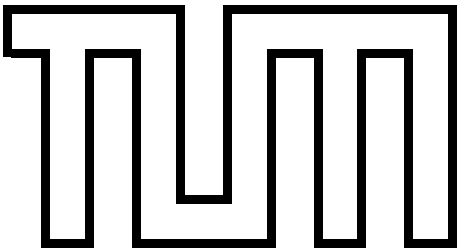


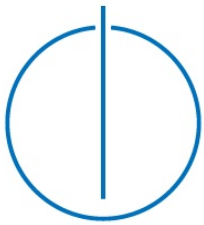
Technische Universität München

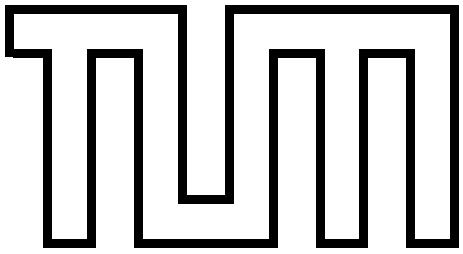
Fakultät für Informatik

Bachelor's Thesis in Informatik

Generischer Linked Data Service Crawler
für strukturierte Webseiten

Jonas Staudenmeir





Technische Universität München

Fakultät für Informatik

Bachelor's Thesis in Informatik

Generischer Linked Data Service Crawler
für strukturierte Webseiten

Generic Linked Data Service Crawler
for Structured Websites

Bearbeiter: Jonas Staudenmeir

Aufgabensteller: Prof. Dr. Johann Schlichter

Betreuer: Dipl.-Inf. Johannes Claudius Hauptmann

Abgabedatum: 16.11.2015

Ich versichere, dass ich diese Bachelor's Thesis selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, 16.11.2015

(Jonas Staudenmeir)

Inhaltsangabe

Viele Webseiten im Internet werden mithilfe von Vorlagen und den Inhalten aus Datenbanken automatisiert generiert. Jedoch bieten nur wenige dieser Webseiten eine öffentliche Schnittstelle für die hinterlegten Informationen an. Aus diesem Grund wird in dieser Arbeit ein Ansatz vorgestellt, um Daten automatisiert aus strukturierten Webseiten zu extrahieren. Im Gegensatz zu existierenden Arbeiten, welche den Aufbau weitgehend automatisch erfassen, soll das entwickelte Verfahren durch detaillierte Konfiguration fehlerfreie Ergebnisse liefern. Die Speicherung der ausgelesenen Informationen erfolgt mithilfe von RDF, um eine Veröffentlichung als Linked Data zu ermöglichen.

Die Grundlage des entworfenen Ansatzes bildet ein Metamodell, mit dem der Nutzer Modelle zur Beschreibung von Webseiten erstellen kann. Zur Feststellung der Anforderung wurden fünf verschiedene Webseiten ausgewählt und analysiert. Neben dem Aufbau einer Webseite können im Modell Anweisungen zur Aufbereitung der extrahierten Daten angegeben werden. Um eine Webseite mehrfach erfassen zu können, werden die Versionen separat gespeichert und deren Unterschiede ermittelt.

Mithilfe einer prototypischen Implementierung wurde der entwickelte Ansatz evaluiert. Es konnte gezeigt werden, dass er für die ausgewählten Webseiten korrekt funktioniert und grundlegend auf strukturierte Webseiten anwendbar scheint.

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Motivation

Nur ein geringer Teil der Webseiten im Internet bietet eine öffentliche Schnittstelle an, um strukturiert auf die hinterlegten Informationen zuzugreifen. Um diese maschinell verarbeiten zu können, besteht jedoch in vielen Fällen ein Interesse an Ausgabemöglichkeiten, die über HTML-Dokumente hinausgehen. Ein mögliches Ziel ist das Sammeln von Daten zu Forschungszwecken.

Wenn Webseiten automatisiert generiert werden, lassen sie sich aufgrund des einheitlichen Aufbaus mit geeigneter Software wieder automatisiert auslesen. Die extrahierten Daten können beispielsweise in einer Datenbank abgespeichert und anschließend weiterverarbeitet werden. Mögliche Anwendungen sind das Durchsuchen, Filtern oder statistische Auswertungen. Darüber hinaus können hinzugefügte, geänderte oder entfernte Datensätze automatisiert erkannt werden.

Da die Informationen bereits im Internet veröffentlicht wurden, sollen sie dort in strukturierter Form und mit einer passenden Schnittstelle auch wieder bereitgestellt werden. Hierfür bietet sich der vom W3C veröffentlichte Standard für Linked Data an [W3Cc]. Dieser verwendet RDF [W3Cf], um Daten und deren Verbindungen untereinander mithilfe von URIs [IETFd] zu beschreiben. Ein Beispiel für einen solchen Datenbestand ist das Projekt DBpedia, welches Informationen aus der Wikipedia bereitstellt.[1]

Ein konkreter Anwendungsfall ist die Webseite des Deutschen Bundestages, auf der alle

[1] http://dbpedia.org

Parteispenden über 50.000 Euro öffentlich einsehbar sind.[2] Seit 2009 werden diese pro Jahr in einem HTML-Dokument dargestellt und können automatisiert ausgelesen werden. Somit ist es beispielsweise möglich, aus den extrahierten Informationen die Gesamtbeträge pro Partei zu berechnen. Auch die Benachrichtigung bei neuen Spenden wäre eine mögliche Nutzung.

1.2 Ziel

Das Ziel der Arbeit ist die Entwicklung eines generischen Ansatzes zum automatisierten Extrahieren von Informationen aus strukturierten Webseiten. Hierzu wird ein Metamodell entworfen, welches zur Erstellung von Modellen genutzt werden kann, die den Aufbau der Webseiten beschreiben. Auf diesem Weg können verschiedene Webseiten mit der gleichen Software erfasst werden, da lediglich ein Austausch des Modells notwendig ist. Die extrahierten Daten sollen in einem SPARQL-Repository [W3Ce] gespeichert werden, so dass eine Veröffentlichung als Linked Data möglich ist. Zudem soll ein Verfahren gefunden werden, das einen Abgleich zwischen bestehenden Daten und dem aktuellen Stand einer Webseite erlaubt.

1.3 Vorgehen

Im ersten Schritt soll die Related Work nach ähnlichen Problemstellungen durchsucht werden. Die von den jeweiligen Autoren vorgestellten Lösungen sollen auf ihre Anwendbarkeit bei dieser Arbeit geprüft werden. Anschließend wird anhand von fünf verschiedenen Webseiten eine Analyse der Anforderungen an das Metamodell durchgeführt. Diese Webseiten sollen unterschiedliche technische Merkmale besitzen und so ein breites Spektrum abdecken. Auf dieser Basis wird ein Ansatz vorgeschlagen, der die herausgearbeiteten Anforderungen erfüllt. Zum Abgleichen von verschiedenen Versionen einer Webseite soll ein Algorithmus entwickelt werden, der hinzugefügte, geänderte und entfernte Daten erkennt. Um die Praxistauglichkeit des entworfenen Metamodells und des Abgleichverfahrens zu zeigen, wird das System exemplarisch in Java implementiert und anhand der ausgewählten Webseiten evaluiert.

[2] https://www.bundestag.de/bundestag/parteienfinanzierung/fundstellen50000

Kapitel 2

Related Work

Bühmann et al. stellen in ihrer Arbeit das Open Source Framework REX vor, das der Extraktion von Informationen aus strukturierten Webseiten dient [Buhm 14]. Um den kostenintensiven Personalaufwand für die Beaufsichtigung und Kontrolle des Crawlers zu eliminieren, setzen sie dabei auf einen selbstlernenden Algorithmus. Dieser benötigt keine ausführliche Konfiguration und erkennt selbstständig die auszulesenden Daten. Dabei überprüft der Crawler deren Validität anhand von Trainingsdaten, bevor er sie als Ergebnis abspeichert. Da ein großer Teil der im Internet veröffentlichten Informationen auf strukturierten Webseiten vorliegt und eine Extraktion unter menschlicher Aufsicht nicht ausreichend skaliert, sind die Autoren der Ansicht, dass sich das Projekt der Linked Open Data Cloud nur auf dem vorgestellten Weg in signifikanten Schritten vorantreiben lässt.

Arasu und Garcia-Molina untersuchen in ihrer Arbeit ebenfalls die automatische Extraktion von Daten aus strukturierten Webseiten [Aras 03]. Ihr Ansatz konzentriert sich dabei auf der Erkennung der gemeinsamen Vorlage, auf der eine Menge von Seiten basiert. Der Prozess soll ohne Trainingsdaten oder Konfiguration funktionieren. Die Autoren begründen diese Zielsetzung mit dem hohen Zeitaufwand und der Fehleranfälligkeit von menschlicher Arbeit. Es sei für Menschen in vielen Fällen schwierig, die gemeinsame Vorlage von mehreren Seiten korrekt herauszuarbeiten. Dies gelte besonders bei optionalen Elementen, die nicht auf allen Seiten vorhanden sind. Da sich die Vorlagen im Laufe der Zeit auch verändern können, würde eine einmalige Vorarbeit durch einen Menschen nicht einmal ausreichen.

Hogue und Karger stellen in ihrer Arbeit das entwickelte System Thresher vor, mit dem auch technisch nicht versierte Nutzer den semantischen Aufbau von Webseiten beschreiben können [Hogu 05]. In einem speziellen Browser können die Anwender auf den Zielwebseiten

Elemente auswählen und deren Bedeutung erläutern. Das System erkennt Seiten mit einer ähnlichen Struktur und wendet die gelernten Regeln auf diese Seiten an. Ziel des Ansatzes ist es, die Akzeptanz und Verbreitung des Semantischen Webs durch niedrigere Einstiegshürden zu erhöhen.

Die beiden zu Beginn genannten Arbeiten versuchen, durch selbstständige Algorithmen den kostenintensiven Personalaufwand zu reduzieren. Im Gegenzug nehmen sie dafür eine höhere Fehlerrate in Kauf. Die dritte vorgestellte Arbeit hingegen verfolgt einen entgegengesetzten Ansatz, hier wird ausdrücklich auf menschliche Vorarbeit gesetzt.

Der in der dieser Arbeit zu entwickelnde Crawler soll durch detaillierte Beschreibungen der Zielwebseiten fehlerfreie Ergebnisse liefern. Im Gegensatz zu Hogue und Karger erfordert er dabei jedoch einen fortgeschritteneren Nutzer. Auf diese Weise soll es möglich sein, auch komplexere Strukturen zu erfassen, beispielsweise über mehrere Seiten hinweg. Ein Bereich, der von keiner der drei Arbeiten behandelt wurde, sind Nutzerinteraktionen mit der Webseite, beispielsweise das Klicken oder Scrollen

Kapitel 3

Grundlagen

Technologien

Linked Data [W3Cc] verwendet **RDF** (Resource Description Framework) [W3Cf], um Daten und deren Verbindungen untereinander zu beschreiben. RDF nutzt **Tripel** aus Subjekt, Prädikat und Objekt, um Aussagen zu formulieren. Subjekte können aus einer URI oder einem Blank Node bestehen, als Prädikate sind nur URIs möglich. Als Objekt kann eine URI, ein Blank Node oder ein Literal genutzt werden. Eine Menge von Tripeln bildet einen **Graph**. Die RDF-Daten können in einem **Repository** gespeichert und per **SPARQL** [W3Ce] abgefragt werden.

URIs (Uniform Resource Identifier) [IETFd] werden zur Identifikation von Ressourcen verwendet. Ein Beispiel ist die folgende Zeichenfolge:
`http://example.com/?foo=bar&abc=123#test`
URIs bestehen aus mehreren Bestandteilen, bei diesem Beispiel bildet `http` das **Schema**. Der Teil `foo=bar&abc=123` wird als Query bezeichnet. In dieser Arbeit werden ihre Komponenten `foo=bar` und `abc=123` **URI-Parameter** genannt. Hierbei bilden `foo` und `abc` jeweils den Namen, `bar` und `123` den Wert. Der Abschnitt nach dem Doppelkreuz wird als **Fragment** bezeichnet, in diesem Fall `test`.

Als Syntax für die Notation von RDF wird in dieser Arbeit Notation3 (N3) verwendet [W3Cd].

Für URIs werden die folgenden Präfixe definiert:

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix :    <predicate://> .
```

Mithilfe der Auszeichnungssprache **HTML** (Hypertext Markup Language) [W3Cb] beschriebene Dokumente bilden die Grundlage von Webseiten. Diese können per **HTTP**-Anfrage (Hypertext Transfer Protocol) [IETFc] von einem Server abgerufen werden. Anhand eines Beispiels sollen einige Begriffe erläutert werden.

Listing 3.1: Beispiel für ein HTML-Dokument

```
<!DOCTYPE html>
<html>
    <head>
        <title>HTML</title>
    </head>
    <body>
        <p class="test">Beispiel</p>
        <a href="http://example.com"></a>
    </body>
</html>
```

HTML-Dokumente bestehen aus geschachtelten Elementen, in diesem Fall sind dies unter anderem `<p class="test">Beispiel</p>` und `<a href="http://example.com"></a>`. Beim ersten Element stellt `Beispiel` den sichtbaren **Text** dar. Letzteres ist ein **Verweis** oder **Link**, mit dem der Nutzer auf eine andere Webseite gelangt. Die Bestandteile `class="test"` und `href="http://example.com"` werden als **Attribute** bezeichnet. Dabei bilden `class` und `href` jeweils den Namen, `test` und `http://example.com` den Wert.

Begriffe

Für die Arbeit sollen einige Begriffe definiert werden:

- **Webseite**: Gesamtheit der Unterseiten einer Domain
- **Unterseite**: Teil einer Webseite, identifiziert durch URI
- **Seite**: Gruppe von Unterseiten, die auf derselben Vorlage basieren
- **Klasse**: Abstraktion der dargestellten Informationen
- **Objekt**: Instanz einer Klasse
- **Eigenschaft**: Bestandteile einer Klasse

Bei einer Webseite mit Kochrezepten beispielsweise, wäre ein einzelnes Rezept eine Unterseite. Alle Rezept-Unterseiten bilden mit ihrer gemeinsamen Vorlage eine Seite. Das Kochrezept wäre eine Klasse und ein konkretes Rezept ein Objekt davon. Eigenschaften der Klasse wären beispielsweise der Name des Rezepts oder die benötigte Arbeitszeit.

Kapitel 4

Crawler für strukturierte Webseiten

4.1 Anforderungen

Damit der Crawler eine strukturierte Webseite durchsuchen und verarbeiten kann, muss deren Aufbau mithilfe von RDF beschrieben werden. Der Crawler muss die HTML-Dokumente der Webseite abrufen und die gewünschten Informationen extrahieren. Anschließend werden die ausgelesenen Daten aufbereitet und im Repository abgespeichert.

Grundlage

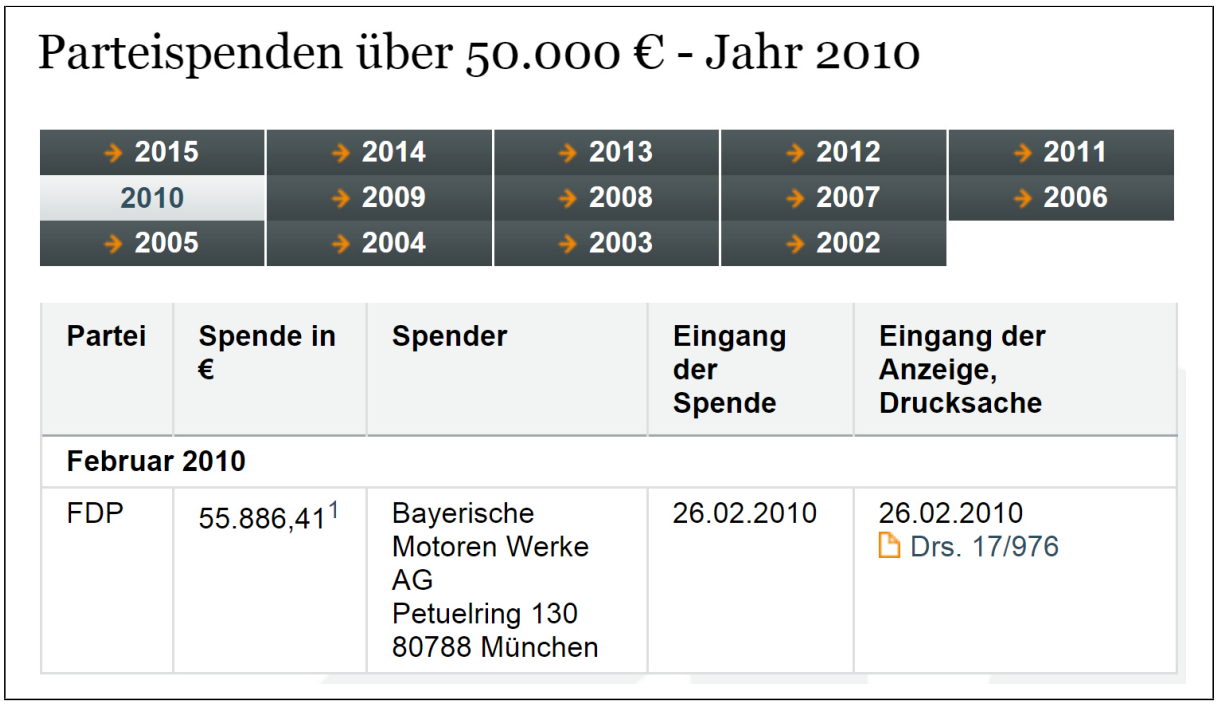
Zur Bestimmung der Anforderungen an den Crawler wurden fünf Webseiten ausgewählt. Diese stammen aus unterschiedlichen Domänen und weisen verschiedene technische Merkmale auf. Auf diese Weise soll ein möglichst allgemeingültiger Ansatz erarbeitet werden, der ein großes Spektrum an Webseiten abdeckt.

Die erste Webseite ist die des **Deutschen Bundestages**, auf der alle Parteispenden über 50.000 Euro öffentlich einsehbar sind.[1] Da die Auflistungen bis zum Jahr 2008 lediglich als PDF zur Verfügung stehen, werden in dieser Arbeit nur die Spenden seit 2009 berücksichtigt. Die Abbildung 4.1 zeigt einen Ausschnitt aus dem Jahr 2010.

Als zweite Webseite wurde das Verzeichnis aller Lehrveranstaltungen der **TU München** ausgewählt, welches in die einzelnen Fakultäten unterteilt ist.[2] Hier sollen alle

[1] https://www.bundestag.de/bundestag/parteienfinanzierung/fundstellen50000
[2] https://campus.tum.de/tumonline/wborg.display_virtuell?PORGNR=1&PORGTYP=28567

Abbildung 4.1: Ausschnitt aus den Parteispenden im Jahr 2010



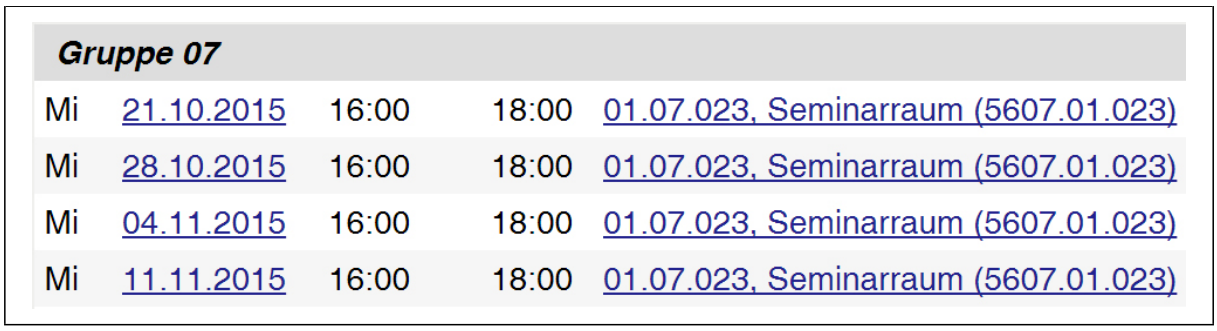
Parteispenden über 50.000 € - Jahr 2010

→ 2015	→ 2014	→ 2013	→ 2012	→ 2011
2010	→ 2009	→ 2008	→ 2007	→ 2006
→ 2005	→ 2004	→ 2003	→ 2002	

Partei	Spende in €	Spender	Eingang der Spende	Eingang der Anzeige, Drucksache
Februar 2010				
FDP	55.886,41[1]	Bayerische Motoren Werke AG Petuelring 130 80788 München	26.02.2010	26.02.2010 Drs. 17/976

Lehrveranstaltungen des aktuellen Semesters zusammen mit ihren Terminen gecrawlt werden. Die Abbildung 4.2 zeigt Ausschnitte aus den Terminen einer Lehrveranstaltung.

Abbildung 4.2: Ausschnitt aus den Terminen einer Lehrveranstaltung der TU München



Gruppe 07				
Mi	21.10.2015	16:00	18:00	01.07.023, Seminarraum (5607.01.023)
Mi	28.10.2015	16:00	18:00	01.07.023, Seminarraum (5607.01.023)
Mi	04.11.2015	16:00	18:00	01.07.023, Seminarraum (5607.01.023)
Mi	11.11.2015	16:00	18:00	01.07.023, Seminarraum (5607.01.023)

Die dritte Webseite bildet **chefkoch.de**, eine Datenbank für Kochrezepte. Von dort sollen alle Rezepte für die Gerichte Donauwelle[3] und Kaiserschmarrn[4] mit den benötigten Zutaten gecrawlt werden. Die Abbildung 4.3 zeigt einen Ausschnitt aus einem Rezept für Donauwelle.

Die nächste Webseite ist die Film- und Seriendatenbank **IMDb**, auf der die Nutzer

[3]http://www.chefkoch.de/rs/s0/Donauwelle/Rezepte.html
[4]http://www.chefkoch.de/rs/s0/Kaiserschmarrn/Rezepte.html

Abbildung 4.3: Ausschnitt aus einem Kochrezept

Filme und Serien bewerten können. Hier soll die Liste der laut Bewertungen 250 besten Serien gecrawlt werden.[5] Jeder Serie sollen zusätzlich ihre Hautdarsteller, Staffeln und deren Episoden zugeordnet werden. Die Abbildung 4.4 zeigt einen Ausschnitt aus einem Serieneintrag.

Als letzte Webseite wurde die der **Stadt München** ausgewählt, die dort eine Übersicht mit aktuellen Veranstaltungen veröffentlicht. Hier sollen vom Crawler die Veranstaltungen aus den Kategorien Comedy[6] und Musical[7] inklusive ihrer Termine und deren Veranstaltungsorten ausgelesen werden. Die Abbildung 4.5 zeigt einen Ausschnitt aus einer Veranstaltung.

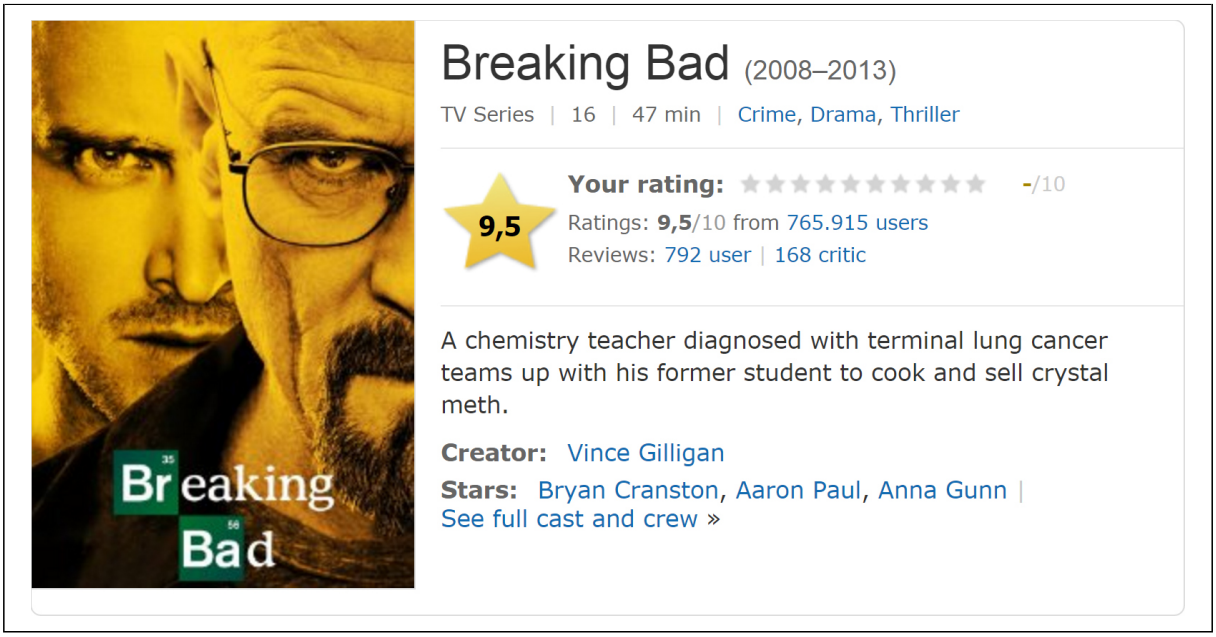
Beschreibung von Bereichen

Um mehrere Webseiten oder verschiedene Teile einer Webseite mit einem einzigen Repository zu crawlen, soll der Nutzer sein Modell in Bereiche unterteilen können. Somit ist es möglich, diese Bereiche separat zu crawlen und damit die unterschiedlichen Änderungsfrequenzen der Webseiten zu berücksichtigen.

Bei den Kochrezepten wird die Einteilung in Bereiche benötigt, um die Rezepte von mehreren verschiedenen Gerichten zu extrahieren. Die Beschreibung der Veranstaltungen in München verwendet je einen Bereich pro Kategorie.

[5] http://www.imdb.com/chart/toptv/
[6] http://www.muenchen.de/veranstaltungen/events/comedy.html
[7] http://www.muenchen.de/veranstaltungen/events/musical.html

Abbildung 4.4: Ausschnitt aus einem Serieneintrag



Beschreibung von Seiten

Um den Crawler verwenden zu können, muss der Nutzer die Struktur der Zielwebseite mit Hilfe von RDF beschreiben. Den ersten Schritt bildet die Identifizierung der verschiedenen Seiten, im zweiten Schritt werden die einzelnen Seiten näher beschrieben. Jeder Seite wird dabei ein Typ zugeordnet. Für die ausgewählten Webseiten werden drei verschiedene Typen benötigt.

Als **Typ 1** werden Seiten bezeichnet, die ein nur einziges Objekt einer bestimmten Klasse enthalten. Die URI der jeweiligen Unterseite ist somit eindeutig diesem Objekt zugeordnet und kann für die Speicherung mit RDF verwendet werden.
Bis auf die des Deutschen Bundestages enthalten alle ausgewählten Webseiten diesen Seitentyp. Beispiele sind die Seiten einer Lehrveranstaltung, eines Kochrezepts oder einer Serie.

Typ 2 bilden Seiten, die mehrere Objekte einer Klasse enthalten. Dabei besitzen die einzelnen Objekte keine eigenen URIs. Um die Objekte dennoch eindeutig identifizieren zu können, muss der Crawler für sie URIs erzeugen.
Die Seite der Parteispenden ist hierfür ein Beispiel. Die Termine einer Lehrveranstaltung der TU München sind in verschiedene Gruppen eingeteilt, diese besitzen ebenfalls keine eigenen URIs. Bei den Kochrezepten besitzt die Auflistung der Zutaten diesen Typ.

Abbildung 4.5: Ausschnitt aus einer Veranstaltung in München

Vom **Typ 3** sind Seiten, die aus einem oder mehreren Verweisen auf Seiten des Typs 1 bestehen. Diese Seiten können bereits Informationen der verlinkten Objekte enthalten, der Crawler extrahiert jedoch lediglich die URIs und verarbeitet anschließend deren Unterseiten. Seiten vom Typ 2 oder 3 werden im Folgenden auch als Liste bezeichnet.
Da Seiten vom Typ 3 und Typ 1 immer zusammen auftreten, wird dieser Typ mit Ausnahme der Webseite des Deutschen Bundestages ebenfalls von allen ausgewählten Webseiten genutzt. Beispiele sind die Listen der Lehrveranstaltungen, Kochrezepte oder Serien.

Beschreibung von Seitenverbindungen

Da die zu crawlenden Webseiten in der Regel aus mehreren Seiten bestehen, müssen auch deren Verbindungen untereinander beschrieben werden.

Die erste Möglichkeit bildet die im vorherigen Abschnitt aufgeführte Verlinkung von Seiten des Typs 3 auf Seiten vom Typ 1. Hier gibt der Nutzer in seinem Modell bei der Liste an, welche Seite das Ziel ihrer Verweise ist.

Weiterhin kann es den Fall einer **Schachtelung** geben. Dabei enthalten Seiten in sich weitere Seiten des gleichen oder eines anderen Typs. Diese Schachtelung kann sich über mehrere Ebenen erstrecken.
Die Webseite der Kochrezepte ist ein Beispiel für diese Methode. Die Seite eines Rezepts (Typ 1) enthält die Liste der dazugehörigen Zutaten (Typ 2). Bei den Veranstaltungen in München enthält die Seite einer Veranstaltung (Typ 1) die Seite ihrer Termine (Typ 2). Diese wiederum beinhaltet eine Seite mit dem Verweis auf den Ort, an dem der Termin stattfindet (Typ 3).

Die dritte Möglichkeit einer Seitenverbindung ist der **Link**. Dabei enthalten Seiten vom Typ 1 oder 2 Verweise auf andere Seiten. Dieser Fall entspricht einer Schachtelung, die sich über zwei Unterseiten erstreckt.
Auf der Webseite der TU München wird diese Funktionalität an zwei Stellen benötigt. Von der Seite der Fakultät (Typ 1) gelangt man über einen Verweis zur Liste der dazugehörigen Lehrveranstaltungen (Typ 3). Die Seite einer Lehrveranstaltung (Typ 1) enthält einen Link zur Seite der Termingruppen (Typ 2).

Entfernen von URI-Parametern

Die URIs von Unterseiten können **Parameter** (3) enthalten, die für den Nutzer des Crawlers nicht von Bedeutung sind oder nicht bei jedem Aufruf den gleichen Wert enthalten. Deshalb soll es die Möglichkeit geben, zu entfernende Parameter zu spezifizieren. Konsistente URIs sind besonders für die im Abschnitt 4.4 beschriebene Versionierung wichtig.

Diese Funktionalität wird bei der Seriendatenbank benötigt. Folgt man auf der Liste der besten Serien dem Link zu einem Eintrag, unterscheidet sich die Ziel-URI bei jedem Aufruf. Die URI enthält auch gleichbleibende Parameter, die jedoch für den Crawler und dessen Nutzer nicht relevant sind. Ein Beispiel ist die Referrer-Angabe (`&ref_=chttp_tt_1`) zur Analyse des Besucherverhaltens.

Beschreibung von Klassen

Die zu extrahierenden Informationen werden beim Modell in Klassen eingeteilt. Der Nutzer muss bei Seiten vom Typ 1 oder 2 beschreiben, welche Klasse sie enthalten und wo die Daten auf der Seite zu finden sind.

Beispiele für Klassen sind Parteispende, Lehrveranstaltung, Kochrezept oder Serie.

Beschreibung von Eigenschaften

Jeder Klasse ordnet der Nutzer Eigenschaften zu und spezifiziert für sie jeweils den Pfad zum entsprechenden Element auf der Seite. Bei seiner Arbeit extrahiert der Crawler den Text der Elemente. Eigenschaften sind beispielsweise die Arbeitszeit eines Kochrezepts oder der Titel einer Serie. An die Beschreibung von Eigenschaften werden mehrere Anforderungen gestellt.

Wenn eine Eigenschaft nicht bei allen Objekten einer Klasse vorhanden ist, muss sie als **optional** gekennzeichnet werden.
Beispielsweise besitzen nicht alle Kochrezepte eine Nutzerbewertung.

Bei HTML-Elementen kann es den Fall geben, dass der im Browser sichtbare Wert zusätzlich in einem **Attribut** (3) angeben ist. Der Wert des Attributs ist in der Regel auf Maschinenlesbarkeit und Weiterverarbeitung durch Programme optimiert. Bei manchen Eigenschaften kann es auch sein, dass der benötigte Wert nur in einem Attribut vorhanden und für den Besucher der Seite nicht sichtbar ist.
Bei den Veranstaltungsterminen in München werden beispielsweise Datum und Uhrzeit des Anfangs in maschinenlesbarer Form in einem Attribut angegeben. Die durchschnittliche Nutzerbewertung eines Kochrezepts kann nur auf diese Weise extrahiert werden.

Der vom Crawler extrahierte Text eines Elements oder Attributs kann mehr Informationen enthalten, als der Nutzer benötigt. Aus diesem Grund soll es für den Nutzer die Möglichkeit der **Begrenzung** geben. Hierfür werden reguläre Ausdrücke verwendet [LRZ].
Bei den Parteispenden enthält die Tabellenzelle mit dem gespendeten Betrag teilweise Verweise auf Anmerkungen oder zusätzlichen Text. Hier wird der Text mit einem regulären Ausdruck auf den reinen Geldbetrag begrenzt.

Im nächsten Schritt soll es die Möglichkeit von **Ersetzungen** geben. Muss der extrahierte Text vor der weiteren Verarbeitung noch angepasst werden, kann dies durch Suchen und Ersetzen erfolgen.
Bei den Parteispenden, Kochrezepten und Serien wird diese Funktion genutzt, um Dezimalzahlen durch das Ersetzen des Kommas mit einem Punkt maschinenlesbar zu machen.

Da RDF verschiedene **Datentypen** [W3Ch] unterstützt, kann der Wert einer Eigenschaft vom Crawler in den entsprechenden Typ konvertiert werden.
Diese Möglichkeit wird von allen ausgewählten Webseiten genutzt. Als Datentypen werden Dauer, Dezimalzahl, Ganzzahl, URI, Datum und Uhrzeit verwendet.

Bei Datentypen, die ein **Datum** und/oder eine Uhrzeit repräsentieren, muss zusätzlich

das auf der Seite verwendete Format angegeben werden. Der Crawler benötigt dieses, um den Wert der Eigenschaft korrekt parsen zu können.
Bei den Terminen von Lehrveranstaltungen der TU München sollen der Tag als Datum und der Anfang als Uhrzeit abgespeichert werden.

Beschreibung von Interaktionen

Weiterhin muss der Crawler in der Lage sein, Benutzerverhalten zu simulieren. Hierbei werden zwei Anforderungen gestellt.

Listen können in mehrere Unterseiten aufgeteilt sein, bei denen man über einen **Weiter**-Button zu den weiteren Teilen gelangt. Der Crawler muss solange auf diesen Button klicken, bis das Ende erreicht ist und der Button verschwindet. Hier kann es zusätzlich den Fall geben, dass der Listeninhalt per JavaScript nachgeladen wird und die einzelnen Teile der Liste somit keine eigenen URIs besitzen.
Bei den ausgewählten Webseiten wird die Funktion von den Parteispenden, den Lehrveranstaltungen und den Kochrezepten benötigt. Die Seite der Parteispenden besitzt keinen separaten Weiter-Button, dieser kann jedoch simuliert werden (Listing A.1). Bei den Lehrveranstaltungen werden die Teile der Liste per JavaScript geladen.

Seiten, die beim **Scrollen** durch das Erreichen des Seitenendes nachgeladen werden, müssen ebenfalls unterstützt werden. Der Crawler muss hier solange scrollen, bis der komplette Inhalt dargestellt wird.
Diese Funktionalität wird bei den Veranstaltungen in München benötigt. Je Kategorie werden beim Aufruf der Seite lediglich die ersten 25 Einträge dargestellt. Erst durch das Scrollen wird die Liste vollständig.

Unterbrechbarkeit

Um auch umfangreiche Webseiten über Wochen hinweg crawlen zu können, muss der Crawler jederzeit unterbrechbar sein, beispielsweise für einen Systemneustart. Der Crawler muss in diesem Fall seinen aktuellen Zustand persistent speichern und seine Arbeit anschließend an der entsprechenden Stelle fortsetzen können.

Übersicht

Die Tabelle 4.1 zeigt, welche Anforderungen die fünf ausgewählten Webseiten an den Crawler stellen.

	bundestag.de	campus.tum.de	chefkoch.de	imdb.com	muenchen.de
Seitentyp 1		•	•	•	•
Seitentyp 2	•	•	•		•
Seitentyp 3		•	•	•	•
Schachtelung		•	•	•	•
Link		•			
Parameter				•	
Optional			•	•	•
Attribut			•	•	•
Begrenzung	•		•	•	
Ersetzung	•		•	•	
Datentyp	•	•	•	•	•
Datum		•			•
Weiter	•	•	•		
Scrollen					•

Tabelle 4.1: Anforderungen der ausgewählten Webseiten

4.2 Metamodell

Aufruf von Webseiten

Der Crawler muss bei seiner Arbeit Webseiten automatisiert aufrufen können. In dieser Arbeit soll der Schwerpunkt darauf liegen, dass der Nutzer das Modell der Zielwebseite auf möglichst einfache Art und Weise beschreiben kann. Die Verarbeitungsgeschwindigkeit spielt dabei nur eine untergeordnete Rolle. Aus diesem Grund wurde hierfür der Browser als Technologie ausgewählt. Eine auf Geschwindigkeit optimierte Alternative wäre der Gebrauch von reinen HTTP-Anfragen, deren Verwendung jedoch für einen Nutzer des Crawlers deutlich komplexer wäre. Ein Nachteil des Browsers ist die geringere

Geschwindigkeit, die besonders durch das Laden von externen Ressourcen verursacht wird.

Die unterschiedliche Komplexität soll an einem Beispiel gezeigt werden:
Die Lehrveranstaltungen der TU München werden pro Fakultät auf Unterseiten à 50 Stück aufgeteilt.[8] Um weitere Veranstaltungen zu laden, muss der Nutzer eines Browsers lediglich auf den Weiter-Button (in diesem Fall ein gelber Pfeil) klicken. Für den Crawler müsste somit nur der Pfad zum entsprechenden HTML-Element beschrieben werden. Die Verwendung von HTTP-Anfragen hingegen gestaltet sich weitaus aufwendiger.
Hier wäre beispielsweise für jeden Teil der Liste eine Anfrage mit folgenden Informationen notwendig:

- `pOrgNr`: `14189`
- `pPersonNr`:
- `pSjNr`: `1601`
- `pStpLvTypNr`: `-1`
- `pPageNr`: `2`
- `pSort`: `5;4;6`
- `pFilter`: `null?f_4_1=W`
- `pGroup`: `W`

Anschließend müssten die erhaltenen Daten noch verarbeitet werden. Dies wird im Browser automatisch vom entsprechenden JavaScript-Code übernommen.

Generierung von URIs

Da die Objekte auf Seiten vom Typ 2 keine eigenen URIs besitzen, muss der Crawler diese erzeugen. Nur mit einer eindeutigen URI können Objekte mit RDF abgespeichert werden. Hier wurde der folgende Ansatz gewählt:
Nach dem Extrahieren wird aus den Daten eines betroffenen Objekts eine String-Darstellung generiert. Diese wird anschließend mithilfe einer Hashfunktion in einen Hash umgewandelt, welcher an die URI der entsprechenden Unterseite angehängt wird. Dabei wird das in 3 beschriebene Fragment genutzt.

[8] Informatik: https://campus.tum.de/tumonline/wblvangebot.wbshowlvoffer?porgnr=14189

Beschreibung von Pfaden

Bei der Verwendung des Crawlers muss der Nutzer im Modell der Webseite beschreiben, welche HTML-Elemente die zu extrahierenden Informationen enthalten. Zur Beschreibung der Pfade zu den Elementen wird die Abfragesprache XPath in der Version 1.0 genutzt [W3Cg]. XPath steht bereits in der umfangreicheren Version 2.0 zur Verfügung. Diese wird jedoch zum aktuellen Zeitpunkt weder von Mozilla Firefox [Mozi] noch von Google Chrome [Goog][Xmls] unterstützt. Eine mögliche Alternative wäre die Verwendung von CSS-Selektoren [W3Ca]. Aufgrund des im Vergleich größeren Funktionsumfangs [Resi] wurde für den Crawler jedoch XPath ausgewählt.

Beschreibung von Bereichen

Bei der Beschreibung von Bereichen verwendet der Nutzer die URI, die dem Crawler als Startpunkt im jeweiligen Bereich dient. Jedes Modell besteht somit aus mindestens einem Bereich.

Bereich

Die URI des Bereichs wird der URI `<model:>` mit dem Prädikat `:section` zugeordnet.

Listing 4.1: Beispiel für die Beschreibung von Bereichen (Listing C.1)

```
<model:>
    :section    <http://www.chefkoch.de/rs/s0/Donauwelle/Rezepte.html> ;
    :section    <http://www.chefkoch.de/rs/s0/Kaiserschmarrn/Rezepte.html> .
```

Seite

Jedem Bereich wird mit dem Prädikat `:page` anschließend die Seite zugeordnet, die den Startpunkt beschreibt.

Listing 4.2: Beispiel für die Beschreibung der Seite eines Bereichs (Listing C.1)

```
<http://www.chefkoch.de/rs/s0/Donauwelle/Rezepte.html>
    :page    <page://recipes> .
```

Beschreibung von Seiten

Um eine Seite zu beschreiben, wählt der Nutzer für sie eine beliebige URI.

Typ

Für jede Seite wird mit dem Prädikat `:type` einer der drei in 4.1 aufgeführten Typen spezifiziert. Dessen URI besteht aus dem Präfix `<type://>` und dem Zusatz `object` (Typ 1), `objects` (Typ 2) oder `links` (Typ 3).

Listing 4.3: Beispiel für die Beschreibung des Typs einer Seite (Listing C.1)

```
<page://recipes>
    :type     <type://links> .
```

Verweisziel

Bei Seiten vom Typ 3 wird im nächsten Schritt mit dem Prädikat `:target` die Zielseite der Verweise angegeben.

Listing 4.4: Beispiel für die Beschreibung des Verweisziels einer Seite (Listing C.1)

```
<page://recipes>
    :target     <page://recipe> .
```

Klasse

Seiten vom Typ 1 oder 2 werden an dieser Stelle mit `:item` die Klasse der enthaltenen Objekte zugeordnet.

Listing 4.5: Beispiel für die Beschreibung der Klasse einer Seite (Listing C.1)

```
<page://recipe>
    :item     <item://recipe> .
```

Beschreibung von Seitenverbindungen

Schachtelung

Bei einer Schachtelung wird der äußeren Seite die innere mit dem Prädikat `:sub` zugeordnet.

Listing 4.6: Beispiel für die Beschreibung einer Schachtelung von Seiten (Listing C.1)

```
<page://recipe>
    :sub    <page://ingredients> .
```

Link

Sollen zwei Seiten verlinkt werden, muss für die Anweisung eine beliebige URI gewählt werden. Der verlinkenden Seite wird diese URI mit dem Prädikat `:link` zugeordnet. Bei der Verlinkung selbst wird mit `:path` der XPath-Pfad zum Link-Element spezifiziert. Mit `:target` wird im nächsten Schritt die URI der verlinkten Seite angegeben. Eine Seite kann mehrere Links enthalten.

Listing 4.7: Beispiel für die Beschreibung einer Verlinkung von Seiten (Listing B.1)

```
<page://faculty>
    :link    <link://courses> .

<link://courses>
    :path    "//a[text()='Lehrveranstaltungen']" ;
    :target    <page://courses> .
```

Entfernen von URI-Parametern

Zum Entfernen von URI-Parametern gibt der Nutzer ihre Namen bei der dazugehörigen Seite mit `:param` an.

Listing 4.8: Beispiel für die Beschreibung von zu entfernenden URI-Parametern (Listing D.1)

```
<page://series>
    :param    " pf_rd_i "  ;
    :param    " ref_ "  .
```

Beschreibung von Klassen

Zur Beschreibung einer Klasse wählt der Nutzer eine beliebige URI mit dem Schema `item` aus (3).

Pfad zum Wurzelelement

Für jede Klasse muss der XPath-Pfad zum Wurzelelement mit `:path` angegeben werden. Dieses Element enthält die Elemente der Eigenschaften und vereinfacht somit deren Pfadbeschreibung. Bei einer Schachtelung von Seiten ist der Pfad der inneren Klasse relativ zum Pfad der äußeren.

Listing 4.9: Beispiel für die Beschreibung des Pfades zum Wurzelelement einer Klasse (Listing C.1)

```
<item://recipe>
    :path     "//div[@id='page']" .

<item://ingredient>
    :path     ".//div[@id='recipe-incredients']//tr" .
```

Eigenschaften

Im nächsten Schritt werden der Klasse ihre Eigenschaften mit `:property` zugeordnet.

Listing 4.10: Beispiel für die Beschreibung der Eigenschaften einer Klasse (Listing C.1)

```
<item://recipe>
    :property     <property://recipe/label> ;
    :property     <property://recipe/rating> .
```

Pfad zum URI-Element

Nach der Definition in 4.1 besitzen Objekte auf Seiten vom Typ 2 keine eindeutigen URIs. Diese werden jedoch für die Speicherung mit RDF benötigt und deshalb vom Crawler erzeugt.

In diesem Abschnitt soll eine Möglichkeit beschrieben werden, den Typ 2 auch bei Objekten mit vorhandenen URIs zu verwenden. Da dieser Anwendungsfall bereits mit einer Kombination aus Typ 3 und Typ 1 abgedeckt werden kann, fügt das Verfahren dem Metamodell keine zusätzliche Funktionalität hinzu. Der Vorteil besteht jedoch in einer mitunter erheblichen Zeitersparnis bei der Laufzeit des Crawlers.

Für die Anwendung der Optimierung muss folgender Fall vorliegen:
Sind auf einer Seite vom Typ 3 alle Informationen enthalten, die eigentlich von der verlinkten Seite des Typs 1 ausgelesen werden sollen, kann letztere weggelassen werden.

Die Liste wird in den Typ 2 konvertiert und zusätzlich wird für die enthaltene Klasse mit `:id` der Pfad zum Link-Element spezifiziert, welches die URI enthält. Dieser Pfad ist relativ zum Wurzelelement der Klasse. So wird bei unverändertem Ergebnis eine Reduzierung der Laufzeit erreicht, da die Unterseiten der einzelnen Objekte nicht aufgerufen werden müssen – alle benötigten Daten können aus der Liste extrahiert werden.

Diese Optimierung wird beispielsweise bei den Terminen von Lehrveranstaltungen der TU München angewendet. Da alle notwendigen Informationen (Datum, Beginn und Ende) bereits in der Liste aller Termine vorhanden sind, müssen deren eigene Unterseiten nicht mehr aufgerufen werden.

Listing 4.11: Beispiel für die Beschreibung des Pfades zum URI-Element einer Klasse (Listing B.1)

```
<item://appointment>
    :id     "./td[2]/a" .
```

Beschreibung von Eigenschaften

Wie bei Seiten und Klassen sind auch bei der Beschreibung von Eigenschaften eigene URIs notwendig, die beliebig gewählt werden können. Die einzige Voraussetzung ist das Schema `property`.

Pfad zum Element

Für jede Eigenschaft muss der XPath-Pfad zum Element mit dem Prädikat `:path` angegeben werden. Dieser ist relativ zum Wurzelelement der Klasse. Beim Crawlvorgang wird standardmäßig der Text dieses Elements extrahiert.

Listing 4.12: Beispiel für die Beschreibung des Pfades zum Element einer Eigenschaft (Listing E.1)

```
<property://appointment/start>
    :path     ".//span[@itemprop='startDate']" .
```

Optionale Eigenschaften

Eine Eigenschaft kann mit `:optional` als optional gekennzeichnet werden, wenn sie nicht bei allen Objekten einer Klasse vorhanden ist. Der Crawler prüft in diesem Fall, ob das dazugehörige Element existiert.

Listing 4.13: Beispiel für die Beschreibung einer optionalen Eigenschaft (Listing C.1)

```
<property://recipe/rating>
    :optional      true  .
```

Attribut

Soll der Crawler den Wert eines HTML-Attributs extrahieren, kann dessen Name mit `:attribute` angegeben werden.

Listing 4.14: Beispiel für die Beschreibung des Attributs einer Eigenschaft (Listing E.1)

```
<property://appointment/start>
    :attribute     "content"  .
```

Begrenzung

Mit dem Prädikat `:pattern` kann einer Eigenschaft ein regulärer Ausdruck zugeordnet werden, mit dem der Text begrenzt werden soll. Wird das angegebene Muster im Text nicht gefunden, speichert der Crawler einen leeren String.

Listing 4.15: Beispiel für die Beschreibung der Begrenzung einer Eigenschaft (Listing A.1)

```
<property://donation/amount>
    :pattern       "[\ d .]{6,}(,\ d{2})?"  .
```

Ersetzung

Ebenfalls möglich sind Ersetzungen im Text. Für eine solche Anweisung benötigt der Nutzer eine eigene Ressource. Diese kann aus einem Blank Node oder einer beliebigen URI bestehen und wird der Eigenschaft mit `:replace` zugeordnet. Bei der Ersetzungsanweisung selbst wird mit `:old` der reguläre Ausdruck angegeben, nach dem gesucht werden soll. Das Prädikat `:new` beschreibt den Text, durch den die gefundenen Stellen ersetzt werden.

Pro Eigenschaft können mehrere Ersetzungsanweisungen angegeben werden. Ist deren Reihenfolge von Bedeutung, kann diese als Ganzzahl zusätzlich mit `:order` angegeben werden. Sie wird in aufsteigender Reihenfolge abgearbeitet.

Listing 4.16: Beispiel für die Beschreibung von Ersetzungen einer Eigenschaft (Listing A.1)

```
<property://donation/amount>
```

```
:replace [
    :old     "\."  ;
    :new     ""    ;
    :order   1
] ;
:replace [
    :old     ","   ;
    :new     "."   ;
    :order   2
] .
```

Datentyp

Standardmäßig speichert der Crawler den extrahierten Text als String-Literal (`xsd:string`). Optional kann der Text in einen der folgenden Datentypen [W3Ch] konvertiert werden:

- URI (`xsd:anyURI`)
- Datum (`xsd:date`)
- Datum und Uhrzeit (`xsd:dateTime`)
- Dezimalzahl (`xsd:decimal`)
- Dauer, nach ISO 8601 [ISO] (`xsd:duration`)
- Ganzzahl (`xsd:int`)
- Uhrzeit (`xsd:time`)

Die Anweisung erfolgt mit dem Prädikat `:type`.

Listing 4.17: Beispiel für die Beschreibung des Datentyps einer Eigenschaft (Listing A.1)

```
<property://donation/amount>
    :type     xsd:decimal .
```

Datum

Stellt der Wert einer Eigenschaft ein Datum und/oder eine Uhrzeit dar (`xsd:date`, `xsd:dateTime`, `xsd:time`), muss für den Parser zusätzlich zum Datentyp das auf der Webseite verwendete Format (nach ISO 8601 [ISO]) mit `:format` angegeben werden.

Listing 4.18: Beispiel für die Beschreibung des Datumsformats einer Eigenschaft (Listing E.1)

```
<property://appointment/start>
    :type      xsd:dateTime ;
    :format   ''yyyy-MM-dd'T'HH:mm:ssZ'' .
```

Beschreibung von Interaktionen

Weiter-Button

Für aufgeteilte Listen gibt der Nutzer bei der Seite den XPath-Pfad zum Weiter-Button mit `:next` an.

Listing 4.19: Beispiel für die Beschreibung des Weiter-Buttons einer Seite (Listing C.1)

```
<page://recipes>
    :next     ''//a[@class='pagination-item pagination-next']'' .
```

Werden die weiteren Teile der Liste per JavaScript nachgeladen, muss zusätzlich eine Wartezeit in Millisekunden mit `:wait` spezifiziert werden. Der Crawler wartet diese nach dem Klick auf den Button ab und beginnt erst danach mit der Verarbeitung.

Listing 4.20: Beispiel für die Beschreibung eines Weiter-Buttons mit Wartezeit (Listing B.1)

```
<page://courses>
    :next     ''//div[@id='idLVOfferTable']//table[@class='coTableNavi'][1]
                //select/following-sibling::a''  ;
    :wait     7000 .
```

Alternativ möglich wäre die Beobachtung eines „Warte-Elements“, das dem Nutzer den aktuellen Zustand anzeigt. Dieses erscheint nach dem Klick auf den Weiter-Button und verschwindet nach dem erfolgreichen Nachladen wieder. Da jedoch nicht alle Webseiten ein solches Element besitzen, wird mit der Wartezeit gearbeitet.

Scrollen

Soll eine Seite solange gescrollt werden, bis sie komplett geladen ist, muss eine Wartezeit angegeben werden, welche die maximale Dauer einer Nachlade-Anfrage darstellt. Der Crawler wartet diese nach jedem Scrollen ab, bevor er prüft, ob weitere Inhalte geladen werden können. Die Wartezeit in Millisekunden wird bei der entsprechenden Seite mit `:wait` spezifiziert.

Listing 4.21: Beispiel für die Beschreibung einer Anweisung zum Scrollen (Listing E.1)

```
<page://events>
    : scroll      1000 .
```

Wie im vorherigen Abschnitt wäre es auch hier theoretisch möglich, ein „Warte-Element“ zu beobachten. Aus dem gleichen Grund wurde jedoch beim Scrollen ebenfalls die Wartezeit bevorzugt.

4.3 Ergebnis

Der Crawler extrahiert bei seiner Arbeit die Objekte der im Modell beschriebenen Klassen und die Werte ihrer Eigenschaften.

Bereiche

Für jeden Bereich des Modells erzeugt der Crawler einen **gerichteten Baum**. Besteht ein Modell aus mehreren Bereichen, kann es je nach Webseite Verbindungen zwischen den Bäumen geben. Das Ergebnis der Veranstaltungen in München enthält solche Verbindungen, wenn an Orten Veranstaltungen aus beiden Kategorien stattfinden.

An der Wurzel eines Baumes steht die URI des Bereichs. Von dort führen mit der URI der Klasse als Prädikat die Kanten zu den Objekten der Startseite.

Listing 4.22: Beispiel für das Ergebnis eines Objekts auf der Startseite (Listing A.5)

```
<https://www.bundestag.de/bundestag/parteienfinanzierung/fundstellen50000>
    <item://donation>  <https://www.bundestag.de/bundestag/parteienfinanzierung/
        fundstellen50000/2010#a0b7dcad346ca3731b6e8e66609a9cc1cd5e9e19> .
```

Ist die Seite des Startpunkts vom Typ 3, dient die verlinkte Seite (Typ 1) als Quelle der Objekte.

Listing 4.23: Beispiel für das Ergebnis eines auf der Startseite verlinkten Objekts (Listing D.5)

```
<http://www.imdb.com/chart/toptv/>
    <item://series>    <http://www.imdb.com/title/tt0903747/> .
```

Seiten und Seitenverbindungen

Die Struktur der Seiten und ihrer Verbindungen untereinander wird im Ergebnis durch Verbindungen zwischen den enthaltenen Objekten dargestellt. Durch die Beschaffenheit von RDF werden Tripel in einem Graphen nicht doppelt angelegt [W3Cf]. Somit existiert ein Objekt mit seinen Daten nur ein einziges Mal, auch wenn es selbst mehreren anderen Objekten zugeordnet ist.

Schachtelung

Bei geschachtelten Seiten werden den äußeren Objekten die inneren Objekten zugeordnet. Als Prädikat dient hier die URI der inneren Klasse.

Listing 4.24: Beispiel für das Ergebnis eines geschachtelten Objekts (Listing D.5)

```
<http://www.imdb.com/title/tt0903747/>
    <item://actor>    <http://www.imdb.com/name/nm0186505/> .
```

Link

Durch einen Link verbundene Seiten werden im Ergebnis wie geschachtelte Seiten behandelt. Der Link wird hierbei nicht separat abgebildet. Die URI der verlinkten Klasse bildet das Prädikat.

Listing 4.25: Beispiel für das Ergebnis eines verlinkten Objekts (Listing B.5)

```
<https://campus.tum.de/tumonline/organisationen.display?corg=14189>
    <item://course>    <https://campus.tum.de/tumonline/lv.detail?clvnr=950208933> .
```

Eigenschaften

Die Blätter der Bäume bilden die Werte der Eigenschaften, diese werden ihren Objekten zugeordnet. Die Werte werden als Literal abgespeichert, dabei wird der spezifizierte Datentyp verwendet. Besitzt ein Objekt eine optionale Eigenschaft nicht, fehlt das Tripel. Als Prädikat dient die URI der Eigenschaft.

Listing 4.26: Beispiel für das Ergebnis einer Eigenschaft (Listing D.5)

```
<http://www.imdb.com/title/tt0903747/>
    <property://series/rating>      "9.5"^^ xsd:decimal .
```

4.4 Versionierung

Um eine Webseite mehrfach crawlen zu können, werden die Ergebnisse für jeden Durchgang separat gespeichert. Hierfür wird im Repository für jeden Durchgang ein eigener Graph angelegt. Dessen URI beinhaltet den Zeitstempel der Sekunde, in welcher der Crawler mit dem Durchgang begonnen hat, beispielsweise: `<run://1444404781>`.

Nach jedem Durchgang vergleicht der Crawler die Ergebnisse mit denen des vorherigen, ermittelt die Unterschiede und protokolliert sie in einem eigenen Repository. Er erkennt dabei, welche Objekte und Eigenschaften in der Zwischenzeit hinzugefügt, geändert oder entfernt wurden. Auch in diesem Repository werden Graphen angelegt, deren URIs den URIs der Graphen im Repository mit den Ergebnissen entsprechen. Der Vorgang wird auch beim ersten Crawlen einer Webseite ausgeführt.

Objekte

Hinzufügung

Wird ein Objekt auf der Startseite eines Bereichs hinzugefügt, wird im Protokoll der URI des Bereichs die URI des Objekts mit dem Prädikat `:added` zugeordnet.

Listing 4.27: Beispiel für ein hinzugefügtes Objekt auf der Startseite (Listing D.8)

```
<http://www.imdb.com/chart/toptv/>
    :added    <http://www.imdb.com/title/tt0903747/> .
```

Wird bei einer Schachtelung von Seiten einem äußeren Objekt ein inneres Objekt hinzugefügt, erzeugt der Crawler ein Tripel aus der URI des äußeren Objekts, dem Prädikat `:added` und der URI des inneren Objekts.

Listing 4.28: Beispiel für ein hinzugefügtes Objekt auf einer geschachtelten Seite (Listing D.7)

```
<http://www.imdb.com/title/tt0903747/>
    :added    <http://www.imdb.com/name/nm0186505/> .
```

Entfernung

Werden Objekte entfernt, entsprechen die Einträge im Protokoll den Einträgen von hinzugefügten Objekten, als Prädikat wird jedoch `:removed` verwendet.

Listing 4.29: Beispiel für ein entferntes Objekt auf der Startseite (Listing D.12)

```
<http://www.imdb.com/chart/toptv/>
    :removed    <http://www.imdb.com/title/tt123456789/> .
```

Listing 4.30: Beispiel für ein entferntes Objekt auf einer geschachtelten Seite (Listing D.11)

```
<http://www.imdb.com/title/tt123456789/>
    :removed    <http://www.imdb.com/name/nm123456789/> .
```

Eigenschaften

Hinzufügung

Wird auf der Webseite eine Eigenschaft zu einem Objekt hinzugefügt, wird im Protokoll der URI des Objekts die URI der Eigenschaft mit dem Prädikat `:added` zugeordnet.

Listing 4.31: Beispiel für eine hinzugefügte Eigenschaft (Listing D.7)

```
<http://www.imdb.com/title/tt0903747/>
    :added    <property://series/label> .
```

Änderung

Ändert sich der Wert einer Eigenschaft, wird als Prädikat `:changed` verwendet.

Listing 4.32: Beispiel für eine geänderte Eigenschaft (Listing D.12)

```
<http://www.imdb.com/title/tt0903747/>
    :changed    <property://series/rating> .
```

Bei Objekten auf Seiten vom Typ 2 können keine geänderten Werte von Eigenschaften erkannt werden, da sich dadurch auch immer die URI des Objekts ändert. Im Protokoll erschienen diese Objekte mit der alten URI als entfernt und mit der neuen als hinzugefügt. Dies ist beispielsweise auf der Webseite des Deutschen Bundestags bei den Parteispenden der Fall.

Entfernung

Bei entfernten Eigenschaften wird als Prädikat `:removed` verwendet.

Listing 4.33: Beispiel für eine entfernte Eigenschaft (Listing D.11)

```
<http://www.imdb.com/title/tt123456789/>
    :removed    <property://series/label> .
```

Optimierung

Um die ermittelten Unterschiede sinnvoll verarbeiten zu können, werden nicht alle Veränderungen einzeln protokolliert. Werden mehrere miteinander verbundene Objekte hinzugefügt oder entfernt, erzeugt der Crawler nur Einträge für die in der Hierarchie jeweils höchst gelegenen Objekte. Hinzugefügte oder entfernte Eigenschaften werden nur protokolliert, wenn ihre dazugehörigen Objekte nicht ebenfalls hinzugefügt oder entfernt wurden.

Wird beispielsweise die Webseite der IMDb zum ersten Mal gecrawlt, wird nur das Hinzufügen der einzelnen Serien protokolliert. Für die untergeordneten Schauspieler, Staffeln, Episoden und alle Eigenschaften werden keine Einträge angelegt. Wird bei einem weiteren Durchlauf eine neue Staffel zu einer bereits vorhandenen Serie erkannt, wird nur für die Staffel ein Tripel erzeugt. Erhält eine bereits existierende Serie zwischen zwei Durchgängen des Crawlers ein Plakat, wird das Hinzufügen dieser Eigenschaft protokolliert, da die dazugehörige Serie nicht neu ist.

Kapitel 5

Implementierung

Um die Praxistauglichkeit des entwickelten Metamodells zu evaluieren, wurde das System in Java implementiert.

Struktur

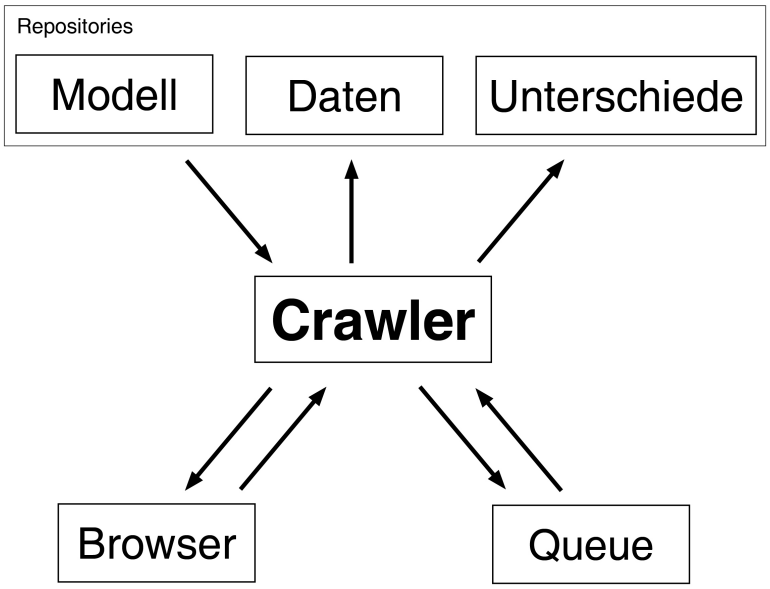
Der implementierte Crawler arbeitet mit drei verschiedenen SPARQL-Repositories. Im ersten legt der Nutzer vor dem Start das Modell der Zielwebseite ab, welches er mithilfe des Metamodells erstellt hat. In das zweite Repository schreibt der Crawler während seiner Arbeit die extrahierten und aufbereiteten Daten. Nach Abschluss des Durchgangs vergleicht er diese mit denen des vorherigen Durchlaufs und protokolliert im dritten Repository die ermittelten Unterschiede.

Um die in 4.1 geforderte Unterbrechbarkeit des Crawlvorgangs zu ermöglichen, arbeitet die Implementierung mit einer persistenten Queue. Darin verwaltet der Crawler die URIs der Unterseiten, die noch abgearbeitet werden müssen. Durch die Verwendung von Transaktionen wird sichergestellt, dass bei einem plötzlichen Programmabbruch der aktuelle Zustand nicht verloren geht und der Crawler seine Arbeit nach dem Neustart an der entsprechenden Stelle fortsetzen kann. Die Nutzung der Queue ermöglicht zusätzlich eine noch zu implementierende Parallelisierung der Anwendung.

Die Methoden zur Ausführung der in Kapitel 6 beschriebenen Evaluation befinden sich in der Klasse `CrawlerTest`.

Abbildung 5.1 zeigt die Struktur der Implementierung.

Abbildung 5.1: Struktur der Implementierung



Software und Bibliotheken

Die Implementierung verwendet das Build-Management-Tool Apache Maven[1]. Zum Aufrufen der Webseiten wird der Browser Mozilla Firefox[2] eingesetzt. Die Software Selenium[3] ermöglicht dem Crawler, den Browser automatisiert auszuführen und die im Modell beschriebenen Elemente per XPath-Ausdruck aus den HTML-Dokumenten zu extrahieren. Die Kommunikation mit den SPARQL-Repositories übernimmt die Bibliothek Sesame[4]. Apache ActiveMQ[5] realisiert für das System eine persistente Queue.

Generierung von URIs

Der in 4.2 beschriebene Ansatz zur Generierung von URIs für Objekte auf Seiten vom Typ 2 wird bei der Implementierung mithilfe von JSON [IETFb] und SHA-1 [IETFa] umgesetzt. Das Format JSON wird verwendet, um die extrahierten Daten eines Objekts in

[1] https://maven.apache.org
[2] https://www.mozilla.org/firefox/
[3] http://seleniumhq.org
[4] http://rdf4j.org
[5] http://activemq.apache.org

eine String-Darstellung zu überführen. Dabei werden die Eigenschaften in alphabetischer Reihenfolge ihrer URIs mit ihren Werte in einem Array angeordnet. Der entstandene String wird anschließend mit SHA-1 gehasht und im letzten Schritt als Fragment an die URI der Unterseite angehängt.

An einem Beispiel aus Listing C.3 soll das Vorgehen veranschaulicht werden:
Einem Kochrezept sind mehrere Zutaten zugeordnet, für die jeweils eine URI generiert werden muss. Die erste Zutat besteht aus einer Bezeichnung („Zucker“) und einer Mengenangabe („250 g“), daraus entsteht der folgende JSON-String:

```
[{"property://ingredient/amount":"250 g"},
    {"property://ingredient/label":"Zucker"}]
```

Hieraus erzeugt der SHA-1 den Hash `aeabf42db4c00f93ee072e77e940e4d3e11c656f`. Dieser wird abschließend als Fragement an die URI der Unterseite (entspricht der URI des Rezepts) angefügt:

```
http://www.chefkoch.de/rezepte/372611122991708/Donauwelle-Konditorenart.html
    #aeabf42db4c00f93ee072e77e940e4d3e11c656f
```

Durch die Kollisionsresistenz von SHA-1 ist es praktisch ausgeschlossen, dass zwei verschiedenen Objekten die gleiche URI zugeordnet wird.

Optimierung der Geschwindigkeit

Eine Möglichkeit zur Optimierung des Crawlvorgangs ist die Verringerung der Ladezeit von Webseiten. Ein großen Einfluss darauf hat das Herunterladen von externen Ressourcen, beispielsweise Grafiken oder Werbeanzeigen. Um dies zu unterbinden, können in Firefox Erweiterungen installiert werden. Zwei mögliche Add-ons sind:

- QuickJava[6]: Diese Erweiterung kann unter anderem das Laden von Bildern oder CSS-Dateien verhindern.
- Adblock Plus[7]: Dieses Add-on blockiert Werbebanner.

Mithilfe eines separaten Profils können diese Konfigurationen von der regulären Nutzung des Browsers getrennt werden. Existiert für Firefox ein Profil mit dem Namen „crawler“, wird dies automatisch vom Crawler geladen.

[6] https://addons.mozilla.org/de/firefox/addon/quickjava/
[7] https://addons.mozilla.org/de/firefox/addon/adblock-plus/

Kapitel 6

Evaluation

6.1 Ziel

Mit der Evaluation wird das Erreichen der in Abschnitt 1.2 formulierten Ziele überprüft. Es soll gezeigt werden, dass es möglich ist, mithilfe des Metamodells (4.2) Modelle zur Beschreibung der ausgewählten Webseiten (4.1) zu erstellen. Dafür wird ermittelt, ob mit den Modellen die geforderten Informationen aus den Webseiten extrahiert und abgespeichert werden können. Um dies zu beantworten, wird die Problemstellung in drei Fragen aufgeteilt:

1. Werden die in den Modellen beschriebenen Informationen aus den Webseiten extrahiert?
2. Werden die ausgelesenen Daten korrekt aufbereitet und wie in 4.3 spezifiziert im Repository abgespeichert?
3. Werden die Unterschiede zwischen verschiedenen Versionen einer Webseite wie in 4.4 beschrieben ermittelt?

6.2 Vorgehen

Um die in 6.1 gestellten Fragen zu beantworten, wird die Implementierung aus Kapitel 5 auf die ausgewählten Webseiten angewendet. Hierfür werden zuerst Modelle erstellt, welche die zu extrahierenden Informationen beschreiben. Anschließend wird der Crawler mit diesen Modellen ausgeführt und seine Ergebnisse werden mit den erwarteten Daten verglichen.

6.2.1 Frage 1: Extraktion

Zur Beantwortung der ersten Frage werden auf allen Webseiten einzelne, möglichst repräsentative Objekte ausgewählt. Die Auswahl erfolgt so, dass alle Bestandteile des jeweiligen Modells zur Anwendung kommen. Um bei Listen mit mehreren Teilen das Klicken des Weiter-Buttons zu testen, werden Objekte gewählt, die nicht auf der ersten Unterseite enthalten sind. Nach dem Durchlauf des Crawlers wird geprüft, ob die Informationen der ausgewählten Objekte in den extrahierten Daten enthalten sind. Die erwarteten Ergebnisse werden mithilfe von JSON [IETFb] angegeben. Dabei werden der URI eines Objekts die Soll-Werte seiner Eigenschaften zugeordnet.

Von der Webseite des Deutschen Bundestages sollen alle Parteispenden seit 2009 extrahiert werden.[1] Pro Jahr existiert eine Unterseite, die in einer Tabelle die jeweiligen Spenden enthält. Hier sollen zu jeder Spende die ersten vier Spalten ausgelesen werden, die Partei, Betrag, Spender und Datum beinhalten.
Für die Evaluation wurde die Spende vom 26.02.2010 ausgewählt (Abbildung 4.1).[2] Da sie keine eigene URI besitzt, wird diese vom Crawler erzeugt (Seite vom Typ 2). Das Listing A.2 zeigt die zu extrahierenden Daten.

Auf der Webseite der TU München soll der Crawler alle Fakultäten[3] mit ihren Lehrveranstaltungen des aktuellen Semesters, deren in Gruppen eingeteilten Terminen und den Räumen erfassen. Zu jeder Fakultät wird der Name ausgelesen, zu jeder Veranstaltung Titel, Nummer und Art. Bei den Gruppen soll der Name extrahiert werden, bei den Terminen Datum, Beginn und Ende und bei den Räumen die Bezeichnung.
Für die Evaluation wurde die Fakultät Informatik[4] mit der Lehrveranstaltung „Tutorübungen zu Grundlagen: Betriebssysteme und Systemsoftware (IN0009)“[5] ausgewählt. Hier soll der vierte Termin der Gruppe 07[6] mit seinem Raum[7] geprüft werden (Abbildung 4.2). Die URIs der Gruppen werden vom Crawler erzeugt (Seite vom Typ 2). Listing B.2 zeigt das zu erreichende Ergebnis.

Von der Webseite chefkoch.de sollen alle Rezepte für Donauwelle[8] und Kaiserschmarrn[9] zusammen mit ihren Zutaten gecrawlt werden. Zu jedem Rezept sollen die Elemente

[1] https://www.bundestag.de/bundestag/parteienfinanzierung/fundstellen50000
[2] https://www.bundestag.de/bundestag/parteienfinanzierung/fundstellen50000/2010
[3] https://campus.tum.de/tumonline/wborg.display_virtuell?PORGNR=1&PORGTYP=28567
[4] https://campus.tum.de/tumonline/organisationen.display?corg=14189
[5] https://campus.tum.de/tumonline/lv.detail?clvnr=950208933
[6] https://campus.tum.de/tumonline/!wbTermin.wbEdit?pTerminNr=884835648
[7] https://campus.tum.de/tumonline/ris.einzelRaum?raumKey=29234
[8] http://www.chefkoch.de/rs/s0/Donauwelle/Rezepte.html
[9] http://www.chefkoch.de/rs/s0/Kaiserschmarrn/Rezepte.html

ausgelesen werden, die Bezeichnung, benötigte Arbeitszeit, Schwierigkeitsgrad, Zubereitung, Anzahl der Portionen und Nutzerbewertung enthalten. Da Rezepte ohne Bewertung existieren, ist diese Eigenschaft optional. Bei den Zutaten werden Bezeichnung und Menge (bezogen auf die Portionenzahl des Rezepts) extrahiert.
Für die Evaluation wurden zwei Rezepte ausgewählt – eines für Donauwelle[10] (Abbildung 4.3) und eines für Kaiserschmarrn[11]. Das zweite Rezept besitzt keine Bewertung. Zu jedem Rezept wurde zudem eine Zutat ausgewählt. Deren URIs werden vom Crawler erzeugt (Seite vom Typ 2). Listing C.2 zeigt das zu erzielende Ergebnis (die Zubereitungstexte wurden für die Darstellung gekürzt).

Von der Webseite der IMDb sollen die 250 Serien mit den besten Nutzerbewertungen[12] zusammen mit ihren Hautdarstellern, Staffeln und deren Episoden gecrawlt werden. Zu jeder Serie sollen die Inhalte der Elemente extrahiert werden, welche Titel, Bewertung, Beschreibung und URI des Plakats enthalten. Bei den drei wichtigsten Schauspielern soll der Name, bei den Staffeln die Nummer und bei den Episoden Titel, Nummer und Bewertung ausgelesen werden.
Für die Evaluation wurde die Serie „Breaking Bad“ ausgewählt (Abbildung 4.4).[13] Hier werden alle drei Hauptdarsteller[14][15][16] und die sechste Episode[17] der dritten Staffel[18] geprüft. Das Listing D.2 zeigt die zu extrahierenden Daten.

Auf der Webseite der Stadt München sollen alle Veranstaltungen aus den Kategorien Comedy[19] und Musical[20] zusammen mit ihren Terminen und deren Veranstaltungsorten erfasst werden. Zu jeder Veranstaltung soll der Crawler den Titel und den optionalen Beschreibungstext auslesen. Bei den Terminen sollen Datum und Uhrzeit des Beginns extrahiert werden, bei den Orten die Bezeichnung und die Adresse.
Für die Evaluation wurden zwei Veranstaltungen ausgewählt – eine aus der Kategorie Comedy[21] (Abbildung 4.5) und ein Musical[22]. Bei der Comedy-Veranstaltung wird der

[10] http://www.chefkoch.de/rezepte/372611122991708/Donauwelle-Konditorenart.html
[11] http://www.chefkoch.de/rezepte/2309811368533750/Kaiserschmarrn-la-Mittermeier.html
[12] http://www.imdb.com/chart/toptv/
[13] http://www.imdb.com/title/tt0903747/
[14] http://www.imdb.com/name/nm0186505/
[15] http://www.imdb.com/name/nm0666739/
[16] http://www.imdb.com/name/nm0348152/
[17] http://www.imdb.com/title/tt1615556/
[18] http://www.imdb.com/title/tt0903747/episodes?season=3
[19] http://www.muenchen.de/veranstaltungen/events/comedy.html
[20] http://www.muenchen.de/veranstaltungen/events/musical.html
[21] http://www.muenchen.de/veranstaltungen/event/22060.html
[22] http://www.muenchen.de/veranstaltungen/event/18009.html

erste Termin zusammen mit seinem Ort geprüft, beim Musical der letzte Termin.[23][24] Die URIs der Termine werden vom Crawler generiert (Seite vom Typ 2). Listing E.2 zeigt das zu erreichende Ergebnis.

6.2.2 Frage 2: Aufbereitung und Speicherung

Nach der Extraktion der Daten aus der Webseite bereitet der Crawler diese auf und speichert sie anschließend im entsprechenden Repository. Anhand der im vorherigen Abschnitt ausgewählten Objekte wird bei der zweiten Frage geprüft, ob deren extrahierte Informationen wie in den Modellen angegeben verarbeitet und im korrekten Datentyp abgespeichert werden. Hierfür werden die erwarteten Ergebnisse mit RDF beschrieben. Anschließend wird geprüft, ob diese in den vom Crawler erzeugten Tripeln vorhanden sind.

Beim Crawlen der Webseite des Deutschen Bundestages muss der extrahierte Text des gespendeten Betrags mithilfe eines regulären Ausdrucks begrenzt werden, da dessen Zelle Verweise auf Anmerkungen (wie im ausgewählten Beispiel) oder zusätzlichen Text (z. B. „Korrigierter Betrag:“ am 01.07.2009) enthalten kann. Anschließend müssen die Punkte entfernt und das Komma durch einen Punkt ersetzt werden, damit der Betrag als Dezimalzahl abgespeichert werden kann. Für die ausgewählte Spende zeigt Listing A.4 das erwartete Ergebnis.

Bei den Lehrveranstaltungen der TU München sollen Datum, Beginn und Ende der Termine in den entsprechenden Formaten (`xsd:date`, `xsd:time`) abgespeichert werden. Das Ergebnis soll wie in Listing B.4 gespeichert werden.

Bei den Rezepten von chefkoch.de muss der Crawler zwei Eigenschaften vor der Speicherung aufbereiten. Da das HTML-Element des Schwierigkeitsgrads weiteren Text enthält, muss dieser mit einem regulären Ausdruck entfernt werden. Die Verarbeitung der extrahierten Bewertung benötigt ebenfalls einen regulären Ausdruck und zusätzlich eine Ersetzung des Unterstrichs durch einen Punkt, um sie in eine Dezimalzahl zu konvertieren. Die Arbeitszeit wird als Dauer abgespeichert, die Portionenzahl als Ganzzahl. Listing C.4 zeigt das zu erzielende Ergebnis.

Zur Aufbereitung der Daten von imdb.com muss bei den Bewertungen von Serien und Episoden das Komma durch einen Punkt ersetzt werden, damit sie anschließend als Dezimalzahlen abgespeichert werden können. Die Nummern von Staffeln und Episoden werden mit einem regulären Ausdruck extrahiert und in Ganzzahlen konvertiert. Die URI

[23] http://www.muenchen.de/veranstaltungen/orte/131027.html
[24] http://www.muenchen.de/veranstaltungen/orte/140242.html

des Serienplakats wird als URI im Repository gespeichert. Das Ergebnis soll die Daten aus Listing D.4 enthalten.

Bei den Veranstaltungen in München soll der Beginn als Datum und Uhrzeit abgespeichert werden (`xsd:dateTime`). Listing E.4 zeigt das zu erreichende Ergebnis.

6.2.3 Frage 3: Versionierung

Nachdem der Crawler eine Webseite vollständig abgearbeitet hat, vergleicht er die Ergebnisse mit denen des vorherigen Durchgangs. Die ermittelten Unterschiede werden in einem separaten Repository protokolliert. Zur Beantwortung der dritten Frage wird wie in 6.2.2 die Existenz von RDF-Tripeln geprüft. Um die in 4.4 beschriebene Optimierung zu testen, wird zusätzlich betrachtet, ob bestimmte Tripel vom Crawler nicht erzeugt wurden.

Zuerst wird die Erkennung von hinzugefügten Einträgen getestet, da beim ersten Durchgang die Ergebnisse des vorherigen leer bzw. nicht vorhanden sind. Um auch die Erkennung von geänderten und entfernten Einträgen zu evaluieren, wird der Crawler für jede Webseite erneut ausgeführt. Da die verwendeten Webseiten für die Tests nicht bearbeitet werden können, werden Änderungen simuliert. Dies geschieht vor dem zweiten Durchlauf durch Anpassungen an den Ergebnissen des ersten Durchgangs.

Nach dem ersten Crawlen der Webseite des Deutschen Bundestages soll der Crawler erkennen, dass die ausgewählte Spende hinzugefügt wurde (Listing A.6). Da die Spende neu hinzugekommen ist, dürfen für ihre Eigenschaften keine Einträge angelegt werden (Listing A.7).
Vor dem zweiten Durchgang wird in das Ergebnis des ersten eine fiktive Spende eingefügt (Listing A.9). Nach dem Durchlauf soll der Crawler erkennen, dass diese Spende nicht mehr existiert (Listing A.10). Da die Spende entfernt wurde, darf das Entfernen ihrer Eigenschaften nicht protokolliert werden (Listing A.11).
Änderungen an den Eigenschaften von Spenden können vom Crawler nicht erkannt werden, da sich dadurch auch immer die URI der Spende ändert (Seite vom Typ 2).

Bei der Webseite der TU München soll der Crawler nach dem ersten Durchgang erkennen, dass die ausgewählte Fakultät hinzugekommen ist (Listing B.6). Da die Fakultät hinzugefügt wurde, dürfen für ihre weiteren Objekte (Lehrveranstaltung, Gruppe, Termin, Raum) und alle Eigenschaften keine Einträge angelegt werden (Listing B.7).
Um das Erkennen von Änderungen zu testen, wird vor dem zweiten Crawlen im Ergebnis des ersten Durchgangs der Beginn des ausgewählten Termins verändert. Zusätzlich wird zur ausgewählten Fakultät eine fiktive Lehrveranstaltung hinzugefügt und das Ergebnis um

eine fiktive Fakultät ergänzt (Listing B.9). Nach der zweiten Ausführung soll der Crawler die Änderung am Termin und das Entfernen der fiktiven Lehrveranstaltung und Fakultät erkennen (Listing B.10). Dabei darf im Protokoll nicht das Entfernen ihrer Eigenschaften und dazugehörigen Objekte festgehalten werden (Listing B.11).

Nach dem ersten Crawlen von chefkoch.de soll der Crawler erkennen, dass die beiden ausgewählten Rezepte hinzugefügt wurden (Listing C.6). Da die Rezepte neu hinzugekommen sind, dürfen für ihre Eigenschaften, Zutaten und deren Eigenschaften keine zusätzlichen Einträge existieren (Listing C.7).
Um das Erkennen von Änderungen zu testen, wird vor dem zweiten Durchgang im Ergebnis des vorherigen die Bewertung des ersten ausgewählten Rezepts verändert. Weiterhin wird zu diesem Rezept eine fiktive Zutat und zum anderen Rezept eine Bewertung hinzugefügt. Zusätzlich wird das Ergebnis um ein fiktives Rezept erweitert (Listing C.9). Nach dem Durchlauf soll der Crawler die Änderung der ersten und das Entfernen der zweiten Bewertung erkennen. Ebenso muss er protokollieren, dass die fiktive Zutat und das fiktive Rezept nicht mehr existieren (Listing C.10). Für die Eigenschaften und die Zutat dieses Rezepts dürfen jedoch keine Einträge angelegt werden (Listing C.11)

Bei der Webseite der IMDb soll der Crawler nach dem ersten Durchgang erkennen, dass die ausgewählte Serie hinzugekommen ist (Listing D.6). Da die Serie hinzugefügt wurde, dürfen für ihre weiteren Objekte (Schauspieler, Staffel, Episode) und alle Eigenschaften keine Einträge angelegt werden (Listing D.7).
Um das Erkennen von Änderungen zu testen, wird vor dem zweiten Crawlen im Ergebnis des ersten Durchgangs die Bewertung der ausgewählten Serie verändert. Zusätzlich wird zur ausgewählten Serie eine fiktive Staffel hinzugefügt und das Ergebnis um eine fiktive Serie ergänzt (Listing D.9). Nach der zweiten Ausführung soll der Crawler die Änderung der Bewertung und das Entfernen der fiktiven Staffel und Serie erkennen (Listing D.10). Dabei darf im Protokoll nicht das Entfernen ihrer Eigenschaften und dazugehörigen Objekte festgehalten werden (Listing D.11).

Nach dem ersten Crawlen der Webseite der Stadt München soll der Crawler erkennen, dass die beiden ausgewählten Veranstaltungen hinzugefügt wurden (Listing E.6). Da die Veranstaltungen hinzugekommen sind, dürfen für ihre Eigenschaften, Termine, Orte und deren Eigenschaften keine zusätzlichen Einträge existieren (Listing E.7).
Um das Erkennen von Änderungen zu testen, wird vor dem zweiten Durchgang im Ergebnis des vorherigen die Adresse des zweiten ausgewählten Veranstaltungsortes verändert. Weiterhin wird zur ersten Veranstaltung ein fiktiver Termin hinzugefügt und das Ergebnis wird um eine fiktive Veranstaltung erweitert (Listing E.9). Nach dem Durchlauf soll der Crawler die Änderung der Adresse des Ortes erkennen. Ebenso muss er protokollieren,

dass der fiktive Termin und die fiktive Veranstaltung nicht mehr existieren (Listing E.10). Für die Eigenschaften, Termine und den Ort dieser Veranstaltung dürfen jedoch keine Einträge angelegt werden (Listing E.11)

6.3 Ergebnisse

In den Listings A.1, B.1, C.1, D.1 und E.1 befinden sich die zur Evaluation verwendeten Modelle. Die Listings A.3, B.3, C.3, D.3 und E.3 beinhalten die Ergebnisse zu Frage 1. Für Frage 2 werden die Ergebnisse in den Listings A.5, B.5, C.5, D.5 und E.5 aufgeführt. Die Listings A.8, A.12, B.8, B.12, C.8, C.12, D.8, D.12, E.8 und E.12 dokumentieren die Ergebnisse für Frage 3.

6.4 Diskussion

Um die in 6.1 gestellten Fragen zu beantworten, werden in diesem Abschnitt die Ergebnisse aus 6.3 mit den erwarteten Daten aus 6.2 verglichen. Hierdurch soll festgestellt werden, ob das Forschungsziel der Arbeit (1.2) erreicht wurde.

6.4.1 Frage 1: Extraktion

Für die Webseiten 1, 3, 4 und 5 entsprechen die Ergebnisse den erwarteten Daten.

Bei der Webseite der TU München ist es aufgrund der HTML-Struktur und den Beschränkungen von XPath 1.0 nicht möglich, die Informationen exakt wie geplant zu extrahieren. Die Gruppen von Lehrveranstaltungen sollten ursprünglich als eigene Klasse modelliert werden. Dies ist so jedoch nicht realisierbar, da bei Lehrveranstaltungen mit mehreren Gruppen kein XPath-1.0-Ausdruck existiert, mit dem jeweils alle dazugehörigen Termine ausgelesen werden. In Folge dessen sind die Termine im Modell direkt der Lehrveranstaltung zugeordnet und der Name der Gruppe wird als Eigenschaft des Termins beschreiben. Auf diese Weise können durch eine Anpassung des Modells dennoch alle gewünschten Informationen gecrawlt werden.

Somit wurde gezeigt, dass der Crawler die in den Modellen beschriebenen Informationen korrekt aus den Webseiten extrahiert.

6.4.2 Frage 2: Aufbereitung und Speicherung

Beim Crawlen von chefkoch.de wurde die Arbeitszeit des ersten Rezepts von `PT60M` zu `PT1H` geändert, die beiden Ausdrücke sind jedoch äquivalent [ISO]. Bei den Veranstaltungen in München wurde der Beginn von `2016-01-22T20:00:00+01` zu `2016-01-22T20:00:00.000+01:00` bzw. von `2016-01-03T14:30:00+01` zu `2016-01-03T14:30:00.000+01:00` geändert. Die beiden Ausdrücke sind jeweils äquivalent. Somit entsprechen die Ergebnisse der Webseiten 1, 3, 4 und 5 den erwarteten Daten.

Durch die Anpassung des Modells unterscheiden sich bei der Webseite 2 auch für diese Frage die Ergebnisse von den erwarteten Daten. Es können dennoch ebenfalls alle geforderten Daten verarbeitet und gespeichert werden. Zusätzlich unterscheiden sich Beginn und Ende von Terminen: `16:00:00` wurde zu `16:00:00.000` und `18:00:00` zu `18:00:00.000` geändert. Die Werte sind jedoch äquivalent.

Die Ergebnisse zeigen, dass die ausgelesenen Daten vom Crawler korrekt aufbereitet und im Repository abgespeichert werden.

6.4.3 Frage 3: Versionierung

Bei allen Webseiten entsprechen die Ergebnisse den erwarteten Daten. Hieraus kann der Schluss gezogen werden, dass der Crawler die Unterschiede zwischen verschiedenen Versionen einer Webseite wie spezifiziert ermittelt.

6.4.4 Fazit

Die positive Beantwortung der drei Fragen zeigt, dass der in dieser Arbeit entwickelte Ansatz für die ausgewählten Webseiten funktioniert. Das Ergebnis deutet darauf hin, dass der Ansatz grundlegend auf strukturierte Webseiten anwendbar ist. Mithilfe des Metamodells können diese so beschrieben werden, dass eine generische Implementierung die spezifizierten Informationen extrahiert, verarbeitet und abspeichert. Auch das mehrfache Crawlen von Webseiten und die damit verbundene Versionierung ist möglich.

6.5 Einschränkungen

Die Anzahl der ausgewählten Webseiten ist zu gering, um Aussagen darüber treffen, ob das Metamodell auf einen Großteil der strukturierten Webseiten anwendbar ist. Auch die Verwendung der zur Anforderungsermittlung genutzten Webseiten schränkt die Aussagekraft der Evaluation ein. Daher sollten noch weitere, möglichst unterschiedliche Webseiten in die Auswahl miteinbezogen werden.

Ein weiterer Aspekt ist die Geschwindigkeit des Crawlers. Hier bietet sich ein Vergleich zwischen dem Browser und der Verwendung von reinen HTTP-Anfragen an. Die Implementierung könnte durch eine Bibliothek wie Jsoup[25] realisiert werden.

[25] http://jsoup.org

Kapitel 7

Zusammenfassung und Ausblick

7.1 Zusammenfassung

In dieser Arbeit wurde untersucht, wie strukturierte Webseiten mit generischer Software automatisiert ausgelesen und in einem SPARQL-Repository abgespeichert werden können.

Anhand von fünf Webseiten wurde ein Metamodell entwickelt, mit dem der Nutzer Modelle zur Beschreibung von Webseiten erstellen kann. Für jede Webseite werden dabei die verschiedenen Seiten und ihre Verbindungen untereinander angegeben. Mithilfe von XPath-Ausdrücken beschreibt der Nutzer, welche Klassen und Eigenschaften vom Crawler extrahiert werden sollen. Dabei kann der sichtbare Text von HTML-Elementen oder der Wert eines Attributs ausgelesen werden. Um die Modellierung zu vereinfachen, können vom Nutzer Interaktionen angegeben werden, die der Crawler in einem Browser simuliert. Auf diese Weise können auf mehrere Unterseiten aufgeteilte Listen durch Klicken des Weiter-Buttons erfasst und Seiten durch Scrollen vollständig geladen werden.

Weiterhin wurde analysiert, welche Verfahren zur Aufbereitung der extrahierten Daten notwendig sind, bevor diese mithilfe von RDF gespeichert werden. Die ausgelesenen Informationen können mit regulären Ausdrücken begrenzt werden. Des Weiteren sind Ersetzungen möglich. Die aufbereiteten Daten können mit verschiedenen Datentypen abgespeichert werden: URI, Datum und/oder Uhrzeit, Dezimalzahl, Dauer, Ganzzahl.

Um eine Webseite mehrfach crawlen zu können, werden die Ergebnisse jedes Durchgangs separat gespeichert. Nach jedem Durchlauf werden vom Crawler die Unterschiede zum vorherigen ermittelt und in einem eigenen Repository protokolliert. Er erkennt dabei hinzugefügte, geänderte und entfernte Einträge.

7.2 Ausblick

Mit dem entwickelten Metamodell können die fünf ausgewählten Webseiten wie gefordert beschrieben werden. Um jedoch Aussagen über dessen allgemeine Anwendbarkeit auf strukturierte Webseiten treffen zu können, ist es erforderlich, die Anforderungsermittlung auf weitere, möglichst unterschiedliche Webseiten auszuweiten. Dabei soll untersucht werden, ob das Metamodell um zusätzliche Funktionalität ergänzt werden muss, damit es auch für diese Webseiten verwendbar ist.
Der Fokus kann dabei beispielsweise auf die Simulierung von Nutzerinteraktionen gelegt werden, da hier die Vorteile des Browsers aufgrund der möglichen Ausführung von JavaScript-Code zur Geltung kommen.

Ein weiteres Forschungsgebiet ist die Erhöhung der Geschwindigkeit des Crawlers durch die Verwendung von reinen HTTP-Anfragen. Hier kann man analysieren, wie sich der Zeitaufwand für die Anpassung der Implementierung zur Ersparnis durch die Geschwindigkeitssteigerung verhält. Dabei können die finanziellen Aspekte untersucht werden, beispielsweise bei der Nutzung von Cloud Computing [BSI] bei Anbietern wie Amazon, Google oder Microsoft.
Denkbar wäre beispielsweise eine Hybridlösung, die sowohl einen Browser als auch reine HTTP-Anfragen nutzt. So können Webseiten, deren Extraktion ohne Ausführung von JavaScript-Code aufwendig ist, von der vereinfachten Modellerstellung profitieren. Bei den restlichen Webseiten verringern HTTP-Anfragen den Ressourcenverbrauch.

Da nach dem ersten Durchgang das regelmäßige Crawlen der gesamten Webseite nicht immer notwendig oder wirtschaftlich sinnvoll ist, bietet sich eine Priorisierung von einzelnen Teilbereichen an. Dieses Thema wurde bereits in einer anderen Bachelorarbeit behandelt. Daher kann noch untersucht werden, wie sich eine Ablaufplanung in das bestehende Metamodell integrieren lässt.

Um das Ergebnis des Crawlers zuverlässig weiterverarbeiten zu können, ist eine gewisse Datenintegrität hilfreich. Im Metamodell ist dazu die Kennzeichnung von optionalen Eigenschaften vorgesehen, alle anderen werden als verpflichtend betrachtet. Hier bietet sich die Möglichkeit, den entwickelten Ansatz um weitere Regeln zu ergänzen, mit denen der Nutzer Voraussetzungen an Klassen und Eigenschaften definieren kann.

Anhang A

Evaluation von bundestag.de

Listing A.1: Modell

```
@prefix :              <predicate://> .
@prefix type:          <type://> .
@prefix page:          <page://> .
@prefix item:          <item://> .
@prefix donation:      <property://donation/> .
@prefix section:       <https://www.bundestag.de/bundestag/parteienfinanzierung/fundstellen50000> .

<model:>
    :section     section: .

section:
    :page        page:donations .

page:donations
    :type        type:objects ;
    :next        "//div[@class='jahreLeiste']// li [@class='aktiv']
                     /following-sibling:: li [1][.// span/text()>='2009']/a" ;
    :item        item:donation .

item:donation
    :path        "//div[@class='inhalt']//tbody/tr[count(td)>1]" ;
    :property    donation:party ;
    :property    donation:amount ;
    :property    donation:donor ;
    :property    donation:date .

donation:party
    :path        "./td[1]" .

donation:amount
    :path        "./td[2]" ;
    :pattern     "[\d .]{6,}(,\ d{2})?" ;
    :replace [
        :old     "\." ;
        :new     "" ;
        :order   1
    ] ;
    :replace [
        :old     "," ;
        :new     "." ;
        :order   2
    ] ;
    :type        xsd:decimal .

donation:donor
    :path        "./td[3]" .

donation:date
    :path        "./td[4]" .
```

Frage 1: Extraktion

Listing A.2: Ziel


```
{
  "https://www.bundestag.de/bundestag/parteienfinanzierung/fundstellen50000
      /2010#a0b7dcad346ca3731b6e8e66609a9cc1cd5e9e19": {
    "property://donation/party": "FDP",
    "property://donation/amount": "55.886,411",
    "property://donation/donor": "Bayerische Motoren Werke AG\nPetuelring 130\n80788 München",
    "property://donation/date": "26.02.2010"
  }
}
```


Listing A.3: Ergebnis


```
{
  "https://www.bundestag.de/bundestag/parteienfinanzierung/fundstellen50000
      /2010#a0b7dcad346ca3731b6e8e66609a9cc1cd5e9e19": {
    "property://donation/party": "FDP",
    "property://donation/amount": "55.886,411",
    "property://donation/donor": "Bayerische Motoren Werke AG\nPetuelring 130\n80788 München",
    "property://donation/date": "26.02.2010"
  }
}
```

Frage 2: Aufbereitung und Speicherung

Listing A.4: Ziel

```
@prefix item:        <item://> .
@prefix donation:    <property://donation/> .
@prefix section:     <https://www.bundestag.de/bundestag/parteienfinanzierung/fundstellen50000> .
@prefix donation1:   <https://www.bundestag.de/bundestag/parteienfinanzierung/fundstellen50000
                        /2010#a0b7dcad346ca3731b6e8e66609a9cc1cd5e9e19> .

section:
    item:donation   donation1: .

donation1:
    donation:party  "FDP"^^xsd:string ;
    donation:amount "55886.41"^^xsd:decimal ;
    donation:donor  "Bayerische Motoren Werke AG\nPetuelring 130\n80788 München"^^xsd:string ;
    donation:date   "26.02.2010"^^xsd:string .
```

Listing A.5: Ergebnis

```
@prefix item:        <item://> .
@prefix donation:    <property://donation/> .
@prefix section:     <https://www.bundestag.de/bundestag/parteienfinanzierung/fundstellen50000> .
@prefix donation1:   <https://www.bundestag.de/bundestag/parteienfinanzierung/fundstellen50000
                        /2010#a0b7dcad346ca3731b6e8e66609a9cc1cd5e9e19> .

section:
    item:donation   donation1: .

donation1:
    donation:party  "FDP"^^xsd:string ;
    donation:amount "55886.41"^^xsd:decimal ;
    donation:donor  "Bayerische Motoren Werke AG\nPetuelring 130\n80788 München"^^xsd:string ;
    donation:date   "26.02.2010"^^xsd:string .
```

Frage 3: Versionierung

Durchgang 1

Listing A.6: Ziel

```
@prefix :             <predicate://> .
@prefix section:      <https://www.bundestag.de/bundestag/parteienfinanzierung/fundstellen50000> .
@prefix donation1:    <https://www.bundestag.de/bundestag/parteienfinanzierung/fundstellen50000
                          /2010#a0b7dcad346ca3731b6e8e66609a9cc1cd5e9e19> .

section:
    :added donation1: .
```

Listing A.7: Nicht-Ziel

```
@prefix :             <predicate://> .
@prefix donation:     <property://donation/> .
@prefix donation1:    <https://www.bundestag.de/bundestag/parteienfinanzierung/fundstellen50000
                          /2010#a0b7dcad346ca3731b6e8e66609a9cc1cd5e9e19> .

donation1:
    :added donation:party ;
    :added donation:amount ;
    :added donation:donor ;
    :added donation:date .
```

Listing A.8: Ergebnis

```
@prefix :             <predicate://> .
@prefix section:      <https://www.bundestag.de/bundestag/parteienfinanzierung/fundstellen50000> .
@prefix donation1:    <https://www.bundestag.de/bundestag/parteienfinanzierung/fundstellen50000
                          /2010#a0b7dcad346ca3731b6e8e66609a9cc1cd5e9e19> .

section:
    :added donation1: .
```

Durchgang 2

Listing A.9: Daten

```
@prefix item:        <item://> .
@prefix donation:    <property://donation/> .
@prefix section:     <https://www.bundestag.de/bundestag/parteienfinanzierung/fundstellen50000> .
@prefix donation3:   <https://www.bundestag.de/bundestag/parteienfinanzierung/fundstellen50000
                         #901d990cf5d896bdd0744d99acfea6ba8bbe36bf> .

section:
    item:donation    donation3: .

donation3:
    donation:party   "Partei"^^xsd:string  ;
    donation:amount "123456.78"^^xsd:decimal ;
    donation:donor   "Spender"^^xsd:string ;
    donation:date    "03.04.2015"^^xsd:string  .
```

Listing A.10: Ziel

```
@prefix :            <predicate://> .
@prefix section:     <https://www.bundestag.de/bundestag/parteienfinanzierung/fundstellen50000> .
@prefix donation3:   <https://www.bundestag.de/bundestag/parteienfinanzierung/fundstellen50000
                         #901d990cf5d896bdd0744d99acfea6ba8bbe36bf> .

section:
    :removed    donation3: .
```

Listing A.11: Nicht-Ziel

```
@prefix :            <predicate://> .
@prefix donation:    <property://donation/> .
@prefix donation3:   <https://www.bundestag.de/bundestag/parteienfinanzierung/fundstellen50000
                         #901d990cf5d896bdd0744d99acfea6ba8bbe36bf> .

donation3:
    :removed    donation:party ;
    :removed    donation:amount ;
    :removed    donation:donor ;
    :removed    donation:date .
```

Listing A.12: Ergebnis

```
@prefix :            <predicate://> .
@prefix section:     <https://www.bundestag.de/bundestag/parteienfinanzierung/fundstellen50000> .
@prefix donation3:   <https://www.bundestag.de/bundestag/parteienfinanzierung/fundstellen50000
                         #901d990cf5d896bdd0744d99acfea6ba8bbe36bf> .

section:
    :removed    donation3: .
```

Anhang B

Evaluation von campus.tum.de

Listing B.1: Modell

```
@prefix :                <predicate://> .
@prefix type:            <type://> .
@prefix page:            <page://> .
@prefix link:            <link://> .
@prefix item:            <item://> .
@prefix faculty:         <property://faculty/> .
@prefix course:          <property://course/> .
@prefix appointment:     <property://appointment/> .
@prefix location:        <property://location/> .
@prefix section:         <https://campus.tum.de/tumonline/wborg.display_virtuell
                             ?PORGNR=1&PORGTYP=28567> .

<model:>
    :section     section: .

section:
    :page        page: faculties  .

page: faculties
    :type        type: links  ;
    :path        "//table[@class='nav']//a" ;
    :target      page:faculty  .

page:faculty
    :type        type:object  ;
    :item        item:faculty  ;
    :link        link:courses  .

page:courses
    :type        type: links  ;
    :path        "//div[@id='idLVOfferTable']//table[@class='cotable']/tbody/tr/td[4]//a" ;
    :next        "//div[@id='idLVOfferTable']//table[@class='coTableNavi'][1]//select
                     /following-sibling::a" ;
    :wait        7000 ;
    :target      page:course  .

page:course
    :type        type:object  ;
    :item        item:course  ;
    :link        link:appointments .

page:appointments
    :type        type:objects  ;
    :item        item:appointment ;
    :sub         page:location  .

page:location
    :type        type:objects  ;
    :item        item:location  .
```

```
link:courses
    :path       "//a[text()='Lehrveranstaltungen']" ;
    :target     page:courses .

link:appointments
    :path       "//label[text()='Abhaltungstermine']/../following-sibling::td//a" ;
    :target     page:appointments .

item:faculty
    :path       "//table[@class='nav']" ;
    :property   faculty:label .

item:course
    :path       "//div[@class=' MaskData']/table" ;
    :property   course:label ;
    :property   course:number ;
    :property   course:type .

item:appointment
    :path       "//table[@id='tabLvTermine']//tr[contains(@class, 'hi coTableR')]" ;
    :id         "./td[2]/a" ;
    :property   appointment:group ;
    :property   appointment:date ;
    :property   appointment:start ;
    :property   appointment:end .

item:location
    :path       "./td[5]/a" ;
    :id         "." ;
    :property   location:label .

faculty:label
    :path       ".//span[@class='bold']" .

course:label
    :path       ".//label[text()='Titel ']/../ following-sibling::td//span" .

course:number
    :path       ".//label[text()='Nummer']/../following-sibling::td//span" .

course:type
    :path       ".//label[text()='Art ']/../ following-sibling::td//span" .

appointment:group
    :path       "./preceding-sibling::tr[@class='coRow coTableGR1 '][1]/td" .

appointment:date
    :path       "./td[2]/a" ;
    :type       xsd:date ;
    :format     "dd.MM.yyyy" .

appointment:start
```

```
    :path       "./td[3]"  ;
    :type       xsd:time ;
    :format     "HH:mm" .

appointment:end
    :path       "./td[4]"   ;
    :type       xsd:time ;
    :format     "HH:mm" .

location:label
    :path       "."  .
```

Frage 1: Extraktion

Listing B.2: Ziel


```
{
  "https://campus.tum.de/tumonline/organisationen.display?corg=14189": {
    "property://faculty/label": "Fakultät für Informatik"
  },
  "https://campus.tum.de/tumonline/lv.detail?clvnr=950208933": {
    "property://course/label": "Tutorübungen zu Grundlagen: Betriebssysteme und
      Systemsoftware (IN0009)",
    "property://course/number": "0000000052",
    "property://course/type": "Übung"
  },
  "https://campus.tum.de/tumonline/wbTvw_List.lehrveranstaltung?pStpSpNr=950208933
      #a07732f5b533dc99a4c84010b82645698ffdc497": {
    "property://group/label": "Gruppe 07"
  },
  "https://campus.tum.de/tumonline/!wbTermin.wbEdit?pTerminNr=884835648": {
    "property://appointment/date": "11.11.2015",
    "property://appointment/start": "16:00",
    "property://appointment/end": "18:00"
  },
  "https://campus.tum.de/tumonline/ris.einzelRaum?raumKey=29234": {
    "property://location/label": "01.07.023, Seminarraum (5607.01.023)"
  }
}
```


Listing B.3: Ergebnis


```
{
  "https://campus.tum.de/tumonline/organisationen.display?corg=14189": {
    "property://faculty/label": "Fakultät für Informatik"
  },
  "https://campus.tum.de/tumonline/lv.detail?clvnr=950208933": {
    "property://course/label": "Tutorübungen zu Grundlagen: Betriebssysteme und
      Systemsoftware (IN0009)",
    "property://course/number": "0000000052",
    "property://course/type": "Übung"
  },
  "https://campus.tum.de/tumonline/!wbTermin.wbEdit?pTerminNr=884835648": {
    "property://appointment/group": "Gruppe 07",
    "property://appointment/date": "11.11.2015",
    "property://appointment/start": "16:00",
    "property://appointment/end": "18:00"
  },
  "https://campus.tum.de/tumonline/ris.einzelRaum?raumKey=29234": {
    "property://location/label": "01.07.023, Seminarraum (5607.01.023)"
  }
}
```

Frage 2: Aufbereitung und Speicherung

Listing B.4: Ziel

```
@prefix item:           <item://> .
@prefix faculty:        <property://faculty/> .
@prefix course:         <property://course/> .
@prefix group:          <property://group/> .
@prefix appointment:    <property://appointment/> .
@prefix location:       <property://location/> .
@prefix section:        <https://campus.tum.de/tumonline/wborg.display_virtuell
                            ?PORGNR=1&PORGTYP=28567> .
@prefix faculty1:       <https://campus.tum.de/tumonline/organisationen.display?corg=14189> .
@prefix course1:        <https://campus.tum.de/tumonline/lv.detail?clvnr=950208933> .
@prefix group1:         <https://campus.tum.de/tumonline/wbTvw_List.lehrveranstaltung
                            ?pStpSpNr=950208933#a07732f5b533dc99a4c84010b82645698ffdc497> .
@prefix appointment1:   <https://campus.tum.de/tumonline/!wbTermin.wbEdit
                            ?pTerminNr=884835648> .
@prefix location1:      <https://campus.tum.de/tumonline/ris.einzelRaum?raumKey=29234> .

section:
    item:faculty        faculty1: .

faculty1:
    faculty:label       "Fakultät für Informatik"^^xsd:string ;
    item:course         course1: .

course1:
    course:label        "Tutorübungen zu Grundlagen: Betriebssysteme und
                            Systemsoftware (IN0009)"^^xsd:string ;
    course:number       "0000000052"^^xsd:string ;
    course:type         "Übung"^^xsd:string ;
    item:group          group1: .

group1:
    group:label         "Gruppe 07"^^xsd:string ;
    item:appointment    appointment1: .

appointment1:
    appointment:date    "2015-11-11"^^xsd:date ;
    appointment:start   "16:00:00"^^xsd:time ;
    appointment:end     "18:00:00"^^xsd:time ;
    item:location       location1: .

location1:
    location:label      "01.07.023, Seminarraum (5607.01.023)"^^xsd:string .
```

Listing B.5: Ergebnis

```
@prefix item:           <item://> .
@prefix faculty:        <property://faculty/> .
@prefix course:         <property://course/> .
@prefix appointment:    <property://appointment/> .
@prefix location:       <property://location/> .
@prefix section:        <https://campus.tum.de/tumonline/wborg.display_virtuell
                            ?PORGNR=1&PORGTYP=28567> .
@prefix faculty1:       <https://campus.tum.de/tumonline/organisationen.display?corg=14189> .
@prefix course1:        <https://campus.tum.de/tumonline/lv.detail?clvnr=950208933> .
@prefix appointment1:   <https://campus.tum.de/tumonline/!wbTermin.wbEdit
                            ?pTerminNr=884835648> .
@prefix location1:      <https://campus.tum.de/tumonline/ris.einzelRaum?raumKey=29234> .

section:
    item:faculty        faculty1: .

faculty1:
    faculty:label       "Fakultät für Informatik"^^xsd:string ;
    item:course         course1: .

course1:
    course:label        "Tutorübungen zu Grundlagen: Betriebssysteme und
                            Systemsoftware (IN0009)"^^xsd:string ;
    course:number       "0000000052"^^xsd:string ;
    course:type         "Übung"^^xsd:string ;
    item:appointment    appointment1: .

appointment1:
    appointment:group   "Gruppe 07"^^xsd:string ;
    appointment:date    "2015-11-11"^^xsd:date ;
    appointment:start   "16:00:00.000"^^xsd:time ;
    appointment:end     "18:00:00.000"^^xsd:time ;
    item:location       location1: .

location1:
    location:label      "01.07.023, Seminarraum (5607.01.023)"^^xsd:string .
```

Frage 3: Versionierung

Durchgang 1

Listing B.6: Ziel

```
@prefix :            <predicate://> .
@prefix section:     <https://campus.tum.de/tumonline/wborg.display_virtuell
                         ?PORGNR=1&PORGTYP=28567> .
@prefix faculty1:    <https://campus.tum.de/tumonline/organisationen.display?corg=14189> .

section:
    :added faculty1: .
```

Listing B.7: Nicht-Ziel

```
@prefix :            <predicate://> .
@prefix faculty:     <property://faculty/> .
@prefix course:      <property://course/> .
@prefix group:       <property://group/> .
@prefix appointment: <property://appointment/> .
@prefix location:    <property://location/> .
@prefix faculty1:    <https://campus.tum.de/tumonline/organisationen.display?corg=14189> .
@prefix course1:     <https://campus.tum.de/tumonline/lv.detail?clvnr=950208933> .
@prefix group1:      <https://campus.tum.de/tumonline/wbTvw_List.lehrveranstaltung
                         ?pStpSpNr=950208933#a07732f5b533dc99a4c84010b82645698ffdc497> .
@prefix appointment1: <https://campus.tum.de/tumonline/!wbTermin.wbEdit
                         ?pTerminNr=884835648> .
@prefix location1:   <https://campus.tum.de/tumonline/ris.einzelRaum?raumKey=29234> .

faculty1:
    :added faculty:label ;
    :added course1: .

course1:
    :added course:label ;
    :added group1: .

group1:
    :added group:label ;
    :added appointment1: .

appointment1:
    :added appointment:date ;
    :added location1: .

location1:
    :added location:label .
```

Listing B.8: Ergebnis

```
@prefix :              <predicate://> .
@prefix section:       <https://campus.tum.de/tumonline/wborg.display_virtuell
                           ?PORGNR=1&PORGTYP=28567> .
@prefix faculty1:      <https://campus.tum.de/tumonline/organisationen.display?corg=14189> .

section:
    :added  faculty1: .
```

Durchgang 2

Listing B.9: Daten

```
@prefix item:           <item://> .
@prefix faculty:        <property://faculty/> .
@prefix course:         <property://course/> .
@prefix appointment:    <property://appointment/> .
@prefix section:        <https://campus.tum.de/tumonline/wborg.display_virtuell
                            ?PORGNR=1&PORGTYP=28567> .
@prefix faculty1:       <https://campus.tum.de/tumonline/organisationen.display?corg=14189> .
@prefix appointment1:   <https://campus.tum.de/tumonline/!wbTermin.wbEdit
                            ?pTerminNr=884835648> .
@prefix course2:        <https://campus.tum.de/tumonline/lv.detail?clvnr=123456789> .
@prefix faculty2:       <https://campus.tum.de/tumonline/organisationen.display?corg=12345> .
@prefix course3:        <https://campus.tum.de/tumonline/lv.detail?clvnr=987654321> .

appointment1:
    appointment:start    "16:15:00.000"^^xsd:time .

faculty1:
    item:course          course2:  .

course2:
    course:label         "Lehrveranstaltung"^^xsd:string ;
    course:number        "0000012345"^^xsd:string ;
    course:type          "Typ"^^xsd:string .

section:
    item:faculty         faculty2:  .

faculty2:
    faculty:label        "Fakultät"^^xsd:string ;
    item:course          course3:  .

course3:
    course:label         "Lehrveranstaltung"^^xsd:string ;
    course:number        "0000054321"^^xsd:string ;
    course:type          "Typ"^^xsd:string .
```

Listing B.10: Ziel

```
@prefix :               <predicate://> .
@prefix appointment:    <property://appointment/> .
@prefix section:        <https://campus.tum.de/tumonline/wborg.display_virtuell
                            ?PORGNR=1&PORGTYP=28567> .
@prefix faculty1:       <https://campus.tum.de/tumonline/organisationen.display?corg=14189> .
@prefix appointment1:   <https://campus.tum.de/tumonline/!wbTermin.wbEdit
                            ?pTerminNr=884835648> .
@prefix course2:        <https://campus.tum.de/tumonline/lv.detail?clvnr=123456789> .
@prefix faculty2:       <https://campus.tum.de/tumonline/organisationen.display?corg=12345> .
```

```
appointment1:
    :changed    appointment:start .

faculty1:
    :removed    course2: .

section:
    :removed    faculty2: .
```

Listing B.11: Nicht-Ziel

```
@prefix :              <predicate://> .
@prefix faculty:       <property://faculty/> .
@prefix course:        <property://course/> .
@prefix course2:       <https://campus.tum.de/tumonline/lv.detail?clvnr=123456789> .
@prefix faculty2:      <https://campus.tum.de/tumonline/organisationen.display?corg=12345> .
@prefix course3:       <https://campus.tum.de/tumonline/lv.detail?clvnr=987654321> .

course2:
    :removed    course:label .

faculty2:
    :removed    faculty:label ;
    :removed    course3: .

course3:
    :removed    course:label .
```

Listing B.12: Ergebnis

```
@prefix :                  <predicate://> .
@prefix appointment:       <property://appointment/> .
@prefix section:           <https://campus.tum.de/tumonline/wborg.display_virtuell
                               ?PORGNR=1&PORGTYP=28567> .
@prefix faculty1:          <https://campus.tum.de/tumonline/organisationen.display?corg=14189> .
@prefix appointment1:      <https://campus.tum.de/tumonline/!wbTermin.wbEdit
                               ?pTerminNr=884835648> .
@prefix course2:           <https://campus.tum.de/tumonline/lv.detail?clvnr=123456789> .
@prefix faculty2:          <https://campus.tum.de/tumonline/organisationen.display?corg=12345> .

appointment1:
    :changed    appointment:start .

faculty1:
    :removed    course2: .

section:
    :removed    faculty2: .
```

Anhang C

Evaluation von chefkoch.de

Listing C.1: Modell

```
@prefix :             <predicate://> .
@prefix type:         <type://> .
@prefix page:         <page://> .
@prefix item:         <item://> .
@prefix recipe:       <property://recipe/> .
@prefix ingredient:   <property://ingredient/> .
@prefix section1:     <http://www.chefkoch.de/rs/s0/Donauwelle/Rezepte.html> .
@prefix section2:     <http://www.chefkoch.de/rs/s0/Kaiserschmarrn/Rezepte.html> .

<model:>
    :section      section1: ;
    :section      section2: .

section1:
    :page         page:recipes  .

section2:
    :page         page:recipes  .

page:recipes
    :type         type:links  ;
    :path         "//table[@class='search-result-table recipe-result']//a[@class='search-result-title']"  ;
    :next         "//a[@class='pagination-item pagination-next']" ;
    :target       page:recipe  .

page:recipe
    :type         type:object  ;
    :item         item:recipe  ;
    :sub          page:ingredients  .

page:ingredients
    :type         type:objects  ;
    :item         item:ingredient  .

item:recipe
    :path         "//div[@id='page']" ;
    :property     recipe:label  ;
    :property     recipe:workingTime ;
    :property     recipe: difficulty  ;
    :property     recipe:instructions  ;
    :property     recipe:portions ;
    :property     recipe:rating  .

item:ingredient
    :path         ".//div[@id='recipe-incredients']//tr[not(.//b)]"  ;
    :property     ingredient:label  ;
    :property     ingredient:amount .

recipe:label
    :path         ".//h1" .
```

```
recipe:workingTime
    :path       ".//span[@class='prepTime']/span" ;
    :attribute  " title " ;
    :type       xsd:duration .

recipe: difficulty
    :path       ".//h2[text()='Zubereitung']/following-sibling::p" ;
    :pattern    "(?<=Schwierigkeitsgrad: ).+?(?= /)" .

recipe:instructions
    :path       ".//div[@id='rezept-zubereitung']" .

recipe:portions
    :path       ".//input[@id='divisor']" ;
    :attribute  "value" ;
    :type       xsd:int .

recipe:rating
    :path       ".//div[@id='rezept-bewertung-box']//span[starts-with(@class, 'rating ')]
                    [not(contains(@class, 'rating-unbewertet'))]" ;
    :optional   true ;
    :attribute  "class" ;
    :pattern    "(?<=rating-)\d(_\d)?" ;
    :replace [
        :old    "_" ;
        :new    "."
    ] ;
    :type       xsd:decimal .

ingredient:label
    :path       "./td[@class='name']" .

ingredient:amount
    :path       "./td[@class='amount']" .
```

Frage 1: Extraktion

Listing C.2: Ziel


```
{
  "http://www.chefkoch.de/rezepte/372611122991708/Donauwelle-Konditorenart.html": {
    "property://recipe/label": "Donauwelle Konditorenart",
    "property://recipe/workingTime": "PT60M",
    "property://recipe/ difficulty ": "Arbeitszeit: ca. 1 Std. / Koch-/Backzeit: ca. 30 Min.
      / Schwierigkeitsgrad: normal / Kalorien p. P.: keine Angabe",
    "property://recipe/instructions": "Für den Teig 350 g Butter mit 250 g Zucker schaumig [...]",
    "property://recipe/portions": "1",
    "property://recipe/rating": "rating rating-4_5"
  },
  "http://www.chefkoch.de/rezepte/372611122991708/Donauwelle-Konditorenart.html
      #aeabf42db4c00f93ee072e77e940e4d3e11c656f": {
    "property://ingredient/label": "Zucker",
    "property://ingredient/amount": "250 g"
  },
  "http://www.chefkoch.de/rezepte/2309811368533750/Kaiserschmarrn-la-Mittermeier.html": {
    "property://recipe/label": "Kaiserschmarrn à la Mittermeier",
    "property://recipe/workingTime": "PT15M",
    "property://recipe/ difficulty ": "Arbeitszeit: ca. 15 Min. / Koch-/Backzeit: ca. 17 Min.
      / Schwierigkeitsgrad: simpel / Kalorien p. P.: keine Angabe",
    "property://recipe/instructions": "Und so geht's:\n1. Vorbereitung: Abrieb von der Zitrone [...]",
    "property://recipe/portions": "4"
  },
  "http://www.chefkoch.de/rezepte/2309811368533750/Kaiserschmarrn-la-Mittermeier.html
      #93255f1deef85ce566d727a2a04528d59da21f5c": {
    "property://ingredient/label": "Rosinen",
    "property://ingredient/amount": "2 EL"
  }
}
```

Listing C.3: Ergebnis


```
{
  "http://www.chefkoch.de/rezepte/372611122991708/Donauwelle-Konditorenart.html": {
    "property://recipe/label": "Donauwelle Konditorenart",
    "property://recipe/workingTime": "PT60M",
    "property://recipe/ difficulty ": "Arbeitszeit: ca. 1 Std. / Koch-/Backzeit: ca. 30 Min.
      / Schwierigkeitsgrad: normal / Kalorien p. P.: keine Angabe",
    "property://recipe/instructions": "Für den Teig 350 g Butter mit 250 g Zucker schaumig [...]",
    "property://recipe/portions": "1",
    "property://recipe/rating": "rating rating-4_5"
  },
  "http://www.chefkoch.de/rezepte/372611122991708/Donauwelle-Konditorenart.html
      #aeabf42db4c00f93ee072e77e940e4d3e11c656f": {
    "property://ingredient/label": "Zucker",
    "property://ingredient/amount": "250 g"
  },
  "http://www.chefkoch.de/rezepte/2309811368533750/Kaiserschmarrn-la-Mittermeier.html": {
    "property://recipe/label": "Kaiserschmarrn à la Mittermeier",
    "property://recipe/workingTime": "PT15M",
    "property://recipe/ difficulty ": "Arbeitszeit: ca. 15 Min. / Koch-/Backzeit: ca. 17 Min.
      / Schwierigkeitsgrad: simpel / Kalorien p. P.: keine Angabe",
    "property://recipe/instructions": "Und so geht's:\n1. Vorbereitung: Abrieb von der Zitrone [...]",
    "property://recipe/portions": "4"
  },
  "http://www.chefkoch.de/rezepte/2309811368533750/Kaiserschmarrn-la-Mittermeier.html
      #93255f1deef85ce566d727a2a04528d59da21f5c": {
    "property://ingredient/label": "Rosinen",
    "property://ingredient/amount": "2 EL"
  }
}
```

Frage 2: Aufbereitung und Speicherung

Listing C.4: Ziel

```
@prefix item:          <item://> .
@prefix recipe:        <property://recipe/> .
@prefix ingredient:    <property://ingredient/> .
@prefix section1:      <http://www.chefkoch.de/rs/s0/Donauwelle/Rezepte.html> .
@prefix section2:      <http://www.chefkoch.de/rs/s0/Kaiserschmarrn/Rezepte.html> .
@prefix recipe1:       <http://www.chefkoch.de/rezepte/372611122991708/Donauwelle
                          -Konditorenart.html> .
@prefix ingredient1:   <http://www.chefkoch.de/rezepte/372611122991708/Donauwelle
                          -Konditorenart.html#aeabf42db4c00f93ee072e77e940e4d3e11c656f> .
@prefix recipe2:       <http://www.chefkoch.de/rezepte/2309811368533750/Kaiserschmarrn-la
                          -Mittermeier.html> .
@prefix ingredient2:   <http://www.chefkoch.de/rezepte/2309811368533750/Kaiserschmarrn-la
                          -Mittermeier.html#93255f1deef85ce566d727a2a04528d59da21f5c> .

section1:
    item:recipe            recipe1: .

section2:
    item:recipe            recipe2: .

recipe1:
    recipe:label           "Donauwelle Konditorenart"^^xsd:string ;
    recipe:workingTime     "PT60M"^^xsd:duration ;
    recipe: difficulty     "normal"^^xsd:string ;
    recipe:instructions    "Für den Teig 350 g Butter mit 250 g Zucker schaumig [...]"^^xsd:string ;
    recipe:portions        "1"^^xsd:int ;
    recipe:rating          "4.5"^^xsd:decimal ;
    item:ingredient        ingredient1: .

ingredient1:
    ingredient:label       "Zucker"^^xsd:string ;
    ingredient:amount      "250 g"^^xsd:string .

recipe2:
    recipe:label           "Kaiserschmarrn à la Mittermeier"^^xsd:string ;
    recipe:workingTime     "PT15M"^^xsd:duration ;
    recipe: difficulty     "simpel"^^xsd:string ;
    recipe:instructions    "Und so geht's:\n1. Vorbereitung: Abrieb von der Zitrone [...]"^^ xsd:string ;
    recipe:portions        "4"^^xsd:int ;
    item:ingredient        ingredient2: .

ingredient2:
    ingredient:label       "Rosinen"^^xsd:string ;
    ingredient:amount      "2 EL"^^xsd:string .
```

Listing C.5: Ergebnis

```
@prefix item:          <item://> .
@prefix recipe:        <property://recipe/> .
@prefix ingredient:    <property://ingredient/> .
@prefix section1:      <http://www.chefkoch.de/rs/s0/Donauwelle/Rezepte.html> .
@prefix section2:      <http://www.chefkoch.de/rs/s0/Kaiserschmarrn/Rezepte.html> .
@prefix recipe1:       <http://www.chefkoch.de/rezepte/372611122991708/Donauwelle
                         -Konditorenart.html> .
@prefix ingredient1:   <http://www.chefkoch.de/rezepte/372611122991708/Donauwelle
                         -Konditorenart.html#aeabf42db4c00f93ee072e77e940e4d3e11c656f> .
@prefix recipe2:       <http://www.chefkoch.de/rezepte/2309811368533750/Kaiserschmarrn-la
                         -Mittermeier.html> .
@prefix ingredient2:   <http://www.chefkoch.de/rezepte/2309811368533750/Kaiserschmarrn-la
                         -Mittermeier.html#93255f1deef85ce566d727a2a04528d59da21f5c> .

section1:
    item:recipe          recipe1: .

section2:
    item:recipe          recipe2: .

recipe1:
    recipe:label         "Donauwelle Konditorenart"^^xsd:string ;
    recipe:workingTime   "PT1H"^^xsd:duration ;
    recipe: difficulty   "normal"^^xsd:string ;
    recipe:instructions  "Für den Teig 350 g Butter mit 250 g Zucker schaumig [...]"^^xsd:string ;
    recipe:portions      "1"^^xsd:int ;
    recipe:rating        "4.5"^^xsd:decimal ;
    item:ingredient      ingredient1: .

ingredient1:
    ingredient:label     "Zucker"^^xsd:string ;
    ingredient:amount    "250 g"^^xsd:string .

recipe2:
    recipe:label         "Kaiserschmarrn à la Mittermeier"^^xsd:string ;
    recipe:workingTime   "PT15M"^^xsd:duration ;
    recipe: difficulty   "simpel"^^xsd:string ;
    recipe:instructions  "Und so geht's:\n1. Vorbereitung: Abrieb von der Zitrone [...]"^^xsd:string ;
    recipe:portions      "4"^^xsd:int ;
    item:ingredient      ingredient2: .

ingredient2:
    ingredient:label     "Rosinen"^^xsd:string ;
    ingredient:amount    "2 EL"^^xsd:string .
```

Frage 3: Versionierung

Durchgang 1

Listing C.6: Ziel

```
@prefix :              <predicate://> .
@prefix section1:      <http://www.chefkoch.de/rs/s0/Donauwelle/Rezepte.html> .
@prefix section2:      <http://www.chefkoch.de/rs/s0/Kaiserschmarrn/Rezepte.html> .
@prefix recipe1:       <http://www.chefkoch.de/rezepte/372611122991708/Donauwelle
                           -Konditorenart.html> .
@prefix recipe2:       <http://www.chefkoch.de/rezepte/2309811368533750/Kaiserschmarrn-la
                           -Mittermeier.html> .

section1:
    :added recipe1: .

section2:
    :added recipe2: .
```

Listing C.7: Nicht-Ziel

```
@prefix :              <predicate://> .
@prefix recipe:        <property://recipe/> .
@prefix ingredient:    <property://ingredient/> .
@prefix recipe1:       <http://www.chefkoch.de/rezepte/372611122991708/Donauwelle
                           -Konditorenart.html> .
@prefix ingredient1:   <http://www.chefkoch.de/rezepte/372611122991708/Donauwelle
                           -Konditorenart.html#aeabf42db4c00f93ee072e77e940e4d3e11c656f> .

recipe1:
    :added recipe:label ;
    :added ingredient1: .

ingredient1:
    :added ingredient:label .
```

Listing C.8: Ergebnis

```
@prefix :              <predicate://> .
@prefix section1:      <http://www.chefkoch.de/rs/s0/Donauwelle/Rezepte.html> .
@prefix section2:      <http://www.chefkoch.de/rs/s0/Kaiserschmarrn/Rezepte.html> .
@prefix recipe1:       <http://www.chefkoch.de/rezepte/372611122991708/Donauwelle
                           -Konditorenart.html> .
@prefix recipe2:       <http://www.chefkoch.de/rezepte/2309811368533750/Kaiserschmarrn-la
                           -Mittermeier.html> .

section1:
    :added recipe1: .

section2:
    :added recipe2: .
```

Durchgang 2

Listing C.9: Daten

```
@prefix item:           <item://> .
@prefix recipe:         <property://recipe/> .
@prefix ingredient:     <property://ingredient/> .
@prefix section1:       <http://www.chefkoch.de/rs/s0/Donauwelle/Rezepte.html> .
@prefix recipe1:        <http://www.chefkoch.de/rezepte/372611122991708/Donauwelle
                            -Konditorenart.html> .
@prefix ingredient3:    <http://www.chefkoch.de/rezepte/372611122991708/Donauwelle
                            -Konditorenart.html#1372ed70264a942badcb4453385974418a25c5df> .
@prefix recipe2:        <http://www.chefkoch.de/rezepte/2309811368533750/Kaiserschmarrn-la
                            -Mittermeier.html> .
@prefix recipe3:        <http://www.chefkoch.de/rezepte/1234567890/Rezept.html> .
@prefix ingredient4:    <http://www.chefkoch.de/rezepte/1234567890/Rezept.html
                            #1372ed70264a942badcb4453385974418a25c5df> .

recipe1:
    recipe:rating       "4"^^xsd:decimal ;
    item:ingredient     ingredient3: .

ingredient3:
    ingredient:label    "Zutat"^^xsd:string ;
    ingredient:amount   "Menge"^^xsd:string .

recipe2:
    recipe:rating       "2"^^xsd:decimal .

section1:
    item:recipe         recipe3: .

recipe3:
    recipe:label        "Rezept"^^xsd:string ;
    recipe:workingTime  "PT30M"^^xsd:duration ;
    recipe: difficulty  "Schwierigkeitsgrad"^^xsd:string ;
    recipe:instructions "Zubereitung"^^xsd:string ;
    recipe:portions     "3"^^xsd:int ;
    item:ingredient     ingredient4: .

ingredient4:
    ingredient:label    "Zutat"^^xsd:string ;
    ingredient:amount   "Menge"^^xsd:string .
```

Listing C.10: Ziel

```
@prefix :                <predicate://> .
@prefix recipe:          <property://recipe/> .
@prefix section1:        <http://www.chefkoch.de/rs/s0/Donauwelle/Rezepte.html> .
@prefix recipe1:         <http://www.chefkoch.de/rezepte/372611122991708/Donauwelle
                             -Konditorenart.html> .
@prefix ingredient3:     <http://www.chefkoch.de/rezepte/372611122991708/Donauwelle
                             -Konditorenart.html#1372ed70264a942badcb4453385974418a25c5df> .
@prefix recipe2:         <http://www.chefkoch.de/rezepte/2309811368533750/Kaiserschmarrn-la
                             -Mittermeier.html> .
@prefix recipe3:         <http://www.chefkoch.de/rezepte/1234567890/Rezept.html> .
@prefix ingredient4:     <http://www.chefkoch.de/rezepte/1234567890/Rezept.html
                             #1372ed70264a942badcb4453385974418a25c5df> .

recipe1:
    :changed    recipe:rating ;
    :removed    ingredient3: .

recipe2:
    :removed    recipe:rating .

section1:
    :removed    recipe3: .
```

Listing C.11: Nicht-Ziel

```
@prefix :                <predicate://> .
@prefix recipe:          <property://recipe/> .
@prefix ingredient:      <property://ingredient/> .
@prefix ingredient3:     <http://www.chefkoch.de/rezepte/372611122991708/Donauwelle
                             -Konditorenart.html#1372ed70264a942badcb4453385974418a25c5df> .
@prefix recipe3:         <http://www.chefkoch.de/rezepte/1234567890/Rezept.html> .
@prefix ingredient4:     <http://www.chefkoch.de/rezepte/1234567890/Rezept.html
                             #1372ed70264a942badcb4453385974418a25c5df> .

ingredient3:
    :removed    ingredient:label .

recipe3:
    :removed    recipe:label ;
    :removed    ingredient4: .
```

Listing C.12: Ergebnis

```
@prefix :                 <predicate://> .
@prefix recipe:           <property://recipe/> .
@prefix section1:         <http://www.chefkoch.de/rs/s0/Donauwelle/Rezepte.html> .
@prefix recipe1:          <http://www.chefkoch.de/rezepte/372611122991708/Donauwelle
                              -Konditorenart.html> .
@prefix ingredient3:      <http://www.chefkoch.de/rezepte/372611122991708/Donauwelle
                              -Konditorenart.html#1372ed70264a942badcb4453385974418a25c5df> .
@prefix recipe2:          <http://www.chefkoch.de/rezepte/2309811368533750/Kaiserschmarrn-la
                              -Mittermeier.html> .
@prefix recipe3:          <http://www.chefkoch.de/rezepte/1234567890/Rezept.html> .
@prefix ingredient4:      <http://www.chefkoch.de/rezepte/1234567890/Rezept.html
                              #1372ed70264a942badcb4453385974418a25c5df> .

recipe1:
    :changed    recipe:rating ;
    :removed    ingredient3: .

recipe2:
    :removed    recipe:rating .

section1:
    :removed    recipe3: .
```

Anhang D

Evaluation von imdb.com

Listing D.1: Modell

```
@prefix :            <predicate://> .
@prefix type:        <type://> .
@prefix page:        <page://> .
@prefix item:        <item://> .
@prefix series:      <property://series/> .
@prefix actor:       <property://actor/> .
@prefix season:      <property://season/> .
@prefix episode:     <property://episode/> .
@prefix section:     <http://www.imdb.com/chart/toptv/> .

<model:>
    :section     section:  .

section:
    :page        page: seriess  .

page: seriess
    :type        type: links  ;
    :path        "//table[@class='chart full-width']//a[@title]"  ;
    :target      page: series  .

page: series
    :type        type:object  ;
    :item        item: series  ;
    :sub         page:actors  ;
    :sub         page:seasons  ;
    :param       "pf_rd_m"  ;
    :param       "pf_rd_p"  ;
    :param       "pf_rd_r"  ;
    :param       "pf_rd_s"  ;
    :param       "pf_rd_t"  ;
    :param       "pf_rd_i"  ;
    :param       "ref_"  .

page:actors
    :type        type:objects  ;
    :item        item:actor  ;
    :param       "ref_"  .

page:seasons
    :type        type: links  ;
    :path        ".//div[@id='title-episode-widget']//a[contains(@href, '?season=')]
                     [not(contains(@href, '?season=-1'))]"  ;
    :target      page:season  .

page:season
    :type        type:object  ;
    :item        item:season  ;
    :sub         page:episodes  ;
    :param       "ref_"  .
```

```
page:episodes
    :type       type:links  ;
    :path       ".//a[@itemprop='name']" ;
    :target     page:episode .

page:episode
    :type       type:object  ;
    :item       item:episode ;
    :param      "ref_" .

item:series
    :path       "//div[@id='content-2-wide']" ;
    :property   series:label  ;
    :property   series:rating  ;
    :property   series:description  ;
    :property   series:poster  .

item:actor
    :path       ".//div[@itemprop='actors']/a" ;
    :id         "." ;
    :property   actor:label  .

item:season
    :path       "//div[@id='episodes_content']" ;
    :property   season:number .

item:episode
    :path       "//td[@id='overview-top']" ;
    :property   episode:label  ;
    :property   episode:number ;
    :property   episode:rating  .

series:label
    :path       ".//h1/span[1]" .

series:rating
    :path       ".//span[@itemprop='ratingValue']" ;
    :replace  [
        :old    "," ;
        :new    "."
    ] ;
    :type       xsd:decimal .

series:description
    :path       ".//p[@itemprop='description']" .

series:poster
    :path       ".//td[@id='img_primary']//img" ;
    :optional   true  ;
    :attribute  "src" ;
    :type       xsd:anyURI .
```

```
actor:label
    :path       "./span" .

season:number
    :path       ".//h3[@id='episode_top']" ;
    :pattern    "(?<=Season )\d+?" ;
    :type       xsd:int  .

episode:label
    :path       "./h1/span[1]" .

episode:number
    :path       "./h2/span" ;
    :pattern    "(?<=Episode )\d+?" ;
    :type       xsd:int  .

episode:rating
    :path       ".//span[@itemprop='ratingValue']" ;
    :optional   true ;
    :replace  [
        :old    "," ;
        :new    "."
    ] ;
    :type       xsd:decimal .
```

Frage 1: Extraktion

Listing D.2: Ziel


```
{
  "http://www.imdb.com/title/tt0903747/": {
    "property://series/label": "Breaking Bad",
    "property://series/rating": "9,5",
    "property://series/description": "A chemistry teacher diagnosed with terminal lung cancer
      teams up with his former student to cook and sell crystal meth.",
    "property://series/poster": "http://ia.media-imdb.com/images/M/MV5BMTQ0ODYz
      ODc0OV5BMl5BanBnXkFtZTgwMDk3OTcyMDE@._V1_SY317_CR0,0,214,317_AL_.jpg"
  },
  "http://www.imdb.com/name/nm0186505/": {
    "property://actor/label": "Bryan Cranston"
  },
  "http://www.imdb.com/name/nm0666739/": {
    "property://actor/label": "Aaron Paul"
  },
  "http://www.imdb.com/name/nm0348152/": {
    "property://actor/label": "Anna Gunn"
  },
  "http://www.imdb.com/title/tt0903747/episodes?season=3": {
    "property://season/number": "Season 3"
  },
  "http://www.imdb.com/title/tt1615556/": {
    "property://episode/label": "Sunset",
    "property://episode/number": "Season 3, Episode 6",
    "property://episode/rating": "7,9"
  }
}
```

Listing D.3: Ergebnis


```
{
  "http://www.imdb.com/title/tt0903747/": {
    "property://series/label": "Breaking Bad",
    "property://series/rating": "9,5",
    "property://series/description": "A chemistry teacher diagnosed with terminal lung cancer
      teams up with his former student to cook and sell crystal meth.",
    "property://series/poster": "http://ia.media-imdb.com/images/M/MV5BMTQ0ODYz
      ODc0OV5BMl5BanBnXkFtZTgwMDk3OTcyMDE@._V1_SY317_CR0,0,214,317_AL_.jpg"
  },
  "http://www.imdb.com/name/nm0186505/": {
    "property://actor/label": "Bryan Cranston"
  },
  "http://www.imdb.com/name/nm0666739/": {
    "property://actor/label": "Aaron Paul"
  },
  "http://www.imdb.com/name/nm0348152/": {
    "property://actor/label": "Anna Gunn"
  },
  "http://www.imdb.com/title/tt0903747/episodes?season=3": {
    "property://season/number": "Season 3"
  },
  "http://www.imdb.com/title/tt1615556/": {
    "property://episode/label": "Sunset",
    "property://episode/number": "Season 3, Episode 6",
    "property://episode/rating": "7,9"
  }
}
```

Frage 2: Aufbereitung und Speicherung

Listing D.4: Ziel

```
@prefix item:        <item://> .
@prefix series:      <property://series/> .
@prefix actor:       <property://actor/> .
@prefix season:      <property://season/> .
@prefix episode:     <property://episode/> .
@prefix section:     <http://www.imdb.com/chart/toptv/> .
@prefix series1:     <http://www.imdb.com/title/tt0903747/> .
@prefix actor1:      <http://www.imdb.com/name/nm0186505/> .
@prefix actor2:      <http://www.imdb.com/name/nm0666739/> .
@prefix actor3:      <http://www.imdb.com/name/nm0348152/> .
@prefix season1:     <http://www.imdb.com/title/tt0903747/episodes?season=3> .
@prefix episode1:    <http://www.imdb.com/title/tt1615556/> .

section:
    item:series          series1: .

series1:
    series:label         "Breaking Bad"^^xsd:string ;
    series:rating        "9.5"^^xsd:decimal ;
    series:description   "A chemistry teacher diagnosed with terminal lung cancer teams up with
                          his former student to cook and sell crystal meth."^^xsd:string ;
    series:poster        <http://ia.media-imdb.com/images/M/MV5BMTQ0ODYzODc0OV5BMl5B
                          anBnXkFtZTgwMDk3OTcyMDE@._V1_SY317_CR0,0,214,317_AL_.jpg> ;
    item:actor           actor1: ;
    item:actor           actor2: ;
    item:actor           actor3: ;
    item:season          season1: .

actor1:
    actor:label          "Bryan Cranston"^^xsd:string .

actor2:
    actor:label          "Aaron Paul"^^xsd:string .

actor3:
    actor:label          "Anna Gunn"^^xsd:string .

season1:
    season:number        "3"^^xsd:int ;
    item:episode         episode1: .

episode1:
    episode:label        "Sunset"^^xsd:string ;
    episode:number       "6"^^xsd:int ;
    episode:rating       "7.9"^^xsd:decimal .
```

Listing D.5: Ergebnis

```
@prefix item:       <item://> .
@prefix series:     <property://series/> .
@prefix actor:      <property://actor/> .
@prefix season:     <property://season/> .
@prefix episode:    <property://episode/> .
@prefix section:    <http://www.imdb.com/chart/toptv/> .
@prefix series1:    <http://www.imdb.com/title/tt0903747/> .
@prefix actor1:     <http://www.imdb.com/name/nm0186505/> .
@prefix actor2:     <http://www.imdb.com/name/nm0666739/> .
@prefix actor3:     <http://www.imdb.com/name/nm0348152/> .
@prefix season1:    <http://www.imdb.com/title/tt0903747/episodes?season=3> .
@prefix episode1:   <http://www.imdb.com/title/tt1615556/> .

section:
    item:series          series1: .

series1:
    series:label         "Breaking Bad"^^xsd:string ;
    series:rating        "9.5"^^xsd:decimal ;
    series:description   "A chemistry teacher diagnosed with terminal lung cancer teams up with
                          his former student to cook and sell crystal meth."^^xsd:string ;
    series:poster        <http://ia.media-imdb.com/images/M/MV5BMTQ0ODYzODc0OV5BMl5B
                          anBnXkFtZTgwMDk3OTcyMDE@._V1_SY317_CR0,0,214,317_AL_.jpg> ;
    item:actor           actor1: ;
    item:actor           actor2: ;
    item:actor           actor3: ;
    item:season          season1: .

actor1:
    actor:label          "Bryan Cranston"^^xsd:string .

actor2:
    actor:label          "Aaron Paul"^^xsd:string .

actor3:
    actor:label          "Anna Gunn"^^xsd:string .

season1:
    season:number        "3"^^xsd:int ;
    item:episode         episode1: .

episode1:
    episode:label        "Sunset"^^xsd:string ;
    episode:number       "6"^^xsd:int ;
    episode:rating       "7.9"^^xsd:decimal .
```

Frage 3: Versionierung

Durchgang 1

Listing D.6: Ziel

```
@prefix :              <predicate://> .
@prefix section:       <http://www.imdb.com/chart/toptv/> .
@prefix series1:       <http://www.imdb.com/title/tt0903747/> .

section:
    :added  series1:  .
```

Listing D.7: Nicht-Ziel

```
@prefix :              <predicate://> .
@prefix series:        <property://series/> .
@prefix actor:         <property://actor/> .
@prefix season:        <property://season/> .
@prefix episode:       <property://episode/> .
@prefix series1:       <http://www.imdb.com/title/tt0903747/> .
@prefix actor1:        <http://www.imdb.com/name/nm0186505/> .
@prefix season1:       <http://www.imdb.com/title/tt0903747/episodes?season=3> .
@prefix episode1:      <http://www.imdb.com/title/tt1615556/> .

series1:
    :added  series:label  ;
    :added  actor1:  ;
    :added  season1:  .

actor1:
    :added  actor:label  .

season1:
    :added  season:number ;
    :added  episode1:  .

episode1:
    :added  episode:label  .
```

Listing D.8: Ergebnis

```
@prefix :              <predicate://> .
@prefix section:       <http://www.imdb.com/chart/toptv/> .
@prefix series1:       <http://www.imdb.com/title/tt0903747/> .

section:
    :added  series1:  .
```

Durchgang 2

Listing D.9: Daten

```
@prefix item:       <item://> .
@prefix series:     <property://series/> .
@prefix season:     <property://season/> .
@prefix episode:    <property://episode/> .
@prefix actor:      <property://actor/> .
@prefix section:    <http://www.imdb.com/chart/toptv/> .
@prefix series1:    <http://www.imdb.com/title/tt0903747/> .
@prefix season2:    <http://www.imdb.com/title/tt0903747/episodes?season=6> .
@prefix series2:    <http://www.imdb.com/title/tt123456789/> .
@prefix actor4:     <http://www.imdb.com/name/nm123456789/> .
@prefix season3:    <http://www.imdb.com/title/tt123456789/episodes?season=1> .
@prefix episode2:   <http://www.imdb.com/title/tt987654321/> .

series1:
    series:rating       "9.4"^^xsd:decimal ;
    item:season         season2: .

season2:
    season:number       "6"^^xsd:int .

section:
    item:series         series2: .

series2:
    series:label        "Serie"^^xsd:string ;
    series:rating       "8.7"^^xsd:decimal ;
    series:description  "Beschreibung"^^xsd:string ;
    series:poster       <http://ia.media-imdb.com/images/M/abc.jpg> ;
    item:actor          actor4: ;
    item:season         season3: .

actor4:
    actor:label         "Schauspieler"^^xsd:string .

season3:
    season:number       "1"^^xsd:int ;
    item:episode        episode2: .

episode2:
    episode:label       "Episode"^^xsd:string ;
    episode:number      "1"^^xsd:int ;
    episode:rating      "7.6"^^xsd:decimal .
```

Listing D.10: Ziel

```
@prefix :              <predicate://> .
@prefix series:        <property://series/> .
@prefix section:       <http://www.imdb.com/chart/toptv/> .
@prefix series1:       <http://www.imdb.com/title/tt0903747/> .
@prefix season2:       <http://www.imdb.com/title/tt0903747/episodes?season=6> .
@prefix series2:       <http://www.imdb.com/title/tt123456789/> .

series1:
    :changed    series:rating ;
    :removed    season2: .

section:
    :removed    series2: .
```

Listing D.11: Nicht-Ziel

```
@prefix :              <predicate://> .
@prefix series:        <property://series/> .
@prefix season:        <property://season/> .
@prefix actor:         <property://actor/> .
@prefix episode:       <property://episode/> .
@prefix season2:       <http://www.imdb.com/title/tt0903747/episodes?season=6> .
@prefix series2:       <http://www.imdb.com/title/tt123456789/> .
@prefix actor4:        <http://www.imdb.com/name/nm123456789/> .
@prefix season3:       <http://www.imdb.com/title/tt123456789/episodes?season=1> .
@prefix episode2:      <http://www.imdb.com/title/tt987654321/> .

season2:
    :removed    season:number .

series2:
    :removed    series:label ;
    :removed    actor4: ;
    :removed    season3: .

actor4:
    :removed    actor:label .

season3:
    :removed    season:number ;
    :removed    episode2: .

episode2:
    :removed    episode:label .
```

Listing D.12: Ergebnis

```
@prefix :              <predicate://> .
@prefix series:        <property://series/> .
@prefix section:       <http://www.imdb.com/chart/toptv/> .
@prefix series1:       <http://www.imdb.com/title/tt0903747/> .
@prefix season2:       <http://www.imdb.com/title/tt0903747/episodes?season=6> .
@prefix series2:       <http://www.imdb.com/title/tt123456789/> .

series1:
    :changed     series:rating  ;
    :removed     season2:  .

section:
    :removed     series2:  .
```

Anhang E

Evaluation von muenchen.de

Listing E.1: Modell

```
@prefix :                <predicate://> .
@prefix type:            <type://> .
@prefix page:            <page://> .
@prefix item:            <item://> .
@prefix event:           <property://event/> .
@prefix appointment:     <property://appointment/> .
@prefix location:        <property://location/> .
@prefix comedy:          <http://www.muenchen.de/veranstaltungen/events/comedy.html> .
@prefix musical:         <http://www.muenchen.de/veranstaltungen/events/musical.html> .

<model:>
    :section     comedy: ;
    :section     musical: .

comedy:
    :page        page:events .

musical:
    :page        page:events .

page:events
    :type        type:links ;
    :path        "//div[@id='oEventList']//a[@itemprop='url']" ;
    :scroll      1000 ;
    :target      page:event .

page:event
    :type        type:object ;
    :item        item:event ;
    :sub         page:appointments .

page:appointments
    :type        type:objects ;
    :item        item:appointment ;
    :sub         page:locations .

page:locations
    :type        type:links ;
    :path        ".//a[@itemprop='location']" ;
    :target      page:location .

page:location
    :type        type:object ;
    :item        item:location .

item:event
    :path        "//div[@id='main']" ;
    :property    event:label ;
    :property    event:description .
```

```
item:appointment
    :path       ".//*[@itemtype='http://schema.org/Event'][.//span[@itemprop='startDate']]" ;
    :property   appointment:start .

item:location
    :path       "//div[@id='pagecontent']" ;
    :property   location:label ;
    :property   location:address .

event:label
    :path       ".//h1" .

event:description
    :path       ".//div[@itemprop='description']" ;
    :optional   true .

appointment:start
    :path       ".//span[@itemprop='startDate']" ;
    :attribute  "content" ;
    :type       xsd:dateTime ;
    :format     "yyyy-MM-dd'T'HH:mm:ssZ" .

location:label
    :path       ".//h1" .

location:address
    :path       ".//div[@id='sidebar']//span[@itemprop='address']" .
```

Frage 1: Extraktion

Listing E.2: Ziel


```
{
  "http://www.muenchen.de/veranstaltungen/event/22060.html": {
    "property://event/label": "Kurt Krömer: Heute stimmt alles",
    "property://event/description": "2014 verabschiedete sich der Comedian aus dem deutschen
      Fernsehen, um sich verstärkt um seine Bühnenkarriere zu kümmern. Das hat offenbar geklappt:
      Krömer tourt mit einem nagelneuen Programm, aber in gewohnt \"altem\" Retro-Look."
  },
  "http://www.muenchen.de/veranstaltungen/event/22060.html
      #7ed598b68430196912e8d6ade0fcbc3930a5968c": {
    "property://appointment/start": "2016-01-22T20:00:00+01"
  },
  "http://www.muenchen.de/veranstaltungen/orte/131027.html": {
    "property://location/label": "Kongresshalle Augsburg",
    "property://location/address": "Gögginger Str. 10\n86159 Augsburg"
  },
  "http://www.muenchen.de/veranstaltungen/event/18009.html": {
    "property://event/label": "Ich war noch niemals in New York"
  },
  "http://www.muenchen.de/veranstaltungen/event/18009.html
      #534bb1b0b45f9cd828b3b75b5edf930c75aa4f29": {
    "property://appointment/start": "2016-01-03T14:30:00+01"
  },
  "http://www.muenchen.de/veranstaltungen/orte/140242.html": {
    "property://location/label": "Deutsches Theater",
    "property://location/address": "Schwanthalerstr. 13\n80336 München"
  }
}
```

Listing E.3: Ergebnis


```
{
  "http://www.muenchen.de/veranstaltungen/event/22060.html": {
    "property://event/label": "Kurt Krömer: Heute stimmt alles",
    "property://event/description": "2014 verabschiedete sich der Comedian aus dem deutschen
      Fernsehen, um sich verstärkt um seine Bühnenkarriere zu kümmern. Das hat offenbar geklappt:
      Krömer tourt mit einem nagelneuen Programm, aber in gewohnt \"altem\" Retro-Look."
  },
  "http://www.muenchen.de/veranstaltungen/event/22060.html
      #7ed598b68430196912e8d6ade0fcbc3930a5968c": {
    "property://appointment/start": "2016-01-22T20:00:00+01"
  },
  "http://www.muenchen.de/veranstaltungen/orte/131027.html": {
    "property://location/label": "Kongresshalle Augsburg",
    "property://location/address": "Gögginger Str. 10\n86159 Augsburg"
  },
  "http://www.muenchen.de/veranstaltungen/event/18009.html": {
    "property://event/label": "Ich war noch niemals in New York"
  },
  "http://www.muenchen.de/veranstaltungen/event/18009.html
      #534bb1b0b45f9cd828b3b75b5edf930c75aa4f29": {
    "property://appointment/start": "2016-01-03T14:30:00+01"
  },
  "http://www.muenchen.de/veranstaltungen/orte/140242.html": {
    "property://location/label": "Deutsches Theater",
    "property://location/address": "Schwanthalerstr. 13\n80336 München"
  }
}
```

Frage 2: Aufbereitung und Speicherung

Listing E.4: Ziel

```
@prefix item:           <item://> .
@prefix event:          <property://event/> .
@prefix appointment:    <property://appointment/> .
@prefix location:       <property://location/> .
@prefix comedy:         <http://www.muenchen.de/veranstaltungen/events/comedy.html> .
@prefix musical:        <http://www.muenchen.de/veranstaltungen/events/musical.html> .
@prefix event1:         <http://www.muenchen.de/veranstaltungen/event/22060.html> .
@prefix appointment1:   <http://www.muenchen.de/veranstaltungen/event/22060.html
                            #7ed598b68430196912e8d6ade0fcbc3930a5968c> .
@prefix location1:      <http://www.muenchen.de/veranstaltungen/orte/131027.html> .
@prefix event2:         <http://www.muenchen.de/veranstaltungen/event/18009.html> .
@prefix appointment2:   <http://www.muenchen.de/veranstaltungen/event/18009.html
                            #534bb1b0b45f9cd828b3b75b5edf930c75aa4f29> .
@prefix location2:      <http://www.muenchen.de/veranstaltungen/orte/140242.html> .

comedy:
    item:event          event1: .

musical:
    item:event          event2: .

event1:
    event:label         "Kurt Krömer: Heute stimmt alles"^^xsd:string ;
    event:description   """2014 verabschiedete sich der Comedian aus dem deutschen Fernsehen,
        um sich verstärkt um seine Bühnenkarriere zu kümmern. Das hat offenbar geklappt: Krömer
        tourt mit einem nagelneuen Programm, aber in gewohnt "altem" Retro-Look."""^^xsd:string ;
    item:appointment    appointment1: .

appointment1:
    appointment:start   "2016-01-22T20:00:00+01"^^xsd:dateTime ;
    item:location       location1: .

location1:
    location:label      "Kongresshalle Augsburg"^^xsd:string ;
    location:address    "Gögginger Str. 10\n86159 Augsburg"^^xsd:string .

event2:
    event:label         "Ich war noch niemals in New York"^^xsd:string ;
    item:appointment    appointment2: .

appointment2:
    appointment:start   "2016-01-03T14:30:00+01"^^xsd:dateTime ;
    item:location       location2: .

location2:
    location:label      "Deutsches Theater"^^xsd:string ;
    location:address    "Schwanthalerstr. 13\n80336 München"^^xsd:string .
```

Listing E.5: Ergebnis

```
@prefix item:           <item://> .
@prefix event:          <property://event/> .
@prefix appointment:    <property://appointment/> .
@prefix location:       <property://location/> .
@prefix comedy:         <http://www.muenchen.de/veranstaltungen/events/comedy.html> .
@prefix musical:        <http://www.muenchen.de/veranstaltungen/events/musical.html> .
@prefix event1:         <http://www.muenchen.de/veranstaltungen/event/22060.html> .
@prefix appointment1:   <http://www.muenchen.de/veranstaltungen/event/22060.html
                            #7ed598b68430196912e8d6ade0fcbc3930a5968c> .
@prefix location1:      <http://www.muenchen.de/veranstaltungen/orte/131027.html> .
@prefix event2:         <http://www.muenchen.de/veranstaltungen/event/18009.html> .
@prefix appointment2:   <http://www.muenchen.de/veranstaltungen/event/18009.html
                            #534bb1b0b45f9cd828b3b75b5edf930c75aa4f29> .
@prefix location2:      <http://www.muenchen.de/veranstaltungen/orte/140242.html> .

comedy:
    item:event          event1: .

musical:
    item:event          event2: .

event1:
    event:label         "Kurt Krömer: Heute stimmt alles"^^xsd:string ;
    event:description   """2014 verabschiedete sich der Comedian aus dem deutschen Fernsehen,
        um sich verstärkt um seine Bühnenkarriere zu kümmern. Das hat offenbar geklappt: Krömer
        tourt mit einem nagelneuen Programm, aber in gewohnt "altem" Retro-Look."""^^xsd:string ;
    item:appointment    appointment1: .

appointment1:
    appointment:start   "2016-01-22T20:00:00.000+01:00"^^xsd:dateTime ;
    item:location       location1: .

location1:
    location:label      "Kongresshalle Augsburg"^^xsd:string ;
    location:address    "Gögginger Str. 10\n86159 Augsburg"^^xsd:string .

event2:
    event:label         "Ich war noch niemals in New York"^^xsd:string ;
    item:appointment    appointment2: .

appointment2:
    appointment:start   "2016-01-03T14:30:00.000+01:00"^^xsd:dateTime ;
    item:location       location2: .

location2:
    location:label      "Deutsches Theater"^^xsd:string ;
    location:address    "Schwanthalerstr. 13\n80336 München"^^xsd:string .
```

Frage 3: Versionierung

Durchgang 1

Listing E.6: Ziel

```
@prefix :              <predicate://> .
@prefix comedy:        <http://www.muenchen.de/veranstaltungen/events/comedy.html> .
@prefix musical:       <http://www.muenchen.de/veranstaltungen/events/musical.html> .
@prefix event1:        <http://www.muenchen.de/veranstaltungen/event/22060.html> .
@prefix event2:        <http://www.muenchen.de/veranstaltungen/event/18009.html> .

comedy:
    :added event1: .

musical:
    :added event2: .
```

Listing E.7: Nicht-Ziel

```
@prefix :              <predicate://> .
@prefix event:         <property://event/> .
@prefix appointment:   <property://appointment/> .
@prefix location:      <property://location/> .
@prefix event1:        <http://www.muenchen.de/veranstaltungen/event/22060.html> .
@prefix appointment1:  <http://www.muenchen.de/veranstaltungen/event/22060.html
                           #7ed598b68430196912e8d6ade0fcbc3930a5968c> .
@prefix location1:     <http://www.muenchen.de/veranstaltungen/orte/131027.html> .

event1:
    :added event:label ;
    :added appointment1: .

appointment1:
    :added appointment:start ;
    :added location1: .

location1:
    :added location:label .
```

Listing E.8: Ergebnis

```
@prefix :             <predicate://> .
@prefix comedy:       <http://www.muenchen.de/veranstaltungen/events/comedy.html> .
@prefix musical:      <http://www.muenchen.de/veranstaltungen/events/musical.html> .
@prefix event1:       <http://www.muenchen.de/veranstaltungen/event/22060.html> .
@prefix event2:       <http://www.muenchen.de/veranstaltungen/event/18009.html> .

comedy:
    :added event1: .

musical:
    :added event2: .
```

Durchgang 2

Listing E.9: Daten

```
@prefix item:           <item://> .
@prefix event:          <property://event/> .
@prefix appointment:    <property://appointment/> .
@prefix location:       <property://location/> .
@prefix comedy:         <http://www.muenchen.de/veranstaltungen/events/comedy.html> .
@prefix event1:         <http://www.muenchen.de/veranstaltungen/event/22060.html> .
@prefix location1:      <http://www.muenchen.de/veranstaltungen/orte/131027.html> .
@prefix appointment3:   <http://www.muenchen.de/veranstaltungen/event/22060.html
                            #24e6404cfe8285be3066844daff4ddaca753fb23> .
@prefix location2:      <http://www.muenchen.de/veranstaltungen/orte/140242.html> .
@prefix event3:         <http://www.muenchen.de/veranstaltungen/event/12345.html> .
@prefix appointment4:   <http://www.muenchen.de/veranstaltungen/event/12345.html
                            #a42cc701ad4ef94e953ef1d7288d0b193935fa72> .
@prefix location3:      <http://www.muenchen.de/veranstaltungen/orte/54321.html> .

event1:
    item:appointment    appointment3: .

appointment3:
    appointment:start   "2016-04-03T20:00:00.000+01:00"^^xsd:dateTime ;
    item:location       location1: .

location2:
    location:address    "Schwanthalerstr. 12\n80336 München"^^xsd:string .

comedy:
    item:event          event3: .

event3:
    event:label         "Veranstaltung"^^xsd:string ;
    item:appointment    appointment4: .

appointment4:
    appointment:start   "2016-03-02T20:00:00.000+01:00"^^xsd:dateTime ;
    item:location       location3: .

location3:
    location:label      "Ort"^^xsd:string ;
    location:address    "Adresse"^^xsd:string .
```

Listing E.10: Ziel

```
@prefix :                <predicate://> .
@prefix location:        <property://location/> .
@prefix comedy:          <http://www.muenchen.de/veranstaltungen/events/comedy.html> .
@prefix event1:          <http://www.muenchen.de/veranstaltungen/event/22060.html> .
@prefix appointment3:    <http://www.muenchen.de/veranstaltungen/event/22060.html
                             #24e6404cfe8285be3066844daff4ddaca753fb23> .
@prefix location2:       <http://www.muenchen.de/veranstaltungen/orte/140242.html> .
@prefix event3:          <http://www.muenchen.de/veranstaltungen/event/12345.html> .

event1:
    :removed    appointment3: .

location2:
    :changed    location:address .

comedy:
    :removed    event3: .
```

Listing E.11: Nicht-Ziel

```
@prefix :                <predicate://> .
@prefix event:           <property://event/> .
@prefix appointment:     <property://appointment/> .
@prefix location:        <property://location/> .
@prefix location1:       <http://www.muenchen.de/veranstaltungen/orte/131027.html> .
@prefix appointment3:    <http://www.muenchen.de/veranstaltungen/event/22060.html
                             #24e6404cfe8285be3066844daff4ddaca753fb23> .
@prefix event3:          <http://www.muenchen.de/veranstaltungen/event/12345.html> .
@prefix appointment4:    <http://www.muenchen.de/veranstaltungen/event/12345.html
                             #a42cc701ad4ef94e953ef1d7288d0b193935fa72> .
@prefix location3:       <http://www.muenchen.de/veranstaltungen/orte/54321.html> .

appointment3:
    :removed    appointment:start ;
    :removed    location1: .

event3:
    :removed    event:label ;
    :removed    appointment4: .

appointment4:
    :removed    appointment:start ;
    :removed    location3: .

location3:
    :removed    location:label .
```

Listing E.12: Ergebnis

```
@prefix :                 <predicate://> .
@prefix location:         <property://location/> .
@prefix comedy:           <http://www.muenchen.de/veranstaltungen/events/comedy.html> .
@prefix event1:           <http://www.muenchen.de/veranstaltungen/event/22060.html> .
@prefix appointment3:     <http://www.muenchen.de/veranstaltungen/event/22060.html
                              #24e6404cfe8285be3066844daff4ddaca753fb23> .
@prefix location2:        <http://www.muenchen.de/veranstaltungen/orte/140242.html> .
@prefix event3:           <http://www.muenchen.de/veranstaltungen/event/12345.html> .

event1:
    :removed    appointment3: .

location2:
    :changed    location:address .

comedy:
    :removed    event3: .
```

Abbildungsverzeichnis

Literaturverzeichnis

[Aras 03] A. Arasu and H. Garcia-Molina. “Extracting structured data from Web pages”. 2003.

[BSI] BSI. “Cloud Computing Grundlagen”. https://www.bsi.bund.de/DE/Themen/CloudComputing/Grundlagen/Grundlagen_node.html. abgerufen am 05.11.2015.

[Buhm 14] L. Bühmann, R. Usbeck, A.-C. Ngonga Ngomo, M. Saleem, A. Both, V. Crescenzi, P. Merialdo, and D. Qiu. “Web-Scale Extension of RDF Knowledge Bases from Templated Websites”. In: P. Mika, T. Tudorache, A. Bernstein, C. Welty, C. Knoblock, D. Vrandečić, P. Groth, N. Noy, K. Janowicz, and C. Goble, Eds., *The Semantic Web – ISWC 2014*, Chap. 5, pp. 66–81, Springer International Publishing, 2014.

[Goog] Google. “README.chromium - Code Search”. https://code.google.com/p/chromium/codesearch#chromium/src/third_party/libxml/README.chromium. abgerufen am 30.09.2015.

[Hogu 05] A. Hogue and D. Karger. “Thresher: automating the unwrapping of semantic content from the World Wide Web”. 2005.

[IETFa] IETF. “RFC 3174 - US Secure Hash Algorithm 1 (SHA1)”. https://tools.ietf.org/html/rfc3174. abgerufen am 13.10.2015.

[IETFb] IETF. “RFC 7159 - The JavaScript Object Notation (JSON) Data Interchange Format”. https://tools.ietf.org/html/rfc7159. abgerufen am 13.10.2015.

[IETFc] IETF. “RFC 7230 - Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing”. https://tools.ietf.org/html/rfc7230. abgerufen am 13.10.2015.

[IETFd] IETF. “Uniform Resource Identifier (URI)”. https://tools.ietf.org/html/rfc3986. abgerufen am 13.10.2015.

[ISO] ISO. “ISO 8601:2004”. http://www.iso.org/iso/catalogue_detail?csnumber=40874. abgerufen am 30.09.2015.

[LRZ] LRZ. “RDF 1.1 Concepts and Abstract Syntax”. http://www.lrz.de/services/schulung/unterlagen/regul/. abgerufen am 13.10.2015.

[Mozi] Mozilla. “Bug 396966 - Xpath 2.0”. https://bugzilla.mozilla.org/show_bug.cgi?id=396966. abgerufen am 30.09.2015.

[Resi] J. Resig. “XPath and CSS Selectors”. http://ejohn.org/blog/xpath-css-selectors/. abgerufen am 30.09.2015.

[W3Ca] W3C. “CSS Selectors”. http://www.w3.org/TR/CSS21/selector.html. abgerufen am 30.09.2015.

[W3Cb] W3C. “HTML5”. http://www.w3.org/TR/html5/. abgerufen am 13.10.2015.

[W3Cc] W3C. “Linked Data”. http://www.w3.org/standards/semanticweb/data. abgerufen am 13.10.2015.

[W3Cd] W3C. “Notation3 (N3): A readable RDF syntax”. http://www.w3.org/TeamSubmission/n3/. abgerufen am 13.10.2015.

[W3Ce] W3C. “SPARQL 1.1”. http://www.w3.org/TR/sparql11-overview/. abgerufen am 30.09.2015.

[W3Cf] W3C. “Uniform Resource Identifier (URI)”. http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/. abgerufen am 13.10.2015.

[W3Cg] W3C. “XML Path Language (XPath) 1.0”. http://www.w3.org/TR/xpath/. abgerufen am 30.09.2015.

[W3Ch] W3C. “XML Schema Definition Language (XSD) 1.1: Datatypes”. http://www.w3.org/TR/xmlschema11-2/. abgerufen am 30.09.2015.

[Xmls] Xmlsoft. “The XML C parser and toolkit of Gnome”. http://www.xmlsoft.org. abgerufen am 30.09.2015.