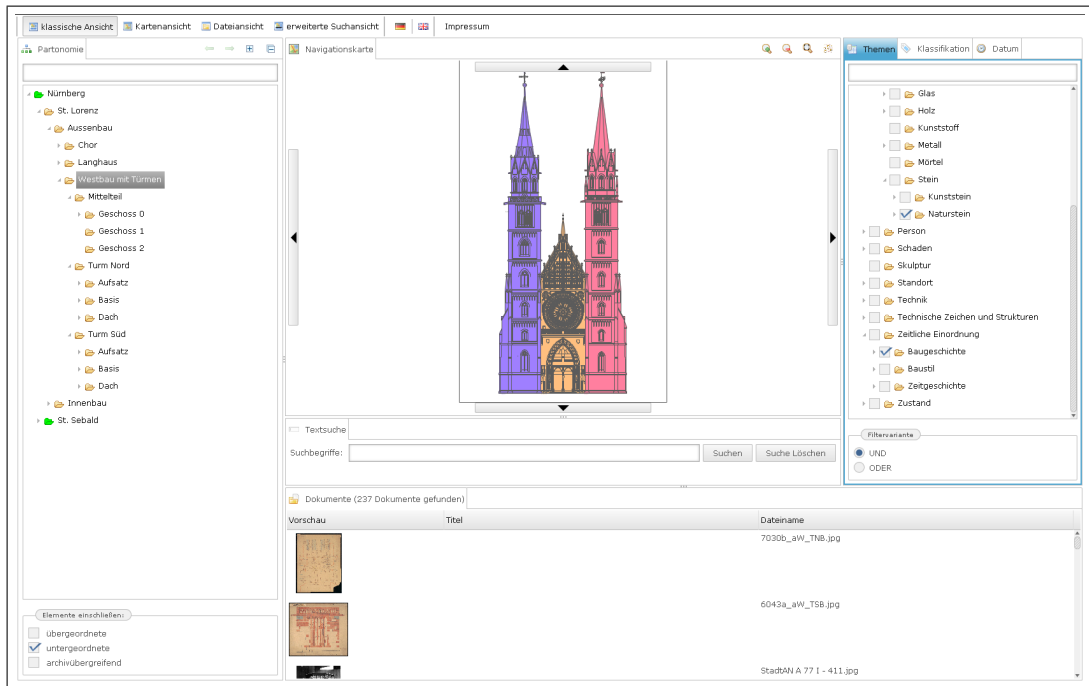


Abbildung 2.22.: MonArch-Client mit Anfrage St. Lorenz

2.5. MonArch-Verbund

Ein Teilziel des MonArch-Projektes war es für die konventionellen Archive von St. Lorenz und St. Sebald in Nürnberg jeweils ein MonArch-Archiv zu erstellen und die digitalisierten Dokumente darin zu speichern. Somit entstand ein digitales Archiv für die Großkirche St. Lorenz und ein digitales Archiv für die Großkirche St. Sebald.

Beide MonArch-Archive erlauben es Dokumente entlang der Strukturhierarchie des jeweiligen Gebäudes zu verorten und thematisch mit Themen auszuzeichnen. In jedem MonArch-Archiv kann der Benutzer die in Abschnitt 2.4 beschriebenen Anfragen stellen und den Dokumentenbestand durchsuchen.

Die MonArch-Methodik ist aber nicht nur auf ein Archiv, respektive auf ein Gebäude, beschränkt. Einzelne MonArch-Archive können zu einem Verbund von MonArch-Archiven (siehe Abbildung 2.23) zusammengeschlossen werden. Dies wird als MonArch-Verbund (siehe Definition 2.15) bezeichnet.

Definition 2.15 [MonArch-Verbund]

Ein *MonArch-Verbund* vernetzt einzelne MonArch-Archive mit folgenden Eigenschaften:

- ein MonArch-Verbund besteht aus mindestens zwei MonArch-Archiven,
- es gibt keine zentrale Verwaltungsstelle im Verbund,
- jedes MonArch-Archiv ist gleichberechtigter Partner im Verbund,

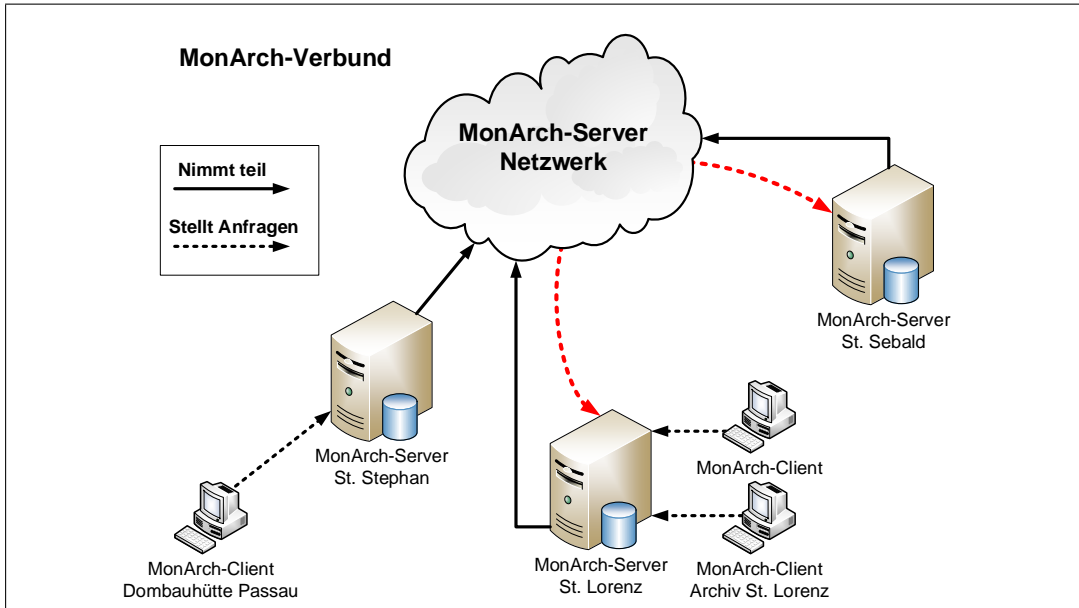


Abbildung 2.23.: MonArch-Verbund

- jedes MonArch-Archiv bestimmt welche Daten bzw. Metadaten im Verbund für andere Teilnehmer sichtbar sind,
- es gibt keine zentrale Benutzerverwaltung,
- jedes MonArch-Archiv verwaltet seine Clients und Benutzer selbst,
- jedes MonArch-Archiv kann die in Abschnitt 2.4 definierten Anfragevarianten an ein anderes MonArch-Archiv stellen,
- jedes MonArch-Archiv kann mit anderen MonArch-Archiven paarweise kommunizieren.

□

Clients sind direkt mit den MonArch-Servern verbunden und können auf den Datenbestand des MonArch-Archivs, mit dem sie verbunden sind, zugreifen. Die einzelnen MonArch-Server sind untereinander verbunden, was eine Weiterleitung einer Verbundanfrage (siehe Definition 2.16) von MonArch-Server zu MonArch-Server erlaubt. Auf diese Weise ist indirekt der Zugriff (siehe gestichelte rote Linie in Abbildung 2.23) auf den Datenbestand der anderen MonArch-Archive im Verbund möglich.

Definition 2.16 [Verbundanfrage]

Eine *Verbundanfrage* ist eine Anfrage die an ein oder alle MonArch-Archive im Verbund gestellt wird. Die Anfrage ist dabei eine Kombination von räumlichen und thematischen Teilanfragen.

□

Betrachtet man die Anfragemöglichkeiten der MonArch-Methodik, erkennt man, dass es Arten von Anfragen gibt, die mit der bis jetzt beschriebenen Methodik nicht realisierbar sind:

- Eine SQL-Differenzoperation für MonArch-Anfragen ist nicht realisiert.
- Eine Räumliche Anfrage die ein bestimmtes Gebäudeteil selektiert, kann nicht ohne weiteres in eine Anfrage transformiert werden, die alle Gebäudeteile derselben Art oder derselben Funktion berücksichtigt. Die MonArch-Methodik erlaubt es nur die Anfrage auf ein übergeordnetes und untergeordnetes Gebäudeteil auszudehnen, (siehe Unterabschnitt 2.4.1[Variante 2 und 3]).
- Thematische Anfragen beziehen automatisch Unterbegriffe mit ein. Dieses Verhalten hat sich im MonArch-Anwendungsfall bewährt schränkt aber die Möglichkeiten im Verbund unnötig ein.
- Eine Verbundanfrage kann nur dann gestellt werden, wenn die für die Anfrageverarbeitung nötigen Struktur- und Themenhierarchien für den Verbund freigegeben sind. Sind die Strukturhierarchie und Themenhierarchie eines MonArch-Archivs nicht bekannt, können auch keine Anfragen an dieses Archiv gestellt werden. Zusätzlich zu der Freigabe der Strukturhierarchie und Themenhierarchie für die Anfrage ist es auch notwendig, dass die Person, welche die Anfrage an den Verbund stellt, sich mit der Themenhierarchie und insbesondere mit der Strukturhierarchie des Fremdarchivs auseinandersetzt.
- Eine Anfrage wird nicht automatisch an ein anderes Archiv im Verbund gestellt und dabei automatisch Gebäudeteile selektiert die dem Gebäudeteil in der Ursprungsanfrage entsprechen.

Aus dieser Betrachtung der Anfragemöglichkeiten lassen sich zwei Anforderungen ableiten.

1. Die MonArch-Anfragemöglichkeiten müssen erweitert werden um neu Anfrageszenarien abzudecken die sich aus dem MonArch-Verbund ergeben. Hierzu werden wir im Kapitel 4 ein Anfragesprache MonQL einführen, die die MonArch-Anfragen erweitern und verallgemeinern.
2. Die Eigenschaften des Verbundes und die Unterschiede in Strukturhierarchie und Themenhierarchie schränken Verbundanfragen in der bis hierher beschriebenen Form ein. Diesen Einschränkungen gilt es entgegenzuwirken. Hierzu werden wir das MonArch-Datenmodell in Kapitel 3 erweitern und die Strukturhierarchie mit Typinformationen anreichern. Die Auszeichnung von Gebäudeteilen mit einem Typ, also die Gruppierung von gleichartigen Teilen, wir sich als günstig für die Aufhebung einiger Einschränkungen zeigen. Für die Themenhierarchie werden wir teilweise auf bestehende Verfahren zurückgreifen und diese an das MonArch-Szenario anpassen. Die Verfahren dazu werden wir in Kapitel 5 präsentieren.

Aus den Anforderungen ergeben sich verschiedener Fragestellungen die sich unmittelbar stellen und in den folgenden Kapiteln beantwortet werden:

- Wie können Strukturhierarchien, die sich von Gebäude zu Gebäude unterscheiden miteinander in Beziehung gesetzt werden?

2. Anwendungsfall MonArch

- Wie können Themenhierarchien, die sich von Archiv zu Archiv unterscheiden können miteinander in Beziehung gesetzt werden?
- Welche Verfahren für eine Anfragetransformation existieren?
- Unter welchen Umständen ist eine Anfragetransformation überhaupt möglich?
- Kann eine Anfragetransformation automatisch erfolgen oder ist der Benutzer involviert?

3. Erweitertes Datenmodell

Die räumlichen und thematischen Anfragen in einem MonArch-Verbund, die in den beiden vorherigen Kapiteln vorgestellt wurden, werfen einige Fragen auf, die auf der Grundlage des in Abschnitt 2.2 vorgestellten MonArch-Datenmodells nicht ausreichend beantwortet werden können. Insbesondere bietet das bisherige Datenmodell keine Möglichkeit, Gruppen gleichartiger Gebäudeteile zu definieren, die für eine gezielte De-Fokussierung einer Anfrage genutzt werden könnten (Anfragerelaxation), um Ungleichheiten der jeweiligen Strukturhierarchie zu kompensieren.

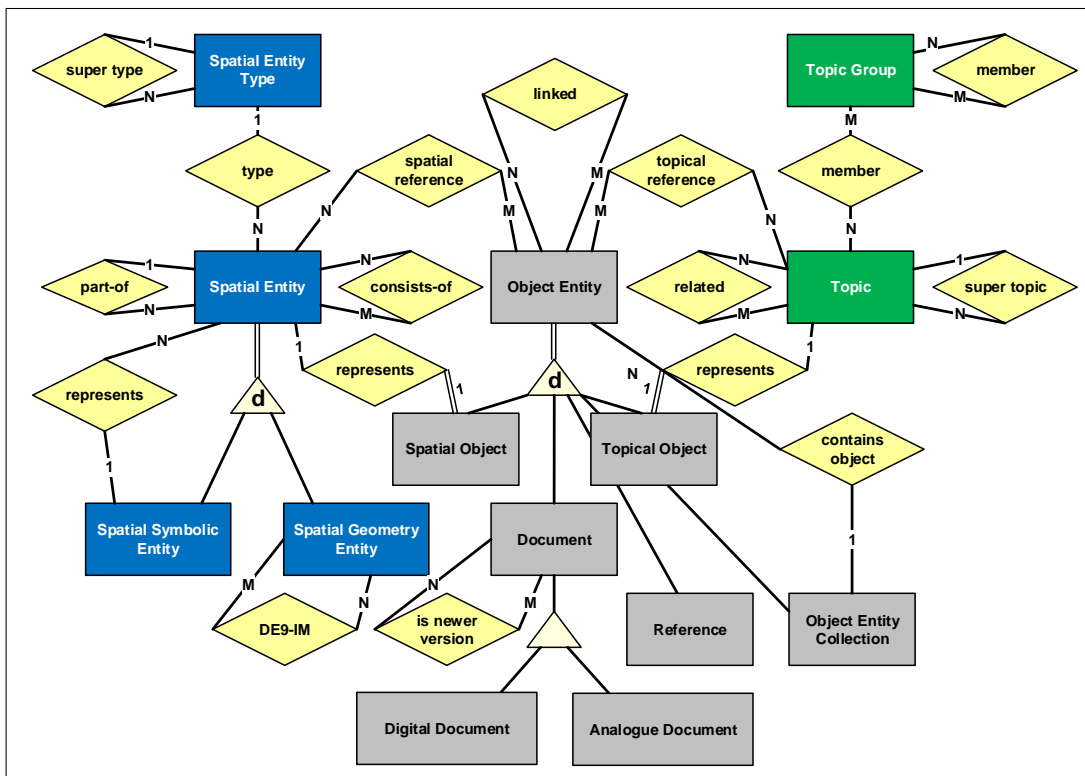


Abbildung 3.1.: Erweitertes Datenmodell als ER-Modell

Ein erster Schritt zur Beantwortung der Fragen ist die Reformulierung und Erweiterung des Datenmodells, um den neuen Anforderungen Rechnung zu tragen. Das erweiterte Datenmodell (siehe Abbildung 3.1) ist ähnlich wie das MonArch-Datenmodell in drei Bereiche aufgeteilt:

- Der räumliche Bereich (linke Seite des ER-Modells) dient zur Modellierung des Strukturgraphen.
- Der thematische Bereich (rechte Seite des ER-Modells) wird zur Modellierung der Themengraphen benötigt.
- Der Objektbereich (Mitte des ER-Modells) stellt Informationen wie z.B. Dokumente und deren Metadaten dar.

Im Folgenden wird auf die Beschreibung von Attributen in den ER-Modellen verzichtet. Jeder Entity-Typ muss durch ein passendes Primärschlüsselattribut identifizierbar sein. Weitere Eigenschaften, wie z.B. ein Name, können bei Bedarf spezifiziert werden. Wir gehen im Folgenden davon aus, dass jeder Entity-Typ zusätzlich Attribute aus dem Dublin Core Metadata Element Set [KB07] hat. Das Dublin Core Metadata Element Set ist als RFC¹ in [Pel17] beschrieben.

3.1. Strukturgraph

Die größte Änderung im erweiterten Datenmodell gegenüber dem MonArch-Datenmodell ist der Übergang von einer Strukturhierarchie zu einem Strukturgraphen (siehe Definition 3.1).

Definition 3.1 [Strukturgraph]

Der *Strukturgraph* erweitert die Strukturhierarchie um eine zusätzliche N:M-Beziehung zwischen Gebäudeteilen. Neben der Teil-von-Beziehung der Strukturhierarchie können Gebäudeteile zusätzlich in einer Besteht-aus-Beziehung stehen. Der Strukturgraph ist azyklisch. □

Das erweiterte Datenmodell verallgemeinert dazu Gebäudeteile zu räumlichen Elementen (**Spatial Entity**). Diese werden disjunkt in symbolische räumliche Elemente (**Spatial Symbolic Entity**) und geometrische räumliche Elemente (**Spatial Geometry Entity**) unterschieden. Räumliche Elemente können mit räumlichen Typen (**Spatial Entity Type**) typisiert werden. Geometrische räumliche Elemente stehen in geometrische Beziehungen (siehe Definition 3.2) zueinander.

Definition 3.2 [DE-9IM]

Das *Dimensionally Extended Nine-Intersection Model (DE-9IM)* ist ein topologisches Modell, welches räumliche Zustände zwischen Geometrien im Zweidimensionalen beschreibt.

Es geht auf die Arbeit von [CFO93; CSE94] und auf die Grundlagen von [EH90; EF91] zurück. Es bildet die Grundlage für OpenGIS kompatible Datenbanksysteme [Her10] und Anwendungen.

Für das erweiterte Datenmodell sind die zehn räumlichen Zustände, die für zwei Geometrien A und B benannt werden können, ausreichend:

- A equals B: A ist in allen Punkten wie B.

¹Requests for Comments

- A disjoint B: A und B haben keine Punkte gemeinsam.
- A touches B: A und B haben mindestens einen Randpunkt gemeinsam.
- A contains B: B liegt in A und das Innere der Geometrien schneidet sich.
- A covers B: Keine Punkte von B liegen im Äußeren von A.
- A intersects B: A und B haben mindestens einen Punkt gemeinsam.
- A within B: A liegt im Inneren von B.
- A coveredby B: Jeder Punkt von A ist ein Punkt von B.
- A crosses B: Sie haben manche aber nicht alle Punkte des Inneren gemeinsam. Die Schnittgeometrie ist in der Dimension kleiner als eine der beiden Geometrien A und B.
- A overlaps B: Sie haben manche aber nicht alle Punkte gemeinsam. A und B haben die gleiche Dimension und die Schnittgeometrie hat die gleiche Dimension wie A und B.

Das eigentliche DE-9IM wurde in [Str07] definiert. □

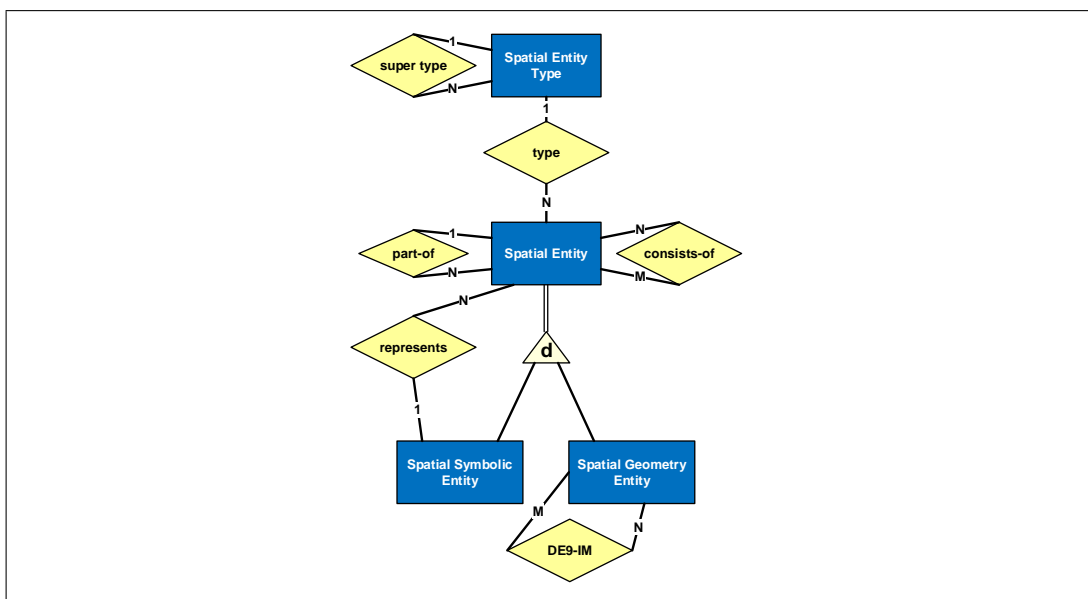


Abbildung 3.2.: Erweitertes Datenmodell (räumlich) als ER-Modell

Abbildung 3.2 zeigt den räumlichen Ausschnitt des erweiterten Datenmodells, dessen Elemente folgende informelle Semantik haben:

Spatial Entity

Der Entity-Typ **Spatial Entity** modelliert Elemente, z.B. Bauteile oder Gebäudeteile, in einer Struktur, z.B. einem Gebäude. Es ist das zentrale Element im Strukturgraphen.

Spatial Entity können dabei in einer 1:N Teil-von-Beziehung (**part-of**) stehen. Dies erlaubt die Abbildung der Strukturhierarchie aus dem MonArch-Datenmodell und stellt damit die Kompatibilität zum MonArch-Datenmodell her.

Die N:M Besteht-aus-Beziehung (**consists-of**) erweitert die Strukturhierarchie zum Strukturgraphen. Dies wurde eingeführt um Gebäudeteile mehr als einem Segment in der Strukturhierarchie zuordnen zu können. Zusätzlich übernimmt sie die Aufgabe die Gruppenbildung der grafischen Elemente des MonArch-Datenmodells über die **contains**-Beziehung abzubilden.

Eine Aufteilung in zwei Arten von Beziehungen wird auch im Bereich des „Building Information Modeling“ (BIM)² und dem BIM-Datenmodell [ISO13] vorgeschlagen.

Beispiel 3.3

*Man stelle sich zwei Räume vor, die durch eine Wand getrennt sind. Im MonArch-Datenmodell müsste die Wand einmal aus Sicht des ersten Raumes und einmal aus Sicht des zweiten Raumes in der Strukturhierarchie abgebildet werden. Im erweiterten Datenmodell kann anstatt der Teil-von-Beziehung jetzt die Beziehung **consists-of** verwendet werden. Eine Wand wird dann nicht mehr doppelt in der Strukturhierarchie repräsentiert, ist jedoch immer noch von jedem Raum erreichbar. Die Zugehörigkeit zu den Räumen geht nicht verloren und die striktere Teil-von-Beziehung kann weiterhin für echte 1:N Beziehungen zwischen Gebäudeteilen verwendet werden. □*

Der Entity-Typ **Spatial Entity** wird total und disjunkt in **Spatial Symbolic Entity** und **Spatial Geometry Entity** spezialisiert.

Spatial Symbolic Entity

Instanzen des Entity-Typs **Spatial Symbolic Entity** stellen Elemente dar, die Gebäudeteile repräsentieren, aber selbst keine grafische Information tragen. Dadurch können materielle Gebäudeteile wie Wände von virtuellen Gebäudeteilen wie dem Raum, den sie umschließen, unterschieden werden. Auch im BIM / IFC Standard ist eine solche Trennung vorgesehen.

Beispiel 3.4

Betrachten wir einen Raum der durch Wände entsteht, die ihn begrenzen. Die Wände könnten geometrische Informationen tragen, z.B. ihre Dicke, Länge und Breite. Auch ihre Position im Gebäude könnte mit Koordinaten beschrieben werden. Der Raum hingegen hat keine solchen geometrischen Informationen. □

Der Relationship-Typ **represents** erlaubt die Verknüpfung einer symbolischen mit mehreren grafischen Repräsentationen.

Spatial Geometry Entity

Der Entity-Typ **Spatial Geometry Entity** modelliert die geometrischen Elemente im Strukturgraphen und somit die grafische Repräsentation eines Gebäudes. Durch die geometrische Natur der Elemente können diese räumlich zueinander in Bezug gesetzt

²<https://www.buildingsmart.de/bim-knowhow/bim-anforderungen>

werden. Der Relationship-Typ **DE9-IM** fasst mögliche Beziehungen zusammen; eine genaue Beschreibung ist in Definition 3.2 eingeführt worden.

Im MonArch-Anwendungsfall werden z.B. Elemente einer Plandarstellung, also Punkte und Linien, als **Spatial Geometry Entity** modelliert.

Spatial Entity Type

Der Entity-Typ **Spatial Entity Type** modelliert Gebäudeteiltypen, oder im Allgemeinen Strukturtypen. Jedem **Spatial Entity** kann ein Typ zugeordnet werden (**type**). Die Typen können in einer Typhierarchie organisiert werden. Der Relationship-Typ **super type** modelliert Ober- bzw. Untertyp in der Typhierarchie im erweiterten Datenmodell.

Zusammengefasst bietet das erweiterte räumliche Datenmodell gegenüber dem MonArch-Datenmodell folgende Vorteile:

- Einführung von Typinformationen von Gebäudeteilen mit Typhierarchie (siehe Beispiel 3.5),
- Möglichkeit der Modellierung eines Strukturgraphen (siehe Beispiel 3.6),
- Gleichstellung der symbolischen und der geometrischen Repräsentation,
- Modellierung von geometrischen Beziehungen zwischen geometrischen Elementen.

Beispiel 3.5 [Strukturhierarchie typisiert - St. Lorenz]

Das erweiterte Datenmodell ermöglicht das Hinzufügen von Typinformation zu einem Strukturgraphen. In Abbildung 3.3 sind ein Ausschnitt aus einer beispielhaften Typhierarchie für Sakralbauten und ein Ausschnitt der Strukturhierarchie von St. Lorenz zu sehen.

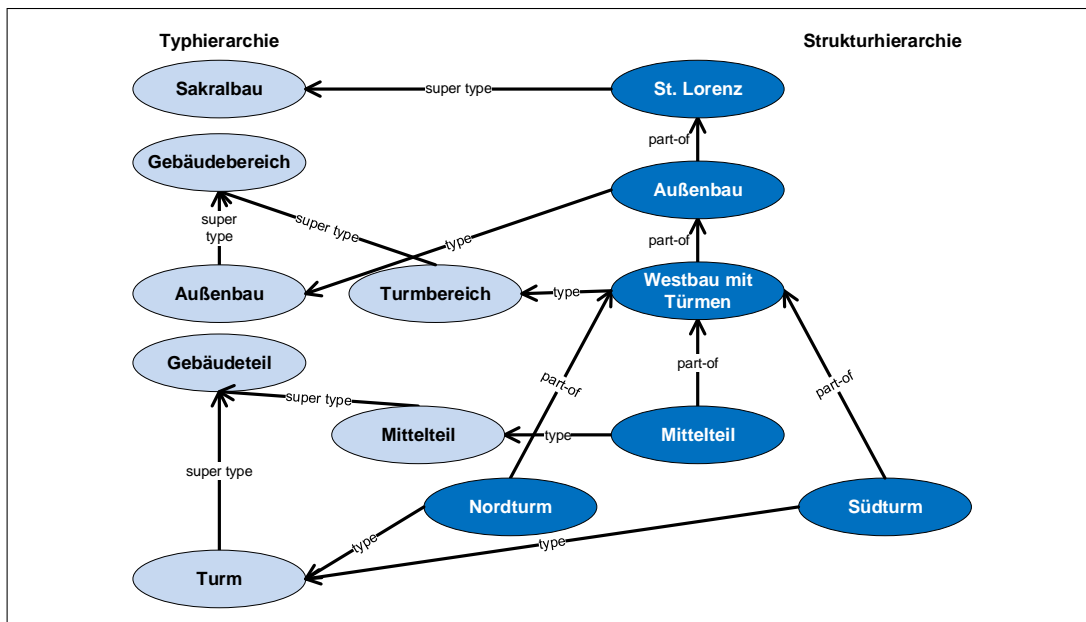


Abbildung 3.3.: Ausschnitt Strukturhierarchie typisiert - St. Lorenz

Die Strukturhierarchie, also die Gebäudeteile *St. Lorenz*, *Außenbau*, *Westbau mit Türmen*, *Mittelteil*, *Nordturm* und *Südturm* werden als *Spatial Symbolic Entity* realisiert und mit *part-of* in Beziehung zueinander gesetzt. Die Strukturtypen werden als *Spatial Entity Type* realisiert. In diesem Ausschnitt wird bei den Typen zwischen *Gebäudeteile* und *Gebäudebereiche* unterschieden. Der *Nordturm* und *Südturm* werden z.B. als *Turm* typisiert. □

Beispiel 3.6 [Strukturgraph - Einfamilienhaus]

Durch die neu eingeführte Beziehung *consists-of* ermöglicht das erweiterte Datenmodell die Definition von Strukturgraphen. In Abbildung 3.3 ist ein Ausschnitt eines Strukturgraphen für ein Einfamilienhaus dargestellt.

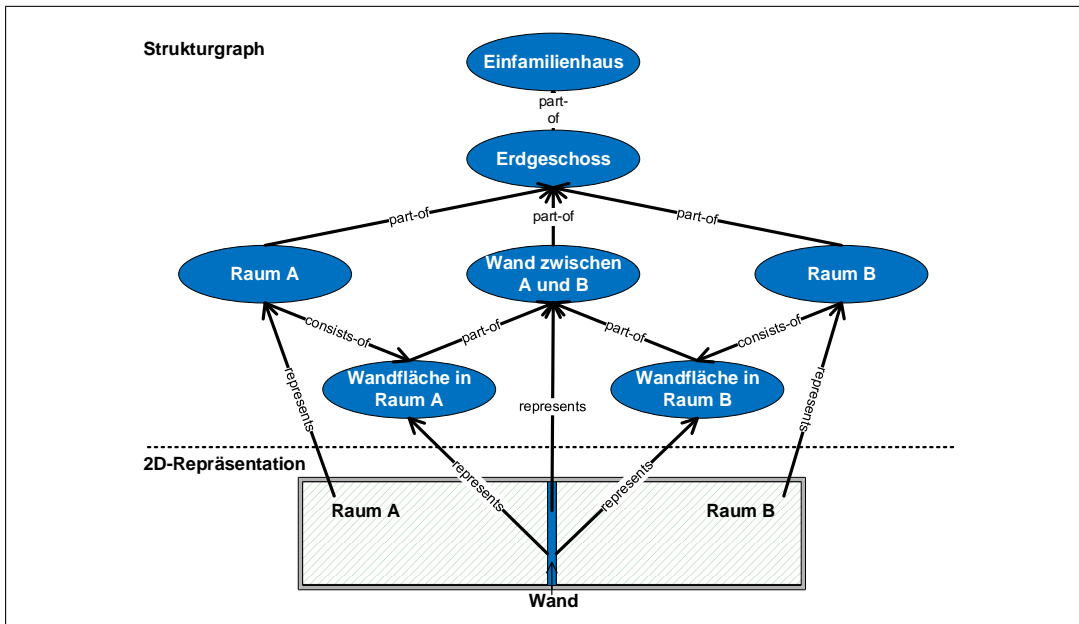


Abbildung 3.4.: Ausschnitt Strukturhierarchie typisiert - St. Lorenz

Der Strukturgraph enthält zwei Räume *Raum A* und *Raum B*. Diese beiden Räume sind durch eine *Wand zwischen A und B* getrennt. Die Räume und die Wand sind Teil des Erdgeschosses und existenzielle abhängig, daher kommt die *part-of*-Beziehung mit ihrer Wertesemantik zum Einsatz. Die Wand hat zu jedem Raum nur eine Wandfläche. Die Wandflächen sind Teil der Wand. Zwischen Raum und Wandflächen kommt die *consists-of*-Beziehung mit ihrer Referenzsemantik zum Einsatz. Ein Raum als symbolisches Gebäudeteil referenziert hier auf die einzelnen Wandflächen. □

Das erweiterte Datenmodell ist weiterhin in der Lage den MonArch-Anwendungsfall abzudecken. Beispiel 3.7 zeigt für einen Ausschnitt von St. Lorenz die Kompatibilität.

Beispiel 3.7 [Kompatibilität zum MonArch-Datenmodell]

In Abbildung 3.5 ist ein Ausschnitt der Strukturhierarchie von St. Lorenz mit einer grafischen Repräsentation zu sehen.

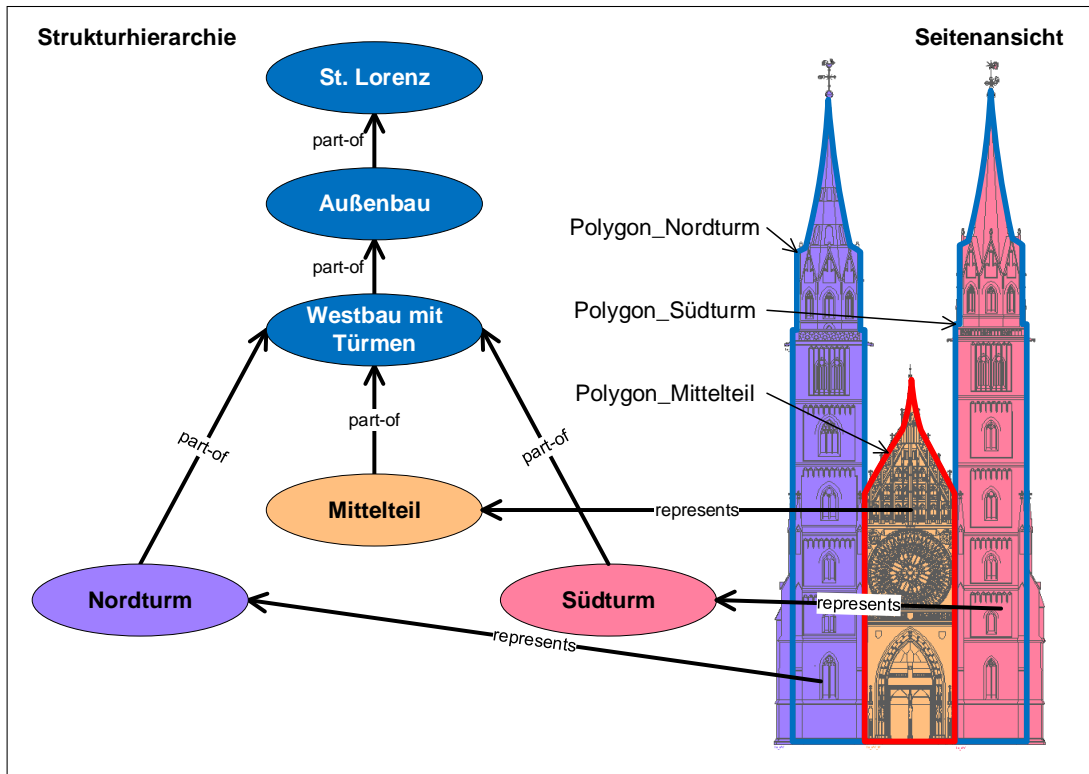


Abbildung 3.5.: Ausschnitt Strukturhierarchie und grafische Repräsentation - St. Lorenz

Das MonArch-Datenmodell sieht hier eine Zweiteilung der Daten vor. Einerseits wird die Strukturhierarchie des Gebäudes modelliert, andererseits die grafische Repräsentation. Im erweiterten Datenmodell lassen sich die grafischen Elemente mit dem Entity-Typen **Spatial Geometry Entity** umsetzen. *Polygon_Nordturm*, *Polygon_Südturm* und *Polygon_Mittelteil* repräsentieren die Westfassade von St. Lorenz, grafisch.

Die Gebäudeteile *St. Lorenz*, *Außenbau*, *Westbau mit Türmen*, *Mittelteil*, *Nordturm* und *Südturm* repräsentieren St. Lorenz symbolisch und werden daher als **Spatial Symbolic Entity** im erweiterten Datenmodell umgesetzt. □

3.2. Themengraph

Eine weitere Änderung gegenüber dem ursprünglichen MonArch-Datenmodell betrifft die Themen.

An das erweiterte Datenmodell werden folgende Anforderungen gestellt:

- Möglichkeit der Definition von Themen durch die Benutzer.
- Möglichkeit der Zusammenfassung von Themen zu Themenhierarchien durch die Benutzer.
- Möglichkeit der Zusammenfassung von Themen zu Themengruppen durch die Benutzer.
- Möglichkeit der Definition von Themenbeziehungen zur Beschreibung einer Verwandtschaft zwischen Themen.

Die erste und zweite Anforderung lassen sich aus dem MonArch-Anwendungsfall ableiten. Die dritte und vierte Anforderung ergeben sich aus der Notwendigkeit die Themenhierarchien weiter zu strukturieren, um bei einer Verbundanfrage diese Strukturen für die Transformation der Anfrage nutzen zu können.

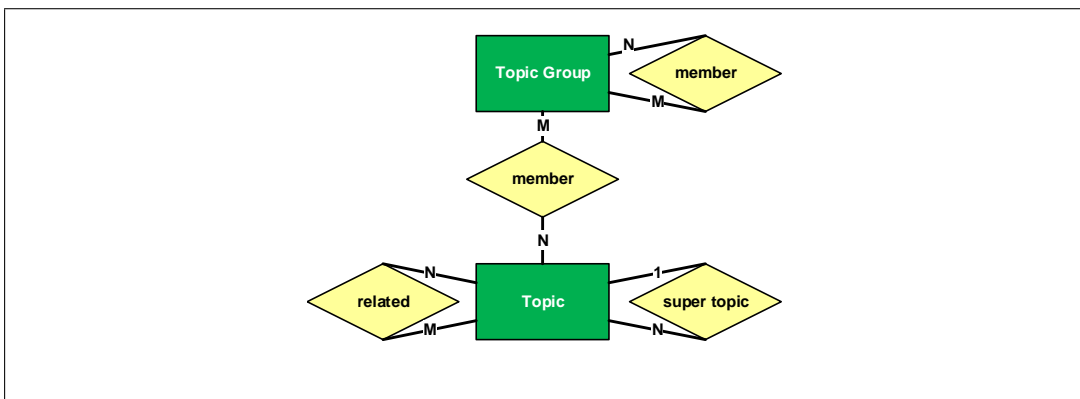


Abbildung 3.6.: Erweitertes Datenmodell (thematisch) als ER-Modell

Abbildung 3.6 zeigt den thematischen Ausschnitt des erweiterten Datenmodells, dessen Elemente folgende informelle Semantik haben.

Topic

Der Entity-Typ **Topic** modelliert die Themen im erweiterten Datenmodell. Die Themen werden über die Beziehung **super topic** hierarchisch strukturiert. Die Themenhierarchie des MonArch-Datenmodells kann damit vollständig abgebildet werden.

Darüber hinaus könnten Themen in einer **related**-Beziehung stehen, die es erlaubt, zwischen Themen eine Verwandtschaftsbeziehung zu spezifizieren.

Topic Group

Der Entity-Typ **Topic Group** erlaubt es Themen zu Themengruppen zusammenzufassen. Themengruppen können dabei rekursiv in Themengruppen geschachtelt werden.

Sowohl ein Oberthema als auch eine Themagruppe kann eine Themenhierarchie zu einer Gruppe zusammenfassen. Der Unterschied liegt darin, dass eine Themagruppe rein aus organisatorischen Zwecken Themen zusammenfasst, ein Oberthema immer einen semantischen Bezug zu den Themen, die es zusammenfasst hat. Diese Unterscheidung der Gruppierungsmöglichkeit ist an den SKOS-Standard [IS09] angelehnt.

Das erweiterte Datenmodell erlaubt eine stärkere Differenzierung zwischen einem Thema und einem Begriff, der eine Gruppe von Themen zusammenfasst. Die Begriffe die Themen zusammenfassen können nun als Themengruppen modelliert werden (siehe Beispiel 3.8).

Beispiel 3.8 [Kompatibilität zum MonArch-Datenmodell]

Abbildung 3.7 zeigt einen Ausschnitt der Themenhierarchie von St. Lorenz.

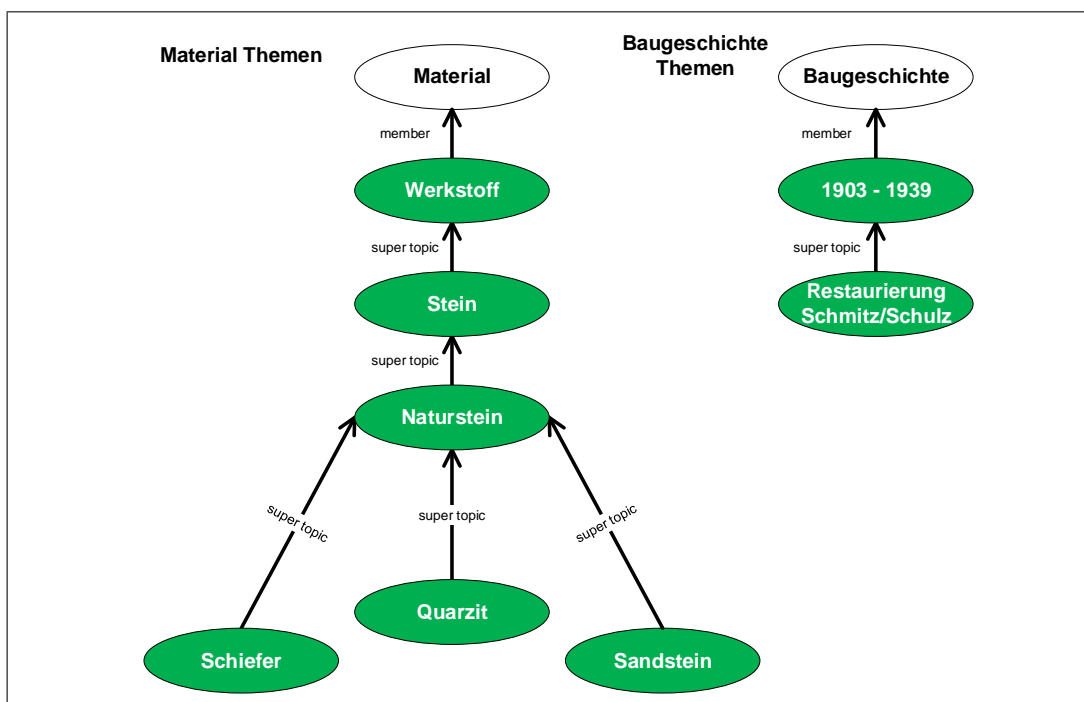


Abbildung 3.7.: Ausschnitt Themenhierarchie - St. Lorenz

In diesem Beispiel ist der Begriff **Material** nicht als Thema (**Topic**) sondern als Themengruppe (**Topic Group**) realisiert. Die Themen **Werkstoff**, **Stein**, **Naturstein**, **Schiefer**, **Quarzit** und **Sandstein** gehören der Themengruppe **Material** an. Der Begriff **Baugeschichte** ist eine zweite Themengruppe mit den Themen **1903 - 1939** und **Restaurierung Schmitz/Schulz**. Die Themenhierarchie ist wie im MonArch-Datenmodell über die **super topic**-Beziehung realisiert. □

3.3. Objektmodell

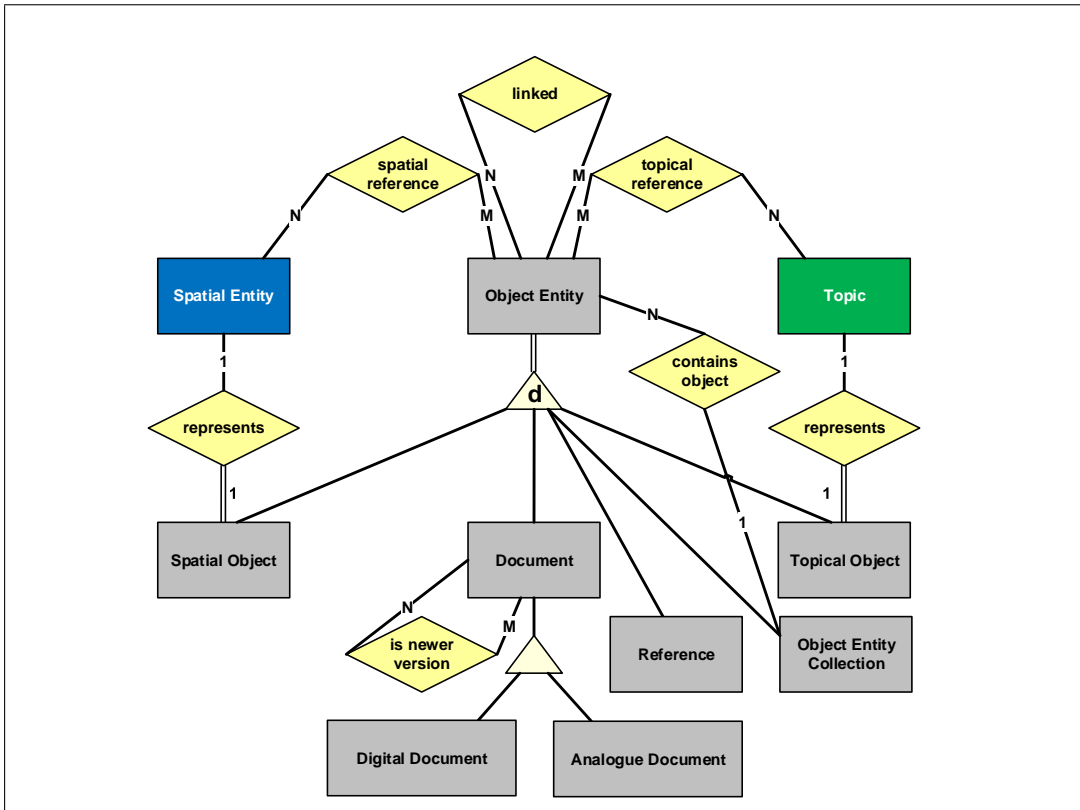


Abbildung 3.8.: Erweitertes Datenmodell (objektzentriert) als ER-Modell

Abbildung 3.8 zeigt das Objektmodell, dessen Elemente folgende informelle Semantik haben:

Object Entity

Der Entity-Typ **Object Entity** ist das zentrale Objekt im erweiterten Datenmodell. Jedes **Object Entity** lässt sich räumlich über die **spatial reference**-Beziehung als auch thematisch über die **topical reference**-Beziehung auszeichnen.

Wie im MonArch-Datenmodell können somit Informationen verortet und durch Themen beschrieben werden.

Um Beziehungen zwischen **Object Entity**s zu modellieren, führt das erweiterte Datenmodell eine **linked**-Beziehung ein.

Damit lassen sich Verweise zwischen Informationen, oder speziell zwischen Dokumenten, modellieren.

Der Entity-Typ **Object Entity** wird disjunkt aber nicht total in **Spatial Object**, **Topical Object**, **Document**, **Reference** und **Object Entity Collection** spezialisiert.

Der Verzicht auf eine totale Spezialisierung erlaubt eine gewisse Flexibilität bei der Repräsentation von Informationen, die nicht als eigener Entity-Typ modelliert sind. Im Rahmen dieser Arbeit sind für alle relevanten Informationen eigene Entity-Typen vorhanden.

Document

Der Entity-Typ **Document** definiert alle Arten von Dokumenten.

Dies sind einerseits digitale Dokumente, wie z.B. Textdokumente und Bilddokumente, andererseits kann der Entity-Typ **Document** auch Platzhalter für analoge Dokumente, wie z.B. Bücher oder Schriftrollen sein.

Daher wird der Entity-Typ **Document** in **Digital Document** und **Analogue Document** spezialisiert um beide Aspekte von Dokumenten abbilden zu können.

Stellen wir uns einen alten, per Hand gezeichneten, Plan vor. Dieser wird zum digitalen Dokument (**Digital Document**) falls er eingescannt und als Datei gespeichert wird. Weiterhin bleibt der Plan immer noch ein analoges Dokument (**Analogue Document**), da er nach dem Scannen ja nicht zerstört wird und z.B. in einem konventionellen Archiv aufbewahrt wird. Der gescannte Plan wird an dem Gebäudeteil verortet, das er zeigt. Zusätzlich kann der analoge Plan auch noch am Archivort verortet werden.

Der Relationship-Typ **is newer version** ermöglicht die Historisierung von Dokumenten in einem Graphen.

Digital Document

Digital Document sind alle Arten von digitalen Dokumenten unabhängig von ihrem Dateityp. Der Dateityp wird im Gegensatz zum MonArch-Datenmodell nicht als Entity-Typ modelliert, sondern als Eigenschaft des Dokuments.

Dies erlaubt eine leichtere Anpassung des Datenmodells für neue Dateitypen. Es entfällt somit die Einführung neuer Entity-Typen was unter Umständen je nach Realisierung eine Schemaänderung im relationalen Schema mit sich bringt.

Analogue Document

Analogue Document sind alle Arten von analogen Dokumenten.

Reference

Der Entity-Typ **Reference** modelliert Verweise auf Informationen.

Die Art der Informationen auf die verwiesen wird, ist dabei nicht spezifiziert und kann genau wie bei Dateitypen als Eigenschaft modelliert werden. Dieser Entity-Typ bietet einen generischen Mechanismus, um Verweise im Datenmodell abzubilden.

Object Entity Collection

Der Entity-Typ **Object Entity Collection** dient als Sammlung von **Object Entity** Elementen mit gleichen Eigenschaften. Hier findet das composite pattern Anwendung und erlaubt somit rekursiv Sammlung auf Sammlung aufzubauen. Dabei werden die Blätter in diesem Sammlungsbaum durch den Entity-Typ **Object Entity** repräsentiert.

Die eigentliche Zugehörigkeit zu einer Sammlung wird erst durch die **contains object**-Beziehung definiert.

Spatial Object

Das erweiterte Datenmodell erlaubt die Auszeichnung von Gebäudeteilen. Der Entity-Typ **Spatial Object** repräsentiert ein Gebäudeteil das thematisch ausgezeichnet werden kann.

Im erweiterten Datenmodell kann mit dem Entity-Typ **Spatial Object** ein Gebäudeteil analog zu einem Dokument behandelt werden und kann somit auch Ergebnis von Anfragen sein.

Diese Dualität der Gebäudeteile wird durch den Relationship-Typ **represents** zwischen **Spatial Object** und **Spatial Entity** in diesem Datenmodell abgebildet.

Die strikte Trennung zwischen diesen beiden Ausprägungen ist eine Designentscheidung die dazu dient, die einzelnen Ebenen und Metaebene zu trennen. Dies wiederum vereinheitlicht und erleichtert die Definition von Anfragen.

Topical Object

Neben den Gebäudeteilen können die Themen für sich gesehen auch ein Informationsobjekt darstellen. Der Entity-Typ **Topical Object** übernimmt diese Aufgabe im erweiterten Datenmodell.

Die Trennung zwischen Thema zur Beschreibung eines thematischen Kontextes und Thema als Informationsobjekt ist hier erneut gewährleistet.

Aus dem MonArch-Anwendungsfall, wie er in Kapitel 2 vorgestellt wurde, ergeben sich folgende Anforderungen an das erweiterte Datenmodell:

- Speicherung von digitalen Dokumenten.
- Räumliche Auszeichnung von Dokumenten.
- Thematische Auszeichnung von Dokumenten.
- Historisierung von Dokumenten.

Die Speicherung und Repräsentation von digitalen Dokumenten ist durch den Entity-Typen **Digital Document** abgedeckt. Durch die Spezialisierung von **Document** in **Digital Document** und **Analogue Document** erlaubt das hier vorgestellte erweiterte Datenmodell zwischen digitalen Dokumenten, also Dokumenten die durch eine Datei repräsentiert werden, und analogen Dokumenten zu unterscheiden. Somit wird die erste Anforderung „Speicherung von digitalen Dokumenten“ erfüllt.

Zusätzlich ist es möglich Dokumente, die beispielsweise noch nicht digitalisiert wurden oder nicht digitalisiert werden können, mit dem Datenmodell zu repräsentieren und genau wie digitale Dokumente zu behandeln.

Die vierte Anforderung „Historisierung von Dokumenten“ (**is newer version**) wird durch das erweiterte Datenmodell abgedeckt.

Die räumliche und thematische Auszeichnung von Dokumenten ist in diesem Datenmodell durch die Einführung des Entity-Typs **Object Entity** und der Relationship-Typen **spatial reference** und **topical reference** realisiert. Damit ist das hier vorgestellte Datenmodell in der Lage die vier Anforderungen, die sich aus dem MonArch-Anwendungsfall ableiten lassen, abzudecken.

Beispiel 3.9 [Zusammenfassung von Objekten zu Sammlungen]

In Abbildung 3.9 sind vier Sammlungen (**Object Entity Collection**) zu sehen, die an Gebäudeteilen verortet und mit Themen ausgezeichnet sind.

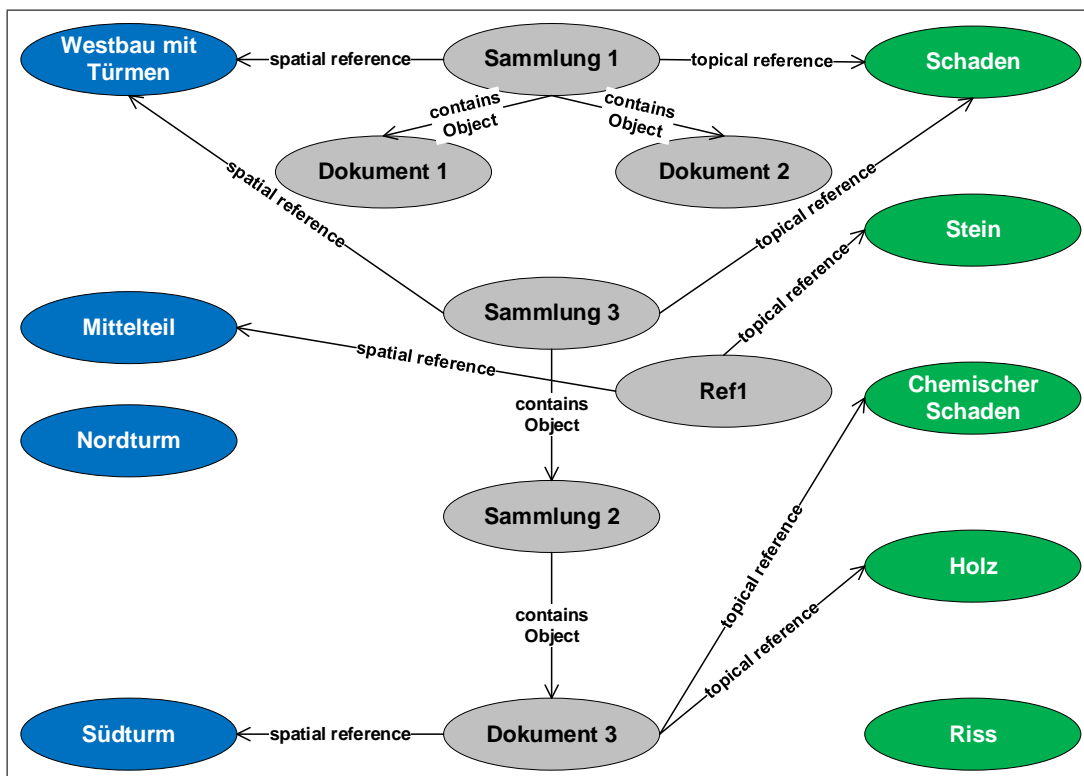


Abbildung 3.9.: Zusammenfassung von Objekten zu Sammlungen

Sammlung 1 enthält zwei Dokumente. Die räumliche und thematische Auszeichnung der Dokumente erfolgt in diesem Fall über die Sammlung.

Sammlung 3 enthält Sammlung 2, die wiederum ein Dokument enthält. Die räumliche und thematische Auszeichnung der Dokumente erfolgt dabei sowohl über die Sammlung (hier Sammlung 3) als auch über das Dokument selbst.

Das Dokument 3 kann sowohl über die Sammlung 3 als auch direkt über die Verortung am Südturm gefunden werden. □

Das erweiterte Datenmodell eröffnet mit den Entity-Typen **Spatial Object** und **Topical Object** die Möglichkeit Gebäudeteile und Themen thematisch und räumlich auszuzeichnen.

Beispiel 3.10 [Thematische Auszeichnung von Gebäudeteilen]

In Abbildung 3.10 nimmt das Gebäudeteil *Nordturm* zwei Rollen ein. Zum einen ist der *Nordturm* Teil des *Westbau mit Türmen* und ein *Spatial Symbolic Entity*. Die Rolle als *Spatial Symbolic Entity* erlaubt es Dokumente an diesem Gebäudeteil zu verorten.

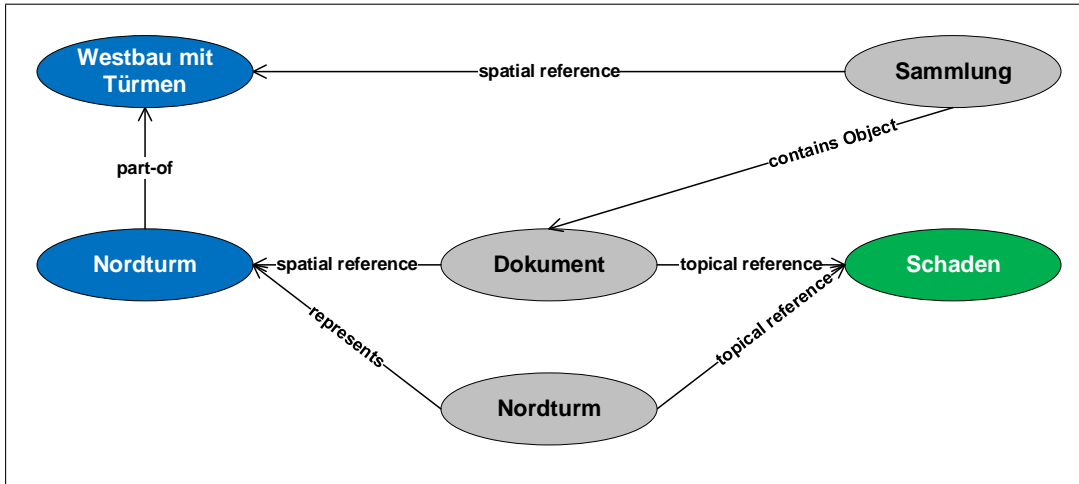


Abbildung 3.10.: Thematische Auszeichnung von Gebäudeteilen

Das Gebäudeteil ist aber auch ein *Spatial Object*. Die Rolle als *Spatial Object* erlaubt es den *Nordturm* thematisch mit dem Thema *Schaden* auszuzeichnen. Interpretiert werden kann diese Auszeichnung als Markierung eines Schadens an einem bestimmten Gebäudeteil. Es wird somit kein separates Dokument benötigt, das einen Schaden, z.B. in Form eines Bildes des Nordturms, zeigt.

Der *Nordturm* nimmt in diesem Beispiel eine Doppelrolle ein. Um sicherzustellen, dass der *Nordturm* sowohl als Gebäudeteil und als Informationsobjekt existiert, wird zwischen den beiden Repräsentationen des *Nordturms* eine **represents**-Beziehung modelliert. Das stellt sicher, dass unabhängig ob der *Nordturm* als Gebäudeteil angesehen wird oder als Ergebnis einer Anfrage, beide Repräsentationen zueinander in Verbindung stehen.

Diese Modellierung erlaubt somit einmal die Interpretation als Gebäudeteil und gleichzeitig auch als Objekt, das thematische Eigenschaften hat. □

3.4. Datenmodell in Beschreibungslogik

Um dem Ziel der Anfragetransformation einen Schritt näher zu kommen, musste ein Formalismus gewählt werden, der eine Vergleichbarkeit und Anwendbarkeit von bestehenden Verfahren in Bezug auf das erweiterte Datenmodell einfach erlaubt.

Im Bereich des Semantic Web werden oft Ontologien (siehe Definition 3.11) zur Modellierung der Wissensbasis oder der Metadaten eingesetzt.

Definition 3.11 [Ontologie[Sow99]]

Eine *Ontologie* ist eine formale Beschreibung eines abstrakten Modells in einer Anwendungsdomäne. Das Wissen der Anwendungsdomäne wird durch eine Menge an Konzepten und Beziehungen zwischen den Konzepten repräsentiert.

Eine Sprache **L** kann dabei zur Beschreibung eingesetzt werden.

Ontologien können in informelle und formale Ontologien unterteilt werden.

- Eine informelle Ontologie beschreibt bezeichnende Typen mit natürlicher Sprache auf eine lose Art und Weise.
- Eine formale Ontologie wird durch eine Sammlung von Namen definiert, die Konzepte und Beziehungen beschreiben, welche teilweise durch eine Typ-Untertyp-Beziehung geordnet sind.

Eine formale Ontologie ist definiert als Vierer-Tupel **(C,R,I,X)**, mit

- **C** als Menge von Konzepten,
- **R** als Menge von Rollen,
- **I** als Menge von Individuen,
- **X** als Menge an Axiomen.

Zusätzlich mit der Bedeutung, dass

- ein Konzept eine Menge von Individuen darstellt,
- eine Rolle eine binäre Beziehung zwischen Individuen darstellt,
- ein Individuum eine Instanz eines Konzepts ist und
- ein Axiom eine Aussage über Konzepte, Rollen und Individuen ist, welches in einer formalen Sprache **L** beschrieben und als wahr ohne Beweis angenommen wird.

□

Es bietet sich daher an, den Strukturgraphen und den Themengraphen mithilfe von Ontologien zu modellieren, wie es ähnlich auch im Ansatz von [Dec+98] geschehen ist. Ein weiterer Aspekt der für den Einsatz von Ontologien für die Modellierung spricht, wird in [AE09] beschrieben.

The most important application of ontologies in KM – besides browsing interfaces in Knowledge Portals – is certainly to improve search and retrieval of documents by exploiting ontological background knowledge about the application domain.[AE09][723]

Als Ontologiesprache hat sich OWL im Bereich des „Semantic Web“ etabliert, insbesondere die Teilsprache OWL-DL und später OWL-2-DL [Car07]. Einer der Vorteile von OWL-DL bzw. OWL-2-DL, im Vergleich zu anderen Teilsprachen, ist die Möglichkeit die Semantik der Sprache mithilfe von Beschreibungslogik zu definieren [HPH03]. Dies erlaubt ein konzeptionelles Arbeiten in Beschreibungslogik mit anschließender Überführung in OWL-2-DL zur maschinellen Verarbeitung.

Die Reformulierung des erweiterten Datenmodells mithilfe von Ontologien und Beschreibungslogik eröffnet folgende Möglichkeiten:

- Eine erweiterte Definition der Eigenschaften des Datenmodells über die Möglichkeiten des ER-Modells.
- Eine Prüfung des Datenmodells auf Konsistenz durch den Einsatz von OWL-2-DL Reasoner
- Eine Vergleichbarkeit mit anderen ontologiebasierten Ansätzen wird vereinfacht. Eine Übersicht gibt [SE13].
- Die ontologische Darstellung eröffnet Möglichkeiten die Suche zu verbessern.
- Die entwickelten Verfahren für Anfragen im Verbund, können auf andere Anwendungsfelder erweitert werden.
- Ontologien bieten abhängig von der ausgewählten Sprache einen verlässlichen Formalismus, um die semantischen Zusammenhänge zu beschreiben.
- Beschreibungslogik ist eine formale Sprache zur Beschreibung der Ontologien.
- Ein so beschriebenes Datenmodell lässt sich in OWL-DL bzw. OWL-2-DL transformieren [HPH03; MPG12].
- Ein in Beschreibungslogik beschriebenes Datenmodell lässt sich, mit Einschränkungen, auch in ein ER-Modell zurücktransformieren [VN06; MM09].

Als Grundlage für das in Beschreibungslogik modellierte erweiterte Datenmodell wird die Sprache *SR0IF* verwendet. *SR0IF* ist ein Fragment der Sprache *SR0IQ* welches erstmals in [HKS06] vorgestellt wurde und die Grundlage für die direkte Semantik [MPG12] von OWL-2-DL darstellt.

Bei der Reformulierung wird größtenteils nach [UK05] und nach [Bag09] vorgegangen.

- Entity-Typen werden als Konzepte modelliert.
- Relationship-Typen werden als Rollen modelliert.
- 1:N-Beziehungen werden durch funktionale Rollen definiert.
- 1:1-Beziehungen können zusätzlich noch inversfunktional definiert werden.
- Da Relationship-Typen keine Richtung haben können diese mit Rollen und inverse Rollen umgesetzt werden.

- Definition- und Wertebereich von Rollen können genutzt werden um Rollen einzuschränken. Die Einschränkung ergibt sich aus den beteiligten Entity-Typen des Relationship-Typ.
- Die is-a-Beziehung kann durch eine geeignete Subsumption von Konzept abgebildet werden.

Die informelle Semantik wird dabei vom erweiterten Datenmodell aus dem ER-Modell übertragen.

Die einzelnen Bereiche des erweiterten Datenmodells werden mit Teilontologien umgesetzt. Es entsteht somit eine Aufteilung in eine Raumontologie, eine Objektontologie und eine Themaontologie. Zum besseren Verständnis und zur Differenzierung von Schema und Instanz des neuen Datenmodells werden die Ontologien nochmals in Schema-Ontologien und Instanz-Ontologien aufgeteilt. Im erweiterten Datenmodell war diese Aufteilung nicht notwendig, da durch das ER-Modell Schema und Instanz klar getrennt sind.

Die Teilontologien werden am Schluss zu einer großen Ontologie zusammengefasst. Dazu werden Operationen auf Ontologien benötigt, die in Definition 3.12 definiert werden.

Definition 3.12 [Ontologie-Operationen]

Seien $A = (\mathbf{C}_A, \mathbf{R}_A, \mathbf{I}_A, \mathbf{X}_A)$ und $B = (\mathbf{C}_B, \mathbf{R}_B, \mathbf{I}_B, \mathbf{X}_B)$ zwei Ontologien. \odot stehe für einen der Operatoren \cup , \cap oder \setminus .

Dann definieren wir die Ontologie-Operation $A \odot B$ von A und B durch

$$A \odot B := (\mathbf{C}_A \odot \mathbf{C}_B, \mathbf{R}_A \odot \mathbf{R}_B, \mathbf{I}_A \odot \mathbf{I}_B, \mathbf{X}_A \odot \mathbf{X}_B)$$

$A \cup B$ bezeichnen wir als *Ontologie-Vereinigung*, $A \cap B$ als *Ontologie-Schnitt* und $A \setminus B$ als *Ontologie-Differenz*.

Als Spezialfall ist eine *Ontologie-Erweiterung* definiert durch

$$0 = A \cup B$$

Eine Ontologie-Erweiterung erlaubt somit z.B. die Aufteilung der Ontologie O in eine TBox und ABox Ontologie $0 = 0(TBox) \cup 0(ABox)$. \square

Die im Folgenden eingeführten Konzepte und Rollen werden mit den in Tabelle 3.1 aufgeführten Sprachkonstrukten repräsentiert. Die Interpretation dieser Sprachkonstrukte ist in [BHS09][S. 27-30] und in [HS04] aufgeführt. In „The Description Logic Handbook: Theory, Implementation and Applications“ [Baa+07] findet sich eine ausführlichere Beschreibung.

3. Erweitertes Datenmodell

DL	Bedeutung
ConceptName	Definition Konzept mit „ConceptName“ (TBox)
RoleName	Definition Rolle mit „RoleName“ (TBox)
IndividualName	Definition Individuum mit „IndividualName“ (ABox)
$\text{ConceptName}(\text{IndividualName})$	Definition Konzeptaussage (ABox)
$\text{RoleName}(\text{INameFrom}, \text{INameTo})$	Definition Rollenaussage (ABox)
$\exists \text{RoleName}. \top \sqsubseteq \text{ConceptName}$	Definition Definitionsbereich einer Rolle (TBox)
$\top \sqsubseteq \forall \text{RoleName}. \text{ConceptName}$	Definition Wertebereich einer Rolle (TBox)
$\text{Asy}(\text{RoleName})$	Definition Eigenschaft Asymmetrie für eine Rolle (TBox)
$\text{RoleName} \sqcap \text{RoleName}^{-} \sqsubseteq \perp$	Definition Eigenschaft Irreflexiv für eine Rolle (TBox)
$\text{RoleName} \equiv \text{RoleName}^{-}$	Definition Eigenschaft Symmetrie für eine Rolle (TBox)
$\top \sqsubseteq \leq 1 \text{RoleName}$	Definition Eigenschaft Funktional (1 zu N) für eine Rolle (TBox)
$\top \sqsubseteq \leq 1 \text{RoleName}^{-}$	Definition Eigenschaft Inversfunktional (N zu 1) für eine Rolle (TBox)
$\text{RoleName} \equiv \text{RoleName}^{+}$	Definition Eigenschaft Transitivität für eine Rolle (TBox)
$\text{ConceptName} \sqsubseteq \text{SuperConceptName}$	Definition Subsumption für Konzepte (TBox)
$\text{RoleName} \sqsubseteq \text{SuperRoleName}$	Definition Rollenhierarchie (TBox)
$\text{ConceptName1} \sqcap \text{ConceptName2} \sqsubseteq \perp$	Definition Disjunktheit von Konzepten (TBox)
$\text{RoleName1} \sqcap \text{RoleName2} \sqsubseteq \perp$	Definition Disjunktheit von Rollen (TBox)
$\text{RoleName1} \equiv \text{RoleName2}^{-}$	Definition Rolleninverse (TBox)
$\text{RoleName1} \circ \text{RoleName2} \sqsubseteq \text{RoleName3}$	Definition Rollenkomposition (TBox)

Tabelle 3.1.: Wichtige Sprachkonstrukte der Beschreibungslogik SR0IF mit Rollenkomposition

3.4.1. Raum- und Typontologie

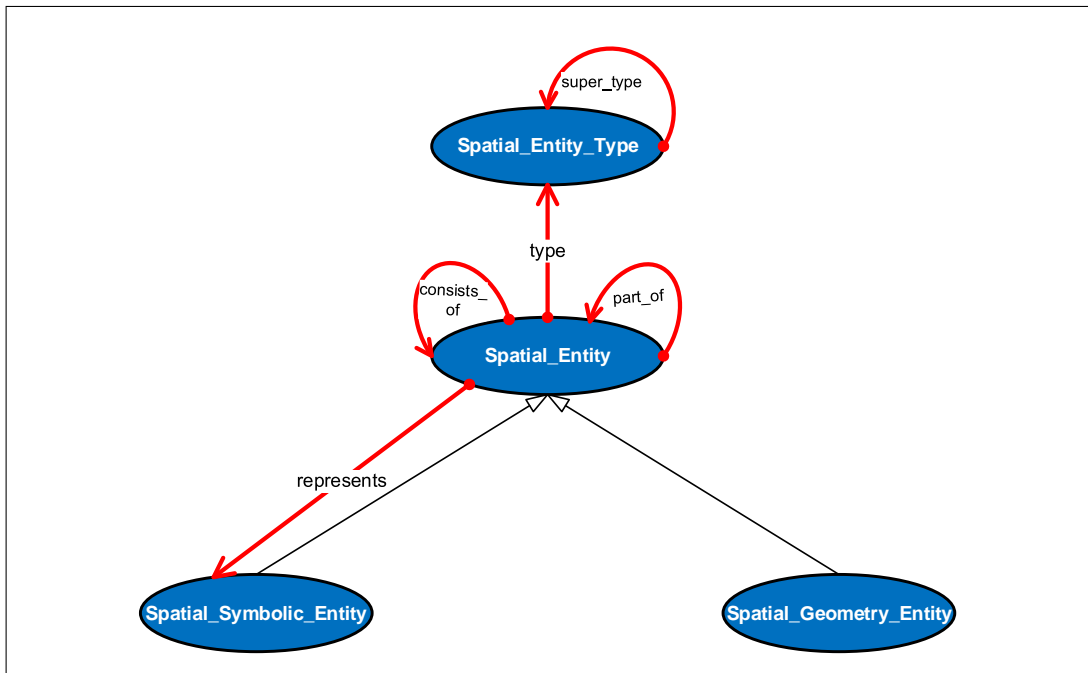


Abbildung 3.11.: Konzepte und Rollen der Raum- und Typontologie

Die Möglichkeiten zur Definition und Zuordnung von Typen des erweiterten Datenmodells lassen sich als Typontologie (siehe Abbildung 3.11) formulieren. Der Entity-Typ **Spatial Entity Type** wird als Konzept `Spatial_Entity_Type` in der Typontologie realisiert.

Die Typhierarchie wird nicht in Form von Konzeptspezialisierungen in der TBOX realisiert, sondern mit Rollen in der ABOX. Dies hat das Ziel, die Typontologie offen für mögliche Erweiterungen zu lassen, die z.B. Aussagen über Typen beinhalten.

Der Relationship-Typ **super type** wird durch die Rolle `super_type` realisiert. Die 1:N Funktionalität von **super type** wird durch die funktionale Rolle `super_type` realisiert.

Zusätzlich wird die inverse Rolle `sub_type` zu `super_type` eingeführt. Zu beiden Rollen existieren in der Typontologie noch transitive Versionen `sub_type_transitive` und `super_type_transitive`, die eine Materialisierung der transitiven Hülle von `super_type` durch eine Inferenz in der Typontologie ermöglicht.

Die Rollen `super_type` und `sub_type` sind asymmetrisch und irreflexiv definiert, was zu einer Präzisierung der Typontologie beiträgt:

- Ein Typ soll nicht Ober- bzw. Untertyp von sich selbst sein.
- Ist ein Typ Untertyp eines Obertyps soll der Obertyp nicht Untertyp des Typs sein.

Diese beiden Eigenschaften konnten im ER-Model nicht definiert werden. Daher erlaubt die Reformulierung des erweiterten Datenmodells in Beschreibungslogik die Definition von weiteren Eigenschaften.

Eine vollständige Definition der Typontologie, die wir mit $O_{\text{schema}}^{\text{spatial_type}}$ bezeichnen, ist im Anhang in Definition B.1 zusammen mit dem Beispiel B.2 zu finden.

Basierend auf der Typontologie kann eine Raumontologie (siehe Abbildung 3.11) definiert werden um die Entity-Typen **Spatial Symbolic Entity** und **Spatial Geometry Entity** als Konzepte umzusetzen. Ein gemeinsames Oberkonzept `Spatial_Entity` wird in `Spatial_Symbolic_Entity` und `Spatial_Geometry_Entity` verfeinert.

Die disjunkte und totale is-a-Beziehung kann durch folgende Axiome abgebildet werden:

$$\begin{aligned} \text{Spatial_Symbolic_Entity} \sqcap \text{Spatial_Geometry_Entity} &\sqsubseteq \perp, \\ \text{Spatial_Entity} & \\ &\equiv \text{Spatial_Symbolic_Entity} \sqcup \text{Spatial_Geometry_Entity} \end{aligned}$$

Für die Beschreibung der Beziehungen im Strukturgraphen stehen die Rollen *part_of* und *consists_of* zur Verfügung. Wie auch in der Typontologie gibt es passende inverse und transitive Rollen *has_part*, *has_part_transitive* und *part_of_transitive*. Für die Rollen *part_of* und *consists_of* kann zusätzlich die Irreflexivität und Asymmetrie gefordert werden.

Zur Verknüpfung der grafischen mit der symbolischen Repräsentation eines Gebäudeteils wird für den Relationship-Typ **represents** die Rolle *represents* eingeführt. Die Rolle *represents* erlaubt es die grafische Repräsentation mit einer symbolischen Repräsentation zu verbinden und somit Gebäudeteilen eine grafische Darstellung zuzuordnen.

Zur Zuordnung eines Typs zu `Spatial_Entity` gibt es die Rolle *type*. Über eine Rollenkomposition lassen sich mithilfe einer Inferenz für ein Gebäudeteil all seine Typen berechnen:

$$\text{type} \circ \text{super_type_transitive} \sqsubseteq \text{type_transitive}$$

Beispiel 3.13 [Berechnung der Typen durch Rollenkomposition]

Betrachten wir die Situation in Abbildung 3.12:

Der *Nordturm* ist ein Individuum des Konzepte `Spatial_Symbolic_Entity`. Die Typen *Gebäudeteil*, *Turm* und *Rundturm* sind Individuen des Konzeptes `Spatial_Entity_Type`. Dem *Nordturm* ist der Typ *Rundturm* zugeordnet.

Eine Inferenz berechnet die *super_type_transitive*-Rollenaussagen (blaue gestrichelte Linien) auf Basis der *super_type*-Rollenaussagen. Mit den so berechneten *super_type_transitive*-Rollenaussagen und der Rollenkomposition $\text{type} \circ \text{super_type_transitive}$ können die *type_transitive*-Rollenaussagen inferiert werden (rote gestrichelte Linien). Damit hat der *Nordturm* die transitiven Typen *Gebäudeteil* und *Turm*.

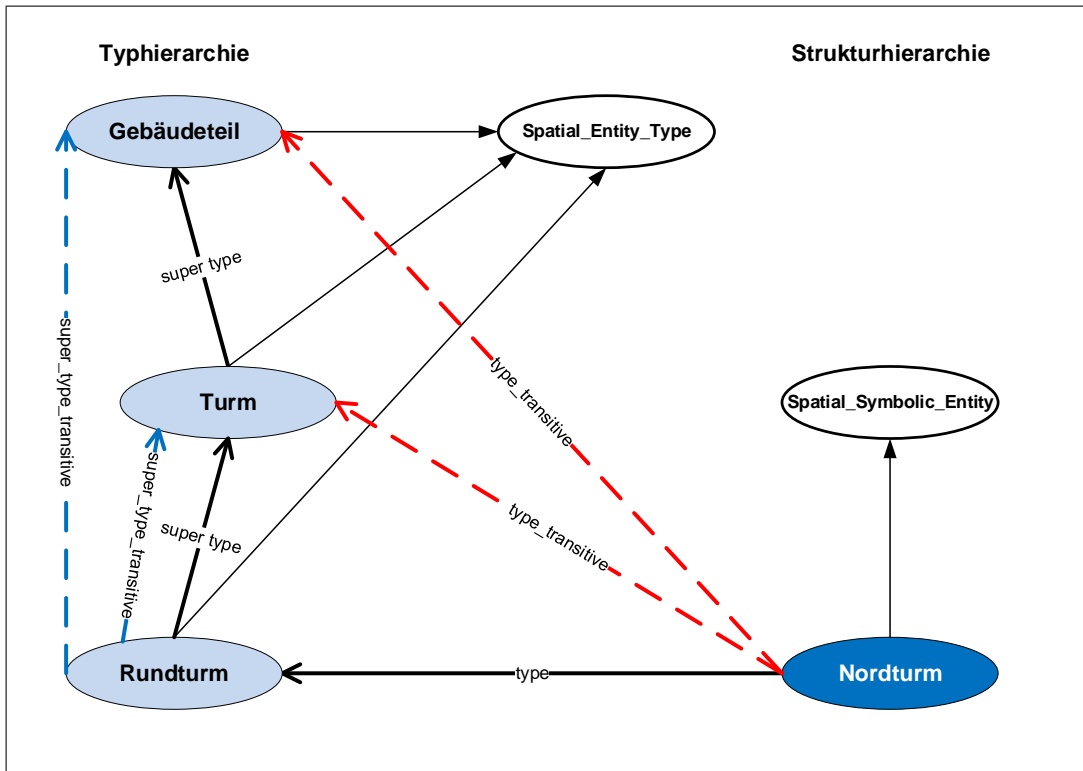


Abbildung 3.12.: Typ und transitive Typen eines Gebäudeteils

□

Eine vollständige Definition der Raumontologie, die wir mit $\mathcal{O}_{\text{schema}}^{\text{spatial}}$ bezeichnen, ist im Anhang in Definition B.3 zusammen mit dem Beispiel B.4 zu finden.

Das erweiterte Datenmodell sieht die Modellierung von geometrischen Beziehungen zwischen **Spatial Geometry Entity** vor. Diese Beziehungen können wieder mit Rollen abgebildet werden. In Abbildung 3.13 ist die Rollen-Hierarchie für die möglichen Beziehungen des DE-9IM Schnittmodells dargestellt. Auch hier kann die Reformulierung des Datenmodells in Beschreibungslogik dazu beitragen, die Eigenschaften der Beziehungen zu präzisieren.

Abbildung 3.13 zeigt mit den grünen Sub-Pfeilen die Rolleninklusion. Die roten Dis-Pfeile zeigen die Disjunktheit zwischen zwei Rollen. Die blauen Inv-Pfeile zeigen inverse Rollen an. Weitere Rolleneigenschaften wie Symmetrie und Asymmetrie werden neben dem Rollenbezeichner vermerkt.

Im praktischen Einsatz würden Rollenaussagen mit diesen Rollen maschinell erzeugt werden. Die Geometriedaten der `Spatial_Geometry_Entity` erlauben eine Berechnung und dann ein Materialisieren in einer Instanzontologie als Rollenaussagen.

Die vollständige Definition der Rollen ist im Anhang in Definition B.5 nachzulesen. Ein Beispiel ist ebenfalls dort zu finden (siehe Beispiel B.6).

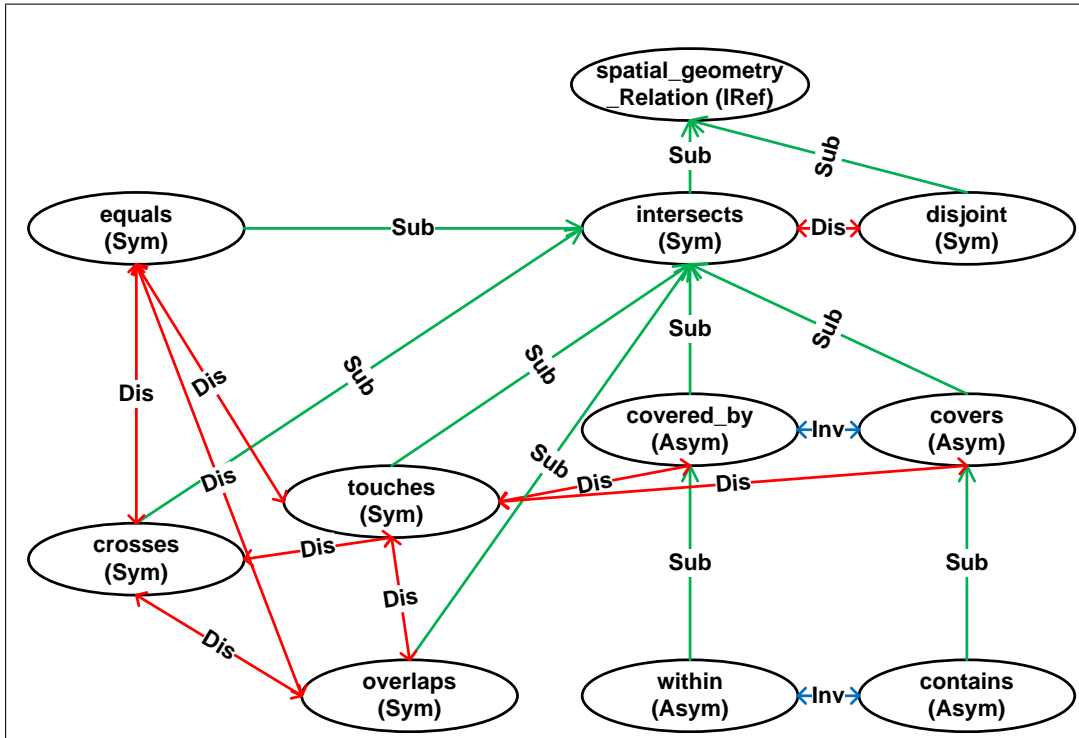


Abbildung 3.13.: Rollen-Hierarchie der geometrischen räumlichen Rollen

Für den dreidimensionalen Fall können die Rollen auf die Arbeit von [BR08; BR09] zurückgeführt werden, indem die existierenden zweidimensionalen Beziehungen des DE-9IM Schnittmodells auf die dritte Dimension erweitert und angepasst werden. Als Konsequenz kann für die dritte Dimension die Anzahl der Rollen verringert werden, ohne dass für den hier beschriebenen Anwendungsfall Nachteile entstehen.

3.4.2. Themaontologie

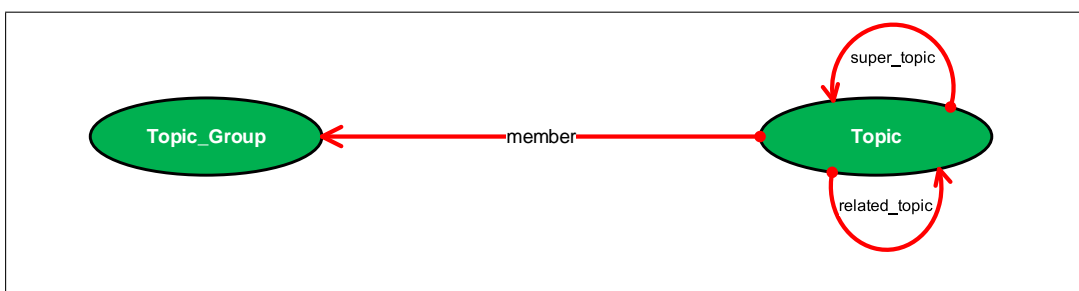


Abbildung 3.14.: Konzepte und Rollen der Themaontologie

Der thematische Bereich des erweiterten Datenmodells lässt sich als Themaontologie (siehe Abbildung 3.14) formulieren.

Die Entity-Typen **Topic** und **Topic Group** lassen sich als Konzepte `Topic` und `Topic_Group` realisieren. Die Relationship-Typen **super topic**, **member** und **related** lassen sich als Rollen `super_topic`, `member_topic` und `related_topic` realisieren.

Für die Rolle `super_topic` wird eine inverse Rolle `sub_topic` eingeführt und jeweils eine transitive Rolle `super_topic_transitive` und `sub_topic_transitive`. Die Rolle `super_topic` definieren wir als irreflexiv, asymmetrisch und funktional. Die inverse Rolle `sub_topic` definieren wir passend dazu als irreflexiv, asymmetrisch und inversfunktional. Als symmetrisch und irreflexiv definieren wir die Rolle `related_topic`. Die Rolle `member_topic` ist asymmetrisch und irreflexiv.

Eine vollständige Ontologiedefinition der Themaontologie $O_{\text{schema}}^{\text{topical}}$ ist im Anhang in Definition B.7 zusammen mit dem Beispiel B.8 zu finden.

Die Realisierung des thematischen Bereichs des erweiterten Datenmodells als Themaontologie ermöglicht eine Abbildung auf den Standard Simple Knowledge Organization System (SKOS) [IS09].

Eine Abbildung kann mit folgenden Schritten realisiert werden:

- Abbildung der Konzepte `Topic` und `Topic_Group` auf die Konzepte `skos:Concept` und `skos:ConceptScheme`. Themen werden auf SKOS-Konzepte abgebildet. Die Themengruppen entsprechen den `skos:ConceptScheme` aus dem SKOS-Standard.
- Die Rolle `super_topic` lässt sich auf die Rolle `skos:broader` aus dem SKOS-Standard abbilden. Für die inverse Rolle zu `super_topic` und den transitiven Rollen finden sich im SKOS-Standard passend Rollen.
- Die Rolle `related_topic` wird auf die Rolle `skos:related` abgebildet.
- Die Rolle `member_topic` wird auf die Rolle `skos:member` abgebildet.

Die Abbildung fassen wir in eine Ontologie $O_{\text{mapping}}^{\text{topical}}$ (SKOS) (siehe Definition B.9) zusammen. Dies erlaubt den Themengraph des erweiterten Datenmodells als SKOS-Ontologie für einen Austausch mit anderen Systemen einzusetzen und die Verwendung von standardisierten Vokabularen im SKOS-Format. Die Abbildung kann dabei durch einen Inferenz-Schritt berechnet werden.

Für den MonArch-Anwendungsfall ermöglicht dies sowohl den Austausch von Themenhierarchien als auch die Wiederverwendung und Erweiterung bestehender Themenhierarchien.

Um die Wiederverwendung einer bestehenden Themenhierarchie zu veranschaulichen, zeigt Beispiel 3.14 ein Anwendungsbeispiel, in dem ein bestehendes kontrolliertes Vokabular als Themaontologie über die SKOS-Abbildung Verwendung findet.

Beispiel 3.14 [Ausschnitt der GND als Themaontologie]

Die Deutsche Nationalbibliothek stellt im Rahmen ihres „Linked Data Service“ die gemeinsame Normdatei (GND) [Hen17] für die Öffentlichkeit unter anderem auch im SKOS-Format zur Verfügung.

3. Erweitertes Datenmodell

In diesem Beispiel wird die Ontologie $\mathcal{O}_{\text{mapping}}^{\text{topical}}(\text{SKOS})$ mit einem Fragment aus der GND erweitert. Diese Erweiterung ermöglicht es Teile der GND über die SKOS-Abbildung als Themenhierarchie zu verwenden. Durch die Abbildung auf den SKOS-Standard werden die Themen aus der GND mit ihren Beziehungen zu Themen in der hier vorgestellten Themaontologie. Das Beispiel beschränkt sich dabei auf die relevanten Konzepte und Rollen aus der GND.

$$\mathcal{O}_{\text{mapping}}^{\text{topical}}(\text{GND}) := (\mathbf{C}, \mathbf{R}, \mathbf{I}, \mathbf{X}) \cup \mathcal{O}_{\text{mapping}}^{\text{topical}}(\text{SKOS}) \text{ mit:}$$

den Konzepten \mathbf{C} :

gnd:SubjectHeading

den Rollen \mathbf{R} :

gnd:broaderTermGeneral

den Individuen $\mathbf{I} = \emptyset$

den Axiomen \mathbf{X} :

skos:Concept = gnd:SubjectHeading

skos:broader = gnd:broaderTermGeneral

Durch die Gleichsetzung des Konzepts *skos:Concept* mit dem Konzept *gnd:SubjectHeading* können Themen der GND als *Topic* in dem hier vorgestellten Datenmodell wiederverwendet werden. Die Übernahme der Themenhierarchie ist durch die Gleichsetzung der Rollen *skos:broader* und *gnd:broaderTermGeneral* gewährleistet. \square

3.4.3. Objektontologie

Mit den vorher definierten Raum- und Themaontologien stehen Ontologien zur Verfügung, um räumliche und thematische Eigenschaften von Objekten zu beschreiben.

Der zentrale Bereich des erweiterten Datenmodells lässt sich in einer Objektontologie (siehe Abbildung 3.15) umsetzen.

Der Entitätentyp **Object Entity** und seine Spezialisierungen werden als Konzepte *Object_Entity*, *Reference*, *Document* und *Object_Entity_Collection* realisiert. Die Relationship-Typen **spatial reference** und **topical reference** werden als Rollen *spatial_reference* und *topical_reference* realisiert.

Die Rolle *is_newer_version*, die aus dem Relationship-Typ **is newer version** hervorgeht, wird zusätzlich als irreflexiv und asymmetrisch definiert:

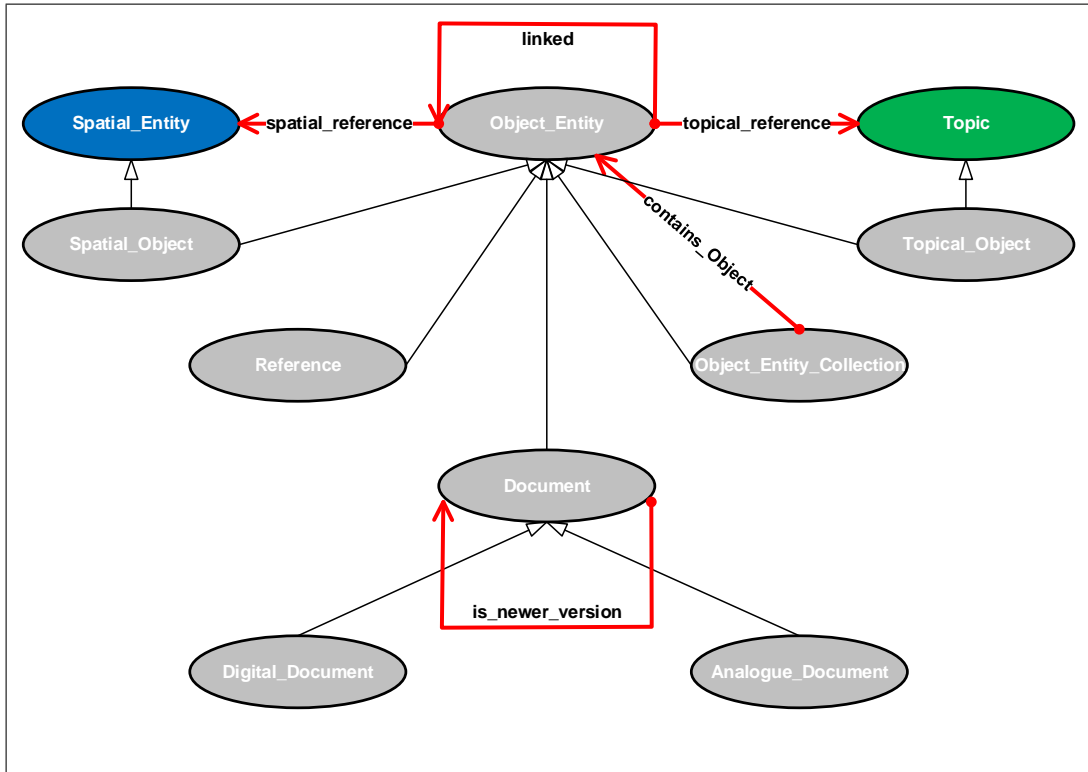


Abbildung 3.15.: Konzepte und Rollen der Objektontologie

- Dies verhindert, dass ein Dokument zu sich selbst in einer direkten Versionsbeziehung steht.
- Die Asymmetrie stellt zusätzlich sicher, dass eine neuere Version eines Dokuments nicht zusätzlich als ältere Version auftreten kann.

Im ER-Modell beschreibt der Relationship-Typ **represents** zwischen **Object Entity** und **Spatial Entity** die Möglichkeit, Gebäudeteile nicht nur für die räumliche Auszeichnung von Informationen zu verwenden, sondern Gebäudeteile selbst wieder thematisch und räumlich auszuzeichnen. Folglich muss diese Dualität der Gebäudeteile in Beschreibungslogik abgebildet werden.

Das Konzept `Spatial_Object` bildet als Unterkonzept der Konzepte `Object_Entity` und `Spatial_Entity` diese Dualität ab.

$$\text{Spatial_Object} \sqsubseteq \text{Object_Entity} \sqcap \text{Spatial_Entity}$$

Gebäudeteile sind daher auch Informationsobjekte.

Für Themen im ER-Modell beschreibt der Relationship-Typ **represents** zwischen **Object Entity** und **Topic** die Möglichkeit, diese selbst wieder thematisch und räumliche auszuzeichnen.

Folglich muss wie bei den Gebäudeteilen diese Dualität der Themen in Beschreibungslogik abgebildet werden.

Das Konzept `Topical_Object` bildet als Unterkonzept der Konzepte `Object_Entity` und `Topic` diese Dualität ab.

$$\text{Topical_Object} \sqsubseteq \text{Object_Entity} \sqcap \text{Topic}$$

Themen sind daher auch Informationsobjekte.

Eine vollständige Definition der Objektontologie, die wir mit $\mathcal{O}_{\text{schema}}^{\text{object}}$ bezeichnen, ist im Anhang in Definition B.10 zusammen mit dem Beispiel B.11 zu finden.

3.4.4. Realisierung in OWL-2-DL

Damit das erweiterte Datenmodell praktisch genutzt werden kann, muss die Definition in Beschreibungslogik in eine Realisierung überführt werden.

Die „OWL 2 Web Ontology Language“ bietet sich hier als Ontologiesprache an. Ihre Semantik ist in der „Direct Semantics (Second Edition)“ [MPG12] beschrieben. OWL-2 bietet mehrere Subsprachen an, wobei sich OWL-2-DL durch ihre Nähe zur Beschreibungslogik in diesem Fall am besten eignet [Hor05].

Die Arbeiten von [HPH03] und die darauf aufbauenden Publikationen [HKS06; MPG12] erläutern die notwendigen Grundlagen eine Realisierung der hier beschriebenen Ontologien in OWL-2-DL.

Da für das Datenmodell die Beschreibungslogiksprache `SR0IF` zum Einsatz kommt, können alle Sprachelemente der Beschreibungslogik in die dazugehörigen Sprachelemente von OWL-2-DL umgesetzt werden.

OWL-2-DL sieht für die Wissensrepräsentation folgende einfache Darstellungsart vor [Hit+12]:

- Axiome: Die wesentlichen Aussagen, die in einer OWL-Ontologie ausgedrückt werden.
- Entitäten: Elemente, um Objekte aus der realen Welt zu benennen.
- Ausdrücke: Eine Kombination von Entitäten, um aus den wesentlichen Aussagen komplexere aufzubauen.

Um diese drei Darstellungsarten in OWL-2-DL nutzen zu können, und damit Ontologien zu modellieren, stellt OWL-2-DL Klassen, Objekteigenschaften, Individuen und Datenwerte zur Verfügung.

- Klassen (classes):

Konzepte `ConceptName` werden in OWL-2-DL als Klassen definiert (siehe Listing 3.1).

```
Declaration(Class (<ConceptName>))
```

Listing 3.1: OWL-2-DL Klassendefinition

Klassen können dazu genutzt werden, Individuen mit Gemeinsamkeiten zu gruppieren. Ein Individuum Wand (*Wall*) könnte z.B. zur Klasse der räumlichen Entitäten (*Spatial_Entity* (*Wall*)) gehören (siehe Listing 3.2).

```
Declaration(NamedIndividual( :Wall )
ClassAssertion( :Spatial_Entity :Wall )
```

Listing 3.2: OWL-2-DL Klasse

Klassen können in Klassenhierarchien analog zu Konzepthierarchien modelliert werden (siehe Listing 3.3).

```
SubClassOf( :Digital_Document :Document )
```

Listing 3.3: OWL-2-DL Klassenhierarchien

Zusätzlich ist es möglich, die Gleichheit und die Disjunktheit von Klassen zu formulieren (siehe Listing 3.4).

```
EquivalentClasses( :Document :File )
DisjointClasses( :Digital_Document :Analogue_Document )
```

Listing 3.4: OWL-2-DL Gleichheit und Disjunktheit

- Objekteigenschaften (object properties):

Rollen werden mit Objekteigenschaften abgebildet und können Beziehungen zwischen Individuen modellieren. Das Individuum Wand (*Wall*) könnte z.B ein Teil eines Stockwerks (*Level*) eines Gebäudes sein (siehe Listing 3.5).

```
ObjectPropertyAssertion( :part_of :Wall :Level )
```

Listing 3.5: OWL-2-DL Eigenschaften

Eigenschaften können, wie auch Klassen, in Hierarchien modelliert werden (siehe Listing 3.6).

```
SubObjectPropertyOf( :part_of :part_of_transitive )
```

Listing 3.6: OWL-2-DL Eigenschaften Hierarchie

Zusätzlich ist es möglich, den Definitions- und Wertebereich einer Eigenschaft zu definieren (siehe Listing 3.7).

```
ObjectPropertyDomain( :part_of :Spatial_Entity )
ObjectPropertyRange( :part_of :Spatial_Entity )
```

Listing 3.7: OWL-2-DL Eigenschaften Definitions- und Wertebereiche

- Datenwerte (data properties):

Mit Dateneigenschaften können Individuen Datenwerte zugeordnet werden. Das Individuum Wand (*Wall*) könnte z.B eine Länge haben (siehe Listing 3.8).

```
DataPropertyAssertion( :length :Wall "6"^^xsd:integer )
```

Listing 3.8: OWL-2-DL Dateneigenschaften

Zusätzlich ist es möglich, den Definitions- und Wertebereich, analog zu den Objekteigenschaften, zu definieren.

- Individuen & Instanzen:

Mit Individuen können Entitäten aus der realen Welt repräsentiert und Aussagen über diese Entitäten mit OWL-2-DL modelliert werden.

Individuen können Instanzen von Klassen sein. In den Listings wurde mehrfach das Individuum Wand (*Wall*) verwendet, das z.B. eine Instanz der Klasse *Spatial_Entity* ist.

Über diese Grundelemente hinaus bietet OWL-2-DL die Möglichkeit komplexere Klassen durch Klassenschnitt, -vereinigung oder Komplementbildung aufzubauen (siehe Listing 3.9).

```
EquivalentClasses (
  :Mother
  ObjectIntersectionOf ( :Woman :Parent )
)
EquivalentClasses (
  :Parent
  ObjectUnionOf ( :Mother :Father )
)
EquivalentClasses (
  :ChildlessPerson
  ObjectIntersectionOf (
    :Person
    ObjectComplementOf ( :Parent )
  )
)
```

Listing 3.9: OWL-2-DL komplexere Klassen

Objekteigenschaften und Datenwerte können weiter charakterisiert werden.

- Inverse:

```
InverseObjectProperties( :part_of :has_part )
```

Listing 3.10: OWL-2-DL Inverse

- Symmetrisch:

```
SymmetricObjectProperty( :related_topic )
```

Listing 3.11: OWL-2-DL Symmetrisch

- Asymmetrisch:

```
AsymmetricObjectProperty( :part_of )
```

Listing 3.12: OWL-2-DL Asymmetrisch

- Reflexiv:

```
ReflexiveObjectProperty( :related_topic )
```

Listing 3.13: OWL-2-DL Reflexiv

- Irreflexiv:

```
IrreflexiveObjectProperty( :part_of )
```

Listing 3.14: OWL-2-DL Irreflexiv

- Funktional (1 zu N):

```
FunctionalObjectProperty( :part_of )  
InverseFunctionalObjectProperty( :has_part )
```

Listing 3.15: OWL-2-DL Funktional

- Transitiv:

```
TransitiveObjectProperty( :part_of_transitive )
```

Listing 3.16: OWL-2-DL Transitiv

- Property Chains

```
SubObjectPropertyOf(  
ObjectPropertyChain( :hasParent :hasParent )  
:hasGrandparent  
)
```

Listing 3.17: OWL-2-DL Property Chains

Eine Zusammenfassung ist im Anhang in Tabelle C.1 gegeben, die sich auf die im Datenmodell verwendeten Elemente beschränkt.

Ein weiterer Einsatzzweck der Realisierung des Datenmodells in OWL-2-DL kann eine operative Datenhaltung sein. Für einen operativen Einsatz muss das Datenmodell um weitere Eigenschaften wie z.B. den Namen eines Gebäudeteils, eine weiterführende Beschreibung, einen Dateinamen für ein digitales Dokument oder weitere Eigenschaften, die zu den einzelnen Individuen gespeichert werden sollen oder müssen, erweitert werden.

Hierbei kann man sich bestehender Standards wie z.B. Dublin Core [KB07], den es auch im OWL-2-DL Format gibt [Rei09], bedienen, um allgemeingültige Eigenschaften zu beschreiben oder FOAF Vokabular [BM14] verwenden um Personen und deren Beziehungen zu beschreiben. Eine Erweiterung der Dublin Core Eigenschaften können in „DCMI Metadata Terms“ und „DCMI Type Vocabulary“ [Boa12] gefunden werden.

Wird OWL-2-DL als Datenhaltung eingesetzt, bieten sich folgende Vorteile:

- **Mehrsprachigkeit**
Die Möglichkeit in OWL textartige Werte (Literals) mit einer Sprache auszuzeichnen macht es sehr einfach, sprachabhängige Werte zu repräsentieren und mit diesen zu arbeiten. Eine Anpassung des Datenmodells auf Mehrsprachigkeit ist nicht notwendig.

- **Erweiterbarkeit**
In OWL gibt es keine strikte Trennung zwischen Schema und Instanz, sondern nur eine Trennung zwischen den Klassen, Properties und Individuen.

Somit ist eine Erweiterung des Datenmodells einfacher zu realisieren als z.B. in einer relationalen Instanziierung des hier vorgestellten Datenmodells. Letztere würde in den meisten Fällen eine Schemaänderung notwendig machen, um neue Aspekte in das Datenmodell zu integrieren.

Die hier vorgestellte Einteilung in Schema- und Instanzontologien dient der Übersichtlichkeit und Partitionierung des Datenmodells und ist nicht eine Eigenschaft der OWL Repräsentation.

- **Kombinierbarkeit**
Durch die Realisierung in OWL eröffnet sich nicht nur, wie vorher beschrieben, die Möglichkeit der Anwendbarkeit von Ontologie-Abbildungsverfahren, sondern es ermöglicht eine gemeinsame Art der Repräsentation von Daten und so einen flexibleren Umgang und Interoperabilität mit diesen Daten. Dies war letztendlich genau das Ziel von OWL: eine Repräsentation für semantische Web-Daten zu schaffen.

3.4.5. Konsistenz und Zusammenspiel der Ontologien

Die Formulierung des erweiterten Datenmodells in Beschreibungslogik als Ontologien erlaubt es uns neben der Herstellung einer Grundlage für die Anwendbarkeit von bestehenden Verfahren zusätzlich die Überprüfung des Datenmodells auf Konsistenz.

Die Schemaontologien Typ-, Raum-, Thema- und Objektontologie wurden mit Protégé [Mus15] in der Version 5.2.0 definiert. Jede Teilontologie wurde dabei auf Konsistenz geprüft. Für die Prüfung wurde der in Protégé integrierte OWL-2-DL Reasoner Pellet [Sir+07; Sir17] in der Version 2.2.0 verwendet. Abschließend wurden die Daten aus den MonArch-Archiv St. Lorenz und St. Sebald in die jeweiligen Instanzontologien migriert. Zusammen mit den Instanzontologien und den Schemaontologien wurde die Konsistenz für eine Ontologie aus allen Teilontologien geprüft.

In Abbildung 3.16 sind Schema- und Instanzontologien in Metadaten-Ebenen unterteilt.

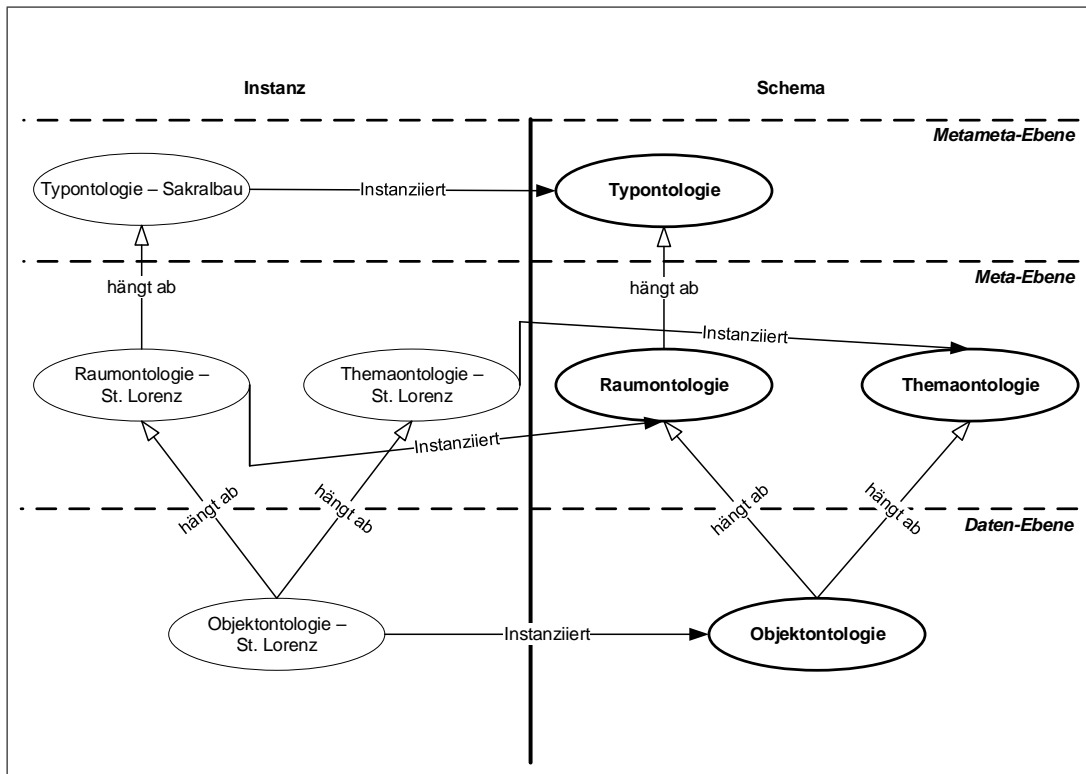


Abbildung 3.16.: Zusammenspiel der Ontologien

In der untersten Ebene, der Daten-Ebene, ist auf der Schemaseite rechts die Objektontologie. Die Objektontologie ist von der Raumontologie und der Themaontologie abhängig.

Raumontologie und Themaontologie finden sich auf der Metadaten-Ebene wieder, da sie zu den Informationsobjekten das Schema der räumlichen und thematischen Metadaten beschreiben.

Die Raumontologie hängt von der Typontologie ab, die sich auf der Metametadata-Ebene befindet, da sie zu den Gebäudeteilen der Raumontologie weitere Daten in Form von Typen speichert.

Zu jeder Schemaontologie gibt es eine Instanzontologie auf derselben Ebene. Die Instanzontologien sind von ihrer jeweiligen Schemaontologie abhängig und instanziiieren diese für z.B. ein Gebäude.

Zusätzlich sind die Instanzontologien von den Instanzontologien auf einer höheren Meta-Ebene abhängig.

Für den MonArch-Anwendungsfall sind z.B. die Dokumente des MonArch-Archivs St. Lorenz in der Objektontologie repräsentiert. Die Strukturhierarchie von St. Lorenz ist in der Raumontologie und die Themenhierarchie in der Themaontologie gespeichert.

Die Strukturtypen eines Sakralbaus sind in der Typontologie gespeichert.

3.5. Fazit

Die Erweiterung des MonArch-Datenmodells war notwendig, um die zusätzlichen Anforderungen einer Verbundabfrage abzudecken. Die Haupteinwirkung ist dabei die Einführung von Typen für Gebäudeteilen oder Bauteilen.

Zusätzlich konnte durch die Erweiterung des Datenmodells eine Verallgemeinerung erreicht werden, die neue Einsatzmöglichkeiten eröffnet.

Hierbei erlaubt der Übergang von einer Strukturhierarchie zum Strukturgraphen eine verfeinerte Abbildung der Gebäude und öffnet das Datenmodell für Daten aus dem Bereich BIM.

Der Übergang zum Themengraphen erlaubt eine verfeinerte Modellierung und Umgang mit Themen.

Die Erweiterung des dokumentenzentrierten MonArch-Datenmodells zu einem objektzentrierten Datenmodell erlaubt die Auszeichnung von Gebäudeteilen und Themen mit den gleichen Methoden wie es das MonArch-Datenmodell für Dokumente ermöglicht.

Die zusätzliche Beschreibung des erweiterten Datenmodells in Beschreibungslogik eröffnete die Prüfung des Datenmodells auf Konsistenz und die Spezifikation von weiteren Eigenschaften die im ER-Modell nicht spezifiziert werden konnten.

Als Ergebnis kann, je nach Anwendungsgebiet und Einsatzzweck, das erweiterte Datenmodell im Relationen Modell oder in OWL-2-DL realisiert werden. Dies erlaubt den Rückgriff auf etablierte Datenbanktechnologie, ohne auf die Vorzüge einer Modellierung mit Beschreibungslogik zu verzichten.

4. Anfragemodell

Um dem eigentlichen Ziel der Anfragetransformation und -relaxation näher zu kommen bedarf es einer Möglichkeit Anfragen kompakt zu formulieren. Dieser Bedarf wird sofort ersichtlich, wenn wir uns eine Anfrage aus dem MonArch-Anwendungsfall in SQL anschauen:

„Suche nach Dokumenten, die direkt am Westbau mit Türmen oder an Gebäudeteilen aus denen der Westbau besteht verortet sind. Zusätzlich sollen nur Informationen zum Material Naturstein und die Restaurierungsphase von Schmitz und Schultz betrachtet werden!“

Für die Übersetzung der Anfrage in SQL (siehe Listing 4.1) gehen wir davon aus, dass das Gebäudeteil *Westbau mit Türmen* die ID **SE-3** hat und die Themen *Naturstein* und *Restaurierungsphase von Schmitz und Schultz* die IDs **T-4** und **T-10** haben.

```
SELECT DISTINCT *
FROM Object_Entity
JOIN
(
  (
    (
      SELECT spatial_reference.Object_Entity_ID AS Object_Entity_ID
      FROM spatial_reference
      JOIN
        Spatial_Entity
      ON spatial_reference.Spatial_Entity_ID = Spatial_Entity.ID
      WHERE Spatial_Entity.ID = SE-3
    )
    UNION
    (
      SELECT spatial_reference.Object_Entity_ID AS Object_Entity_ID
      FROM spatial_reference
      JOIN
        transitive_spatial_part_of
      ON spatial_reference.Spatial_Entity_ID = transitive_spatial_part_of.Ancestor_ID
      JOIN
        Spatial_Entity
      ON transitive_spatial_part_of.ID = Spatial_Entity.ID
      WHERE Spatial_Entity.ID = SE-3
    )
  )
)
INTERSECT
(
  (
    (
      SELECT topical_reference.Object_Entity_ID AS Object_Entity_ID
      FROM topical_reference
      JOIN
        Topic
```

```
    ON topical_reference.Topic_ID = Topic.ID
  WHERE Topic.ID = T-4
)
UNION
(
  SELECT topical_reference.Object_Entity_ID AS Object_Entity_ID
  FROM topical_reference
  JOIN
    transitive_topical_superTopic
  ON topical_reference.Topic_ID = transitive_topical_superTopic.Ancestor_ID
  JOIN
    Topic
  ON transitive_topical_superTopic.ID = Topic.ID
  WHERE Topic.ID = T-4
)
)
INTERSECT
(
  (
    SELECT topical_reference.Object_Entity_ID AS Object_Entity_ID
    FROM topical_reference
    JOIN
      Topic
    ON topical_reference.Topic_ID = Topic.ID
    WHERE Topic.ID = T-10
  )
  UNION
  (
    SELECT topical_reference.Object_Entity_ID AS Object_Entity_ID
    FROM topical_reference
    JOIN
      transitive_topical_superTopic
    ON topical_reference.Topic_ID = transitive_topical_superTopic.Ancestor_ID
    JOIN
      Topic
    ON transitive_topical_superTopic.ID = Topic.ID
    WHERE Topic.ID = T-10
  )
)
)
) AS spatial_and_topical
ON Object_Entity.ID = spatial_and_topical.Object_Entity_ID
```

Listing 4.1: Standard-MonArch-Anfrage in SQL bezogen auf das erweiterte Datenmodell

Diese Anfrage könnte man mit einem SQL-Parser in ihre Einzelteile zerlegen und dann gezielt Teile davon durch ein Transformationsverfahren ersetzen.

Diese Möglichkeit wird jedoch aus den folgenden Gründen nicht weiterverfolgt:

- **Komplexität**
SQL bietet eine Vielzahl an Sprachkonstrukten und -möglichkeiten. Daraus ergeben sich oftmals mehrere mögliche Arten ein- und dieselbe Anfrage zu formulieren. Dies bedeutet für eine Transformation, dass alle diese Möglichkeiten berücksichtigt werden müssen. Eine Folge daraus ist ein sehr komplexes und unter Umständen sehr unübersichtliches Transformationsverfahren.

- **Abhängigkeit**
SQL ist eng mit dem Relationen-Modell verknüpft. Daraus ergibt sich eine Abhängigkeit von der Realisierung des Datenmodells. Wird als Realisierung des hier beschriebenen Datenmodells das Relationen-Modell gewählt, sollte idealerweise SQL als Anfragesprache gewählt werden. Somit muss eine Anfragetransformation auf SQL definiert werden. Entscheidet man sich aber für eine Realisierung des Datenmodells in OWL-2-DL sollte auch SPARQL als Anfragesprache gewählt werden. Möchte man beide Möglichkeiten zulassen müsste eine Anfragetransformation zusätzlich auf SPARQL definiert werden.

Mit dem Ziel diesen Nachteilen einer direkten Realisierung in SQL oder SPARQL entgegenzuwirken, wird im Folgenden eine Anfragesprache vorgestellt, die für den Zweck der Anfragetransformation optimiert ist und eine kompakte Formulierung der Anfragen zulässt.

Die in diesem Kapitel vorgestellte Anfragesprache wird im Folgenden als MonQL bezeichnet. In MonQL sieht die SQL Anfrage aus Listing 4.1 folgendermaßen aus (siehe Listing 4.2).

```
SELECT DISTINCT *
FROM
(
    SPATIAL ID SE-3 DOWN
)
INTERSECT
(
    TOPICAL ID T-4 DOWN
    INTERSECT
    TOPICAL ID T-10 DOWN
)
```

Listing 4.2: Standard-MonArch-Anfrage in MonQL bezogen auf das erweiterte Datenmodell

4.1. Anforderungen an MonQL

Ausgangspunkt für die Definition der hier vorgestellten Anfragesprache sind folgende Anforderungen:

- **Abbildung MonArch-Anwendungsfall**
Die hier definierte Anfragesprache muss mindestens die Varianten der im MonArch-Anwendungsfall beschriebenen Anfragen abdecken.
- **Unabhängigkeit von der eigentlichen Realisierung**
Die Anfragesprache muss unabhängig von der gewählten Realisierung sein. Eine Transformierbarkeit der Anfragen der hier definierten Sprache in SQL und SPARQL muss gewährleistet sein, um eine Einsetzbarkeit zu ermöglichen.
- **Vorbereitung Verbundanfrage**
Zur Realisierung der Verbundanfrage muss diese mithilfe der Anfragesprache formulierbar sein.

- Transformation
Auf Basis der Anfragesprache müssen Regeln zur Transformation definiert werden können.
- Einschränkung der Ergebnisse
Neben der Möglichkeit räumliche als auch thematische Anfragen und deren Kombination zu stellen, müssen Anfrageergebnisse durch weitere Filter eingeschränkt werden können.

4.2. Grundstruktur von MonQL

Die Grundstruktur von MonQL orientiert sich an SQL mit dem Unterschied, dass sich die FROM-Klausel nicht auf Tabellen, Views oder Unteranfragen bezieht, sondern auf räumliche und thematische Selektoren, welche die Anfrageergebnisse beschreiben.

Für MonQL wurde eine LL(*) Grammatik [PF11] mit ANTLR 3 [Par12] definiert die im Anhang D als EBNF (Erweiterte Backus-Naur-Form) zu finden ist. Ausgehend von der MonQL Grammatik wurde mit ANTLR ein Parser erzeugt, mit dem die Übersetzung von MonQL nach SQL und SPARQL realisiert wurde.

Die Semantik von MonQL führen wir auf SQL zurück.

Die Ausdrucksmächtigkeit der hier vorgestellten Anfragesprache ist auf das Nötigste reduziert, um einerseits die Transformation durchführen zu können und um andererseits auch für den MonArch-Anwendungsfall reichhaltig genug zu sein.

Auf Basis des erweiterten Datenmodells betrachten wir nun in SQL räumliche und thematische Anfragen. Dabei ist das Ergebnis einer Anfrage immer Elemente bzw. dessen Attribute vom Entity-Typ **Object Entity**. Diese Einschränkung erlaubt es, eine Grundstruktur einer Anfrage in SQL zu definieren, die sich sowohl räumliche als auch thematische Anfragen teilen.

Im Folgenden wir bei den Definitionen von SQL-Anfragen bei Optionen oder Alternativen die EBNF-Notation verwendet.

Definition 4.1 [Grundstruktur von Anfragen auf Basis des erweiterten Datenmodells]

Die Grundstruktur der Anfrage ist in SQL wie folgt definiert:

```
SELECT (DISTINCT)? (*|Object_Entity.<attribute>
                        (, Object_Entity.<attribute>)*
FROM Object_Entity
JOIN (
  (<spatial>|<topical>|<type>)
  ((UNION|INTERSECT|EXCEPT)
  (<spatial>|<topical>|<type>))*
  ) AS spatial_and_topical
ON Object_Entity.ID = spatial_and_topical.Object_Entity_ID
(WHERE <where>)?
(ORDER BY <order>)?
(LIMIT <Integer>)?
```

Die SQL-Anfrage enthält folgende Platzhalter:

<attribute>: In der Projektion der Anfrage können entweder alle Attribute des Entity-Typs **Object Entity** oder ausgewählte Attribute angegeben werden.

<spatial>: Dies ist ein Platzhalter für eine räumliche Unterabfrage (siehe Abschnitt 4.3).

<topical>: Dies ist ein Platzhalter für eine thematische Unterabfrage (siehe Abschnitt 4.4).

<type>: Dies ist ein Platzhalter für eine Typ-Unterabfrage (siehe Abschnitt 4.5).

<where>: In der Selektion können optional weitere Einschränkungen über die Attribute des Entity-Typs **Object Entity** formuliert werden.

<order>: Eine Sortierung kann optional über die Attribute des Entity-Typs **Object Entity** formuliert werden.

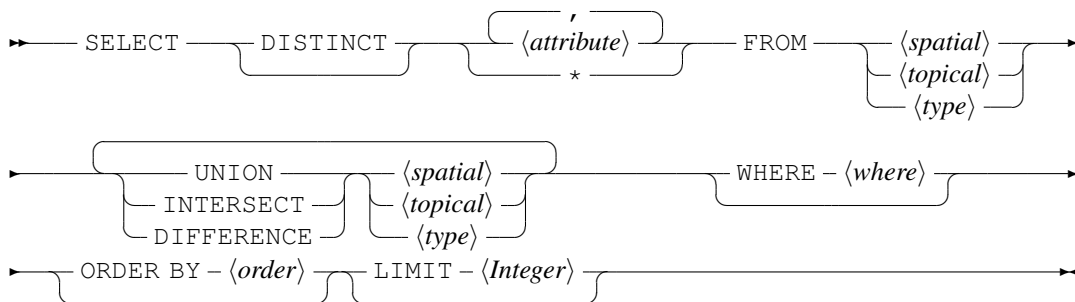
<Integer>: Optional kann die Anzahl der Ergebnisse beschränkt werden.

□

MonQL übernimmt die Grundstruktur wie sie in Definition 4.1 definiert wurde. Daraus ergibt sich die in Definition 4.2 definierte Struktur.

Definition 4.2 [Syntaxdiagramm MonQL]

MonQL :=



Die Grundstruktur von MonQL enthält dieselben Platzhalter wie die Grundstruktur der korrespondierenden SQL-Anfrage.

□

Beispiel 4.3 [Anfrage MonArch-Anwendungsfall]

In natürlicher Sprache könnte die Anfrage wie folgt lauten:

„Suche nach den ersten fünf Dokumenten sortiert nach der Beschriftung, die direkt mit dem Westbau mit Türmen verknüpft sind und deren Autor Schmitz ist!“

Für den räumlichen Teil steht in der Anfrage ein Platzhalter, auf den wir im Folgenden (Abschnitt 4.3) näher eingehen.

```
SELECT DISTINCT caption, author
FROM <spatial>
WHERE author = "Schmitz"
ORDER BY caption
LIMIT 5
```

Diese wird in SQL übersetzt:

```
SELECT DISTINCT Object_Entity.caption, Object_Entity.author
FROM Object_Entity
JOIN (<spatial>) AS spatial_and_topical
ON Object_Entity.ID = spatial_and_topical.Object_Entity_ID
WHERE Object_Entity.author = "Schmitz"
ORDER BY Object_Entity.caption
LIMIT 5
```

□

Ähnlich wie bei SQL beginnt eine Anfrage in MonQL mit einem „SELECT“ gefolgt von der Möglichkeit Duplikate bereits bei der Anfrage auszuschließen. Durch den „*“-Operator oder durch die Aufzählung von Attributnamen, wie z.B. Dateiname eines Dokuments oder Beschriftung eines Objekts, kann analog zu SQL eine Projektion auf einzelne oder alle verfügbaren Eigenschaften durchgeführt werden. Nach der Projektion folgt wie in SQL die FROM-Klausel.

In der FROM-Klausel können drei Anfragearten:

- die räumliche Anfrage (siehe Abschnitt 4.3),
- die thematische Anfrage (siehe Abschnitt 4.4)
- und eine typbasierte Anfrage mit Strukturtypen (siehe Abschnitt 4.5)

beliebig mit Vereinigung (UNION), Schnitt (INTERSECTION) und Differenz (DIFFERENCE) kombiniert werden. Die FROM-Klausel von MonQL vereinfacht die Kombination der Anfragearten, indem im Vergleich zu SQL der Join nicht explizit gemacht werden muss. Dies ist die erste Vereinfachung, die MonQL einführt gegenüber der korrespondierenden SQL-Anfrage. Bei der Übersetzung von MonQL nach SQL wird der Join automatisch eingeführt.

Abbildungsdefinition 4.4 [Übersetzung MonQL in SQL - Grundstruktur]

Die Grundstruktur einer Anfrage in MonQL:

```
SELECT (DISTINCT)? (*|<attribute>(, <attribute>)*
FROM
(<spatial>|<topical>|<type>)
((UNION|INTERSECT|DIFFERENCE)
(<spatial>|<topical>|<type>))*
(WHERE <where>)?
(ORDER BY <order>)?
(LIMIT <Integer>)?
```

wird in SQL übersetzt:


```

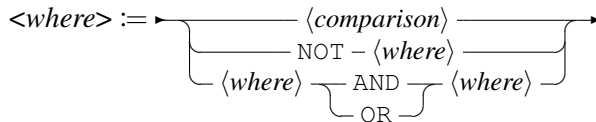
SELECT (DISTINCT)? (*|Object_Entity.<attribute>
                    (, Object_Entity.<attribute>)*
FROM Object_Entity
JOIN (
  (<spatial>|<topical>|<type>)
  ((UNION|INTERSECT|EXCEPT)
  (<spatial>|<topical>|<type>))*
  ) AS spatial_and_topical
ON Object_Entity.ID = spatial_and_topical.Object_Entity_ID
(WHERE <where>)?
(ORDER BY <order>)?
(LIMIT <Integer>)?

```

Bei der Übersetzung werden die Attribute in der Projektion in MonQL auf die Attribute in SQL abgebildet. Die Kombinationsmöglichkeiten UNION, INTERSECT und DIFFERENCE werden von MonQL nach SQL übernommen. Hier ist zu beachten das DIFFERENCE in EXCEPT abgebildet wird. □

Der FROM-Klausel kann in MonQL eine WHERE-Klausel (siehe Definition 4.5) folgen.

Definition 4.5 [Syntaxdiagramm der WHERE-Klausel in MonQL]



□

Die WHERE-Klausel beschränkt sich auf einen Vergleich mit „<, <=, =, =>, >“ von Attributen des Entity-Typs **Object Entity** und seinen Subtypen. Die Übersetzung in SQL ist daher ohne Einschränkung Möglich.

Abbildungsdefinition 4.6 [Übersetzung MonQL in SQL - WHERE-Klausel]

Die WHERE-Klausel einer Anfrage in MonQL:

```

WHERE (<attribute> (<|<=| |=|>|>) <value>
|NOT <where>
|<where> (AND |OR) <where>

```

wird in SQL übersetzt:

```

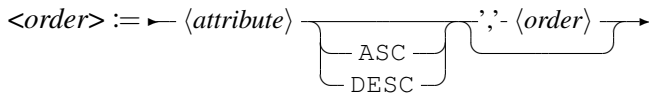
WHERE (Object_Entity.<attribute> (<|<=| |=|>|>) <value>
|NOT <where>
|<where> (AND |OR) <where>

```

Bei der Übersetzung werden die Attribute in der Selektion n in MonQL auf die Attribute in SQL abgebildet. Die Kombinationsmöglichkeiten AND, OR und NOT werden von MonQL nach SQL übernommen. □

Eine Sortierung der Ergebnisse in aufsteigender und absteigender Richtung kann mit der ORDERBY-Klausel (siehe Definition 4.7) nach der WHERE-Klausel erreicht werden. Hierbei können dieselben Attribute verwendet werden wie bei der WHERE-Klausel.

Definition 4.7 [Syntaxdiagramm der ORDERBY-Klausel]



□

Eine Übersetzung der ORDERBY-Klausel von MonQL in SQL ist ohne größere Anpassungen möglich.

Abbildungsdefinition 4.8 [Übersetzung MonQL in SQL - ORDERBY-Klausel]

Die ORDERBY-Klausel einer Anfrage in MonQL:

```
ORDER BY <attribute> ( |ASC|DESC) (, <attribute> ( |ASC|DESC)) *
```

wird in SQL übersetzt:

```
ORDER BY Object_Entity.<attribute> ( |ASC|DESC)
(, Object_Entity.<attribute> ( |ASC|DESC)) *
```

□

Abgeschlossen wird die Anfrage in MonQL durch die Möglichkeit, die Anzahl der Ergebnisse auf eine bestimmte Zahl zu limitieren. Hierbei wird analog zu SQL das Schlüsselwort „LIMIT“ gefolgt von einem Integerwert verwendet.

Abbildungsdefinition 4.9 [Übersetzung MonQL in SQL - LIMIT-Klausel]

Die LIMIT-Klausel einer Anfrage in MonQL:

```
LIMIT <Integer>
```

wird in SQL übersetzt:

```
LIMIT <Integer>
```

□

4.3. Räumliche Anfrage

Eine räumliche Anfrage kann als nicht korrelierte SQL-Unteranfrage definiert werden (siehe Definition 4.10)

Definition 4.10 [Struktur räumliche SQL-Unteranfrage]

Eine räumliche SQL-Unteranfrage ist wie folgt definiert:

```

SELECT spatial_reference.Object_Entity_ID AS Object_Entity_ID
FROM spatial_reference
JOIN Spatial_Entity
ON spatial_reference.Spatial_Entity_ID = Spatial_Entity.ID
WHERE (Spatial_Entity.ID = <ID>
|Spatial_Entity.caption = '<String>' | LIKE '<Pattern>')
(
UNION
SELECT spatial_reference.Object_Entity_ID AS Object_Entity_ID
FROM spatial_reference
JOIN transitive_spatial_part_of
ON spatial_reference.Spatial_Entity_ID = transitive_spatial_part_of.Ancestor_ID
JOIN Spatial_Entity
ON transitive_spatial_part_of.ID = Spatial_Entity.ID
WHERE (Spatial_Entity.ID = <ID>
|Spatial_Entity.caption = '<String>' | LIKE '<Pattern>')
)?
(
UNION
SELECT spatial_reference.Object_Entity_ID AS Object_Entity_ID
FROM spatial_reference
JOIN transitive_spatial_part_of
ON spatial_reference.Spatial_Entity_ID = transitive_spatial_part_of.ID
JOIN Spatial_Entity
ON transitive_spatial_part_of.Ancestor_ID = Spatial_Entity.ID
WHERE (Spatial_Entity.ID = <ID>
|Spatial_Entity.caption = '<String>' | LIKE '<Pattern>')
)?

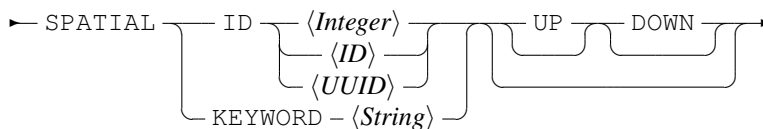
```

Eine räumliche Anfrage enthält zwei Varianten der Selektion von Elementen des Entity-Typs **Spatial Entity**. Zusätzlich gibt es zwei Möglichkeiten die Beziehungen **part-of-** oder **consists-of** bei der Anfrage mit einzubeziehen.

Eine räumliche Anfrage ist in MonQL wie folgt (siehe Definition 4.11) definiert.

Definition 4.11 [Syntaxdiagramm der spatial-Klausel]

<spatial> :=



Eine räumliche Anfrage wird durch das Schlüsselwort „SPATIAL“ eingeleitet, gefolgt von zwei Möglichkeiten, um Elemente im Strukturgraphen zu identifizieren bzw. auszuwählen.

Die Möglichkeiten sind:

Selektion per ID:

Jede räumliche Entität (`Spatial_Entity`) lässt sich per ID eindeutig identifizieren. Abhängig von der Implementierung ist das eine Integer-Zahl, ein spezieller ID-Datentyp oder eine global eindeutige ID.

Abbildungsdefinition 4.12 [Übersetzung MonQL in SQL - räumliche Selektion per ID]

Eine räumliche Anfrage in MonQL die per ID selektiert:

```
SPATIAL ID (<Integer>|<ID>|<UUID>)
```

wird in SQL übersetzt:

```
SELECT spatial_reference.Object_Entity_ID AS Object_Entity_ID
FROM spatial_reference
JOIN Spatial_Entity
ON spatial_reference.Spatial_Entity_ID = Spatial_Entity.ID
WHERE (Spatial_Entity.ID = <Integer>
|Spatial_Entity.ID = <ID>
|Spatial_Entity.UUID = <UUID>)
```

□

Selektion per KEYWORD:

Die Selektion per KEYWORD erlaubt die Selektion einer räumlichen Entität über den Namen. Da der Name nicht eindeutig sein muss, kann bei dieser Art der Identifizierung eine mehrfache Selektion auftreten.

Abbildungsdefinition 4.13 [Übersetzung MonQL in SQL - räumliche Selektion per KEYWORD]

Eine räumliche Anfrage in MonQL die per KEYWORD selektiert:

```
SPATIAL KEYWORD <Keyword>
```

wird in SQL übersetzt:

```
SELECT spatial_reference.Object_Entity_ID AS Object_Entity_ID
FROM spatial_reference
JOIN Spatial_Entity
ON spatial_reference.Spatial_Entity_ID = Spatial_Entity.ID
WHERE Spatial_Entity.caption ( = '<Keyword>' | LIKE '<Keyword>' )
```

□

Die Variante der Selektion per ID und per KEYWORD können mit dem Entitätstyp **Spatial Entity** in SQL realisiert werden.

Beispiel 4.14 [räumliche Selektion]

Die MonArch-Anfrage:

„Suche nach Dokumenten, die direkt mit dem Westbau mit Türmen verknüpft sind!“

lässt sich unter der Annahme, dass der Gebäudeteil „Westbau mit Türmen“ die ID SE-3 hat, durch eine räumliche Anfrage (siehe Listing 4.3) mit Selektion des Gebäudeteils über dessen ID formulieren.

```
SELECT DISTINCT caption, author
FROM
  SPATIAL ID SE-3
```

Listing 4.3: Beispiel räumliche Anfrage per ID in MonQL

Die Selektion des Gebäudeteils kann alternativ über den Namen durchgeführt werden. Hierbei könnten mehrere Gebäudeteile mit demselben Namen selektiert werden (siehe Listing 4.4).

```
SELECT DISTINCT caption, author
FROM
  SPATIAL KEYWORD "Westbau mit Tuermen"
```

Listing 4.4: Beispiel räumliche Anfrage per KEYWORD in MonQL

□

Die Möglichkeit den Suchraum entlang der Beziehungen **part-of**- oder **consists-of** zu erweitern, ist durch die UP- und DOWN-Sprachelemente realisiert. In folgendem Übersetzungen ist nur die Variante für die **part-of**-Hierarchie gezeigt. Die Variante für die **consists-of**-Hierarchie wird analog dazu behandelt.

Die Möglichkeiten sind:

UP-Variante:

Bei dieser Variante werden im Strukturgraphen ausgehen von der Selektion, Elemente auf dem Pfad entlang der Rolle *part_of* betrachtet.

Abbildungsdefinition 4.15 [Übersetzung MonQL in SQL - räumliche Selektion UP]
Eine räumliche UP-Anfrage in MonQL:

```
SPATIAL ID <ID> UP
```

wird in SQL übersetzt:

```
SELECT spatial_reference.Object_Entity_ID AS Object_Entity_ID
FROM spatial_reference
JOIN Spatial_Entity
ON spatial_reference.Spatial_Entity_ID = Spatial_Entity.ID
WHERE Spatial_Entity.ID = <ID>
UNION
SELECT spatial_reference.Object_Entity_ID AS Object_Entity_ID
FROM spatial_reference
JOIN transitive_spatial_part_of
ON spatial_reference.Spatial_Entity_ID = transitive_spatial_part_of.ID
JOIN Spatial_Entity
ON transitive_spatial_part_of.Ancestor_ID = Spatial_Entity.ID
WHERE Spatial_Entity.ID = <ID>
```

□

Beispiel 4.16 [UP-Variante]

Die MonArch-Anfrage:

„Suche nach Dokumenten, die direkt mit dem Westbau mit Türmen oder Gebäudeteilen, in denen der Westbau als Gebäudeteil vorkommt, verknüpft sind!“

lässt sich ebenfalls in MonQL als Variante der räumlichen Anfrage realisieren (siehe Listing 4.5).

```
SELECT DISTINCT caption, author
FROM
  SPATIAL ID SE-3 UP
```

Listing 4.5: Beispiel räumliche Anfrage per ID mit UP in MonQL

Alternativ wäre hier eine Anfrage über den Namen möglich.

DOWN-Variante:

Bei dieser Variante werden im Strukturgraphen ausgehen von der Selektion, Elemente auf dem Pfad entlang der Rolle *has_part* betrachtet.

Abbildungsdefinition 4.17 [Übersetzung MonQL in SQL - räumliche Selektion DOWN]

Eine räumliche DOWN-Anfrage in MonQL:

```
SPATIAL ID <ID> DOWN
```

wird in SQL übersetzt:

```
SELECT spatial_reference.Object_Entity_ID AS Object_Entity_ID
FROM spatial_reference
JOIN Spatial_Entity
ON spatial_reference.Spatial_Entity_ID = Spatial_Entity.ID
WHERE Spatial_Entity.ID = <ID>
UNION
SELECT spatial_reference.Object_Entity_ID AS Object_Entity_ID
FROM spatial_reference
JOIN transitive_spatial_part_of
ON spatial_reference.Spatial_Entity_ID = transitive_spatial_part_of.
  Ancestor_ID
JOIN Spatial_Entity
ON transitive_spatial_part_of.ID = Spatial_Entity.ID
WHERE Spatial_Entity.ID = <ID>
```

Beispiel 4.18 [DOWN-Variante]

Die MonArch-Anfrage:

„Suche nach Dokumenten, die direkt mit dem Westbau mit Türmen oder mit Gebäudeteilen aus denen der Westbau besteht, verknüpft sind!“

kann in MonQL durch eine Variante der räumlichen Anfrage formuliert werden (siehe Listing 4.6). Die Annahme, dass der Gebäudeteil „Westbau mit Türmen“ die ID SE-3 hat, gilt auch hier wieder.

```
SELECT DISTINCT caption, author
FROM
  SPATIAL ID SE-3 DOWN
```

Listing 4.6: Beispiel räumliche Anfrage per ID mit DOWN in MonQL

```
SELECT DISTINCT Object_Entity.caption, Object_Entity.author
FROM Object_Entity
JOIN (
  SELECT spatial_reference.Object_Entity_ID AS Object_Entity_ID
  FROM spatial_reference
  JOIN Spatial_Entity
  ON spatial_reference.Spatial_Entity_ID = Spatial_Entity.ID
  WHERE Spatial_Entity.ID = SE-3
  UNION
  SELECT spatial_reference.Object_Entity_ID AS Object_Entity_ID
  FROM spatial_reference
  JOIN transitive_spatial_part_of
  ON spatial_reference.Spatial_Entity_ID = transitive_spatial_part_of.
  Ancestor_ID
  JOIN Spatial_Entity
  ON transitive_spatial_part_of.ID = Spatial_Entity.ID
  WHERE Spatial_Entity.ID = SE-3
)
) AS spatial
ON Object_Entity.ID = spatial.Object_Entity_ID
```

Listing 4.7: Räumliche Anfrage per ID mit DOWN in SQL

□

Für beide Varianten muss in der Sicht „transitive_spatial_part_of“ (siehe Definition 4.19) die transitive Hülle der **part-of**-Beziehung berechnet werden.

Definition 4.19 [Sicht:transitive_spatial_part_of]

Die **part-of**-Hierarchie lässt sich rekursiv berechnen:

```
CREATE VIEW transitive_spatial_part_of AS
WITH Spatial_part_of(ID, Ancestor_ID) AS
(
  SELECT p.ID, p.Parent_ID
  FROM Spatial_Entity p
  WHERE Parent_ID IS NOT NULL
  UNION ALL
  SELECT p.ID, ph.Ancestor_ID
  FROM Spatial_Entity p INNER JOIN Spatial_part_of ph ON p.Parent_ID = ph.ID
)
SELECT ID, Ancestor_ID
FROM Spatial_part_of
```

Listing 4.8: SQL-Sicht für **part-of**-Hierarchie

□

Betrachtet man die beiden Variante aus der Sicht der Raumontologie, erweitert das Schlüsselwort „UP“ den Suchraum, indem räumliche Elemente über die Rollen *part_of*⁺ und *consists_of*⁻⁺ miteinbezogen werden. Das Schlüsselwort „DOWN“ erweitert den Suchraum über die Rolle *has_part*⁺ und *consists_of*⁺.

Die zwei Varianten der Selektion lassen sich mit den beiden Varianten UP und DOWN kombinieren.

Beispiel 4.20 [Kombination UP und DOWN]

Die MonArch-Anfrage:

„Suche nach Dokumenten, die direkt mit dem Westbau mit Türmen, mit Gebäudeteilen aus denen der Westbau besteht oder mit Gebäudeteilen, in denen der Westbau als Gebäudeteil vorkommt, verknüpft sind!“

ist eine Kombination der beiden vorherigen Varianten und lässt sich mit MonQL wie folgt realisieren (siehe Listing 4.9).

```
SELECT DISTINCT caption, author
FROM
  SPATIAL ID SE-3 UP DOWN
```

Listing 4.9: Beispiel räumliche Anfrage per ID mit UP und DOWN in MonQL

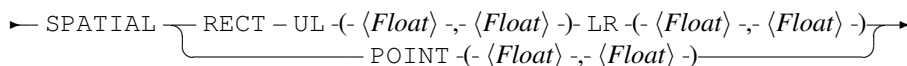
Alternativ wäre hier eine Anfrage über den Namen möglich. □

Die beiden ersten Varianten der Selektion identifizieren die Elemente symbolisch. Es ist auch möglich räumliche Entitäten anhand ihrer Koordinaten zu identifizieren.

Hierfür sind zwei geometrische Auswahlmöglichkeiten vorgesehen: Zum einen durch Angabe eines Punktes, zum anderen durch Angabe eines rechteckigen Anfragefensters. Im Syntaxdiagramm (siehe Definition 4.21) ist der zweidimensionale Fall definiert.

Definition 4.21 [Syntaxdiagramm der koordinatenbasierten Selektion]

<spatial> :=



Für den dreidimensionalen Fall müssen die Koordinaten entsprechend erweitert werden.

Abbildungsdefinition 4.22 [Übersetzung MonQL in SQL - räumliche koordinatenbasierte Selektion]

Eine räumliche koordinatenbasierte Anfrage in MonQL:

```
SPATIAL RECT UL (<Float>,<Float>) LR (<Float>,<Float>)
```

wird in SQL übersetzt:


```

SELECT spatial_reference.Object_Entity_ID AS Object_Entity_ID
FROM spatial_reference
JOIN Spatial_Geometry_Entity
ON spatial_reference.Spatial_Entity_ID = Spatial_Geometry_Entity.ID
WHERE ST_Within(Spatial_Geometry_Entity.Geometry, <Rectangle>)

```

Bei dieser Übersetzung müssen die Angaben des Rechtecks in MonQL in die passende Darstellungsform in SQL umgerechnet werden. □

Für die geometrische Selektion per Koordinaten muss der Entity-Typ **Spatial Geometry Entity** in SQL verwendet werden. Hier wurde die geometrische Operation „ST_Within“ aus der SQL/MM Spezifikation [ISO02; Sto03] oder der OpenGIS-Spezifikation [Her10] verwendet. Für die geometrische Operation können auch alternative Operationen verwendet werden, die an die Bedürfnisse des Anwendungsfalls angepasst sind.

Beispiel 4.23 [Grafische Anfrage]

Betrachten wir einen 2D-Ansichtsplan der Westfassade von St. Lorenz. In Abbildung 4.1 markiert das rote Rechteck unser Anfragerechteck. Die grafischen Elemente im roten Rechteck, hier die Polygone die den **Nordturm** repräsentieren, sind mit der symbolischen Repräsentation des **Nordturm**s verknüpft. Die Anfrage selektiert somit die Polygone und darüber hinaus den **Nordturm** in der Strukturhierarchie. Ausgehend vom **Nordturm** werden die Gebäudeteile auf dem Weg zur Wurzel in die Anfrage miteinbezogen.

Ergebnis der Anfrage sind die Dokumente A bis C.

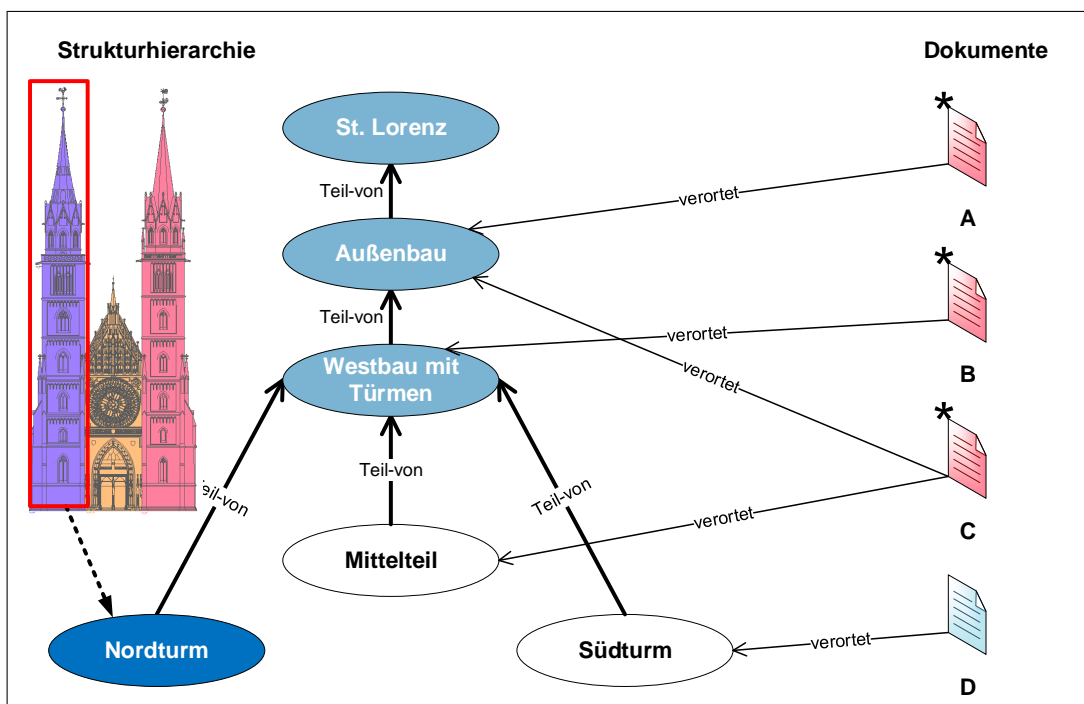


Abbildung 4.1.: Räumliche Anfrage Variante vier am Nordturm von St. Lorenz

Durch die Möglichkeit der koordinatenbasierten Spezifikation der räumlichen Anfrage kann die in Abbildung 4.1 gezeigte Anfrage mit dem roten Rechteck als Anfrage in MonQL (siehe Listing 4.10) umgesetzt werden. (ULX, ULY) sind die Koordinaten der linken oberen Ecke des Rechtecks und (LRX, LRY) sind die Koordinaten der rechten unteren Ecke des Rechtecks.

```
SELECT DISTINCT caption, author
FROM
  SPATIAL RECT UL (ULX, ULY) LR (LRX, LRY) UP DOWN
```

Listing 4.10: Beispiel grafische Anfrage über Koordinaten

□

In Unterabschnitt 2.4.1 werden drei Varianten von räumlichen Anfragen beschrieben, die im MonArch-Anwendungsfall vorkommen. Alle diese Varianten lassen sich mit MonQL umsetzen.

4.4. Thematische Anfrage

Im Gegensatz zum MonArch-Anwendungsfall, in dem thematische Anfragen bei der Selektion eines Themas dessen Unterthemen immer mitberücksichtigen, erlaubt MonQL, ähnlich wie bei den räumlichen Anfragen, vier Varianten der Themenselektion.

Eine thematische Anfrage kann als nicht korrelierte SQL-Unteranfrage definiert werden (siehe Definition 4.24)

Definition 4.24 [Struktur thematische SQL-Unteranfrage]

Eine thematische SQL-Unteranfrage ist wie folgt definiert:

```
SELECT topical_reference.Object_Entity_ID AS Object_Entity_ID
FROM topical_reference
JOIN Topic
ON topical_reference.Topic_ID = Topic.ID
WHERE (Topic.ID = <ID>
      |Topic.caption = '<String>' | LIKE '<Pattern>')
(
UNION
SELECT topical_reference.Object_Entity_ID AS Object_Entity_ID
FROM topical_reference
JOIN transitive_topical_superTopic
ON topical_reference.Topic_ID = transitive_topical_superTopic.Ancestor_ID
JOIN Topic
ON transitive_topical_superTopic.ID = Topic.ID
WHERE (Topic.ID = <ID>
      |Topic.caption = '<String>' | LIKE '<Pattern>')
)?
(
UNION
SELECT topical_reference.Object_Entity_ID AS Object_Entity_ID
FROM topical_reference
JOIN transitive_topical_superTopic
ON topical_reference.Topic_ID = transitive_topical_superTopic.ID
JOIN Topic
```

```

ON transitive_topical_superTopic.Ancestor_ID = Topic.ID
WHERE (Topic.ID = <ID>
      | Topic.caption = '<String>' | LIKE '<Pattern>')
)?

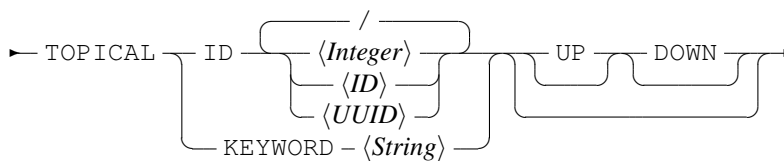
```

Eine thematische Anfrage enthält zwei Varianten der Selektion von Elementen des Entity-Typs **Topic**. Zusätzlich gib es zwei Möglichkeiten die Beziehungen **super topic** bei der Anfrage mit einzubeziehen.

Eine thematische Anfrage ist in MonQL wie folgt (siehe Definition 4.25) definiert.

Definition 4.25 [Syntaxdiagramm der topical-Klausel]

$\langle \text{topical} \rangle :=$



Eine thematische Anfrage wird durch das Schlüsselwort „TOPICAL“ eingeleitet, gefolgt von zwei Varianten ein Thema (**Topic**) zu identifizieren.

Die Möglichkeiten sind:

Selektion per ID:

Jedes Thema (**Topic**) lässt sich per ID eindeutig identifizieren. Abhängig von der Implementierung ist das eine Integer-Zahl, ein spezieller ID-Datentyp oder eine global eindeutige ID.

Abbildungsdefinition 4.26 [Übersetzung MonQL in SQL - thematische Selektion per ID]

Eine thematische Anfrage in MonQL die per ID selektiert:

```

TOPICAL ID (<Integer> | <ID> | <UUID>)

```

wird in SQL übersetzt:

```

SELECT topical_reference.Object_Entity_ID AS Object_Entity_ID
FROM topical_reference
JOIN Topic
ON topical_reference.Topic_ID = Topic.ID
WHERE (Topic.ID = <Integer>
      | Topic.ID = <ID>
      | Topic.UUID = <UUID>)

```

Selektion per KEYWORD:

Da Themen in der Regel zusätzlich zur ID durch einen Begriff beschrieben werden, kann eine Selektion auch per Schlüsselwort erfolgen. Dieser Art der Identifizierung kann zu einer mehrfachen Selektion führen.

Abbildungsdefinition 4.27 [Übersetzung MonQL in SQL - thematische Selektion per KEYWORD]

Eine thematische Anfrage in MonQL die per KEYWORD selektiert:

```
TOPICAL KEYWORD <Keyword>
```

wird in SQL übersetzt:

```
SELECT topical_reference.Object_Entity_ID AS Object_Entity_ID
FROM topical_reference
JOIN Topic
ON topical_reference.Topic_ID = Topic.ID
WHERE Topic.caption ( = '<Keyword>' | LIKE '<Keyword>' )
```

□

Analog zum räumlichen Fall sind hier die unterschiedlichen Möglichkeiten zur Auswahl der Themen durch die Selektion nach ID oder Name umgesetzt.

Eine Besonderheit hierbei ist die Möglichkeit nicht nur ein Thema zu identifizieren, sondern mehrere Themen in einem Art Pfad-Ausdruck. Die Interpretation und somit die Auswirkung auf die Anfrage wird in Kapitel 5 behandelt.

Die UP- und DOWN-Sprachelemente der thematischen Anfrage in MonQL erlauben die Erweiterung des Suchraums über die Themenhierarchie, die durch die **super topic**-Beziehung aufgebaut wird.

Die Möglichkeiten sind:

UP-Variante:

Bei dieser Variante werden im Themengraphen ausgehen von der Selektion, Elemente auf dem Pfad entlang der Rolle *super_topic* betrachtet.

Abbildungsdefinition 4.28 [Übersetzung MonQL in SQL - thematische Selektion UP]

Eine thematische UP-Anfrage in MonQL:

```
TOPICAL ID <ID> UP
```

wird in SQL übersetzt:

```
SELECT topical_reference.Object_Entity_ID AS Object_Entity_ID
FROM topical_reference
JOIN Topic
ON topical_reference.Topic_ID = Topic.ID
WHERE Topic.ID = <ID>
UNION
SELECT topical_reference.Object_Entity_ID AS Object_Entity_ID
FROM topical_reference
```

```

JOIN transitive_topical_superTopic
ON topical_reference.Topic_ID = transitive_topical_superTopic.ID
JOIN Topic
ON transitive_topical_superTopic.Ancestor_ID = Topic.ID
WHERE Topic.ID = <ID>

```

□

DOWN-Variante:

Bei dieser Variante werden im Themengraphen ausgehen von der Selektion, Elemente auf dem Pfad entlang der Rolle *sub_topic* betrachtet.

Abbildungsdefinition 4.29 [Übersetzung MonQL in SQL - thematische Selektion DOWN]

Eine thematische DOWN-Anfrage in MonQL

```
TOPICAL ID <ID> DOWN
```

wird in SQL übersetzt:

```

SELECT topical_reference.Object_Entity_ID AS Object_Entity_ID
FROM topical_reference
JOIN Topic
ON topical_reference.Topic_ID = Topic.ID
WHERE Topic.ID = <ID>
UNION
SELECT topical_reference.Object_Entity_ID AS Object_Entity_ID
FROM topical_reference
JOIN transitive_topical_superTopic
ON topical_reference.Topic_ID = transitive_topical_superTopic.Ancestor_ID
JOIN Topic
ON transitive_topical_superTopic.ID = Topic.ID
WHERE Topic.ID = <ID>

```

□

Die Einbeziehung der Ober- und Unterbegriffe in den Suchraum ist durch die Sicht 'transitive_topical_superTopic' (siehe Definition 4.30) realisiert.

Definition 4.30 [Sicht:transitive_topical_superTopic]

Die **super topic**-Hierarchie lässt sich rekursiv berechnen:

```

CREATE VIEW transitive_topical_superTopic AS
WITH Topical_superTopic(ID, Ancestor_ID) AS
(
  SELECT p.ID, p.Parent_ID
  FROM Topic p
  WHERE Parent_ID IS NOT NULL
  UNION ALL
  SELECT p.ID, ph.Ancestor_ID
  FROM Topic p INNER JOIN Topical_superTopic ph ON p.Parent_ID = ph.ID
)
SELECT ID, Ancestor_ID
FROM Topical_superTopic

```

Listing 4.11: SQL-Sicht für **super topic**-Hierarchie



Der Suchraum wird somit immer über *sub_topic*⁺ und *super_topic*⁺ erweitert.

Das folgende Beispiel zeigt die Grundanfrage aus dem MonArch-Anwendungsfall und die Umsetzung als thematische Anfrage.

Beispiel 4.31 [Thematische Grundanfrage]

Die MonArch-Anfrage

„Suche nach Dokumenten, die Informationen über das Material Naturstein im Zusammenhang mit der Restaurierungsphase der Herren Schmitz und Schultz aufweisen!“

wird in zwei Grundanfragen zerlegt.

Der erste Teil der Anfrage ist in Listing 4.12 realisiert, unter der Annahme, dass das Thema **Naturstein** die ID **T-4** hat.

```
SELECT DISTINCT caption, author
FROM
  TOPICAL ID T-4 DOWN
```

Listing 4.12: Beispiel thematische Anfrage per ID (Naturstein)

Alternativ wäre hier eine Anfrage über den Namen möglich (siehe Listing 4.13).

```
SELECT DISTINCT caption, author
FROM
  TOPICAL KEYWORD "Naturstein" DOWN
```

Listing 4.13: Beispiel thematische Anfrage per KEYWORD (Naturstein)

Die Selektion per **KEYWORD** ist als **SQL-Anfrage** in Listing 4.14 zu sehen.

```
SELECT DISTINCT Object_Entity.caption, Object_Entity.author
FROM Object_Entity
JOIN (
  SELECT topical_reference.Object_Entity_ID AS Object_Entity_ID
  FROM topical_reference
  JOIN Topic
  ON topical_reference.Topic_ID = Topic.ID
  WHERE Topic.Name = 'Naturstein'
  UNION
  SELECT topical_reference.Object_Entity_ID AS Object_Entity_ID
  FROM topical_reference
  JOIN transitive_topical_superTopic
  ON topical_reference.Topic_ID = transitive_topical_superTopicc.Ancestor_ID
  JOIN Topic
  ON transitive_topical_superTopic.ID = Topic.ID
  WHERE Topic.Name = 'Naturstein'
) AS topical
ON Object_Entity.ID = topical.Object_Entity_ID
```

Listing 4.14: Thematische SQL-Anfrage per KEYWORD

Der zweite Teil der Anfrage ist in Listing 4.15 realisiert, unter der Annahme, dass das Thema *Restaurierungsphase der Herren Schmitz und Schultz* die ID **T-10** hat.

```
SELECT DISTINCT caption, author
FROM
  TOPICAL ID T-10 DOWN
```

Listing 4.15: Beispiel thematische Anfrage per ID (Restaurierungsphase)

□

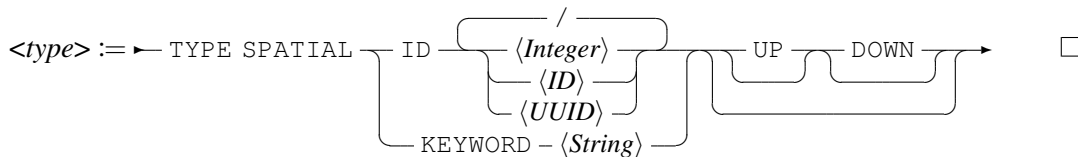
Mit den thematischen Anfragen können alle im MonArch-Anwendungsfall vorkommenden Anfragen abgebildet werden. MonQL bietet darüber hinaus noch weitere Möglichkeiten die Selektion von Ober- und Unterbegriffen zu steuern.

4.5. Typbasierte Anfrage

Eine ganz neue Anfrageart führt die Anfragesprache mit der typbasierten Anfrage ein. Diese neue Art der Anfrage ist essentiell, um die spätere Anfragetransformation für räumliche Anfragen überhaupt durchführen zu können. Dieser Abschnitt greift in gewisser Hinsicht vorab ein Ergebnis auf, dessen Notwendigkeit sich erst im Kapitel 5 erschließt.

Eine typbasierte Anfrage lässt sich wie folgt definieren:

Definition 4.32 [Syntaxdiagramm der type-Klausel]



□

Eingeleitet wird sie mit den Schlüsselwörtern „TYPE SPATIAL“ gefolgt von zwei Varianten, die räumlichen Typen (*Spatial_Entity_Type*) zu identifizieren. Folgende zwei Varianten sind vorgesehen:

Selektion per ID:

Jeder räumliche Typ (*Spatial_Entity_Type*) lässt sich per ID eindeutig identifizieren. Abhängig von der Implementierung ist das eine Integer-Zahl, ein spezieller ID-Datentyp oder eine global eindeutige ID.

Selektion per KEYWORD:

Der räumliche Typ wird durch einen Namen selektiert. Da der Name nicht eindeutig sein muss, kann bei dieser Art der Identifizierung eine mehrfache Selektion auftreten.

Eine Besonderheit hierbei ist die Möglichkeit nicht nur einen Typ zu identifizieren, sondern mehrere Typen in einer Art Pfad-Ausdruck, ähnlich eines XPATH-Ausdruckes aus XML, in einer typbasierten Anfrage zu spezifizieren. Die Interpretation und somit die Auswirkung auf die Anfrage ist Teil von Kapitel 5.

Abgeschlossen wird die typbasierte Anfrage, genau wie die räumliche Anfrage, durch die Möglichkeit entlang der *part_of*- oder *consists_of*-Hierarchie die Anfrage zu erweitern.

Hierzu werden verschiedene Ansätze in Kapitel 5 vorgestellt und diskutiert.

Durch die Optionen „UP“ und „DOWN“ entstehen verschieden Varianten.

Beispiel 4.33 [Typbasierte Anfrage]

Eine Anfrage, die im MonArch-Anwendungsfall noch nicht vorgekommen ist, lässt sich aus der Anforderung, die in Abschnitt 2.5 beschrieben wurde, ableiten.

Im Folgenden Beispiel soll nicht nach einem speziellen Pfeiler sondern nach allen Pfeilern gesucht werden. Die Anfrage kann lauten:

„Suche nach Dokumenten, die direkt an allen Pfeilern im Gebäude vorkommen.“

Listing 4.16 zeigt wie diese Anfrage mit der Anfragesprache realisiert wird:

```
SELECT DISTINCT caption, author
FROM
  TYPE SPATIAL KEYWORD "Pfeiler"
```

Listing 4.16: Beispiel typbasierte Anfrage per KEYWORD

Die Auswertung dieser Anfrage setzt voraus, dass jeder konkrete Pfeiler 1 bis z.B. Pfeiler 5 mit dem Typ Pfeiler ausgezeichnet wurde.

4.6. Anfragekombination

Die einzelnen Anfragearten können mit der hier vorgestellten Anfragesprache MonQL beliebig mit Vereinigung, Schnitt und Differenz kombiniert werden. Dies erlaubt einerseits die Abbildung der Anfragekombinationen aus dem MonArch-Anwendungsfall. Andererseits können neue Arten von Kombinationen formuliert werden, die notwendig sind, um Verbundanfragen mit einer automatischen Anfragetransformation stellen zu können.

Anfragen aus dem MonArch-Anwendungsfall sind Kombinationen einer räumlichen Anfrage mit keiner oder mehreren thematischen Anfragen. Die thematischen Anfragen können dabei entweder mit UNION oder INTERSECT verbunden werden. In Listing 4.17 ist eine derartige Anfragekombination schematisch realisiert.

```
SELECT DISTINCT document.*
FROM (
  SPATIAL ID <ID> UP DOWN
  INTERSECT (
    TOPICAL ID <ID> DOWN
    (( UNION | INTERSECT )
     TOPICAL ID <ID> DOWN ) * ) )
```

Listing 4.17: Anfrageschema aus dem MonArch-Anwendungsfall

Das Beispiel 4.34 zeigt eine MonArch-Anfrage, die bereits in Beispiel 2.14 eingeführt wurde:

Beispiel 4.34 [MonArch-Anfragen in St. Lorenz]

Die Anfrage

„Suche nach Dokumenten, die direkt mit dem Westbau mit Türmen oder mit Gebäudeteilen aus denen der Westbau besteht, verknüpft sind und das Material Naturstein und die Restaurierungsphase der Herren Schmitz und Schultz aufweisen!“

lässt sich, wie in Listing 4.18 gezeigt wird, mit MonQL realisieren. Es gilt wieder die Annahme, dass der **Westbau mit Türmen** die ID **SE-3**, **Naturstein** die ID **T-4** und die **Restaurierungsphase der Herren Schmitz und Schultz** die ID **T-10** hat.

```
SELECT DISTINCT *
FROM
(
  SPATIAL ID SE-3 DOWN
  INTERSECT
  (
    TOPICAL ID T-4 DOWN
    INTERSECT
    TOPICAL ID T-10 DOWN
  )
)
```

Listing 4.18: Anfrage aus dem MonArch-Anwendungsfall

Falls die IDs nicht bekannt sind, könnte die Anfrage alternativ, wie in Listing 4.19 zu sehen, formuliert werden:

```
SELECT DISTINCT *
FROM
(
  SPATIAL KEYWORD "Westbau mit Tuermen" DOWN
  INTERSECT
  (
    TOPICAL KEYWORD "Material Naturstein" DOWN
    INTERSECT
    TOPICAL KEYWORD "Restaurierungsphase der Herren Schmitz und Schultz" DOWN
  )
)
```

Listing 4.19: Anfrage aus dem MonArch-Anwendungsfall

□

Zur Veranschaulichung der Möglichkeiten folgen zwei weitere Beispiele, die aus dem MonArch-Anwendungsfall abgeleitet sind und sich die neuen Möglichkeiten der Anfragesprache MonQL zunutze machen:

Beispiel 4.35 [Kombination zweier räumlicher Anfragen]

Die Anfrage

„Suche nach Dokumenten, die direkt mit dem Nordturm und dem Südturm oder mit Gebäudeteilen aus denen der Nordturm und der Südturm besteht, verknüpft sind und das Material Naturstein und die Restaurierungsphase der Herren Schmitz und Schultz aufweisen!“

lässt sich mit MonQL übersetzen.

Listing 4.20 zeigt unter Zugrundelegung der in Abbildung 2.14 gezeigten Strukturhierarchie und unter der Annahme, dass der Westbau mit Türmen die ID SE-3, Nordturm die ID SE-4, Mittelteil die ID SE-5, Südturm die ID SE-6, Naturstein die ID T-4 und die Restaurierungsphase der Herren Schmitz und Schultz die ID T-10 hat, die realisierte Anfrage.

```
SELECT DISTINCT *
FROM
  (
    (
      SPATIAL ID SE-4 DOWN
      UNION
      SPATIAL ID SE-6 DOWN
    )
    INTERSECT
    (
      TOPICAL ID T-4 DOWN
      INTERSECT
      TOPICAL ID T-10 DOWN
    )
  )
```

Listing 4.20: Neue Anfragemöglichkeiten in MonQL

Listing 4.21 zeigt die Differenz zweier räumlicher Anfragen:

```
SELECT DISTINCT *
FROM
  (
    (
      SPATIAL ID SE-3 DOWN
      DIFFERENCE
      SPATIAL ID SE-5 DOWN
    )
    INTERSECT
    (
      TOPICAL ID T-4 DOWN
      INTERSECT
      TOPICAL ID T-10 DOWN
    )
  )
```

Listing 4.21: Differenz in MonQL

□

Die neuen Möglichkeiten von MonQL erlauben es auch Anfragen zu stellen, die thematische Eigenschaften ausschließen (siehe Beispiel 4.36).

Beispiel 4.36 [Kombination zweier thematischer Anfragen]

Die Anfrage

*„Suche nach Dokumenten, die direkt mit dem Westbau mit Türmen oder mit Gebäudeteilen aus denen der Westbau besteht, verknüpft sind und **nicht** das Material Naturstein **oder** die Restaurierungsphase der Herren Schmitz und Schultz aufweisen!“*

lässt sich mit MonQL, wie in Listing 4.22 gezeigt, umsetzen. Die Annahme, dass „Westbau mit Türmen“ die ID **SE-3**, „Material Naturstein“ die ID **T-4** und „Restaurierungsphase der Herren Schmitz und Schultz“ die ID **T-10** hat, muss auch hier wieder gültig sein.

```
SELECT DISTINCT *
FROM
  (
    SPATIAL ID SE-3 DOWN
  DIFFERENCE
    (
      TOPICAL ID T-4 DOWN
    UNION
      TOPICAL ID T-10 DOWN
    )
  )
```

Listing 4.22: Differenz in MonQL

□

4.7. Übersetzung in SPARQL

Das Datenmodell lässt sich als relationale Datenbank oder als OWL-2-DL realisieren, womit sich neben der Übersetzung in SQL eine Übersetzung in SPARQL anbietet.

Abbildung 4.2 zeigt den schematischen Ablauf, der hier für die Konvertierung der Anfrage in die jeweilige Anfragesprache gewählt wurde.

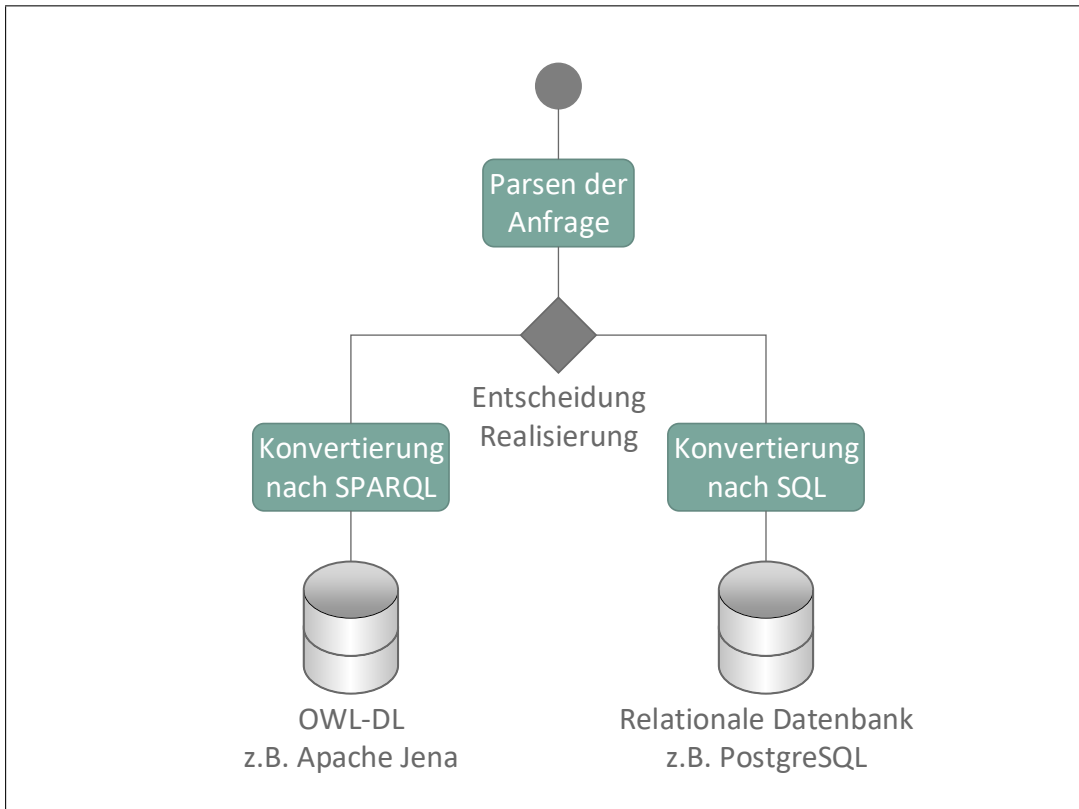


Abbildung 4.2.: Konvertierung nach SQL oder SPARQL

Die SQL-Übersetzung ergibt sich direkt aus den SQL-Unterabfragen und Fragmenten, die bei der Definition von MonQL eingeführt wurden. Die Übersetzung in SPARQL lässt sich durch eine Konvertierung der zugehörigen SQL-Anfragen und Konstrukte erreichen. MonQL verwendet bei der Konvertierung in SQL nur begrenzte Sprachkonstrukte, die sich entsprechen in SPARQL wiederfinden. Die informelle Semantik eine SPARQL-Anfrage ergibt sich daher aus der Übersetzung von MonQL und auf die Zurückführung auf die dazugehörige SQL-Anfrage.

Bei der Konvertierung von MonQL in SPARQL muss die Auswertungsart der SPARQL-Anfrage miteinbezogen werden, da je nach gewähltem Entailment Regime [GO13] das Ergebnis einer Anfrage unterschiedlich ausfallen kann. Was dies genau bedeuten kann, soll hier an einem kurzen Beispiel erläutert werden.

Beispiel 4.37 [Anfrageauswertung unter verschiedenen Entailment Regimes]

Das hier vorgestellte Beispiel orientiert sich an dem Beispiel aus Punkt 1.2 des SPARQL 1.1 Entailment Regimes der W3C Recommendation vom 21. März 2013 [GO13].

Betrachtet wird erneut das Beispiel von St. Lorenz (siehe Abbildung 4.3).

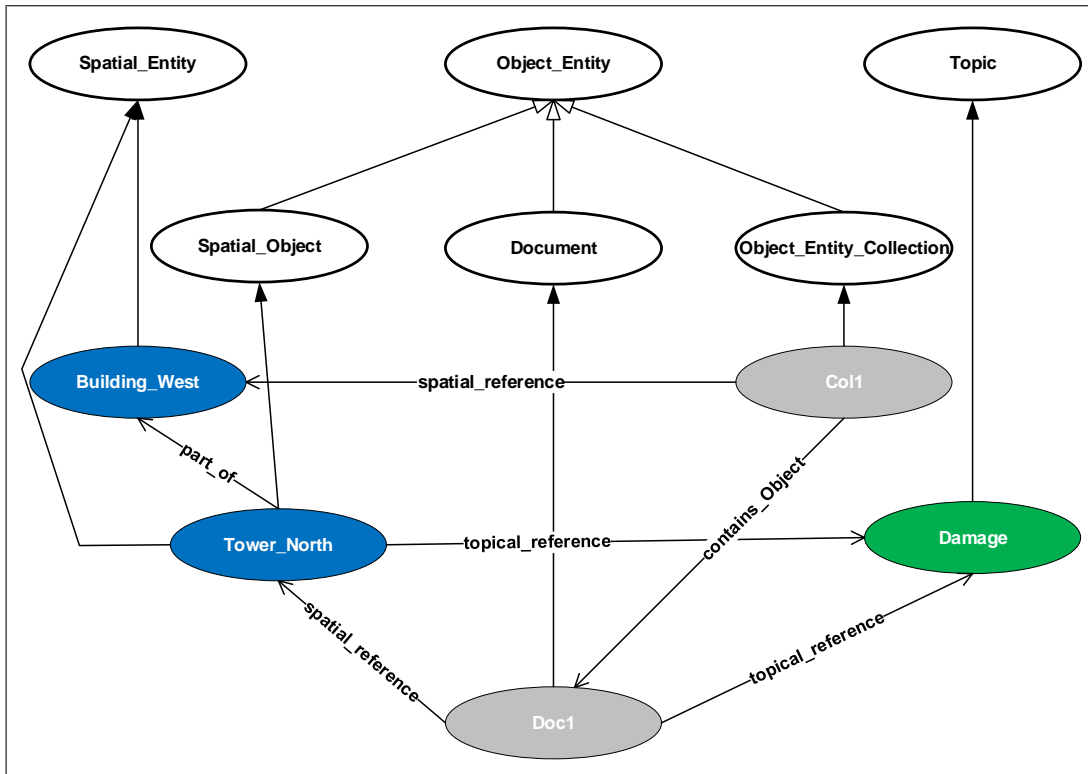


Abbildung 4.3.: Konzepte, Individuen und Rollenaussagen der $O_{\text{instance}}^{\text{object}}$ (StLorenz)

Dieses lässt sich in OWL-2-DL als RDF verkürzt wie folgt darstellen:

```

:Doc1 rdf:type :Document .
:Col1 rdf:type :Object_Entity_Collection .
:Tower_North rdf:type :Spatial_Object .
:Col1 :spatial_reference :Building_West .
:Doc1 :spatial_reference :Tower_North .
:Doc1 :topical_reference :Damage .
:Tower_North :topical_reference :Damage .

```

Betrachtet man die Anfrage:

„Suche nach allen *Object_Entity*“

```

SELECT ?entity WHERE { ?entity rdf:type :Object_Entity }

```

Unter einem „simple entailment regime“ ist das Ergebnis leer. Unter einem „OWL 2 DL RDF-Based entailment regime“ [GO13] hingegen besteht das Ergebnis aus dem intuitiv

erwarteten *Doc1*, *Coll* und *Tower_North*. Dies lässt sich in diesem Beispiel aus der Beziehung zwischen *Object_Entity* und *Document* ableiten, da *Document* eine Unterklasse bzw. ein Unterkonzept von *Object_Entity* ist. □

Im Folgenden wird davon ausgegangen, dass die Anfragen unter dem „OWL 2 DL RDF-Based entailment regime“ beantwortet werden. Dies erlaubt es unter anderem, dass transitive Properties für die SPARQL-Anfrage verwendet werden können, da eine Inferenz Axiome die durch die Transitivität von z.B. *part_of* entstehen, der Ontologie hinzufügt. Zu welchem Zeitpunkt die Inferenz stattfindet, also zur Anfragezeit oder in Form einer Materialisierung in den jeweiligen Ontologien, kann dabei gewählt werden.

Analog zu SQL betrachten wir die Grundstruktur von MonQL und die Übersetzung in SPARQL.

Abbildungsdefinition 4.38 [Übersetzung MonQL in SPARQL - Grundstruktur]

Die Grundstruktur einer Anfrage in MonQL:

```
SELECT (DISTINCT)? (*|<attribute>(, <attribute>)*)  
FROM  
(<spatial>|<topical>|<type>)  
( (UNION|INTERSECT|DIFFERENCE)  
<spatial>|<topical>|<type>)*  
(WHERE <where>)?  
(ORDER BY <order>)?  
(LIMIT <Integer>)?
```

wird in SPARQL übersetzt:

```
SELECT (DISTINCT)? (*|( ?<attribute>)* )  
WHERE  
(OPTIONAL { ?object_entity :<attribute> ?<attribute> .}) *  
(<spatial>|<topical>|<type>)  
( (UNION| |MINUS)  
<spatial>|<topical>|<type>)*  
(FILTER <filter>)?  
(ORDER BY <order>)?  
(LIMIT <Integer>)?
```

Bei der Übersetzung werden die Attribute in der Projektion in MonQL auf Variablen (?<attribute>) in SPARQL abgebildet. Die Kombinationsmöglichkeiten UNION, INTERSECT und DIFFERENCE in MonQL werden auf UNION, eine „graph pattern“-Sequenz und MINUS in SPARQL abgebildet.

Die Platzhalter <spatial>, <topical>, <type>, <where> und <order> können analog zu SQL unabhängig voneinander umgesetzt werden. □

Die Übersetzung der WHERE-Klausel von MonQL kann in SPARQL mit einem FILTER-Ausdruck realisiert werden.

Abbildungsdefinition 4.39 [Übersetzung MonQL in SPARQL - WHERE-Klausel]
Die WHERE-Klausel einer Anfrage in MonQL:

```
WHERE (<attribute> (<|<=<|=|=>|>) <value>
|NOT <where>
|<where> (AND|OR) <where>)
```

wird in SPARQL übersetzt:

```
FILTER (?<attribute> (<|<=<|=|=>|>) <value>
|NOT <filter>
|<filter> (&&||) <filter>)
```

□

Die Sortierung und Ergebnisbegrenzung kann in SPARQL mit ORDER BY und LIMIT eins zu eins wie in SQL realisiert werden.

Abbildungsdefinition 4.40 [Übersetzung MonQL in SPARQL - ORDERBY-Klausel]
Die ORDERBY-Klausel einer Anfrage in MonQL:

```
ORDER BY <attribute> ( |ASC|DESC) (, <attribute> ( |ASC|DESC) ) *
```

wird in SPARQL übersetzt:

```
ORDER BY (?<attribute>|ASC(?<attribute>)|DESC(?<attribute>)) *
```

□

Die räumlichen und thematischen Anfragen können durch group graph patterns ohne blank nodes umgesetzt werden.

Abbildungsdefinition 4.41 [Übersetzung MonQL in SPARQL - räumliche Anfrage]
Eine räumliche Anfrage in MonQL:

```
SPATIAL (ID (<Integer>|<ID>|<UUID>)
|KEYWORD <Keyword>) UP? DOWN?
```

wird in SPARQL übersetzt:

```
{ ?spatial_entity (:id|:caption).
{ ?object_entity :spatial_reference ?spatial_entity .}
(UNION /*UP-Variante (optional)*/
{ ?object_entity :spatial_reference ?spatial_entity_up .
?spatial_entity_up :transitive_has_part ?spatial_entity .})?
(UNION /*DOWN-Variante (optional)*/
{ ?object_entity :spatial_reference ?spatial_entity_down .
?spatial_entity_down :transitive_part_of ?spatial_entity .})?
}
```

4. Anfragemodell

Dabei können die Selektion per ID und KEYWORD des `Spatial_Entity` durch die Prädikate `id`, `caption` umgesetzt werden.

Die grafische Selektion kann mit Prädikaten aus der SPARQL-Erweiterung GeoSPARQL [Mat12] übersetzt werden und erlauben somit geometrische Abfragen in SPARQL. Die Möglichkeit den Suchraum entlang der *part_of*- oder *consists_of*-Hierarchie zu erweitern, ist durch den UP-Variante und DOWN-Variante realisiert. Hier ist nur der Fall für die *part_of*-Hierarchie gezeigt. Der Fall für die *consists_of*-Hierarchie funktioniert analog. \square

Betrachtet man die Anfrage aus Beispiel 4.18 erneut, lässt sich diese wie im Folgenden Beispiel 4.42 umsetzen.

Beispiel 4.42 [Räumliche Anfrage SPARQL]

Ausgehend von der Anfrage in Listing 4.23 ergibt sich nach der Transformation die SPARQL-Anfrage (siehe Listing 4.24):

```
SELECT DISTINCT caption, author
FROM
  SPATIAL ID SE-3 DOWN
```

Listing 4.23: Räumliche Anfrage per ID

```
SELECT DISTINCT ?caption ?author
WHERE { { {
  OPTIONAL { ?object_entity :caption ?caption . }
  OPTIONAL { ?object_entity :author ?author . } }
  { ?spatial_entity :id SE-3 .
    { ?object_entity :spatial_reference ?spatial_entity . }
  UNION
  { ?object_entity :spatial_reference ?spatial_entity_down .
    ?spatial_entity_down :transitive_part_of ?spatial_entity . }
} } }
```

Listing 4.24: Räumliche SPARQL-Anfrage per ID

In diesem Beispiel ist bereits der Vorteil von MonQL in Bezug auf die Länge und Lesbarkeit gegenüber SPARQL erkennbar. \square

Thematische Anfragen lassen sich analog zu den räumlichen Anfragen in SPARQL übersetzen.

Abbildungsdefinition 4.43 [Übersetzung MonQL in SPARQL - thematische Anfrage]

Eine thematische Anfrage in MonQL:

```
TOPICAL (ID (<Integer>|<ID>|<UUID>)
|KEYWORD <Keyword>) UP? DOWN)?*)
```

wird in SPARQL übersetzt:

```
{ ?topic (:id|:caption).
  { ?object_entity :topical_reference ?topic . }
(UNION /*UP-Variante (optional)*/
  { ?object_entity :topical_reference ?topic_up .
    ?topic_up :sup_topic_transitive ?spatial_entity .})?
```



```
(UNION /*DOWN-Variante (optional)*/
 { ?object_entity :topical_reference ?topic_down .
   ?topic_down :super_topic_transitive ?topic .})?
}
```

□

Betrachtet man erneut die Anfrage aus Beispiel 4.31, lässt sich diese wie im Folgenden Beispiel 4.44 umsetzen:

Beispiel 4.44 [Thematische Anfrage]

Ausgehend von der Anfrage in Listing 4.25 ergibt sich nach der Transformation die SPARQL-Anfrage (siehe Listing 4.26).

```
SELECT DISTINCT caption, author
FROM
  TOPICAL KEYWORD "Naturstein" DOWN
```

Listing 4.25: Thematische Anfrage per KEYWORD

```
SELECT DISTINCT ?caption ?author
WHERE { { {
  OPTIONAL { ?object_entity :caption ?caption . }
  OPTIONAL { ?object_entity :author ?author . } }
 { ?topic :caption "Naturstein" .
   { ?object_entity :topical_reference ?topic . }
  UNION
   { ?object_entity :topical_reference ?sub_topic .
     ?sub_topic :super_topic_transitive ?topic . }
 } } }
```

Listing 4.26: Thematische SPARQL-Anfrage per KEYWORD

□

Abschließend betrachten wir in Beispiel 4.45 eine Kombination von räumlicher und thematischer Anfragen.

Beispiel 4.45 [Räumliche und thematische Anfrage kombiniert]

Ausgehend von der Anfrage in Listing 4.27 ergibt sich nach der Transformation die SPARQL-Anfrage (siehe Listing 4.28).

```
SELECT DISTINCT *
FROM
  (
    ( SPATIAL ID SE-4 DOWN
      UNION
      SPATIAL ID SE-6 DOWN )
    INTERSECT
    ( TOPICAL ID T-4 DOWN
      INTERSECT
      TOPICAL ID T-10 DOWN )
  )
```

Listing 4.27: Räumliche und thematische Anfragekombination per ID

```
SELECT DISTINCT *
WHERE
{
  { ?spatial_entity1 :id SE-4 .
    { ?object_entity :spatial_reference ?spatial_entity1 .}
  UNION
  { ?object_entity :spatial_reference ?spatial_entity2 .
    ?spatial_entity2 :transitive_part_of ?spatial_entity1 .} }
UNION
{ ?spatial_entity3 :id SE-6 .
  { ?object_entity :spatial_reference ?spatial_entity3 .}
UNION
{ ?object_entity :spatial_reference ?spatial_entity4 .
  ?spatial_entity4 :transitive_part_of ?spatial_entity3 .} }
{ { ?topic1 :id T-4 .
  { ?object_entity :topical_reference ?topic1 .}
  UNION
  { ?object_entity :topical_reference ?topic2 .
    ?topic2 :super_topic_transitive ?topic1 .} }
  { ?topic3 :id T-10 .
    { ?object_entity :topical_reference ?topic3 .}
  UNION
  { ?object_entity :topical_reference ?topic4 .
    ?topic4 :super_topic_transitive ?topic3 .}}
}
}
```

Listing 4.28: Räumliche und thematische SPARQL-Anfragekombination per ID

Die Lesbarkeit der SPARQL-Anfrage ist bei Verwendung von *INTERSECT* stark erschwert, da hier die Gruppen einfach aneinandergereiht werden. □

Auf die Übersetzung der TYP-Anfragen aus MonQL gehen wir im Folgenden Kapitel genauer ein.

4.8. Fazit

In diesem Kapitel wurde passend zum erweiterten Datenmodell die Anfragesprache MonQL vorgestellt, die eine kompakte Formulierung von räumlichen und thematischen Anfragen erlaubt.

Wir haben MonQL eingeführt, um einerseits von der konkreten Realisierung des Datenmodells zu abstrahieren und andererseits, was noch viel wichtiger ist, um MonQL als Abstraktionsschicht für räumliche und thematische Anfragen in diesem Kontext fungieren zu lassen. MonQL erlaubt es eine Anfragetransformation (siehe Kapitel 5) auf dieser Abstraktionsebene durchzuführen und dann in SQL und SPARQL zu konvertieren.

Zusätzlich konnten die MonArch-Anfragen ein Stück weit verallgemeinert werden. Dies erlaubt neue Anfragearten, die bisher im MonArch-Kontext so nicht möglich waren.

5. Anfragerelaxation und -transformation

In den beiden vorherigen Kapiteln wurden ein erweitertes Datenmodell und eine darauf basierende Anfragesprache vorgestellt, die es erlauben, die Fragestellung hinter dem MonArch-Anwendungsfall kompakt und präzise zu formulieren und dafür Lösungen zu erarbeiten.

Dieses Kapitel formalisiert in Abschnitt 5.1 die in Definition 2.16 skizzierte Verbundanfrage und die daraus resultierenden Anforderungen auf Basis des erweiterten Datenmodells (siehe Kapitel 3) und mithilfe der Anfragesprache MonQL (siehe Kapitel 4) neu. Aus der Fragestellung lassen sich Ziele ableiten, die in Abschnitt 5.2 beschrieben sind.

Abschnitt 5.3 stellt existierende Lösungsansätze vor und bewertet diese auf ihre Anwendbarkeit hinsichtlich der Fragestellung.

Die Anfragetransformation wird in Abschnitt 5.4 vorgestellt. In Abschnitt 5.5 und Abschnitt 5.6 wird dann explizit auf die räumliche und thematische Anfragerelaxation eingegangen.

Abgeschlossen wird dieses Kapitel durch eine Betrachtung der Eigenschaften und Grenzen der Anfragetransformation (siehe Abschnitt 5.8).

Beginnen wir mit der Begriffsdefinition für die in dieser Arbeit betrachtete Anfragerelaxation (siehe Definition 5.1).

Definition 5.1 [Anfragerelaxation]

Seien zwei Anfragen in MonQL gegeben:

Q: Die Anfrage, die relaxiert wird.

Q': Die relaxierte Anfrage.

Zusätzlich sei der Suchraum der Anfragen definiert als:

R: Der Suchraum der Anfrage Q.

R': Der Suchraum der Anfrage Q'.

Eine *Anfragerelaxation* (Φ_r):

$$Q \Phi_r Q'$$

ist eine Veränderung der Anfrage Q hin zur Anfrage Q', die folgende Eigenschaft erfüllt:

$$R \subseteq R'$$

Eine relaxierte Anfrage Q' kann durch folgende Veränderungen der Anfrage Q erreicht werden:

- Entfernen von Selektionsbedingungen,
- Verändern von Selektionsbedingungen,
- Hinzufügen von Selektionsbedingungen,
- Veränderung des Anfragetyps.

□

Die in dieser Arbeit definierte Anfragerelaxation ist vergleichbar mit der in [ERW11; MT14] vorgestellten. Der dort vorgestellte Ansatz bewirkt zwar ebenfalls die Erweiterung des Suchraums, die Autoren beziehen sich dabei aber auf ein anderes Szenario.

Die im Folgenden vorgestellte Anfragerelaxation entspricht in ihrer Wirkung teilweise einer Query Expansion in der klassischen Schlüsselwortsuche, sie beschränkt sich also nicht nur auf das Entfernen von Selektionsbedingungen.

Beispiel 5.2 zeigt für den MonArch-Anwendungsfall eine Anfragerelaxation auf Basis von MonQL.

Beispiel 5.2 [Anfragerelaxation]

Ausgehend von einer Anfrage Q :

```
SELECT DISTINCT *  
FROM SPATIAL KEYWORD „Nordturm“
```

betrachten wir die relaxierte Anfrage Q' :

```
SELECT DISTINCT *  
FROM  
    SPATIAL KEYWORD „Nordturm“  
UNION  
    SPATIAL KEYWORD „Südturm“
```

Die Relaxation der Anfrage bewirkt durch disjunktives Hinzufügen einer weiteren räumlichen Anfrage nach dem **Südturm** eine Erweiterung des Suchraums. Da sowohl der **Nordturm** als auch der **Südturm** betrachtet werden, gilt:

$$R \subseteq R'$$

□

Die Anfragerelaxation wird durch eine Anfragetransformation (siehe Definition 5.3) realisiert.

Definition 5.3 [Anfragetransformation]

Als **Anfragetransformation** (Φ_t):

$$Q \Phi_t Q'$$

wird die Veränderung einer Anfrage Q in eine Anfrage Q' definiert.

Dabei unterliegt die Veränderung keiner Einschränkung. Dies bedeutet, dass eine Anfragerelaxation Φ_r immer eine Anfragetransformation Φ_t ist, eine Anfragetransformation jedoch nicht zwangsweise eine Anfragerelaxation sein muss.

Die Richtung

$$\Phi_r \Rightarrow \Phi_t$$

ergibt sich aus der Definition der Anfragerelaxation.

Dass die Richtung

$$\Phi_t \Rightarrow \Phi_r$$

nicht immer gilt, zeigt Beispiel 5.4. □

Beispiel 5.4 [Anfragetransformation - Gegenbeispiel]

Ausgehend von einer Anfrage Q :

```
SELECT DISTINCT *
FROM
  SPATIAL KEYWORD „Nordturm“
UNION
  SPATIAL KEYWORD „Südturm“
```

betrachten wir die transformierte Anfrage Q' :

```
SELECT DISTINCT *
FROM
  SPATIAL KEYWORD „Nordturm“
INTERSECT
  SPATIAL KEYWORD „Südturm“
```

Wird die Vereinigung durch einen Schnitt ersetzt, gilt nicht:

$$R \subseteq R'$$

Durch den Schnitt wird der Suchraum R' eingeschränkt. Daher kann eine Veränderung der Operatoren dazu führen, dass eine Transformation keine Relaxation mehr ist. □

Wird im Folgenden von räumlicher oder thematischer Ontologie gesprochen, sind damit immer die Ontologien, die in Abschnitt 3.4 eingeführt wurden, gemeint. Analog verhält es sich, wenn von räumlichen und thematischen Anfragen die Rede ist. Hierbei beziehen wir uns auf Kapitel 4.

Die Algorithmen und Transformationsverfahren, die in diesem Kapitel eingeführt werden, beschränken sich auf die Identifizierung von Elementen im Struktur- und Themengraphen per ID.

Obwohl MonQL auch eine Identifizierung per KEYWORD oder eine grafische Selektion vorsieht, lassen sich diese alternativen Selektionsmechanismen auf die Selektion per ID zurückführen.

Dies erlaubt es auf eine Fallunterscheidung bei den Algorithmen und Transformationsverfahren zu verzichten und verbessert somit die Lesbarkeit. Analog dazu werden die Elemente der Struktur- und Themengraphen auch in Beispielen mit einer Pseudo-ID versehen. Die Pseudo-ID dient zur besseren Lesbarkeit der Grafiken und ersetzt eine echte ID. Für eine durchgängige und einheitliche Benennung wird folgendes ID-Schema eingeführt.

Definition 5.5 [ID-Schema für Struktur- und Themengraphen]

Für Elemente im Strukturgraphen wird folgendes ID-Schema eingeführt:

- `Spatial_Entity_Type`: Strukturtypen werden mit dem Präfix `SET-` gefolgt von einem Integerwert identifiziert.
- `Spatial_Entity`: Gebäudeteile werden mit dem Präfix `SE-` gefolgt von einem Integerwert identifiziert

Für Elemente im Themengraphen wird folgendes ID-Schema eingeführt:

- `Topic`: Themen werden mit dem Präfix `T-` gefolgt von einem Integerwert identifiziert.
- `Topic_Group`: Themengruppen werden mit dem Präfix `TG-` gefolgt von einem Integerwert identifiziert.

Falls mehrere Struktur- bzw. Themengraphen in Beispielen vorkommen, werden die IDs der Elemente mit weiteren zusätzlichen Präfixen versehen.

Betrachten wir den `Nordturm` von St. Lorenz und den `Nordturm` von St. Sebald:

- Der `Nordturm` von St. Lorenz hat die ID: `Lo:SE-5`.
- Der `Nordturm` von St. Sebald hat die ID: `Se:SE-5`.

Um die Lesbarkeit der Anfragen zu erhöhen und einen besseren Abgleich mit den dazugehörigen Graphen zu ermöglichen, wird in manchen Beispielen nicht die ID, sondern der Name, als Pseudo-ID verwendet.

- Der `Nordturm` von St. Lorenz hat die ID: `Lo:Nordturm`.
- Der `Nordturm` von St. Sebald hat die ID: `Se:Nordturm`.
- Das Thema `Georg`, das nur in St. Lorenz vorkommt hat die ID: `Lo:Georg`.
- Das Thema `Heiliger`, das in St. Lorenz und St. Sebald vorkommt, hat die ID: `MA:Heiliger`.

□

5.1. Fragestellung

Die Verbundanfrage, die in Kapitel 2 vorgestellt wurde, ermöglicht es, Anfragen nicht nur an einzelne Archive zu stellen, sondern an einen Verbund von MonArch-Archiven.

Die Herausforderung dabei ist es, die Unterschiede im Struktur- und Themengraphen der einzelnen Archive im Verbund zu berücksichtigen. Es stellt sich folgende Frage:

Wie kann eine Anfrage, die sich auf einen Struktur- bzw. Themengraphen bezieht, in MonQL so relaxiert werden, dass die veränderte Anfrage auf weitere Struktur- bzw. Themengraphen im Verbund anwendbar ist und dabei die Intention der Ursprungsanfrage erhalten bleibt?

Zur Lösung dieser Frage ist es notwendig, einige Definitionen einzuführen, um die Begrifflichkeiten der Fragestellung genauer zu fassen.

Da eine MonQL-Anfrage als Kombination aus räumlichen und thematischen Unteranfragen aufgebaut ist, muss geklärt werden, wie die Intention sowohl einer räumlichen Anfrage als auch einer thematischen Anfrage erhalten werden kann.

Hierzu führen wir zuerst den Begriff der elementaren räumlichen bzw. thematischen Unteranfrage (siehe Definition 5.6) ein.

Definition 5.6 [Elementare räumliche bzw. thematische Unteranfrage]

Als *elementare räumliche bzw. thematische Unteranfrage* wird ein Anfragefragment einer räumlichen bzw. thematischen Anfrage in MonQL bezeichnet.

Als elementare räumliche Unteranfrage s bezeichnen wir ein Anfragefragment der Form:

```
(TYPE)? SPATIAL ID <id|path> UP? DOWN?
```

Als elementare thematische Unteranfrage t bezeichnen wir ein Anfragefragment der Form:

```
TOPICAL ID <id|path> UP? DOWN?
```

- S bezeichnet die Menge der elementaren räumlichen Unteranfragen.
- T bezeichnet die Menge der elementaren thematischen Unteranfragen.

Als *räumliche bzw. thematische Unteranfrage* werden Kombinationen von elementaren räumlichen bzw. thematischen Unteranfragen bezeichnet. □

Die Intention einer Anfrage in MonQL wird durch die von ihr spezifizierte räumliche und thematische Selektion beschrieben. Die räumliche Intention ergibt sich aus der Selektion im Strukturgraphen und die thematische Intention ergibt sich aus der Selektion im Themengraphen.

Ziel ist es, die Selektion im Strukturgraphen intentionserhaltend auf einen anderen Strukturgraphen zu übertragen. Wird in der Ausgangsanfrage z.B. ein Pfeiler selektiert, sollten auch

wieder bevorzugt Pfeiler in der relaxierten Anfrage selektiert werden. Darüber hinaus ist der räumliche Kontext, in dem sich der selektierte Pfeiler befindet, wichtig für die Intention der Anfrage. Wird ein Pfeiler im Außenbau selektiert, sollten in der relaxierten Anfrage bevorzugt Pfeiler im Außenbau selektiert werden.

Ähnlich verhält es sich für den Themengraphen. Ziel ist es hier, die Selektion im Themengraphen intentionserhaltend auf einen anderen Themengraphen zu übertragen. Dabei spielt die Oberbegriffs- und Unterbegriffsbeziehung eine wichtige Rolle. Wird ein Thema selektiert, das im Zielthemengraphen nicht vorhanden ist, muss ein Thema im Zielthemengraphen gefunden werden, das der Intention des ursprünglichen Themas weitgehend entspricht. Z.B. kann ein Oberbegriff gewählt werden, der in beiden Themengraphen vorhanden ist.

Die Definition einer intentionserhaltenden Anfrage lässt sich auf intentionserhaltende räumliche Unteranfragen (siehe Definition 5.7) und intentionserhaltende thematische Unteranfragen (siehe Definition 5.8) zurückführen.

Definition 5.7 [Intentionserhaltende räumliche Unteranfrage]

Die räumliche Unteranfrage s' ist *intentionserhaltend* zur elementaren räumlichen Anfrage s genau dann, wenn einer der folgenden drei Fälle zutrifft:

1. Fall: Die selektierten Gebäudeteile SE und SE' der Anfragen s und s' sind identisch.
2. Fall: Die selektierten Gebäudeteile (SE) der elementaren Anfrage s und die selektierten Gebäudeteile (SE') der Anfrage s' haben den identischen Strukturtypen (*Spatial_Entity_Type*).
3. Fall: Die Strukturtypen der selektierten Gebäudeteile (SE') der Anfrage s' sind identisch mit den Strukturtypen
 - a) der selektierten Gebäudeteile (SE) der Anfrage s ,
 - b) der Gebäudeteile im Strukturgraph, die durch die *part_of*- bzw. *consists_of*-Beziehungen, startend bei den selektierten Gebäudeteilen (SE), erreichbar sind.

$$s : \approx s' \Leftrightarrow$$

$$SE \equiv SE' \vee, \quad \text{1. Fall}$$

$$\exists \text{type}^-.SE \equiv \exists \text{type}^-.SE' \vee, \quad \text{2. Fall}$$

$$\exists \text{type}^-.SE' \sqsubseteq \exists \text{type}^-.(\exists \text{has_part_transitive}.SE \sqcup \exists \text{consists_of}^+.SE \sqcup SE), \quad \text{3. Fall}$$

□

Definition 5.8 [Intentionserhaltende thematische Unteranfrage]

Die thematische Unteranfrage t' ist *intentionserhaltend* zur elementaren thematischen Anfrage t genau dann, wenn einer der folgenden drei Fälle zutrifft:

1. Fall: Die selektierten Themen T und T' der Anfragen t und t' sind identisch.

2. Fall: Die selektierten Themen (\mathbb{T}) der Anfrage t sind Unterthemen der selektierten Themen (\mathbb{T}') der Anfrage t' .
3. Fall: Die selektierten Themen (\mathbb{T}) der Anfrage t stehen in einer *related_topic*- oder *sameAS*-Beziehung zu den selektierten Themen (\mathbb{T}') der Anfrage t' .

$$t : \approx t' \Leftrightarrow$$

$$\mathbb{T} \equiv \mathbb{T}' \vee, \quad \text{1. Fall}$$

$$\mathbb{T} \sqsubseteq \exists \textit{super_topic_transitive}.\mathbb{T}' \vee, \quad \text{2. Fall}$$

$$\mathbb{T} \sqsubseteq \exists \textit{related_topic}.\mathbb{T}' \sqcup \exists \textit{sameAS}.\mathbb{T}', \quad \text{3. Fall}$$

□

Die Relaxation der Anfrage im Verbund kann durch eine Anfragetransformation realisiert werden.

Dazu ist es notwendig, eine intentionserhaltende Anfragetransformation (siehe Definition 5.11) zu definieren. Bevor diese definiert werden kann, muss zusätzlich der Begriff einer strukturerhaltenden Anfragetransformation (siehe Definition 5.9) definiert werden.

Definition 5.9 [Strukturerhaltende Anfragetransformation]

Sei Q die Ursprungsanfrage, die aus elementaren Unteranfragen Q_1, \dots, Q_{n+1} besteht, die mit Operatoren $\Theta_1, \dots, \Theta_n$ verknüpft sind. Eine Anfragetransformation Φ ist *strukturerhaltend* wenn die Zielanfrage Q' aus elementaren Unteranfragen Q_1', \dots, Q_{n+1}' besteht, die mit den gleichen Operatoren $\Theta_1, \dots, \Theta_n$ verknüpft werden und für die gilt:

Q_i' ist vom gleichen Anfragetyp wie Q_i (räumlich bzw. thematisch) oder eine UNION von elementaren Unteranfragen gleichen Anfragetyps, $1 \leq i \leq n + 1$.

□

Beispiel 5.10 zeigt für eine MonQL-Anfrage eine strukturerhaltende Anfragetransformation, die in diesem Fall auch eine Anfragerelaxation darstellt.

Beispiel 5.10 [Strukturerhaltende Anfragetransformation]

Betrachten wir die Ursprungsanfrage Q :

```
SELECT DISTINCT *
FROM
  (
    SPATIAL KEYWORD „Nordturm“ DOWN
  INTERSECT
    (
      TOPICAL KEYWORD „Naturstein“ DOWN
    INTERSECT
      TOPICAL KEYWORD
      „Restaurierungsphase von Schmitz und Schultz“ DOWN
    )
  )
```

Die transformierte Anfrage Q' :

```

SELECT DISTINCT *
FROM
(
  (
    SPATIAL KEYWORD „Nordturm“ DOWN
    UNION
    SPATIAL KEYWORD „Südturm“ DOWN
  )
  INTERSECT
  (
    (
      TOPICAL KEYWORD „Naturstein“ DOWN
      UNION
      TOPICAL KEYWORD „natural stone“ DOWN
    )
    INTERSECT
    TOPICAL KEYWORD
    „Restaurierungsphase von Schmitz und Schultz“ DOWN
  )
)

```

ist strukturerhaltend, da die Verknüpfung der räumlichen und thematischen Anfragen mit den Operatoren UNION und INTERSECT zur Ursprungsanfrage bis auf eine UNION gleichen Anfragetyps identisch ist.

Bei dieser Anfragetransformation wurde die räumliche Selektion auf den **Südturm** und die thematische Selektion auf den englischen Begriff von **Naturstein** erweitert. Diese Transformation stellt also auch eine Relaxation dar, da die ursprüngliche Anfrage erhalten bleibt und durch die Erweiterung mit UNION defokussiert wird. \square

Die intentionserhaltende Anfragetransformation lässt sich wie folgt definieren:

Definition 5.11 [Intentionserhaltende Anfragetransformation]

Eine Anfragetransformation ist *intentionserhaltend*, wenn für jede elementare räumliche und jede elementare thematische Unteranfrage der Ursprungsanfrage Q eine auf die elementare Unteranfrage beschränkte intentionserhaltende Transformation in eine Unteranfrage in der Zielanfrage Q' existiert. Zusätzlich muss die gesamte Anfragetransformation strukturerhaltend sein.

Φ ist intentionserhaltend \Leftrightarrow

$$(\forall s (s \in S \rightarrow \exists \Phi_{spatial} \exists s' (s' \in S' \wedge s \Phi_{spatial} s' \wedge s \approx s'))) \wedge$$

$$(\forall t (t \in T \rightarrow \exists \Phi_{topical} \exists t' (t' \in T' \wedge t \Phi_{topical} t' \wedge t \approx t'))) \wedge$$

Φ ist strukturerhaltend (siehe Definition 5.9)

- S ist die Menge der elementaren räumlichen Anfragen der Ursprungsanfrage Q .
- T ist die Menge der elementaren thematischen Anfragen der Ursprungsanfrage Q .

- S' ist die Menge der elementaren räumlichen Anfragen der Zielanfrage Q' .
- T' ist die Menge der elementaren thematischen Anfragen der Zielanfrage Q' .
- $\Phi_{spatial}$ ist eine Transformation einer elementaren räumlichen Unteranfrage.
- $\Phi_{topical}$ ist eine Transformation einer elementaren thematischen Unteranfrage.
- $s \approx s'$ ist eine intentionserhaltende elementaren räumliche Unteranfrage (siehe Definition 5.7).
- $t \approx t'$ ist eine intentionserhaltende elementaren thematische Unteranfrage (siehe Definition 5.8).

□

Mithilfe der vorbereitenden Definitionen lässt sich definieren, wann eine Anfrage intentionserhaltend zu einer Ursprungsanfrage ist.

Definition 5.12 [Intentionserhaltende Anfrage]

Eine Anfrage Q' ist zu einer Anfrage Q *intentionserhaltend* genau dann, wenn es eine intentionserhaltende Anfragetransformation Φ (siehe Definition 5.11) von Q nach Q' gibt. □

Die Fragestellung lässt sich mithilfe der eingeführten Definitionen neu formulieren.

Gesucht ist eine intentionserhaltende Anfragetransformation, die eine Verbundanfrage so relaxiert, dass die unterschiedlichen Struktur- und Themengraphen im Verbund berücksichtigt werden.

5.2. Zieleigenschaften

Aufgabe dieses Abschnittes ist es, Zieleigenschaften für eine intentionserhaltende Anfragetransformation und -relaxation zu konkretisieren.

Aus dem in Abschnitt 2.5 vorgestellten Szenario der Verbundanfrage an einen MonArch-Verbund lassen sich folgende Ziele und Eigenschaften einer Anfragetransformation ableiten:

Intentionserhaltung der Anfrage

Eine Eigenschaft, die ein Anfragetransformationsverfahren erfüllen muss, ist die Erhaltung der Intention der Anfrage. Ziel ist es, eine MonArch-Anfrage intentionserhaltend zu transformieren.

Nehmen wir an, dass erkannt werden kann, dass dem Pfeiler mit der ID [Lo:SE-5](#) im Archiv St. Lorenz der Pfeiler mit der ID [Se:SE-5](#) im Archiv St. Sebald entspricht. Dann sollte das Anfragetransformationsverfahren die Anfrage Q transformieren in die Anfrage Q' :

- Q :

```
SELECT *
FROM
  SPATIAL ID Lo:SE-5 /* Pfeiler */
INTERSECT
TOPICAL ID T-25 /* Sandstein */
```

- Q' :

```
SELECT *
FROM
  SPATIAL ID Se:SE-5 /* Pfeiler */
INTERSECT
TOPICAL ID T-25 /* Sandstein */
```

Die Intention bleibt erhalten, da hier die räumliche Unteranfrage transformiert wurde und beide selektierten Gebäudeteile denselben Strukturtypen aufweisen (Definition 5.7, 2. Fall).

Relaxation der Anfrage

Ein für den MonArch-Anwendungsfall passendes Anfragetransformationsverfahren muss in der Lage sein, eine Relaxation der Anfrage zuzulassen.

Nehmen wir an, dass eine Anfrage an einen Pfeiler mit der ID *Lo:SE-5* im Archiv St. Lorenz auf alle Pfeiler mit den IDs *Lo:SE-5* bis *Lo:SE-10* relaxiert werden soll, um den Suchraum auf alle Pfeiler zu erweitern. Dann sollte das Anfragetransformationsverfahren die Anfrage Q in Q' transformieren.

- Q:

```
SELECT *
FROM
  SPATIAL ID Lo:SE-5 /* Pfeiler 1 */
```

- Q' :

```
SELECT *
FROM
  ((
    SPATIAL ID Lo:SE-5 /* Pfeiler 1 */
  UNION
    SPATIAL ID Lo:SE-6 /* Pfeiler 2 */ )
  UNION
    SPATIAL ID ... )
  UNION
    SPATIAL ID Lo:SE-10 /* Pfeiler 5 */ )
```

Dies bedeutet im Allgemeinen, dass die Anfragerelaxation ausgehend von einem selektierten Gebäudeteil, mehrere Gebäudeteile desselben Typs in die relaxierte Anfrage mitaufnehmen kann.

Einbeziehung der Strukturhierarchie

Für den MonArch-Anwendungsfall spielt die Strukturhierarchie der Gebäudeteile eine zentrale Rolle. Daher muss bei einer Anfragetransformation die Abbildung der Gebäudeteile im Zusammenhang mit der Strukturhierarchie betrachtet werden.

Dies bedeutet, dass ein Gebäudeteil nicht nur, weil es denselben Namen hat, auch die gleiche Funktion innerhalb eines Gebäudes einnimmt.

Nehmen wir wieder einen Pfeiler im Archiv St. Lorenz. Es macht einen Unterschied, ob der Pfeiler sich im Innenbau oder im Außenbau befindet. Eine Transformation muss also Pfeiler im Außenbau bevorzugt Pfeilern im Außenbau zuordnen. Dies gilt analog für den Innenbau.

Einbeziehung der Themenhierarchie

Für den MonArch-Anwendungsfall spielt die Themenhierarchie zur weiteren Verfeinerung einer Anfrage eine wichtige Rolle. Daher muss bei einer Anfragetransformation die Abbildung der Themen in Zusammenhang mit der Themenhierarchie betrachtet werden.

Dies bedeutet, dass ein Thema immer im Kontext seiner Oberthemen betrachtet werden muss.

Nehmen wir das Thema *Riss*, dann kann hier ein Riss im Mauerwerk gemeint sein oder ein Riss in einem Fotopapier. Der erste Riss wäre ein spezieller *Gebäudeschaden*, der zweite Riss wäre ein spezieller *Dokumentschaden*.

Unabhängigkeit

In einem MonArch-Verbund steht jedes MonArch-Archiv organisatorisch meist für sich alleine.

Ein Anfragetransformationsverfahren muss daher die Unabhängigkeit der einzelnen MonArch-Archive bei der Anfragetransformation wahren.

Im Idealfall bedeutet dies, dass das Anfragetransformationsverfahren eine Anfrage transformieren kann, ohne dabei auf die Struktur- und Themengraphen der einzelnen MonArch-Archive des Verbundes zugreifen zu müssen.

Für die Beispielanfrage nach dem Pfeiler in St. Lorenz sollte das Anfragetransformationsverfahren nicht auf den Strukturgraphen von St. Sebald angewiesen sein, um eine Transformation auf dem MonArch-Archiv von St. Lorenz durchzuführen.

Benennungsschemata und Mehrsprachigkeit

In den MonArch-Archiven St. Lorenz und St. Sebald sind durchaus Gebäudeteile unterschiedlich benannt, obwohl sie die gleiche Funktion einnehmen. Daher muss ein Anfragetransformationsverfahren in der Lage sein, vom Benennungsschemata zu abstrahieren.

Ein ähnliches Phänomen entsteht auch, falls eine Anfragetransformation über Sprachgrenzen hinweg erfolgen soll und somit Gebäudeteile unterschiedliche Namen aufweisen.

Z.B. könnten die Pfeiler von St. Lorenz im Uhrzeigersinn benannt worden sein, die Pfeiler von St. Sebald im Gegenuhrzeigersinn. Auch könnte, statt *Seitenschiff Nord*, die Benennung *nave north* oder für den *Außenbau* die Benennung *Außen* verwendet werden.

Allgemein bedeutet dies, dass sich das Anfragetransformationsverfahren nicht auf eine sprachabhängige Benennung von Gebäudeteilen (*Spatial_Entity*) zum Erstellen einer Abbildung verlassen kann.

Automatisierung

Im MonArch-Anwendungsfall setzt ein manueller Eingriff in die Anfragetransformation voraus, dass der Anfragende sowohl Kenntnis der Ontologie der Quellenfrage als auch der Ontologie der Zielanfrage besitzen muss.

Daher muss es Ziel des Anfragetransformationsverfahrens sein, die Transformation automatisiert durchzuführen.

Diese sieben Ziele dienen zur Bewertung von verschiedenen Ansätzen zur Realisierung eines Anfragetransformations- und -relaxationsverfahrens.

5.3. Anwendbarkeit existierender Verfahren

Da ein ontologiebasiertes Datenmodell vorliegt (siehe Abschnitt 3.4), scheint ein naheliegender Ansatz die Abbildung der räumlichen und thematischen Ontologien des Quellsystems auf diejenigen des Zielsystems zu sein.

Viele Forschungsarbeiten befassen sich unter dem Begriff *Ontology-Matching* (siehe Definition 5.13) mit diesem Problem.

Definition 5.13 [Ontology-Matching]

Eine Definition von Paaren von Elementen zweier Ontologien wird als *Ontology-Matching* bezeichnet [SE05; SE13; Tay10]. Dabei wird eine Ontologie O als Quellontologie und die andere als Zielontologie O' bezeichnet.

Eine Abbildung (*Match*) kann als Tupel (E_1, R, E_2) beschrieben werden, wobei $E_1 \in O$ und $E_2 \in O'$ ist. Der Ausdruck R beschreibt dabei die Beziehung zwischen E_1 und E_2 . Für R ist in den meisten Verfahren die Gleichheit (\equiv), die Verallgemeinerung (\sqsubseteq) oder die Disjunktheit ($E_1 \sqcap E_2 \sqsubseteq \perp$) erlaubt. \square

In diesem Zusammenhang werden oft die Begriffe *Ontology-Alignment* (siehe Definition 5.14) und *Ontology-Merging* (siehe Definition 5.15) verwendet.

Definition 5.14 [Ontology-Alignment (siehe Abbildung 5.1)]

Als Ontology-Alignment wird das Ergebnis und teilweise auch der Prozess eines Ontology-Matching-Verfahrens bezeichnet.

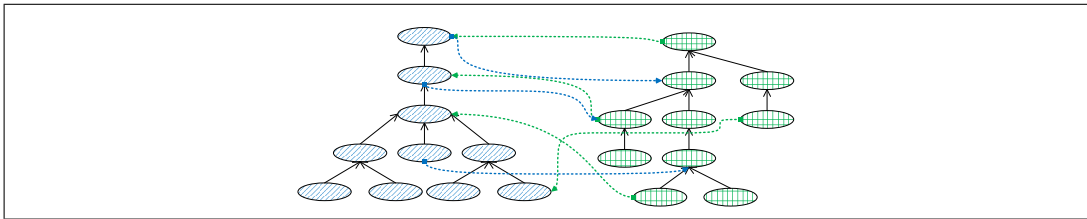


Abbildung 5.1.: Ontology-Alignment

Wieder ausgehend von zwei Ontologien O und O' berechnet ein Verfahren in einem Matching-Prozess ein Alignment A . Das Alignment A kann 1:1, 1:N oder N:M Abbildungen zwischen den Ontologien O und O' beinhalten [SE05; SE13; KS03]. \square

Definition 5.15 [Ontology-Merging (siehe Abbildung 5.2)]

Als Ontology-Merging wird das Zusammenführen von mehreren unterschiedlichen Ontologien derselben Domäne zu einer kohärenten Ontologie bezeichnet [Noy04].

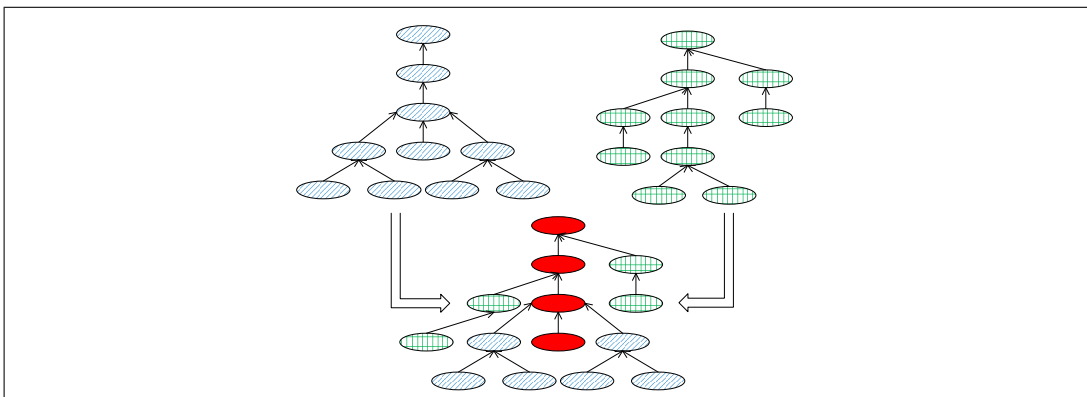


Abbildung 5.2.: Ontology-Merging

Die ursprünglichen Ontologien werden dabei durch die neu modellierte Ontologie ersetzt. Dem Ontology-Merging geht im Allgemeinen ein Ontology-Alignment voraus, welches die auftretenden Korrespondenzen ermittelt. Die identifizierten Entsprechungen werden jedoch nicht zur Abbildung zwischen den Elementen der Ontologien genutzt, sondern zu einem gemeinsamen Symbol zusammengefasst, das die bisherigen ersetzt [Tay10]. \square

In [ES13] wird eine Klassifikation von Ontology-Matching-Verfahren vorgestellt, die für diese Arbeit als Grundlage zur Bewertung und Anwendbarkeit auf die Fragestellung dient.

5.3.1. Klassen von Verfahren des Ontology-Matchings

Die Autoren von [ES13] bauen ihre Klassifikation (siehe Abbildung 5.3) dabei auf [RB01] auf und liefern eine grundlegende Aufteilung in elementbasierte und strukturbasierte Verfahren.

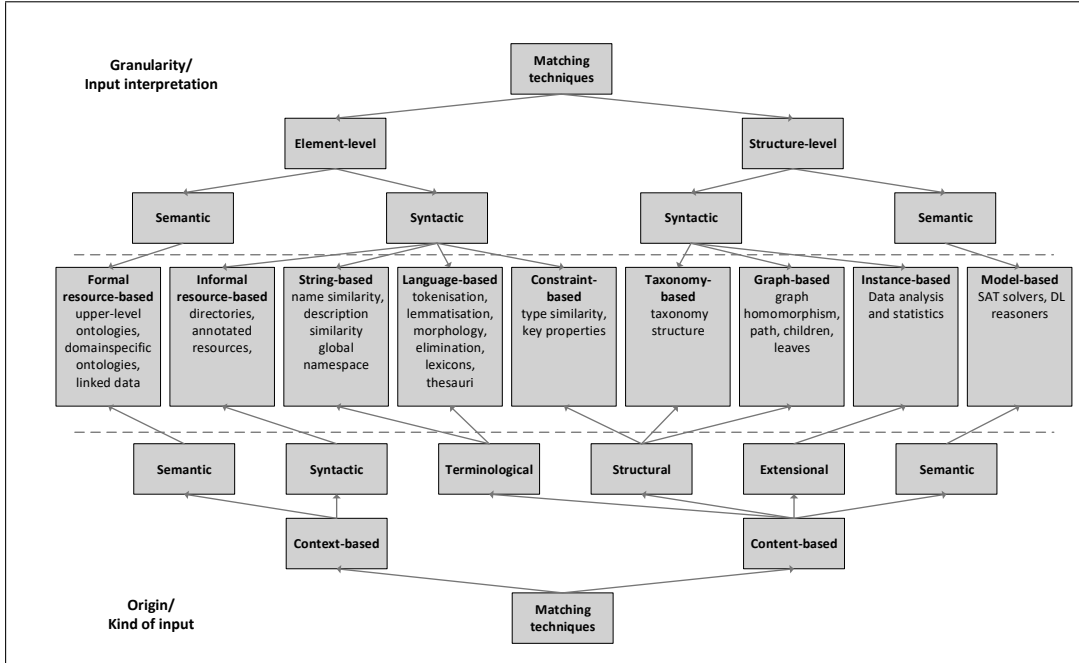


Abbildung 5.3.: Ontology-Matching-Verfahren (angelehnt und inhaltlich entnommen aus [ES13])

Im Folgenden wird auf die Klassifikation der Verfahren eingegangen und anhand der definierten Ziele eine Bewertung vorgenommen.

Die Eigenschaften der Verfahren sind dabei aus [Ngu16] entnommen. In der Arbeit von [Ngu16] wurde ebenfalls auf die Klassifikation von [ES13] zurückgegriffen.

Element-level

Diese Klasse von Verfahren betrachtet die in den beiden Ontologien spezifizierten Beziehungen sowohl zwischen Konzepten als auch Individuen einer Ontologie nicht.

Daher kann das Ziel der Einbeziehung der Struktur- bzw. Themenhierarchie bei der Zuordnung von Gebäudeteilen mit diesen Verfahren nicht erreicht werden.

Diese Verfahrensklasse wird deshalb hier nicht weiter betrachtet.

Structural-level

Diese Klasse von Verfahren betrachtet auch Beziehungen einer Ontologie, um Ähnlichkeiten zwischen Konzepten, Rollen und Individuen zu berechnen.

Damit wären sie prinzipiell in der Lage, die Struktur- und Themenhierarchie miteinzubeziehen. Folgende vier Unterklassen gibt es in dieser Verfahrensklasse.

Graph-based

Diese Klasse von Verfahren betrachtet die Ontologien als kanten- und knotenmarkierte Graphen, indem die Knoten Konzepte und Individuen und die Kanten Rollen darstellen.

Taxonomy-based

Diese Verfahrensklasse kann nur auf Taxonomien angewendet werden und beschränkt sich auf die „is-a“-Beziehung in Ontologien.

Für räumliche Ontologien sind diese Verfahren nicht geeignet, für thematische Ontologien können diese Verfahren angewendet werden.

Instance-based

Diese Klasse nutzt Individuen, um eine Ähnlichkeit von Konzepten zu bestimmen.

Model-based

Die Verfahren dieser Klasse nutzen ähnliche Semantikinterpretationen der Konzepte, um Ähnlichkeiten zwischen Konzepten zu bestimmen.

Dreh- und Angelpunkt bei allen Structural-level Verfahren ist die Berechnung eines Ontology-Alignments (siehe Definition 5.14) zwischen den räumlichen und thematischen Ontologien, und hier insbesondere zwischen den Individuen der Ontologien. Die Berechnung kann nur dann erfolgen, wenn alle Ontologien vorhanden sind.

Für den MonArch-Anwendungsfall müssen also räumliche und thematische Ontologien der einzelnen MonArch-Archive ausgetauscht werden. Daher kann das Ziel der Unabhängigkeit der MonArch-Archive mit diesen Verfahren nicht erfüllt werden.

Nehmen wir an, dass sowohl die räumlichen als auch thematischen Ontologien der einzelnen MonArch-Archive ausgetauscht werden können, dann könnte eine Kombination der Structural-level Verfahren Erfolg versprechend sein [NM02; ES13].

Daher betrachten wir im folgenden Abschnitt kombinierte strukturelle Verfahren.

5.3.2. Kombinierte Structural-level Verfahren

Der nächste Schritt zur Überprüfung existierender Ontology-Matching-Verfahren auf ihre Anwendbarkeit als Teil einer Anfragetransformation, ist die Betrachtung von kombinierten Verfahren.

Kombinierte Verfahren können ausgeschlossen werden:

- Falls sie einen manuellen Eingriff in die Erstellung eines Alignments vorsehen.

Die Benutzergruppe im MonArch-Anwendungsfall sind Experten für Gebäude, deren Struktur und den dazugehörigen Informationen. Sie sind jedoch keine Ontology-Mapping Experten, welche das Alignment definieren könnten. Zusätzlich wäre der Aufwand bei N MonArch-Archiven nicht vertretbar, da $O(N^2)$ Ontology-Alignments erstellt werden müssten. Dieser Aufwand müsste unter Umständen bei jeder Änderung an einem Archiv durchgeführt werden.

- Falls sie nur ein 1:1-Alignment zulassen.

Dies führt dazu, dass die Anfrage nicht relaxiert werden kann, was dem formulierten Ziel der Relaxation der Anfrage widerspricht

Die Arbeiten von [KS03; CSH06; Tay10; SE13] bieten dazu verschiedene Übersichten und Beschreibungen der Eigenschaften von kombinierten Verfahren.

PROMPT

Das PROMPT-Framework [NM00] ist ein Framework, welches ein semi-automatisches Verfahren anwendet, um Korrespondenzen zu ermitteln [KS03; CSH06].

Für den MonArch-Anwendungsfall ist es nicht geeignet, da es nicht vollautomatisch ist.

Anchor-PROMPT

In [NM01] wird ein Nachfolger Anchor-PROMPT von PROMPT vorgestellt. Dieser erweitert den Ansatz um Anchor-Paare, die miteinander verwandte Terme in Beziehung setzen [CSH06].

Dieser Schritt kann durch eine lexikalische Analyse automatisiert werden [NM01]. Daher ist das Ausschlusskriterium des manuellen Eingriffs behoben, dafür stößt dieses Verfahren schnell an seine Grenzen bei unterschiedlichen Benennungsschemata und Mehrsprachigkeit.

MAFRA

Das Mapping-Framework for Distributed Ontologies in the Semantic Web (MAFRA) [Mae+02; MS00] ist ein weiteres semi-automatisches Verfahren zur Berechnung eines Alignments zwischen zwei Ontologien.

Über sogenannte Semantic-Bridging können Korrespondenzen zwischen Konzepten und Individuen hergestellt werden.

Da dieses Framework eine Benutzerinteraktion vorsieht, ist es für den MonArch-Anwendungsfall ebenfalls nicht geeignet.

Chimaera

Chimaera [Mcg+00] ist ein weiteres interaktives Verfahren, das es erlaubt, auf Basis von linguistischen Heuristiken potentielle Korrespondenzen zwischen den Ontologien

zu identifizieren und nur bei Bedarf den Benutzer in den Prozess involviert [KS03; CSH06].

Da der automatische Teil des Verfahrens nur elementbasiert vergleicht, ist dieses Verfahren für den MonArch-Anwendungsfall nicht einsetzbar, da die Struktur- und Themenhierarchie nicht miteinbezogen werden kann und zusätzlich ein manueller Eingriff notwendig ist.

FCA-Merge

Das Verfahren FCA-Merge [SM01] nutzt eine Menge von Dokumenten, die in allen beteiligten Ontologien klassifiziert sein müssen.

Die Ergebnisse des Verfahrens hängen stark von der Menge an Dokumenten ab, die vorhanden sind. Dies ist daher der kritische Faktor dieses Verfahrens, was auch explizit angemerkt wird [KS03; SM01].

Da für den MonArch-Anwendungsfall in den meisten Fällen die Dokumentenmengen der einzelnen MonArch-Archive disjunkt sind, kann dieses Verfahren nicht angewendet werden. Darüber hinaus werden nur Konzepte und Rollen im Verfahren berücksichtigt, bei Auftreten von Konflikten ist zudem ein Domänenexperte notwendig [KS03; SM01].

OntoMerge

OntoMerge [DMQ03] ist ein in mehreren Schritten durchgeführtes Verfahren, das die beteiligten Ontologien zu einer gemeinsamen Ontologie vereinheitlicht (siehe Definition 5.15).

Die Autoren von [DMQ03] weisen explizit darauf hin, dass beim Identifizieren von Korrespondenzen, die durch sogenannte Bridging-Axiome ausgedrückt werden, Domänenexperten und Knowledge-Engineers erforderlich sind. Daher ist auch dieses Verfahren für den MonArch-Anwendungsfall nicht anwendbar, weil ein manueller Eingriff notwendig ist.

SAMBO

SAMBO [LT06] ist ein System, das in der Lage ist, für Ontologien aus dem biomedizinischen Bereich ein Ontology-Matching durchzuführen. Es berechnet ein 1:1-Alignment für Konzepte und Relationen, aber nicht für Individuen [SE13]. Daher ist es für den MonArch-Anwendungsfall nicht einsetzbar.

Falcon

Falcon [HQC08] ist ein automatischer divide-and-conquer Ansatz der speziell für große Ontologien entwickelt wurde.

Es berechnet ein 1:1-Alignment indem ein linguistisches Verfahren mit einem iterativen strukturellen Verfahren kombiniert wird [SE13]. Durch die Einschränkung auf die Berechnung eines 1:1-Alignments ist es für den MonArch-Anwendungsfall nicht einsetzbar.

DSSim

DSSim [NV10; NVS09] ist ein agentenbasiertes Framework für das Ontology-Matching. Ausgehend von OWL oder SKOS werden 1:1-Alignments berechnet. Es kombiniert verschiedene Verfahren zur Berechnung eines Vergleichsmaßes [SE13].

Durch die explizite Unterstützung für SKOS-Ontologien könnte es im MonArch-Anwendungsfall für die thematischen Ontologien eingesetzt werden. Für räumliche Ontologien ist es durch die Beschränkung auf 1:1-Alignments nicht geeignet.

RiMOM

RiMOM [Li+09] ist ein kombiniertes Verfahren aus einem elementbasierten und einem strukturellen Verfahren. Es erzeugt ein 1:1-Alignment auf Konzepte und Rollen, mit der Besonderheit, dass auch Konzepte auf Rollen abgebildet werden können [SE13].

Durch die Beschränkung auf 1:1-Alignments ist es nicht geeignet.

ASMOV

Automatic Semantic Matching of Ontologies with Verification (ASMOV) [JSK09] ist ein weiteres automatisches Verfahren aus dem Bereich der Biomedizin. Es erzeugt N:M-Alignments für Konzepte und Rollen in zwei Schritten, in denen die Ähnlichkeit berechnet wird und auf die Berechnung eine semantische Verifikation folgt [SE13].

Da die Berechnung der Ähnlichkeiten auf lexikalischen Verfahren beruht, führt dies bei unterschiedlichen Benennungsschemata nicht zum Ziel und kann daher für den MonArch-Anwendungsfall nicht verwendet werden.

Anchor-Flood

Der Anchor-Flood-Ansatz [SA09] ist für große Ontologien konzipiert und erzeugt ein 1:1-Alignment. Dieses Verfahren konzentriert sich auf die Abbildung von Konzepten und Rollen [SE13].

Durch die Beschränkung auf 1:1-Alignments ist es nicht geeignet.

AgreementMaker

AgreementMaker [CAS09; Far+13] ist ein System das eine Vielzahl an automatischen und semi-automatischen Verfahren zur Berechnung eines N:M-Alignments verwendet.

AgreementMaker beschränkt sich beim Ontology-Matching auf Konzepte und Rollen [Far+13]. Da ein manueller Eingriff notwendig ist, ist dieses Verfahren nicht für den MonArch-Anwendungsfall einsetzbar.

Darüber hinaus gibt es noch eine Vielzahl an weiteren Verfahren wie z.B. CIDER [GM08], SPIDER [MA04], GeRoMeSuite [Ken+07], MLMA+ [AHS09], Lily [WX08], MapPSO [BH08], TaxoMap [Ham+08] und ASCO [LDG04], die nicht geeignet sind [Ngu16], da:

- sie nicht vollautomatisch sind,
- sie sich auf ein 1:1-Alignment beschränken,
- sie als Hauptmechanismus ein lexikalisches Verfahren zugrunde legen,

- sie auf einen anderen Anwendungsfall, z.B. Biomedizin, ausgelegt sind.

5.3.3. Zusammenfassung

Die Analyse existierender Verfahren des Ontology-Matching zeigt, dass die betrachteten Verfahren voraussetzen, dass ein vollständiger Zugriff auf die räumlichen und thematischen Ontologien gewährleistet ist. Da dies im MonArch-Verbund nicht gewährleistet ist, können die Verfahren nicht eingesetzt werden.

Nimmt man an, dass ein Zugriff auf räumliche und thematische Ontologien möglich ist, sprechen jedoch weitere Gründe gegen bestehende Verfahren:

- 1:1-Alignments verhindern eine Relaxation der Anfrage.
- Nicht automatisierbare Verfahren machen es unmöglich, diese in der Praxis einzusetzen, da der Aufwand für die Abbildung nicht erbracht werden kann.
- Elementbasierte Verfahren sind für unterschiedliche Benennungsschemata und Mehrsprachigkeit in diesem Kontext nicht geeignet.

5.4. Anfragetransformation

Die folgenden Abschnitte stellen ein Anfragetransformationsverfahren vor, das alle definierten Ziele aus Abschnitt 5.2 erfüllen wird. Dabei wird auf eine Berechnung eines Ontology-Alignments verzichtet, wodurch die Nachteile der existierenden Verfahren überwunden werden können.

Das Anfragetransformationsverfahren basiert dabei auf zwei Eigenschaften, die sich aus dem MonArch-Anwendungsfall ergeben:

Eigenschaft des Strukturgraphen

Jedes Gebäude, sei es ein Sakralbau oder ein einfaches Einfamilienhaus, ist aus wiederkehrenden Gebäudeteilen oder funktionalen Gebäudesegmenten aufgebaut. In einem Sakralbau gibt es z.B. mehrere Pfeiler und in einem Einfamilienhaus z.B. mehrere Wände. In der Regel können also die verschiedenen Gebäudeteile typisiert und auf Typebene zusammengefasst werden.

Eine weitere Eigenschaft, die essentiell ist für eine Alternative zur Berechnung eines Ontology-Alignments, ist die Beobachtung, dass sich die Typen innerhalb eines Gebäudetyps nicht unterscheiden.

Alle Einfamilienhäuser teilen sich Grundtypen wie *Wand*, *Raum* und *Stockwerk*. Sogar spezielle Räume wie *Küche*, *Schlafzimmer* oder *Wohnzimmer* sind in den meisten Einfamilienhäusern zu finden. Genauso verhält es sich bei Sakralbauten, die

sich auch speziellere Gebäudeteiltypen teilen, wie z.B. *Pfeiler*, *Schiff*, *Turm* oder *Fassadenabschnitt*.

Eigenschaft des Themengraphen

Betrachtet man die Themenhierarchien der MonArch-Archive St. Lorenz und St. Sebald, so teilen sich beide Archive eine gemeinsame Oberthemenhierarchie. Beide Gebäude verwenden dieselben Themen für z.B. Materialien und Schäden, unterscheiden sich aber in speziellen Themen, wie z.B. ausführende Baumeister.

Der Einsatz von kontrollierten Vokabularen wie z.B. die GND führt auch dazu, dass unterschiedliche Archive gemeinsame Themenhierarchien zur thematischen Auszeichnung verwenden.

Ziel ist es, die Eigenschaften des Strukturgraphen und Themengraphen, die sich aus der Anwendungsdomäne ergeben, zu nutzen, um damit die Berechnung des Ontology-Alignment durch eine Transformation der Anfragen ersetzen zu können.

In Definition 5.16 werden Annahmen getroffen, die im Folgenden für die Anfragetransformation angenommen werden. Es sei darauf hingewiesen, dass die beschriebenen Annahmen im MonArch-Anwendungsfall alle erfüllt sind.

Definition 5.16 [Annahmen Anfragetransformation]

Für die Anfragetransformation werden folgende Annahmen getroffen:

- Im Strukturgraphen hat jedes Gebäudeteil genau einen direkten Strukturtypen.
- Alle Verbundmitglieder teilen sich die gleiche Typhierarchie der Strukturtypen.
- Alle Verbundmitglieder teilen sich eine gemeinsame Themenhierarchie, die aber in den Unterthemen verfeinert werden kann.
- Werden Themen aus kontrollierten Vokabularen eingesetzt, existieren *super_topic*-, *sameAS*- oder *related_topic*-Beziehungen zu Themen aus einer gemeinsamen Themenhierarchie.

□

Die in dieser Arbeit vorgestellte intentionserhaltende Anfragetransformation wird in zwei große Phasen aufgeteilt. Einmal in eine Transformation im Quellsystem und einmal in eine Transformation im Zielsystem.

Ausgangspunkt der Transformation ist eine Anfrage Q_{source} , die an das Quellsystem gestellt wird. Darauf folgt eine Transformation der Anfrage im Quellsystem.

Als Ergebnis der Anfragetransformation im Quellsystem entsteht eine Anfrage $Q_{\text{intermediate}}$. Die Anfrage $Q_{\text{intermediate}}$ wird an ein Zielsystem übermittelt. Dabei kann das Zielsystem z.B. ein MonArch-Archiv im MonArch-Verbund sein. Die Anfrage $Q_{\text{intermediate}}$ kann auch parallel an mehrere Zielsysteme geschickt werden.

Zusätzlich kann das Quellsystem auch als Zielsystem fungieren. Damit lässt sich die Anfragetransformation auch zur Relaxation einer Anfrage im Quellsystem einsetzen.

Ergebnis der Anfragetransformation im Zielsystem ist eine Anfrage Q_{target} .

Die Aufteilung in zwei Phasen stellt sicher, dass die Transformation unabhängig, also nur mit den Informationen des Quell- bzw. Zielsystems, durchgeführt wird. Es müssen bei dieser Transformation außer der Anfrage selbst keine weiteren Informationen zwischen Quell- und Zielsystem ausgetauscht werden.

5.4.1. Anfragetransformation im Quellsystem

Im Quellsystem wird die Anfrage für das Zielsystem aufbereitet. Algorithmus 5.1 zeigt den Ablauf im Quellsystem.

```

Input :  $Q_{\text{source}}$ 
/* Setze Verbundanfrage gleich der Ursprungsanfrage */
1  $Q_{\text{intermediate}} := Q_{\text{source}}$ ;
2 Let S be the set of spatial sub queries s of  $Q_{\text{intermediate}}$ ;
3 Let T be the set of topical sub queries t of  $Q_{\text{intermediate}}$ ;
/* Transformation der elementaren räumlichen Unteranfragen */
4 foreach s  $\in$  S do
5 | s := transformSpatialQuerySource(s);
/* Transformation der elementaren thematischen Unteranfragen */
6 foreach t  $\in$  T do
7 | t := transformTopicalQuerySource(t);
Result :  $Q_{\text{intermediate}}$ 

```

Algorithmus 5.1 : Anfragetransformation im Quellsystem

Eingabe des Algorithmus ist die Ursprungsanfrage Q_{source} und Ausgabe ist die Verbundanfrage $Q_{\text{intermediate}}$.

Der erste Schritt des Transformationsalgorithmus im Quellsystem setzt die Verbundanfrage gleich der Ursprungsanfrage (Zeile 1). Durch diese Gleichsetzung der Anfragen wird sichergestellt, dass die Unteranfragen der Verbundanfrage mit den gleichen Operatoren verknüpft sind wie die elementaren Unteranfragen der Ursprungsanfrage.

Im zweiten Schritt werden alle elementaren räumlichen (Zeile 4-5) und thematischen Unteranfragen (Zeile 6-7) der Anfrage $Q_{\text{intermediate}}$ transformiert. Das Ergebnis der Transformation einer elementaren Unteranfrage ersetzt die Unteranfrage in $Q_{\text{intermediate}}$. Da jede elementare Unteranfrage für sich transformiert wird, könnte die Transformation der elementaren Unteranfragen prinzipiell parallelisiert werden. Im Algorithmus ist die Transformation sequenziell dargestellt.

Durch die Transformation jeder einzelnen elementaren Unteranfrage im zweiten Schritt mit anschließendem Ersetzen in der Verbundanfrage ist sichergestellt, dass die ersetzte Unteranfrage den gleichen Anfragetyp hat.

Die Anfragetransformation der räumlichen und thematischen Unteranfragen ist in Abschnitt 5.6 und Abschnitt 5.5 genauer beschrieben.

5.4.2. Anfragetransformation im Zielsystem

Im Zielsystem wird die Verbundanfrage für den Struktur- und Themengraphen des Zielsystems aufbereitet. Algorithmus 5.2 zeigt den Ablauf im Zielsystem.

```
Input :  $Q_{\text{intermediate}}$ 
/* Setze Zielfrage gleich der Verbundanfrage */
1  $Q_{\text{target}} := Q_{\text{intermediate}};$ 
2 Let  $S_{\text{target}}$  be the set of spatial sub queries  $s$  of  $Q_{\text{target}};$ 
3 Let  $T_{\text{target}}$  be the set of topical sub queries  $t$  of  $Q_{\text{target}};$ 
/* Transformation der elementaren räumlichen Unteranfragen */
4 foreach  $s \in S_{\text{target}}$  do
5 |  $s := \text{transformSpatialQueryTarget}(s);$ 
/* Transformation der elementaren thematischen Unteranfragen */
6 foreach  $t \in T_{\text{target}}$  do
7 |  $t := \text{transformTopicalQueryTarget}(t);$ 
Result :  $Q_{\text{target}}$ 
```

Algorithmus 5.2 : Anfragetransformation im Zielsystem

Eingabe des Algorithmus ist die Verbundanfrage $Q_{\text{intermediate}}$ und Ausgabe ist die Zielfrage Q_{target} . Auch im Zielsystem läuft die Anfragetransformation in zwei Schritten ab.

Der erste Schritt des Transformationsalgorithmus im Quellsystem setzt die Verbundanfrage gleich der Ursprungsanfrage (Zeile 1). Die Gleichsetzung der Anfragen stellt auch hier wieder sicher, dass die Unteranfragen der Zielfrage mit den gleichen Operatoren verknüpft sind wie die elementaren Unteranfragen der Verbundanfrage.

Im zweiten Schritt werden, wie im Quellsystem, alle elementaren räumlichen (Zeile 4-5) und thematischen Unteranfragen (Zeile 6-7) der Anfrage Q_{target} transformiert. Das Ergebnis der Transformation einer elementaren Unteranfrage ersetzt die Unteranfrage in Q_{target} . Der zweite Schritt lässt sich wieder parallelisieren.

Die Transformation stellt auch hier sicher, dass die ersetzten Unteranfragen den gleichen Anfragetyp haben.

Die Anfragetransformation der räumlichen und thematischen Unteranfragen im Zielsystem ist in Abschnitt 5.6 und Abschnitt 5.5 genauer beschrieben.

5.4.3. Betrachtung Intentionserhaltung

Anhand der bis hierher beschriebenen Anfragetransformation im Quell- und Zielsystem, lassen sich einige der in Definition 5.11 eingeführten Eigenschaften überprüfen. Hierzu werden

die unabhängigen Anfragetransformationen im Quell- und Zielsystem als eine Anfragetransformation betrachtet.

Die Definition der intentionserhaltenden Anfragetransformation sieht vor, dass sowohl die elementaren räumlichen als auch die elementaren thematischen Unteranfragen unabhängig transformiert werden.

Überprüfung 5.17 [Unabhängigkeit der Transformation von Unteranfragen]

Sowohl der Transformationsalgorithmus im Quell- als auch im Zielsystem sehen eine unabhängige Transformation der elementaren räumlichen und elementaren thematischen Unteranfragen vor. Der Algorithmus ist so konzipiert, dass jede elementare Unteranfrage für sich transformiert wird.

Werden Quell- als auch Zieltransformation hintereinandergeschaltet, entstehen keine zusätzlichen Abhängigkeiten, da Unteranfragen durch Unteranfragen ersetzt werden, die wiederum unabhängig voneinander transformiert werden können. \square

Als weitere Eigenschaft sieht die Definition die Strukturhaltung vor.

Überprüfung 5.18 [Strukturhaltung der Transformation von Anfragen]

Betrachten wir die Anfragetransformation im Quellsystem, stellt der Algorithmus die Strukturhaltung dadurch sicher, dass die Ursprungsanfrage, die aus den Unteranfragen Q_1, \dots, Q_{n+1} besteht, welche mit Operatoren $\Theta_1, \dots, \Theta_n$ verknüpft sind, nicht geändert, sondern einfach kopiert wird. Dasselbe Argument lässt sich auch für die Anfragetransformation im Zielsystem anwenden.

Die jeweiligen Transformationen der Unteranfragen müssen jetzt noch sicherstellen, dass die transformierte Unteranfrage vom gleichen Anfragetyp wie die Unteranfrage in der Quellanfrage (räumlich bzw. thematisch) oder eine UNION von Unteranfragen gleichen Anfragetyps ist. \square

Im Folgenden muss überprüft werden, dass die Anfragetransformation der räumlichen und thematischen Unteranfragen im Zusammenspiel zwischen Transformation im Quellsystem und Transformation im Zielsystem intentionserhaltend ist, damit der gesamte Transformationsprozess intentionserhaltend ist.

5.4.4. Spreading-Activation

Bevor die räumliche und thematische Anfragetransformation beschrieben wird, führen wir in diesem Abschnitt einen auf die Transformation angepassten Spreading-Activation-Algorithmus ein, der für die folgenden Anfragetransformationen notwendig ist. In [RSA04; SSS08; SLR14] werden der Grundalgorithmus und verschiedene Anpassungen vorgestellt, auf denen diese Arbeit aufbaut.

Das Spreading-Activation-Verfahren wurde erstmals in [CL75] von Collins und Loftus beschrieben mit dem Ziel, die Prozesse im Gehirn bei der Auswahl eines Begriffes abzubilden.

In [RSA04] zeigen die Autoren, wie das Spreading-Activation-Verfahren mit Suchtechniken kombiniert und der Spreading-Activation-Algorithmus auf Ontologien angewendet werden können.

Die Autoren von [Dix+10] zeigen, wie das Spreading-Activation-Verfahren auf große Ontologien angewendet werden kann. Die vorgeschlagene Abbruchbedingung über die Anzahl der Aktivierungsschritte wurde in dem hier vorgestellten Spreading-Activation-Algorithmus übernommen.

Schumacher und Kollegen [SSS08] zeigen wie eine Suche nach Dokumenten mit Hilfe von Spreading-Activation auf Ontologien realisiert werden kann.

Die Autoren von [SLR14] zeigen die Unterschiede zwischen PageRank [Pag+99] und dem Spreading-Activation-Verfahren und bilden die Grundlage für die Entscheidung für das Spreading-Activation-Verfahren, die in der vorliegenden Arbeit getroffen wurde. Hauptargument für die Auswahl von Spreading-Activation ist die Flexibilität bei der Behandlung von unterschiedlichen Kanten [SLR14, 4.1 Path constraint].

Einen ähnlichen Ansatz wie das Spreading-Activation-Verfahren verfolgt das Similarity Flooding von Melnik und Kollegen [MGR02]. Similarity Flooding wird wegen des Initialisierungsschrittes, der auf textueller Ähnlichkeit basiert, hier nicht gewählt (vgl. auch Abschnitt 5.3).

Dass das Spreading-Activation-Verfahren und die darin berechneten Aktivierungswerte für ein Ranking von Ergebnissen angewendet werden kann, zeigen die Autoren von [Noo+16] und die Spreading-Activation-Anpassung Discovery Hub von [Mar+14].

[RSA04] bildet die Grundlage für das in diesem Kapitel vorgestellte Verfahren. Die Raum- und Typontologie (Unterabschnitt 3.4.1) als auch die Themaontologie (Unterabschnitt 3.4.2) bilden dabei die Grundlage für den Spreading-Activation-Graphen (siehe Definition 5.19).

Die Anpassung des Spreading-Activation-Algorithmus (siehe Definition 5.20) besteht in der Überführung der Raum-, Typ- und Themaontologie in einen Spreading-Activation-Graphen (siehe Definition 5.19) und der Bestimmung der Gewichtungsfaktoren (siehe Abschnitt 6.2).

Definition 5.19 [Spreading-Activation-Graph]

Ein *Spreading-Activation-Graph* G ist ein gerichteter Graph mit der Knotenmenge V , der Kantenmenge E und der Aktivierungsmenge A . Zusätzlich existiert die Menge der Gewichtungsfaktoren W und die Menge der Zerfallsfaktoren D .

$$G = (V, E, A, W, D)$$

- $v_i \in V$ bezeichnet dabei den Knoten mit Index bzw. ID i aus der Knotenmenge V .
- $e_t^{k \rightarrow i} \in E$ bezeichnet dabei die Kante von Knoten k zum Knoten i vom Typ t aus der Kantenmenge E .
- $E_i^{incoming} \subseteq E$ bezeichnet dabei die Menge der eingehenden Kanten vom Knoten v_i .

- $E_i^{outgoing} \subseteq E$ bezeichnet dabei die Menge der ausgehenden Kanten vom Knoten v_i .
- $a_i \in A$ bezeichnet den Aktivierungswert des Knotens v_i . Falls $a_i \notin A$ wird $a_i := 0$ definiert.
- $w_t^{forward} \in W$ bezeichnet den Gewichtungsfaktor der Kanten vom Typ t in Richtung der Kanten und ist im Intervall von $[0, 1]$. Falls $w_t^{forward} \notin W$ wird $w_t^{forward} := 0$ definiert.
- $w_t^{backward} \in W$ bezeichnet den Gewichtungsfaktor der Kanten vom Typ t in Rückrichtung der Kanten und ist im Intervall von $[0, 1]$. Falls $w_t^{backward} \notin W$ wird $w_t^{backward} := 0$ definiert.
- $d_t^{forward} \in D$ bezeichnet den Zerfallsfaktor der Kanten vom Typ t in Richtung der Kanten und ist im Intervall von $[0, 1]$. Falls $d_t^{forward} \notin D$ wird $d_t^{forward} := 0$ definiert.
- $d_t^{backward} \in D$ ist dabei der Zerfallsfaktor der Kanten vom Typ t in Rückrichtung der Kanten und ist im Intervall von $[0, 1]$. Falls $d_t^{backward} \notin D$ wird $d_t^{backward} := 0$ definiert.

□

Definition 5.20 [Spreading-Activation-Algorithmus]

Der *Spreading-Activation-Algorithmus* berechnet iterativ auf Basis eines Spreading-Activation-Graphen ausgehend von einer initialen Aktivierung für jeden Knoten des Graphen einen Aktivierungswert. Hierzu wird in jedem Iterationsschritt für jeden Knoten des Graphen der Aktivierungswert neu berechnet, falls ein Aktivierungswert eines Nachbarknotens sich geändert hat. [RSA04; SSS08; Aki+11] Der Aktivierungswert des Nachbarknotens wird mit Gewichtungs- und Zerfallsfaktor multipliziert und als Aktivierungsdelta für eine Nachverfolgung zwischengespeichert. Die Summe der Aktivierungsdeltas wird am Schluss jeder Iteration zum Aktivierungswert addiert.

Algorithmus 5.3 zeigt den für die Transformation angepassten Spreading-Activation-Graphen. Die Anpassung besteht in einer separaten Betrachtung von mehreren ausgehenden bzw. eingehenden Kanten und die kantentypabhängigen Gewichtungs- und Zerfallsfaktoren. Wie schon eingangs beschrieben ist dies genau der Vorteil des Spreading-Activation-Verfahrens. Damit ist es möglich, die Aktivierungsausbreitung, z.B. zwischen Strukturtypen und -elementen, und die Ausbreitung zwischen Strukturelementen untereinander zu beeinflussen. Dies ist essentiell, um die Intention der Anfrage in den Spreading-Activation-Graphen zu übertragen.

Je nach Art des Spreading-Activation-Graphen greifen unterschiedliche Mechanismen, die eine Terminierung des Algorithmus sicherstellen.

Wenn der Spreading-Activation-Graph azyklisch ist und die Zerfallsfaktoren $d < 1$ gewählt werden, terminiert der Algorithmus in endlicher Laufzeit, indem bei jeder Iteration die Aktivierung reduziert wird, bis der Grenzwert (*delta_Threshold*) greift.

Sinkt das Aktivierungsdelta unter den Grenzwert, werden keine neuen Knoten mehr aktiviert und somit wird in der nächsten Iteration keine Aktivierung von Nachbarknoten propagiert. Wird der Gewichtungsfaktor $w < 1$ gewählt, kann $d = 1$ als Sonderfall gewählt werden.

```

Input :  $G = (V, E, A, W, D)$ ;  $iteration\_Threshold$ 
/* Initialisiere Mengen der aktuellen aktivierten Knoten  $AV$ , der
neu aktivierten Knoten  $AV^{new}$  und der Aktivierungsdeltas  $\Delta_i$  */
1  $AV := AV^{new} := \emptyset$ ;  $\Delta_i := \emptyset$  for all  $v_i \in V$ ;
/* Ermittle aktivierte Knoten und Deltas anhand initialer
Aktivierung */
2 foreach  $a_i \in A$  do
| /* Füge Knoten zu der Menge der aktivierten Knoten  $AV$  hinzu */
3 |  $AV := AV \cup \{v_i\}$ ;
| /* Initialisiere  $\Delta_i$  mit  $\delta_{i \leftarrow i} := a_i$  */
4 |  $\Delta_i := \{\delta_{i \leftarrow i}\}$  with  $\delta_{i \leftarrow i} := a_i$ ;
/* Berechne Delta durch Delta der Nachbarknoten */
5 while  $AV \neq \emptyset \wedge iterations < iteration\_Threshold$  do
6 | foreach  $v_i \in V$  do
| /* Initialisiere  $\Delta_i^{new}$  für Knoten  $v_i$  */
7 |  $\Delta_i^{new} := \emptyset$ ;
| /* Berechne Delta für eingehende Kanten */
8 | foreach  $e_t^{k \rightarrow i} \in E_i^{incoming}$  do
9 | |  $v_k := Source(e_t^{k \rightarrow i})$  with  $t := Type(e_t^{k \rightarrow i})$ ;
| | /* Falls Quellknoten aktiviert wurde */
10 | | if  $v_k \in AV$  then
11 | | | if  $\delta_{i \leftarrow k}^{new} \notin \Delta_i^{new}$  then  $\delta_{i \leftarrow k}^{new} := 0$ ;
| | | /* Berechne Delta übergeben von Knoten  $v_k$  */
12 | | | foreach  $\delta \in (\Delta_k \setminus \{\delta_{k \leftarrow i}\})$  do
13 | | | |  $\delta_{i \leftarrow k}^{new} := \delta_{i \leftarrow k}^{new} + \delta * w_t^{forward} * d_t^{forward}$ ;
14 | | | |  $\Delta_i^{new} := \Delta_i^{new} \cup \{\delta_{i \leftarrow k}^{new}\}$ ;
| | /* Berechne Delta für ausgehende Kanten */
15 | | foreach  $e_t^{i \rightarrow k} \in E_i^{outgoing}$  do
16 | | |  $v_k := Target(e_t^{i \rightarrow k})$  with  $t := Type(e_t^{i \rightarrow k})$ ;
| | | /* Falls Zielknoten aktiviert wurde */
17 | | | if  $v_k \in AV$  then
18 | | | | if  $\delta_{i \leftarrow k}^{new} \notin \Delta_i^{new}$  then  $\delta_{i \leftarrow k}^{new} := 0$ ;
| | | | /* Berechne Delta übergeben von Knoten  $v_k$  */
19 | | | | foreach  $\delta \in (\Delta_k \setminus \{\delta_{k \leftarrow i}\})$  do
20 | | | | |  $\delta_{i \leftarrow k}^{new} := \delta_{i \leftarrow k}^{new} + \delta * w_t^{backward} * d_t^{backward}$ ;
21 | | | | |  $\Delta_i^{new} := \Delta_i^{new} \cup \{\delta_{i \leftarrow k}^{new}\}$ ;
| | /* Summiere Aktivierungsdeltas für Knoten  $i$  */
22 | |  $a_\delta := \sum_{\delta \in \Delta_i^{new}} \delta$ ;
| | /* Falls Summe der Aktivierungsdeltas groß genug */
23 | | if  $a_\delta > delta\_Threshold$  then
| | | /* Speichere neuen Aktivierungswert */
24 | | |  $A := A \setminus \{a_i\}$ ;  $a_i := a_i + a_\delta$ ;  $A := A \cup \{a_i\}$ ;
| | | /* Füge Knoten zu Liste der aktivierten Knoten hinzu */
25 | | |  $AV^{new} := AV^{new} \cup \{v_i\}$ ;
| /* Sichere  $\Delta_i^{new}$  für nächste Iteration */
26 | foreach  $v_i \in V$  do
27 | |  $\Delta_i := \Delta_i^{new}$ ;  $\Delta_i^{new} := \emptyset$ ;
| /* Sichere  $AV^{new}$  für nächste Iteration */
28 |  $AV := AV^{new}$ ;  $AV^{new} := \emptyset$ ;
Result :  $A$ 

```

Algorithmus 5.3 : Spreading-Activation-Algorithmus

Ist der Spreading-Activation-Graph hingegen nicht azyklisch, greift der Grenzwert (*iteration_Threshold*) und beendet die Berechnung von neuen Aktivierungswerten nach einer definierten Anzahl an Iterationsschritten. Daher beeinflusst der Iterationsgrenzwert bei Zyklen im Spreading-Activation-Graph die zyklische Ausbreitung.

Zusätzlich dürfen die Aktivierungswerte sich nur über Kanten zu Nachbarknoten, aber nicht zurück zu sich selbst ausbreiten. Daher muss zusätzlich bei der Erstellung des Spreading-Activation-Graphen sichergestellt werden, dass keine Kanten mit denselben Quell- und Zielknoten (d.h. $e^{v \rightarrow v} \notin E$) im Graphen vorhanden sind.

Die Ausgabe des Spreading-Activation-Algorithmus ist eine Menge an Aktivierungswerten für jeden aktivierten Knoten im Spreading-Activation-Graph. \square

Um die grundlegende Idee zu illustrieren, zeigt Beispiel 5.21 das Spreading-Activation-Verfahren an einer Strukturhierarchie.

Beispiel 5.21 [Spreading-Activation]

Ausgangspunkt ist ein Ausschnitt der Strukturhierarchie aus St. Lorenz (siehe Abbildung 5.4).

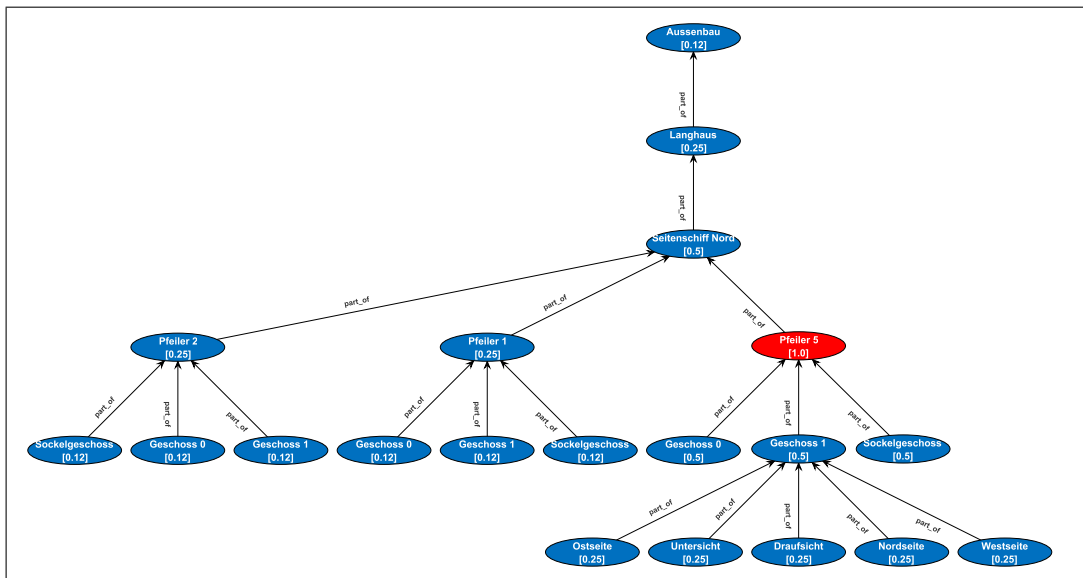


Abbildung 5.4.: Spreading-Activation

Für die initiale Aktivierung wählen wir einen Pfeiler (im Beispiel **Pfeiler 5**) aus, der mit dem Wert 1 aktiviert wird. Die Gewichtungsfaktoren werden so gewählt, dass sich die Aktivierung über die *part_of*-Beziehung in beide Richtungen ausbreitet. In diesem Beispiel wird die Aktivierung immer um die Hälfte verringert. Die endgültigen Aktivierungswerte werden in eckigen Klammern dargestellt. Die Aktivierungswerte werden iterativ durch die Ausbreitung von Aktivierungsdeltas berechnet. Der Spreading-Activation-Algorithmus durchläuft exemplarisch folgende Schritte:

Initialisierung:

- Erzeugen des Spreading-Activation-Graphen $G = (V, E, A, W, D)$:

$$V := \{v_{\text{Aussenbau}}, v_{\text{Langhaus}}, v_{\text{Seitenschiff Nord}}, v_{\text{Pfeiler 5}}, \\ v_{\text{Geschoss 0}}, v_{\text{Geschoss 1}}, v_{\text{Sockelgeschoss}}, v_{\text{Pfeiler 1}}, \dots\}$$

$$E := \left\{ e_{\text{part_of}}^{\text{Langhaus} \rightarrow \text{Aussenbau}}, e_{\text{part_of}}^{\text{Seitenschiff Nord} \rightarrow \text{Langhaus}}, \right. \\ \left. e_{\text{part_of}}^{\text{Pfeiler 5} \rightarrow \text{Seitenschiff Nord}}, e_{\text{part_of}}^{\text{Pfeiler 1} \rightarrow \text{Seitenschiff Nord}}, \right. \\ \left. e_{\text{part_of}}^{\text{Geschoss 0} \rightarrow \text{Pfeiler 5}}, e_{\text{part_of}}^{\text{Geschoss 1} \rightarrow \text{Pfeiler 5}}, e_{\text{part_of}}^{\text{Sockelgeschoss} \rightarrow \text{Pfeiler 5}}, \dots \right\}$$

$$A := \{a_{\text{Pfeiler 5}}\} \text{ mit } a_{\text{Pfeiler 5}} := 1$$

$$W := \left\{ w_{\text{part_of}}^{\text{backward}}, w_{\text{part_of}}^{\text{forward}} \right\} \text{ mit } w_{\text{part_of}}^{\text{backward}} := w_{\text{part_of}}^{\text{forward}} := 0,5$$

$$D := \left\{ d_{\text{part_of}}^{\text{backward}}, d_{\text{part_of}}^{\text{forward}} \right\} \text{ mit } d_{\text{part_of}}^{\text{backward}} := d_{\text{part_of}}^{\text{forward}} := 1,0$$

und setzen des Iterationsgrenzwertes: $\text{iteration_Threshold} := 100$

- Initialisieren der Menge (AV) der aktuell aktivierten Knoten, der Menge (AV^{new}) der Knoten, die neu aktiviert werden, und der Mengen der Aktivierungsdeltas (Δ_i):

$$(1) AV := AV^{\text{new}} := \emptyset; \quad \Delta_i := \emptyset \text{ für alle } v_i \in V$$

$$(2-4) \text{ Sei } a_i := a_{\text{Pfeiler 5}} \in A$$

$$(3) AV := AV \cup \{v_{\text{Pfeiler 5}}\}$$

$$(4) \Delta_{\text{Pfeiler 5}} := \{\delta_{\text{Pfeiler 5} \leftarrow \text{Pfeiler 5}}\} \text{ mit } \delta_{\text{Pfeiler 5} \leftarrow \text{Pfeiler 5}} := 1$$

Erste Iteration:

- $AV \neq \emptyset \wedge \text{iteration} := 1 < \text{iteration_Threshold} := 100$
- Iteration über die Knoten $v_i \in V$ (6-25). Es werden nur eingehende (8-10) bzw. ausgehende (15-17) Kanten betrachtet, deren Quell- bzw. Zielknoten in der Menge der aktivierten Knoten (AV) liegen. Für dieses Beispiel sind das anfangs nur die Knoten $v_{\text{Seitenschiff Nord}}, v_{\text{Geschoss 0}}, v_{\text{Geschoss 1}}$ und $v_{\text{Sockelgeschoss}}$ als direkte Nachbarn von $v_{\text{Pfeiler 5}}$, die in Frage kommen.

Betrachte Knoten $v_{\text{Seitenschiff Nord}}$ (6) Sei $v_i := v_{\text{Seitenschiff Nord}}$ (7) $\Delta_{\text{Seitenschiff Nord}}^{\text{new}} := \emptyset$

- Betrachte eingehende Kanten:

$$E_{\text{Seitenschiff Nord}}^{\text{incoming}} := \left\{ e_{\text{part_of}}^{\text{Pfeiler 5} \rightarrow \text{Seitenschiff Nord}}, e_{\text{part_of}}^{\text{Pfeiler 1} \rightarrow \text{Seitenschiff Nord}}, e_{\text{part_of}}^{\text{Pfeiler 2} \rightarrow \text{Seitenschiff Nord}} \right\}$$

(8) Sei $e_t^{k \rightarrow i} := e_{\text{part_of}}^{\text{Pfeiler 5} \rightarrow \text{Seitenschiff Nord}}$ (9) $v_k := v_{\text{Pfeiler 5}} = \text{Source}(e_{\text{part_of}}^{\text{Pfeiler 5} \rightarrow \text{Seitenschiff Nord}})$
 $t := \text{part_of} = \text{Type}(e_{\text{part_of}}^{\text{Pfeiler 5} \rightarrow \text{Seitenschiff Nord}})$ (10) Knoten $v_{\text{Pfeiler 5}} \in AV$ wurde aktiviert(11) Initialisiere Delta $\delta_{\text{Seitenschiff Nord} \leftarrow \text{Pfeiler 5}}^{\text{new}} := 0$ da $\Delta_{\text{Seitenschiff Nord}}^{\text{new}} = \emptyset$ (12) Sei $\delta := \delta_{\text{Pfeiler 5} \leftarrow \text{Pfeiler 5}} \in (\Delta_{\text{Pfeiler 5}} \setminus \{\delta_{\text{Pfeiler 5} \leftarrow \text{Seitenschiff Nord}}\})$

(13) Berechne Delta durch Gewichtung der Deltas von Pfeiler 5:

$$\begin{aligned} \delta_{\text{Seitenschiff Nord} \leftarrow \text{Pfeiler 5}}^{\text{new}} &:= \delta_{\text{Seitenschiff Nord} \leftarrow \text{Pfeiler 5}}^{\text{new}} + \delta * w_{\text{part_of}}^{\text{forward}} * d_{\text{part_of}}^{\text{forward}} \\ &= 0 + 1 * 0,5 * 1 = 0,5 \end{aligned}$$

(14) Füge neues Delta ein: $\Delta_{\text{Seitenschiff Nord}}^{\text{new}} := \emptyset \cup \{\delta_{\text{Seitenschiff Nord} \leftarrow \text{Pfeiler 5}}^{\text{new}}\}$ (8-10) Die Kanten $e_{\text{part_of}}^{\text{Pfeiler 1} \rightarrow \text{Seitenschiff Nord}}$ und $e_{\text{part_of}}^{\text{Pfeiler 2} \rightarrow \text{Seitenschiff Nord}}$ werden nicht weiter betrachtet, da die Knoten $v_{\text{Pfeiler 1}} \notin AV$ und $v_{\text{Pfeiler 2}} \notin AV$ nicht aktiviert wurden.

- Betrachte ausgehende Kanten: $E_{\text{Seitenschiff Nord}}^{\text{outgoing}} := \left\{ e_{\text{part_of}}^{\text{Seitenschiff Nord} \rightarrow \text{Langhaus}} \right\}$

(15-17) Da der Knoten $v_{\text{Langhaus}} \notin AV$ kann hier für Knoten $v_{\text{Seitenschiff Nord}}$ abgebrochen werden.

- Berechne Aktivierungswertdelta durch Summierung der Deltas:

$$(22) a_\delta := \sum_{\delta \in \Delta_{\text{Seitenschiff Nord}}^{\text{new}}} \delta = \delta_{\text{Seitenschiff Nord} \leftarrow \text{Pfeiler 5}}^{\text{new}} = 0,5$$

(23) Falls $a_\delta > \text{delta_Threshold} := 0,1$ (24) Addiere Aktivierungswertdelta zum alten Aktivierungswert: $A := A \setminus \{a_{\text{Seitenschiff Nord}}\};$
 $a_{\text{Seitenschiff Nord}} := a_{\text{Seitenschiff Nord}} + a_\delta = 0 + 0,5 = 0,5; \quad A := A \cup \{a_{\text{Seitenschiff Nord}}\}$ (25) Füge aktuellen Knoten zu Liste der aktivierten Knoten hinzu: $AV^{\text{new}} := AV^{\text{new}} \cup \{v_{\text{Seitenschiff Nord}}\}$

Betrachte Knoten $v_{\text{Geschoss 1}}$ (gleiches Verfahren für die Knoten $v_{\text{Geschoss 0}}$ und $v_{\text{Sockelgeschoss}}$)

(6) Sei $v_i := v_{\text{Geschoss 1}}$

(7) $\Delta_{\text{Geschoss 1}}^{\text{new}} := \emptyset$

• Betrachte eingehende Kanten:

$$E_{\text{Geschoss 1}}^{\text{incoming}} := \left\{ e_{\text{part_of}}^{\text{Ostseite} \rightarrow \text{Geschoss 1}}, e_{\text{part_of}}^{\text{Untersicht} \rightarrow \text{Geschoss 1}}, \dots \right\}$$

(8-10) Die Kanten in $E_{\text{Geschoss 1}}^{\text{incoming}}$ werden nicht weiter betrachtet, da die Knoten $v_{\text{Ostseite}} \notin AV$, $v_{\text{Untersicht}} \notin AV$ usw. nicht aktiviert wurden.

• Betrachte ausgehende Kanten:

$$E_{\text{Geschoss 1}}^{\text{outgoing}} := \left\{ e_{\text{part_of}}^{\text{Geschoss 1} \rightarrow \text{Pfeiler 5}} \right\}$$

(15) Sei $e_t^{i \rightarrow k} := e_{\text{part_of}}^{\text{Geschoss 1} \rightarrow \text{Pfeiler 5}}$

(16) $v_k := v_{\text{Pfeiler 5}} = \text{Target}(e_{\text{part_of}}^{\text{Geschoss 1} \rightarrow \text{Pfeiler 5}})$
 $t := \text{part_of} = \text{Type}(e_{\text{part_of}}^{\text{Geschoss 1} \rightarrow \text{Pfeiler 5}})$

(17) Knoten $v_{\text{Pfeiler 5}} \in AV$ wurde aktiviert

(18) Initialisiere Delta $\delta_{\text{Geschoss 1} \leftarrow \text{Pfeiler 5}}^{\text{new}} := 0$ da $\Delta_{\text{Geschoss 1}}^{\text{new}} = \emptyset$

(19) Sei $\delta := \delta_{\text{Pfeiler 5} \leftarrow \text{Pfeiler 5}} \in (\Delta_{\text{Pfeiler 5}} \setminus \{\delta_{\text{Pfeiler 5} \leftarrow \text{Geschoss 1}}\})$

(20) Berechne Delta durch Gewichtung der Deltas von Pfeiler 5:

$$\begin{aligned} \delta_{\text{Geschoss 1} \leftarrow \text{Pfeiler 5}}^{\text{new}} &:= \delta_{\text{Geschoss 1} \leftarrow \text{Pfeiler 5}}^{\text{new}} + \delta * w_{\text{part_of}}^{\text{backward}} * d_{\text{part_of}}^{\text{backward}} \\ &= 0 + 1 * 0,5 * 1 = 0,5 \end{aligned}$$

(21) Füge neues Delta ein: $\Delta_{\text{Geschoss 1}}^{\text{new}} := \emptyset \cup \{\delta_{\text{Geschoss 1} \leftarrow \text{Pfeiler 5}}^{\text{new}}\}$

• Berechne Aktivierungswertdelta durch Summierung der Deltas:

(22) $a_\delta := \sum_{\delta \in \Delta_{\text{Geschoss 1}}^{\text{new}}} \delta = \delta_{\text{Geschoss 1} \leftarrow \text{Pfeiler 5}}^{\text{new}} = 0,5$

(23) Falls $a_\delta > \text{delta_Threshold} := 0,1$

(24) Addiere Aktivierungswertdelta zum alten Aktivierungswert: $A := A \setminus \{a_{\text{Geschoss 1}}\};$
 $a_{\text{Geschoss 1}} := a_{\text{Geschoss 1}} + a_\delta = 0 + 0,5 = 0,5; \quad A := A \cup \{a_{\text{Geschoss 1}}\}$

(25) Füge aktuellen Knoten zu Liste der aktivierten Knoten hinzu: $AV^{\text{new}} := AV^{\text{new}} \cup \{v_{\text{Geschoss 1}}\}$

Abschluss erste Iteration und Vorbereitung der nächsten Iteration

(26-27) *Sichern der Aktivierungsdeltas für nächste Iteration.*

$$\begin{aligned}\Delta_{\text{Seitenschiff Nord}} &:= \{\delta_{\text{Seitenschiff Nord} \leftarrow \text{Pfeiler 5}}\} \\ \Delta_{\text{Geschoss 0}} &:= \{\delta_{\text{Geschoss 0} \leftarrow \text{Pfeiler 5}}\} \\ \Delta_{\text{Geschoss 1}} &:= \{\delta_{\text{Geschoss 1} \leftarrow \text{Pfeiler 5}}\} \\ \Delta_{\text{Sockelgeschoss}} &:= \{\delta_{\text{Sockelgeschoss} \leftarrow \text{Pfeiler 5}}\} \\ \Delta_i^{\text{new}} &:= \emptyset \text{ für alle Knoten } v_i\end{aligned}$$

(28) *Setzen der aktivierten Knoten im Graphen.*

$$AV := \{v_{\text{Seitenschiff Nord}}, v_{\text{Geschoss 0}}, v_{\text{Geschoss 1}}, v_{\text{Sockelgeschoss}}\}; AV^{\text{new}} := \emptyset$$

Zweite Iteration:

- $AV \neq \emptyset \wedge \text{iteration} := 2 < \text{iteration_Threshold} := 100$
- *Iteration über die Knoten $v_i \in V$ (6-25). Es werden nur eingehende (8-10) bzw. ausgehende (15-17) Kanten betrachtet, deren Quell- bzw. Zielknoten in der Menge der aktivierten Knoten (AV) liegen. In diesem Beispiel werden in der zweiten Iteration die Knoten $v_{\text{Pfeiler 1}}, v_{\text{Pfeiler 2}}, v_{\text{Pfeiler 5}}, v_{\text{Langhaus}}$, und die weiteren Gebäudeteile von $v_{\text{Geschoss 1}}$ als direkte Nachbarn der Knoten in AV betrachtet.*

Betrachte Knoten $v_{\text{Pfeiler 5}}$

(6) *Sei $v_i := v_{\text{Pfeiler 5}}$*

(7) $\Delta_{\text{Pfeiler 5}}^{\text{new}} := \emptyset$

- *Betrachte eingehende Kanten:*

$$E_{\text{Pfeiler 5}}^{\text{incoming}} := \left\{ e_{\text{part_of}}^{\text{Geschoss 0} \rightarrow \text{Pfeiler 5}}, e_{\text{part_of}}^{\text{Geschoss 1} \rightarrow \text{Pfeiler 5}}, e_{\text{part_of}}^{\text{Sockelgeschoss} \rightarrow \text{Pfeiler 5}} \right\}$$

(8) *Sei $e_t^{k \rightarrow i} := e_{\text{part_of}}^{\text{Geschoss 0} \rightarrow \text{Pfeiler 5}}$*

(9) $v_k := v_{\text{Geschoss 0}} = \text{Source}(e_{\text{part_of}}^{\text{Geschoss 0} \rightarrow \text{Pfeiler 5}})$

$t := \text{part_of} = \text{Type}(e_{\text{part_of}}^{\text{Geschoss 0} \rightarrow \text{Pfeiler 5}})$

(10) *Knoten $v_{\text{Geschoss 0}} \in AV$ wurde aktiviert*

(11) *Initialisiere Delta $\delta_{\text{Pfeiler 5} \leftarrow \text{Geschoss 0}}^{\text{new}} := 0$ da $\Delta_{\text{Pfeiler 5}}^{\text{new}} = \emptyset$*

(12-14) *Da $\Delta_{\text{Geschoss 0}} \setminus \{\delta_{\text{Geschoss 0} \leftarrow \text{Pfeiler 5}}\} = \emptyset$ erfolgt keine Aktivierungsausbreitung ausgehend vom Knoten $v_{\text{Geschoss 0}}$ zurück zum Knoten $v_{\text{Pfeiler 5}}$, da der Knoten $v_{\text{Geschoss 0}}$ in der ersten Iteration vom Knoten $v_{\text{Pfeiler 5}}$ aktiviert wurde.*

(8-14) Die Kanten $e_{\text{part_of}}^{\text{Geschoss 1} \rightarrow \text{Pfeiler 5}}$ und $e_{\text{part_of}}^{\text{Sockelgeschoss} \rightarrow \text{Pfeiler 5}}$ tragen aus demselben Grund nicht zur Veränderung des Aktivierungswertes des Knoten $v_{\text{Pfeiler 5}}$ bei.

- Betrachte ausgehende Kanten: $E_{\text{Pfeiler 5}}^{\text{outgoing}} := \left\{ e_{\text{part_of}}^{\text{Pfeiler 5} \rightarrow \text{Seitenschiff Nord}} \right\}$

(15-17) Der Knoten $v_{\text{Seitenschiff Nord}} \in AV$ ist zwar in der ersten Iteration aktiviert worden, wurde aber vom Knoten $v_{\text{Pfeiler 5}}$ aktiviert, daher gilt auch hier wieder $\Delta_{\text{Seitenschiff Nord}} \setminus \{ \delta_{\text{Seitenschiff Nord} \leftarrow \text{Pfeiler 5}} \} = \emptyset$.

- Da keine neuen Aktivierungsdeltas berechnet wurden, bleibt der Aktivierungswert des Knotens $v_{\text{Pfeiler 5}}$ unverändert.

Betrachte Knoten $v_{\text{Pfeiler 1}}$ (gleiches Verfahren für die Knoten $v_{\text{Pfeiler 2}}, v_{\text{Langhaus}}$ und die weiteren Gebäudeteile von $v_{\text{Geschoss 1}}$)

(6) Sei $v_i := v_{\text{Pfeiler 1}}$

(7) $\Delta_{\text{Pfeiler 1}}^{\text{new}} := \emptyset$

- Betrachte eingehende Kanten:

$$E_{\text{Pfeiler 1}}^{\text{incoming}} := \left\{ e_{\text{part_of}}^{\text{Geschoss 0} \rightarrow \text{Pfeiler 1}}, e_{\text{part_of}}^{\text{Geschoss 1} \rightarrow \text{Pfeiler 1}}, e_{\text{part_of}}^{\text{Sockelgeschoss} \rightarrow \text{Pfeiler 1}} \right\}$$

Hinweis: In diesem Beispiel sind Knotenbezeichner doppelt verwendet worden. In einer Implementierung werden die Knoten nicht mit ihrem Namen sondern mit ihrer ID identifiziert, die dann eindeutig sein muss. Im Folgenden sind die Knoten $v_{\text{Geschoss 0}}, v_{\text{Geschoss 1}}$ und $v_{\text{Sockelgeschoss}}$ gemeint, die in Abbildung 5.4 mit Knoten $v_{\text{Pfeiler 1}}$ verknüpft sind.

(8-10) Die Kanten in $E_{\text{Pfeiler 1}}^{\text{incoming}}$ werden nicht weiter betrachtet, da die Knoten $v_{\text{Geschoss 0}} \notin AV$, $v_{\text{Geschoss 1}} \notin AV$ usw. nicht aktiviert wurden.

- Betrachte ausgehende Kanten:

$$E_{\text{Pfeiler 1}}^{\text{outgoing}} := \left\{ e_{\text{part_of}}^{\text{Pfeiler 1} \rightarrow \text{Seitenschiff Nord}} \right\}$$

(15) Sei $e_t^{i \rightarrow k} := e_{\text{part_of}}^{\text{Pfeiler 1} \rightarrow \text{Seitenschiff Nord}}$

(16) $v_k := v_{\text{Seitenschiff Nord}} = \text{Target}(e_{\text{part_of}}^{\text{Pfeiler 1} \rightarrow \text{Seitenschiff Nord}})$
 $t := \text{part_of} = \text{Type}(e_{\text{part_of}}^{\text{Pfeiler 1} \rightarrow \text{Seitenschiff Nord}})$

(17) Knoten $v_{\text{Seitenschiff Nord}} \in AV$ wurde aktiviert

(18) Initialisiere Delta $\delta_{\text{Pfeiler 1} \leftarrow \text{Seitenschiff Nord}}^{\text{new}} := 0$ da $\Delta_{\text{Pfeiler 1}}^{\text{new}} = \emptyset$

(19) Sei $\delta := \delta_{\text{Seitenschiff Nord} \leftarrow \text{Pfeiler 5}} \in (\Delta_{\text{Seitenschiff Nord}} \setminus \{ \delta_{\text{Pfeiler 1} \leftarrow \text{Seitenschiff Nord}} \})$

(20) Berechne Delta durch Gewichtung der Deltas von Pfeiler 5:

$$\begin{aligned} \delta_{\text{Pfeiler 1} \leftarrow \text{Seitenschiff Nord}}^{\text{new}} &:= \delta_{\text{Pfeiler 1} \leftarrow \text{Seitenschiff Nord}}^{\text{new}} + \delta * w_{\text{part_of}}^{\text{backward}} * d_{\text{part_of}}^{\text{backward}} \\ &= 0 + 0,5 * 0,5 * 1 = 0,25 \end{aligned}$$

(21) Füge neues Delta ein: $\Delta_{\text{Pfeiler 1}}^{\text{new}} := \emptyset \cup \{\delta_{\text{Pfeiler 1} \leftarrow \text{Seitenschiff Nord}}^{\text{new}}\}$

- Berechne Aktivierungswertdelta durch Summierung der Deltas:

(22) $a_{\delta} := \sum_{\delta \in \Delta_{\text{Pfeiler 1}}^{\text{new}}} \delta = \delta_{\text{Pfeiler 1} \leftarrow \text{Seitenschiff Nord}}^{\text{new}} = 0,25$

(23) Falls $a_{\delta} > \text{delta_Threshold} := 0,1$

(24) Addiere Aktivierungswertdelta zum alten Aktivierungswert: $A := A \setminus \{a_{\text{Pfeiler 1}}\}$;
 $a_{\text{Pfeiler 1}} := a_{\text{Pfeiler 1}} + a_{\delta} = 0 + 0,25 = 0,25$; $A := A \cup \{a_{\text{Pfeiler 1}}\}$

(25) Füge aktuellen Knoten zu Liste der aktivierten Knoten hinzu: $AV^{\text{new}} := AV^{\text{new}} \cup \{v_{\text{Pfeiler 1}}\}$

Abschluss zweite Iteration und Vorbereitung der nächsten Iteration

(26-27) Sichern der Aktivierungsdeltas für nächste Iteration.

$$\begin{aligned} \Delta_{\text{Pfeiler 1}} &:= \{\delta_{\text{Pfeiler 1} \leftarrow \text{Seitenschiff Nord}}\} \\ \Delta_{\text{Pfeiler 2}} &:= \{\delta_{\text{Pfeiler 2} \leftarrow \text{Seitenschiff Nord}}\} \\ \Delta_{\text{Langhaus}} &:= \{\delta_{\text{Langhaus} \leftarrow \text{Seitenschiff Nord}}\} \\ \Delta_{\text{Ostseite}} &:= \{\delta_{\text{Ostseite} \leftarrow \text{Geschoss 1}}\} \\ \Delta_{\text{Untersicht}} &:= \{\delta_{\text{Untersicht} \leftarrow \text{Geschoss 1}}\} \\ \Delta_{\text{Draufsicht}} &:= \{\delta_{\text{Draufsicht} \leftarrow \text{Geschoss 1}}\} \\ \Delta_{\text{Nordseite}} &:= \{\delta_{\text{Nordseite} \leftarrow \text{Geschoss 1}}\} \\ \Delta_{\text{Westseite}} &:= \{\delta_{\text{Westseite} \leftarrow \text{Geschoss 1}}\} \\ \Delta_i^{\text{new}} &:= \emptyset \text{ für alle Knoten } v_i \end{aligned}$$

(28) Setzen der aktivierten Knoten im Graphen.

$$AV := \{v_{\text{Pfeiler 1}}, v_{\text{Pfeiler 2}}, v_{\text{Langhaus}}, v_{\text{Ostseite}}, v_{\text{Untersicht}}, v_{\text{Draufsicht}}, v_{\text{Nordseite}}, v_{\text{Westseite}}\};$$

$$AV^{\text{new}} := \emptyset$$

Nach der zweiten Iteration verändert sich der Aktivierungswert des Knotens $v_{\text{Pfeiler 5}}$ nicht, da die Rückausbreitung der Aktivierungswerte verhindert wird. Die weiteren Iterationen verlaufen analog. Der Algorithmus terminiert in diesem Fall, da die Aktivierungsdeltas durch die Halbierung unter $0,1$ fallen (siehe Zeile 23) und eine Rückausbreitung der Aktivierungswerte verhindert wird (siehe Zeile 12 und 19). \square

5.5. Transformation thematischer Anfragen

Die Anfragetransformation von elementaren thematischen Anfragen ist, wie auch die allgemeine Anfragetransformation, in eine Anfragetransformation im Quellsystem und eine Anfragetransformation im Zielsystem aufgeteilt.

Algorithmus 5.4 zeigt den Ablauf im Quellsystem.

```

1 Function transformTopicalQuerySource(t)
   Input :  $t_{input}$  = TOPICAL ID T-id UP? DOWN?
   Output :  $t_{output}$  = TOPICAL ID Path UP? DOWN?
   /* Setze ID des Themas der Eingabeanfrage in den Ausgabepfad */
2   Path := T-id;
   /* Setze aktuelles Thema */
3   currentTopic := the Topic with ID T-id;
   /* Bestimme Oberthema des aktuellen Themas */
4   superTopic := the Topic which is super_topic of currentTopic;
   /* Bestimme Pfad von Oberthemen */
5   while currentTopic has superTopic do
6     currentTopic := superTopic;
7     T-id := getID(currentTopic);
   /* Füge ID des aktuellen Themas an den Pfad an */
8     Path := T-id + '/' + Path;
9     superTopic := the Topic which is super_topic of currentTopic;
   /* Rückgabe einer elementaren thematischen Anfrage mit Pfad von
   IDs und Übernahme der UP bzw. DOWN Option */
   Result : TOPICAL ID Path UP? DOWN?

```

Algorithmus 5.4 : Thematische Anfragetransformation im Quellsystem

Als Eingabe bekommt der Algorithmus für die thematische Anfragetransformation im Quellsystem eine elementare thematische Anfrage.

Die Grundidee des Algorithmus ist es, ausgehend vom selektierten Thema in der elementaren thematischen Anfrage den Pfad entlang der *super_topic*-Beziehung bis zum Top-Thema zu folgen und dabei die Anfrage mit einer Pfad-Anfrage zu transformieren.

Der Algorithmus durchläuft dabei folgende Schritte.

1. Setzen der ID eines Themas, das in der Eingabeanfrage t_{input} spezifiziert ist als Startpunkt des Pfades entlang der *super_topic*-Beziehung (Zeile 2).
2. Bestimmen des direkten Oberthemas des selektierten Themas (Zeile 3-4).
3. Anfügen der ID des Oberthemas an den Anfang des Pfades (Zeile 6-9).
4. Wiederholen der Bestimmung von Oberthemen bis das Top-Thema in der Themenhierarchie erreicht ist und Anfügen der IDs der Themen an den Pfad (Zeile 5).
5. Rückgabe einer elementaren thematischen Anfrage mit einem ID-Pfad.

Die Transformation der thematischen Anfrage im Quellsystem greift dabei auf keine Informationen im Zielsystem zu. Die Transformation kann unabhängig durchgeführt werden.

Beispiel 5.22 zeigt eine elementare thematische Anfragetransformation.

Beispiel 5.22 [Thematische Anfragetransformation im Quellsystem]

Betrachten wir einen Ausschnitt des Themengraphen von St. Lorenz (siehe Abbildung 5.5).

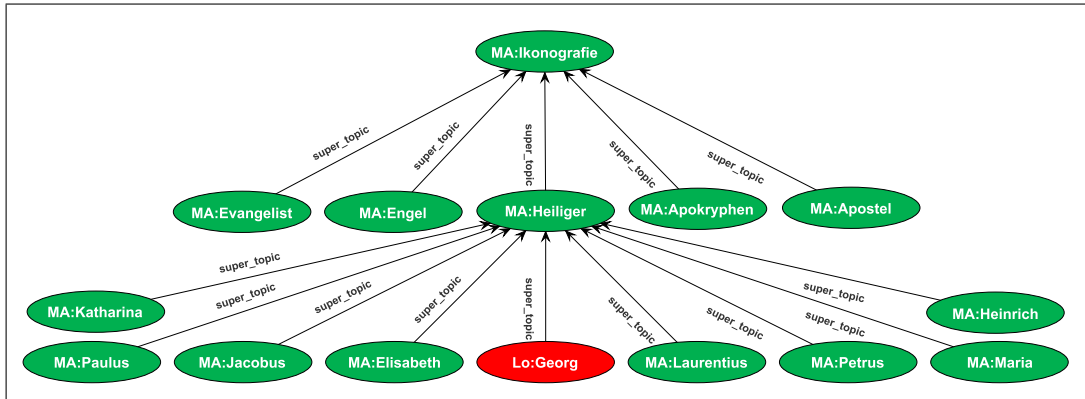


Abbildung 5.5.: Themengraph von St. Lorenz

Ausgehend von einer elementaren thematischen Anfrage t_{input} nach einem Heiligen, in diesem Beispiel *Georg*, wird diese in eine Anfrage t_{output} , die den Pfad ausgehend vom Thema *Georg* hin zum Thema *Ikonografie* enthält, transformiert.

- t_{input} :

TOPICAL ID Lo:Georg DOWN

- t_{output} :

TOPICAL ID MA:Ikonografie / MA:Heilig / Lo:Georg DOWN

□

Die Transformation im Quellsystem ist eine vorbereitende Transformation für die Transformation im Zielsystem. Beide Transformationen zusammen ergeben letztendlich die thematische Anfragetransformation.

Es sei hier erneut darauf hingewiesen, dass das Zielsystem nicht zwangsweise ein anderes System sein muss als das Quellsystem.

Die Aufteilung in zwei separate Transformationen dient einerseits dazu, genau aufzuzeigen welche Schritte notwendig sind und welche Daten übertragen werden, andererseits erlaubt es verschiedene Transformationsstrategien genauer zu betrachten.

Für die Anfragetransformation im Zielsystem von thematischen Anfragen werden in den folgenden Unterabschnitten zwei Strategien betrachtet und zusammen mit der Transformation im Quellsystem auf ihre intentionserhaltende Eigenschaft untersucht.

5.5.1. Relaxation durch Verallgemeinerung

In [MT14] beschreiben die Autoren ein Relaxationsverfahren für eine Anfragetransformation, die auf Taxonomien zugreift, um die Selektionsbedingungen der Anfrage zu verändern. Dieses Verfahren lässt sich auf die Transformation von thematischen elementaren Unteranfragen übertragen. Als Ergebnis erhält man ein Verfahren, das Themen in einer Anfrage durch Oberthemen ersetzt und somit die Anfrage relaxiert.

In Algorithmus 5.5 ist die Transformation der Anfrage im Zielsystem beschrieben.

```
1 Function transformTopicalQueryTarget (tinput)
   Input : tinput = TOPICAL ID Path UP? DOWN?
   Output : toutput = TOPICAL ID T-id UP? DOWN?
   /* Finde erstes Thema im Pfad, das lokal existiert */
2   repeat
     /* Nehme letztes Element im Pfad und setze ID */
3     T-id := Last (Path);
     /* Setze aktuelles Thema */
4     currentTopic := Topic with ID T-id;
     /* Lösche letzte ID im Pfad */
5     DeleteLast (Path);
6   until currentTopic exists locally;
   Result : TOPICAL ID T-id UP? DOWN?
```

Algorithmus 5.5 : Thematische Anfragetransformation im Zielsystem

Die Eingabe des Algorithmus ist eine Anfrage, die durch die Transformation der Ursprungsanfrage mit Algorithmus 5.4 entsteht. Die transformierte Anfrage ist eine thematische Anfrage nach einem Thema.

Die Grundidee des Algorithmus besteht darin, dass möglichst das speziellste Thema im Pfad ausgewählt wird, welches im Themengraph des Zielsystems existiert.

Der Algorithmus durchläuft dabei folgende Schritte.

1. Aus der Eingabeanfrage t_{input} wird die ID eines Themas dem Pfad entnommen. Dabei wird die letzte ID im Pfad betrachtet (Zeile 3-5).
2. Es erfolgt eine Prüfung, ob das Thema mit der gegebenen ID im Themengraphen existiert (Zeile 6).
3. Existiert das Thema, wird eine neue elementare thematische Anfrage erstellt und zurückgegeben.
4. Falls das Thema nicht im Themengraphen existiert, wird die nächste ID im Pfad überprüft (Zeile 2).

Die Transformation kann wieder unabhängig durchgeführt werden. Außer der Anfrage vom Quellsystem sind keine weiteren Informationen aus dem Quellsystem notwendig.

Dieses Verfahren setzt voraus,

- dass sich Quell- und Zielsystem eine Themenhierarchie teilen, die sich ab einer gewissen Ebene in den Unterthemen unterscheiden kann,
- dass jedes Thema nur ein Oberthema besitzt, d.h. die *super_topic*-Beziehung funktional ist,
- dass die Themengraphen trotz der Abweichungen letztendlich die gleiche semantische Grundlage aufweisen.

Beispiel 5.23 [Thematische Anfragetransformation im Zielsystem]

Betrachten wir einen Ausschnitt des Themengraphen von St. Sebald (siehe Abbildung 5.6). Im Gegensatz zum Themengraphen von St. Lorenz existiert hier das Thema *Georg* nicht.

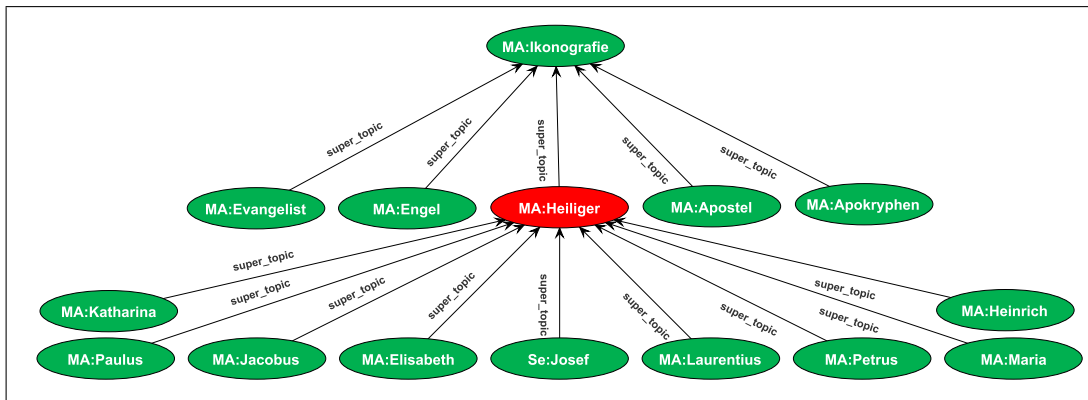


Abbildung 5.6.: Themengraph von St. Sebald

Ausgehend von einer thematischen Pfad-Anfrage τ_{input} aus dem Beispiel 5.22 wird diese in eine Anfrage τ_{output} , die das erste gemeinsame Thema *Heiliger* enthält, transformiert.

- τ_{input} :

`TOPICAL ID MA:Ikonografie / MA:Heiliger / Lo:Georg DOWN`

- τ_{output} :

`TOPICAL ID MA:Heiliger DOWN`

Obwohl das Thema *Georg* nicht im Themengraphen von St. Sebald vorkommt, kann eine Relaxation der Anfrage durchgeführt werden. Der Suchraum wird dabei durch das Oberthema *Heiliger* und der *DOWN*-Option in der Anfrage τ_{output} erweitert. Das Thema *Josef*, das nur im Themengraphen von St. Sebald existiert, wird so auch in den Suchraum mitaufgenommen. \square

Für diese Variante der thematischen Anfragetransformation muss noch die Intentionserhaltung betrachtet werden.

Überprüfung 5.24 [Intentionserhaltende thematische Anfragetransformation]

Die Intentionserhaltung der thematischen Anfragetransformation wird für die möglichen Ausgabeanfragen bei einer gegebenen Eingabeanfrage betrachtet. Wie bereits bei der Betrachtung der Strukturhaltung, wird die unabhängige Anfragetransformation im Quell- und Zielsystem zusammen betrachtet.

Gehen wir dazu von einer generischen Form einer elementaren thematischen Unteranfrage als Eingabeanfrage t_{input} im Quellsystem aus:

```
TOPICAL ID T-0 UP? DOWN?
```

Nach der Transformation der Anfrage im Quellsystem ist das Ergebnis eine thematische Anfrage mit einem Themenpfad $t_{\text{intermediate}}$:

```
TOPICAL ID T-n/.../T-0 UP? DOWN?
```

Im Zielsystem kann diese Anfrage in verschiedene Anfragen transformiert werden.

- Eine Anfrage $t_{\text{output}-0}$:

```
TOPICAL ID T-0 UP? DOWN?
```

- Anfragen $t_{\text{output}-1}, \dots, t_{\text{output}-n}$:

```
TOPICAL ID T-1 UP? DOWN?  
...  
TOPICAL ID T-n UP|DOWN)?
```

Welche Anfrage als Ergebnis der Transformation ausgewählt wird, hängt davon ab, welches Thema im Themengraphen des Zielsystems vorhanden ist.

Ist die Anfrage $t_{\text{output}-0}$ das Ergebnis der Transformation, ist die Intentionserhaltung nach dem 1. Fall der Definition 5.8 gewährleistet, da das Thema der Zielanfrage der Quellenfrage entspricht. Dieser Fall tritt ein, falls im Themengraphen des Zielsystems das Thema der Quellenfrage vorhanden ist.

Ist die Anfrage $t_{\text{output}-1}$ das Ergebnis der Transformation, ist die Intentionserhaltung nach dem 2. Fall der Definition 5.8 sichergestellt, da das Thema $T-1$ ein Oberthema vom Thema $T-0$ ist:

$$\{T-0\} \sqsubseteq \exists \text{super_topic_transitive}.\{T-1\}$$

Tritt dieser Fall ein, so ist die Transformation eine Relaxation.

Für die Anfrage $t_{\text{output}-2}$ bis $t_{\text{output}-n}$ lässt sich die Intentionserhaltung analog zu $t_{\text{output}-1}$ begründen.

Es gilt somit $t_{\text{input}} \approx t_{\text{output}}$.

Diese Art der Intentionserhaltung setzt voraus, dass es mindestens ein gemeinsames Obertema im Themenpfad der Anfrage $t_{\text{intermediate}}$ in den Themengraphen des Quell- bzw. Zielsystems gibt. \square

Diese Strategie der Anfragerelaxation von thematischen Anfragen ist eine Möglichkeit eine Verbundanfrage zu realisieren, die intentionserhaltend ist. Sie betrachtet nur die Themenhierarchie aufgebaut aus der *super_topic*-Beziehung.

5.5.2. Spreading-Activation für die thematische Anfragerelaxation

Die zweite Strategie, die in dieser Arbeit betrachtet wird, nutzt zusätzlich zu dem Relaxationsverfahren aus Unterabschnitt 5.5.1 ein für den Anwendungsfall angepasstes Spreading-Activation-Verfahren für ein Ranking der Suchergebnisse. In [Sun+13] wurde das Spreading-Activation-Verfahren ebenfalls eingesetzt, um ein Ranking auf den Suchergebnissen zu berechnen. Die Grundidee, die Aktivierungswerte des Spreading-Activation-Verfahrens angewendet auf eine Ontologie als Ranking zu verwenden, lässt sich auf den Themengraphen übertragen.

Um das Verfahren anwenden zu können, wird der Themengraph als Spreading-Activation-Graph (siehe Abbildungsdefinition 5.25) interpretiert.

Abbildungsdefinition 5.25 [Thematische Ontologie als Spreading-Activation-Graph] Ausgangspunkt für die Abbildung der Themaontologie auf einen Spreading-Activation-Graphen $G^{\text{topical}} = (V, E, A, W, D)$ ist die Ontologie $O_{\text{schema}}^{\text{topical}}$ aus Unterabschnitt 3.4.2 mit:

$$V = \text{Topic}$$

Individuen des Konzeptes *Topic* werden als Knoten im Spreading-Activation-Graphen G^{topical} abgebildet.

$$E = \text{super_topic} \sqcup \text{related_topic} \sqcup \text{sameAS}$$

Rollenaussagen der Rollen *super_topic*, *related_topic* und *sameAS* werden als Kanten im Spreading-Activation-Graphen G^{topical} abgebildet.

Die Mengen A , W und D können durch eine Konfiguration vor der Anwendung des Spreading-Activation-Algorithmus definiert werden und beeinflussen damit das Verfahren. \square

In Beispiel 5.26 ist ein Ausschnitt eines Themengraphen als Spreading-Activation-Graph mit Aktivierungswerten zu sehen.

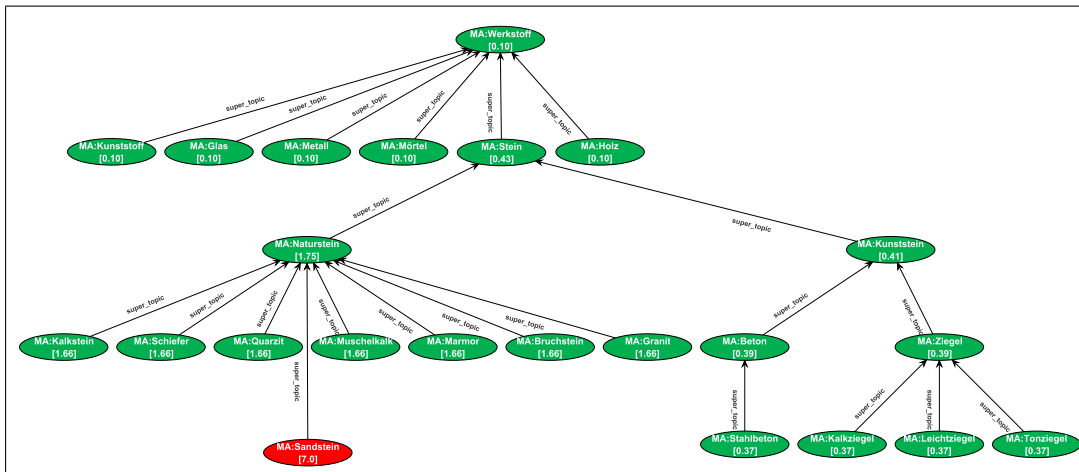


Abbildung 5.7.: Spreading-Activation-Graph für den Themengraph von St. Sebald

Beispiel 5.26 [Spreading-Activation-Graph für St. Sebald]

Abbildung 5.7 zeigt einen Ausschnitt des Themengraphen von St. Sebald als Spreading-Activation-Graph.

Die Themen sind als Knoten repräsentiert, die *super_topic*- und *sameAS*-Beziehungen sind als Kanten repräsentiert. Für die Knotenbeschriftung wurde zur Illustration der Name des Themas verwendet, der nicht eindeutig sein muss. Der Wert im Knoten ist der berechnete Aktivierungswert. Je höher der Aktivierungswert, desto besser passt das Thema zur Anfrage. Die roten Knoten zeigen die initial aktivierten Knoten. □

Ähnlich zu der Transformation der elementaren thematischen Anfrage in Algorithmus 5.5 ist in Algorithmus 5.6 die Transformation der Anfrage im Zielsystem durch den Einsatz von Spreading-Activation beschrieben.

Die Eingabe des Algorithmus ist eine thematische Pfad-Anfrage, die durch die Transformation der Ursprungsanfrage mit Algorithmus 5.4 entsteht.

Die Grundidee des Algorithmus ist es, ausgehend von den Themen im Pfad zusätzlich zu der Anfragetransformation durch Einsatz des Spreading-Activation-Verfahrens, Themen im Themengraphen mit Hilfe des Aktivierungswertes zu bewerten.

Die berechneten Aktivierungswerte können als Ranking interpretiert und anschließend auf die Ergebnisse der transformierten Anfrage, z.B. Dokumente, übertragen werden. Damit lassen sich die Ergebnisse nach Relevanz sortieren. Hier unterscheidet sich das Spreading-Activation-Verfahren zum vorigen Verfahren (siehe Unterabschnitt 5.5.1).

```

1 Function transformTopicalQueryTarget (tinput)
   Input : tinput = TOPICAL ID Path UP? DOWN?
   Output : toutput
   /* Phase 1: Transformation der thematischen Anfrage */
2   foreach Topic in Path do
3     if currentTopic exists locally then
4       /* Erstellen der transformierten Anfrage */
5       T-id := getID (currentTopic) ;
6       toutput := toutput + UNION TOPICAL ID T-id UP? DOWN?;
7       foreach Topic is sameAS or related_topic to currentTopic do
8         /* Ermittle ID des Themas, das in einer sameAS- oder
9         related_topic-Beziehung zum aktuellen Thema steht */
10        sameAsTopic := the Topic which is sameAS or related_topic to currentTopic;
11        T-id := getID (sameAsTopic) ;
12        /* Erstellen der transformierten Anfrage */
13        toutput := toutput + UNION TOPICAL ID T-id UP? DOWN?;
14    /* Phase 2: Berechnung der Aktivierungen */
15    /* Aufbau des Spreading-Activation-Graphen */
16    V := Topic;
17    E := super_topic □ related_topic □ sameAS;
18    /* Laden der initialen Aktivierung, Gewichtungs- und
19    Zerfallsfaktoren abhängig von der UP- bzw. DOWN-Option der
20    Eingabeanfrage */
21    Load A, W, D;
22    /* Anwendung des Spreading-Activation-Algorithmus */
23    Set Gtopical := (V,E,A,W,D);
24    R := SA (Gtopical);
25    /* Rückgabe der transformierten Anfrage und der berechneten
26    Aktivierungswerte */
27    Result : toutput; R

```

Algorithmus 5.6 : Thematische Anfragetransformation im Zielsystem mit Spreading-Activation

Der Algorithmus durchläuft dabei folgende Schritte.

1. Für jedes Thema des Themenpfades, das im Themengraphen des Zielsystems vorhanden ist, wird eine Anfrage erstellt und die UP- bzw. DOWN-Option der Eingabeanfrage übernommen (Zeile 2).
2. Zusätzlich wird für jedes Thema, das in einer *sameAS*- oder *related_topic*-Beziehung zu einem Thema aus dem Themenpfad steht, eine Anfrage erstellt und die UP- bzw. DOWN-Option der Eingabeanfrage übernommen (Zeile 6).
3. Die zweite Phase erstellt einen Spreading-Activation-Graph und lädt die initiale Aktivierung, Gewichtungs- und Zerfallsfaktoren nach Definition 5.28 oder Definition 5.30 (Zeile 10-13). Die initiale Aktivierung, Gewichtungs- und Zerfallsfaktoren erlauben es auf die Ausbreitung der Aktivierung Einfluss zu nehmen.
4. Folgt die Anwendung des Spreading-Activation-Verfahrens auf den Spreading-Activation-Graph (Zeile 14).

5. Als Ergebnis liegt nun die Anfrage t_{output} und die Menge der Aktivierungswerte für Themen im Zielsystem vor.

Die Transformation kann auch hier wieder unabhängig durchgeführt werden. Außer der Anfrage vom Quellsystem sind keine weiteren Informationen vom Quellsystem notwendig.

Es muss, wie auch für das vorherige Verfahren, für diese Variante der thematischen Anfrage-transformation die Intentionserhaltung betrachtet werden.

Überprüfung 5.27 [Intentionserhaltende thematische Anfrage-transformation mit Spreading-Activation]

Die Intentionserhaltung der thematischen Anfrage-transformation wird für die möglichen Ausgabeanfragen bei einer gegebenen Eingabeanfrage betrachtet. Wie bereits bei der Betrachtung der Strukturserhaltung, wird die Anfrage-transformation im Quell- und Zielsystem in einer Einheit betrachtet.

Die Ausgangssituation ist analog zur Betrachtung der Intentionserhaltung in Unterabschnitt 5.5.1.

Gehen wir dazu wieder von einer generischen Form einer elementaren thematischen Unteranfrage als Eingabeanfrage t_{input} im Quellsystem aus:

TOPICAL ID $T-0$ UP? DOWN?

Nach der Transformation der Anfrage im Quellsystem ist das Ergebnis eine thematische Anfrage mit einem Themenpfad $t_{\text{intermediate}}$:

TOPICAL ID $T-n/\dots/T-0$ UP? DOWN?

Im Zielsystem beinhaltet die transformierte Anfrage:

- Eine Anfrage $t_{\text{output}-0}$:

TOPICAL ID $T-0$ UP? DOWN?

deren Thema $T-0$ auch in der Ausgangsanfrage im Quellsystem vorhanden war. (1. Fall, Definition 5.8)

- Anfragen $t_{\text{output}-1}, \dots, t_{\text{output}-n}$:

TOPICAL ID $T-1$ UP? DOWN?
...
TOPICAL ID $T-n$ UP? DOWN?

Wobei gilt, dass die Themen $T-1$ bis $T-n$ Oberthemen zum Thema $T-0$ sind.

$$\{T-0\} \sqsubseteq \exists \text{super_topic_transitive}.\{T-1, \dots, T-n\}, \quad \text{2. Fall, Definition 5.8}$$

- Anfragen $t_{\text{output}-(n+1)}, \dots, t_{\text{output}-(n+m)}$:

TOPICAL ID $T-(n+1)$ UP? DOWN?
 ...
 TOPICAL ID $T-(n+m)$ UP? DOWN?

Wobei gilt, dass die Themen $T-(n+1)$ bis $(n+m)$ in einer *related_topic*- bzw. *sameAS*-Beziehung zu den Themen $T-0$ bis $T-n$ stehen.

$$\{T-(n+1), \dots, (n+m)\} \sqsubseteq \exists \textit{related_topic}.\{T-0, \dots, T-n\} \sqcup \exists \textit{sameAS}.\{T-0, \dots, T-n\},$$

3. Fall, Definition 5.8

Es gilt somit $t_{\text{input}} \approx t_{\text{output}}$.

Diese Art der Intentionserhaltung setzt voraus, dass es mindestens ein gemeinsames Oberthema im Themenpfad der Anfrage $t_{\text{intermediate}}$ in den Themengraphen des Quell- bzw. Zielsystems gibt. \square

Die Wahl der Gewichtung- und Zerfallsfaktoren und die initiale Aktivierung erlauben es, verschiedene Relaxationsstrategien für die thematischen Anfragen abzubilden.

Im Folgenden werden zwei Strategien genauer betrachtet.

5.5.2.1. Relaxation durch Verallgemeinerung

Die erste Strategie erweitert das Relaxationsverfahren aus Unterabschnitt 5.5.1 mithilfe einer auf Spreading-Activation basierenden Relaxation.

Die Grundidee, ausgehend vom Themenpfad der Anfrage aus dem Quellsystem möglichst das speziellste Thema auszuwählen, das im Themengraph des Zielsystems existiert, bleibt dabei bestehen.

Zusätzlich zu der Auswahl eines oder mehrerer Themen, wird für jedes Thema durch das Spreading-Activation-Verfahren ein Aktivierungswert berechnet. Für die Anwendung des Spreading-Activation-Verfahrens werden in diesem Fall ausschließlich die Kanten, die sich aus den *super_topic*-Beziehungen des Themengraphen ableiten, betrachtet.

Gewichtungs- und Zerfallsfaktoren werden anschließend so gewählt, dass sich die Aktivierung nur entlang des Pfades vom speziellsten Thema zu Oberthemen ausbreiten kann.

Die initiale Aktivierung wird so gewählt, dass das speziellste Thema im Pfad der thematischen Anfrage mit dem höchsten Aktivierungswert initialisiert wird. Alle weiteren Themen werden dann mit absteigenden Aktivierungswerten entlang des Pfades zum Oberthema initialisiert.

In Definition 5.28 wird eine Kombination von initialer Aktivierung, Gewichtungs- und Zerfallsfaktoren definiert, die genau die beschriebene Aktivierungsausbreitung und damit die beschriebene Relaxation realisieren.

Definition 5.28 [Initiale Aktivierung, Gewichtungsfaktoren]

Ausgangspunkt für die Parameter ist folgende generische elementare thematische Anfrage:

TOPICAL ID T-1 / T-2 / ... / T-n UP? DOWN?

Basierend darauf werden folgende Parameter gesetzt:

Initiale Aktivierung A

$$A = \{a_{T-1}, a_{T-2}, \dots, a_{T-n}\}; \quad a_{T-1} = 1, a_{T-2} = 2, \dots, a_{T-n} = n$$

Gewichtungsfaktoren W

$$W = \{w_{super_topic}^{forward}, w_{super_topic}^{backward}\};$$

$$w_{super_topic}^{forward} = 0,25 \text{ (UP-Option)}, w_{super_topic}^{backward} = 0,95 \text{ (DOWN-Option)}$$

Die Gewichtungsfaktoren sind abhängig von den gesetzten Optionen der Anfrage.

Zerfallsfaktoren D

$$D = \{d_{super_topic}^{forward}, d_{super_topic}^{backward}\}; \quad d_{super_topic}^{forward} = d_{super_topic}^{backward} = 1$$

Da für die „Relaxation durch Verallgemeinerung“ keine weiteren Kanten in Betracht gezogen werden, werden alle anderen Werte auf 0 gesetzt. □

Ausgehend von der Anfragetransformation im Quellsystem, die schon in Beispiel 5.22 eingeführt wurde, zeigt Beispiel 5.29 die Relaxation durch Verallgemeinerung unter Anwendung des Spreading-Activation-Verfahrens.

Beispiel 5.29 [Thematische Anfragetransformation im Zielsystem]

Betrachten wir erneut den Ausschnitt des Themengraphen von St. Sebald (siehe Abbildung 5.8), mit dem Unterschied, dass jetzt den Themen zusätzlich ein Aktivierungswert zugewiesen wurde.

Im Gegensatz zum Themengraphen von St. Lorenz existiert auch hier das Thema **Georg** nicht. Zusätzlich existiert das Thema **Josef**, das ebenfalls betrachtet wird.

Ausgehend von einer thematischen Pfad-Anfrage τ_{input} aus dem Beispiel 5.22 wird diese in eine Anfrage τ_{output} , die eine ganze Serie von Themen enthält, transformiert. Da die DOWN-Option in der Anfrage gesetzt ist, wird in diesem Beispiel $w_{super_topic}^{backward} = 0,95$ verwendet.

- τ_{input} :

TOPICAL ID MA:Ikonografie / MA:Heiliger / MA:Georg DOWN

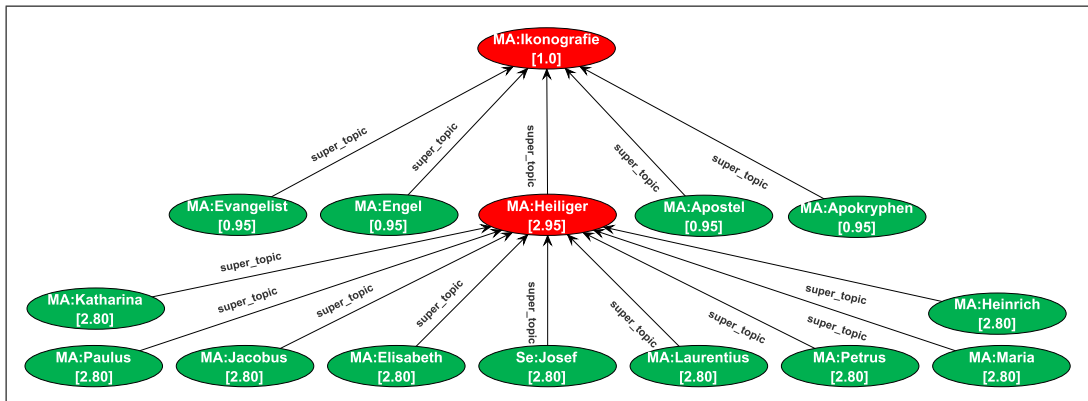


Abbildung 5.8.: Themengraph von St. Sebald mit Aktivierungswerten

- t_{output} :

```
TOPICAL ID MA:Ikografie DOWN
UNION
TOPICAL ID MA:Heiliger DOWN
```

Obwohl das Thema **Georg** nicht im Themengraphen von St. Sebald vorkommt, kann auch hier eine Relaxation der Anfrage durchgeführt werden. Die Aktivierungswerte der Themen bieten hierbei zusätzlich die Möglichkeit, nach Relevanz zu sortieren und erlauben dadurch auch eine Sortierung der Ergebnisse einer Anfrage.

Dies bedeutet, dass ein Dokument, das mit dem Thema **Katharina** ausgezeichnet ist höher bewertet wird als ein Dokument, das mit dem Thema **Engel** ausgezeichnet ist. \square

Der Vorteil dieser Art der Anfragerelaxation besteht in der Möglichkeit, für die Ergebnisse aus den Aktivierungswerten des Spreading-Activation-Verfahrens ein Ranking abzuleiten. Dabei bleibt die Anfragetransformation intentionserhaltend.

5.5.2.2. Relaxation durch Verwandtschaft

Eine weitere Relaxationsstrategie nutzt nicht nur die *super_topic*-Beziehung für die Berechnung eines Rankings, sondern zusätzlich werden die *sameAS*- und *related_topic*-Beziehungen in das Spreading-Activation miteinbezogen.

Dies erlaubt neben der Relaxation durch Verallgemeinerung auch die Relaxation der Anfrage über Verwandtschaftsbeziehungen von Themen. Der Ablauf ist dabei identisch zur vorherigen Strategie, es ändern sich nur die Gewichtungs- und Zerfallsfaktoren.

Die initiale Aktivierung wird wie vorher gewählt, d.h. das speziellste Thema hat den höchsten Aktivierungswert. Für die *super_topic*-Beziehung ändert sich die Aktivierungsausbreitung nicht und ist weiterhin von der UP- bzw. DOWN-Option abhängig.

Die *sameAS*- und *related_topic*-Beziehungen werden symmetrisch betrachtet, wobei die *sameAS*-Beziehung eine stärkere Ausbreitung der Aktivierung zulässt.

Die Idee hinter dieser Strategie der Relaxation ist eine Ausbreitung von Aktivierungen von Themenhierarchie zu Themenhierarchie, die miteinander in Beziehung gesetzt wurden, zu ermöglichen. Damit lassen sich externe Vokabulare, wie z.B. die GND¹, in die Relaxation miteinbeziehen.

In Definition 5.30 sind die initiale Aktivierung, Gewichtung- und Zerfallsfaktoren definiert, die für diese Strategie die beschriebene Aktivierungsausbreitung und damit eine weitere Anfragerelaxation realisieren.

Definition 5.30 [Initiale Aktivierung, Gewichtungs- und Zerfallsfaktoren]

Ausgangspunkt für die Parameter ist folgende generische elementare thematische Anfrage:

TOPICAL ID T-1 / T-2 / ... / T-n UP? DOWN?

Basierend darauf werden folgende Parameter gesetzt:

Initiale Aktivierung A

$$A = \{a_{T-1}, a_{T-2}, \dots, a_{T-n}\}; \quad a_{T-1} = 1, a_{T-2} = 2, \dots, a_{T-n} = n$$

Gewichtungsfaktoren W

$$W = \{w_{super_topic}^{forward}, w_{super_topic}^{backward}, w_{sameAS}^{forward}, w_{sameAS}^{backward}, w_{related_topic}^{forward}, w_{related_topic}^{backward}\};$$

$$w_{super_topic}^{forward} = 0,25 \text{ (UP-Option)}, w_{super_topic}^{backward} = 0,95 \text{ (DOWN-Option)},$$

$$w_{sameAS}^{forward} = w_{sameAS}^{backward} = 1, w_{related_topic}^{forward} = w_{related_topic}^{backward} = 0,75$$

Die Gewichtungsfaktoren sind abhängig von den gesetzten Optionen der Anfrage.

Zerfallsfaktoren D

$$D = \{d_{super_topic}^{forward}, d_{super_topic}^{backward}, d_{sameAS}^{forward}, d_{sameAS}^{backward}, d_{related_topic}^{forward}, d_{related_topic}^{backward}\};$$

$$d_{super_topic}^{forward} = d_{super_topic}^{backward} = d_{sameAS}^{forward} = d_{sameAS}^{backward} = d_{related_topic}^{forward} = d_{related_topic}^{backward} = 1$$

□

¹http://www.dnb.de/DE/Standardisierung/GND/gnd_node.html

Zur Illustration dieser Strategie werden im Folgenden zwei Beispiele gezeigt.

Beide Beispiele nutzen zusätzlich zu einem gemeinsamen Themengraphen, der sowohl im Quell- als auch im Zielsystem vorhanden ist, ein externes Vokabular, das nur im Zielsystem vorhanden ist.

Hierbei ermöglicht die erweiterte Strategie auch eine Relaxation der Anfrage unter Einbeziehung externer Vokabulare.

Beispiel 5.31 [Thematische Anfragetransformation - Verfeinerung im Zielsystem]

Ausgangspunkt für dieses Beispiel ist ein Ausschnitt aus der **Material**-Themenhierarchie aus dem MonArch-Projekt und ein Ausschnitt aus der GND über **Sandstein** und dessen Unterthemen (siehe Abbildung 5.9).

In der Grafik ist in grün mit weißer Schrift der Ausschnitt aus dem Themengraphen, der sowohl im Quell- als auch im Zielsystem vorhanden ist, zu sehen. Die Themen in Gelb mit schwarzer Schrift stammen aus der GND.

Die Anfrage im Quellsystem t_{source} ist eine thematische elementare Anfrage nach dem Thema **Sandstein**.

```
TOPICAL ID MA:Sandstein DOWN
```

Ausgehend von dieser Anfrage wird eine thematische Pfad-Anfrage $t_{\text{intermediate}}$ im Quellsystem an das Zielsystem übermittelt.

```
TOPICAL ID MA:Material / MA:Werkstoff / MA:Stein / MA:Naturstein / MA:Sandstein
DOWN
```

Im Zielsystem wird die Anfrage $t_{\text{intermediate}}$ in die Anfrage t_{target} relaxiert.

Dabei wird der gemeinsame Themengraph betrachtet. Zusätzlich wird auch die Themenhierarchie über Sandsteine aus der GND in die Relaxation miteinbezogen.

In diesem Beispiel gehen wir davon aus, dass die Themenhierarchie aus der GND nur im Zielsystem vorhanden ist, da hier z.B. eine feinere Unterscheidung der Sandsteine notwendig ist und diese aus der GND übernommen wurde.

Die relaxierte Anfrage t_{target} betrachtet also sowohl den Pfad als auch die Verfeinerung in die GND.

```
TOPICAL ID MA:Material DOWN
UNION
TOPICAL ID MA:Werkstoff DOWN
UNION
TOPICAL ID MA:Stein DOWN
UNION
TOPICAL ID MA:Naturstein DOWN
UNION
TOPICAL ID MA:Sandstein DOWN
UNION
TOPICAL ID GND:Sandstein DOWN
```

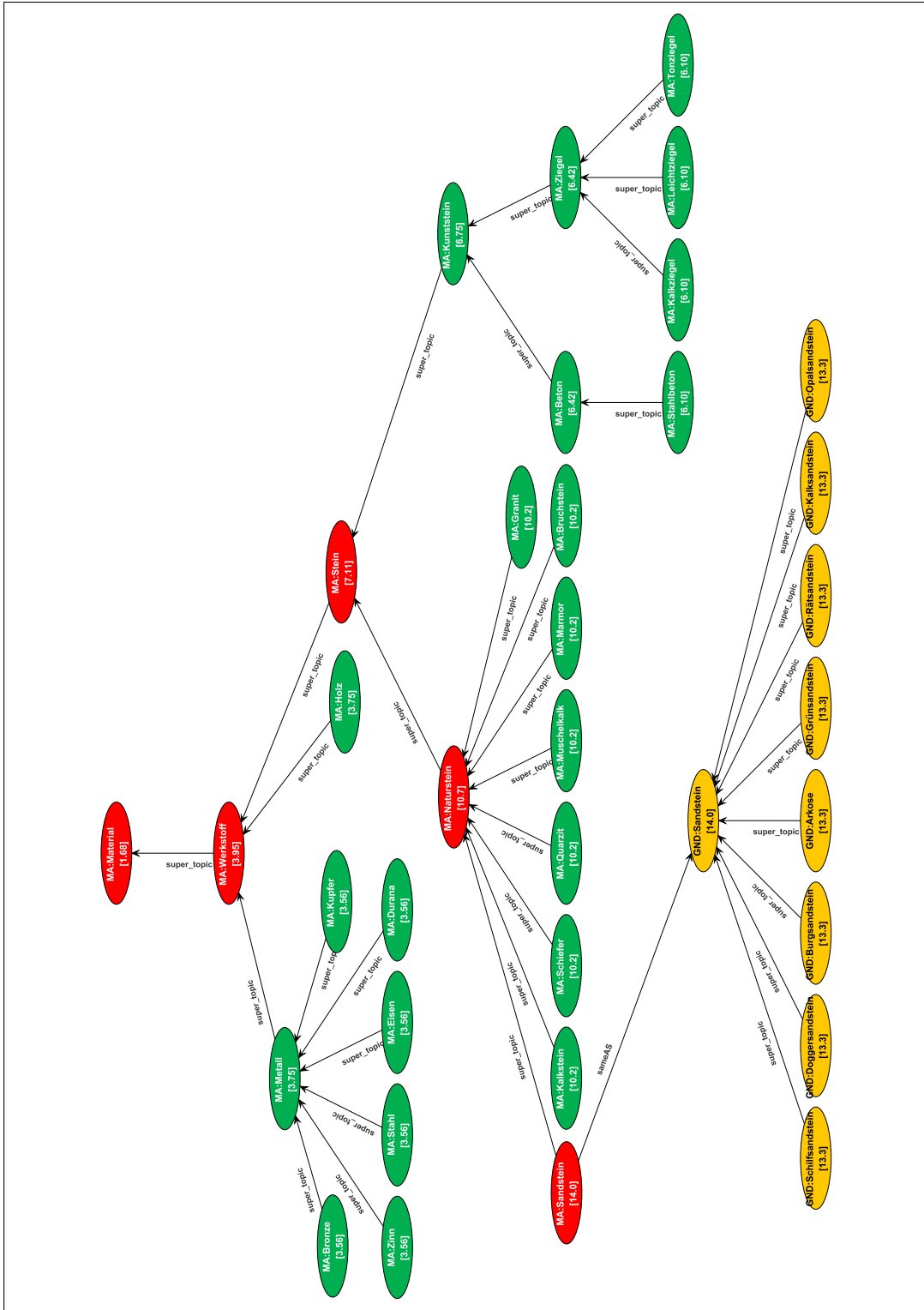


Abbildung 5.9.: Material-Themengraph mit Aktivierungswerten - Anfrage Sandstein

Die Kombination zwischen Anfragetransformation und Berechnung eines Rankings über das Spreading-Activation-Verfahren ist in diesem Beispiel an den Aktivierungswerten der Themen zu beobachten. Einerseits wird **Sandstein**, wie auch in der Quellenfrage, weiterhin im Zielsystem betrachtet. Andererseits kann die Erweiterung des Sandsteins im Zielsystem über die GND mit in die Relaxation einbezogen werden.

Als Konsequenz können z.B. Dokumente im Zielsystem nur mit den Themen aus der GND ausgezeichnet werden und trotzdem würden diese Dokumente höher im Ranking liegen als z.B. Dokumente, die mit **Naturstein** ausgezeichnet worden sind.

Die Sandsteinarten haben einen Aktivierungswert zwischen 13 und 14. Der Aktivierungswert von anderen Gesteinsarten liegt bei 10. Weitere Materialien liegen in den Aktivierungswerten noch niedriger.

Hier wird die Intention der Quellenfrage, nach Informationen über **Sandstein** zu suchen, in das Zielsystem übertragen.

Das Quellsystem hat dabei keinerlei Information über den Themengraph im Zielsystem. Voraussetzung ist auch hier die Einigung über eine gemeinsame Teilthemenhierarchie, wie in Definition 5.16 gefordert.

Dass dieses Miteinbeziehen der Sandsteinarten aus der GND auch noch funktioniert, falls die Anfrage im Quellsystem allgemeiner nach **Naturstein** selektiert, zeigt der Ausschnitt in Abbildung 5.10.

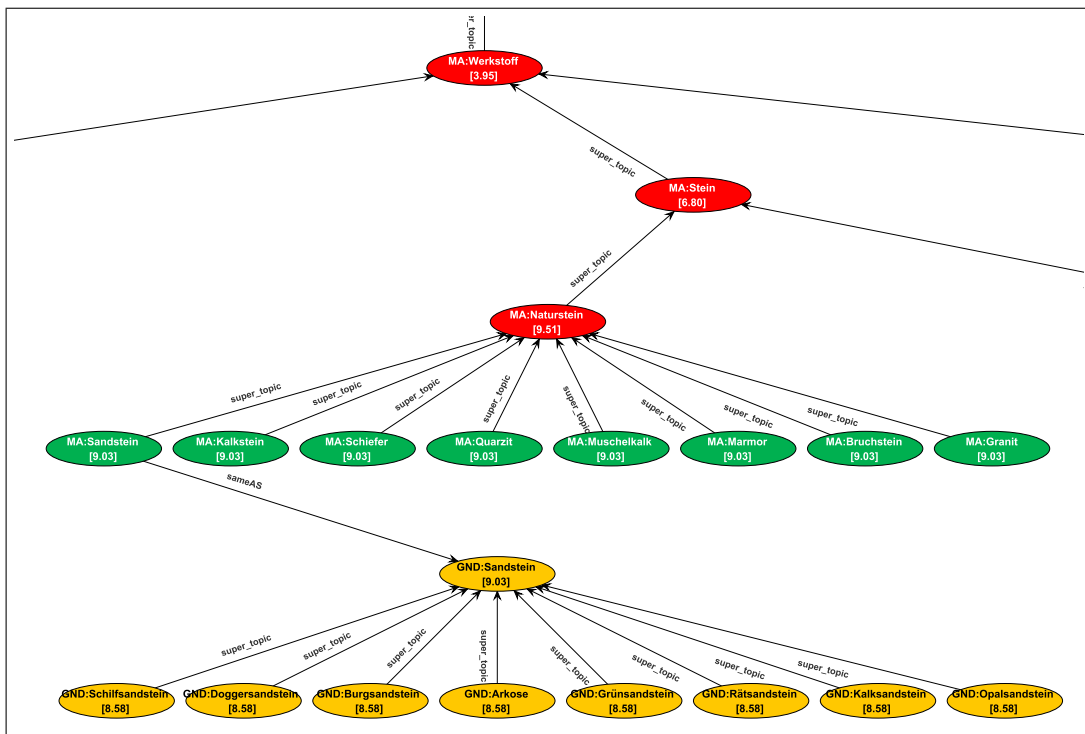


Abbildung 5.10.: Material-Themengraph mit Aktivierungswerten - Anfrage **Naturstein**

5. Anfragerelaxation und -transformation

Durch die Ausbreitung der Aktivierungswerte über die *sameAS*-Beziehung, werden die Sandsteinarten der GND miteinbezogen. Als Konsequenz fallen hier die Aktivierungswerte niedriger aus als bei der direkten Anfrage nach *Sandstein*. Andere Natursteine haben den gleichen Aktivierungswert wie *Sandstein* \square

Beispiel 5.32 [Thematische Anfragetransformation - Übergeordnetes Vokabular im Zielsystem]

Bei diesem Beispiel dreht sich die Beziehung zwischen externem und gemeinsamem Vokabular um. In Abbildung 5.11 ist ein Ausschnitt aus der Themenhierarchie des ICONCLASS-Vokabulars² zu sehen (schwarze Schrift auf gelben Knoten).

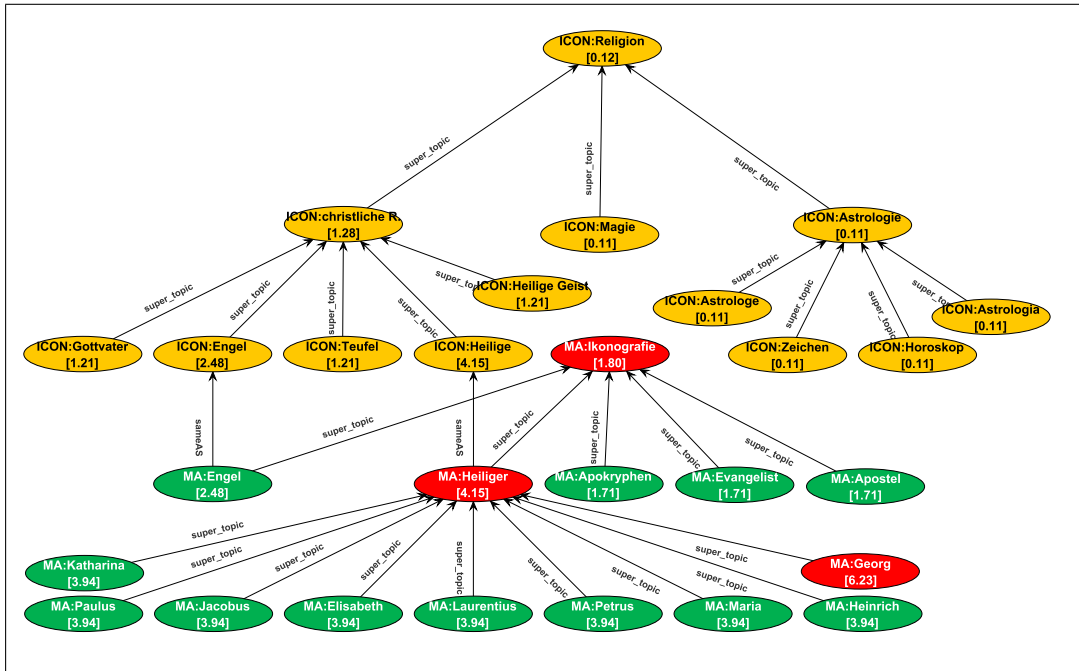


Abbildung 5.11.: ICONCLASS-Themengraph mit Aktivierungswerten - Anfrage *Georg*

In diesem Fall dient das externe Vokabular als übergeordnete Themenhierarchie für die gemeinsame Themenhierarchie (weiße Schrift auf grünen Knoten) im Zielsystem. Ausgangspunkt im Quellsystem ist die gemeinsame Themenhierarchie.

Die Anfrage t_{source} selektiert nach dem Heiligen *Georg* in Kombination mit einer UP- und DOWN-Option.

TOPICAL ID MA:Georg **UP DOWN**

Übertragen an das Zielsystem wird wieder ein Pfad-Anfrage $t_{\text{intermediate}}$.

TOPICAL ID MA:Ikografie / MA:Heiliger / MA:Georg **UP DOWN**

²<http://www.iconclass.nl/home>

Das Ergebnis der Anfragetransformation ist die Anfrage τ_{target} .

```

TOPICAL ID MA:Georg UP DOWN
UNION
TOPICAL ID MA:Heiliger UP DOWN
UNION
TOPICAL ID MA:Ikonografie UP DOWN
UNION
TOPICAL ID ICON:Heilige UP DOWN

```

In diesem Fall ermöglicht die Ausbreitung der Aktivierungen das Miteinbeziehen der ICONCLASS-Themenhierarchie im Zielsystem. In diesem Beispiel gibt es keine direkte Verbindung zwischen dem Thema **Georg** und den Themen aus ICONCLASS. Über das Oberthema **Heiliger** kann jedoch eine Beziehung zwischen der gemeinsamen Themenhierarchie und der ICONCLASS-Themenhierarchie hergestellt werden.

Die relaxierte Anfrage τ_{target} bildet die erste Auswahl. Mithilfe der Aktivierungswerte der Themen kann ein Ranking für die Elemente im Ergebnis der Anfrage abgeleitet werden.

Das Thema **Engel** wird hier höher aktiviert, da neben der Beziehung zwischen **Heiliger** und **Heilige** eine zweite Verknüpfung zwischen den Themenhierarchien existiert, die bei der gegebenen Anfrage Einfluss auf die Aktivierungswerte nimmt. \square

5.6. Transformation räumlicher Anfragen

Die Anfragetransformation von elementaren räumlichen Anfragen ist, wie auch die allgemeine Anfragetransformation, in eine Anfragetransformation im Quellsystem und eine Anfragetransformation im Zielsystem aufgeteilt.

Algorithmus 5.7 zeigt den Ablauf im Quellsystem.

Die Eingabe des Algorithmus ist eine elementare räumliche Anfrage, auf deren Basis die Anfragetransformation im Quellsystem durchgeführt wird.

Die Grundidee des Algorithmus ist ähnlich zu der Transformation von thematischen Anfragen. Ausgehend vom selektierten Gebäudeteil werden alle Gebäudeteile auf dem Pfad entlang der *part_of*-Beziehung betrachtet.

Im Gegensatz zur thematischen Anfragetransformation wird nicht ein Pfad-Anfrage von Gebäudeteilen, sondern eine typbasierte Anfrage basierend auf dem Pfad der Gebäudeteile erstellt.

Dies hat den Hintergrund, dass Gebäudeteile sich im Allgemeinen zwischen Quell- und Zielsystem unterscheiden und die Strukturtypen eine gemeinsame Basis bilden.

Der Algorithmus durchläuft dabei folgende Schritte.

1. Setzen der ID eines Strukturtypen vom selektierten Gebäudeteil in der Eingabeanfrage s_{input} (Zeile 2-4).

```

1 Function transformSpatialQuerySource (sinput)
   Input : sinput = SPATIAL ID SE-id UP? DOWN?
   Output : soutput = TYPE SPATIAL ID Path UP? DOWN?
   /* Setze aktuelles Gebäudeteil */
2   currentSpatialEntity := Spatial_Entity with ID SE-id;
   /* Setze ID des Typs des Gebäudeteils der Eingabeanfrage in den
      Ausgabepfad */
3   SET-id := getID (TypeOf (currentSpatialEntity) );
4   Path := SET-id;
   /* Bestimme nächstes Gebäudeteil */
5   nextSpatialEntity := Spatial_Entity which currentSpatialEntity is part_of;
   /* Bestimme Pfad von Typen */
6   while currentSpatialEntity has nextSpatialEntity do
7     currentSpatialEntity := nextSpatialEntity;
8     SET-id := getID (TypeOf (currentSpatialEntity) );
       /* Füge ID des aktuellen Typs an den Pfad an */
9     Path := SET-id + '/' + Path;
10    nextSpatialEntity := Spatial_Entity which currentSpatialEntity is part_of;
   Result : TYPE SPATIAL ID Path UP? DOWN?

```

Algorithmus 5.7 : Räumliche Anfragetransformation im Quellsystem

2. Bestimmen des übergeordneten Gebäudeteils entlang der *part_of*-Beziehung (Zeile 5).
3. Anfügen der ID des Strukturtyps vom aktuellen Gebäudeteil an den Anfang des Pfades (Zeile 7-9).
4. Wiederholen der Bestimmung übergeordnetes Gebäudeteil bis das Top-Gebäudeteil in der Strukturhierarchie erreicht ist und Anfügen der IDs der Strukturtypen an den Pfad (Zeile 6,10).
5. Rückgabe einer elementaren räumlichen typbasierten Anfrage mit einem ID-Pfad.

Tritt der Sonderfall ein, dass keine *part_of*-Beziehung existiert aber eine *consists_of*-Beziehung, so muss der Algorithmus bei der Bestimmung des nächsten Gebäudeteils angepasst werden.

Da die *consists_of*-Beziehung nicht funktional ist, entsteht hierbei für jeden Pfad eine eigene räumliche typbasierte Pfad-Anfrage. Dies würde den Algorithmus umfangreicher machen, die Grundidee und den generellen Ablauf jedoch nicht betreffen. Deshalb beschränkt sich der Algorithmus auf die *part_of*-Beziehung. Da davon ausgegangen wird, dass der Strukturgraph zyklenfrei ist, müssen keine weiteren Maßnahmen ergriffen werden.

Die Transformation der räumlichen Anfrage im Quellsystem greift dabei ebenfalls auf keine Informationen im Zielsystem zu. Die Transformation kann ausschließlich unabhängig durchgeführt werden.

Beispiel 5.33 zeigt eine elementare räumliche Anfragetransformation.

Beispiel 5.33 [räumliche Anfragetransformation im Quellsystem]

Betrachten wir einen Ausschnitt des Strukturgraphen von St. Lorenz (siehe Abbildung 5.12).

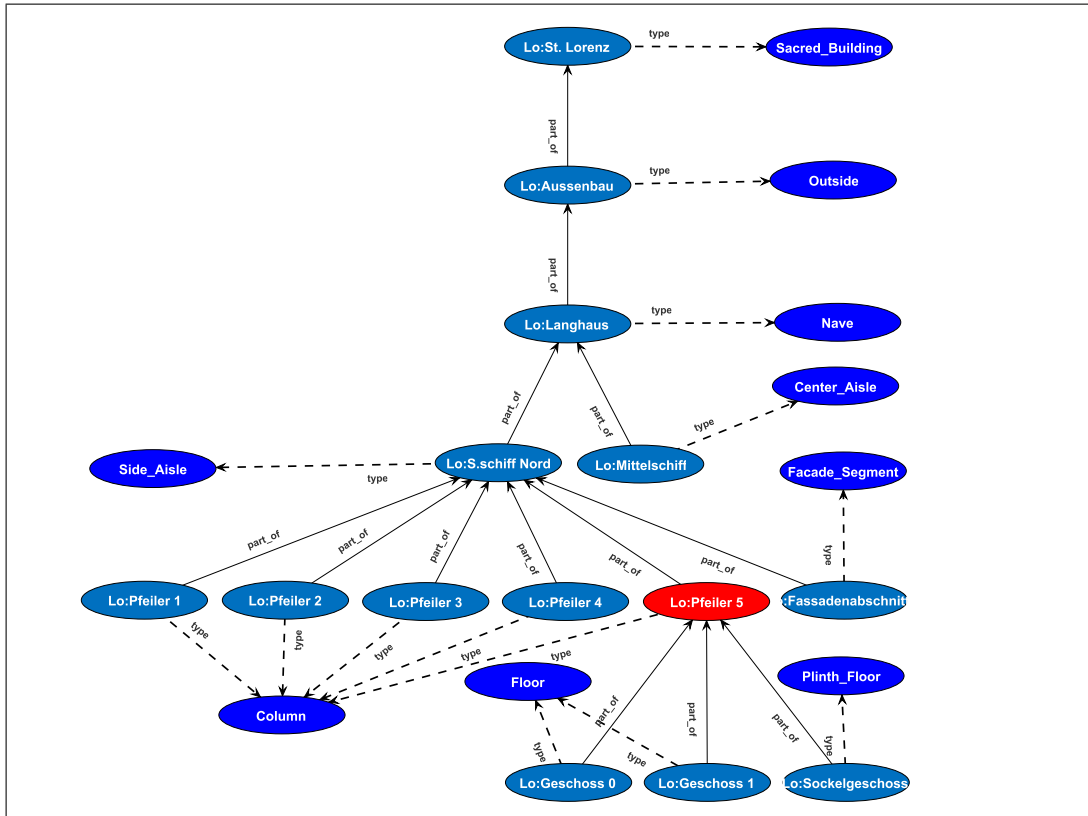


Abbildung 5.12.: Strukturgraph von St. Lorenz mit Strukturtypen

Ausgehend von einer elementaren räumlichen Anfrage s_{input} nach einem Pfeiler, in diesem Beispiel *Pfeiler 5* im *Seitenschiff Nord*, wird diese in eine typbasierte Anfrage s_{output} mit einem Typpfad transformiert.

- s_{input} :

```
SPATIAL ID Lo:Pfeiler 5(S Nord)
```

- s_{output} :

```
TYPE SPATIAL ID Sacred_Building / Outside / Nave / Side_Aisle / Column
```

Der Typpfad wird ausgehend vom selektierten Pfeiler zum Top-Element erstellt. Die Strukturtypen im Quellsystem sind identisch zu den Strukturtypen im Zielsystem. \square

Die Transformation im Quellsystem ist auch hier wieder eine vorbereitende Transformation für die Transformation im Zielsystem. Beide Transformationen zusammen ergeben letztendlich die räumliche Anfragetransformation.

Es sei hier erneut darauf hingewiesen, dass das Zielsystem nicht zwangsweise ein anderes System sein muss als das Quellsystem.

Für die Anfragetransformation im Zielsystem von räumlichen Anfragen werden in den folgenden Unterabschnitten zwei Strategien betrachtet und zusammen mit der Transformation im Quellsystem auf ihre intentionserhaltende Eigenschaft untersucht.

5.6.1. Transformation durch Typanfragen

Eine naheliegende Transformation im Zielsystem ergibt sich aus den Überlegungen, die in Unterabschnitt 5.5.1 zur thematischen Anfragetransformation durch Verallgemeinerung vorgestellt wurden.

Verallgemeinerung im Kontext räumlicher Anfragen bedeutet, basierend auf dem Strukturtypen eine Anfrage durchzuführen. Die Anfrage nach einem bestimmten Pfeiler wird in eine Anfrage nach allen Pfeilern transformiert.

In Algorithmus 5.8 ist die Transformation der Anfrage im Zielsystem beschrieben.

```
1 Function transformSpatialQueryTarget (sinput)
   Input : sinput = TYPE SPATIAL ID Path UP? DOWN?
   Output : soutput
   /* Initialisiere Menge für Gebäudeteile */
2   SE := ∅;
3   repeat
4     /* Nehme letztes Element im Pfad und setze ID */
     SET-id := Last (Path);
5     currentType := Spatial_Entity_Type with SET-id;
     /* Bestimme Gebäudeteile, die vom aktuellen Strukturtypen
       sind */
6     SE := SE ∪ { SE-id | SE-id hasType (currentType) };
     /* Lösche letzte ID im Pfad */
7     DeleteLast (Path);
8   until SE ≠ ∅ ∨ Path is empty;
   /* Erzeuge räumliche Anfrage */
9   foreach SE-id ∈ SE do
10    soutput := soutput + UNION SPATIAL ID SE-id UP? DOWN?;
   Result : soutput
```

Algorithmus 5.8 : Räumliche Anfragetransformation im Zielsystem

Die Eingabe des Algorithmus ist eine typbasierte Anfrage, die durch die Transformation der Ursprungsanfrage mit Algorithmus 5.7 entsteht. Die transformierte Anfrage ist eine räumliche Anfrage von elementaren räumlichen Anfragen nach Gebäudeteilen.

Die Grundidee des Algorithmus besteht darin, dass möglichst Gebäudeteile ausgewählt werden, die denselben Strukturtypen haben wie das Gebäudeteil im Quellsystem.

Der Algorithmus durchläuft dabei folgende Schritte.

1. Aus der Eingabeanfrage s_{input} wird die ID eines Strukturtypen dem Pfad entnommen. Dabei wird die letzte ID im Pfad betrachtet (Zeile 4).
2. Es erfolgt die Bestimmung aller Gebäudeteile, die vom gewählten Strukturtyp sind (Zeile 6).
3. Existieren Gebäudeteile, so werden diese als räumliche Anfragen zurückgegeben (Zeile 9-10).
4. Falls kein Gebäudeteil im Strukturgraphen existiert, wird die nächste ID im Pfad überprüft (Zeile 8).

Die Transformation kann wieder unabhängig durchgeführt werden. Außer der Anfrage vom Quellsystem sind keine weiteren Informationen vom Quellsystem notwendig.

Dieses Verfahren setzt voraus,

- dass sich Quell- und Zielsystem eine Typhierarchie teilen,
- dass jedes Gebäudeteil nur einen Strukturtyp besitzt, d.h. die *type*-Beziehung funktional ist.

Beispiel 5.34 [Räumliche Anfragetransformation im Zielsystem]

Betrachten wir einen Ausschnitt des Strukturgraphen von St. Sebald (siehe Abbildung 5.13). Im Gegensatz zum Themengraphen von St. Lorenz, wurden teilweise die Pfeiler römisch nummeriert.

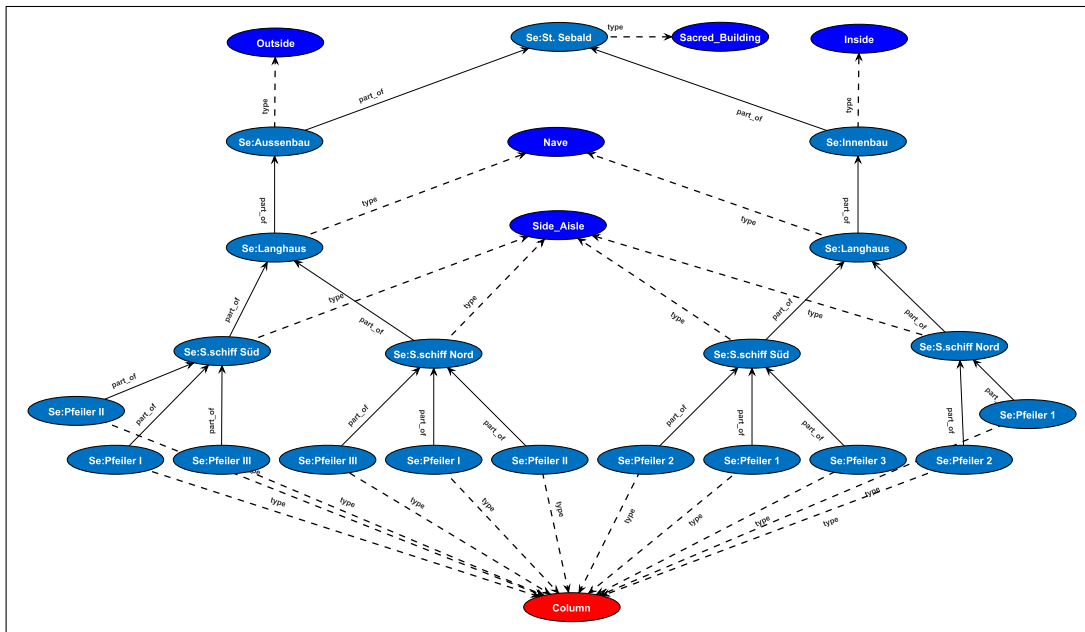


Abbildung 5.13.: Strukturgraph von St. Sebald mit Strukturtypen

5. Anfragerelaxation und -transformation

Ausgehend von einer räumlichen typbasierten Pfad-Anfrage s_{input} aus dem Beispiel 5.33, wird diese in eine Anfrage s_{output} , die alle Pfeiler unabhängig von der Nummerierung und ob sie Teil des **Außenbaus** oder **Innenbaus** sind, betrachtet.

- s_{input} :

```
TYPE SPATIAL ID Sacred_Building / Outside / Nave / Side_Aisle / Column
```

- s_{output} :

```
SPATIAL ID Se:Pfeiler I
UNION
SPATIAL ID Se:Pfeiler II
UNION
SPATIAL ID Se:Pfeiler I
UNION
...
```

Obwohl der **Lo:Pfeiler 5** nicht im Strukturgraphen von St. Sebald vorkommt, kann eine Relaxation der Anfrage durchgeführt werden. Die Relaxation bewirkt, dass Pfeiler sowohl im **Außenbau** als auch im **Innenbau** im Suchraum mit aufgenommen werden. Ausgangspunkt der Anfrage ist der Strukturtyp **Column**.

Auch die Benennung der Pfeiler, z.B. Art der Nummerierung oder Sprache, muss bei der Auswahl nicht betrachtet werden. Würde man versuchen einen speziellen Pfeiler in St. Lorenz **Lo:Pfeiler 5** auf einen Pfeiler mit gleicher Benennung in St. Sebald **Se:Pfeiler 5** abzubilden, so würde der Suchraum zu stark eingeschränkt und somit weitere Pfeiler nicht betrachtet werden.

□

Für diese Variante der räumlichen Anfragetransformation muss noch die Intentionserhaltung betrachtet werden.

Überprüfung 5.35 [Intentionserhaltende räumliche Anfragetransformation]

Die Intentionserhaltung der räumlichen Anfragetransformation wird für die möglichen Ausgabeanfragen bei einer gegebenen Eingabeanfrage betrachtet. Wie bereits bei der Betrachtung der Strukturerhaltung, wird die Anfragetransformation im Quell- und Zielsystem in einer Einheit betrachtet.

Gehen wir dazu von einer generischen Form einer elementaren räumlichen Unteranfrage als Eingabeanfrage s_{input} im Quellsystem aus:

```
SPATIAL ID SE-0 UP? DOWN?
```

Nach der Transformation der Anfrage im Quellsystem ist das Ergebnis eine räumliche typbasierte Pfad-Anfrage $s_{intermediate}$:

```
TYPE SPATIAL ID SET-n/.../SET-0 UP? DOWN?
```

Im Zielsystem wird diese in eine räumliche Anfrage s_{output} , bestehend aus einzelnen elementaren räumlichen Anfragen, transformiert:

```

SPATIAL ID SE-1 UP? DOWN?
UNION
SPATIAL ID SE-2 UP? DOWN?
UNION
...
SPATIAL ID SE-m UP? DOWN?
    
```

Welche Gebäudeteile als Ergebnis der Transformation ausgewählt werden, hängt davon ab, welche Strukturtypen im Strukturgraphen des Zielsystems vorhanden sind.

Falls im Strukturgraphen des Zielsystems Gebäudeteile vom Strukturtyp $SET-0$ existieren, sind die Gebäudeteile $SE-1$ bis $SE-m$ von diesem Typ und die Intentionserhaltung ist daher nach dem 2. Fall der Definition 5.7 gewährleistet.

Sind die Gebäudeteile vom Strukturtyp $SET-1$, $SET-2$ oder $SET-n$, ist die Intentionserhaltung nach dem 3. Fall der Definition 5.7 sichergestellt, da jedes Gebäudeteil $SE-1$ bis $SE-m$ genau einen Strukturtyp $SET-1$, $SET-2$ oder $SET-n$ besitzt:

$$\begin{aligned}
 \exists type^- . \{SE-1, SE-2, \dots, SE-m\} &\sqsubseteq \exists type^- . (\exists has_part_transitive . SE-0 \\
 &\quad \sqcup \exists consists_of^+ . SE-0) \\
 &= \{SET-0, SET-1, \dots, SET-n\}
 \end{aligned}$$

Es gilt somit $s_{\text{input}} : \approx s_{\text{output}}$.

In beiden Fällen ist die Transformation eine Relaxation, da der Suchraum von einem Gebäudeteil auf eine Menge von Gebäudeteilen erweitert wird.

Diese Art der Intentionserhaltung setzt voraus, dass es mindestens einen gemeinsamen Strukturtypen in der Anfrage $s_{\text{intermediate}}$ in den Strukturgraphen des Quell- bzw. Zielsystems gibt. \square

Diese Strategie der Anfragerelaxation von räumlichen Anfragen ist eine Möglichkeit eine Verbundanfrage zu realisieren, die intentionserhaltend ist. Sie bezieht die Strukturhierarchie aber nur teilweise mit ein, was dazu führt, dass Gebäudeteile nur anhand ihres Strukturtyps, nicht aber anhand ihrer Position in der Strukturhierarchie ausgewählt werden.

Die Pfeiler werden unabhängig davon ob sie im *Außenbau* oder *Innenbau* vorkommen in die Zielanfrage übernommen.

5.6.2. Spreading-Activation für die räumliche Anfragerelaxation

Da die Grundidee der räumlichen Anfragetransformation der thematischen Anfragetransformation entspricht, liegt es nahe das Spreading-Activation-Verfahren in die räumliche Anfragetransformation miteinzubeziehen.

Erste Ansätze um die Anfragerelaxation durch Spreading-Activation zu verbessern wurden vom Autor dieser Arbeit in [SF12] vorgestellt. Die Autoren in [Sun+17] verfolgen einen ähnlichen Ansatz um die Ergebnisse an räumlichen Daten mithilfe von Spreading-Activation zu verbessern.

Der Nachteil der Anfragerelaxation, die in Unterabschnitt 5.6.1 beschrieben ist, besteht darin, dass Strukturtypen isoliert betrachtet werden. Um diesem Nachteil entgegenzuwirken, ist es notwendig die Strukturtypen in Bezug auf die Strukturhierarchie zu betrachten.

Nutzt man dazu die Strukturtypen als initiale Aktivierung im Strukturgraphen, können die Strukturtypen in Bezug auf die *part_of*- bzw. *consists_of*-Beziehung betrachtet werden.

Welche Strukturtypen betrachtet werden, lässt sich aus der Anfrage des Quellsystems ableiten.

Als Vorbereitung zur Anwendung des Spreading-Activation für die räumliche Anfragerelaxation, muss der Strukturgraph als Spreading-Activation-Graph (siehe Abbildungsdefinition 5.36) abgebildet werden.

Abbildungsdefinition 5.36 [Räumliche Ontologie als Spreading-Activation-Graph]

Ausgangspunkt für die Abbildung der Typ- und Raumontologie auf einen Spreading-Activation-Graphen $G^{spatial} = (V, E, A, W, D)$ sind die Ontologien $O_{schema}^{spatial_type}$ und $O_{schema}^{spatial}$ aus Unterabschnitt 3.4.1 mit:

$$V = Spatial_Entity_Type \sqcup Spatial_Entity$$

Individuen der Konzepte *Spatial_Entity_Type* und *Spatial_Entity* werden als Knoten im Spreading-Activation-Graph $G^{spatial}$ abgebildet.

$$E = part_of \sqcup consists_of \sqcup type$$

Rollenaussagen der Rollen *part_of*, *consists_of* und *type* werden als Kanten im Spreading-Activation-Graph $G^{spatial}$ abgebildet.

Die Mengen *A*, *W* und *D* werden später durch eine Konfiguration vor der Anwendung des Spreading-Activation-Algorithmus definiert und beeinflussen damit das Verfahren analog zur Anwendung des Spreading-Activation-Verfahrens auf den Themengraphen. \square

In Beispiel 5.37 ist ein Ausschnitt eines Strukturgraphen als Spreading-Activation-Graph mit Aktivierungswerten zu sehen.

Beispiel 5.37 [Spreading-Activation-Graph für St. Lorenz]

Abbildung 5.14 zeigt einen Ausschnitt des Strukturgraphen von St. Lorenz als Spreading-Activation-Graph.

Die Strukturtypen und Gebäudeteile sind als Knoten repräsentiert, die *part_of*- und *type*-Beziehungen sind als Kanten repräsentiert. Für die Knotenbeschriftung wurde zur Illustration der Name des Gebäudeteils verwendet. Dieser muss nicht eindeutig sein.

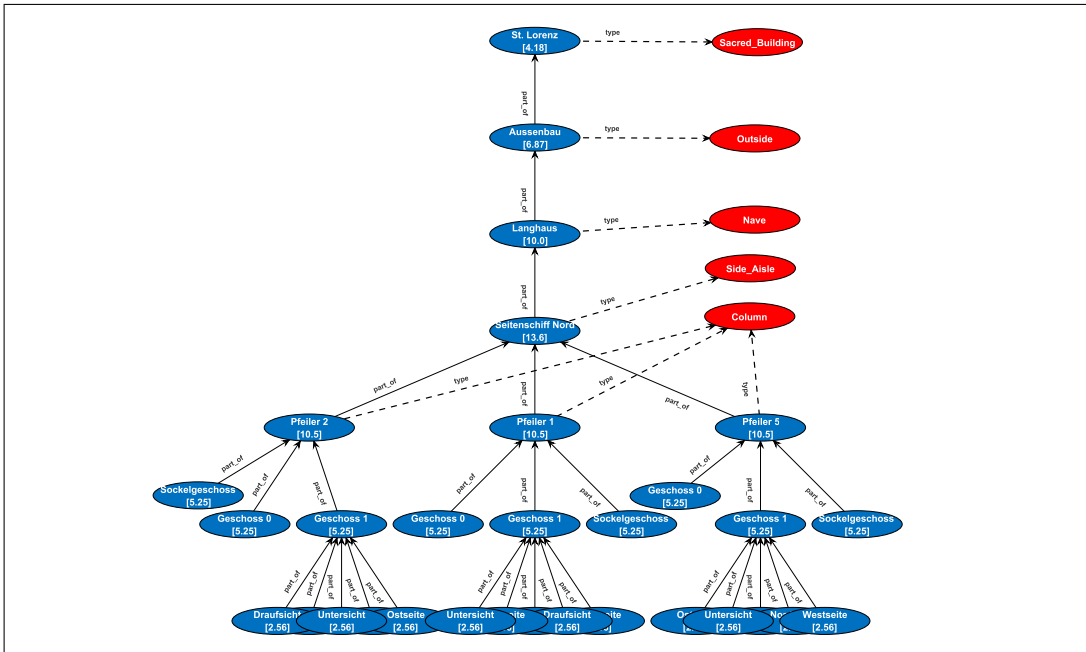


Abbildung 5.14.: Spreading-Activation-Graph für den Strukturgraphen von St. Lorenz

Der Wert im Knoten ist der berechnete Aktivierungswert. Je höher der Aktivierungswert, desto besser passt das Gebäudeteil zur Anfrage. Die roten Knoten zeigen die initial aktivierten Knoten. □

Algorithmus 5.9 erweitert die räumliche Anfragetransformation im Zielsystem durch das Spreading-Activation-Verfahren.

Die Grundidee des Algorithmus ist es, ausgehend von den Strukturtypen im Pfad durch den Einsatz des Spreading-Activation-Verfahrens, Gebäudeteile im Strukturgraphen mithilfe des Aktivierungswertes zu bewerten.

Da sich im Allgemeinen der Strukturgraph im Quell- und Zielsystem in Bezug auf die konkreten Gebäudeteile unterscheidet, kann nicht einfach ein Pfad an Gebäudeteilen als Eingabeanfrage s_{input} verwendet werden, wie es bei den Themen möglich war.

Die Alternative besteht darin, den Pfad an Gebäudeteilen zu nehmen und jedes Gebäudeteil im Pfad durch den Strukturtyp des Gebäudeteils zu ersetzen. Dieser Schritt muss im Quellsystem

erfolgen, da im Zielsystem dieses Ersetzen nicht mehr durchgeführt werden kann, da die Gebäudeteile im Zielsystem nicht existieren.

Die Eingabe des Algorithmus ist eine typbasierte Pfad-Anfrage, die durch die Transformation einer elementaren räumlichen Anfrage mit Algorithmus 5.7 entsteht.

```

1 Function transformSpatialQueryTarget (sinput)
  Input : sinput = TYPE SPATIAL ID Path UP? DOWN?
  Output : soutput
  /* Phase 1: Transformation der räumlichen typbasierten Anfrage
   */
  /* Initialisiere Menge für Gebäudeteile */
2  SE := ∅;
3  repeat
  | /* Nehme letztes Element im Pfad und setze ID */
  | SET-id := Last (Path);
  | currentType := Spatial_Entity_Type with SET-id;
  | /* Bestimme Gebäudeteile, die vom aktuellen Strukturtypen
  | sind */
  | SE := SE ∪ { SE-id | SE-id hasType (currentType) };
  | /* Lösche letzte ID im Pfad */
  | DeleteLast (Path);
4  until SE ≠ ∅ ∨ Path is empty;
  /* Erzeuge räumliche Anfrage */
5  foreach SE-id ∈ SE do
  | soutput := soutput + UNION SPATIAL ID SE-id UP? DOWN?;
  | /* Phase 2: Berechnung der Aktivierungen */
  | /* Aufbau des Spreading-Activation-Graphen */
  | V := Spatial_Entity_Type ∪ Spatial_Entity;
  | E := part_of ∪ consists_of ∪ type;
  | /* Laden der initialen Aktivierung, Gewichtung- und
  | Zerfallsfaktoren */
  | Load A, W, D;
  | /* Anwendung des Spreading-Activation-Algorithmus */
  | Gspatial := (V,E,A,W,D);
  | R := SA (Gspatial);
  | /* Rückgabe der transformierten Anfrage und der berechneten
  | Aktivierungswerte */
  | Result : soutput; R

```

Algorithmus 5.9 : Räumliche Anfragetransformation im Zielsystem mit Spreading-Activation

Der Algorithmus durchläuft dabei folgende Schritte.

1. Die erste Phase des Algorithmus läuft wie bei der Transformation durch Typanfragen (siehe Algorithmus 5.8) ab. Es wird eine räumliche Anfrage ausgehend von den Strukturtypen aus dem Pfad der Anfrage s_{input} erstellt (Zeile 2-10).
2. Die zweite Phase erstellt einen Spreading-Activation-Graph und lädt die initiale Aktivierung, Gewichtung- und Zerfallsfaktoren nach Definition 5.38 (Zeile 11-14). Die initiale

Aktivierung, Gewichtung- und Zerfallsfaktoren erlauben es, auf die Ausbreitung der Aktivierung Einfluss zu nehmen.

3. Es folgt die Anwendung des Spreading-Activation-Verfahrens auf den Spreading-Activation-Graph (Zeile 15).
4. Als Ergebnis liegt die Anfrage s_{output} und die Menge der Aktivierungswerte für Gebäudeteile im Zielsystem vor.

Die Transformation kann auch hier wieder unabhängig durchgeführt werden. Außer der Anfrage vom Quellsystem werden keine weiteren Informationen vom Quellsystem benötigt.

Die Aktivierungswerte werden, wie schon bei der thematischen Anfrage, als Ranking für die Gebäudeteile und damit auch als Ranking für das Ergebnis der Anfrage verwendet.

In Definition 5.38 sind die initiale Aktivierung, Gewichtungs- und Zerfallsfaktoren definiert, die für diese Variante der räumlichen Anfragerelaxation mit Spreading-Activation gewählt wird.

Definition 5.38 [Initiale Aktivierung, Gewichtungs- und Zerfallsfaktoren]

Ausgangspunkt für die Parameter ist folgende generische elementare räumliche typbasierte Anfrage:

TYPE SPATIAL ID SET-n / SET-(n-1) / ... / SET-0 UP? DOWN?

Basierend darauf werden folgende Parameter gesetzt:

Initiale Aktivierung A

$$A = \{a_{\text{SET-0}}, a_{\text{SET-1}}, \dots, a_{\text{SET-n}}\}; \quad a_{\text{SET-0}} = n + 1, a_{\text{SET-1}} = n, \dots, a_{\text{SET-n}} = 1$$

Gewichtungsfaktoren W

$$W = \{w_{\text{type}}^{\text{forward}}, w_{\text{type}}^{\text{backward}}, w_{\text{part_of}}^{\text{forward}}, w_{\text{part_of}}^{\text{backward}}, w_{\text{consists_of}}^{\text{forward}}, w_{\text{consists_of}}^{\text{backward}}\};$$

$$w_{\text{type}}^{\text{forward}} = 0, w_{\text{type}}^{\text{backward}} = 1,$$

$$w_{\text{part_of}}^{\text{forward}} = 0, 0, w_{\text{part_of}}^{\text{backward}} = 0, 5,$$

$$w_{\text{consists_of}}^{\text{forward}} = 0, 5, w_{\text{consists_of}}^{\text{backward}} = 0, 0$$

Zerfallsfaktoren D

$$D = \{d_{type}^{forward}, d_{type}^{backward}, d_{part_of}^{forward}, d_{part_of}^{backward}, d_{consists_of}^{forward}, d_{consists_of}^{backward}\};$$

$$d_{type}^{forward} = 0, d_{type}^{backward} = 1,$$

$$d_{part_of}^{forward} = 0, d_{part_of}^{backward} = 1,$$

$$d_{consists_of}^{forward} = 1, d_{consists_of}^{backward} = 0$$

Wie diese Parameter zustande gekommen sind, wird in Kapitel 6 genauer beschrieben.

□

Es muss, wie auch für das vorherige Verfahren, für diese Variante der räumlichen Anfragetransformation die Intentionserhaltung betrachtet werden.

Überprüfung 5.39 [Intentionserhaltende räumliche Anfragetransformation mit Spreading-Activation]

Die Intentionserhaltung der räumlichen Anfragetransformation wird für die möglichen Ausgabeanfragen bei einer gegebenen Eingabeanfrage betrachtet. Wie bereits bei der Betrachtung der Strukturserhaltung, wird die Anfragetransformation im Quell- und Zielsystem in einer Einheit betrachtet.

Die Ausgangssituation ist analog zur Betrachtung der Intentionserhaltung in Unterabschnitt 5.6.1.

Gehen wir dazu wieder von einer generischen Form einer elementaren thematischen Unteranfrage als Eingabeanfrage s_{input} im Quellsystem aus:

```
SPATIAL ID SE-0 UP? DOWN?
```

Nach der Transformation der Anfrage im Quellsystem ist das Ergebnis eine räumliche typbasierte Pfad-Anfrage $s_{intermediate}$:

```
TYPE SPATIAL ID SET-n/.../SET-0 UP? DOWN?
```

Im Zielsystem wird diese in eine räumliche Anfrage s_{output} , bestehend aus einzelnen elementaren räumlichen Anfragen, transformiert:

```
SPATIAL ID SE-1 UP? DOWN?
UNION
SPATIAL ID SE-2 UP? DOWN?
UNION
...
SPATIAL ID SE-m UP? DOWN?
```


Welche Gebäudeteile durch Spreading-Activation aktiviert werden, hängt von der initialen Aktivierung, den Gewichtung- und Zerfallsfaktoren ab.

In die Ausgabeanfrage s_{output} kommen nur Gebäudeteile $SE-1$ bis $SE-m$, deren Aktivierungswert über einem bestimmten Schwellenwert liegt und dessen Strukturtyp in der Anfrage $s_{\text{intermediate}}$ vorkommt.

Es gilt daher der 3. Fall der Definition 5.7 und somit $s_{\text{input}} \approx s_{\text{output}}$:

$$\begin{aligned} \exists \text{type}^- . \{SE-1, SE-2, \dots, SE-m\} &\sqsubseteq \exists \text{type}^- . (\exists \text{has_part_transitive} . \{SE-0\} \\ &\sqcup \exists \text{consists_of}^+ . \{SE-0\} \sqcup \{SE-0\}) \\ &= \{SET-0, SET-1, \dots, SET-n\} \end{aligned}$$

Diese Art der Intentionserhaltung setzt ebenfalls voraus, dass es mindestens einen gemeinsamen Strukturtypen in der Anfrage $s_{\text{intermediate}}$ in den Strukturgraphen des Quell- bzw. Zielsystems gibt. \square

Zur Illustration dieser Strategie werden im Folgenden zwei Beispiele gezeigt.

Beide Beispiele gehen davon aus, dass im Quell- als auch im Zielsystem die gleichen Strukturtypen vorhanden sind, die Strukturgraphen sich aber unterscheiden.

Beispiel 5.40 [Räumliche Anfragetransformation - Pfeiler im Außen- und Innenbau]

Dieses Beispiel greift die Anfrage nach einem Pfeiler in St. Lorenz auf.

Die Ausgangssituation ist wie in Beispiel 5.33.

Im Quellsystem, in diesem Beispiel St. Lorenz, wurde eine räumliche Anfrage nach einem Pfeiler gestellt.

Im Zielsystem, in diesem Beispiel St. Sebald, soll diese Anfrage relaxiert werden.

In Abbildung 5.15 ist dazu ein Ausschnitt aus dem Strukturgraphen von St. Sebald mit Aktivierungswerten zu sehen.

*Die Anfrage s_{source} selektiert nach dem Gebäudeteil **Lo:Pfeiler 5** im Quellsystem.*

SPATIAL ID Lo:Pfeiler 5

Übertragen an das Zielsystem wird wieder eine typbasierte Pfad-Anfrage $s_{\text{intermediate}}$.

TYPE SPATIAL ID Sacred_Building / Outside / Nave / Side_Aisle / Column

Gesucht werden bevorzugt alle Pfeiler, die an einem Seitenschiff im Außenbereich zu finden sind.

Ein Ergebnis der Anwendung der Anfragerelaxation (Phase 1) im Zielsystem ist die Anfrage s_{target} .

5. Anfragerelaxation und -transformation

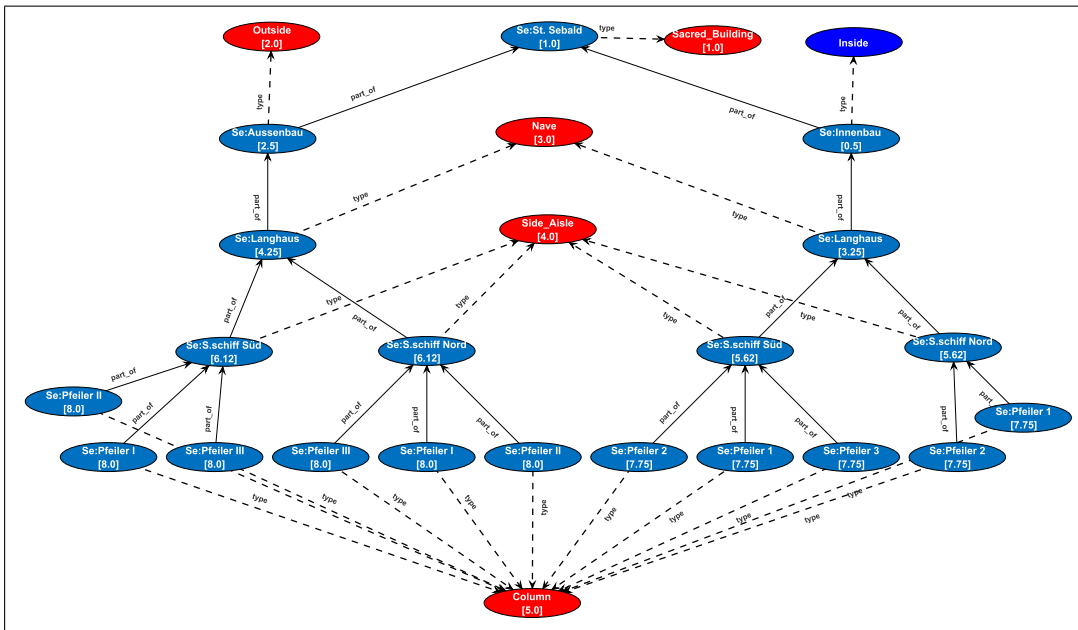


Abbildung 5.15.: Strukturgraph mit Aktivierungswerten - Anfrage *Lo:Pfeiler 5*

```

SPATIAL ID Se:Pfeiler I
UNION
SPATIAL ID Se:Pfeiler II
UNION
SPATIAL ID Se:Pfeiler III
UNION
...

```

In der zweiten Phase werden die Aktivierungswerte ausgehend von der Anfrage $S_{\text{intermediate}}$ berechnet. Die Anfrage $S_{\text{intermediate}}$ bestimmt dabei die initialen Aktivierungswerte (in der Abbildung bei den Strukturtypen zu sehen). Durch die zweite Phase kann die Strukturhierarchie miteinbezogen werden.

An diesem Beispiel ist zu sehen, dass die Pfeiler im **Außenbau** höher aktiviert (Aktivierungswert ist 8) wurden als die Pfeiler im **Innenbau** (Aktivierungswert ist 7,75).

Da der Pfeiler in der Anfrage S_{source} Teil des Außenbaus war, kann diese Eigenschaft über den Aktivierungswert transportiert werden. Dies ist bei dem Verfahren ohne Spreading-Activation (siehe Unterabschnitt 5.6.1) nicht möglich. \square

Beispiel 5.41 [Räumliche Anfragetransformation - Geschoss am Pfeiler im Außen- und Innenbau]

Dieses Beispiel erweitert das vorherige um eine weitere Unterscheidung.

Die Ausgangssituation ist hier eine Anfrage nach einem bestimmten **Geschoss** an einem Pfeiler im Außenbau.

Übertragen an das Zielsystem wird wieder eine typbasierte Pfad-Anfrage $t_{intermediate}$. Hierbei ist jetzt noch die Selektion nach einem **Geschoss** im Typ-Pfad sichtbar.

TYPE SPATIAL ID Sacred_Building / Outside / Nave / Side_Aisle / Column / Floor

Nach der Berechnung der Aktivierungswerte ergeben sich folgende zwei Ausschnitte aus dem Strukturgraphen.

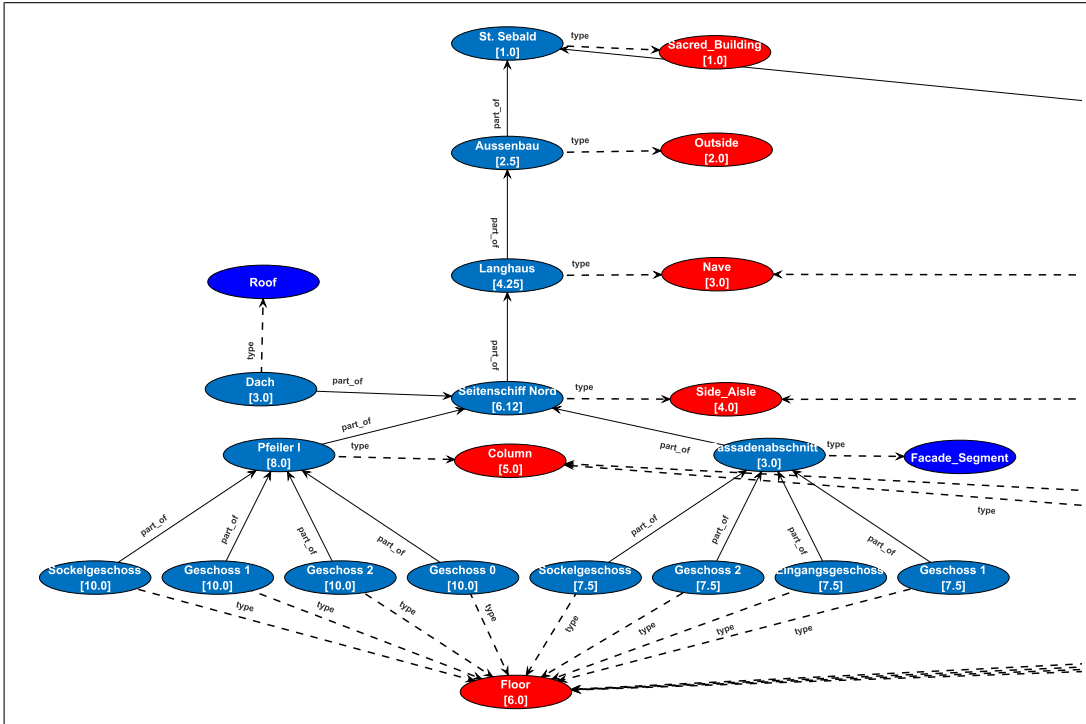


Abbildung 5.16.: Strukturgraph (Außenbau) mit Aktivierungswerten - Anfrage **Geschoss**

In Abbildung 5.16 sieht man, dass die Geschosse der Pfeiler höher aktiviert (Aktivierungswert ist 10) werden als die der Fassadenabschnitte (Aktivierungswert ist 7,5). Die Relaxation bevorzugt somit Geschosse der Pfeiler im Außenbau.

In Abbildung 5.17 sieht man, dass die Geschosse der Pfeiler ebenfalls aktiviert werden. Der Aktivierungswert der Geschosse (Aktivierungswert ist 9,87) ist dabei niedriger als bei den Geschossen im Außenbau (Aktivierungswert ist 10), aber höher als bei den Geschossen der Fassadenabschnitte (Aktivierungswert ist 7,5). Die Relaxation bevorzugt somit Geschosse der Pfeiler vor Geschossen der Fassadenabschnitte, abhängig davon ob sie im Außenbau oder Innenbau vorkommen. □

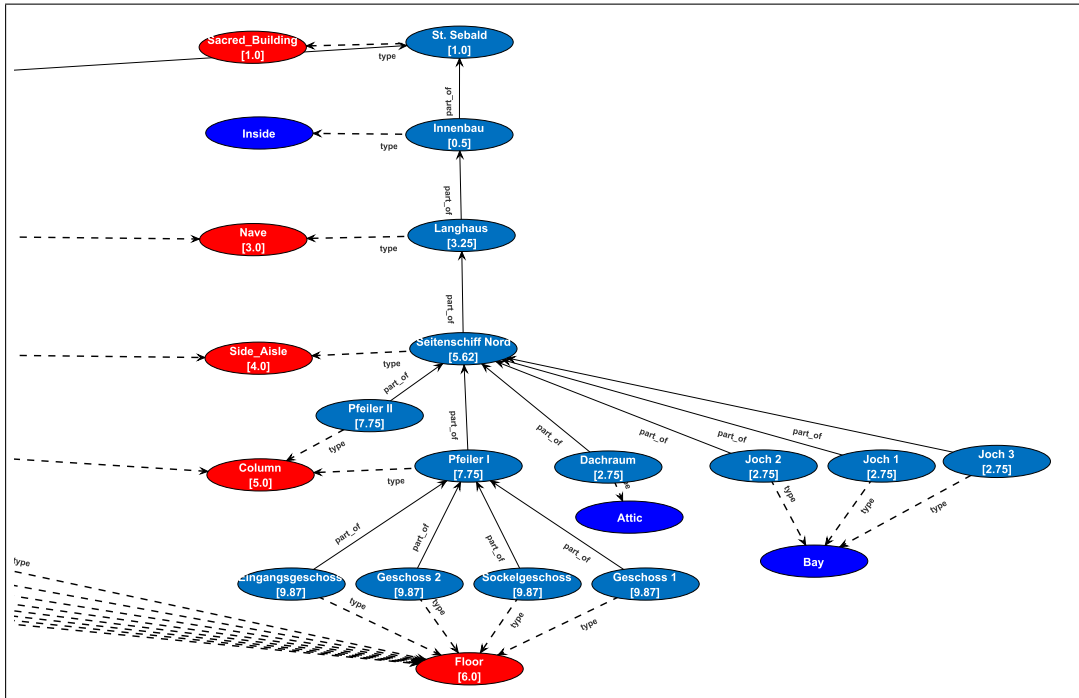


Abbildung 5.17.: Strukturgraph (Innenbau) mit Aktivierungswerten - Anfrage *Geschoss*

5.7. Kombination der Anfragerelaxation

Betrachtet man die räumliche und thematische Anfragerelaxation zusammen, ergibt sich der wie in Abbildung 5.18 gezeigte Ablauf.

Begonnen wird mit der Transformation der Quellenanfrage im Quellsystem mit Algorithmus 5.1. Hierbei wird die Anfragetransformation in eine räumliche (siehe Algorithmus 5.7) und eine thematische Anfragetransformation (siehe Algorithmus 5.4) aufgeteilt. Im Quellsystem werden die einzelnen Teilanfragen zusammengesetzt und an das Zielsystem geschickt.

Im Zielsystem wird eine Aufteilung der Anfrage mit Algorithmus 5.2 vorgenommen. Die Relaxation der räumlichen (siehe Algorithmus 5.9) als auch der thematischen (siehe Algorithmus 5.6) Anfragen finden jeweils in zwei Phasen statt.

Dabei werden in der ersten Phase die Teilanfragen transformiert und in der zweiten Phase das Spreading-Activation-Verfahren auf den Strukturgraphen und Themengraphen angewendet.

Abgeschlossen wird das gesamte Transformationsverfahren durch eine Zusammenführung der Teilanfragen und Aktivierungswerte. Da für jede Transformation einer elementaren Anfrage eigene Aktivierungswerte A berechnet werden, müssen diese vor der Zusammenführung normiert werden.

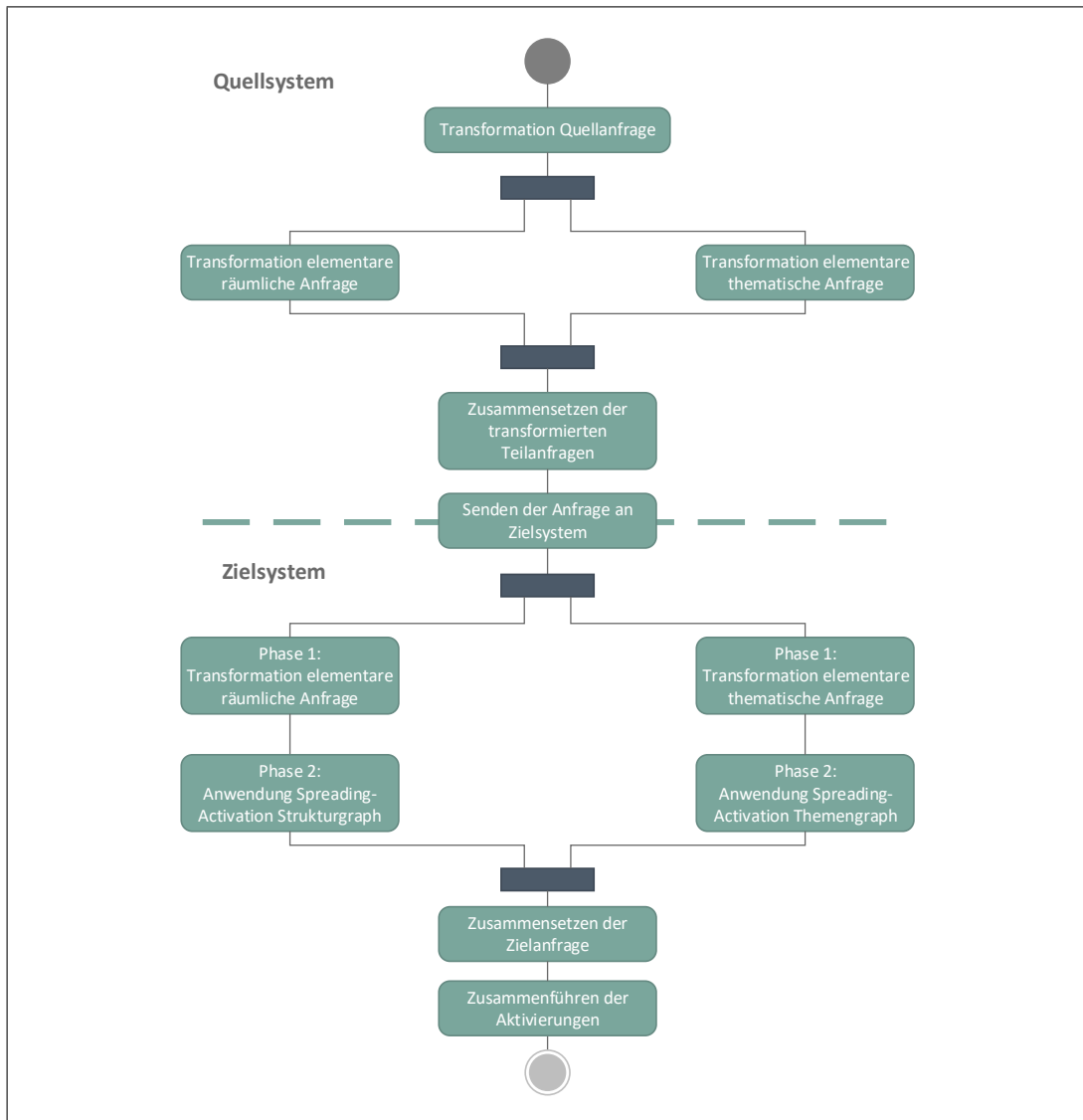


Abbildung 5.18.: Ablauf Anfragerelaxation - Gesamtbild

$$A' = \{a'_i | a'_i = \frac{a_i - \min(A)}{\max(A) - \min(A)}\}$$

Anschließend können die einzelnen normierten Aktivierungswerte der Transformationen A'_1, \dots, A'_n zusammengeführt werden.

$$A^{total} = \sum_{i=1}^n A'_i$$

Zusammen mit dem Ergebnis der relaxierten Anfrage können die so erhaltenen Aktivierungswerte als Ranking verwendet werden.

Beispiel 5.42 [Relaxation einer Anfrage aus dem MonArch-Anwendungsfall]

Betrachten wir die Anfrage nach **Sandstein** in Bezug auf einen **Pfeiler** im MonArch-Archiv von St. Lorenz:

```
SELECT *
FROM
    SPATIAL ID Lo:Pfeiler 5
INTERSECT
    TOPICAL ID MA:Sandstein DOWN
```

Diese wird im MonArch-Archiv von St. Sebald in eine Anfrage nach Pfeilern und **Sandstein** relaxiert. Die genaue Relaxation ist in Beispiel 5.40 und Beispiel 5.31 beschrieben.

Als Ergebnis der Transformation entsteht folgende Anfrage:

```
SELECT *
FROM (
    SPATIAL ID Se:Pfeiler I
    UNION
    SPATIAL ID Se:Pfeiler II
    UNION
    SPATIAL ID Se:Pfeiler 1
    UNION
    ...
)
INTERSECT
(
    TOPICAL ID MA:Material DOWN
    UNION
    TOPICAL ID MA:Werkstoff DOWN
    UNION
    TOPICAL ID MA:Stein DOWN
    UNION
    TOPICAL ID MA:Naturstein DOWN
    UNION
    TOPICAL ID MA:Sandstein DOWN
    UNION
    TOPICAL ID GND:Sandstein DOWN
)
```

Das Spreading-Activation sorgt nun dafür, dass z.B. Dokumente, die mit einem Pfeiler im Außenbau und Sandstein ausgezeichnet wurden, als relevanter betrachtet werden, als z.B. Dokumente, die allgemein mit Stein und Pfeiler im Innenbau ausgezeichnet wurden. □

5.8. Eigenschaften und Grenzen des Transformationsansatzes

Die in diesem Kapitel vorgestellte Anfragerelaxation und -transformation stellt eine Alternative zu einer Berechnung eines Alignments zwischen Strukturgraphen und Themengraphen dar.

Es konnte ein Transformationsverfahren gefunden werden, das intentionserhaltend ist und zusätzlich eine Anfragerelaxation erlaubt.

Dabei werden die in Abschnitt 5.2 definierten Zieleigenschaften wie folgt erfüllt.

Intentionserhaltung der Anfrage

Die Anfragetransformation einer Anfrage in MonQL ist intentionserhaltend, da sowohl die Anfragetransformation der thematischen als auch räumlichen Anfragen intentionserhaltend sind.

Das vorgestellte Transformationsverfahren stellt sicher, dass die Anfragen einzeln und unabhängig betrachtet werden können. Die Struktur der Anfrage bleibt dadurch ebenfalls erhalten.

Relaxation der Anfrage

Die verschiedenen Strategien erlauben zusätzlich eine Relaxation der Anfrage.

Insbesondere der Einsatz des Spreading-Activation erlaubt zusätzlich ein Ranking der Elemente im Strukturgraphen und Themengraphen. Dieses Ranking kann auf das Ergebnis der transformierten Anfrage übertragen werden.

Für die räumlichen Anfragen erlaubt die Transformation in eine Typ-Anfrage die Relaxation im Strukturgraphen. Im Themengraphen wird die Relaxation durch eine thematische Pfad-Anfrage erreicht.

Die Relaxation der Anfrage erlaubt eine Abbildung der Gebäudeteile und Themen mit 1:N zum Anfragezeitpunkt.

Einbeziehung der Strukturhierarchie

Durch die Transformation einer räumlichen Anfrage im Zielsystem in eine Typ-Anfrage unter Berücksichtigung der *part_of*- bzw. *consists_of*-Beziehung, wird die Strukturhierarchie des Zielsystems in die Anfragetransformation miteinbezogen.

Im Zielsystem kann durch die initiale Aktivierung der Typen im Spreading-Activation-Verfahren einerseits eine Übertragung der Gebäudestruktur erreicht werden, andererseits kann durch eine Gewichtung der Einfluss der Gebäudestruktur auf die transformierte Anfrage beeinflusst werden.

Einbeziehung der Themenhierarchie

Bei der Transformation einer thematischen Anfrage im Zielsystem stellt die Pfad-Anfrage sicher, dass die Oberthemen mit in die Anfragerelaxation einfließen.

Unabhängigkeit

Die Anfragetransformation benötigt keinen Austausch des Strukturgraphen und Themengraphen zwischen Quell- und Zielsystem. Daher berücksichtigt die Anfragetransformation die Unabhängigkeit der einzelnen MonArch-Archive, im Gegensatz zum Einsatz von Ontology-Matching-Verfahren.

Benennungsschemata und Mehrsprachigkeit

Weder die räumlichen noch die thematischen Anfragetransformationen nutzen die Benennung der Gebäudeteile oder der Themen, um die Transformation durchzuführen.

Die Transformation kann unter den gegebenen Voraussetzungen (siehe Definition 5.16) rein ID-basiert arbeiten. Deshalb muss die sprachabhängige Benennung von Gebäudeteilen (`Spatial_Entity`) oder Themen (`Topic`) nicht zur Identifizierung herangezogen werden.

Automatisierung

Weder im Quellsystem noch im Zielsystem ist eine Interaktion mit dem Benutzer notwendig um eine Anfragetransformation durchzuführen.

Das Verfahren erlaubt aber durch eine Anpassung der Gewichtungsfaktoren des Spreading-Activation eine Einflussnahme des Benutzers auf die Relaxation der Anfrage.

Die Transformation der Anfrage wird zum Anfragezeitpunkt durchgeführt. Dies hat zusätzlich den Vorteil, dass Änderungen am Strukturgraphen und Themengraphen sofort und ohne Zeitverlust von der transformierten Anfrage berücksichtigt werden können, ohne einen zeitaufwendigen Schritt zur Berechnung eines neuen Alignments vorschalten zu müssen.

Da sich die Anfragerelaxation durch die Wahl der Parameter beeinflussen lässt, ist die Optimierung dieser eine Aufgabe, die im Rahmen eines Feldtests mit den MonArch-Anwendern zu lösen ist. Dieser Feldtest steht in zukünftigen MonArch-Projekten bevor.

In Kapitel 6 ist beschrieben, wie man prinzipiell bei der Bestimmung dieser Werte vorgehen kann. Für die Optimierung der Gewichtungsfaktoren wurden mit einem kleinen Kreis von MonArch-Anwendern das vorgeschlagene Verfahren durchgespielt und somit eine initiale Konfiguration der Gewichtungsfaktoren ermittelt. Dies muss mit einem größeren Benutzerkreis wiederholt werden.

Zusammenfassend lässt sich sagen, dass unter den getroffenen Annahmen, eine Alternative zu Ontology-Matching-Verfahren gefunden wurde, die sich durch die Veränderung der Parameter flexibel an die Bedürfnisse der Anwender anpassen lässt.

Für den MonArch-Anwendungsfall ermöglicht die Kombination der räumlichen und thematischen Anfragetransformation eine erweiterte Verbundanfrage.

Ausgehend von einer Anfrage an ein MonArch-Archiv kann eine Verbundanfrage automatisch und intentionserhaltend durch Transformation erstellt werden, die in anderen MonArch-Archiven angewendet werden kann. Dabei ist es nicht notwendig, dass die MonArch-Archive ihre Struktur- und Themengraphen untereinander austauschen.

Da die beschriebene Anfragetransformation immer eine Relaxation der Ursprungsanfrage bewirkt, wird der Suchraum erweitert. Diese Suchraumerweiterung war eine Forderung, die in Abschnitt 1.2 gestellt wurde.

Sind die Annahmen, die in Definition 5.16 getroffen wurden, nicht oder nur teilweise erfüllt, liefert das vorgestellte Verfahren nicht die gewünschten Ergebnisse.

Ein Fall in dem die Annahmen nicht erfüllt sind, ist z.B. wenn versucht wird eine Anfrage an einen Strukturgraphen eines Sakralbaus in eine Anfrage an einen Strukturgraphen eines Hochhauses zu transformieren. Hierbei unterscheiden sich die Strukturtypen so stark voneinander, dass keine Gemeinsamkeit gefunden werden kann, um eine intentionserhaltende Anfragerelaxation durchzuführen.

Ein ähnliches Argument lässt sich auch für Themengraphen anbringen, indem keine gemeinsamen Oberthemen existieren.

Werden die gleichen Strukturtypen verwendet, stehen jedoch die Gebäudeteile in konträren *part_of*-Beziehungen, so tritt ein weiterer Fall ein, in dem die Transformation von räumlichen Anfragen nicht die gewünschten Ergebnisse erzielt. Beispiel 5.43 zeigt diesen Fall.

Beispiel 5.43 [Räumliche Anfragetransformation - Pfeiler im Außen- und Innenbau (umgedreht)]

In diesem Beispiel sind die Voraussetzungen wie in Beispiel 5.40. Im Quellsystem wird räumlich auf einen Pfeiler selektiert. Die Phase 1 der Anfragetransformation ist ebenfalls identisch.

*Der Unterschied besteht hier im Strukturgraphen im Zielsystem (siehe Abbildung 5.19). In diesem Strukturgraphen ist die Unterscheidung zwischen **Außenbau** und **Innenbau** nicht auf der Ebene des Gebäudes, sondern auf der Ebene der Bauteile, in diesem Beispiel die Pfeiler, getroffen worden.*

*Dies führt nun dazu, dass zwar der **Außenbau** höher aktiviert wird als der **Innenbau**, die Pfeiler aber nochmals höher aktiviert werden. Als Folge sind z.B. Dokumente die allgemein an einem Pfeiler verortet wurden höher im Ranking, als Dokumente die im Außenbau eines Pfeilers verortet wurden.*

Kompensiert werden kann dies durch eine Anpassung der initialen Aktivierung des Außenbaus, die höher sein muss als die Aktivierung der Pfeiler (siehe Abbildung 5.20).

□

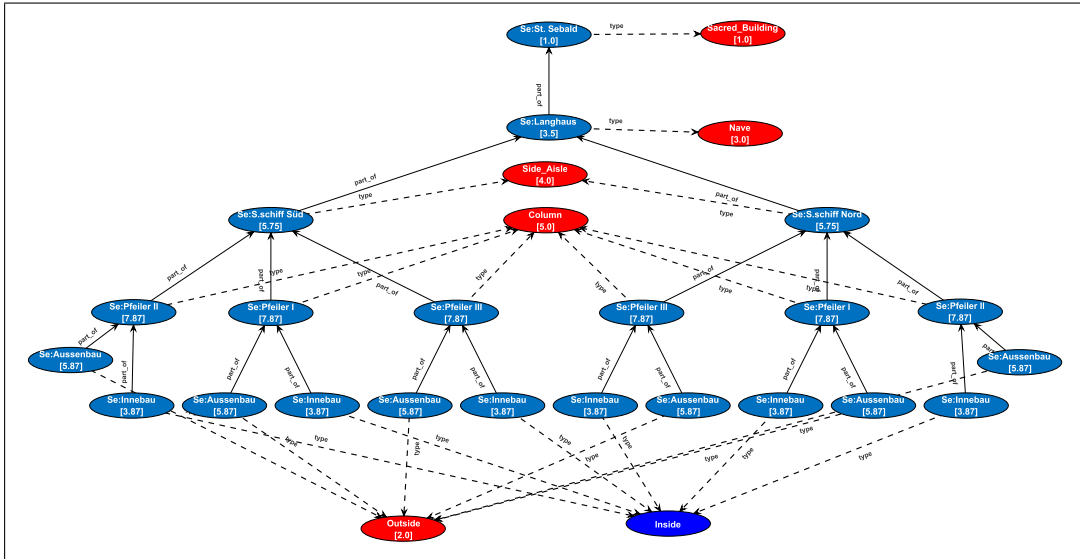


Abbildung 5.19.: Strukturgraph (umgedreht) mit Aktivierungswerten - Anfrage *Pfeiler*

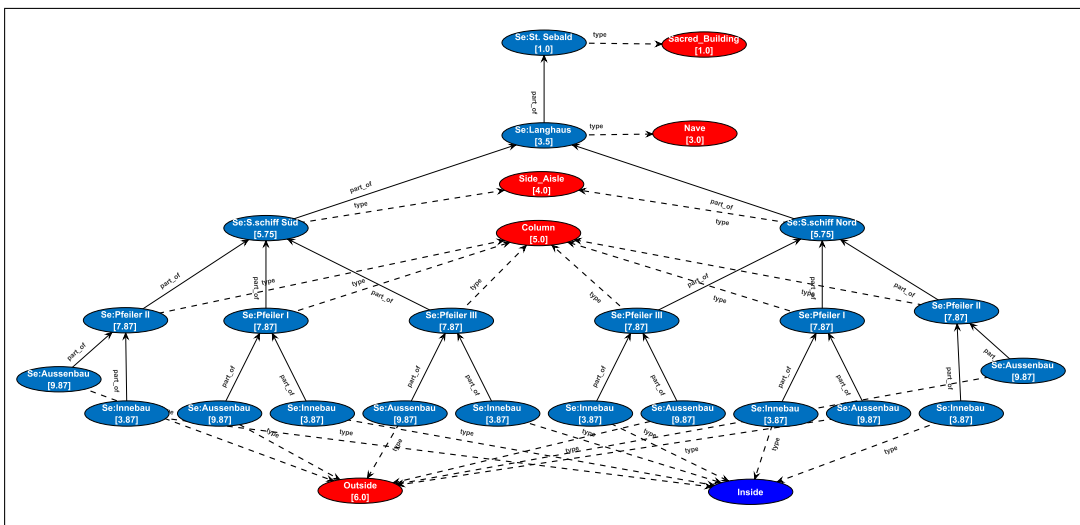


Abbildung 5.20.: Strukturgraph (umgedreht) mit Aktivierungswerten - Alternative Aktivierung

6. Eignung für den praktischen Einsatz

In Kapitel 5 wurde eine Anfragetransformation vorgestellt, die eine Anfrage eines Quellsystems in eine Anfrage eines Zielsystems umwandelt.

Dazu werden selektierte Gebäudeteile in der Quellanfrage in ein oder mehrere Gebäudeteile in der Zielanfrage abgebildet. Die Auswahl der Gebäudeteile erfolgt auf Basis eines angepassten Spreading-Activation-Verfahrens.

Dieses Verfahren benötigt mehrere Parameter, die eine Auswahl der Gebäudeteile beeinflussen können. Daher müssen die Auswahl der Parameterwerte und deren Einfluss diskutiert werden.

In der Masterarbeit [Hel11] wurde eine Referenz- und Ergebnismenge für eine Auswahl an Anfragen definiert und evaluiert, die als Grundlage für die Optimierung der Parameterwerte des Spreading-Activation-Verfahrens herangezogen wurden.

6.1. Verwendete Gütekriterien

Für die Bewertung der Ergebnisse werden statistische Gütekriterien wie *Precision* und *Recall* verwendet (Definitionen entnommen aus [BR10]), die für das vorgestellte Verfahren wie folgt angepasst wurden.

Im Rahmen der Bewertung der ausgewählten Gebäudeteile einer Anfrage Q lassen sich die Gebäudeteile der Anfrage in vier Klassen einteilen.

- **True Positives** (t_p): Gefundene relevante Gebäudeteile
$$t_p = SE_{Ref} \cap SE_{SA}$$
- **False Positives** (f_p): Gefundene irrelevante Gebäudeteile
$$f_p = SE_{SA} \setminus SE_{Ref}$$
- **True Negatives** (t_n): Nicht gefundene irrelevante Gebäudeteile
$$t_n = SE \setminus (SE_{SA} \cup SE_{Ref})$$
- **False Negatives** (f_n): Nicht gefundene relevante Gebäudeteile
$$f_n = SE_{Ref} \setminus SE_{SA}$$

Die Menge SE_{Ref} beinhaltet die Referenzgebäudeteile aus den Vergleichsanfragen. Die Menge SE_{SA} sind die selektierten Gebäudeteile, die durch die Anfragetransformation entstehen.

Precision

Precision P beschreibt den Anteil der gefundenen relevanten Gebäudeteile SE_{Ref} des Konzeptes `Spatial_Entity` unter allen gefundenen Gebäudeteilen SE_{SA} einer Anfrage Q [BR10; Hen08].

$$\begin{aligned}
 P &= \frac{\text{gefundene relevante Gebäudeteile}}{\text{gefundene Gebäudeteile insgesamt}} \\
 &= \frac{t_p}{t_p \cup f_p} \\
 &= \frac{SE_{Ref} \cap SE_{SA}}{(SE_{Ref} \cap SE_{SA}) \cup (SE_{SA} \setminus SE_{Ref})} \\
 &= \frac{SE_{Ref} \cap SE_{SA}}{SE_{SA}}
 \end{aligned}$$

Rated Precision

Da die Referenz- und Ergebnismenge von Hand ausgewertet wurden, sind diese nicht vollständig. D.h. nicht alle Gebäudeteile in der theoretischen SE_{Ref} Menge sind bewertet worden. Daher wird die Menge SE_{Ref} auf bewertete Ergebnisse SE_{rated} beschränkt.

$$SE_{Ref} = \{e \in SE_{rated} \mid rating(e) \geq 1\}$$

Damit ergibt sich eine Rated Precision PR :

$$\begin{aligned}
 RP &= \frac{\text{gefundene relevante Gebäudeteile mit Bewertung}}{\text{gefundene Gebäudeteile mit Bewertung insgesamt}} \\
 &= \frac{SE_{Ref} \cap SE'_{SA}}{SE'_{SA}} \\
 &= \frac{SE_{Ref} \cap (SE_{rated} \cap SE_{SA})}{SE_{rated} \cap SE_{SA}} \\
 &= \frac{SE_{Ref} \cap SE_{SA}}{SE_{rated} \cap SE_{SA}}
 \end{aligned}$$

Layered Precision

Bei der Erstellung der Referenzmenge wurde die Relevanz einzelner Gebäudeteile mit 0 bis 5 bewertet. Dabei hat 0 für die Anfrage die niedrigste Relevanz und 5 die höchste.

Um die Score-Werte, die durch das Spreading-Activation-Verfahren für die Gebäudeteile berechnet werden, mit den Werten der Referenzmenge vergleichen zu können, müssen diese auf das Intervall $[0, 5]$ abgebildet werden.

$f(x)$ bildet die Score-Werte auf das Intervall $[0, 5]$ ab, wobei:

- \min_SA der minimale Score-Wert des Spreading-Activation-Verfahrens ist.
- \max_SA der maximale Score-Wert des Spreading-Activation-Verfahrens ist.

- min_LS der minimale Wert der Bewertung der Referenzgebäudeteile ist. Hier als 0 gewählt.
- max_LS der maximale Wert der Bewertung der Referenzgebäudeteile ist. Hier als 5 gewählt.

$$\begin{aligned}
 f(x) &= \frac{(x - min_{SA}) * (max_{LS} - min_{LS})}{max_{SA} - min_{SA}} + min_{LS} \\
 &= \frac{(x - min_{SA}) * (5 - 0)}{max_{SA} - min_{SA}} + 0 \\
 &= \frac{(x - min_{SA}) * 5}{max_{SA} - min_{SA}}
 \end{aligned}$$

Werden im Spreading-Activation-Verfahren auch negative Werte für die Gewichtungsfaktoren w zugelassen, muss dies bei der Umrechnung berücksichtigt werden. Im Allgemeinen bedeutet ein negativer Score-Wert, dass ein selektiertes Gebäudeteil nicht relevant ist. Dieses muss daher auf 0 abgebildet werden.

In der Layered Precision wird für jede Äquivalenzklasse 1, 2, 3, 4 oder 5 separat die Precision berechnet.

$$\begin{aligned}
 SE_{Ref,i} &= \{e \in SE_{Ref} \mid likert(e) = i\} \\
 SE_{Ref} &= \bigcup_{i=1}^5 SE_{Ref,i}
 \end{aligned}$$

$$\begin{aligned}
 SE_{SA,i} &= \{e \in SE_{SA} \mid likert(e) = i\} \\
 SE_{SA} &= \bigcup_{i=1}^5 SE_{SA,i}
 \end{aligned}$$

$$likert(e) := \left\{ \begin{array}{ll} \lceil relevance(e) \rceil & e \in SE_{Rated} \\ \lceil f(relevance(e)) \rceil & e \in SE_{SA} \end{array} \right\} \in \{1, 2, 3, 4, 5\}$$

Die Menge $SE_{Ref,i}$ beschreibt diejenigen Entitäten der Referenzmenge SE_{Ref} , die aufgrund der *likert*-Funktion in die Äquivalenzklasse i eingeordnet werden.

Dies hat den Vorteil, dass in fünf Stufen überprüft werden kann, ob das Verfahren die Gebäudeteile passend bei der Transformation der Anfrage auswählt.

Die *Layered Precision* ergibt sich aus dem arithmetischen Mittel der *Precision* der einzelnen Äquivalenzklassen.

$$P_i = \frac{E_{Ref,i} \cap E_{SA,i}}{E_{SA,i}}$$

$$LP = \frac{1}{5} \sum_{i=1}^5 P_i$$

Layered Rated Precision

Anstelle der hier definierten *Precision* führt die Verwendung der *Rated Precision* zur *Layered Rated Precision LRP*.

$$LRP = \frac{1}{5} \sum_{i=1}^5 RP_i$$

Recall

Recall R beschreibt das Verhältnis zwischen relevanten Gebäudeteilen im Ergebnis zu allen relevanten Gebäudeteilen.[BR10; Hen08].

$$R = \frac{\text{gefundene relevante Gebäudeteile}}{\text{relevante Gebäudeteile insgesamt}}$$

$$= \frac{t_p}{t_p \cup f_n}$$

$$= \frac{SE_{Ref} \cap SE_{SA}}{(SE_{Ref} \cap SE_{SA}) \cup (SE_{Ref} \setminus SE_{SA})}$$

$$= \frac{SE_{Ref} \cap SE_{SA}}{SE_{Ref}}$$

Rated Recall

Analog zur Rated Precision werden beim Rated Recall RR erneut nur Gebäudeteile in Betracht gezogen, zu denen auch eine Bewertung in der Referenzmenge vorliegt.

$$RR = \frac{\text{gefundene relevante Gebäudeteile mit Bewertung}}{\text{relevante Gebäudeteile mit Bewertung insgesamt}}$$

$$= \frac{SE_{Ref} \cap SE'_{SA}}{SE_{Ref}}$$

$$= \frac{SE_{Ref} \cap (SE_{rated} \cap SE_{SA})}{SE_{Ref}}$$

$$= \frac{SE_{Ref} \cap SE_{SA}}{SE_{Ref}}$$

$$= R$$

Layered Recall

Wie bereits bei der *Layered Precision* wird ein *Layered Recall LR* definiert als:

$$R_i = \frac{SE_{Ref,i} \cap SE_{SA,i}}{SE_{Ref,i}}$$

$$LR = \frac{1}{5} \sum_{i=1}^5 R_i = LRR$$

F-Measure, Rated F-Measure, Layered F-Measure

Der F-Measure kombiniert *Precision* und *Recall* in einer einzigen Kennzahl. Es ist als das gewichtete harmonische Mittel aus *Precision* und *Recall* definiert. Die folgende Definition ist aus [Chi92] entnommen.

$$F = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}, \text{ mit } \beta^2 = \frac{1 - \alpha}{\alpha} \in [0, \infty[$$

Eine zusätzliche Gewichtung kann durch einen Parameter $\alpha \in [0, 1]$ erreicht werden. [Hen08]

Die *Precision* wird mit einem Wert $\beta < 1$ stärker bewertet, wohingegen der *Recall* mit einem Wert $\beta > 1$ stärker berücksichtigt wird.

Das ausbalancierte *F-Measure* ($\alpha = 0.5$ und $\beta = 1$) gewichtet *Precision* und *Recall* jeweils gleich und wird häufig verwendet [MRS08].

$$F_1 = \frac{2 * P * R}{P + R}$$

Für den Rated- und Layered-Fall lässt sich das F-Measure ebenfalls berechnen.

$$RF_1 = \frac{2 * RP * RR}{RP + RR}$$

$$LF_1 = \frac{2 * LP * LR}{LP + LR}$$

$$LRF_1 = \frac{2 * LRP * LRR}{LRP + LRR}$$

6.2. Anpassung des Spreading-Activation-Verfahren

Für den MonArch-Anwendungsfall muss das Verfahren zur Transformation der Typ-Anfrage angepasst werden. Diese Anpassung kann durch Veränderung der Parameter des Spreading-Activation-Verfahren erzielt werden. Hierzu wurden in mehreren Schritten, ausgehend von

einer vorher definierten Referenzmenge, verschiedene Parameterkombinationen ermittelt und anhand der vorher definierten *F-Measure* Gütekriterien bewertet.

1. Im ersten Schritt wurde eine Befragung von sieben Probanden mit jeweils zehn Fragen durchgeführt. Dabei waren die Probanden gemischt aus Domänenexperten aus dem MonArch-Projekt und Technikexperten.

Die Aufgabe der Probanden war es, ausgehend von der Gebäudeteilhierarchie von St. Lorenz, die jeweils passenden Gebäudeteile der Gebäudeteilhierarchie von St. Sebald auf einer Skala von 0 bis 5 zu bewerten.

Die Fragen wurden dabei immer mit dem Hintergrund gestellt, welches Gebäudeteil ausgehend von einem selektierten Gebäudeteil in St. Lorenz für die Beantwortung einer Anfrage relevant sein könnte.

Über die Ergebnisse wurde ein arithmetisches Mittel der Relevanz gebildet. Dieser Schritt der Ermittlung der Menge SE_{rated} war notwendig, da es noch keine vergleichbaren Daten gab, um Gebäudeteile aufeinander abzubilden.

Die Menge SE_{ref} enthält alle Gebäudeteile die eine Relevanz > 0 aufweisen.

2. Im zweiten Schritt wurden ausgehend von den Anfragen in den zehn gestellten Fragen jeweils die transformierten Anfragen ermittelt und die ausgewählten Gebäudeteile mit den berechneten Score-Werten durch das Spreading-Activation-Verfahren in der Menge SE_{SA} gespeichert.
3. Im dritten Schritt konnten mit den Mengen SE_{ref} und SE_{SA} die verschiedenen Varianten des F-Measure berechnet werden.
4. Im vierten Schritt wurden der zweite und dritte Schritt mit unterschiedlichen Parametern für das Spreading-Activation-Verfahren mehrmals wiederholt. Ziel war es dabei eine optimale Parameterkombination für das Verfahren zu ermitteln.

Da nicht alle Kombinationsmöglichkeiten durchgerechnet werden konnten, musste bei der Veränderung der Parameter in definierten Schritten vorgegangen werden.

Für die Ermittlung der Parameterwerte wurden folgende Annahmen getroffen:

- Der Gewichtungsfaktor w und der Zerfallsfaktor d liegen im Intervall von $[0, 1]$.
- Es werden die Kanten, abgeleitet von den Rollen *type*, *part_of* und *consists_of*, betrachtet.
- Es werden die Kanten in Hin- und Rückrichtung separat behandelt.
- Für Kante *type*, also vom *Spatial_Entity* zum *Spatial_Entity_Type*, soll der Gewichtungsfaktor $w = 0$ sein. Eine Ausbreitung der Aktivierung von Gebäudeteil zu Gebäudeteiltyp ist nämlich nicht gewünscht, da für die Transformation am Ende die Gebäudeteile und nicht deren Typen relevant sind.

- Für die Kanten *part_of* und *consists_of* soll der Gewichtungsfaktor für die Richtung Gebäudeteil zum übergeordneten Gebäudeteil kleiner sein als in die andere Richtung. Dies soll bewirken, dass im Allgemeinen speziellere Gebäudeteile selektiert werden.
- Die Parameter werden in Schrittweiten von 0,1 erhöht.

6.3. Auswertung der Ergebnisse

Eine ausführliche Präsentation der Ergebnisse und Zahlen aus der Bestimmung der Parameterwerte kann in [Hel11] nachgelesen werden. In diesem Abschnitt werden die wichtigsten Erkenntnisse zusammengefasst.

Für jede Kante im Spreading-Activation-Graphen wurde für Gewichtungsfaktor und Zerfallsfaktor im Intervall von $[0, 1]$ in Schrittweiten von 0,1 der F-Measure für die Variante 1 (vgl. Unterabschnitt 5.6.1) und Variante 2 (vgl. Unterabschnitt 5.6.2) der Anfragetransformation berechnet.

Im Folgenden werden für eine Kante aus dem Spreading-Activation-Graphen, die aus der Rolle *part_of* hervorgegangen ist, die Ergebnisse exemplarisch diskutiert.

Auswertung Top-Down: *part_of*

Abbildung 6.1 zeigt in der Top-Down-Richtung, also vom Gebäudeteil zum untergeordneten Gebäudeteil, die Auswertung des Gewichtungsfaktors bei angenommenen Zerfallsfaktor von $d = 1$.

Auf der X-Achse sind die möglichen Werte für den Gewichtungsfaktor in Schrittweiten von 0,1 aufgetragen. Die Y-Achse zeigt den F-Measure-Wert im Intervall von $[0, 1]$.

Variante 1, die rein typbasierte Transformation, ist zum Vergleich mit gestrichelten Linien dargestellt. Variante 2, das Spreading-Activation-Verfahren, ist mit durchgezogenen Linien dargestellt.

Die Grafik zeigt, dass, je nach F-Measure Variante, die höchsten Werte für den Gewichtungsfaktor bei $w = 0,5$ bzw. $w = 0,4$ erreicht werden. Dies lässt den Schluss zu, dass eine Weitergabe der Aktivierung entlang der *part_of*-Hierarchie in Top-Down-Richtung nicht zu stark und nicht zu schwach ausfallen darf.

Ein Gewichtungsfaktor von $w = 0,5$ gewährleistet, dass die Gebäudestruktur bei der Transformation mit einbezogen wird.

Dies ermöglicht z.B., dass die Pfeiler im Außenbau höher aktiviert werden als Pfeiler im Innenbau wenn die ursprüngliche Anfrage einen Pfeiler im Außenbau selektierte.

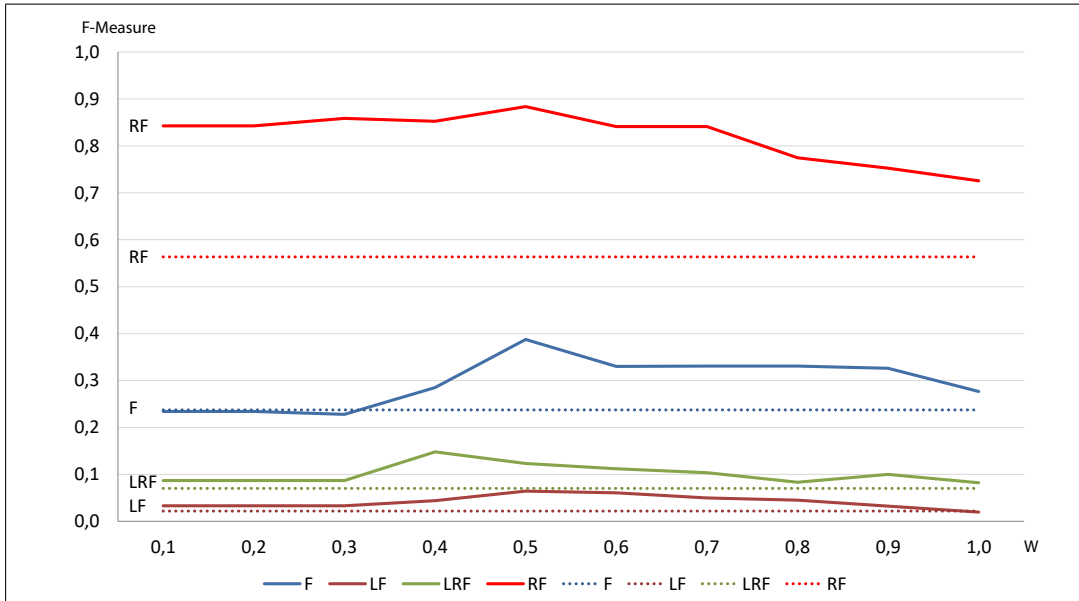


Abbildung 6.1.: Top-Down: *part_of*-Gewichtung in Richtung untergeordnetes Gebäudeteil (entnommen aus [Hel11])

Der RF-Measure zeigt hierbei sehr schön, wie gut das Verfahren genau die Gebäudeteile selektiert, die auch in der Referenzmenge selektiert wurden. Ein Wert von fast 0,9 für den Gewichtungsfaktor von $w = 0,5$ gibt erste Hinweise für die letztendliche Parameterwahl. Die weiteren F-Measure Verfahren zeigen ähnliche Maxima bei einem Gewichtungsfaktor von $w = 0,5$.

Die niedrigen absoluten Werte der anderen Verfahren lassen sich dadurch erklären, dass die Anzahl der Gebäudeteile in der Referenzmenge im Vergleich zu der Anzahl der Gebäudeteile von St. Lorenz und St. Sebald stark reduziert ist.

Die Grafik zeigt darüber hinaus auch, dass die Variante 2 der Anfragetransformation der Variante 1 überlegen ist, da die Gebäudestruktur bei der Anfragetransformation mit einbezogen wird.

Auswertung Bottom-Up: *part_of*

Betrachtet man die Bottom-Up-Richtung entlang der *part_of*-Hierarchie zeigt sich die Auswirkung des Gewichtungsfaktors auf die Güte der Transformation noch viel deutlicher.

Abbildung 6.2 zeigt in der Bottom-Up-Richtung, also vom Gebäudeteil zum übergeordneten Gebäudeteil, die Auswertung des Gewichtungsfaktors bei angenommenen Zerfallsfaktor von $d = 0,9$. Es gelten, wie vorher, weiter dieselben Annahmen.

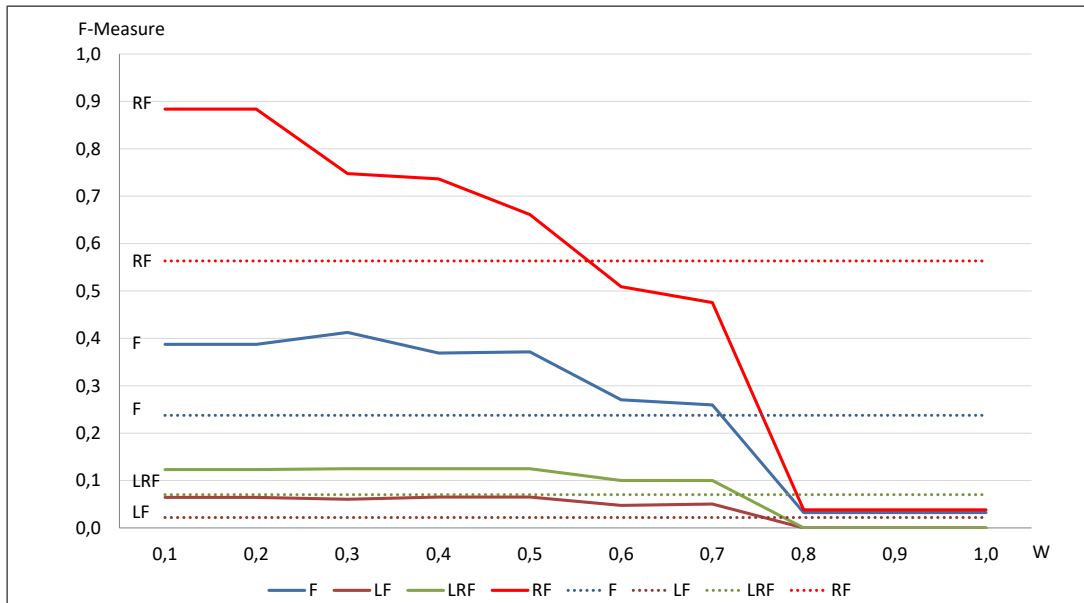


Abbildung 6.2.: Bottom-Up: *part_of*-Gewichtung in Richtung übergeordnetes Gebäudeteil (entnommen aus [Hel11])

Die Grafik zeigt, dass, je nach F-Measure Variante, die höchsten Werte für den Gewichtungsfaktor bei $w = 0, 1$ erreicht werden. Für höhere Gewichtungsfaktoren für diese Kantenrichtung und Kante verschlechtern sich die Werte der berechneten F-Measure Varianten. Dies zeigt, dass eine Ausbreitung der Aktivierung von Gebäudeteilen zu übergeordneten Gebäudeteilen nicht zu stark ausfallen darf, da sonst die falschen Gebäudeteile nach der Transformation der Anfrage selektiert werden.

Dies würde unter anderem dazu führen, dass, ausgehend von einem Pfeiler, die übergeordneten Gebäudeteile wie Seitenschiff stärker bei der transformierten Anfrage berücksichtigt werden. Dabei werden die Pfeiler im Seitenschiff immer mehr vernachlässigt. Die Befragung zeigt jedoch, dass ein solches Verhalten nicht erwünscht ist.

Auch in der Betrachtung der Rückrichtung ist die Variante 2 der Anfragetransformation der Variante 1 überlegen.

Zusammenfassung

Diese beiden Auswertungen zeigen auszugsweise wie die Parameter für das Verfahren ermittelt wurden. Bei der Auswertung der Kombinationsmöglichkeiten von Gewichtungsfaktor und Zerfallsfaktor wurden folgende Werte (siehe Tabelle 6.1) ermittelt, die dann in die Realisierung des Verfahrens übernommen wurden. Die Tabelle zeigt, für Kanten *part_of*, *consists_of* und *type* in Hin- und Rückrichtung, die ermittelten Werte für Gewichtungsfaktor und Zerfallsfaktor.

Kante	Gewichtungsfaktor	Zerfallsfaktor
<i>part_of</i>	0,1	0,9
<i>has_part</i> \equiv <i>part_of</i> ⁻	0,5	1,0
<i>consists_of</i>	0,5	1,0
<i>consists_of</i> ⁻	0,1	0,9
<i>type</i>	0,0	0,0
<i>type</i> ⁻	1,0	1,0

Tabelle 6.1.: Ermittelte Parameter für das Spreading-Activation-Verfahren im MonArch-Anwendungsfall

Für die Kante *part_of*⁻ und somit auch für die Kante *consists_of* hat sich ein Gewichtungsfaktor $w = 0,5$ in der Richtung vom übergeordneten Gebäudeteil zum untergeordneten Gebäudeteil als ideal erwiesen. So wird einerseits die Gebäudestruktur mitberücksichtigt andererseits wirkt es nicht zu dominant bei der Auswahl der Gebäudeteile.

In der Gegenrichtung dagegen, ergaben die Messungen einen Wert des Gewichtungsfaktors von $w \leq 0,1$. Als Zerfallsfaktor ergab sich ein Wert von $d = 1$ für $w = 0,5$ und $d = 0,9$ für $w = 0,1$.

Für die Kante *type*⁻, also die Verbreitung der Aktivierung von den Typen zu den Gebäudeteilen, hat sich ein Gewichtungsfaktor von $w = 1$ und ein Zerfallsfaktor von $d = 1$ für die Richtung vom Typ zum Gebäudeteil bewährt. Die Rückrichtung wird immer auf $w = 0$ gesetzt, da hier eine Verbreitung der Aktivierung ausgeschlossen werden sollte.

Es muss hier noch angemerkt werden, dass die Bestimmung der Parameter sehr stark von der durchgeführten Befragung zur Ermittlung der Referenzmenge abhängt.

Änderungen am Datenmodell können durch neue Parameter für neue Kanten abgebildet werden.

6.4. Täglicher Einsatz

In Zusammenarbeit mit der Masterarbeit wurde das Verfahren für die Strukturhierarchie in das MonArch-System integriert und ist nun Bestandteil aller laufenden MonArch-Installationen.

Das MonArch-System ist in den folgenden Bauwerken, teilweise täglich, im Einsatz:

- St. Lorenz und St. Sebald in Nürnberg,
- Kaiser- und Barbarathermen in Trier,
- Weißenhofsiedlung in Stuttgart,
- Pfalzenforschung in Aachen,

- Bamberger Dom in Bamberg,
- Passauer Dom in Passau,
- St. Kassian in Regensburg,
- Denkmal und Kontext in Buchara (Usbekistan).

Für St. Lorenz und St. Sebald sowie für die Kaiser- und Barbarathermen wird die Verbundanfrage über mehrere Gebäude hinweg durchgeführt. In den anderen MonArch-Archiven kann die Anfragetransformation zur Relaxation der Anfragen im selben Gebäude genutzt werden.

Der zusätzliche Aufwand, der durch die Transformation bei der Anfragebearbeitung entsteht, fällt in der praktischen Nutzung nicht ins Gewicht. Hier ist die Netzwerkanbindung zwischen den einzelnen Archiven der limitierende Faktor. Die Berechnung der Aktivierung kann für jeden Knoten unabhängig durchgeführt werden, da nur eine Abhängigkeit auf Werte in einer vorangegangenen Iteration vorhanden ist. Daher kann das Verfahren parallelisiert werden, um die Bearbeitung zu beschleunigen.

7. Zusammenfassung und Ausblick

Die Möglichkeit einzelne MonArch-Archive zu einem MonArch-Verbund zusammenzuschließen eröffnet neue Anfragemöglichkeiten. Insbesondere die Verbundanfrage, also eine verteilte Anfrage über alle Mitglieder im Verbund, ermöglicht es, die verteilten Informationen über Bauwerke im Verbund zu nutzen. Verbundanfragen werden mit fortschreitender Vernetzung von Informationsbeständen zunehmend relevant, insbesondere auch im Anwendungsfall MonArch. Für die Realisierung einer Verbundanfrage ist es notwendig, Anfragen zu transformieren, um von gebäudespezifischen Strukturgraphen abstrahieren zu können.

In dieser Arbeit wurden die Voraussetzungen, Eigenschaften und Grenzen der Verbundanfrage untersucht und ein Anfragetransformations- und -relaxationsverfahren vorgestellt. Ermöglicht wird das vorgestellte Anfragetransformations- und -relaxationsverfahren durch folgende Schritte:

- Konsolidierung, Formalisierung und Erweiterung des Datenmodells
 - Einführung zusätzlicher Typinformationen für Gebäudeteile.

Dies erlaubt für räumliche Anfragen eine Relaxation der Anfrage auf Strukturtypen und damit eine Abstrahierung von gebäudespezifischen Strukturgraphen.
 - Übergang von einer Strukturhierarchie zu einem Strukturgraphen.

Diese Konsolidierung erlaubt neben einer besseren Abbildung von Gebäudemodellen, die Einbindung der Typinformationen in die Strukturhierarchie bzw. den Strukturgraphen. Darüber hinaus dient die Erweiterung als Grundlage für die Anfragetransformation.
 - Übergang von einer Themenhierarchie zu einem Themengraphen.

Dies öffnet das Datenmodell für Verknüpfungen von Themenhierarchien zu Themengraphen und somit für die Einbindung externer Vokabulare, die die thematische Anfragetransformation unterstützen können.
- Reformulierung des erweiterten Datenmodells in Beschreibungslogik
 - Anreicherung um weitere Eigenschaften und Überprüfung der Eigenschaften auf Konsistenz.

Die weiteren Eigenschaften können bei der Transformation der Anfrage genutzt werden, um Annahmen zu treffen, die eine Anfragetransformation erleichtern.

- Grundlage für den Datenaustausch in OWL-2-DL.

Die Reformulierung ist ein weiterer Baustein zur Einbindung von externen Vokabularen und ermöglicht die Anwendbarkeit bestehender Verfahren.

- Definition einer Anfragesprache für räumliche und thematische Anfragen

- kompakte Formulierung der Anfragen im MonArch-Anwendungsfall.

Dies erlaubt es, die Anfragetransformation und insbesondere die Anfragetransformationsregeln auf Basis des reduzierten Sprachumfangs zu definieren.

- Reduktion der Komplexität für eine unabhängige Anfragetransformation.

Im Rahmen der vorliegenden Arbeit konnte gezeigt werden, dass sich das vorgestellte Anfragetransformations- und -relaxationsverfahren in Kombination mit einem angepassten Spreading-Activation-Verfahren sowohl für die räumliche als auch für die thematische Anfragetransformation im MonArch-Anwendungsfall eignet.

Vorteile der vorgestellten Herangehensweise:

- Eine aufwendige Abbildung zwischen Strukturgraphen kann vermieden werden.
- Verbundanfragen können unter bestimmten Bedingungen auch ermöglicht werden, wenn Strukturgraphen und Themengraphen nicht im Verbund geteilt werden, da die Relaxation der Anfrage lokal durchgeführt werden kann.
- Verbundanfragen können sprachübergreifend durchgeführt werden.
- Die Transformation und Relaxation der Anfrage kann zur Laufzeit stattfindet und kann somit bei jeder Anfragetransformation beeinflusst werden.
- Die Intention der zu transformierende Anfrage bleibt in der Verbundanfrage erhalten.
- Kann nicht nur im Verbund angewendet werden, sondern als reiner Relaxationsmechanismus in nicht vernetzten Archiven.

Abschließend wurde das Anfragetransformationsverfahren an die Anforderungen aus dem MonArch-Anwendungsfall angepasst.

Eine Aufgabe, die sich an diese Arbeit anschließen könnte, ist eine Untersuchung, wie weit das vorgestellte Verfahren in anderen Anwendungsfällen trägt. Insbesondere der Bereich des Linked Open Data könnte in Bezug auf thematische Anfragen und deren Transformation als Hintergrundwissen analog zu den Typinformationen für Gebäudeteile eingesetzt werden um für den Fall komplett unterschiedlicher Themenhierarchien auch eine Transformation zu ermöglichen.

A. Themenhierarchie

Beispiel A.1 [Themenhierarchie von St. Lorenz und St. Sebald]

Zur Schadensbeschreibung können folgende Themen und deren Unterthemen verwendet werden:

Schaden Beschreibt einen Schaden der in einem Dokument beschrieben wird und sich auf ein Bauteil oder Segment des Gebäudes bezieht. Ein Beispiel ist ein Dokument, das einen **Riss** in Form einer CAD-Zeichnung dokumentiert. Es können auch Schäden auf den Originaldokumenten beschrieben werden, deren digitale Kopie im Archiv als Dokument gespeichert ist. Ein Beispiel hierfür ist ein **Knick** auf einem Plan, der digitalisiert wurde und als Bild im Archiv abgespeichert ist.

Zustand Der Zustand beschreibt den aktuellen Zustand von Bauteilen, die z.B. auf einem Bild zu sehen sind.

Material Der Material-Thesaurus beinhaltet eine Vielzahl an Materialien die in den Großkirchen vorhanden sind.

Um bestimmte Elemente im Inneren und Äußeren zu beschreiben, stehen folgende Themen und deren Unterthemen zur Verfügung:

Bauornamentik Im Thesaurus Bauornamentik sind z.B. Begriffe wie **Fiale** und **Wasserspeier** zu finden.

Ausstattung Im Thesaurus Ausstattung sind hauptsächlich Elemente im Innenbereich wie **Altar** und **Glocke** zu finden.

Ikonografie Der Thesaurus Ikonografie enthält z.B. **Heilige** und **Apostel**.

Liturgie Der Thesaurus Liturgie umfasst Terme zur Beschreibung von Zeremonien und Riten des Gottesdienstes.

Technische Zeichen und Strukturen Hiermit können u.a. **Steinmetzzeichen** ausgezeichnet werden.

Skulptur Der Thesaurus Skulptur beschreibt ähnlich wie Ikonografie besondere Elemente im Innenraum.

Technik Im Technik-Thesaurus finden sich Elemente der Technikinstallation, wie z.B. **Heizung**.

Um Reparaturen am Gebäude beschreiben zu können, steht folgendes Thema und dessen Unterthemen zur Verfügung:

Maßnahme *Der Thesaurus Maßnahme beschreibt die verschiedenen Phasen, die bei einer Reparaturmaßnahme durchlaufen werden. **Sicherungsmaßnahme** kann beispielsweise in Form eines Dokuments beschrieben werden. Somit wird dieses Dokument mit diesem Thema ausgezeichnet.*

Um den Standort, Urheber und Besitzer beschreiben zu können, existieren folgende Themen und deren Unterthemen:

Person *Beschreibt Urheber und Rechthebesitzer.*

Standort *Mit dem Thesaurus Standort können neben dem räumlichen Bezug über die hierarchische Gebäudeteilzerlegung auch der Standort von Archivalien beschrieben werden, die sich nicht im Gebäude befinden. Diese sind z.B. in einem externen Archiv oder Museum.*

Rechte *Der Rechte-Thesaurus beschreibt den Status der Zugriffsrechte in Bezug auf das Urheberrecht. Dies ist insbesondere interessant, wenn Archivalien für die Allgemeinheit freigegeben werden sollen.*

Um die zeitliche Einordnung abzubilden, steht ein weiteres Thema und dessen Unterthemen zur Verfügung.

Zeitliche Einordnung *Hiermit werden wichtige baugeschichtliche Ereignisse abgebildet.*

□

B. Ontologien

Definition B.1 [Typontologie - Schema]

$$\mathcal{O}_{\text{schema}}^{\text{spatial_type}} := (\mathbf{C}, \mathbf{R}, \mathbf{I}, \mathbf{X}) \text{ mit:}$$

den Konzepten \mathbf{C} :

Typ für Gebäudeteile: `Spatial_Entity_Type`

den Rollen \mathbf{R} :

Überordnete Typpolle:	<i>sub_type_transitive</i>	<i>super_type</i>
<i>spatial_type_relation</i>	<i>sub_type</i>	
Typhierarchierollen:	<i>super_type_transitive</i>	

den Individuen $\mathbf{I} = \emptyset$

den Axiomen \mathbf{X} :

Rollenhierarchie:

sub_type_transitive \sqsubseteq *spatial_type_relation*
sub_type \sqsubseteq *sub_type_transitive*
super_type_transitive \sqsubseteq *spatial_type_relation*
super_type \sqsubseteq *super_type_transitive*

Inverse Rollen:

super_type \equiv *sub_type*⁻
sub_type_transitive \equiv *sub_type_transitive*⁻

Transitive Rollen:

sub_type_transitive \equiv *sub_type_transitive*⁺
super_type_transitive \equiv *super_type_transitive*⁺

Funktionale Rollen:

$\top \sqsubseteq \leq 1$ *super_type*
 $\top \sqsubseteq \leq 1$ *sub_type*⁻

Asymmetrische Rollen:

Asy(*sub_type*)
Asy(*super_type*)

Irreflexive Rollen:

sub_type \sqcap *sub_type*⁻ $\sqsubseteq \perp$
super_type \sqcap *super_type*⁻ $\sqsubseteq \perp$

Definition- & Wertebereich:

\exists *spatial_type_relation*. $\top \sqsubseteq$ `Spatial_Entity_Type`
 $\top \sqsubseteq \forall$ *spatial_type_relation*. `Spatial_Entity_Type`

□

Beispiel B.2 [Typontologie - Instanz]

Die Ontologie $\mathcal{O}_{\text{instance}}^{\text{spatial_type}}$ (StLorenz) modelliert exemplarisch mögliche Typen eines Sakralbaus, angefangen vom Gebäudetyp über einen allgemeinen Gebäudeteiltyp hin zu konkreten Gebäudeteiltypen wie Türmen.

B. Ontologien

$$\mathcal{O}_{\text{instance}}^{\text{spatial_type}}(\text{StLorenz}) := (\mathbf{C}, \mathbf{R}, \mathbf{I}, \mathbf{X}) \cup \mathcal{O}_{\text{schema}}^{\text{spatial_type}} \text{ mit:}$$

den Konzepten $\mathbf{C} = \emptyset$

den Rollen $\mathbf{R} = \emptyset$

den Individuen \mathbf{I} :

<i>Tower</i>	<i>Building</i>	<i>Section</i>	<i>Building_Part</i>
<i>North_Tower</i>	<i>Sacred_Building</i>	<i>Center_Section</i>	
<i>South_Tower</i>	<i>Cathedral</i>	<i>West_Building</i>	

den Axiomen \mathbf{X} :

<i>Spatial_Entity_Type(Tower)</i>	<i>Spatial_Entity_Type(Building_Part)</i>
<i>Spatial_Entity_Type(North_Tower)</i>	<i>super_type(Sacred_Building, Building)</i>
<i>Spatial_Entity_Type(South_Tower)</i>	<i>super_type(Cathedral, Sacred_Building)</i>
<i>Spatial_Entity_Type(Building)</i>	<i>super_type(Section, Building_Part)</i>
<i>Spatial_Entity_Type(Sacred_Building)</i>	<i>super_type(West_Building, Building_Part)</i>
<i>Spatial_Entity_Type(Cathedral)</i>	<i>super_type(Tower, Building_Part)</i>
<i>Spatial_Entity_Type(Section)</i>	<i>super_type(Center_Section, Section)</i>
<i>Spatial_Entity_Type(Center_Section)</i>	<i>super_type(North_Tower, Tower)</i>
<i>Spatial_Entity_Type(West_Building)</i>	<i>super_type(South_Tower, Tower)</i>

□

Definition B.3 [Raumontologie - Schema]

$$\mathcal{O}_{\text{schema}}^{\text{spatial}} := (\mathbf{C}, \mathbf{R}, \mathbf{I}, \mathbf{X}) \cup \mathcal{O}_{\text{schema}}^{\text{spatial_type}} \text{ mit:}$$

den Konzepten \mathbf{C} :

Gebäudeteile allg.: *Spatial_Entity*

Gebäudeteile symbolisch: *Spatial_Symbolic_Entity* Gebäudeteile grafisch: *Spatial_Geometry_Entity*

den Rollen \mathbf{R} :

Überordnete räumliche Rolle:

spatial_symbolic_relation

Strukturgraph:

part_of

consists_of

has_part

Typ:

type

grafische Repräsentation:

represents

Strukturgraph transitiv:

part_of_transitive

has_part_transitive

type_transitive

den Individuen $\mathbf{I} = \emptyset$

den Axiomen **X** :

Konzepthierarchie:

$Spatial_Symbolic_Entity \sqsubseteq Spatial_Entity$

$Spatial_Geometry_Entity \sqsubseteq Spatial_Entity$

Rollenhierarchie:

$part_of_transitive \sqsubseteq spatial_symbolic_relation$

$part_of \sqsubseteq part_of_transitive$

$has_part_transitive \sqsubseteq spatial_symbolic_relation$

$has_part \sqsubseteq has_part_transitive$

$consists_of \sqsubseteq spatial_symbolic_relation$

Inverse Rollen:

$part_of_transitive \equiv has_part_transitive^-$

$part_of \equiv has_part^-$

$has_part_transitive \equiv part_of_transitive^-$

$has_part \equiv part_of^-$

Transitive Rollen:

$part_of_transitive \equiv part_of_transitive^+$

$has_part_transitive \equiv has_part_transitive^+$

Funktionale Rollen:

$\top \sqsubseteq \leq 1 \ part_of$

$\top \sqsubseteq \leq 1 \ has_part^-$

$\top \sqsubseteq \leq 1 \ type$

$\top \sqsubseteq \leq 1 \ represents$

Asymmetrische Rollen:

Asy($part_of$)

Asy(has_part)

Asy($consists_of$)

Asy($represents$)

Irreflexive Rollen:

$part_of \sqcap part_of^- \sqsubseteq \perp$

$has_part \sqcap has_part^- \sqsubseteq \perp$

$consists_of \sqcap consists_of^- \sqsubseteq \perp$

$represents \sqcap represents^- \sqsubseteq \perp$

$type \sqcap type^- \sqsubseteq \perp$

Rollenkomposition:

$type \circ super_type_transitive \sqsubseteq type_transitive$

Definition- & Wertebereich:

$\exists represents.T \sqsubseteq Spatial_Entity$

$\top \sqsubseteq \forall represents.Spatial_Symbolic_Entity$

$\exists type_transitive.T \sqsubseteq Spatial_Entity$

$\top \sqsubseteq \forall type_transitive.Spatial_Entity_Type$

$\exists type.T \sqsubseteq Spatial_Entity$

$\top \sqsubseteq \forall type.Spatial_Entity_Type$

$\exists spatial_symbolic_relation.T \sqsubseteq Spatial_Entity$

$\top \sqsubseteq \forall spatial_symbolic_relation.Spatial_Entity$

Disjunkte Konzepte:

$Spatial_Entity_Type \sqcap Spatial_Entity \sqsubseteq \perp$

$Spatial_Symbolic_Entity \sqcap$

$Spatial_Geometry_Entity \sqsubseteq \perp$

Totale Konzeptspezialisierung:

$Spatial_Entity \equiv Spatial_Symbolic_Entity \sqcup$

$Spatial_Geometry_Entity$

□

Beispiel B.4 [Raumontologie - Instanz]

Die Ontologie $\mathcal{O}_{instance}^{spatial}(StLorenz)$ beschreibt exemplarisch für St. Lorenz einen typisier-
ten Strukturgraphen. Die Ontologie baut dabei auf den Ontologien $\mathcal{O}_{instance}^{spatial_type}(StLorenz)$,
 $\mathcal{O}_{schema}^{spatial}$ und indirekt auf der Ontologie $\mathcal{O}_{schema}^{spatial_type}$ auf.

$$\mathcal{O}_{instance}^{spatial}(StLorenz) := (C, R, I, X) \cup \mathcal{O}_{instance}^{spatial_type}(StLorenz) \cup \mathcal{O}_{schema}^{spatial} \text{ mit:}$$

den Konzepten $C = \emptyset$

den Rollen $R = \emptyset$

den Individuen I :

Nürnberg
St._Lorenz
Outside
Building_West
Tower_North

Middle_Section
Tower_South
Polygon1_Tower_North
Polygon2_Tower_North
Polygon1_Tower_South

Polygon2_Tower_South
Polygon1_Center_Section
Polygon2_Center_Section
Group_West_Building
Group_Tower_North

Group_Center_Section
Group_Tower_South

den Axiomen **X** :

Gebäudeteile symbolisch:

$Spatial_Symbolic_Entity(Nürnberg)$

B. Ontologien

Spatial_Symbolic_Entity(St._Lorenz)
Spatial_Symbolic_Entity(Outside)
Spatial_Symbolic_Entity(Building_West)
Spatial_Symbolic_Entity(Tower_North)
Spatial_Symbolic_Entity(Middle_Section)
Spatial_Symbolic_Entity(Tower_South)
Gebäudeteile grafisch:
Spatial_Geometry_Entity(Polygon1_Tower_North)
Spatial_Geometry_Entity(Polygon2_Tower_North)
Spatial_Geometry_Entity(Polygon1_Tower_South)
Spatial_Geometry_Entity(Polygon2_Tower_South)
Spatial_Geometry_Entity(Polygon1_Center_Section)
Spatial_Geometry_Entity(Polygon2_Center_Section)
Gruppierungselemente:
Spatial_Symbolic_Entity(Group_West_Building)
Spatial_Symbolic_Entity(Group_Tower_North)
Spatial_Symbolic_Entity(Group_Center_Section)
Spatial_Symbolic_Entity(Group_Tower_South)
Strukturhierarchie:
part_of(St._Lorenz, Nürnberg)
part_of(Outside, St._Lorenz)
part_of(Building_West, Outside)
part_of(Tower_North, Building_West)

part_of(Middle_Section, Building_West)
part_of(Tower_South, Building_West)
Verknüpfung symbolisch mit grafisch:
represents(Group_West_Building, Building_West)
represents(Group_Tower_North, Tower_North)
represents(Group_Center_Section, Middle_Section)
represents(Group_Tower_South, Tower_South)
Verknüpfung Gruppierungselemente:
consists_of(Group_West_Building, Group_Tower_North)
consists_of(Group_West_Building, Group_Center_Section)
consists_of(Group_West_Building, Group_Tower_South)
consists_of(Group_Tower_North, Polygon1_Tower_North)
consists_of(Group_Tower_North, Polygon2_Tower_North)
consists_of(Group_Center_Section, Polygon1_Middle_Sec.)
consists_of(Group_Center_Section, Polygon2_Middle_Sec.)
consists_of(Group_Tower_South, Polygon1_Tower_South)
consists_of(Group_Tower_South, Polygon2_Tower_South)
consists_of(Building_West, Group_West_Building)
Typ:
type(St._Lorenz, Cathedral)
type(Outside, Section)
type(Building_West, West_Building)
type(Tower_North, North_Tower)
type(Middle_Section, Center_Section)
type(Tower_South, South_Tower)

□

Definition B.5 [Rollen für DE9-IM - Schema]

$$\mathcal{O}_{\text{schema}}^{\text{spatial}}(\text{DE9-IM}) := (\mathbf{C}, \mathbf{R}, \mathbf{I}, \mathbf{X}) \cup \mathcal{O}_{\text{schema}}^{\text{spatial}} \text{ mit:}$$

den Konzepten $\mathbf{C} = \emptyset$

den Rollen \mathbf{R} :

<i>spatial_geometry_relation</i>	<i>crosses</i>	<i>overlaps</i>
<i>contains</i>	<i>disjoint</i>	<i>touches</i>
<i>covered_by</i>	<i>equals</i>	<i>within</i>
<i>covers</i>	<i>intersects</i>	

den Individuen $\mathbf{I} = \emptyset$

den Axiomen \mathbf{X} :

Rollenhierarchie:	<i>touches</i> \sqsubseteq <i>intersects</i>
<i>disjoint</i> \sqsubseteq <i>spatial_geometry_relation</i>	<i>contains</i> \sqsubseteq <i>covers</i>
<i>intersects</i> \sqsubseteq <i>spatial_geometry_relation</i>	<i>within</i> \sqsubseteq <i>covered_by</i>
<i>equals</i> \sqsubseteq <i>intersects</i>	Inverse Rollen:
<i>covered_by</i> \sqsubseteq <i>intersects</i>	<i>contains</i> \equiv <i>within</i> ⁻
<i>covers</i> \sqsubseteq <i>intersects</i>	<i>covered_by</i> \equiv <i>covers</i> ⁻
<i>crosses</i> \sqsubseteq <i>intersects</i>	Symmetrische Rollen:
<i>overlaps</i> \sqsubseteq <i>intersects</i>	<i>crosses</i> \equiv <i>crosses</i> ⁻

$disjoint \equiv disjoint^-$
 $overlaps \equiv overlaps^-$
 $touches \equiv touches^-$
 $equals \equiv equals^-$
 $intersects \equiv intersects^-$

Asymmetrische Rollen:

Asy(*contains*)
 Asy(*covered_by*)
 Asy(*covers*)
 Asy(*within*)

Irreflexive Rollen:

$spatial_geometry_relation \sqcap spatial_geometry_relation^- \sqsubseteq \perp$

Definition- & Wertebereich:

$\exists spatial_geometry_relation. \top \sqsubseteq$
 $Spatial_Geometry_Entity$
 $\top \sqsubseteq \forall spatial_geometry_relation.$
 $Spatial_Geometry_Entity$

Disjunkte Rollen:

$covered_by \sqcap crosses \sqsubseteq \perp$
 $covered_by \sqcap overlaps \sqsubseteq \perp$
 $covered_by \sqcap equals \sqsubseteq \perp$
 $covered_by \sqcap touches \sqsubseteq \perp$
 $covers \sqcap crosses \sqsubseteq \perp$
 $covers \sqcap overlaps \sqsubseteq \perp$
 $covers \sqcap equals \sqsubseteq \perp$
 $covers \sqcap touches \sqsubseteq \perp$
 $crosses \sqcap equals \sqsubseteq \perp$
 $crosses \sqcap touches \sqsubseteq \perp$
 $crosses \sqcap overlaps \sqsubseteq \perp$
 $disjoint \sqcap intersects \sqsubseteq \perp$
 $overlaps \sqcap touches \sqsubseteq \perp$
 $overlaps \sqcap equals \sqsubseteq \perp$
 $touches \sqcap equals \sqsubseteq \perp$

□

Beispiel B.6 [Rollen für DE9-IM - Instanz]

In Abbildung B.1 sind geometrische Elemente angedeutet, die z.B. die Westfassade von St. Lorenz repräsentieren können.

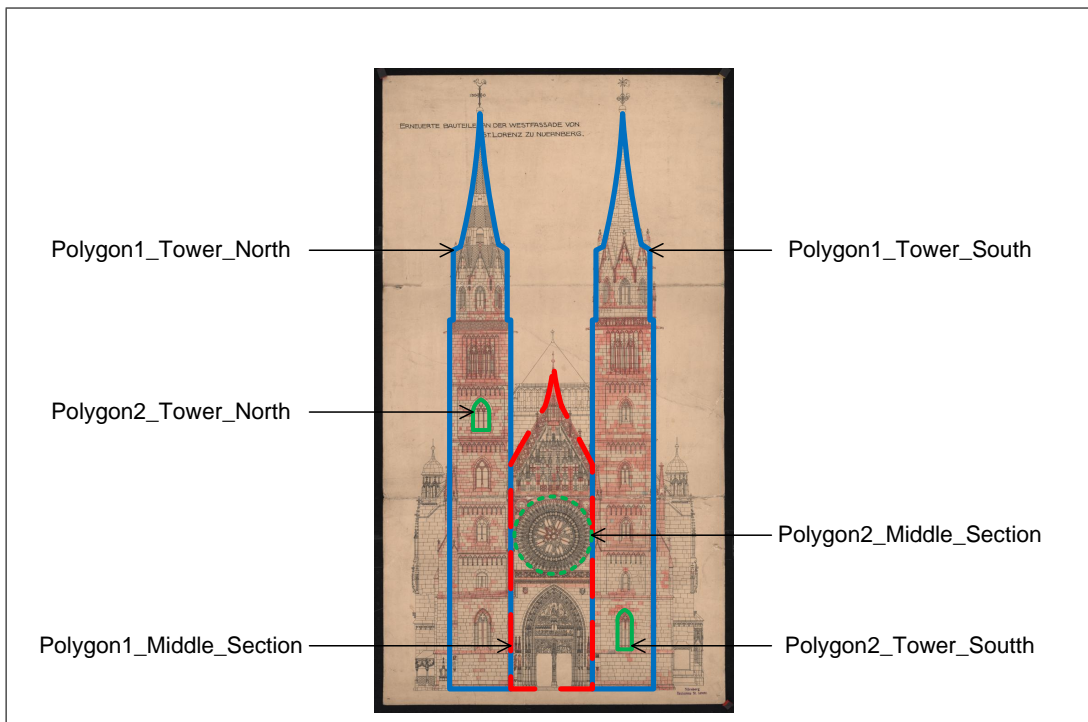


Abbildung B.1.: Polygone der Westfassade von St. Lorenz

$$\mathcal{O}_{instance}^{spatial}(DE9-IM) := (C, R, I, X) \cup \mathcal{O}_{schema}^{spatial}(DE9-IM) \text{ mit:}$$

B. Ontologien

den Konzepten $\mathbf{C} = \emptyset$

den Rollen $\mathbf{R} = \emptyset$

den Individuen $\mathbf{I} = \emptyset$

den Axiomen \mathbf{X} :

covers(Polygon1_Middle_Section, Polygon1_Middle_Section) *touches*(Polygon1_Middle_Section, Polygon1_Tower_South)

touches(Polygon1_Middle_Section, Polygon1_Tower_North) *contains*(Polygon1_Tower_North, Polygon2_Tower_North)
contains(Polygon1_Tower_South, Polygon2_Tower_South)

□

Definition B.7 [Themaontologie - Schema]

$\mathcal{O}_{\text{schema}}^{\text{topical}} := (\mathbf{C}, \mathbf{R}, \mathbf{I}, \mathbf{X})$ mit:

den Konzepten \mathbf{C} :

Thema: *Topic*

Themagruppe: *Topic_Group*

den Rollen \mathbf{R} :

Rollen Themengraph:

topic_relation

sub_topic

sub_topic_transitive

super_topic

super_topic_transitive

related_topic

Rollen Themengruppen:

member_topic

den Individuen $\mathbf{I} = \emptyset$

den Axiomen \mathbf{X} :

Rollenhierarchie:

sub_topic_transitive \sqsubseteq *topic_relation*

sub_topic \sqsubseteq *sub_topic_transitive*

super_topic_transitive \sqsubseteq *topic_relation*

super_topic \sqsubseteq *super_topic_transitive*

related_topic \sqsubseteq *topic_relation*

Inverse Rollen:

sub_topic_transitive \equiv *super_topic_transitive*⁻

sub_topic \equiv *super_topic*⁻

super_topic_transitive \equiv *sub_topic_transitive*⁻

super_topic \equiv *sub_topic*⁻

Transitive Rollen:

sub_topic_transitive \equiv *sub_topic_transitive*⁺

super_topic_transitive \equiv *super_topic_transitive*⁺

Funktionale Rollen:

$\top \sqsubseteq \leq 1$ *super_topic*

$\top \sqsubseteq \leq 1$ *sub_topic*⁻

Symmetrische Rollen:

related_topic \equiv *related_topic*⁻

Asymmetrische Rollen:

Asy(*member_topic*)

Asy(*sub_topic*)

Asy(*super_topic*)

Irreflexive Rollen:

member_topic \sqcap *member_topic*⁻ $\sqsubseteq \perp$

super_topic \sqcap *super_topic*⁻ $\sqsubseteq \perp$

sub_topic \sqcap *sub_topic*⁻ $\sqsubseteq \perp$

related_topic \sqcap *related_topic*⁻ $\sqsubseteq \perp$

Definition- & Wertebereich:

\exists *topic_relation*. $\top \sqsubseteq$ *Topic*

$\top \sqsubseteq \forall$ *topic_relation*. *Topic*

\exists *member_topic*. $\top \sqsubseteq$ *Topic* \sqcup *Topic_Group*

$\top \sqsubseteq \forall$ *member_topic*. *Topic_Group*

Disjunkte Konzepte:

Topic \sqcap *Topic_Group* $\sqsubseteq \perp$

Disjunkte Rollen:

related_topic \sqcap *super_topic* $\sqsubseteq \perp$

related_topic \sqcap *sub_topic* $\sqsubseteq \perp$

□

Beispiel B.8 [Themaontologie - Instanz]

Dieses Beispiel zeigt die Umsetzung eines Ausschnittes der Themenhierarchie als Ontologie $\mathcal{O}_{\text{instance}}^{\text{topical}}(\text{StLorenz})$ die auf der Ontologie $\mathcal{O}_{\text{schema}}^{\text{topical}}$ basiert.

$$\mathcal{O}_{\text{instance}}^{\text{topical}}(\text{StLorenz}) := (\mathbf{C}, \mathbf{R}, \mathbf{I}, \mathbf{X}) \cup \mathcal{O}_{\text{schema}}^{\text{topical}} \text{ mit:}$$

den Konzepten $\mathbf{C} = \emptyset$

den Rollen $\mathbf{R} = \emptyset$

den Individuen \mathbf{I} :

<i>MonArch_Standard</i>	<i>Restoration_Schmitz_Schulz</i>	<i>Substance</i>	<i>Slate</i>
<i>Building_History</i>		<i>Stone</i>	<i>Sandstone</i>
<i>1903_1939</i>	<i>Material</i>	<i>Natural_Stone</i>	<i>Quartzite</i>

den Axiomen \mathbf{X} :

<i>Topic_Group(MonArch_Standard)</i>	<i>member_topic(Material, MonArch_Standard)</i>
<i>Topic_Group(Building_History)</i>	<i>member_topic(Building_History, MonArch_Standard)</i>
<i>Topic(1903_1939)</i>	<i>member_topic(Substance, Material)</i>
<i>Topic(Restoration_Schmitz_Schulz)</i>	<i>super_topic(Stone, Substance)</i>
<i>Topic_Group(Material)</i>	<i>super_topic(Natural_Stone, Stone)</i>
<i>Topic(Substance)</i>	<i>super_topic(Slate, Natural_Stone)</i>
<i>Topic(Natural_Stone)</i>	<i>super_topic(Quartzite, Natural_Stone)</i>
<i>Topic(Stone)</i>	<i>super_topic(Sandstone, Natural_Stone)</i>
<i>Topic(Slate)</i>	<i>member_topic(1903_1939, Building_History)</i>
<i>Topic(Sandstone)</i>	<i>super_topic(Restoration_Schmitz_Schulz, 1903_1939)</i>
<i>Topic(Quartzite)</i>	

□

Definition B.9 [Abbildung Themaontologie auf SKOS]

Aufbauend auf der Themaontologie $\mathcal{O}_{\text{schema}}^{\text{topical}}$ werden weitere Konzepte und Rollen aus dem SKOS-Standard definiert. Die eigentliche Abbildung wird mit Gleichsetzung der Konzepte und Rollen in Beschreibungslogik realisiert.

$$\mathcal{O}_{\text{mapping}}^{\text{topical}}(\text{SKOS}) := (\mathbf{C}, \mathbf{R}, \mathbf{I}, \mathbf{X}) \cup \mathcal{O}_{\text{schema}}^{\text{topical}} \text{ mit:}$$

den Konzepten \mathbf{C} :

skos:Concept

skos:ConceptScheme

den Rollen \mathbf{R} :

skos:narrower
skos:broader

skos:member
skos:related

B. Ontologien

den Individuen $\mathbf{I} = \emptyset$

den Axiomen \mathbf{X} :

`skos:Concept = Topic`
`skos:ConceptScheme = Topic_Group`
`skos:narrower = sub_topic`
`skos:broader = super_topic`
`skos:member = member_topic`
`skos:related = related_topic`

□

Definition B.10 [Objektontologie - Schema]

$\mathcal{O}_{\text{schema}}^{\text{object}} := (\mathbf{C}, \mathbf{R}, \mathbf{I}, \mathbf{X}) \cup \mathcal{O}_{\text{schema}}^{\text{spatial}} \cup \mathcal{O}_{\text{schema}}^{\text{topical}}$ mit:

den Konzepten \mathbf{C} :

`Object_Entity`
`Document`
`Digital_Document`
`Analogue_Document`
`Reference`
`Object_Entity_Collection`
`Spatial_Object`
`Topical_Object`

den Rollen \mathbf{R} :

`spatial_reference`
`topical_reference`
`contains_Object_transitive`
`contains_Object`
`linked`
`is_newer_version`

den Individuen $\mathbf{I} = \emptyset$

den Axiomen \mathbf{X} :

Konzepthierarchie:

`Document` \sqsubseteq `Object_Entity`
`Digital_Document` \sqsubseteq `Document`
`Analogue_Document` \sqsubseteq `Document`
`Reference` \sqsubseteq `Object_Entity`
`Object_Entity_Collection` \sqsubseteq `Object_Entity`
`Topical_Object` \sqsubseteq `Object_Entity` \sqcap `Topic`
`Spatial_Object` \sqsubseteq `Object_Entity` \sqcap
`Spatial_Entity`

Rollenhierarchie:

`contains_Object` \sqsubseteq `contains_Object_transitive`

Transitive Rollen:

`contains_Object_transitive` \equiv `contains_Object_transitive`⁺

Funktionale Rollen:

$\top \sqsubseteq \leq 1$ `contains_Object`⁻

Asymmetrische Rollen:

`Asy(contains_Object)`

`Asy(is_newer_version)`

Irreflexive Rollen:

`contains_Object` \sqcap `contains_Object`⁻ $\sqsubseteq \perp$

`is_newer_version` \sqcap `is_newer_version`⁻ $\sqsubseteq \perp$

Rollenkomposition:

`contains_Object`⁻ \circ `spatial_reference` \sqsubseteq `spatial_reference`

`contains_Object`⁻ \circ `topical_reference` \sqsubseteq `topical_reference`

Definition- & Wertebereich:

\exists `spatial_reference`. $\top \sqsubseteq$ `Object_Entity`

$\top \sqsubseteq \forall$ `spatial_reference`. `Spatial_Entity`

\exists `topical_reference`. $\top \sqsubseteq$ `Object_Entity`

$\top \sqsubseteq \forall$ `topical_reference`. `Topic`

\exists `contains_Object_transitive`. $\top \sqsubseteq$

`Object_Entity_Collection`

$\top \sqsubseteq \forall$ `contains_Object_transitive`. `Object_Entity`

\exists `is_newer_version`. $\top \sqsubseteq$ `Document`

$\top \sqsubseteq \forall$ `is_newer_version`. `Document`

\exists `linked`. $\top \sqsubseteq$ `Object_Entity_Col.`

$\top \sqsubseteq \forall$ `linked`. `Object_Entity`

Disjunkte Konzepte:

`Object_Entity` \sqcap `Spatial_Entity_Type` $\sqsubseteq \perp$

`Object_Entity` \sqcap `Topic_Group` $\sqsubseteq \perp$

`Document` \sqcap `Spatial_Entity` $\sqsubseteq \perp$

$\text{Document} \sqcap \text{Topic} \sqsubseteq \perp$
 $\text{Document} \sqcap \text{Reference} \sqsubseteq \perp$
 $\text{Document} \sqcap \text{Object_Entity_Collection} \sqsubseteq \perp$
 $\text{Document} \sqcap \text{Spatial_Object} \sqsubseteq \perp$
 $\text{Document} \sqcap \text{Topical_Object} \sqsubseteq \perp$
 $\text{Digital_Document} \sqcap \text{Analogue_Document} \sqsubseteq \perp$
 $\text{Reference} \sqcap \text{Spatial_Entity} \sqsubseteq \perp$
 $\text{Reference} \sqcap \text{Topic} \sqsubseteq \perp$
 $\text{Reference} \sqcap \text{Object_Entity_Collection} \sqsubseteq \perp$
 $\text{Reference} \sqcap \text{Spatial_Object} \sqsubseteq \perp$

$\text{Reference} \sqcap \text{Topical_Object} \sqsubseteq \perp$
 $\text{Object_Entity_Col.} \sqcap \text{Spatial_Entity} \sqsubseteq \perp$
 $\text{Object_Entity_Col.} \sqcap \text{Topic} \sqsubseteq \perp$
 $\text{Object_Entity_Col.} \sqcap \text{Spatial_Object} \sqsubseteq \perp$
 $\text{Object_Entity_Col.} \sqcap \text{Topical_Object} \sqsubseteq \perp$
 $\text{Spatial_Object} \sqcap \text{Topic} \sqsubseteq \perp$
 $\text{Spatial_Object} \sqcap \text{Topical_Object} \sqsubseteq \perp$
 $\text{Topical_Object} \sqcap \text{Spatial_Entity} \sqsubseteq \perp$

□

Beispiel B.11 [Objektontologie - Instanz]

Auf Basis der Instanzontologie aus Beispiel B.4 für den Gebäudegraphen von St. Lorenz und auf der Instanzontologie aus Beispiel B.8 für den Themagraphen, definiere wir eine Beispiel Objektontologie.

Zusammen mit der Ontologie $\mathcal{O}_{\text{instance}}^{\text{topical}}(\text{StLorenz})$ und der Ontologie $\mathcal{O}_{\text{instance}}^{\text{spatial}}(\text{StLorenz})$ zeigt dieses Beispiel verschiedene Möglichkeiten der Ontologie $\mathcal{O}_{\text{instance}}^{\text{object}}(\text{StLorenz})$:

$\mathcal{O}_{\text{instance}}^{\text{object}}(\text{StLorenz}) := (\mathbf{C}, \mathbf{R}, \mathbf{I}, \mathbf{X}) \cup \mathcal{O}_{\text{schema}}^{\text{object}} \cup \mathcal{O}_{\text{instance}}^{\text{topical}}(\text{StLorenz}) \cup \mathcal{O}_{\text{instance}}^{\text{spatial}}(\text{StLorenz})$ mit:

den Konzepten $\mathbf{C} = \emptyset$

den Rollen $\mathbf{R} = \emptyset$

den Individuen \mathbf{I} :

Doc1	Doc3	Col1	Col3
Doc2	Doc4	Col2	Ref1

den Axiomen \mathbf{X} :

$\text{Digital_Document}(\text{Doc1})$ $\text{Digital_Document}(\text{Doc3})$ $\text{Digital_Document}(\text{Doc4})$ $\text{Analogue_Document}(\text{Doc2})$ $\text{Object_Entity_Collection}(\text{Col1})$ $\text{Object_Entity_Collection}(\text{Col2})$ $\text{Object_Entity_Collection}(\text{Col3})$ $\text{Spatial_Object}(\text{Tower_North})$ $\text{spatial_reference}(\text{Doc2}, \text{Building_West})$ $\text{topical_reference}(\text{Doc2}, \text{Damage})$ $\text{spatial_reference}(\text{Doc4}, \text{Tower_South})$ $\text{topical_reference}(\text{Doc4}, \text{Chemical_Defect})$	$\text{topical_reference}(\text{Doc4}, \text{Wood})$ $\text{spatial_reference}(\text{Col1}, \text{Building_West})$ $\text{topical_reference}(\text{Col1}, \text{Damage})$ $\text{contains_Object}(\text{Col1}, \text{Doc1})$ $\text{contains_Object}(\text{Col1}, \text{Doc2})$ $\text{spatial_reference}(\text{Col2}, \text{Building_West})$ $\text{topical_reference}(\text{Col2}, \text{Damage})$ $\text{contains_Object}(\text{Col2}, \text{Col3})$ $\text{contains_Object}(\text{Col3}, \text{Doc4})$ $\text{topical_reference}(\text{Tower_North}, \text{Crack})$ $\text{spatial_reference}(\text{Ref1}, \text{Middle_Section})$ $\text{topical_reference}(\text{Ref1}, \text{Stone})$
--	---

□

C. Abbildung DL nach OWL-2-DL

Die folgende Tabelle zeigt die Abbildung DL nach OWL-2-DL

DL	OWL-2-DL
ConceptName	Declaration (Class(<ConceptName>)
RoleName	Declaration (ObjectProperty(<RoleName>)
IndividualName	Declaration (NamedIndividual(<IndividualName>)
ConceptName(IndividualName)	ClassAssertion (<ConceptName> <IndividualName>)
RoleName(INameFrom, INameTo)	ObjectPropertyAssertion (<RoleName> <INameFrom> <INameTo>)
$\exists \text{RoleName} . \top \sqsubseteq \text{ConceptName}$	ObjectPropertyDomain (<RoleName> <ConceptName>)
$\top \sqsubseteq \forall \text{RoleName} . \text{ConceptName}$	ObjectPropertyRange (<RoleName> <ConceptName>)
$\text{RoleName} \sqcap \text{RoleName}^{-} \sqsubseteq \perp$	IrreflexiveObjectProperty (<RoleName>)
$\text{RoleName} \equiv \text{RoleName}^{-}$	SymmetricObjectProperty (<RoleName>)
Asy(RoleName)	AsymmetricObjectProperty (<RoleName>)
$\top \sqsubseteq \leq 1 \text{ RoleName}$	FunctionalObjectProperty (<RoleName>)
$\top \sqsubseteq \leq 1 \text{ RoleName}^{-}$	InverseFunctionalObjectProperty (<RoleName>)
$\text{RoleName} \equiv \text{RoleName}^{+}$	TransitiveObjectProperty (<RoleName>)
ConceptName \sqsubseteq SuperConceptName	SubClassOf (<ConceptName> <SuperConceptName>)

Tabelle C.1.: Abbildung DL nach OWL-2-DL

DL	OWL-2-DL
$RoleName \sqsubseteq SuperRoleName$	SubObjectPropertyOf (<RoleName> <SuperRoleName>)
$ConceptName1 \sqcap ConceptName2 \sqsubseteq \perp$	DisjointClasses (<ConceptName1> <ConceptName2>)
$RoleName1 \sqcap RoleName2 \sqsubseteq \perp$	DisjointObjectProperties (<RoleName1> <RoleName2>)
$RoleName1 \equiv RoleName2^{-}$	InverseObjectProperties (<RoleName1> <RoleName2>)
$RoleName1 \circ RoleName2 \sqsubseteq RoleName3$	SubObjectPropertyOf (ObjectPropertyChain (<RoleName1> <RoleName2>) <RoleName3>)

Tabelle C.1.: Abbildung DL nach OWL-2-DL

D. MonQL Grammatik

ANTLR Grammatik - Lexer

Definition D.1 [ANTLR Grammatik - Lexer]

```
lexer grammar MonqlLexer;

SELECT : 'SELECT';
DISTINCT : 'DISTINCT';
LIMIT : 'LIMIT';
FROM : 'FROM';
WHERE : 'WHERE';
ORDERBY : 'ORDER BY';
UNION : 'UNION';
INTERSECT : 'INTERSECT';
DIFFERENCE : 'DIFFERENCE';
SPATIAL : 'SPATIAL';
TYPE : 'TYPE';
TOPICAL : 'TOPICAL';
ASC : 'ASC';
DESC : 'DESC';
IDENT : 'ID';
KEYWORD : 'KEYWORD';
POINT : 'POINT';
RECT : 'RECT';
LR : 'LR';
UL : 'UL';
UP : 'UP';
DOWN : 'DOWN';
AND : 'AND';
OR : 'OR';
NOT : 'NOT';

ATTRIBUTE
    : 'caption'
    | 'description'
    | 'filename'
    | 'filesize'
    | 'author'
    | 'hres'
    | 'vres'
    ;

//
// Digits

fragment HexDigits : HexDigit+ ;
fragment DecDigits : DecDigit+ ;
fragment OctalDigits : OctalDigit+ ;
fragment BinaryDigits : BinaryDigit+ ;
```

D. MonQL Grammatik

```
fragment HexOctet      : HexDigit HexDigit;

fragment HexDigit      : [0 9a fA F]  ;
fragment DecDigit      : [0 9]        ;
fragment OctalDigit    : [0 7]        ;
fragment BinaryDigit   : [01]         ;

//
//
// Whitespace
fragment Hws           : [ \t]         ;
fragment Vws           : '\r'? [\n\f]  ;
WS : ( Hws | Vws )+ > channel(HIDDEN) ;

fragment SQuote       : '\'' ;
fragment Esc           : '\\' ;
fragment UnicodeEsc   : 'u' (HexDigit (HexDigit (HexDigit HexDigit?)?)?)?;
fragment SQuoteLiteral : SQuote ( EscSeq | ~['\r\n\\] )* SQuote ;
fragment EscSeq       : Esc
    : Esc
      ( [bfnr"'\] // The standard escaped character set such as tab, newline,
        etc.
        | UnicodeEsc // A Unicode escape sequence
        | . // Invalid escape character
        | EOF // Incomplete at EOF
      );

INTEGER : DecDigit+;
ID      : [a-zA-Z_] [0 9a-zA-Z_]*;
UUID    : HexOctet HexOctet HexOctet HexOctet ' ' HexOctet HexOctet ' ' HexOctet
    HexOctet ' ' HexOctet HexOctet ' ' HexOctet HexOctet HexOctet HexOctet
    HexOctet;
BOOL    : 'true'|'false';
STRING  : SQuoteLiteral;

fragment ExponentPart : [eE] [+ ]? DecDigits;
fragment FloatTypeSuffix : [fFdD] ;
fragment DecimalFloatingPointLiteral
    : DecDigits '.' DecDigits? ExponentPart? FloatTypeSuffix?
    | '.' DecDigits ExponentPart? FloatTypeSuffix?
    | DecDigits ExponentPart FloatTypeSuffix?
    | DecDigits FloatTypeSuffix
    ;
FLOAT : DecimalFloatingPointLiteral;

LPAREN : '(' ;
RPAREN : ')' ;
COMMA  : ',' ;
STAR   : '*' ;
SLASH  : '/' ;
LT     : '<' ;
GT     : '>' ;
LTE    : '<=' ;
GTE    : '>=' ;
EQ     : '=' ;
NE     : '!=' ;
```

□

ANTLR Grammatik - Parser

Definition D.2 [ANTLR Grammatik - Parser]

```
parser grammar MonqlParser;

options {tokenVocab = MonqlLexer;}

monql
: SELECT WS* (DISTINCT WS*)? WS* (STAR|attributes)
  FROM WS* from WS*
  (WHERE WS* where)?
  (ORDERBY WS* order)?
  (LIMIT WS* INTEGER)?
  EOF
;

order
: ATTRIBUTE WS* (ASC|DESC) (WS* COMMA WS* order)?
;

from
: (spatial|topical|type)
| LPAREN from WS* (UNION|INTERSECT|DIFFERENCE) WS* from RPAREN
;

type
: TYPE WS* SPATIAL WS* (IDENT WS* (INTEGER (SLASH INTEGER)?|ID|UUID)
|KEYWORD WS* STRING
) UP? DOWN?
;

topical
: TOPICAL WS* (IDENT WS* (INTEGER (SLASH INTEGER)?|ID|UUID)
|KEYWORD WS* STRING
) UP? DOWN?
;

spatial
: SPATIAL WS* (IDENT WS* (INTEGER|ID|UUID)
|KEYWORD WS* STRING
|POINT WS* LPAREN FLOAT COMMA FLOAT RPAREN
|RECT WS* UL WS* LPAREN FLOAT COMMA FLOAT RPAREN WS* LR WS*
LPAREN FLOAT COMMA FLOAT RPAREN
) UP? DOWN?
;

where
: comp
| NOT WS* LPAREN where RPAREN
| LPAREN where WS* (AND|OR) WS* where RPAREN
;

comp
: ATTRIBUTE WS* (EQ|NE|LT|LTE|GT|GTE) WS* (INTEGER|FLOAT|STRING|BOOL)
;

attributes
: ATTRIBUTE ( WS* COMMA WS* attributes)?
;
```

□

Literatur

- [AE09] Andreas Abecker und Ludger van Elst. „Ontologies for Knowledge Management“. English. In: *Handbook on Ontologies*. Hrsg. von Steffen Staab und Rudi Studer. International Handbooks on Information Systems. Springer Berlin Heidelberg, 2009, S. 713–734. ISBN: 978-3-540-70999-2. DOI: 10.1007/978-3-540-92673-3_32. URL: http://www.ebook.de/de/product/12471152/handbook_on_ontologies.html (siehe S. 53).
- [AHS09] Ahmed Alasoud, Volker Haarslev und Nematollaah Shiri. „An empirical comparison of ontology matching techniques“. In: *Journal of Information Science* 35.4 (2009), S. 379–397. DOI: 10.1177/0165551508100383. URL: <https://doi.org/10.1177/0165551508100383> (siehe S. 120).
- [Aki+11] Nazihah Md Akim u. a. „Spreading activation for web scale reasoning: Promise and problems“. In: *Proceedings of the ACM WebSci11*. Koblenz, Germany, Juni 2011, S. 1–4 (siehe S. i, iii, 127).
- [Baa+07] Franz Baader u. a., Hrsg. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2007 (siehe S. 55).
- [Bag09] Sikha Bagui. „Mapping OWL to the Entity Relationship and Extended Entity Relationship models“. In: *I. J. Knowledge and Web Intelligence* 1.1/2 (2009), S. 125–149. DOI: 10.1504/IJKWI.2009.027929. URL: <https://doi.org/10.1504/IJKWI.2009.027929> (siehe S. 54).
- [BH08] Jürgen Bock und Jan Hettenhausen. „MapPSO Results for OAEI 2008“. In: *3rd International Workshop on Ontology Matching (OM-2008) Collocated with the 7th International Semantic Web Conference (ISWC-2008)*. Karlsruhe, Germany, 26. Okt. 2008. URL: http://ceur-ws.org/Vol-431/oeai08_paper8.pdf (siehe S. 120).
- [BHS09] Franz Baader, Ian Horrocks und Ulrike Sattler. „Description Logics“. In: *Handbook on Ontologies*. Hrsg. von Steffen Staab und Rudi Studer. Springer Berlin Heidelberg, 2009, S. 21–43. ISBN: 978-3-540-92673-3. DOI: 10.1007/978-3-540-92673-3_1. URL: http://www.ebook.de/de/product/12471152/handbook_on_ontologies.html (siehe S. 55).
- [BM14] Dan Brickley und Libby Miller. *FOAF Vocabulary Specification 0.99*. 2014. URL: <http://xmlns.com/foaf/spec/> (siehe S. 68).
- [Boa12] DCMI Usage Board. *DCMI Metadata Terms*. <http://dublincore.org>. Juni 2012. URL: <http://dublincore.org/documents/dcmi-type-vocabulary/> (siehe S. 68).

- [BR08] André Borrmann und Ernst Rank. „Topological operators in a 3D spatial query language for building information models“. In: *12th Int. Conf. on Computing in Civil and Building Engineering (ICCCBE)*. 2008 (siehe S. 60).
- [BR09] André Borrmann und Ernst Rank. „Topological analysis of 3D building models using a spatial query language“. In: *Advanced Engineering Informatics* 23.4 (2009), S. 370–385 (siehe S. 60).
- [BR10] Ricardo Baeza-Yates und Berthier Ribeiro-Neto. *Modern Information Retrieval*. 2. Aufl. Addison-Wesley, 2010. ISBN: 978-0321416919 (siehe S. 175, 176, 178).
- [Car07] Jorge Cardoso. „The Semantic Web Vision: Where Are We?“ In: *Intelligent Systems, IEEE* 22.5 (Sep. 2007), S. 84–88. ISSN: 1541-1672. DOI: 10.1109/MIS.2007.4338499 (siehe S. 54).
- [CAS09] Isabel F. Cruz, Flavio Palandri Antonelli und Cosmin Stroe. „AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies“. In: *PVLDB* 2.2 (2009), S. 1586–1589. URL: <http://www.vldb.org/pvldb/2/vldb09-1003.pdf> (siehe S. 120).
- [CFO93] Eliseo Clementini, Paolino Felice und Peter van Oosterom. „A small set of formal topological relationships suitable for end-user interaction“. English. In: *Advances in Spatial Databases*. Hrsg. von David Abel und Beng Chin Ooi. Bd. 692. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1993, S. 277–295. ISBN: 978-3-540-56869-8. DOI: 10.1007/3-540-56869-7_16 (siehe S. 40).
- [Chi92] Nancy Chinchor. „MUC-4 evaluation metrics“. In: *Proceedings of the 4th Conference on Message Understanding, MUC 1992, McLean, Virginia, USA, June 16-18, 1992*. 1992, S. 22–29. DOI: 10.1145/1072064.1072067. URL: <http://doi.acm.org/10.1145/1072064.1072067> (siehe S. 179).
- [CL75] Allan M. Collins und Elizabeth F. Loftus. „A spreading-activation theory of semantic processing.“ In: *Psychological Review* 82.6 (1975), S. 407–428. ISSN: 0033-295X. URL: <http://search.ebscohost.com/login.aspx?direct=true&db=pdh&AN=rev-82-6-407&loginpage>Login.asp&site=ehost-live> (siehe S. 125).
- [Com10] European Commission. *Digitale Agenda: Europeana ermöglicht Online-Zugriff auf über 14 Millionen Beispiele des europäischen Kulturerbes*. 18. Okt. 2010. URL: http://europa.eu/rapid/press-release_IP-10-1524_de.htm?locale=en (besucht am 08.07.2017) (siehe S. 1).
- [CSE94] Eliseo Clementini, Jayant Sharma und Max J. Egenhofer. „Modelling topological spatial relations: Strategies for query processing“. In: *Computers & Graphics* 18.6 (1994), S. 815–822. ISSN: 0097-8493. DOI: 10.1016/0097-8493(94)90007-8 (siehe S. 40).
- [CSH06] Namyoun Choi, Il-Yeol Song und Hyoil Han. „A survey on ontology mapping“. In: *SIGMOD Record (ACM Special Interest Group on Management of Data)* 35.3 (Sep. 2006), S. 34–41. ISSN: 0163-5808 (siehe S. 118, 119).

-
- [Dec+98] Stefan Decker u. a. „Ontobroker: Ontology based Access to Distributed and Semi-Structured Information“. In: *Database Semantics: Semantic Issues in Multimedia Systems*. Kluwer Academic Publisher, 1998, S. 351–369 (siehe S. 53).
- [DFS11] Rainer Drewello, Burkhard Freitag und Christoph Schlieder. „New Tools for Ancient Ruins“. In: *DFG magazine german research* 33 (1) (2011), S. 24–28 (siehe S. 22).
- [Dix+10] Alan J. Dix u. a. „Spreading Activation over Ontology-Based Resources: from Personal Context to Web Scale Reasoning“. In: *Int. J. Semantic Computing* 4.1 (2010), S. 59–102. DOI: 10.1142/S1793351X10000973. URL: <https://doi.org/10.1142/S1793351X10000973> (siehe S. 126).
- [DMQ03] Dejing Dou, Drew V. McDermott und Peishen Qi. „Ontology Translation on the Semantic Web.“ In: *CoopIS/DOA/ODBASE*. Hrsg. von Robert Meersman, Zahir Tari und Douglas C. Schmidt. Bd. 2888. Lecture Notes in Computer Science. Springer, 2003, S. 952–969. ISBN: 3-540-20498-9. URL: <http://dblp.uni-trier.de/db/conf/coopis/coopis2003.html#DouMQ03> (siehe S. 119).
- [DT14] Michael Dodt und Martino La Torre. „Neue Untersuchungen und Restaurierungen an den Trierer Kaiserthermen“. In: *Bericht über die 48. Tagung für Ausgrabungswissenschaft und Bauforschung*. Erfurt, 2014, S. 197–206 (siehe S. 22).
- [EF91] Max J. Egenhofer und Robert D. Franzosa. „Point-set topological spatial relations“. In: *International Journal of Geographical Information Systems* 5.2 (Jan. 1991), S. 161–174. DOI: 10.1080/02693799108927841. eprint: <http://dx.doi.org/10.1080/02693799108927841> (siehe S. 40).
- [EH90] M.J Egenhofer und J.R. Herring. *A mathematical framework for the definition of topological relationships*. 1990 (siehe S. 40).
- [ERW11] Shady Elbassuoni, Maya Ramanath und Gerhard Weikum. „Query Relaxation for Entity-Relationship Search“. In: *The Semantic Web: Research and Applications - 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece, May 29 - June 2, 2011, Proceedings, Part II*. 2011, S. 62–76. DOI: 10.1007/978-3-642-21064-8_5. URL: https://doi.org/10.1007/978-3-642-21064-8_5 (siehe S. 104).
- [ES13] Jérôme Euzenat und Pavel Shvaiko. *Ontology Matching*. Springer Berlin Heidelberg, 8. Nov. 2013. 511 S. ISBN: 978-3-642-38721-0. URL: http://www.ebook.de/de/product/21989831/jerome_euzenat_pavel_shvaiko_ontology_matching.html (siehe S. 115–117).

- [Far+13] Daniel Faria u. a. „The AgreementMakerLight Ontology Matching System“. In: *On the Move to Meaningful Internet Systems: OTM 2013 Conferences - Confederated International Conferences: CoopIS, DOA-Trusted Cloud, and ODBASE 2013* (9. Sep. 2013). Graz, Austria, 2013, S. 527–541. DOI: 10.1007/978-3-642-41030-7_38. URL: https://doi.org/10.1007/978-3-642-41030-7_38 (siehe S. 120).
- [Fou17] Europeana Foundation. *europeana collections - The Current Europeana Dataset*. 2017. URL: <http://www.europeana.eu/portal/de> (besucht am 08.07.2017) (siehe S. 1).
- [Fre16] Burkhard Freitag. *MonArch - Monumentalbauwerke Archivsystem*. Hrsg. von Universität Passau. Lehrstuhl für Informatik mit Schwerpunkt Informationsmanagement. 26. Juli 2016. URL: <http://www.monarch-project.eu> (besucht am 08.07.2017) (siehe S. i, iii, 1, 9).
- [FS09] Burkhard Freitag und Christoph Schlieder. „MonArch - Digital Archives for Monumental Buildings“. In: *Künstliche Intelligenz* 4 (2009), S. 30–35 (siehe S. 9).
- [FS17] Burkhard Freitag und Alexander Stenzer. „MonArch – A Digital Archive for Cultural Heritage“. Tagungsdokumentation der Jahrestagung 2016 - Das Digitale und die Denkmalpflege. 2017 (siehe S. i, iii, 1).
- [GM08] Jorge Gracia und Eduardo Mena. „Ontology Matching with CIDER: Evaluation Report for the OAEI 2008“. In: *Proceedings of the 3rd International Workshop on Ontology Matching (OM-2008) Collocated with the 7th International Semantic Web Conference (ISWC-2008), Karlsruhe, Germany, October 26, 2008*. 2008. URL: http://ceur-ws.org/Vol-431/oaei08_paper4.pdf (siehe S. 120).
- [GO13] Birte Glimm und Chimezie Ogbuji. *SPARQL 1.1 Entailment Regimes W3C Recommendation 21 March 2013*. W3C Recommendation. März 2013. URL: <https://www.w3.org/TR/sparql11-entailment/> (besucht am 30.05.2017) (siehe S. 96, 97).
- [Ham+08] Fayçal Hamdi u. a. „TaxoMap in the OAEI 2008 Alignment Contest“. In: *Proceedings of the 3rd International Workshop on Ontology Matching (OM-2008) Collocated with the 7th International Semantic Web Conference (ISWC-2008), Karlsruhe, Germany, October 26, 2008*. 2008. URL: http://ceur-ws.org/Vol-431/oaei08_paper12.pdf (siehe S. 120).
- [Hel11] Markus Helm. „Semantic Search - Semantische Suche auf taxonomisch strukturierten Daten im verteilten Archivsystem MonArch“. Masterarbeit. Lehrstuhl für Informationsmanagement, 21. Sep. 2011 (siehe S. 175, 181–183).
- [Hen08] Andreas Henrich. *Information Retrieval 1 - Grundlagen, Modelle und Anwendungen*. Bamberg: Otto-Friedrich-Universität Bamberg, 2008. URL: <http://www.uni-bamberg.de/minf/IR1-Buch> (siehe S. 176, 178, 179).

-
- [Hen17] Jana Hentschke. *LINKED-DATA-SERVICE DER DEUTSCHEN NATIONALBIBLIOTHEK*. 12. Jan. 2017. URL: http://www.dnb.de/DE/Service/DigitaleDienste/LinkedData/linkedata_node.html (besucht am 23.07.2017) (siehe S. 61).
- [Her10] John R. Herring. *OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 2: SQL option*. Open Geospatial Consortium Inc. Version: 1.2.1. Aug. 2010 (siehe S. 40, 85).
- [Hit+12] Pascal Hitzler u. a., Hrsg. *OWL 2 Web Ontology Language Primer (Second Edition)*. W3C Recommendation 11 December 2012. W3C. 11. Dez. 2012. URL: <https://www.w3.org/TR/owl2-primer/> (besucht am 31.05.2017) (siehe S. 64).
- [HKS06] Ian Horrocks, Oliver Kutz und Ulrike Sattler. „The Even More Irresistible SROIQ“. In: *Kr* 6 (2006), S. 57–67 (siehe S. 54, 64).
- [Hor05] Ian Horrocks. „OWL: A Description Logic Based Ontology Language“. English. In: *Logic Programming*. Hrsg. von Maurizio Gabbrielli und Gopal Gupta. Bd. 3668. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, S. 1–4. ISBN: 978-3-540-29208-1. DOI: 10.1007/11562931_1 (siehe S. 64).
- [HPH03] Ian Horrocks, Peter F. Patel-Schneider und Frank van Harmelen. „From SHIQ and RDF to OWL: the making of a Web Ontology Language“. In: *Web Semantics: Science, Services and Agents on the World Wide Web* 1.1 (2003), S. 7–26. ISSN: 1570-8268. DOI: 10.1016/j.websem.2003.07.001. URL: <http://www.sciencedirect.com/science/article/pii/S1570826803000027> (siehe S. 54, 64).
- [HQC08] Wei Hu, Yuzhong Qu und Gong Cheng. „Matching large ontologies: A divide-and-conquer approach“. In: *Data and Knowledge Engineering* 67.1 (2008), S. 140–160. DOI: 10.1016/j.datak.2008.06.003. URL: <https://doi.org/10.1016/j.datak.2008.06.003> (siehe S. 119).
- [HS04] Ian Horrocks und Ulrike Sattler. „Decidability of \mathcal{SHIQ} with complex role inclusion axioms“. In: *Artificial Intelligence* 160.1–2 (2004), S. 79–104. DOI: 10.1016/j.artint.2004.06.002 (siehe S. 55).
- [IS09] Antoine Isaac und Ed Summers. *SKOS Simple Knowledge Organization System Primer*. W3C Working Group Note. Aug. 2009. URL: <https://www.w3.org/TR/skos-primer/> (siehe S. 47, 61).
- [ISO02] ISO. *Information technology – Database languages – SQL Multimedia and Application Packages – Part 3: Spatial*. ISO/IEC 13249-3:2002 FDIS. 2nd edition. 2002 (siehe S. 85).
- [ISO13] ISO. *Industry Foundation Classes (IFC). ISO 16739:2013*. International Organization for Standardization. Apr. 2013. URL: <https://www.buildingsmart.de/bim-knowhow/standards> (besucht am 12.07.2017) (siehe S. 42).
-

- [JSK09] Yves R. Jean-Mary, E. Patrick Shironoshita und Mansur R. Kabuka. „Ontology matching with semantic verification“. In: *Web Semantics: Science, Services and Agents on the World Wide Web* 7.3 (2009), S. 235–251. DOI: 10.1016/j.websem.2009.04.001. URL: <https://doi.org/10.1016/j.websem.2009.04.001> (siehe S. 120).
- [KB07] J. Kunze und T. Baker. *The Dublin Core Metadata Element Set*. Aug. 2007. DOI: 10.17487/rfc5013. URL: <https://www.rfc-editor.org/info/rfc5013> (siehe S. 40, 68).
- [KDS17] Lorenz Korn, Rainer Drewello und Manfred Schuller. *Monuments and Context in Bukhara*. 2017. URL: www.uni-bamberg.de/en/buchara-projekt/ (besucht am 15. 07. 2017) (siehe S. 22).
- [Ken+07] David Kensche u. a. „GeRoMeSuite: A System for Holistic Generic Model Management“. In: *Proceedings of the 33rd International Conference on Very Large Data Bases, University of Vienna, Austria, September 23-27, 2007*. 2007, S. 1322–1325. URL: <http://www.vldb.org/conf/2007/papers/demo/p1322-kensche.pdf> (siehe S. 120).
- [KS03] Yannis Kalfoglou und Marco Schorlemmer. „Ontology Mapping: The State Of The Art“. In: *Knowledge Engineering Review* 18.1 (2003), S. 1–31. URL: <http://citeseer.ist.psu.edu/kalfoglou03ontology.html> (siehe S. 115, 118, 119).
- [LDG04] Bach Thanh Le, Rose Dieng-Kuntz und Fabien Gandon. „On Ontology Matching Problems - for Building a Corporate Semantic Web in a Multi-Communities Organization“. In: *ICEIS 2004, Proceedings of the 6th International Conference on Enterprise Information Systems, Porto, Portugal, April 14-17, 2004*. 2004, S. 236–243 (siehe S. 120).
- [Li+09] Juanzi Li u. a. „RiMOM: A Dynamic Multistrategy Ontology Alignment Framework“. In: *IEEE Transactions on Knowledge and Data Engineering* 21.8 (2009), S. 1218–1232. DOI: 10.1109/TKDE.2008.202. URL: <https://doi.org/10.1109/TKDE.2008.202> (siehe S. 120).
- [LT06] Patrick Lambrix und He Tan. „SAMBO - A system for aligning and merging biomedical ontologies“. In: *Web Semantics: Science, Services and Agents on the World Wide Web*. Bd. 4. 3. 2006, S. 196–206. DOI: 10.1016/j.websem.2006.05.003. URL: <https://doi.org/10.1016/j.websem.2006.05.003> (siehe S. 119).
- [LW14] Judith Ley und Marc Wietheger. *Palace Research in Aachen*. 11. Dez. 2014. URL: <http://arch.rwth-aachen.de/cms/Architektur/Forschung/Verbundforschung/Cultural-Heritage/~cbjv/Pfalzenforschung-in-Aachen/?lidx=1> (siehe S. 22).

-
- [MA04] Peter Mika und Hans Akkermans. „Towards a new synthesis of ontology technology and knowledge management“. In: *The Knowledge Engineering Review* 19.4 (2004), S. 317–345. DOI: 10.1017/S0269888905000305. URL: <https://doi.org/10.1017/S0269888905000305> (siehe S. 120).
- [Mae+02] Alexander Maedche u. a. „MAFRA - A MAPPING FRAMework for Distributed Ontologies.“ In: *EKAW*. Hrsg. von Asunción Gómez-Pérez und V. Richard Benjamins. Bd. 2473. Lecture Notes in Computer Science. Springer, 2002, S. 235–250. ISBN: 3-540-44268-5. URL: <http://dblp.uni-trier.de/db/conf/ekaw/ekaw2002.html#MaedcheMSV02> (siehe S. 118).
- [Mar+14] Nicolas Marie u. a. „Exploratory search on topics through different perspectives with DBpedia“. In: *Proceedings of the 10th International Conference on Semantic Systems, SEMANTICS 2014, Leipzig, Germany, September 4-5, 2014*. 2014, S. 45–52. DOI: 10.1145/2660517.2660518. URL: <http://doi.acm.org/10.1145/2660517.2660518> (siehe S. 126).
- [Mat12] John Herring Matthew Perry. *GeoSPARQL - A Geographic Query Language for RDF Data*. Open Geospatial Consortium. Sep. 2012. URL: <http://www.opengeospatial.org/standards/geosparql> (siehe S. 100).
- [Mcg+00] D. L. McGuinness u. a. „An Environment for Merging and Testing Large Ontologies“. In: *17th Intl. Conf. on Principles of Knowledge Representation and Reasoning (KR'2000)*. Colorado, USA, Apr. 2000, S. 483–493 (siehe S. 118).
- [MGR02] *Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching*. 2002, S. 117–128. DOI: 10.1109/ICDE.2002.994702. URL: <https://doi.org/10.1109/ICDE.2002.994702> (siehe S. 126).
- [MM09] I. Myroshnichenko und M.C. Murphy. „Mapping ER Schemas to OWL Ontologies“. In: *Semantic Computing*. Sep. 2009, S. 324–329. DOI: 10.1109/ICSC.2009.61 (siehe S. 54).
- [MPG12] Boris Motik, Peter F. Patel-Schneider und Bernardo Cuenca Grau, Hrsg. *OWL 2 Web Ontology Language Direct Semantics (Second Edition)*. W3C Recommendation 11 December 2012. W3C. 11. Dez. 2012. URL: <https://www.w3.org/TR/owl2-direct-semantics/> (besucht am 31.05.2017) (siehe S. 54, 64).
- [MRS08] Christopher D. Manning, Prabhakar Raghavan und Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press, 2008. ISBN: 978-0-521-86571-5. URL: <http://nlp.stanford.edu/IR-book/information-retrieval-book.html> (siehe S. 179).
- [MS00] Alexander Maedche und Stefan Staab. „Semi-Automatic Engineering of Ontologies from Text“. In: *Proceedings of the 12th International Conference on Software and Knowledge Engineering* (5. Juli 2000). Chicago, USA: KSI, 2000 (siehe S. 118).

- [MT14] Davide Martinenghi und Riccardo Torlone. „Taxonomy-based relaxation of query answering in relational databases“. In: *VLDB J.* 23.5 (2014), S. 747–769. DOI: 10.1007/s00778-013-0350-x. URL: <https://doi.org/10.1007/s00778-013-0350-x> (siehe S. 104, 138).
- [Mus15] M.A Musen. „The Protégé project: A look back and a look forward.“ In: *Association of Computing Machinery Specific Interest Group in Artificial Intelligence*. AI Matters. Bd. 1(4). Stanford Center for Biomedical Informatics Research. Juni 2015. DOI: 10.1145/2557001.25757003.. URL: <http://protege.stanford.edu/products.php#desktop-protege> (siehe S. 69).
- [Ngu16] Thi Thuy Anh Nguyen. „Ontology Matching based on Combination of Lexical and Structural Techniques in Semantic Web“. Diss. Düsseldorf, Heinrich-Heine-Universität Düsseldorf, 2016 (siehe S. 116, 120).
- [NM00] N. F. Noy und M. A. Musen. „PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment“. In: *Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*. Austin, Texas, 2000 (siehe S. 118).
- [NM01] N. F. Noy und M. A. Musen. „Anchor-PROMPT: Using Non-Local Context for Semantic Matching“. In: *Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001)*. Seattle, WA, 2001 (siehe S. 118).
- [NM02] Natalya Fridman Noy und Mark A. Musen. „Evaluating Ontology-Mapping Tools: Requirements and Experience.“ In: *Ontoweb-Sig3 Workshop at the 13th International Conference on Knowledge Engineering and Knowledge Management*. 2002, S. 1–14 (siehe S. 117).
- [Noo+16] Farhad Nooralahzadeh u. a. „Adapting Semantic Spreading Activation to Entity Linking in Text“. In: *Natural Language Processing and Information Systems - 21st International Conference on Applications of Natural Language to Information Systems, NLDB 2016, Salford, UK, June 22-24, 2016, Proceedings*. 2016, S. 74–90. DOI: 10.1007/978-3-319-41754-7_7. URL: https://doi.org/10.1007/978-3-319-41754-7_7 (siehe S. 126).
- [Noy04] Natalya F. Noy. „Semantic integration: a survey of ontology-based approaches“. In: *SIGMOD Record (ACM Special Interest Group on Management of Data)* 33.4 (Dez. 2004), S. 65–70. ISSN: 0163-5808 (siehe S. 115).
- [NV10] Miklos Nagy und Maria Vargas-Vera. „Towards an Automatic Semantic Data Integration: Multi-agent Framework Approach, Semantic Web“. In: *Semantic Web*. Hrsg. von Gang Wu. InTech, 2010. DOI: 10.5772/7310. URL: <https://www.intechopen.com/books/semantic-web/towards-an-automatic-semantic-data-integration-multi-agent-framework-approach> (siehe S. 120).

-
- [NVS09] Miklos Nagy, Maria Vargas-Vera und Piotr Stolarski. „DSSim Results for OAEI 2009“. In: *Proceedings of the 4th International Workshop on Ontology Matching (OM-2009) collocated with the 8th International Semantic Web Conference (ISWC-2009) Chantilly, USA, October 25, 2009*. 2009. URL: http://ceur-ws.org/Vol-551/oaiei09_paper5.pdf (siehe S. 120).
- [ÖV11] M. Tamer Özsu und Patrick Valduriez. *Principles of Distributed Database Systems, Third Edition*. Springer, 2011. ISBN: 978-1-4419-8833-1. DOI: 10.1007/978-1-4419-8834-8 (siehe S. 5).
- [Pag+99] Lawrence Page u. a. *The PageRank citation ranking: Bringing order to the web*. Techn. Ber. Stanford InfoLab, 1999 (siehe S. 126).
- [Par12] Terence Parr. *ANTLR v3*. 2012. URL: <http://www.antlr3.org/> (besucht am 25.07.2017) (siehe S. 74).
- [Pel17] Ray Pelletier, Hrsg. *Request for Comments (RFC)*. IETF Administrative Director. 2017. URL: <https://www.ietf.org/rfc.html> (besucht am 08.07.2017) (siehe S. 40).
- [PF11] Terence Parr und Kathleen Fisher. „LL(*): the foundation of the ANTLR parser generator“. In: *Proceedings of the 32nd ACM SIGPLAN Conference on Programming Language Design and Implementation* (4. Juni 2011). San Jose, CA, USA, 2011, S. 425–436. DOI: 10.1145/1993498.1993548. URL: <http://doi.acm.org/10.1145/1993498.1993548> (siehe S. 74).
- [RB01] Erhard Rahm und Philip A. Bernstein. „A Survey of Approaches to Automatic Schema Matching“. In: *The International Journal on Very Large Data Bases*. 2001. DOI: 10.1007/s007780100057 (siehe S. 116).
- [Rei09] Simon Reinhardt. *Dublin Core in OWL 2*. http://bloody-byte.net/rdf/dc_owl2dl/. Sep. 2009. URL: http://bloody-byte.net/rdf/dc_owl2dl/ (siehe S. 68).
- [RSA04] Cristiano Rocha, Daniel Schwabe und Marcus Poggi de Aragao. „A Hybrid Approach for Searching in the Semantic Web“. In: (Jan. 2004). URL: <http://wwwconf.ecs.soton.ac.uk/archive/00000589/01/p374-rocha.pdf> (siehe S. 125–127).
- [SA09] Md. Hanif Seddiqui und Masaki Aono. „An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size“. In: *Web Semantics: Science, Services and Agents on the World Wide Web 7.4* (2009). Semantic Web challenge 2008, S. 344–356. ISSN: 1570-8268. DOI: 10.1016/j.websem.2009.09.001. URL: <http://www.sciencedirect.com/science/article/pii/S1570826809000432> (siehe S. 120).
- [SE05] Pavel Shvaiko und Jérôme Euzenat. „A Survey of Schema-Based Matching Approaches“. In: (2005), S. 146–171. DOI: 10.1007/11603412_5. URL: https://doi.org/10.1007/11603412_5 (siehe S. 114, 115).

- [SE13] P. Shvaiko und J. Euzenat. „Ontology Matching: State of the Art and Future Challenges“. In: *Knowledge and Data Engineering, IEEE Transactions on* 25.1 (Jan. 2013), S. 158–176. ISSN: 1041-4347. DOI: 10.1109/TKDE.2011.253 (siehe S. 54, 114, 115, 118–120).
- [SES17] Alexander Stenzer, Christina Ehrlinger und Matthias Schmid. „Ansätze zur semantischen 3D-Repräsentation von Bauwerken in Datenbanken. Der Modelle Tugend 2.0“. 11. Juli 2017 (siehe S. 16).
- [SF12] Alexander Stenzer und Burkhard Freitag. „An approach to query relaxation using ontologies in a GIS-based archiving system“. In: *ACM SIGSPATIAL International Workshop on GeoStreaming (IWGS)*. Redondo Beach, California, USA, Nov. 2012 (siehe S. 160).
- [Sir+07] Evren Sirin u. a. „Pellet: A practical OWL-DL reasoner“. In: *J. Web Sem.* 5.2 (2007), S. 51–53. DOI: 10.1016/j.websem.2007.03.004. URL: <https://doi.org/10.1016/j.websem.2007.03.004> (siehe S. 69).
- [Sir17] Evren Sirin. *Pellet is an OWL 2 reasoner in Java*. 2017. URL: <https://github.com/stardog-union/pellet/wiki/FAQ> (besucht am 23.07.2017) (siehe S. 69).
- [SLR14] Serwah Sabetghadam, Mihai Lupu und Andreas Rauber. „Which One to Choose: Random Walks or Spreading Activation?“ In: *Multidisciplinary Information Retrieval - 7th Information Retrieval Facility Conference, IRFC 2014, Copenhagen, Denmark, November 10-12, 2014, Proceedings*. 2014, S. 112–119. DOI: 10.1007/978-3-319-12979-2_11. URL: https://doi.org/10.1007/978-3-319-12979-2_11 (siehe S. 125, 126).
- [SM01] G. Stumme und A. Maedche. „FCA-Merge: Bottom-Up Merging of Ontologies“. In: *17th Intl. Conf. on Artificial Intelligence (IJCAI '01)*. Hrsg. von B. Nebel. Seattle, WA, USA, 2001, S. 225–230. URL: <http://www.kde.cs.uni-kassel.de/stumme/papers/2001/IJCAI01.pdf> (siehe S. 119).
- [Sow99] John F. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Pacific Grove, CA: Brooks Cole Publishing Co., 1999, S. 594 (siehe S. 53).
- [SSF10] Tobias Seppenhauser, Alexander Stenzer und Burkhard Freitag. „Retrieving Cultural Heritage Information with Google Earth“. In: *Proceedings of the Twelfth Asia-Pacific Web Conference (APWeb 2010) on Advances in Web Technologies and Applications*. Busan, Korea: IEEE Computer Society Press, Apr. 2010. DOI: <http://www.computer.org/portal/web/csdl/doi/10.1109/APWeb.2010.29> (siehe S. 11).

-
- [SSS08] Kinga Schumacher, Michael Sintek und Leo Sauermann. „Combining Metadata and Document Search with Spreading Activation for Semantic Desktop Search“. In: *ESWC 2008: The Semantic Web: Research and Applications*. Hrsg. von Sean Bechhofer u. a. Springer, Juni 2008, S. 569–583. DOI: 10.1007/978-3-540-68234-9_42. URL: <http://www.springerlink.com/content/a6683407861281q2/> (siehe S. 125–127).
- [Sto03] Knut Stolze. „SQL/MM Spatial-The Standard to Manage Spatial Data in a Relational Database System.“ In: *BTW*. Bd. 2003. 2003, S. 247–264 (siehe S. 85).
- [Str07] Christian Strobl. „Dimensionally Extended Nine-Intersection Model (DE-9IM)“. In: *Encyclopedia of GIS (Springer Reference)*. Springer, 2007, S. 240–245. ISBN: 978-0-387-30858-6 (siehe S. 41).
- [Sun+13] Tao Sun u. a. „Personalized search in digital libraries via spreading activation model“. In: *Web Intelligence and Agent Systems 11.2* (2013), S. 137–147. DOI: 10.3233/WIA-130267 (siehe S. 141).
- [Sun+17] Shengtao Sun u. a. „Associative retrieval in spatial big data based on spreading activation with semantic ontology“. In: *Future Generation Comp. Syst.* 76 (2017), S. 499–509 (siehe S. 160).
- [SWF11a] Alexander Stenzer, Claudia Woller und Burkhard Freitag. „Digital Archives for Cultural Heritage“. In: *EU-CHIC Olimia Workshop: Proceedings of the 2th workshop*. Olimia, Slovenia, Mai 2011, S. 36–41 (siehe S. 10).
- [SWF11b] Alexander Stenzer, Claudia Woller und Burkhard Freitag. „MonArch: Digital Archives for Cultural Heritage“. In: *iiWAS 2011: Proceedings of the 13th International Conference on Information Integration and Web-based Applications & Services*. Ho Chi Minh City, Vietnam: ACM International Conference Proceedings, Dez. 2011 (siehe S. 17).
- [Tay10] Mohammad Mustafa Taye. „State-of-the-Art: Ontology Matching Techniques and Ontology Mapping Systems“. In: *International Journal of ACM Jordan 3* (2010), S. 8. ISSN: 2078-7952 (siehe S. 114, 115, 118).
- [UK05] Sujatha R. Upadhyaya und P. Sreenivasa Kumar. „ERONTO: a tool for extracting ontologies from extended E/R diagrams“. In: *Proceedings of the 2005 ACM Symposium on Applied Computing (SAC)* (13. März 2005). Santa Fe, New Mexico, USA, 2005, S. 666–670. DOI: 10.1145/1066677.1066828. URL: <http://doi.acm.org/10.1145/1066677.1066828> (siehe S. 54).
- [VN06] Ernestas Vysniauskas und Lina Nemuraite. „Transforming Ontology Representation from OWL to Relational Database“. In: *Information Technology and Control*. Bd. 35. 3A. 2006, S. 333–343 (siehe S. 54).

- [WX08] Peng Wang und Baowen Xu. „Lily: Ontology Alignment Results for OAEI 2008“. In: *Proceedings of the 3rd International Workshop on Ontology Matching (OM-2008) Collocated with the 7th International Semantic Web Conference (ISWC-2008), Karlsruhe, Germany, October 26, 2008*, 2008. URL: http://ceur-ws.org/Vol-431/oaei08_paper7.pdf (siehe S. 120).

Abbildungsverzeichnis

2.1. MonArch-Arbeitsschritte bei der Erschließung von Archivalien	9
2.2. Zusammenhang zwischen Zerlegung und Seitenansicht	11
2.3. Themenhierarchie für die thematische Auszeichnung	13
2.4. Kartierung der Westfassade von St. Lorenz in Nürnberg mit räumlichen und thematischen Metadaten	14
2.5. Oberfläche der MonArch-Anwendung	15
2.6. MonArch-Arbeitsschritte bei der Erstellung der digitalen Repräsentation des Gebäudes (entnommen aus [SES17])	16
2.7. ER-Modell des MonArch-Datenmodells	17
2.8. ER-Modell des MonArch-Datenmodells (räumlicher Teil)	18
2.9. ER-Modell des MonArch-Datenmodells (thematischer Teil)	19
2.10. ER-Modell des MonArch-Datenmodells (Dokumente)	20
2.11. Metaebenen im MonArch-Datenmodell	21
2.12. Ausschnitt der ersten drei Ebenen der Strukturhierarchie von St. Lorenz und St. Sebald	23
2.13. Ausschnitt der ersten Themenebene von St. Lorenz und St. Sebald	24
2.14. Vier Dokumente verortet an der Westfassade von St. Lorenz	26
2.15. Räumliche Anfrage Variante eins an der Westfassade von St. Lorenz	27
2.16. Räumliche Anfrage Variante zwei an der Westfassade von St. Lorenz	28
2.17. Räumliche Anfrage Variante drei an der Westfassade von St. Lorenz	29
2.18. Räumliche Anfrage am Nordturm von St. Lorenz durch grafische Selektion	30
2.19. Vier Dokumente ausgezeichnet mit Themen von St. Lorenz	31
2.20. Thematische Anfrage Variante eins an St. Lorenz	32
2.21. Thematische Anfrage Variante zwei an St. Lorenz	33
2.22. MonArch-Client mit Anfrage St. Lorenz	35
2.23. MonArch-Verbund	36
3.1. Erweitertes Datenmodell als ER-Modell	39
3.2. Erweitertes Datenmodell (räumlich) als ER-Modell	41
3.3. Ausschnitt Strukturhierarchie typisiert - St. Lorenz	43
3.4. Ausschnitt Strukturhierarchie typisiert - St. Lorenz	44
3.5. Ausschnitt Strukturhierarchie und grafische Repräsentation - St. Lorenz	45
3.6. Erweitertes Datenmodell (thematisch) als ER-Modell	46
3.7. Ausschnitt Themenhierarchie - St. Lorenz	47
3.8. Erweitertes Datenmodell (objektzentriert) als ER-Modell	48
3.9. Zusammenfassung von Objekten zu Sammlungen	51

3.10. Thematische Auszeichnung von Gebäudeteilen	52
3.11. Konzepte und Rollen der Raum- und Typontologie	57
3.12. Typ und transitive Typen eines Gebäudeteils	59
3.13. Rollen-Hierarchie der geometrischen räumlichen Rollen	60
3.14. Konzepte und Rollen der Themaontologie	60
3.15. Konzepte und Rollen der Objektontologie	63
3.16. Zusammenspiel der Ontologien	69
4.1. Räumliche Anfrage Variante vier am Nordturm von St. Lorenz	85
4.2. Konvertierung nach SQL oder SPARQL	96
4.3. Konzepte, Individuen und Rollenaussagen der $O_{instance}^{object}(StLorenz)$	97
5.1. Ontology-Alignment	115
5.2. Ontology-Merging	115
5.3. Ontology-Matching-Verfahren (angelehnt und inhaltlich entnommen aus [ES13])	116
5.4. Spreading-Activation	129
5.5. Themengraph von St. Lorenz	137
5.6. Themengraph von St. Sebald	139
5.7. Spreading-Activation-Graph für den Themengraph von St. Sebald	142
5.8. Themengraph von St. Sebald mit Aktivierungswerten	147
5.9. Material-Themengraph mit Aktivierungswerten - Anfrage <i>Sandstein</i>	150
5.10. Material-Themengraph mit Aktivierungswerten - Anfrage <i>Naturstein</i>	151
5.11. ICONCLASS-Themengraph mit Aktivierungswerten - Anfrage <i>Georg</i>	152
5.12. Strukturgraph von St. Lorenz mit Strukturtypen	155
5.13. Strukturgraph von St. Sebald mit Strukturtypen	157
5.14. Spreading-Activation-Graph für den Strukturgraph von St. Lorenz	161
5.15. Strukturgraph mit Aktivierungswerten - Anfrage <i>Lo:Pfeiler 5</i>	166
5.16. Strukturgraph (Außenbau) mit Aktivierungswerten - Anfrage <i>Geschoss</i>	167
5.17. Strukturgraph (Innenbau) mit Aktivierungswerten - Anfrage <i>Geschoss</i>	168
5.18. Ablauf Anfragerelaxation - Gesamtbild	169
5.19. Strukturgraph (umgedreht) mit Aktivierungswerten - Anfrage <i>Pfeiler</i>	174
5.20. Strukturgraph (umgedreht) mit Aktivierungswerten - Alternative Aktivierung	174
6.1. Top-Down: <i>part_of</i> -Gewichtung in Richtung untergeordnetes Gebäudeteil (entnommen aus [Hel11])	182
6.2. Bottom-Up: <i>part_of</i> -Gewichtung in Richtung übergeordnetes Gebäudeteil (entnommen aus [Hel11])	183
B.1. Polygone der Westfassade von St. Lorenz	195

Tabellenverzeichnis

3.1. Wichtige Sprachkonstrukte der Beschreibungslogik SR0IF mit Rollenkomposition	56
6.1. Ermittelte Parameter für das Spreading-Activation-Verfahren im MonArch-Anwendungsfall	184
C.1. Abbildung DL nach OWL-2-DL	201
C.1. Abbildung DL nach OWL-2-DL	202

Liste der Listings

3.1. OWL-2-DL Klassendefinition	65
3.2. OWL-2-DL Klasse	65
3.3. OWL-2-DL Klassenhierarchien	65
3.4. OWL-2-DL Gleichheit und Disjunktheit	65
3.5. OWL-2-DL Eigenschaften	65
3.6. OWL-2-DL Eigenschaften Hierarchie	65
3.7. OWL-2-DL Eigenschaften Definitions- und Wertebereiche	66
3.8. OWL-2-DL Dateneigenschaften	66
3.9. OWL-2-DL komplexere Klassen	66
3.10. OWL-2-DL Inverse	67
3.11. OWL-2-DL Symmetrisch	67
3.12. OWL-2-DL Asymmetrisch	67
3.13. OWL-2-DL Reflexiv	67
3.14. OWL-2-DL Irreflexiv	67
3.15. OWL-2-DL Funktional	67
3.16. OWL-2-DL Transitiv	67
3.17. OWL-2-DL Property Chains	67
4.1. Standard-MonArch-Anfrage in SQL bezogen auf das erweiterte Datenmodell	71
4.2. Standard-MonArch-Anfrage in MonQL bezogen auf das erweiterte Datenmodell	73
4.3. Beispiel räumliche Anfrage per ID in MonQL	81
4.4. Beispiel räumliche Anfrage per KEYWORD in MonQL	81
4.5. Beispiel räumliche Anfrage per ID mit UP in MonQL	82
4.6. Beispiel räumliche Anfrage per ID mit DOWN in MonQL	83
4.7. Räumliche Anfrage per ID mit DOWN in SQL	83
4.8. SQL-Sicht für part-of -Hierarchie	83
4.9. Beispiel räumliche Anfrage per ID mit UP und DOWN in MonQL	84
4.10. Beispiel grafische Anfrage über Koordinaten	86
4.11. SQL-Sicht für super topic -Hierarchie	89
4.12. Beispiel thematische Anfrage per ID (Naturstein)	90
4.13. Beispiel thematische Anfrage per KEYWORD (Naturstein)	90
4.14. Thematische SQL-Anfrage per KEYWORD	90
4.15. Beispiel thematische Anfrage per ID (Restaurierungsphase)	91
4.16. Beispiel typbasierte Anfrage per KEYWORD	92
4.17. Anfrageschema aus dem MonArch-Anwendungsfall	92
4.18. Anfrage aus dem MonArch-Anwendungsfall	93

4.19. Anfrage aus dem MonArch-Anwendungsfall	93
4.20. Neue Anfragemöglichkeiten in MonQL	94
4.21. Differenz in MonQL	94
4.22. Differenz in MonQL	95
4.23. Räumliche Anfrage per ID	100
4.24. Räumliche SPARQL-Anfrage per ID	100
4.25. Thematische Anfrage per KEYWORD	101
4.26. Thematische SPARQL-Anfrage per KEYWORD	101
4.27. Räumliche und thematische Anfragekombination per ID	101
4.28. Räumliche und thematische SPARQL-Anfragekombination per ID	102

Liste der Algorithmen

5.1. Anfragetransformation im Quellsystem	123
5.2. Anfragetransformation im Zielsystem	124
5.3. Spreading-Activation-Algorithmus	128
5.4. Thematische Anfragetransformation im Quellsystem	136
5.5. Thematische Anfragetransformation im Zielsystem	138
5.6. Thematische Anfragetransformation im Zielsystem mit Spreading-Activation	143
5.7. Räumliche Anfragetransformation im Quellsystem	154
5.8. Räumliche Anfragetransformation im Zielsystem	156
5.9. Räumliche Anfragetransformation im Zielsystem mit Spreading-Activation .	162