

Martin Stabauer / Johann Höller

Verordnungen erstellen ohne Rechtskenntnisse

Demonstration eines Curriculum-Designprozesses

Dieser Beitrag zeigt die praxisnahe Umsetzung eines Frameworks zur Unterstützung von Rechtssetzungsprozessen. Es wird anhand universitärer Curricula präsentiert, wie semantische Technologien und klassische Softwareentwicklung zusammenwirken können, um juristischen Laien das Erstellen von Verordnungen zu erleichtern.

Collection: Conference Proceedings IRIS 2015

Category: Articles

Field of law: Advanced Legal Informatics Systems and Applications

Region: Austria

Citation: Martin Stabauer / Johann Höller, Verordnungen erstellen ohne Rechtskenntnisse, in: Jusletter IT 26. Februar 2015 – IRIS

Inhaltsübersicht

- 1 Einführung
 - 1.1 Curricula nach österreichischem Recht
 - 1.2 Normsetzungsprozess und Zusammenarbeit
- 2 Semantische Modellierung
 - 2.1 Grundmodell
 - 2.2 Curriculum
 - 2.3 Fächer, Module, Lehrveranstaltungen
 - 2.4 Textbausteine
 - 2.5 Beziehungen zu externen Ontologien
- 3 Benutzerunterstützung
 - 3.1 Weitergehende Automatisierung
 - 3.2 Regelwerk
 - 3.3 Benutzeroberflächen
- 4 Fazit und Ausblick

1 Einführung

[Rz 1] Dieses Paper ist als Fortsetzung bzw. Detaillierung des Beitrags «HÖLLER, JOHANN/STABAUER, MARTIN, IT in der Normsetzung — Semantische Technologien als Lösungsansatz» zu verstehen, dessen theoretische Konzepte an dieser Stelle anhand eines Fallbeispiels praxisnah demonstriert werden sollen.

[Rz 2] Der entwickelte integrative Ansatz¹ vereint klassische Methoden aus der Softwareentwicklung mit semantischen Modellierungswerkzeugen und ermöglicht so das praxisnahe Umsetzen von Rechtstexten in Ontologien. Wir stellen den angesprochenen Ansatz, bestehend aus einem semantischen Grundmodell und einem dazu passenden Vorgehensmodell, vor und demonstrieren seine Umsetzbarkeit an Hand eines Fallbeispiels von universitären Curricula nach österreichischem Recht.

[Rz 3] Es wurde im Vorfeld ermittelt, welche Stakeholder auf welche Weise partizipieren, wie die Bedürfnisse und Anforderungen der einzelnen Beteiligten aussehen und welche Anforderungen sich daraus für entsprechende IT-Systeme ergeben. Ein beispielhaftes webbasiertes System, das im Zuge der Umsetzung als Unterstützung für die Verfasser von Curricula entwickelt wurde, wird in einer Live-Demonstration vorgestellt und zeigt so die Implementierbarkeit unseres Ansatzes.

1.1 Curricula nach österreichischem Recht

[Rz 4] Das österreichische Hochschulwesen hat sich in den vergangenen Jahren massiv gewandelt. Im Zuge der 2002 mittels dem Universitätsgesetz (UnivG) eingeführten Vollrechtsfähigkeit der österreichischen Universitäten wurde die Erlassung von Satzungen für alle Universitäten Pflicht. Ein IT-System zur Modellierung von Curricula muss somit nicht nur Bundesgesetze, sondern auch die Satzungen der jeweiligen Universitäten berücksichtigen. Dies bringt die Anforderung zur Wandlungsfähigkeit mit sich — ein System, das über mehrere Universitäten hinweg betrieben werden soll, muss sich den jeweiligen Gegebenheiten anpassen können. Im Zuge der in diesem

¹ Vgl. STABAUER, MARTIN, Der Nutzen semantischer Technologien bei Erstellung und Pflege von Rechtstexten am Beispiel von Curricula, Linz (2014).

Beitrag beschriebenen Modellierung von Curricula wurde insbesondere auf die an der Johannes Kepler Universität Linz (JKU) gültigen Rechtsnormen eingegangen, es stellt jedoch keine große Schwierigkeit dar, das System an andere Universitäten und sonstige Bildungseinrichtungen anzupassen.

[Rz 5] Weitere tiefgreifende Änderungen der letzten Jahre waren die Weichenstellung in Richtung Bologna-Architektur, also die Dreiteilung des Ausbildungssystems in Bachelor-, Master- und Doktoratsstudien, sowie das einheitliche Leistungspunktesystem «European Credit Transfer System»² (ECTS).

1.2 Normsetzungsprozess und Zusammenarbeit

[Rz 6] Wie HÖLLER/STABAUER³ in diesem Band ausführen, gibt es zumindest theoretisch eine breite Palette von Einsatzmöglichkeiten für semantische Modelle im Recht. Nicht immer rechtfertigen die diskutierten Vorteile jedoch den stark erhöhten Aufwand gegenüber der herkömmlichen Formulierung von Rechtstext. Mögliche Gebiete mit hohem Nutzen sind eng abgegrenzte Rechtsmaterien mit hoher Wiederverwendbarkeit oder auch abrechnungsorientierte Materien.

[Rz 7] Die Erstellung und Pflege von Curricula hat gegenüber anderen Normsetzungsprozessen einige Besonderheiten. Im § 22 Abs. 1 Z. 12 UnivG ist geregelt, dass nur das Rektorat ein neues Studium einrichten kann, dieses wird dann einem gem. § 25 Abs. 8 Z. 3 UnivG vom Senat eingesetzten Kollegialorgan zur Ausarbeitung zugewiesen. Dieses Kollegialorgan wird an der JKU als «Studienkommission» bezeichnet und besteht im Wesentlichen aus Experten der jeweiligen Fachrichtung, die jedoch nicht zwangsläufig auch hinreichende juristische Erfahrung aufweisen. Die Problematik bei der Normsetzung liegt in diesem Fall daher oftmals darin, dass juristische Laien ohne angemessene Hilfestellung zur Formulierung juristischer Texte angehalten werden. Hier können semantische Technologien mit ihrer eindeutigen Begriffswelt und klaren Zusammenhängen einen Lösungsansatz bieten. Das in diesem Paper vorgestellte Framework wirkt unterstützend und automatisierend.

2 Semantische Modellierung

[Rz 8] In diesem Abschnitt soll nun die eigentliche Modellierung von Curricula in der eben diskutierten Form beschrieben werden. Die in OWL⁴ modellierten Beispiele werden der besseren Lesbarkeit wegen in leicht modifiziertem Turtle angeführt. Turtle steht für Terse RDF Triple Language und stellt RDF-Triples in für Menschen leicht lesbarer Spaltenform dar⁵.

² Vgl. ECTS-Leitfaden, http://ec.europa.eu/education/tools/docs/ects-guide_de.pdf, Abruf am 15. Dezember 2014 (2009).

³ Vgl. HÖLLER, JOHANN/STABAUER, MARTIN, IT in der Normsetzung — Semantische Technologien als Lösungsansatz, IRIS (2015).

⁴ Vgl. <http://www.w3.org/TR/owl2-overview>, Abruf am 15. Dezember 2014 (2012).

⁵ Vgl. BECKET, D. ET AL., <http://www.w3.org/TR/turtle>, Abruf am 15. Dezember 2014 (2004).

2.1 Grundmodell

[Rz 9] Als Ausgangsbasis wird das von STABAUER/HÖLLER⁶ beschriebene semantische Grundmodell in leicht modifizierter Form herangezogen und um notwendige Ergänzungen für den Einsatz bei Curricula erweitert. Das so entstandene Framework trägt den Namen Curriculum Modeling Language, kurz CML. Es beinhaltet neben Klassen für die Struktur des Rechtstextes und aller Bestandteile von Curricula auch weitere Ontologien zur Abbildung von Universitäten und anderer Bildungseinrichtungen, von Personen, Regeln, Mustertexte und Übersetzungen und Mappings zu externen Ontologien.

2.2 Curriculum

[Rz 10] Das Curriculum nach österreichischem Recht und gemäß den Bestimmungen der Satzung der JKU wird in CML zusammen mit Beschreibungen und einigen Metadaten modelliert. Diese Metadaten beinhalten z.B. das In- und Außerkrafttreten, mit Hilfe derer eine einfache Versionierung der Curricula erreicht wird. Zudem werden die Beziehungen zur Fächer- und Modulstruktur (siehe nächster Abschnitt) und zu den Textbausteinen des geschriebenen Curriculumtexts (siehe übernächster Abschnitt) modelliert. Hier ein kurzer Ausschnitt mit einigen wesentlichen Bestandteilen des modellierten Individuums der Klasse *cml:Curriculum* für das Masterstudium Webwissenschaften an der JKU:

```
webwi:WebWiM      rdf:type          cml:Curriculum
                  rdfs:label        «Webwissenschaften»@de
                  cml:inkrafttreten «2013-10-01»
                  cml:hatCurriculumTyp  cml:CurrM
                  cml:hatStudiengruppe  cml:SoWiStudium
                  cml:uebergeordnetPflicht webwi:Fach575WEBW10
                  cml:istQuellcurriculum webwi:Fach575WEBW10
                  cml:uebergeordnetText webwi:AkadGrad
```

2.3 Fächer, Module, Lehrveranstaltungen

[Rz 11] Einer der beiden großen Blöcke eines Curriculums ist die Hierarchie von Fächern, Modulen und Lehrveranstaltungen, die den Studierenden angeboten und nach bestimmten Regeln absolviert werden müssen. Da einige Teile wie Name, ECTS, Code, Reihenfolge, Quellcurriculum oder Hierarchisierung auf alle drei Varianten zutreffen, wurde in CML eine gemeinsame Superklasse *cml:Baustein* eingeführt, die diese übergreifenden Eigenschaften beinhaltet. Bis auf wenige Ausnahmefälle sind aber alle Bausteine einer der drei Subklassen zugeordnet, *cml:Baustein* ist daher als abstrakte Klasse zu sehen. Jede der drei Varianten hat ihre Eigenheiten, die in den entsprechenden Modellen abgebildet sind, deren genauere Beschreibung an dieser Stelle jedoch zu weit führen würde. Hierunter fallen beispielsweise alle Informationen zu Lehrveranstaltungen,

⁶ Vgl. STABAUER, MARTIN/HÖLLER, JOHANN, Ein integrativer Ansatz zur semantischen Modellierung von Rechtstexten, in: Jusletter IT 11. Dezember 2014.

die gemäß der Satzung veröffentlicht werden müssen, wie etwa Prüfungs- oder Anmelde-mo-dalitäten.

```
webwi:LV575WQUEWSCK10  rdf:type          cml:LV
                          rdfs:label       «Web Security»@de
                          cml:hatTyp       cml:KV
                          cml:hatReihenfolge 2
                          cml:wst         2.0
                          cml:ects        3.0
                          cml:code        «575WQUEWSCK10»
                          cml:untergeordnet webwi:Fach575WQUE10
                          cml:ziele       «Die Studierenden besitzen ...»@de
                          cml:hatVerantwortl jku:MaxMustermann
```

[Rz 12] Die vorangehende Auflistung zeigt unter anderem, dass mit CML auch natürliche Per-sonen modelliert werden. Hierbei handelt es sich vornehmlich um Personen mit besonderen Be-rechtigungen, wie in diesem Fall den Verantwortlichen für eine Lehrveranstaltung. Die so mo-dellierten Informationen können beispielsweise für eine Authentifizierung beim Bearbeiten von Daten in einer speziellen Benutzeroberfläche genutzt werden.

2.4 Textbausteine

[Rz 13] Um sicherstellen zu können, dass der ursprüngliche Normtext ohne Verlust von Inhalt oder Form wiederhergestellt werden kann, müssen die einzelnen Textbausteine des Curricu-lums jeweils für sich modelliert werden. Zu diesem Zweck sind im Grundmodell OWL-Klassen für viele denkbare Hierarchiestufen eines Curriculumtextes mit einer gemeinsamen Superklasse *cml:Textbaustein* vordefiniert: Paragraph, Absatz, Ziffer, Litera, Sublitera, Inhalt. Als Beispiel soll gezeigt werden, wie folgender kurzer Auszug aus Webwissenschaften grundsätzlich modellierbar ist. Das Beispiel wird in weiterer Folge noch stärker annotiert und weiter automatisiert werden.

§ 10 Akademischer Grad (1) An die AbsolventInnen des Masterstudiums Webwissenschaften ist der akademische Grad «Master of Science», abgekürzt «MSc» oder «MSc (JKU)», zu verleihen. (2) Der Bescheid über den akademischen Grad wird in deutscher Sprache und englischer Übersetzung ausgefertigt.

<code>webwi:AkadGrad</code>	<code>rdf:type</code>	<code>cml:Paragraph</code>
	<code>rdfs:label</code>	<code>«Akademischer Grad»@de</code>
	<code>cml:hatReihenfolge</code>	<code>10</code>
	<code>cml:untergeordnetText</code>	<code>webwi:WebWiM</code>
<code>webwi:AkadGrad1</code>	<code>rdf:type</code>	<code>cml:Absatz</code>
	<code>cml:hatReihenfolge</code>	<code>1</code>
	<code>cml:untergeordnetText</code>	<code>webwi:AkadGrad</code>
<code>webwi:InhaltAkadGrad1</code>	<code>rdf:type</code>	<code>cml:Inhalt</code>
	<code>cml:untergeordnetText</code>	<code>webwi:AkadGrad1</code>
	<code>rdfs:label</code>	<code>«An die AbsolventInnen ...»@de</code>
<code>webwi:AkadGrad2</code>	<code>rdf:type</code>	<code>cml:Absatz</code>
	<code>cml:hatReihenfolge</code>	<code>2</code>
	<code>cml:untergeordnetText</code>	<code>webwi:AkadGrad</code>
<code>webwi:InhaltAkadGrad2</code>	<code>rdf:type</code>	<code>cml:Inhalt</code>
	<code>cml:untergeordnetText</code>	<code>webwi:AkadGrad2</code>
	<code>rdfs:label</code>	<code>«Der Bescheid über den ...»@de</code>

[Rz 14] Mit `webwi:AkadGrad` wird zunächst der Paragraph 10 modelliert, welcher dem entsprechenden Curriculum direkt untergeordnet wird. Das Objekt wird als Paragraph mit der Bezeichnung «Akademischer Grad» und der Reihenfolge 10 definiert.

[Rz 15] Zu beachten ist, dass eine eventuell notwendige Internationalisierung sehr einfach vollzogen werden kann, indem weitere Datenbeziehungen `rdfs:label` hinzugefügt werden:

<code>webwi:AkadGrad</code>	<code>rdfs:label</code>	<code>«Academic Degree»@en</code>
-----------------------------	-------------------------	-----------------------------------

[Rz 16] Auch die Inhaltselemente mit dem eigentlichen Text des Bausteins können auf diese Weise übersetzt und so mehrere Sprachen in einer einzigen Ontologie mit einer gemeinsamen Struktur abgespeichert werden. Dies hat unter anderem den großen Vorteil, dass spätere Änderungen an nur einer Stelle durchgeführt werden müssen, um dennoch alle modellierten Sprachversionen einzubeziehen.

[Rz 17] Auf diese Weise sind auch weitere Ergänzungen der einzelnen Textbausteine möglich, beispielsweise kann eine inhaltliche Kategorisierung vorgenommen werden, wie es im folgenden Abschnitt beschrieben wird.

2.5 Beziehungen zu externen Ontologien

[Rz 18] Bislang wurde der eigentliche Curriculumstext zusammen mit seiner Hierarchie von Fächern und Modulen abgespeichert. Es wurde gezeigt, wie mit Hilfe von Objektbeziehungen Verbindungen zwischen den Elementen hergestellt und so die wichtigsten Bedeutungen zu den Textbausteinen hinzugefügt werden können. Ist in einem Absatz oder sonstigem Inhaltselement von einem Fach oder Modul die Rede, werden die Inhalte direkt von diesem entnommen. Dies erleichtert die Wartbarkeit und vermeidet Missverständnisse.

[Rz 19] Häufig sollen aber noch weitere Metainformationen zu Textbausteinen modelliert, bzw. die Bedeutung von einzelnen Elementen verdeutlicht werden. Dies kann in vielen Fällen mit Verbindungen zu externen Ontologien passieren. Anhand zweier beispielhafter externer Quellen

sollen diese Beziehungen näher erläutert werden: DBpedia⁷ und LKIF⁸.

[Rz 20] DBpedia ist eine Community, die es sich zum Ziel gemacht hat, strukturierte semantische Informationen aus Wikipedia zu extrahieren und so umfangreiche Modelle unterschiedlichster Wissensgebiete anzubieten. Viele der Konzepte von DBpedia haben sich als Standard für andere Ontologien durchgesetzt. Hier das Beispiel der Bildungseinrichtungen, auf diese Weise können mit einem Schlag alle in einem CML-Curriculum definierten Bildungseinrichtungen der Entsprechung von DBpedia verbunden werden:

```
webwi:JKU                rdf:type                cml_uni:Bildungseinrichtung
                        rdfs:label                «Johannes Kepler Universität»@de
cml_uni:Bildungseinrichtung owl:equivalentClass «http://dbpedia.org/ontology/
                                                EducationalInstitution»
```

[Rz 21] Die zweite an dieser Stelle erwähnte externe Ontologie ist das Legal Knowledge Interchange Format, kurz LKIF, das vom EU-weit aktiven Estrella Konsortium («European Project for Standardized Transparent Representations in order to Extend Legal Accessibility») implementiert wurde und sich inzwischen zu einer Standardsprache zur Modellierung von juristischem Wissen entwickelt hat. LKIF berücksichtigt dabei spezifische Terminologie, Regeln und auch normative Aussagen. Die Regeln erweitern hierbei den «Semantic Web Rule Language» (SWRL) Standard⁹, der sich wiederum selbst aus OWL und RuleML («Rule Markup Language»)¹⁰ zusammensetzt.

[Rz 22] Die Mappings von CML zu LKIF betreffen insbesondere die Klassen von *lkif:Legal_Expression*, in denen Definitionen, Existenzdeklarationen und ähnliche Ausdrücke abgebildet sind, und *lkif:Norm*, welches qualifizierte Aussagen wie Verpflichtungen, Verbote und Rechte beinhaltet.

3 Benutzerunterstützung

[Rz 23] Die Modellierung von Sachverhalten per se nutzt noch nicht das volle Potenzial der Semantik. Erst durch die im Folgenden beschriebenen Techniken können wesentliche Vorteile ausgeschöpft werden.

3.1 Weitergehende Automatisierung

[Rz 24] Eine weitergehende Automatisierung kann beispielsweise auf zwei Arten passieren. Zum einen können häufig benötigte Formulierungen und standardisierte Mustertexte in einer eigenen Ontologie ausgelagert und an den Stellen importiert werden, in denen sie schlussendlich innerhalb eines konkreten Curriculums benötigt werden. Ein Beispiel hierfür ist der im vorigen Abschnitt erwähnte erste Absatz des Paragraphs betreffend den akademischen Grad. Dieser tritt

⁷ Vgl. <http://dbpedia.org>, Abruf am 15. Dezember 2014.

⁸ Vgl. http://www.estrellaproject.org/?page_id=5, Abruf am 15. Dezember 2014.

⁹ Vgl. HORROCKS, IAN ET AL., SWRL: A Semantic Web Rule Language — Combining OWL and RuleML, <http://www.w3.org/Submission/SWRL>, Abruf am 15. Dezember 2014 (2004).

¹⁰ Vgl. BOLEY, HAROLD ET AL., Design Rationale of RuleML — A Markup Language for Semantic Web rules. In: Proceedings of the Semantic Web Working Symposium, S. 381—401 (2001).

in einer sehr ähnlichen Form in nahezu allen denkbaren Curricula auf, lediglich die konkrete Platzierung des Absatzes und die inhaltlichen Komponenten (Name und Typ des Curriculums, akademischer Grad in Lang- und Kurzform) müssen angepasst werden. Zu diesem Zweck bietet sich die Verwendung von Variablen an, die dann von den algorithmischen Komponenten des Frameworks durch die entsprechenden Werte ersetzt werden.

[Rz 25] Das erwähnte Beispiel könnte also an einer zentralen Stelle unter Verwendung von Variablen, die durch eckige Klammern hervorgehoben werden, abgespeichert werden. Die konkrete Implementierung innerhalb eines Curriculums kann dieses dann problemlos importieren:

```
mustertext:AkadGrad1    rdf:type          cml:Inhalt
                        rdfs:label        «An die AbsolventInnen des
[CURRTYP]s [CURRNAME] ist der akademische Grad «[AKADGRADLANG]», abgekürzt
                        «[AKADGRADKURZ]» oder «[AKADGRADKURZ] (JKU)», zu verleihen.»@de
webwi:AkadGrad1        rdf:type          cml:Absatz
                        cml:hatReihenfolge 1
                        cml:uebergeordnetText mustertext:AkadGrad1
```

[Rz 26] Somit bleibt für den texterzeugenden Algorithmus die Aufgabe, die Variablen wie [CURRNAME] oder [AKADGRADLANG] durch die entsprechenden Werte, die beim modellierten Objekt des Curriculums abgespeichert sind, zu ersetzen. Dies kann auch mehrsprachig erfolgen, indem im Falle des nicht-deutschsprachigen Texts für jede Variable der Inhalt des jeweiligen *rdfs:label* eingesetzt wird.

[Rz 27] Zum anderen betreffen große Teile des Curriculumtexts konkrete Fächer und Module, die ja ebenfalls innerhalb der Ontologie modelliert werden. Hier bieten sich direkte Verweise auf die entsprechenden Bausteine an, sowohl was Inhalte hinsichtlich einzelner Fächer und Module betrifft, als auch in jenen Fällen, in denen ganze Listen ausgegeben werden sollen. Diese Verweise werden mittels Objektbeziehungen modelliert, hierfür sind in CML bereits einige häufig benötigte Varianten vordefiniert. So gibt es etwa die Objektbeziehung *cml:listeUntergeordnete*, die die untergeordneten Bausteine des angegebenen Bausteins oder Curriculums ausgibt. Das folgende Beispiel etwa definiert ein Inhaltselement, welches die oberste Ebene der Fächer und Module des Curriculums ausgibt, wie dies im Paragraphen «Aufbau und Gliederung» benötigt wird:

```
webwi:Aufbaul_1        rdf:type          cml:Inhalt
                        cml:hatReihenfolge 2
                        cml:listeUntergeordnete webwi:WebWiM
```

[Rz 28] Die texterzeugenden Algorithmen geben dann in einer formatierten Liste die entsprechenden Daten (Name und ECTS) dieser Bausteine aus:

Propädeutikum Webwissenschaften	2.0
Grundlagen der Webwissenschaften	24.0
Studienzweige	30.0
Webwissenschaften	24.0
Masterarbeit (inkl. Masterarbeitsseminar)	27.0
Masterprüfung	1.0
Freie Studienleistungen	12.0

Abbildung 1: `cml:listeUntergeordnete`

3.2 Regelwerk

[Rz 29] Das zu CML entwickelte Regelwerk stellt eine weitere Unterstützung der Benutzer dar. Zum einen erlaubt es den Erstellern des Curriculums, insbesondere den Studienkommissionen, auf einfachere Art und Weise standardkonforme Curricula zu erzeugen, die allen anzuwendenden Rechtsvorschriften entsprechen. Zum anderen erleichtert es den prüfenden Stellen wie dem Senat und der Rechtsabteilung der Universität ihre Arbeit, indem eventuelle Fehler oder Unregelmäßigkeiten erkannt und kenntlich gemacht werden.

[Rz 30] Die Besonderheit des Regelwerks von CML ist die Kombination aus SWRL, der Regelsprache des Semantic Web, und Algorithmen der klassischen Softwareentwicklung. SWRL ist für eher einfache Regeln wie beispielsweise die Überprüfung, ob eine Lehrveranstaltung nicht fälschlicherweise anderen Bausteinen übergeordnet ist, optimal geeignet:

```
LV(?lv) uebergeordnet(?lv,?baustein) BausteinUngueeldig(?lv)
```

[Rz 31] Allerdings stößt diese Sprache bei manchen Anwendungsgebieten an ihre Grenzen. Diese Grenzen hängen häufig mit der in OWL gültigen Open World Assumption (OWA) zusammen. Diese besagt, dass zwischen nicht vorhandenem und negativem Wissen unterschieden werden kann, da im Unterschied zu geschlossenen Systemen aus einer fehlenden Information nichts geschlossen wird.¹¹ Das bedeutet aber, dass beispielsweise summenbasierte Regeln schwierig werden, da nie festgestellt werden kann, ob die momentan vorhandenen Teile auch tatsächlich alle sind, oder ob im offenen System noch ein weiteres Teil auftaucht.

[Rz 32] Eine Möglichkeit, diese Problematik zu umgehen, wäre der Einsatz der Semantic Query-Enhanced Web Rule Language (SQWRL)¹², die zeitpunktbezogene Abfragen wie in Datenbanksystemen ermöglicht. Allerdings ist SQWRL kein anerkannter Standard und wird daher auch von den meisten Frameworks und Reasonern nicht voll unterstützt. Die zweite Möglichkeit wäre der Einsatz von Closure Axiomen, also Fakten, die eine künstliche geschlossene Umgebung schaffen, indem das nicht vorhandene Wissen ausgeschlossen wird. Dies vergrößert die Ontologie jedoch massiv, und Änderungen müssen dann immer an mehreren Stellen durchgeführt werden. Daher kommen in CML Algorithmen in der Sprache Java zum Einsatz, in der auch die Algorithmen zur

¹¹ Vgl. GRIMM, STEPHAN, Knowledge Representation and Ontologies. In: Scientific Data Mining and Knowledge Discovery — Principles and Foundations, Springer, S. 210 ff. (2010).

¹² Vgl. O'CONNOR, MARTIN J./DAS, AMAR K., SQWRL: A Query Language for OWL. In: OWLED'09 (2009).

Erzeugung von Rechtstext formuliert sind.

3.3 Benutzeroberflächen

[Rz 33] Zu CML kompatible Modelle können zwar auch mit generalistischen Werkzeugen wie Protégé¹³ erzeugt und bearbeitet werden, diese stellen jedoch unerfahrene Benutzer vor unüberwindbare Schwierigkeiten. Auf die Bedürfnisse der verschiedenen Benutzergruppen zugeschnittene Benutzeroberflächen erleichtern die Arbeit mit semantischen Modellen stark, da so viele technische Aufgaben gekapselt und gewissermaßen vor dem Benutzer «versteckt» werden. Eine exemplarische Oberfläche stellt der «Curriculum Designer» dar, der es Studienkommissionen ermöglicht, die groben Blöcke eines Curriculums auf der grünen Wiese zu erstellen und anzuordnen. Abbildung 2 zeigt diese Oberfläche.

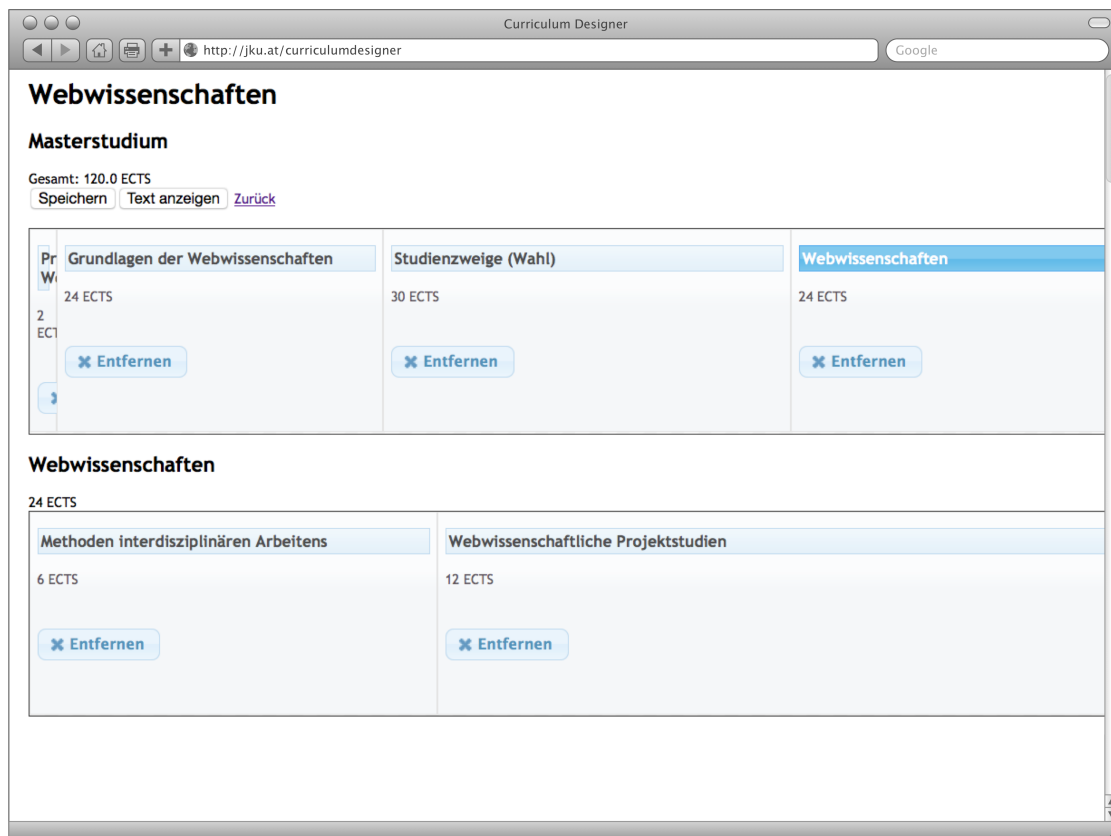


Abbildung 2: Weboberfläche Hierarchie

¹³ Vgl. <http://protege.stanford.edu>, Abruf am 15. Dezember 2014.

4 Fazit und Ausblick

[Rz 34] Mit Hilfe semantischer Technologien sind neben den gezeigten Funktionalitäten noch viele weitere Ergänzungen und neue Einsatzzwecke denkbar. Beispielsweise würde sich eine Verteilung der Verantwortung für Daten in der Hinsicht anbieten, dass jedes Institut bzw. jeder Lehrveranstaltungsleiter für die Informationen der eigenen Lehrveranstaltungen selbst verantwortlich ist und die Daten selbst bereitstellt. Dieses Prinzip ließe sich bei Studien, die von mehreren Bildungseinrichtungen gemeinsam angeboten werden, noch erweitern, indem Informationen aus mehreren Ontologien, die von den jeweiligen Einrichtungen veröffentlicht werden, zusammengeführt werden.

[Rz 35] Eine weitere Möglichkeit wäre die Erweiterung des Modells um individuelle Informationen jedes Studierenden. Diese könnten so ihren individuellen Stand im Studium markieren und automatisch generierte Vorschläge zum weiteren Verlauf und dem optimalen Weg durch das Studium erhalten.

[Rz 36] Durch personalisierte Benutzeroberflächen könnten die unterschiedlichen Nutzergruppen das volle Potenzial der Technik nutzen, die in Zukunft den kompletten Prozess der Erstellung und Pflege von Curricula, deren Genehmigungs- und Prüfverfahren sowie das Abrufen und die Weiterverwendung der gespeicherten Informationen abdecken könnten. Doch schon mit dem heutigen Stand der Forschung stellen die diskutierten Techniken eine große Hilfestellung und Mittel zu qualitativ hochwertigeren Normtexten mit Fokus auf Reduzierung bzw. Vermeidung von Missverständnissen, Unklarheiten und Fehlern dar.

JOHANN HÖLLER, Professor, Johannes Kepler Universität Linz, Institut für Datenverarbeitung in den Sozial- und Wirtschaftswissenschaften, Altenberger Straße 69, 4040 Linz, AT, johann.hoeller@jku.at; <http://www.idv.edu>

MARTIN STABAUER, Universitätsassistent, Johannes Kepler Universität Linz, Institut für Datenverarbeitung in den Sozial- und Wirtschaftswissenschaften, Altenberger Straße 69, 4040 Linz, AT, martin.stabauer@jku.at; <http://www.idv.edu>

Es sei darauf hingewiesen, dass bei der Entwicklung dieses Beitrags die praktische Umsetzbarkeit im Vordergrund stand. Zur theoretischen Einordnung siehe etwa Fußnote 6.