



# **Medizinische Entscheidungsunterstützung unterrichten: Konzepte und Werkzeugnutzung**

## **Bachelorarbeit**

im Studiengang  
Medizinische Informatik

vorgelegt von

**Kai Nico Groß**

Matrikelnummer: 185320

am 03.08.2018

an der Hochschule Heilbronn und Universität Heidelberg

Referent: Prof. Dr. Thomas Wetter  
Korreferent: Prof. Dr. Martin Haag

## Kurzfassung

Gegenstand der hier vorgestellten Arbeit ist ein Konzept zum Lehren von Ontologie in der Informatik, des Ontologiemanagementsystems Protégé und des Wissensverwaltungssystems ProKEt. Dies geschieht in Form von Präsentationen, welche mit Hilfe von PowerPoint erstellt wurden. Dabei wird auf die Installation, die Basics, Kernkonzepte und die Teilwissensbasen im Detail eingegangen. Diese Präsentationen sollen ein alternatives Konzept für den Unterricht darstellen, um die Lernenden aktiv zu beteiligen, und stellen ein Angebot zum Lehren der entsprechenden Thematiken dar. Zur Überprüfung der Effizienz wird eine Evaluation durchgeführt.

**Schlagwörter:** Ontologie, Protégé, ProKEt, Lehrkonzept

## Abstract

Subject of this thesis is a concept on teaching ontology in IT, the ontology management software Protégé and the knowledge administration system ProKEt. This is implemented with presentations created with PowerPoint. For that it discusses the installation, the basics, the central concept and the differentiated knowledge bases in detail. These presentations shall depict an alternative concept for the lessons where the students are engaged actively and offering a way of teaching these corresponding topics. An evaluation is carried out to verify the efficiency.

**Keywords:** ontology, Protégé, ProKEt, teaching concept

# Inhaltsverzeichnis

<b>Kurzfassung</b> .....	<b>2</b>
<b>Abstract</b> .....	<b>2</b>
<b>Inhaltsverzeichnis</b> .....	<b>3</b>
<b>1 Einleitung</b> .....	<b>5</b>
1.1 Motivation .....	5
1.2 Zielsetzung .....	6
1.3 Strukturierung.....	7
<b>2 Grundlagen</b> .....	<b>8</b>
2.1 Ontologie in der Informatik.....	8
2.2 Protégé.....	9
2.2.1 Die Entwicklung Protégés .....	10
2.3 ProKEt .....	11
<b>3 Konzeptziele</b> .....	<b>12</b>
3.1 Konzeptziele eines Lernenden.....	12
3.2 Konzeptziele eines Lehrenden.....	14
<b>4 Umsetzung der Konzeptziele für Protégé</b> .....	<b>15</b>
4.1 Ontologie und Protégé: Installation.....	15
4.1.1 Installation .....	16
4.1.2 Fehlerbehandlung .....	17
4.2 Protégé: Kernkonzepte .....	17
4.2.1 Anlegen und Speichern.....	18
4.2.2 Klassen .....	19
4.2.3 Object Properties .....	20
4.2.4 Data Properties .....	21
4.3 Protégé: Vertiefung .....	22
4.3.1 Property Chains .....	23
4.3.2 Class Expressions .....	24
4.3.3 Object Restriction.....	25
4.3.4 Reasoner .....	26
4.3.5 Visualisierung.....	27
4.3.6 Exportierung als Java-Code.....	28
<b>5 Umsetzung der Konzeptziele für ProKEt</b> .....	<b>29</b>
5.1 ProKEt: Allgemeines.....	29
5.2 ProKEt: Spalten .....	31

---

5.2.1	Standarttypen .....	31
5.2.2	Regeltypen .....	34
5.3	ProKEt: Typen .....	35
5.3.1	Fragentypen .....	35
5.3.2	Erklärungstypen .....	37
5.3.3	Sonstige Typen .....	38
<b>6</b>	<b>Fazit .....</b>	<b>41</b>
<b>Anhang A: Rohdaten der Umfrage zur Ermittlung der Konzeptziele der Lernenden .....</b>		<b>43</b>
<b>Anhang B: Ontologie und Protégé: Installation.....</b>		<b>45</b>
<b>Anhang C: Protégé: Kernkonzepte .....</b>		<b>52</b>
<b>Anhang D: Protégé: Vertiefung.....</b>		<b>69</b>
<b>Anhang E: ProKEt: Allgemeines.....</b>		<b>86</b>
<b>Anhang F: ProKEt: Spalten.....</b>		<b>92</b>
<b>Anhang G: ProKEt: Typen .....</b>		<b>106</b>
<b>Anhang H: Rohdaten der Umfrage zur Erwartungsmessung .....</b>		<b>124</b>
<b>Literaturverzeichnis .....</b>		<b>127</b>
<b>Erklärung .....</b>		<b>131</b>

# 1 Einleitung

## 1.1 Motivation

*„Ich komme kurz vor 8 Uhr in den Vorlesungsraum, setze mich an einen Platz am Fenster und hole meinen Block heraus. Kurz darauf kommt der Professor herein, begrüßt uns und der Großteil meiner Kommilitonen begrüßt ihn nahezu im Chor. Der Beamer wird eingeschaltet und die Vorlesung der letzten Stunde wird vom Professor mit einer solch gleichgültigen und monotonen Stimme fortgesetzt, dass ich denke, dass man die Vorlesung als Schlafmittel verstehen könnte. Ich schreibe das Datum des heutigen Tages auf mein Papier, falls ich mir Notizen machen werde, sehe dann aber doch lieber kurz aus dem Fenster. Es scheint ein schöner Tag zu werden, also sehe ich auf die Uhr wie lange ich dieses Thema noch über mich ergehen lassen muss. Ich stelle fest, dass wir seit 15 Minuten noch immer auf der selben Folie im Skript sind und entscheide mich dann das zu tun, was ich immer tue, wenn mir langweilig ist: Ich sehe nach ob ich E-Mails habe. Ich habe ein paar E-Mails mit Werbung aber sonst nichts weiter Interessantes, also wechsle ich zu WhatsApp und beginne mit ein paar Freunden zu schreiben. Als ich wieder auf die Uhr blicke stelle ich fest, dass die Vorlesung nur noch 10 Minuten dauert, also entschliefse ich mich, doch nochmal dem Professor zu lauschen. Ich bemerke, dass wir auf einmal bei einem völlig anderen Thema sind, bei dem ich wohl den Einstieg verpasst habe. Auch im Skript sind wir bereits über 20 Seiten weiter. Daraufhin zucke ich mit den Achseln, hoffe einfach das es nichts Klausurrelevantes war und entschliefse mich dazu weiter mit meinem Handy im Internet zu surfen.“ – Aus dem Leben eines Studenten.*

Ebenso geht es vielen Schüler/-innen und Studierenden, insbesondere wenn es sich um Themen handelt, die nur kurz unterrichtet werden oder für Prüfungen nicht von Relevanz sind. Wenn Schüler/-innen und Studenten/-innen von der Art des Unterrichts gelangweilt sind, bspw. wenn die Lehrkraft nur das Skript vorliest, ohne auf die Lernenden einzugehen, sinkt ihre Aufmerksamkeit wie auch die Bereitschaft aktiv dem Unterricht zu folgen. Um dies zu vermeiden, sollte der Unterricht so gestaltet sein, dass Schüler/-innen und Studenten/-innen aktiv daran teilhaben können, um so die Motivation und Lernbereitschaft zu steigern und zu erhalten. Durch die selbige wie oben genannte Erfahrung am eigenen Leib entstand der Ansporn und die Idee zumindest einen Teil einer Vorlesung für Studierende ansprechender zu gestalten und den Lehrenden eine alternative Möglichkeit anzubieten, den Stoff der Vorlesung zu vermitteln.

## 1.2 Zielsetzung

Ziel dieser Arbeit ist es, am Fachbereich Medizinische Informatik der Universität Heidelberg und der Hochschule Heilbronn, ein Konzept und dazu passende Präsentationen zu entwickeln, welche es ermöglichen, die Ontologie, die Software Protégé als Ontologie-Editor und das Wissensmanagementsystem ProKEt interaktiver vorzustellen und zu lehren.

Darüber hinaus soll diese Arbeit eine gut zu handhabende Ressource, sowohl zur Lehre als auch zum Lernen, darstellen. Es soll eine verwendbare Alternative zum reinen Frontalunterricht oder dem problemorientierten Unterricht entstehen. Dadurch sollen Protégé und ProKEt sowohl theoretisch als auch direkt im praktischen Bezug dem Motto „Learning by Doing“ entsprechend gelehrt und gelernt werden.

Folgende Leitideen liegen den Erstellten Präsentationen zu Grunde:

- Ein wesentliches Ziel dieser Arbeit besteht darin, die Konzeptziele der Lernenden in den Präsentationen umzusetzen. Dies soll für eine höhere Aufmerksamkeit und Lernbereitschaft sorgen.
- Ein weiteres Ziel besteht darin ebenfalls die Konzeptziele der Lehrenden in den Präsentationen zu erreichen.
- Die Verwendung dieser Arbeit soll unabhängig von einer Vorlesung und möglichst ohne besondere Vorkenntnisse möglich sein. Es soll ein eigenes Verständnis für die hier vorgestellten Themen entwickelt werden.

Das Ergebnis dieser Arbeit in Form von Präsentationen soll ein Angebot für alle Lehrenden und Lernenden darstellen, die sich mit den Thematiken Ontologie, Protégé und ProKEt beschäftigen. Zusätzlich dazu sollen die Präsentationen für die Lernenden gegebenenfalls als Nachschlagewerk dienen können, sowie den Lehrenden als Grundlage für eine Klausur dienen.

### **1.3 Strukturierung**

Ausgehend von den in Kapitel 1.2 genannten Zielen werden Ontologie, Protégé und ProKEt in Kapitel 2 Grundlagen kurz dargestellt. Dabei wird auf die Ontologie in der Informatik eingegangen sowie Protégé und ProKEt kurz beschrieben. Der Hauptteil der Arbeit beginnt in Kapitel 3 Konzeptziele mit der Ermittlung der Ziele, die diese Arbeit erfüllen soll. Dabei werden die Konzeptziele der Lernenden und der Lehrenden erarbeitet. Darauffolgend wird in den Kapiteln 4 und 5 die jeweilige Umsetzung dieser Konzeptziele für Protégé und ProKEt dargestellt. Abschließend wird im Kapitel 6 Fazit mit Hilfe von Umfragen ermittelt, ob die Umsetzung der Konzeptziele gelungen und damit das Ziel dieser Arbeit erreicht wurde.

## 2 Grundlagen

### 2.1 Ontologie in der Informatik

Die Ontologie ist ein Begriff der aus der Philosophie übernommen wurde. Er beschreibt dort die Lehre des Seins bzw. von den grundsätzlichen, allgemeinsten, elementarsten, fundamentalen und konstitutiven Eigenschaften, den Prinzipien, den grundsätzlichen Wesens-, Ordnungs- und Begriffsbestimmungen des Seins. [1] Sie beschäftigt sich dabei sowohl mit der Frage was es heißt, dass etwas existiert und mit der Frage, welche Kategorien von Objekten da sind und wie sie zueinander im Verhältnis stehen. [2]

In vielen Bereichen der Informatik steht man vor der Aufgabe, Erkanntes oder Erdachtes zu repräsentieren und Wissen zu kommunizieren, z.B. über Fakten, Sachverhalte oder Regeln in einem technischen Anwendungsbereich. Im Alltag macht man sich gespeichertes und abgelegtes Wissen zu Nutze in Form von Regelwerken, Lexika und anderer Sammlungen von Wissen. Soll dieses Verwenden dagegen von einem Programm übernommen werden, um z.B. Entscheidungsaufgaben in Bezug auf gespeichertes Wissen und Daten, die selbst Informationen zur Verfügung stellen, wie dieses Wissen zu interpretieren ist (die sogenannten Metadaten), so benötigt dieses Programm eine Repräsentation der zugrunde liegenden Begriffe der Wissensansammlung und deren Zusammenhänge. Für diese Repräsentation hat sich in der Informatik der Begriff Ontologie geprägt. [3]

T. R. Gruber hat 1993 die Definition einer Ontologie wie folgt beschrieben:

*„Eine Ontologie ist eine explizite formale Spezifikation einer gemeinsamen Konzeptualisierung.“* [4]

Eine Konzeptualisierung wird nach Genesereth und Nilsson betrachtet als „Ein Körper formell repräsentierten Wissens basiert auf einer Konzeptualisierung: den Objekten, Konzepten und anderen Entitäten, von denen angenommen wird, dass sie in einem bestimmten Interessensbereich existieren und die Beziehungen zwischen Ihnen bestehen. Eine Konzeptualisierung ist eine abstrakte, vereinfachte Sicht auf die Welt, die aus einem Zweck heraus dargestellt werden soll. Jede Wissensbasis, jedes wissensbasierte System oder jeder Akteur auf einer Wissensebene ist an eine explizite oder implizite Konzeptualisierung gebunden“ [Vom Autor übersetzt von [5].]

Davon ausgehend beschreibt eine Ontologie einen Wissensbereich mit Hilfe standardisierter Terminologie sowie Beziehungen und Ableitungsregeln zwischen den dort definierten Begriffen. Dieses Vokabular basiert meist auf einer Taxonomie oder im Bereich



der Medizinischen Informatik auf einem medizinischen Leitfaden, der grundlegende Ausgangselemente wie Klassen, Relationen, Funktionen und Axiome enthält. [3]

Die Verwendung von Ontologien in der Informatik wird von Gruninger und Lee in drei Anwendungsfelder unterschieden:

- Kommunikation
- Automatisches Schließen bzw. Herleiten
- Repräsentation und Wiederverwendung von Wissen

Beim Automatischem Schließen und Herleiten ist es Programmen möglich, logische Schlüsse und Lösungen anhand der durch die Ontologie gegebenen Ableitungsregeln herzuleiten. Dies ist insbesondere im Bereich der künstlichen Intelligenz von Bedeutung, ebenso wie in der medizinischen Entscheidungsunterstützung. [3]

Zur Entwicklung und Prüfung von Ontologien entwickelten sich diverse Sprachen, Methoden und Werkzeuge, von denen insbesondere der RDF/XML-Syntax zu nennen ist. XML (Extensible Markup Language) wird dabei für die Annotierung und Strukturierung von Daten und Dokumenten verwendet, wohingegen RDF (Resource Description Framework) die Möglichkeit bietet, Ressourcen durch Eigenschaften zu beschreiben und diesen Werte zuzuordnen. Darauf basiert die Idee semantische Netze und damit auch Ontologien als Graphen aufzufassen. [3]

## 2.2 Protégé

Protégé ist eine Open-Source-Anwendung die am Center for Biomedical Informatics Research (BMIR) an der Universität Stanford in Kalifornien entwickelt wurde um Ontologien zu entwickeln, modellieren und pflegen. Das BMIR entwickelt Möglichkeiten zur Erfassung, Darstellung und Verarbeitung von Informationen über die menschliche Gesundheit. Protégé ermöglicht es bei richtiger Verwendung biomedizinische Daten in entscheidungsrelevante Erkenntnisse umzusetzen. Zusätzlich unterstützt das BMIR die Weltgesundheitsorganisation bei der Schaffung einer technologischen Infrastruktur für die Entwicklung und Verbreitung der Internationalen Klassifikation der Krankheiten (International Classification of Diseases (ICD), aktuell in Verwendung in Form der Version 11), der Internationalen Klassifikation der traditionellen Medizin und anderer Ontologien. [6] Laut der Startseite von Protégé wird die Software von bereits mehr als 300.000 registrierten Benutzer verwendet. [7]

### 2.2.1 Die Entwicklung Protégés

1972 begannen Bruce G. Buchanan und Edward H. Shortliffe in Stanford mit der Entwicklung des MYCIN-Projekts in Stanford, eines Expertensystems zur Diagnose von Infektionskrankheiten. [8] Es ist ein Programm mit einer künstlichen Intelligenz das mit folgenden Eigenschaften ausgestattet werden sollte:

- Zur Lösung komplexer Probleme auf Expertenniveau.
- Nachvollziehbar und verständlich.
- Ausreichend flexibel um leicht neues Wissen einfügen zu können.

Ein Expertensystem wie MYCIN beinhaltet dabei zwei Hauptbestandteile, eine Wissensbasis und einen Inferenzmechanismus. [9] Jedoch konnte das Ziel der Flexibilität nicht erfüllt werden, woraufhin sich EMYCIN entwickelte. Die Flexibilität, die MYCIN fehlt, um Wissensbasen zu erweitern oder zu modifizieren wurde in EMYCIN umgesetzt, einem Programm zur Erstellung von Regelbasierten Expertensystemen. Dabei wurden die Konzepte des Kontexts, der Parameter und der Regeln generalisiert, um den Zugriff auf diese zu erleichtern. Ein Schwerpunkt bei der Entwicklung von EMYCIN bestand darin, dass der Fokus nun auf der Person lag, die eine Wissensbasis erstellt, anstelle des Nutzers dieser Wissensbasis. Dadurch kam es zu einem besseren Interface, für die Systemerstellung, das transparenter und leichter zu verwenden war. Es wurde versucht, den Aufwand, der benötigt wird, um ein Expertensystem zu erstellen, zu reduzieren. MYCIN selbst wurde erfolgreich als eine neue Instanz von EMYCIN reimplementiert. [10]

Aus EMYCIN entstand unter Mitarbeit von Edward H. Shortliffe das Oncology Protocol Management System (ONCOCIN). Ein System zur besseren Krebsbehandlung durch Therapieplanung. ONCOCIN war dazu gedacht, Ärzten mit einer verbesserten Datensammlung und Entscheidungsfindung zu unterstützen. [11]

Laut Professor Wetter führte die Abstraktion für andere Krankheiten durch Edward Shortliffe und Mark Musen dazu, dass das integrierte System in den allgemein informatischen Teil und das spezifisch medizinische Fachwissen getrennt wurde. Durch diese Abspaltung des informatischen Teils wurde um 1998 Protégé als eigene Software veröffentlicht.

## 2.3 ProKEt

ProKEt ist ein Prototyping- und Engineering-Tool für wissensbasierte Systeme. Es kann genutzt werden, um Prototypen, durch das Hinzufügen geeigneter Datenmodelle in Form von Wissensbasen, in funktionale wissensbasierte Systeme umzuwandeln. [12]

Mit ProKEt wird versucht die folgenden Probleme zu lösen:

- Wie kann Expertenwissen formalisiert werden?
- Wie kann ein leicht anpassbares Frontend ermöglicht werden, so dass die Nutzer ihren eigenen angepassten webbasierten Zugriff auf das wissensbasierte System nutzen können? [Übersetzt vom Autor von [13]]

Zur Formalisierung verwendet ProKEt eine tabellarische Form, die im .xlsx-Format erstellt wurde. Dadurch können bereits bestehende und weit verbreitete Programme wie Microsoft Excel verwendet werden, die eine hohe Qualität bieten. [13]

## 3 Konzeptziele

Um ein geeignetes Konzept für eine Präsentation zu entwickeln, muss zuerst festgelegt werden, welche Ziele und Zwecke das Konzept der Präsentation erfüllen soll. Dabei ist zu unterscheiden, was für einen Lernenden, wie einen Studenten oder Schüler, ein gelungenes Konzept einer Präsentation darstellt und welche Ziele der Lehrende, zum Beispiel ein Professor oder ein Lehrer, mit dem Konzept erreichen will. Um ein Präsentationskonzept als erfolgreich zu betrachten, ist es von Nöten, dass diese Ziele von beiden Seiten möglichst gut erfüllt werden. Im Folgenden werden zuerst die Ziele aus der Sicht der Lernenden betrachtet und daraufhin die Ziele des Konzepts eines Lehrenden.

### 3.1 Konzeptziele eines Lernenden

Um herauszuarbeiten was für einen Lernenden an einem Präsentationskonzept wichtig ist, wurde eine Reihe von zufällig ausgewählten Schülern und Studenten mündlich befragt. Es stellte sich heraus, dass es den meisten Lernenden überaus wichtig ist, dass die Präsentation möglichst gut verständlich und selbsterklärend ist. Aus demselben Grund legen die Lernenden Wert darauf, dass die Präsentationen vollständig sind, ohne jedoch zusätzliches „unbrauchbares“ Wissen. Die Präsentation sollte so gestaltet sein, dass der Lernende dieser seine volle Aufmerksamkeit widmen kann. Dies sollte möglich sein ohne jedes Wort des Lehrenden zu verfolgen, um nötigenfalls Wissen, welches nicht in der Präsentation vorhanden ist, mitschreiben oder notieren zu müssen. Dies wurde vor allem auch im Zusammenhang erwähnt, sollte man bei der Präsentation selbst nicht anwesend sein können. Weiterhin wurde erwähnt, dass die Präsentationen nicht mit Wissen überladen sein sollten, sondern nur das enthalten sollen, was tatsächlich gebräuchlich und anwendbar ist. Es soll also erneut unnützes Wissen vermieden werden, um den Lernenden nicht zu überfordern. Ein weiterer Punkt, welcher immer wieder im Gespräch mit den Lernenden zu Tage kam, ist die mögliche Mitarbeit bzw. Einbindung der Lernenden in die Präsentation, sodass diese das gerade erworbene Wissen direkt anwenden können und sich „*nicht nur von dem Stoff berieseln lassen müssen*“ (Ein Lernender während des Gesprächs über die Ziele einer Präsentation). Der gelehrte Stoff soll direkt angewendet werden, um das Wissen darüber zu festigen und das Interesse daran zu erhalten. Insgesamt soll die Präsentation eine gute handhabbare Ressource darstellen.

Zusammenfassend wurden von den Lernenden folgende Ziele genannt:

1. verständlich und selbsterklärend
2. Vollständigkeit
3. Kein überflüssiges Wissen

#### 4. Direktes Anwenden des Wissens

Zusätzlich zu der mündlichen Befragung wurde eine Onlineumfrage erstellt, bei der gefragt wurde, was den Lernenden an einer Präsentation im Rahmen einer Vorlesung wichtig ist. Insgesamt nahmen 33 Personen an der Umfrage teil. In der untenstehenden Grafik wurden die Ergebnisse der Umfrage, nach herausfiltern unsachgemäßer Antworten, dargestellt. Dabei ist zu beachten, dass mehrere Antworten gegeben werden konnten. Die Rohdaten der Umfrage können in *Anhang A: Rohdaten der Umfrage zur Ermittlung der Konzeptziele der Lernenden* nachgeschlagen werden. Auch in den Rohdaten wurde unsachliche Antworten bereits herausgefiltert.

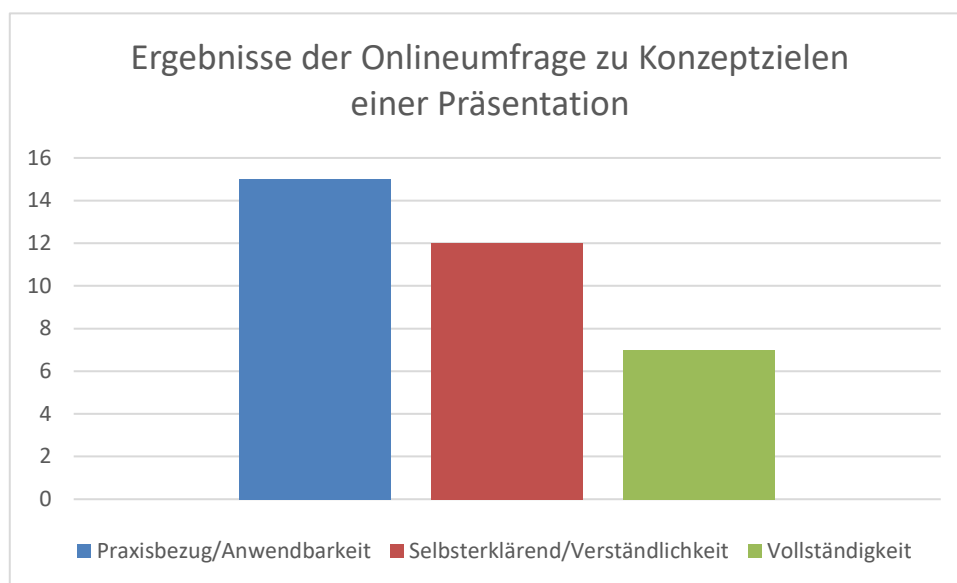


Abbildung 1: Ergebnisse der Onlineumfrage zu Konzeptzielen einer Präsentation

Deutlich zu erkennen ist, dass ein großer Teil der Studierenden den Praxisbezug bzw. die Anwendbarkeit für sehr wichtig erachtet, ebenso wie Verständlichkeit gefordert wird. Das Ergebnis dieser Umfrage deckt sich mit den Ergebnissen der mündlichen Befragung, mit Ausnahme des Punktes „Kein überflüssiges Wissen“.

Durch die mündliche Befragung und der Onlineumfrage kann schlussendlich gesagt werden, dass die Konzeptziele einer Präsentation für Lernende die folgenden Punkte sind:

1. Praxisbezug und Anwendbarkeit
2. Selbsterklärend und Verständlichkeit
3. Vollständigkeit (ohne überflüssiges Wissen)

## 3.2 Konzeptziele eines Lehrenden

Zur Erarbeitung der Konzeptziele eines Lehrenden wurde Prof. Dr. Thomas Wetter mündlich befragt sowie nach den Zielen des Lehrkonzepts in seinem Interesse, als Nutzer dieser Arbeit, in schriftlicher Antwortform gebeten.

*„Im Bereich der Methoden und zugehöriger Werkzeuge zur Repräsentation von Wissen und zur Entscheidungsunterstützung sollen sie Studenten die Charakteristika ausgewählter Methoden erkennen und dadurch in die Lage versetzt werden, in künftigen Anwendungssituation die je bestgeeignete Methode auszuwählen und anzuwenden.*

*Bezüglich Protégé sollen die Studenten die Grundidee einer Ontologie im Sinne der Informatik als eine formale Beschreibung der Eigenschaften und Zusammenhänge von Gegebenheiten eines Anwendungsbereichs verstehen und anwenden können. Hierzu soll beitragen, dass sie an konkreten medizinischen Beispielen wie etwa Behandlungsleitlinien das dort sprachlich kodifizierte Wissen extrahieren, bewerten und in formalsprachliche Konstrukte von Protégé abbilden. Dabei sollen ihnen die vielfältigen Gestaltungsmöglichkeiten durch abstrakte und konkrete Klassen, Properties und deren Typen- und Wertebereichsbeschränkungen, Instanzen und Mehrfachvererbung bewusstwerden, einschließlich einer Urteilsfähigkeit, welche formalsprachlichen Konstrukte sich für welche Gegebenheiten wie gut eignen.*

*Bezüglich ProKEt sollen die Studenten die Grundeigenschaften eines regelbasierten Systems verstehen und in Beziehung setzen können zu regelhaften Gegebenheiten eines Anwendungsgebietes. Hierzu sollen sie regelhafte Formulierungen in sprachlichen Beschreibungen des Anwendungsgebietes in die von ProKEt vorgegebenen formalen Darstellungen abbilden können und sich ein Bild machen können von der Bedeutung und Verwendung unterschiedlicher Fragetypen, der Rolle von Gewissheitskennzeichnungen von Regeln und deren Abhängigkeit von unterschiedlichen Inferenzstrategien. Sie sollen in der Lage sein, konkrete regelbasierte Prototypen zu entwickeln und den Wert der Nutzung der ProKEt-Entwicklungsumgebung für schnelles Prototyping erkennen.“ – Prof. Dr. Thomas Wetter.*

Aus dieser Antwort, wie auch aus der mündlichen Besprechung mit Prof. Wetter ergeben sich als Hauptziele des Lehrkonzepts eines Lehrenden die Verständlichkeit und Anwendbarkeit. Es wird gefordert, dass die Lernenden den Stoff verstehen und anwenden können. Dies überschneidet sich mit den Zielen der Lernenden in Form der Verständlichkeit und des Praxisbezuges bzw. der Anwendbarkeit.

## 4 Umsetzung der Konzeptziele für Protégé

In den folgenden Abschnitten wird die Erstellung der einzelnen Präsentationen über Protégé beschrieben unter den Kriterien der Konzeptziele sowohl von der Seite der Lernenden als auch von der Seite der Lehrenden. Die Präsentationen finden sich im Anhang abgedruckt wieder, sowie auf dem dieser Arbeit beiliegendem Datenträger als Power-Point-Präsentation.

### 4.1 Ontologie und Protégé: Installation

In dieser ersten Präsentation über Protégé geht es darum, den Lernenden einen Überblick zu verschaffen was Ontologien sind, wofür sie eingesetzt werden und wozu Protégé verwendet wird. Im Anschluss wird die Installation von Protégé schrittweise beschrieben. Die Präsentation befindet sich in *Anhang B: Ontologie und Protégé: Installation*.

Zu Beginn der Präsentation wird in einem Inhaltsverzeichnis darauf aufmerksam gemacht worum es sich in dieser ersten Präsentation von Protégé handelt. Dies dient dazu, den Lernenden später zu helfen, schneller Wissen über die behandelten Themen wiederzufinden bzw. herauszufinden ob das gesuchte Wissen in dieser Präsentation vorhanden ist. Auf den folgenden Seiten wird kurz erklärt, wozu die Ontologie in der Informatik verwendet wird und weshalb sie zur Entwicklung von entscheidungsunterstützender Software verwendet wird. Dieses Wissen wurde bereits im Grundlagenkapitel 2.1 *Ontologie in der Informatik* behandelt und kann von Lehrenden verwendet werden, um mündlich ausführlicher auf die Ontologie einzugehen. Gegebenenfalls kann dann zusätzlich Lehrmaterial verwendet werden, je nach der für diesen Lehrpunkt eingeplante und zur Verfügung stehende Zeit.

Im Abschnitt Protégé wird zuerst die Verwendung von Protégé vorgestellt, um zu vermitteln aus welchen Gründen sich Protégé entwickelt hat und weshalb es in der Entwicklung entscheidungsunterstützender Software verwendet wird. Der Abschnitt zur Geschichte Protégés ist bewusst nicht in die Präsentation integriert, da es sich hier um Wissen handelt welches vom Lernenden nicht oder nur in geringem Ausmaße benötigt wird, es handelt sich gewissermaßen um irrelevantes Wissen. Auch hier kann der Lehrende, je nach verfügbarem Zeitrahmen, ausführlicher auf die Entwicklung von Protégé eingehen mit Hilfe des Grundlagenkapitels 2.2 *Protégé*, in welchem diese ausführlich geschildert ist.

### 4.1.1 Installation

Auf allen verbliebenden Seiten der Präsentation wird nun ausführlich die Installation von Protégé beschrieben, sowohl die für Windows kompatible Version als auch die der Plattform unabhängigen Independent-Version, welche für Linux, Mac OS und auch für Windows verwendet werden kann. Hierbei ist zu beachten, dass für die Independent-Version Java 8 oder eine spätere Version bereits installiert sein muss. Anschließend werden mögliche Installationsfehler als auch deren Lösung beschrieben. Hierbei wurde darauf geachtet, dass jeder Schritt grafisch dargestellt wird und ausführlich beschrieben ist, um es jedem Lernenden selbstständig zu ermöglichen die Installation problemlos durchzuführen, um Protégé dann verwenden zu können. Dies entspricht zum einem dem Punkt der von den Lernenden gewünschten Vollständigkeit, sowie auch der Tatsache, dass, falls es am Ort der Präsentation aus technischen Gründen nicht möglich sein sollte das Internet zum Herunterladen verwenden zu können, die Installation auch unabhängig an jedem anderen Ort, an dem eine Internetverbindung zur Verfügung steht, durchgeführt werden kann.

Die Installation besteht aus den im Folgenden acht beschriebenen Schritten:

1. Das Aufrufen der Seite <https://protege.stanford.eu>, der Startseite von Protégé.
2. Über die Startseite kann man nun über den „Download Now“-Button zur Downloadseite gelangen.
3. Auf der Downloadseite von Protégé kann nun entweder die Windows-Version oder alternativ die Independent-Version zum Herunterladen ausgewählt werden. Wie bereits erwähnt wird für die Independent-Version Java 8 oder spätere Versionen benötigt.
4. Darauf bietet Protégé eine Registrierung an, die jedoch auch abgelehnt werden kann. Die Registrierung hilft den Entwicklern von Protégé durch die Verwendung der Nutzerdaten Protégé weiterhin zu optimieren und den Nutzer per E-Mail über Updates zu informieren.
5. Nun kann festgelegt werden an welchem Zielort Protégé abgespeichert werden soll.
6. Nach der Navigation in den entsprechend zuvor gewählten Ordner muss die Datei extrahiert werden.
7. Nun kann Protégé gestartet werden. In Windows muss dazu *Protege.exe* ausgeführt werden während unter Mac OS *run.command* und unter Linux *run.sh* verwendet werden.



8. Nach dem erstmaligen Start schlägt Protégé eine Reihe von Updates und Plugins vor die installiert werden. Es wird hier empfohlen sämtliche verfügbaren Updates zu installieren um Protégé bestmöglich verwenden zu können.

### 4.1.2 Fehlerbehandlung

Bei der Installation kann es selten zu einem Fehler kommen bei dem der Reasoner FaCT++ (Kapitel 3.2.3 *Protégé: Vertiefung*) nicht richtig installiert wird. Dabei wird beim Versuch Protégé zu starten der Fehler „FaCT++ requires platform specific libraries which cannot be found!“ gezeigt. Dieser Fehler kann durch das Löschen und anschließende Neuinstallieren des FaCT++-Plugins wie folgt behoben werden:

1. Protégé muss beendet werden.
2. Im Installationsordner von Protégé muss im Unterordner Plugins die Datei *uk.ac.manchester.cs.owl.factplusplus-x.y.z.jar* gelöscht werden, dabei stehen x, y und z für die Versionsnummer von FaCT++.
3. Zusätzlich müssen nun unter *C:\Users\Username\AppData\Local\Temp\* alle Dateien und Ordner, die mit „ProtegeCache“ beginnen, gelöscht werden. Diese Dateien sind von Protégé zwischengespeichert und enthalten noch Informationen über das nun gelöschte FaCT++-Plugin, was beim Starten von Protégé zu erneutem Fehlerauswurf führen würde.
4. Anschließend kann Protégé neugestartet werden. [14]

Der Nutzer von Protégé kann selbst entscheiden ob FaCT++ über den sich in Protégé befindenden Tab „Help“ und dem Unterpunkt „Protégé Plugins“ neu installiert wird oder alternativ ein anderer Reasoner, wie zum Beispiel HerMiT, verwendet werden soll. Im Regelfall jedoch wird Protégé beim Start anbieten FaCT++ und eventuell weitere Updates zu installieren.

## 4.2 Protégé: Kernkonzepte

In dieser Präsentation von Protégé soll es dem Lehrenden ermöglicht werden den Lernenden die grundlegenden Möglichkeiten und Funktionen von Protégé verständlich und nachvollziehbar darzustellen. Die Präsentation hierzu befindet sich in *Anhang C: Protégé: Kernkonzepte*.

Zu Beginn wird aus den bereits oben genannten Gründen ein Inhaltsverzeichnis der Präsentation gezeigt. Um jedoch die Möglichkeiten von Protégé grundlegend zu lehren und lernen ist es wichtig zu differenzieren, welche Funktionen von Protégé als grundlegend

zu betrachten sind und welche erst benötigt werden, wenn man bereits ein Grundverständnis für Protégé erlangt hat. Daher wurde im folgenden Wert darauf gelegt, alles möglichst vollständig und nachvollziehbar, wie von den Lernenden gewünscht, darzustellen. Dabei werden zusätzlich Grafiken verwendet, um dies zusätzlich zu veranschaulichen und den Lernenden noch während der Präsentation die Möglichkeit zu bieten mitzuarbeiten und das gelernte Wissen direkt anzuwenden. Zusätzlich wurden dafür Aufgaben eingebracht, die helfen sollen, das Wissen über die Anwendung von Protégé zu festigen. Um die Präsentation möglichst vollständig und nachvollziehbar zu gestalten wird im Folgenden davon ausgegangen, dass die Lernenden zumindest geringe Vorkenntnisse in Informatik besitzen. Auf diese Weise wird es auch Lernenden ermöglicht sich mit Protégé zu befassen, die nur geringe Erfahrung in Informatik gesammelt haben. Als benötigtes Vorwissen wird die Kenntnis von Klassen, Objekten und Properties vorausgesetzt.

#### 4.2.1 Anlegen und Speichern

Im Abschnitt „Anlegen und Speichern eines Projektes“ wird zuerst darauf hingewiesen, dass in Protégé erstellte Projekte später auch veröffentlicht werden können. Dabei wird bei der Erstellung des Projektes im Feld „Ontology IRI“ der Projektname direkt im Zusammenhang mit einem lokalen Pfad oder einer Online-Adresse verwendet. Protégé bietet hierbei immer an, auf dem Server von [www.semanticweb.de](http://www.semanticweb.de) sein Projekt abzulegen, wo es nach Abschluss auch veröffentlicht werden kann. Dies wird in der Präsentation auf Seite 3 beschrieben und mit einer Grafik auf Seite 4 verdeutlicht. Auf dieser Grafik lässt sich erkennen, dass das Projekt „Basics\_of\_Protégé“ auf dem Server von [www.semanticweb.de](http://www.semanticweb.de) abgespeichert wird.

Darauffolgend wird auf den Seiten 5 und 6 beschrieben und gezeigt wird man in Protégé einen Kommentar anlegen kann. Dies soll dazu dienen, dass jeder Benutzer sich Informationen notieren kann, um später nachzuvollziehen wofür z.B. eine bestimmte Klasse erstellt wurde. Lehrenden und Lernenden, die bereits Vorkenntnisse in Informatik haben, sollte an dieser Stelle bereits bewusst sein, wie wichtig Kommentare während einer Entwicklung sein können. [15]

Auf Seite 7 der Präsentation wird darauf hingewiesen sowie grafisch betont, dass Projekte im „RDF/XML-Syntax“ abgespeichert werden sollen. Dieser Syntax ist die modernste in Protégé verfügbare Syntax und sollte dementsprechend verwendet werden, um zu gewährleisten, dass es beim Speichern nicht zu Kompatibilitätsproblemen kommen kann. Genaueres zum „RDF/XML“-Syntax lässt sich im Grundlagenkapitel 2.1 *Ontologie in der Informatik* nachlesen.

### 4.2.2 Klassen

Es folgt in der Präsentation ab Seite 8 nun der Abschnitt „Klassen in Protégé“. Auf dieser Seite wird beschrieben, dass um Klassen zu erstellen und bereits erstellte Klassen sehen zu können, zuerst in Protégé auf den Reiter „Entities“ und daraufhin auf den Subreiter „Classes“ gewechselt werden muss. Eine Grafik soll verdeutlichen wo der entsprechende Reiter zu finden ist. Als Nächstes wird darauf hingewiesen, dass in Protégé alle Klassen eine Subklasse von „owl:Thing“ sind und damit auch von dieser Klasse erben. Es können in Protégé Eigenschaften von Klassen, wie z.B. in der Programmiersprache Java, vererbt werden.

Um nun Klassen zu erstellen, kann entweder gezielt eine einzelne Klasse mit Hilfe des „Add Subclass“-Buttons an einer ausgewählten Stelle der Hierarchie erstellt werden oder durch die Funktion „Add Subclasses“, welche sich über einen Rechtsklick auf entsprechende Superklasse aufrufen lässt. Bei der gleichzeitigen Erstellung mehrere Subclasses gibt es die Möglichkeit diese auch direkt in der gewünschten Klassenhierarchie entsprechend anzuordnen indem durch „Tab“-Einschübe die Hierarchie abgebildet wird. [16] Das in diesem Abschnitt Beschriebene findet sich in der Präsentation auf den Seiten 10 bis 15 und wird auch wieder durch beispielhafte Grafiken verdeutlicht. Auf den beiden folgenden Seiten widmet sich die Präsentation nun dem „Disjoinen“ gewidmet. Dieses dient dazu, dass die erstellten Klassen untereinander differenziert werden und nicht dieselbe Instanz teilen können. [17] Um dies zu verdeutlichen kann der Lehrende folgendes Beispiel erwähnen:

Apfel und Banane sind beides Subklassen von Obst. Wenn man nun ein Stück Obst hat, dann kann dieses entweder ein Apfel oder eine Banane sein, aber nicht beides.

Auf Seite 18 ist beispielhaft gezeigt wie die erstellten Klassen angezeigt und hierarchisch aufgebaut sind. Um den Lernenden nun die Möglichkeit zu geben, das obige Wissen zu Klassen zu festigen, ist auf den Seiten 19 bis 23 eine Aufgabe zum Erstellen von Klassen formuliert. Sie kommt aus dem Bereich der Medizin, handelt von der Untersuchung und Diagnostizierung von Brustkrebs und lautet wie folgt:

„Zu den klinischen Untersuchungen gehören die Inspektion der Mamillen sowie der Mamillen-haut. Bei der Untersuchung der Mamillen ist zwischen benigner und maligner Qualität des Sekrets zu unterscheiden und die Seitigkeit des Sekrets zu bestimmen. Bei der Untersuchung der Mamillenhaut ist der Hautbefund und die Nicht-Therapierbarkeit zu überprüfen. Weiterhin gehören zu den klinischen Untersuchungen die Palpation mit einem Tastbefund, dessen Ergebnis benigne oder maligne sein kann, und die klinische Untersuchung zu Schmerzen. Zu den weiteren Untersuchungen gehören die Feinnadelpunktion, Kernspin, Mammographie und Ultraschall. Es

soll eine Klasse „Patient“ entstehen mit der Subklasse „Alter“, in der unterschieden wird, ob die Patientin unter 31 oder 31 und älter ist. Die Klasse Diagnose erhält Untersuchungs-indikationen zu jeder der „weiteren Untersuchungen“ sowie zu „keiner Indikation“. Die Klasse „Diagnostische Beziehung“ besteht aus der heuristischen Beziehung und überdeckender Beziehung. Die heuristischen Beziehungen bestehen aus den Symptominterpretationen von verdächtigen Mamillen, verdächtiger Mamilienhaut, verdächtiger Palpation und Vorerkrankungen.“

Diese Aufgabe ist geschachtelt geschrieben, um die Lernenden etwas zu fordern, aber nicht zu überfordern, und um das Wissen über das Erstellen von Klassen möglichst intensiv zu festigen, da es eine der grundlegendsten Eigenschaften von Protégé darstellt. Um die Möglichkeit der Überprüfung zu bieten, wird auf Seite 24 der Präsentation die Lösung der obigen Aufgabe mit einer Grafik dargestellt. Der Lehrende kann an dieser Stelle der Präsentation, je nach vorhandener Zeit, die gesamte geschilderte Aufgabe verwenden oder alternativ auch nur einen Teil davon. Die Aufgabe lässt sich gut unterteilen, da in der Präsentation selbst die Aufgabe auf den fünf verwendeten Folien jeweils als ein Abschnitt formuliert wurde.

### 4.2.3 Object Properties

Ab Seite 25 werden in der Präsentation die Object Properties behandelt. Object Properties beschreiben die Verbindungen zwischen Klassen und deren Instanzen. Ähnlich wie bei den Klassen muss auch bei den Object Properties in Protégé über den „Entities“-Reiter zu dem „Object Properties“-Subreiter navigiert werden, um die bereits bestehenden Object Properties sehen und weitere erstellen zu können. Die Object Properties erben, wie auch bei den Klassen, alle von einer obersten Object Property, der „owl:topObjectProperty“, und können äquivalent zu den Klassen erstellt werden. Auf Seite 28 werden beispielhaft 3 Object Properties erstellt, was zusätzlich durch eine Grafik dargestellt wurde. Da Object Properties Klassen miteinander verknüpfen, muss der Object Property die entsprechenden Klassen zugewiesen werden, wobei auch die Richtung der Verknüpfung zu beachten ist. Um die gewünschten Klassen mit einer Property zu verbinden gibt es die „Domain“ und die „Range“, die in der Property selbst festgelegt werden. Dabei kann eine Property auch mehrere Domains und Ranges besitzen. Die Domain einer Property beschreibt welche Klasse die Object Property verwendet, wohingegen die Range angibt, welche Klasse durch die Object Property verwendet wird. Die Domain zeigt also über die Property auf die Range. Auf der folgenden Seite 30 soll dies an einem Beispiel beschrieben werden. „Eine Diagnostische Beziehung erlaubt einen Schluss einer Diagnose“. Es soll also die Object Property „erlaubt\_Schluss“ von der Klasse „Diagnostische\_Beziehung“ verwendet werden um auf die Klasse „Diagnose“ zu zeigen. Um dies

umzusetzen muss die Property „erlaubt\_Schluss“ ausgewählt sein. Dann kann im Bereich „Domain“ über den „+“-Button ein Fenster geöffnet werden in welchem im Reiter „Class hierarchy“ die gewünschte Klasse „Diagnostische\_Beziehung“ ausgewählt werden kann. Dies wird zusätzlich durch eine Grafik auf Seite 31 verdeutlicht. Anschließend kann parallel dazu die Klasse „Diagnose“ als Range festgelegt werden, wie auf Seite 32 beschrieben und auf Seite 33 erneut grafisch betont. [18] [19]

Um den Lernenden auch hier wieder die Möglichkeit zu geben, das bis zu diesem Punkt erfragene Wissen über Object Properties zu festigen, findet sich eine Übungsaufgabe dazu auf den Seiten 34 und 35 mit der entsprechenden Lösung auf Seite 36. Bei der Aufgabe sollen die folgenden Object Properties erstellt werden:

- Ein Patient hat einen Datensatz, der Untersuchungen, Diagnosen und diagnostische Beziehungen enthält.
- Eine diagnostische Beziehung nutzt die Werte der Untersuchung(en).
- Symptominterpretation führt zu einer klinischen Untersuchung
- Ein Untersuchungsindikator führt zu einer weiteren Untersuchung.

In diesem Beispiel können die beiden letzten Object Properties Subproperties von „fuehrt\_zu“ sein, um gegebenenfalls zu einem späteren Zeitpunkt die Vererbung nutzen zu können.

#### 4.2.4 Data Properties

Der folgende Abschnitt ab Seite 37 handelt von Data Properties in Protégé. Data Properties beschreiben die Beziehungen von Klassen und Instanzen mit Datenwerten. Sie sind vergleichbar mit Variablen in Java. Um zu den Data Properties zu gelangen muss in Protégé erneut über den „Entities“-Tab zu dem Subtab „Data properties“ navigiert werden. Wie schon bei den Klassen und den Object Properties erben alle Data Properties von einer obersten Data Property, der „owl:topDataProperty“ und werden äquivalent wie Klassen und Object Properties erstellt. Auf Seite 40 sollen dann die Data Properties Patientendaten, bestehend aus einer Patientenummer sowie Vor- und Nachname, erstellt werden was auch wieder grafisch dargestellt wird. Wie auch zuvor bei den Object Properties besitzen die Data Properties Domains und Ranges, allerdings wird bei der Data Property im Gegensatz zu der Object Property mit der Range angegeben in welchem Datentyp die entsprechenden Daten vorliegen müssen, damit sie als gültig betrachtet werden. Auf der folgenden Seite 42 wird dann beschrieben, dass bei den erstellten Data Properties Patientenummer, Vor- und Nachname keine Domain eingetragen werden muss, wenn bei der darüberstehenden Data Property „Patientendaten“ bereits eine Domain eingetragen wurde. Dies ist der Fall da die Domain einfach vererbt

wird. Auf Seite 43 wird daraufhin beschrieben wie bei einer Data Property der Datentyp festgelegt wird. Dazu wird über den „+“-Button bei Range in dem sich öffnenden Fenster zu dem Tag „Built in datatypes“ navigiert und der gewünschte Datentyp ausgewählt. Dies wird grafisch zum besseren Verständnis auf Seite 44 dargestellt. Anschließend wird der Lernende erneut gebeten sein gelerntes Wissen zu festigen indem er bei den Data Properties für die Patientennummer einen integer als Datentyp und für den Vor- und den Nachname einen String als Datentyp festlegt. [20] [21]

Zum Abschluss der Präsentation wird dem Lernenden auf Seite 47 eine kurze Aufgabe gegeben bei der er sein Wissen über Klassen, Object Properties und Data Properties anwenden kann, um den Stoff der Präsentation noch ein weiteres Mal zu festigen. Zusätzlich soll diese Aufgabe als Grundlage für die Präsentation *Protégé: Vertiefung*, die im folgenden Kapitel behandelt wird, dienen.

### 4.3 Protégé: Vertiefung

Diese letzte Präsentation über Protégé behandelt die weiteren und tieferen Funktionen und Möglichkeiten die Protégé bietet. Dabei wird bei dieser Präsentation das Wissen aus der Präsentation *Protégé: Kernkonzepte* vorausgesetzt. Die Präsentation befindet sich in *Anhang D: Protégé: Vertiefung*.

Weiterhin soll an dieser Stelle angemerkt werden, dass das hier verwendete Beispiel, welches auch die letzte Aufgabe der vorherigen Präsentation war und mit dessen Hilfe die weiteren Methoden von Protégé gezeigt werden, auf dem englischsprachigen Beispiel von Antonis Bikakis, einem am University College London unterrichtenden Dozenten, basiert.

Auch in dieser Präsentation wird wieder mit einem Inhaltsverzeichnis zum Überblick begonnen, dabei lässt sich hier sehen, dass deutlich mehr Themen und Möglichkeiten von Protégé angesprochen werden. Es werden die meist benötigten und wahrscheinlich wichtigsten Thematiken angesprochen, denn alle Möglichkeiten von Protégé zu erläutern würde wohl jeglichen zeitlichen Rahmen sprengen.

Ab Seite 3 wird auf die Property Characteristics eingegangen. Diese sind Eigenschaften die sich den Object Properties und begrenzt auch den Data Properties geben lassen. Diese Characteristics können für die Properties ausgewählt werden, indem bei der gewählten Property im Fenster „Characteristics“ die gewünschte Characteristic ausgewählt wird. Dies wird auf Seite 4 mit einer Grafik dargestellt. Für die Object Properties existieren die Characteristics Functional, Inverse Functional, Transitive, Symmetric,

Asymmetric, Reflexive und Irreflexive während für Data Properties nur die Characteristic Functional zur Verfügung steht. Im Folgenden werden diese Characteristics nun genauer beschrieben. Zum Einen, um die gewünschte Vollständigkeit zu erreichen, zum Anderen wird davon ausgegangen, dass ein Teil der Studenten nicht das mathematischer Grundwissen besitzt, welches diese Funktionalitäten erklären würde. In der Präsentation finden sich diese auf den Seiten 5 bis 12:

- Functional: Die Eigenschaft Functional bedeutet, dass die Property für jedes Individuum höchstes einen ausgehenden Wert haben kann.
- Inverse Functional: Die Eigenschaft Inverse Functional bedeutet, dass die Property für jedes Individuum höchstens einen eingehenden Wert haben kann.
- Transitive: Die Eigenschaft Transitive bedeutet, dass, wenn ein Individuum X mit einem Individuum Y mit einer Property in Verbindung steht und das Individuum Y mit einem Individuum Z mit derselben Property in Verbindung steht, das Individuum X auch mit dem Individuum Z über diese Property in Verbindung steht.
- Symmetric: Die Eigenschaft Symmetric bedeutet, dass, wenn ein Individuum X mit dem Individuum Y durch eine Property in Verbindung steht, das Individuum Y auch mit dem Individuum X durch dieselbe Property in Verbindung stehen muss.
- Asymmetric: Die Eigenschaft Asymmetric bedeutet, dass, wenn ein Individuum X mit dem Individuum Y durch eine Property in Verbindung steht, das Individuum Y nicht mit dem Individuum X über die gleiche Property in Verbindung steht.
- Reflexive: Die Eigenschaft Reflexive führt dazu, dass jedes einzelne Individuum mit Hilfe dieser Property mit sich selbst in Verbindung gesetzt wird.
- Irreflexive: Die Eigenschaft Irreflexive bedeutet, dass ein Individuum nicht über diese Property mit sich selbst in Verbindung stehen kann. [Übersetzt vom Autor aus [22]] [23]

#### 4.3.1 Property Chains

Darauffolgend werden ab Seite 13 Property Chains beschrieben. Property Chains werden verwendet um eine Property durch bereits bestehenden Properties zu beschreiben. Um eine Property zu erstellen muss bei der ausgewählten Property auf den „+“-Button bei „SuperProperty Of(Chain)“ geklickt werden, daraufhin können die einzelnen Properties mit dem o-Operator verknüpft werden. Das wird auf Seite 15 mit folgendem Beispiel verdeutlicht:

Die Property „hatGroßvater“ lässt sich durch die existierende Property „hatVater“ beschreiben indem sie hintereinander mit sich selbst verkettet wird. Die Property-Chain für "hatGroßvater" würde also wie folgt geschrieben: hatVater o hatVater. [24]

Um den Lernenden zu ermöglichen, den Stoff über Property Chains zu festigen, folgt auf Seite 16 die Aufgabe, eine Property „hatNachbarlaender“ aus den bereits bestehenden Properties zu erstellen. Die dazu befindliche Lösung folgt auf Seite 17 mit einer Grafik.

### 4.3.2 Class Expressions

Die Class Expressions werden in der Präsentation ab Seite 19 behandelt und können dazu genutzt werden, um Individuen mit gemeinsamen Merkmalen zu beschreiben. Dazu können die Klassennamen zusammen mit den Class Expressions selbst verwendet werden. Dabei können Class Expressions beliebig geschachtelt werden, um entsprechend gewünschte Aussagen treffen zu können. Die meist verwendeten Class Expressions sind folgende sich selbsterklärende Operatoren:

- and
- or
- not

Class Expressions können erstellt werden, indem bei der gewünschten Klasse auf den „+“-Button bei „SubClass Of“ geklickt wird und dann zu dem Tab „Class Expressions“ navigiert wird. [25] Dies wird auf Seite 20 der Präsentation verdeutlicht. Daraufhin gibt es für die Lernenden zwei einfache Aufgaben um das Verwenden von Class Expressions anzuwenden. Die beiden Aufgaben lauten wie folgt:

1. Erstellen Sie die Subklassen „EuropaeischesLand“ und „NichtEuropaeischesLand“ und verwenden Sie eine Class Expression um „NichtEuropaeischesLand“ zu definieren.
2. Erstellen Sie die Klasse „Gegend“ und beschreiben Sie diese durch die Klassen „Land“ oder „Staat“.

Die Lösung für die erste Aufgabe findet sich in der Präsentation in Form einer Grafik auf der folgenden Seite 23. Dabei ist zu sehen, dass das „NichtEuropaeischesLand“ mit Hilfe der Class Expression „not“ und der Klasse „EuropaeischesLand“ definiert wurde. Für Aufgabe 2 müssen die zwei genannten Klassen „Land“ und „Staat“ zusammen mit der Class Expression „or“ verwendet werden um die Klasse „Gegend“ wie gewünscht zu definieren.



### 4.3.3 Object Restriction

Mit den Object Restrictions wird auf Seite 24 begonnen. Object Restrictions werden ebenso wie Class Expressions genutzt, um Individuen genauer zu beschreiben, während die Object Restrictions aber nicht nur Klassen, sondern auch Properties und den gewünschten RestrictionType verwenden. Die folgenden RestrictionTypes können in Protégé verwendet werden:

- some  
Die „some“-Restriction funktioniert wie eine Art Filter. Sie legt fest, dass, wenn eine Verbindung zu etwas besteht, muss dieses vom entsprechend gewählten Typ sein. Bsp.: *hasPet some Dog* legt fest, dass wenn etwas ein Haustier hat muss dieses ein Hund sein.
- value  
Diese Restriction funktioniert wie die „some“-Restriction, jedoch wird zusätzlich ein spezifisches Individuum festgelegt.
- only  
„only“ ist eine universelle Restriction die festlegt, mit was etwas in Verbindung steht.
- min  
Mit der „min“-Restriction wird festgelegt, dass mindestens entsprechend viele Kardinalitäten gegeben sein müssen. Bsp.: *hasPet min 3 Cat* beschreibt, dass eine Instanz mindestens drei Katzen als Haustiere haben muss.
- max  
Die „max“-Restriction beschreibt wie viele Kardinalitäten höchstens gegeben sein dürfen.
- Exactly  
Besagt, dass genau die gewählte Anzahl an Kardinalitäten gegeben sein muss.

Die Erstellung einer Object Restriction erfolgt durch das Öffnen des Object Restriction Creators. Dazu wird zuerst bei der gewünschten Klasse auf den „+“-Button bei „SubClass Of“ geklickt und dann zu dem Tab „Object Restriction Creator“ navigiert. [25] Veranschaulicht ist dies auf Seite 27. Alternativ kann man die Object Restrictions auch unter den Class Expressions erstellen, jedoch muss dort alles benötigte von Hand eingetippt werden, wohingegen im Object Restriction Creator ein Userinterface zur Auswahl verwendet werden kann. Das Verwenden einer Object Restriction soll an folgendem Beispiel verdeutlicht werden:

„Es soll die Klasse „MittleuropaeischesLand“ durch bereits bestehende Klassen und Properties beschrieben werden. Ein „MittleuropaeischesLand“ ist ein eu. Land, dass nur an eu. Länder angrenzt.“

Daraus lässt sich schließen, dass die Klasse „EuropaeischesLand“ und die Property „grenzt\_an“ verwendet werden sollen. Zusätzlich muss noch entschieden werden, welcher RestrictionType verwendet werden muss. Da ein mitteleuropäisches Land nur an andere europäische Länder angrenzt kommt als RestrictionType nur „only“ in Betracht. Es wird im Object Restriction Creator, wie auf Seite 30 dargestellt, die Property „grenzt\_an“, die Klasse „EuropaeischesLand“ und der RestrictionType „only“ ausgewählt. Die Auswahl der Klasse, der Property und des RestrictionTypes wird auf Seite 30 grafisch veranschaulicht.

Daraufhin folgt eine Aufgabe für die Lernenden, bei der sie die Klasse „Europaesiches-Nachbarland“ erstellen und mit Hilfe der Object Restriction definieren sollen. Die Lösung dazu befindet sich in Form einer Grafik auf der Folgeseite.

#### 4.3.4 Reasoner

Im nächsten Abschnitt der Präsentation, ab Seite 33, wird auf die Reasoner eingegangen. Reasoner sind Programme zur Überprüfung von logischen Konsequenzen aus Behauptungen oder Axiomen. Protégé verwendet Reasoner um die erstellte Ontologie auf ihre Konsistenz zu überprüfen. Reasoner sind im Funktionsumfang einer Standardinstallation von Protégé enthalten. Die bekanntesten OpenSource Reasoner, die in Protégé als Plugin verfügbar sind, sind Pellet und Hermit, welche beide in Java implementiert sind sowie das in C++ implementierte FaCT++. Um in Protégé einen Reasoner zu verwenden, muss der gewünschte Reasoner zuerst im entsprechenden Tab ausgewählt werden, grafisch auf Seite 36 in der Präsentation dargestellt. Darauffolgend wird er im selben Tab mit dem Befehl „Start reasoner“ gestartet werden. Der Reasoner kann an selbiger Stelle mit dem Befehl „Stop reasoner“ wieder angehalten werden. Wenn ein Reasoner in Protégé gestartet wurde, wird dies in der unteren rechten Ecke des Protégé-Fensters in der Form des Schriftzuges „active“ angezeigt. Sollte es eine Interferenz in der erstellen Ontologie geben, so wird der Reasoner einschreiten und ähnlich einem Debugging anzeigen an welcher Stelle die Interferenz auftritt. [26] [27]

### 4.3.5 Visualisierung

In der folgenden Behandlung von Visualisierungen, in der Präsentation zu finden ab Seite 38, werden auf die Visualisierungsplugins OntoGraf und die Visualisierungsprogramme „Graphical Ontology Editor“ (OWLGrEd) und „Visual Notation for OWL Ontologies“ (VOWL) als Webversionen eingegangen. Visualisierungen werden dafür genutzt die in Protégé erstellen Ontologien graphisch darzustellen, wodurch diese dann leichter zu überschauen und zu überprüfen sind. Dies liegt auf der einen Seite im Interesse der Lehrenden, da darauf aufmerksam gemacht werden kann, dass auf diese Weise die Ontologie gut veranschaulicht ist und es leichter verständlich ist wo Veränderungen oder Erweiterungen möglich sind. Auf der anderen Seite liegt es im Interesse der Lernenden da sie durch die Visualisierung abstrakter sehen können, wie die erstellte Ontologie aufgebaut ist. So ist diese leichter zu verstehen ist. Es werden verschiedene Visualisierungen vorgestellt, da jeder eine eigene Präferenz für die Darstellung von Ontologien oder Diagrammen hat.

Zu Beginn soll das Plugin OntoGraf näher betrachtet werden. Dieses Plugin sollte standardmäßig in Protégé installiert sein und kann über den Tab „Window“ über „Views“ und anschließend über „Class views“ aufgerufen werden. Dies wird auf Seite 40 grafisch dargestellt. Daraufhin öffnet sich ein Fenster in Protégé in welchem die aktuelle Ontologie in Form eines Baumes dargestellt wird. Über die verschiedenen Optionen innerhalb dieses Fensters kann der Graph verschieden angeordnet werden und es können verschiedene Ebenen ausblenden werden. Das in dieser Präsentation verwendete Beispiel ist als Visualisierung mit OntoGraf auf Seite 41 dargestellt.

Die Webversion von OWLGrEd kann über [http://owlgred.lumii.lv/online\\_visualization](http://owlgred.lumii.lv/online_visualization) aufgerufen werden. Hier muss dann die abgespeicherte Ontologie hochgeladen werden, um sie graphisch direkt im Browser betrachten zu können. Die Darstellung von OWLGrEd ähnelt einer UML-Darstellung. Wie auch bei OntoGraf ist das verwendete Beispiel mit der Visualisierung von OWLGrEd in der Präsentation abgebildet und kann auf Seite 43 gefunden werden.

Eine andere Art der Visualisierung bietet VOWL, wobei auch hier auf die Browserversion verwiesen wird die zu finden ist unter <http://www.visualdataweb.de/webvowl/>. Wie oben muss hier die Ontologie, um sie als Visualisierung zu betrachten, hochgeladen werden. Außerdem können wieder diverse Filter und Einstellungen verwendet werden um gezielt Bereiche der Ontologie zu betrachten.

### 4.3.6 Exportierung als Java-Code

Zum Abschluss der Präsentation wird noch ab Seite 45 darauf eingegangen, wie die erstellte Ontologie als Java-Code exportiert werden kann. Da das Erstellen der Ontologie nur einen Schritt in der Softwareentwicklung von entscheidungsunterstützenden Systemen darstellt, kann dies sehr hilfreich sein, um die gewünschten Klassen zumindest in Grundzügen bereits zu erstellen. Um dies in Protégé auszuführen, muss die Option „Generate Java code...“ und der Tab „Tools“ aufgerufen werden. Dies ist in der Präsentation auf Seite 46 beispielhaft zu sehen. Darauffolgend muss man Protégé den gewünschten Namen des Projektes geben und den Zielort zum Abspeichern festlegen. Protégé generiert dann den gewünschten Java-Code und legt diesen im Ziel Ordner als Java-Dateien ab. Auf der letzten Seite der Präsentation wird dargestellt wie die Java-Dateien abgelegt sind. [28] [29]

## 5 Umsetzung der Konzeptziele für ProKEt

In den folgenden Abschnitten wird die Erstellung der einzelnen Präsentationen über ProKEt unter den Kriterien der Konzeptziele beschrieben. Sie wird sowohl von der Seite der Lernenden als auch von der Seite der Lehrenden beleuchtet. Die Präsentationen finden sich im Anhang abgedruckt und auf dem dieser Arbeit beiliegendem Datenträger als PowerPoint-Präsentation wieder. Es wurde entschieden, dass zuerst auf die Funktionen der Spalten eingegangen wird und erst im Anschluss auf die Möglichkeiten diverser Typen zur Zeilenverwendung. Diese Entscheidung basiert darauf, dass auf diese Weise ProKEt so praxisorientiert wie möglich vorgestellt werden kann. Außerdem kann so das volle Potential genutzt werden, da die Funktionen der Typen auf diese Weise am nachvollziehbarsten beschrieben werden können.

### 5.1 ProKEt: Allgemeines

In dieser ersten Präsentation über ProKEt geht es darum den Lernenden einen Überblick zu verschaffen und ihnen zu veranschaulichen, wozu ProKEt verwendet wird. Im Anschluss wird schrittweise beschrieben wie ProKEt verwendet werden kann. Die Präsentation befindet sich in *Anhang E: ProKEt: Allgemeines*.

Zu Beginn soll den Lernenden mit Hilfe des Inhaltsverzeichnisses ein Überblick dieser Präsentation vermittelt werden. In der Verwendung auf Seite 3 der Präsentation wird kurz und knapp erklärt, dass ProKEt ein Programm zur Verwendung von Wissensbasen in tabellarischer Form ist, welche verwendet wird, um Lösungen herleiten zu können. Hiermit soll bereits alles kurz und knapp über die Verwendung im Allgemeinen zu ProKEt gesagt sein.

Darauf folgt die Vorbereitung zur Verwendung von ProKEt ab Präsentationsseite 4. Dabei muss an erster Stelle darauf aufmerksam gemacht werden, dass nicht mit ProKEt selbst, sondern mit dem Programm KnowOF gearbeitet wird, was jedoch auf dem Kern von ProKEt arbeitet. Des Weiteren wird hier der Dank an die Uni Würzburg, insbesondere an Frank Puppe, ausgerichtet, durch die der Zugang zu KnowOF ermöglicht wurde. Auf Seite 5 wird erklärt, dass ProKEt/KnowOF Wissensbasen in Form des Excelformats .xlsx verwendet und dass zur Erstellung dieser Wissensbasen daher ein Programm benötigt wird, welches dieses Speicherformat ermöglicht. Bekannte Programme die in der Lage sind Tabellen als .xlsx-Format zu speichern sind:

- Microsoft Office Excel
- Open Office
- Numbers for Mac

- Google Sheets

Zu erwähnen ist an dieser Stelle, dass Google Sheets webbasiert arbeitet. Folgend wird auf Seite 6 erklärt, dass eine aktive Internetverbindung sowie ein Browser benötigt werden, um KnowOF zu verwenden. Das liegt daran, dass KnowOF eine webbasierte Anwendung ist. Das Vorstellen von .xlsx-fähigen Programmen und den getesteten Browsern soll die problemlose Anwendung von ProKEt gewährleisten und dient an dieser Stelle der Vervollständigung. Weiter werden an dieser Stelle gängige kompatible Browser wie folgend genannt:

- Google Chrome
- Mozilla Firefox
- Internet Explorer / Edge
- Opera
- Safari

Auf den folgenden Seiten der Präsentation werden die Links dargestellt, über welche man KnowOF als auch die dazugehörige Dokumentation aufrufen kann. Um KnowOF aufzurufen wird folgender Link verwendet:

[http://winc031.informatik.uni-wuerzburg.de:8080/testing\\_master/knowOfStatic/](http://winc031.informatik.uni-wuerzburg.de:8080/testing_master/knowOfStatic/)

Um die Dokumentation von KnowOF aufzurufen wird folgender Link verwendet:

<http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/Einfuehrung>

Zusätzlich wird auf Seite 9 der Präsentation in Form eines Bildes die Startseite nach Aufrufen des Links für KnowOF gezeigt.

Der letzte Abschnitt dieser ersten Präsentation von ProKEt bzw. KnowOF beschreibt, wie Wissensbasen hochgeladen und ausgeführt werden können. Dies geschieht Schritt für Schritt, um es den Lernenden so nachvollziehbar wie nur möglich zu gestalten und wird in Form von Abbildungen über den Vorgang nochmals verdeutlicht. Zu Beginn muss auf der Startseite von KnowOF im Bereich „Wissensbasis hochladen“ unter „Wissensbasis wählen“ mit Hilfe des Buttons „Durchsuchen...“ die gewünschte Wissensbasis im .xlsx-Format ausgewählt werden. Nach Auswahl der gewünschten Wissensbasis kann die Datei über den „Hochladen“-Button schließlich auf den Webserver hochgeladen werden. Die Auswahl der Wissensbasis und des Hochladens wird auf Seite 11 mit Hilfe eine Abbildung verdeutlicht. Sollte die Wissensbasis in korrekter Syntax sein, zeigt KnowOF nach dem hochladen die Meldung „Upload Erfolgreich“ in einem grünen Feld an. Sollte die Syntax der Datei fehlerhaft sein, so wird in einem gelblichen Feld die Meldung „Verifikation ist fehlgeschlagen. Genaueres dazu im Bericht“ angezeigt. Dies wurde erneut durch

eine Abbildung auf Seite 13 verdeutlicht, um möglichst praxisorientiert und nachvollziehbar ProKEt vorzustellen. Sollte ein Fehler auftreten, so kann über den Button „Bericht ansehen“ ein Fehlerbericht angezeigt werden in dem auf die genauen Syntaxfehler eingegangen wird. Dies wurde zum Abschluss dieser Präsentation auf Seite 14 veranschaulicht. An dieser Stelle wird nicht weiter auf die Fehlerbehandlung eingegangen, da im Verlauf der folgenden Präsentationen über ProKEt ausreichend Wissen vermittelt werden sollte, um den Fehlerbericht anwendungsbezogen eigenständig auszuwerten und nutzen zu können.

## 5.2 ProKEt: Spalten

Diese Präsentation behandelt die Funktion der Spalten in ProKEt/KnowOF. Diese Präsentation enthält nur wenige Übungsaufgaben auf Grund der Tatsache, dass ProKEt auf Tabelleneingaben arbeitet. Dazu müssen sowohl die Spalten als auch die verschiedenen Typen bekannt sein. Es wurde entschieden zunächst nur die Spalten einzuführen, um danach die problemlose Verwendung der Typen zu ermöglichen. In diesem Kapitel werden nur die Typen kurz beschrieben, die für das Verständnis der Funktion der Spalten benötigt werden. Die Präsentation hierzu befindet sich in *Anhang F: ProKEt: Spalten*.

Zu Beginn wird im Inhaltsverzeichnis darauf aufmerksam gemacht, dass es grundsätzlich 2 Arten von Spalten gibt: die Standarttypen, die immer benötigt werden, und die Regeltypen. Direkt im Anschluss wird auf Seite 3 der Präsentation erklärt, dass das verwendete Beispiel in dieser Präsentation frei erfunden ist und nur dem Zweck der Veranschaulichung dient. Dennoch ist das Beispiel so gewählt, dass es den Praxisbezug darstellt.

### 5.2.1 Standarttypen

Im Folgenden werden nun die verschiedenen Standarttypen der Spalten anhand eines Beispiels erklärt. Die Standarttypen der Spalten beschreiben, was in welcher Spalte der Tabelle eingetragen werden darf, damit diese vom System korrekt verwendet werden kann. Bei falscher Verwendung kommt es zu einem Syntaxfehler und die Wissensbasis kann, wie in Kapitel 3.3.1 *ProKEt: Allgemein* gezeigt, nicht erfolgreich hochgeladen und verwendet werden. Des Weiteren wird an dieser Stelle der Präsentation beschrieben, dass die Reihenfolge der Spalten in ProKEt beliebig gewählt werden kann. Es wird aus Gründen der Übersicht aber die Reihenfolge des Beispiels empfohlen, welches auf Seite 6 abgebildet ist.

Die Spalte „Typ“ ist die wichtigste Spalte bei der Benutzung von ProKEt/KnowOF. In dieser Spalte wird durch den Zeilentyp definiert, wie die entsprechende Zeile auszulesen und zu behandeln ist. [30] Dies wird anhand des vorherig gezeigten Beispiels verdeutlicht. Dabei wird in Zeile 3 des Beispiels mit dem Zeilentyp „U“ eine Untersuchung eingeleitet, während in Zeile 6 mit dem Zeilentyp „A“ eine mögliche Antwort für die zuvor erstellte Frage beschrieben wird. An dieser Stelle wird erneut erwähnt, dass weitere Typen erst in der Präsentation *ProKEt: Typen* beschrieben werden.

In der „Fragen-Antworten“-Spalte werden alle möglichen Fragen und Antworten aufgelistet. In dieser Spalte steht außerdem der größte Teil des Textes, welcher dem Benutzer später angezeigt wird. [31] Auf Seite 10 wird dies durch eine Abbildung verdeutlicht, in der ein Teil des obigen gezeigten Beispiels aus Sicht eines Benutzers gezeigt wird.

Die „ID“-Spalte dient dazu, einer Frage oder einer Antwort eine beliebige ID zuzuweisen, um diese im späteren Verlauf leichter referenzieren zu können. Dies wird dazu verwendet, um gezielte Inhalte erneut wiederverwenden zu können. [32] Auf Seite 12 wird in einer Abbildung gezeigt wie der Frage „Spielen Sie Computerspiele?“ die ID „CS“ zugewiesen wird.

Eine Spalte des Typs „Wenn“ wird verwendet, um Bedingungen benutzen zu können. Dabei wird die Frage oder Untersuchung der Zeile, in der eine Bedingung in der „Wenn“-Spalte steht, dem Benutzer nur angezeigt, wenn die Bedingung erfüllt ist. [33] Auf Seite 14 der Präsentation wird beschrieben, dass die Frage des Zeilentyps „One-Choice“ nur angezeigt wird, wenn die Frage mit der ID „CS“ mit „Ja“ beantwortet wurde. Dies wird zusätzlich durch die Abbildung der Tabelle mit entsprechend markierten Bereichen veranschaulicht, während auf der Folgeseite eine Abbildung deutlich macht, wie sich dies auf die Benutzeranzeige auswirkt.

Mit der „Aktion“-Spalte kann festgelegt werden, welche Untersuchungen des Zeilentyps „U“ direkt angezeigt werden, ohne dass dafür Bedingungen erfüllt sein müssen. Um dies zu ermöglichen, wird in die Aktionsspalte eine Zahl, beginnend von 1, eingetragen. Dabei ist jedoch die Reihenfolge der Nummerierung irrelevant, da die Reihenfolge innerhalb der Tabelle ausschlaggebend für die Reihenfolge der Benutzeranzeige ist. [34] Durch ein Beispiel auf Seite 17 soll dies verdeutlicht werden. Dabei wird sowohl der entsprechende Tabellenausschnitt, als auch die Ansicht des Benutzers gezeigt. Eine weitere wichtige Funktion der „Aktion“-Spalte besteht darin, dass sie genutzt wird um gezielte Aktionen durch RF-Regeln auszuführen. Darauf wird jedoch im Kapitel 5.3.3 *Sonstige Typen* genauer eingegangen. Um den Lernenden zur ermöglichen, ihr bisheriges Wissen in Form einer Übungsaufgabe zu verwenden, soll eine neue Untersuchung des Typs „U“ mit der



Bezeichnung „Drogenabhängigkeit“ erstellt werden. Dazu soll eine Ja-Nein-Frage „Nehmen Sie Drogen?“ erstellt werden und der Untersuchung soll eine Aktionsnummer zugewiesen werden, um die Untersuchung anzeigen zu lassen. Die Lösung ist in Form der Sicht des Benutzers auf Seite 19 angezeigt.

Abschließend zu den Standarttypen der Spalten werden in den folgenden Abschnitten die Möglichkeiten der „Layout“-Spalte und einige der dazu verwendeten Zeilentypen beschrieben. Layouttypen werden verwendet, um die Anzeige zu bestimmten Abschnitten anzupassen. Dabei wird zwischen einzelnen Layouttypen, die nur die Anzeige einer einzelnen Zeile anpassen und den D-Layouttypen („D“ für Dialog) unterschieden, deren Eingaben für den gesamten Benutzerdialog gültig sind. [35] Dies wird in der Praxis dazu verwendet die Befragungen möglichst optimal zuzuschneiden, um für den Benutzer ein optimales Userinterface bieten zu können.

Zu Beginn wird hier der Layouttyp „dropdown“ für den Zeilentyp „OC“ (One-Choice) dargestellt, bei dem die Anzeige der Antwortmöglichkeiten durch ein Dropdownmenü ersetzt wird, indem in der „Layout“-Spalte der gewünschten „OC“-Zeile der Befehl „dropdown“ eingetragen wird. [36] Um dies zu veranschaulichen wird auf Seite 22 sowohl die tabellarische Ansicht als auch die Ansicht des Benutzers abgebildet.

Auf Seite 23 der Präsentation wird erläutert, dass die D-Layouttypen für den gesamten Dialog gültig sind und als Block am Ende einer Wissensbasis aufgelistet werden. Dabei werden aber nur die Layout-Typen „D-Header“, „D-SP“ und „D-SSP“ betrachtet, da diese zu den wichtigsten D-Layouttypen gehören und die weiteren Typen für die grundlegende Verwendung von ProKET nicht von Bedeutung sind.

Der Typ „D-Header“ ermöglicht es, dem gesamten Dialog eine Überschrift zu geben. Dazu wird in der „Typ“-Spalte „D-Header“ eingetragen und in der Layout-Spalte die gewünschte Überschrift festgelegt. [37] Dies wird zusammen mit den anderen Layout-Typen auf den Seiten 27 und 28 sowohl in Tabellenform als auch in Benutzersicht durch Abbildungen verdeutlicht.

Mit Hilfe des Layout-Typs „D-SP“ wird festgelegt, wie viele Spalten das Layout verwenden soll, wenn ein Dialog mit Spalten genutzt wird. Dabei sind die Werte 1 bis 4 möglich. [38]

Das Verwenden des Layout-Typs „D-SSP“ ermöglicht es das „Solution Panel“ anzeigen zu lassen in welchem hergeleitete Diagnosen und Lösungen dem Benutzer angezeigt werden. Um das Solution Panel anzeigen zu lassen, muss in der „Typ“-Spalte „D-SPP“ gesetzt und in der „Layout“-Spalte ein „x“ geschrieben werden. [39]

## 5.2.2 Regeltypen

Im Weiteren sollen nun die Regeln genauer betrachtet werden. Die Möglichkeit Regeln zu verwenden und zu erstellen, bildet eines der Kernkonzepte von ProKEt selbst. Durch das Verwenden von Regeln können Lösungen und Diagnosen aus gegebenen Antworten des Benutzers hergeleitet werden. Dabei sind die Regeln selbst für den Benutzer der graphischen Oberfläche nicht sichtbar. Lediglich die Ergebnisse der Regeln werden im bereits erwähnten „Solution Panel“ angezeigt. Diese Regeln stellen zusammen mit den Typen der Zeilen den Kern des regelbasierten Wissensmanagements dar und sind dementsprechend für die praktische Anwendung unverzichtbar.

„RH“-Spalten werden verwendet um Lösungen durch eine Reihe von Und-Verknüpfungen von Antworten mehrerer Fragen herzuleiten. Dazu werden in der RH-Spalte zu entsprechenden Fragen die zu erfüllenden Antwortmöglichkeiten angegeben. Nur wenn der Nutzer alle Fragen entsprechend der RH-Spalte beantwortet hat wird die Lösung als anerkannt betrachtet und im Solution-Panel angezeigt. In der „LZeile“ der RH-Regel steht der Name der Lösung während in der „LB“-Zeile der Wert der Lösung zugewiesen wird. Die Typen „LZeile“ und „LB“ werden zu späterem Zeitpunkt genauer erklärt, siehe Kapitel 5.3.3 *Sonstige Typen*. [40] Um die RH-Spalte zu veranschaulichen, wurde auf den folgenden Seiten 32 und 33 das Beispiel aus der Dokumentation von KnowOF verwendet um sowohl die tabellarische Form als auch die Benutzeroberfläche entsprechend nachvollziehbar vorzustellen.

In den „H“-Spalten (H steht an dieser Stelle für heuristisch) werden einer Lösung Punkte in Abhängigkeit der vom Benutzer gegebenen Antworten gegeben. Dabei wird bei ausreichend positiven Punkten die Lösung als bestätigt betrachtet, während bei genügend negativen Punkten die Lösung als widerlegt betrachtet wird. Auch hier wird in der „LZeile“ im Tabellenfeld der „H“-Spalte der Name der Lösung vergeben und in den Felder der Antwort-Zeilen werden in der „H“-Spalte entsprechende Punkte vergeben. [41] Dies wurde durch ein tabellarisches Beispiel auf Seite 37 veranschaulicht, bei dem sichtbar wird, dass die Punktevergabe für die Lösung je nach gewählter Antwort unterschiedlich ausfallen kann. Das verwendete Bewertungssystem wird im Folgekapitel 3.3.3 ProKEt-Typen beschrieben.

Abschließend in dieser Präsentation werden die „HN“-Spalten beschrieben. Diese Spalten funktionieren wie die „H“-Spalten mit dem Unterschied, dass Punkte bei einer „Ja-Nein-Frage“ mit der Antwort „Nein“ oder einer nicht gegebenen Antwort einer Multiple-Choice- bzw. einer One-Choice-Frage trotzdem für die Herleitung der Lösung herangezogen werden. [42]

## 5.3 ProKEt: Typen

Diese letzte Präsentation über ProKEt behandelt Funktionen der verschiedenen Zeilen-Typen. Dabei wird bei dieser Präsentation das Wissen aus der Präsentation *ProKEt: Spalten* vorausgesetzt. Die Präsentation befindet sich in *Anhang G: ProKEt: Typen*.

Durch das Inhaltsverzeichnis soll die Präsentation besser strukturiert werden und es den Lehrenden ermöglicht werden sich besser innerhalb der Präsentation zurechtzufinden. Dabei wird im Verlauf der Präsentation erst im Allgemeinen auf die Typen eingegangen. Daraufhin werden die verschiedenen Typen dem Inhaltsverzeichnis entsprechend behandelt.

Typen sind in der Typ-Zeile festgelegt und stellen die wichtigste Eigenschaft in ProKEt dar, denn Typen geben an, wie die Zeile interpretiert werden muss.

### 5.3.1 Fragentypen

Die Fragen-Typen bilden den Kern der Wissensengabetabelle und müssen in der ganzen Wissensbasis eindeutig sein. Zusätzlich zu den im folgenden aufgeführten Frage-Typen können verschiedene Frage-Typen zusätzlich mit der Endung „-I“ ausgestattet sein, diese Fragen werden nicht vom Benutzer beantwortet, sondern durch bereits beantwortete Fragen von ProKEt selbst hergeleitet. [43] Die in ProKEt existierenden Typen sind:

- Date
- JN (Ja-Nein)
- JNU (Ja-Nein-Unbekannt)
- MC (Multiple Choice)
- Num (Numerisch)
- Num-I (Numerisch-Indirekt)
- OC (One-Choice)
- Text

In der Präsentation wird auf Seite 6 beschrieben, dass eine Zeile des „Date“-Typs ein Datum als Antwort erfordert. Die Oberfläche des Nutzers zeigt zur Antwortmöglichkeit automatisch ein Dropdownmenü an, sodass es keine Eingabefehler durch verschiedene Datumsformatierungen gibt. Zusätzlich kann in der Layout-Spalte der Layout-Befehl „YearOnly“ eingetragen werden, damit nur mit einer Jahreszahl geantwortet werden kann. [44] Dies wurde zum veranschaulichen auf Seite 7 durch eine Tabelle und durch eine Abbildung des Dropdownmenüs dargestellt.

Eine Frage des Typs „JN“ gibt dem Benutzer die Möglichkeit zwischen „Ja“ und „Nein“ zu wählen. Hierbei ist zu beachten, dass durch die verwendete Version von ProKEt trotzdem die Option „Unbekannt“ angezeigt wird. [45] Auch hier wird eine Tabelle abgebildet, um dies zu veranschaulichen. Auf die Abbildung des Benutzerinterfaces wurde hier verzichtet, da im späteren Verlauf genügend Beispiele auftreten, in denen die Ja-Nein-Frage erkennbar ist.

Der Fragen-Typ „JNU“ unterscheidet sich von dem Typ „JN“ nur in der Tatsache, dass noch die Dritte Option „unbekannt“ für den Benutzer zu Auswahl steht. [46] Wie bereits bei dem „JN“-Typ erwähnt, wird auch hier auf ein Beispiel aus dem obigen Grund verzichtet.

Durch den Fragen-Typ „MC“ wird eine Frage formuliert bei der durch den Benutzer eine, mehrere oder auch keine Antwort ausgewählt werden kann. ProKEt fordert jedoch, dass mindestens zwei Antworten des Antworttyps „A“ zur Verfügung stehen. [47] Erneut wurde in dies durch eine tabellarische Form auf Seite 11 abgebildet.

Fragen des Typs „Num“ erfordern als Antwort eine Zahl. Dazu wird dem Benutzer ein Feld angezeigt, in welches eine entsprechende Zahl eingegeben werden kann. In diesem Feld sind Kommazahlen erlaubt. Zusätzlich können nach der „Num“-Zeile noch folgende Elemente in genau dieser Reihenfolge genutzt werden, um die Eingabe besser zu definieren:

1. Einheit:

Hier kann in der „Fragen-Antworten“-Spalte die Einheit der Zahl festgelegt werden.

2. Min-Wert:

Definiert in der „Fragen-Antworten“-Spalte den minimalen zulässigen Wert.

3. Max-Wert:

Definiert in der „Fragen-Antworten“-Spalte den maximalen zulässigen Wert. [48]

Eine tabellarische Abbildung zur Veranschaulichung findet sich auf Seite 14 der Präsentation. Dies soll den Lernenden, dem Ziel der Verständlichkeit entsprechend, darstellen wie diese zusätzlichen Elemente genutzt werden.

Der „Num-I“-Typ verhält sich genau wie der „Num“-Typ, wird jedoch vom System berechnet und wird dem Benutzer in der verwendeten Version von ProKEt auch nicht angezeigt, sondern zur Herleitung von Lösungen verwendet. Die Berechnung dazu ergibt sich durch eine „RF“-Regel die im späteren Verlauf noch beschrieben wird. [49]

Der in der vorherigen Präsentation kurz angerissene Fragen-Typ „OC“ verhält sich wie der bereits genannte Typ „MC“, jedoch kann nur eine einzige Antwort ausgewählt werden. Hier kann, wie ebenfalls in der vorherigen Präsentation erwähnt, zusätzlich in der Layout-Spalte der Typ „dropdown“ eingetragen werden, um in der Benutzeroberfläche ein Dropdownmenü zur Auswahl zu verwenden. [50] Auf Seite 17 befindet sich eine Tabelle zur visuellen Darstellung.

Der verbleibende Fragen-Typ „Text“ erfordert vom Benutzer eine Eingabe in einem Textfeld. [51] Durch Testen wurde herausgefunden, dass die maximale Anzahl an Zeichen 4683 beträgt.

Sämtliche obige genannten Fragen-Typen wurden auf Seite 19 in Form der Benutzeroberfläche abgebildet. Daraufaufgehend wird in der Präsentation eine Aufgabe gestellt, bei der die Lernenden das bisher erlangte Wissen anwenden und testen können. Dabei sollten für die Untersuchung „Körperlicher Zustand“ folgende Fragen erstellt werden:

- Trinken Sie Alkohol? Wenn Ja, einmal pro Monat, pro Woche oder pro Tag.
- Taillenumfang in cm mit der Mindesteingabe 0.
- Haarfarbe, mit den Auswahlmöglichkeiten Blond, Braun, Rot oder Schwarz.

Auf Seite 21 folgt die Lösung dazu in tabellarischer Form und auf Seite 22 findet sich die dazugehörige Abbildung der Benutzeroberfläche.

### 5.3.2 Erklärungstypen

In diesem Abschnitt werden die verschiedenen Erklärungs-Typen behandelt. Erklärungstypen bieten die Möglichkeit verschiedene Elemente zur weiteren Erklärung von Fragen-, Antwort- und Untersuchungs-Typen zu verwenden. Dabei ist die Anzahl und Reihenfolge der Erklärungselemente beliebig, jedoch müssen sie gebündelt auf den entsprechenden Typ folgen, der genauer erklärt werden soll. [52] Es existieren in ProKEt folgende vier Erklärungs-Typen:

- EB (Erklärungsbild)
- EH (Erklärungs-HTML-Text)
- EL (Erklärungs-Link)
- ET (Erklärungstext)

Der Erklärungs-Typ „EB“ ermöglicht es, eine Bilddatei anzeigen zu lassen. Dazu wird in der Fragen-Antwort-Spalte der Dateiname des Bildes eingefügt. Das Bild muss in der Mediendatei als .zip-Datei hochgeladen werden. Um ein Bild in die Mediendatei hochzuladen muss dieses vor dem Hochladen der Wissensbasis ausgewählt werden. [53] Dies

geschieht wie das Auswählen der Wissensbasis in dem Bereich „Wissensbasis hochladen:“ durch verwenden des „Durchsuchen...“-Buttons bei „Mediendatei wählen:“. Auf Seite 27 wurde dies durch eine Abbildung zum besseren Verständnis demonstriert. Es gilt zu beachten, dass das Bild nicht skaliert, sondern in exakter Auflösung angezeigt wird.

Die Verwendung des Erklärungs-Typs „EH“ ermöglicht das Anzeigen eines durch HTML formatierten Textes, der zur genaueren Erklärung für den Benutzer sichtbar wird. Dabei wird die gesamte Eingabe im Feld der „Fragen-Antwort“-Spalte als HTML interpretiert. [54]

Mit Hilfe des Erklärungs-Typs „EL“ kann in der „Fragen-Antwort“-Spalte ein Link eingefügt werden, welcher dem Benutzer dann angezeigt wird. [55]

Durch den Erklärungs-Typ „ET“ kann, wie auch durch den Typ „EH“, ein Text angezeigt werden, um etwas zum besseren Verständnis für den Benutzer zu beschreiben. Jedoch ist die Eingabe im „Fragen-Antwort“-Feld im Gegensatz zu „EH“ nicht formatiert und wird als durchgehender Text angezeigt. [56] Daher wird empfohlen „ET“ nur für kurze einfache Erklärungen zu verwenden und für ausführlichere Erklärungen den Typ „EH“ zu verwenden, der dafür etwas mehr Aufwand fordert.

Auf Seite 31 und 32 wurden sämtliche Erklärungs-Typen durch ein Beispiel in tabellarischer als auch in abgebildeter Form der Benutzeroberfläche dargestellt, um den Lernenden anwendungsnah die Benutzung der Erklärungs-Typen zu zeigen.

### 5.3.3 Sonstige Typen

Zur Vervollständigung werden im Weiteren nun sämtliche sonstige Typen beschrieben, die zum aktuellen Zeitpunkt in ProKEt vorhanden sind. Teilweise wurden diese Typen bereits schon kurz angerissen. Folgende Typen werden noch weiter beschrieben:

- U (Untersuchung)
- A (Antwort)
- K bzw. Kom (Kommentar)
- LZeile (Lösungszeile)
- L (Lösung)
- LB (Lösungsbewertung)
- RF-Regeln

Der Typ „U“ dient dazu, den Fragebogen, den der Benutzer sieht, besser zu untergliedern. Dazu wird in der Aktionsspalte angegeben, ob die Untersuchung bereits zu Beginn

des Dialogs gezeigt wird oder erst im späteren Verlauf in Abhängigkeit von vorangegangenen Antworten.

Mit dem Typ „A“ wird eine Zeile als Antwort definiert und die dazugehörige Antwort in der „Fragen-Antwort“-Zeile eingetragen.

Durch die Verwendung des Typs „K“ oder „Kom“ wird die gesamte Zeile der Tabelle als Kommentar definiert und wird nicht von ProKEt interpretiert. Dies dient dazu in einer größeren Tabelle Abschnitte zu erstellen und dadurch die Tabelle übersichtlicher zu gestalten. [57] [58] Auf Seite 37 wurde das beispielhaft durch eine Abbildung dargestellt.

In der Zeile des Typs „LZeile“ werden Lösungen und Diagnosen angegeben. Die Bezeichnungen der Lösungen werden dann in die Regelspalten geschrieben. Veranschaulicht wird dies auf Seite 38. [59]

Der Typ „L“ ermöglicht es, existierende Lösungen aus der Zeile „LZeile“ zu referenzieren. Dabei wird die gewünschte Lösung in der Fragen-Antwort-Spalte gesetzt. Zu beachten ist, dass Zeilen des „L“-Typs erst nach sämtlichen Fragen- und Antwort-Typen in der Tabelle gelistet werden. [60] In tabellarischer Form wird dies auf Seite 40 verdeutlicht.

Ein weiterer sehr wichtiger Typ in ProKEt stellt der Typ „LB“ dar. Durch diese Zeile wird ein Bewertungswert für Lösungen und Diagnosen festgelegt. Die möglichen Bewertungen reichen, wie auf Seite 42 der Präsentation tabellarisch gezeigt, von p7, dem Wert 999 entsprechend, bis zu n7, dem Wert -999 entsprechend. Je höher der Wert ist, desto sicherer wird die Diagnose als anerkannt betrachtet. Bei einer positiven Bewertung von 80 (entspricht p6) oder höher wird eine Diagnose als bestätigt betrachtet, während eine negative Bewertung von -80 oder weniger eine Widerlegung der Diagnose zur Folge hat. [61] Auf den Seiten 44 und 45 wird dies erneut zum besseren Verständnis sowohl in tabellarischer Form, als auch in Form der Benutzeroberfläche gezeigt. Dabei ist zu bemerken, dass das Ergebnis einer Herleitung im Solution Panel zu sehen ist.

Abschließend zu ProKEt wird der Typ „RF“ betrachtet. „RF“-Typen lassen sich dazu benutzen selbst Regeln zu formulieren. Dazu wird die Bedingung in die Fragen-Antwort-Spalte geschrieben und die auszuführende Aktion wird in der „Aktion“-Spalte festgelegt. [62] Auf der letzten Seite der Präsentation wird, um den „RF“-Typ besser verständlich zu gestalten, ein Beispiel zur besseren Erklärung verwendet. Dieses Beispiel wurde aus der Dokumentation übernommen. [62] Dabei wird der BMI des Benutzers berechnet mit den Bedingungen, dass die verwendeten Werte nicht kleiner als Null sein dürfen. Es gilt zu beachten, dass das Ergebnis der Berechnung dem Benutzer nicht angezeigt wird, sondern nur innerhalb der Wissensbasis verwendet wird. Dies liegt erneut der verwendeten

Version KnowOFs zu Grunde und ließe sich über die Einstellungen des Webservers ändern, sodass die berechneten Ergebnisse auch dem Benutzer direkt angezeigt werden. Allerdings stand diese Möglichkeit stand zum Zeitpunkt der Fertigstellung nicht zur Verfügung.



## 6 Fazit

Um zu messen, ob die Erwartungen der Lehrenden und Lernenden erreicht wurden, sollte eine Evaluation stattfinden, nachdem die in dieser Arbeit erstellten Materialien in einer Vorlesung verwendet wurden. Jedoch ist diese empirische Feststellung der Erwartungen auf Grund der zeitlichen Gegebenheit nicht möglich. Dies liegt der Tatsache zu Grunde, dass zum Zeitpunkt der Fertigstellung der Materialien das Semester bereits abgeschlossen war.

Die Erwartungen der Lernenden wurden daraufhin experimentell verifiziert, indem freiwillige Probanden aus dem Bekanntenkreis des Autors die Materialien zum Nutzen vorgelegt wurden und daraufhin durch eine Umfrage zu ihren Erfahrungen befragt wurden. Die Umfrage wurde dabei größten Teils an aktive Studenten ausgegeben. Nicht alle demographischen Merkmale sind bekannt, da die Befragung in einer Art Schnellballsystem weitergeleitet wurde. Es wurden insgesamt 10 Personen befragt. Dabei wurden die Probanden zu jeder Präsentation einzeln gebeten eine Bewertung in den Kategorien „Verständlichkeit/Selbsterklärend“, „Anwendbarkeit/Praxisbezug“, „Vollständigkeit“ sowie „Gewählte Beispiele“ abzugeben. Zusätzlich gab es zu jeder Präsentation die Möglichkeit zusätzliche Bemerkungen zu verfassen. Die zu dieser Umfrage gehörenden Rohdaten befinden sich in *Anhang H: Rohdaten der Umfrage zur Erwartungsmessung*.

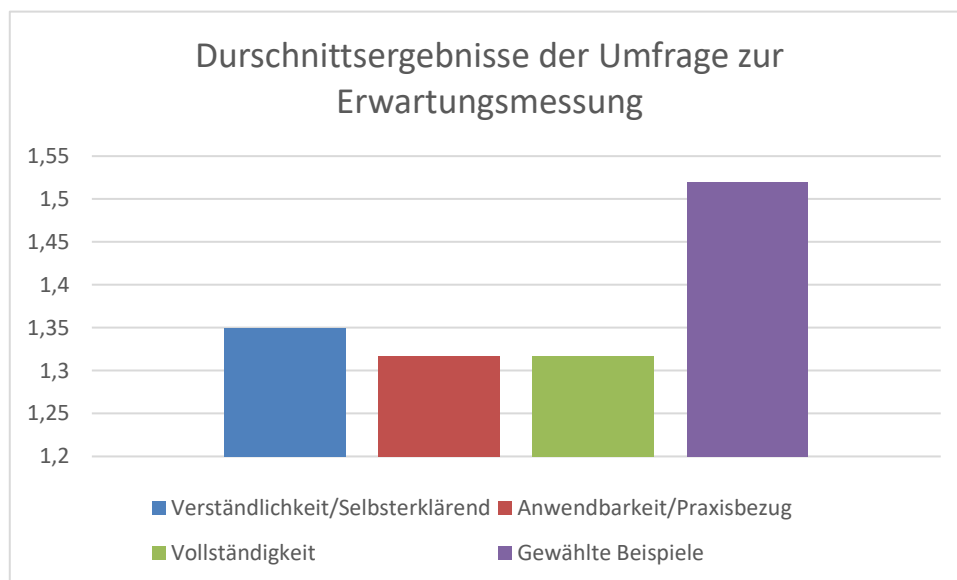


Abbildung 2: Durschnittsergebnisse der Umfrage zur Erwartungsmessung

Erkenntlich wird, dass sowohl die Verständlichkeit als auch Anwendbarkeit und Vollständigkeit überwiegend positiv zwischen „Gut“ (2) und „Sehr gut“ (1) bewertet wurden. Die

gewählten Beispiele fielen im Durchschnitt etwas schlechter aus, wurden dennoch durchschnittlich mit 1,52 bewertet.

Daraus wird deutlich, dass die erwarteten Ziele der Lernenden erfüllt sind und damit auch der Erfolg dieser Arbeit, bezüglich dieser Konzeptziele, bestätigt wird.

Die Erfüllung der Konzeptziele der Lehrenden war leider nicht messbar. Um diese zu bestätigen war geplant, das gelehrte Wissen in einer Klausur in Form von Anwendungsaufgaben zu prüfen. Auf Grund der bereits genannten zeitlichen Umstände war dies nicht möglich. Aber da die Ziele der Lehrenden sich mit denen der Lernenden überschneiden, kann man, aufgrund des Erfolgs beim Erreichen der Konzeptziele der Lernenden, davon ausgehen, dass auch die Konzeptziele des Lehrenden sich erfüllt und bestätigt hätten.

## Anhang A: Rohdaten der Umfrage zur Ermittlung der Konzeptziele der Lernenden

### Was ist Ihnen als Studierender an einer Vorlesung/Präsentation wichtig?

Bitte Stichwortartig antworten mit dem wichtigsten an oberster Stelle und dem am wenigsten wichtigen an unterster Stelle.

Sie sind der Zuhörer der Präsentation!

Praxisbezug; gute Visualisierung; Folien nachvollziehbar ohne Präsentation  
Selbsterklärend; Anwendbar/Praxisbezug; Vollständig

Interessant (-> Bezug zu Studenten herstellen) ; Anschaulich (-> Beispiele, Grafiken usw.) ; Verständlich (sowohl von Inhalt als auch vom Vortrag -> deutliche Aussprache)

Aufmerksamkeit zu bekommen,; Mein wissen weiter zu geben,; Vermitteln des Wissens zu können; Praxisnahe Anwendung

- Interesse weckend; - Hintergrundwissen des Dozenten; - Thematik in höheren Kontext setzen/ "gut Verpacken"; - Bedingungen müssen stimmen(räumlich, Rahmenbedingungen,...)

interessante Darstellung; visuelle Darstellung; lebhaftere Darstellung (statt "runterleiern")

Darbietung; Aufbau ; Gestaltung

Verständliches Reden ; Klare Zusammenhänge

-> Thema mit Bezug zur Realität; -> Nicht nur Theorie erklären -> praktischen Bezug und praktische Anwendung erklären; -> angenehmes Sprachtempo (zu schnell oder langsam -> man schaltet ab)

Kurzweiligkeit; Anschaulichkeit; Einbeziehung der Studenten

Sollte sein Informativ, gut erklärt, bei kompliziertem Thema Schaubilder oder Grafiken zum besseren Verständnis. Dozent muss Bock auf das Thema haben.

Viele Formeln

Bilder; praktische Beispiele; Wenig Text; Anschaulich ; Übersichtlich  
Informationen; Präsentationstechnik; Unterhaltsam

Logischer Aufbau der den Inhalt gut erklärt; Bezug zur Praxis

- Dialog fördern; - Angenehme Stimme; - Nicht zu viel / zu wenig Bewegungen; - Alles gut lesbar (Präsentation); - Ausführlich erklären; - Nicht "bloßstellen" ; - Übungsaufgaben; - Praxisnahe Beispiele

Praxisbezogen; Locker; Informativ; Nicht zu trocken

Interessanter Aufbau

Klare struktur; Alle wichtigen infos enthalten; Zusendung per email o.ä.

Ein unterhaltsamer Vortrag, z.B. durch einen nicht zu einstudiert wirkenden Text ;

Gute, prägnant formulierte Folien

sympathischer Dozent; Lockerheit zu Anwesenheit

Lustig; Interessant; Guter Einstieg

Präsentation vollständig/Skriptähnlich; Beispiele zur Realität; Gut verständlich

Guter Vortrag; Praxisbezogen

Stoff soll anwendbar sein; Einbeziehung in die Vorlesung; komplett (will nichts mit-schreiben müssen)

Präsentation selbsterklärend; In der Realität anwendbar (nicht nur theoretisch lernen); Guter Prof

Bezug zur Praxis vorhanden; Nicht langweilig; Alles wird erklärt (nicht nur sagen es gibt dies und jenes und dann selber nachschlagen müssen); Gut verständlich

Übungsaufgaben; Verständlich; Wofür wird es in der Realität gebraucht?

Selbsterklärend (falls der Prof schlecht ist); Praxisnah; Guter Prof :D; Nicht nur berie-seln lassen

## Anhang B: Ontologie und Protégé: Installation



UNIVERSITÄTS  
KLINIKUM  
HEIDELBERG

### Ontologie und Protégé: Installation

Kai Nico Groß

Institut für Medizinische Biometrie und Informatik  
Sektion Medizinische Informatik



### Inhaltsverzeichnis

- Ontologie in der Informatik
- Protégé
  - Verwendung
  - Installation



### Ontologie: In der Informatik

- Ontologien beschreiben einen Wissensbereich mit Hilfe einer standardisierenden Terminologie sowie Beziehungen und Ableitungsregeln zwischen den dort definierten Begriffen.
- Das gemeinsame Vokabular liegt meist einer Taxonomie oder einem Leitfaden zugrunde die Klassen, Relationen, Funktionen und Axiome enthalten.



## Ontologie: In der Informatik

- Ontologien werden in diesen Anwendungsgebieten verwendet:
  - Kommunikation
  - Automatisches Schließen
  - Repräsentation und Wiederverwendung von Wissen

imbi



## Ontologie: In der Informatik

- Ontologien werden in diesen Anwendungsgebieten verwendet:
  - Kommunikation
  - Automatisches Schließen
  - Repräsentation und Wiederverwendung von Wissen

imbi



## Ontologie: In der Informatik

- In der künstlichen Intelligenz und der Entscheidungsunterstützung werden Ontologien verwendet um logische Schlüsse aufgrund bekannter Ableitungsregeln zu ziehen.

imbi



## Ontologie: In der Informatik

- In der künstlichen Intelligenz und der Entscheidungsunterstützung werden Ontologien verwendet um logische Schlüsse aufgrund bekannter Ableitungsregeln zu ziehen.



## Protégé: Verwendung

Protégé ist eine Open-Source-Anwendung die an der Universität Stanford in Kalifornien entwickelt wurde zur Entwicklung und Pflege von Ontologien.

Außerdem wird es zur Entwicklung und Verbreitung der International Classification of Diseases (ICD), aktuell in Version 11, verwendet.



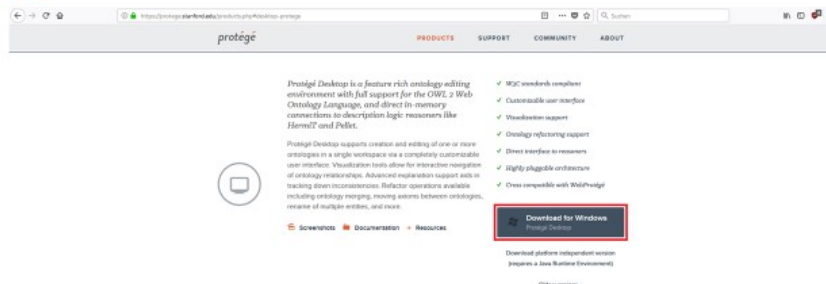
## Protégé: Installation

1. Aufrufen der Startseite über: <https://protege.stanford.edu>
2. Dem „Download Now“-Button zur Downloadseite folgen



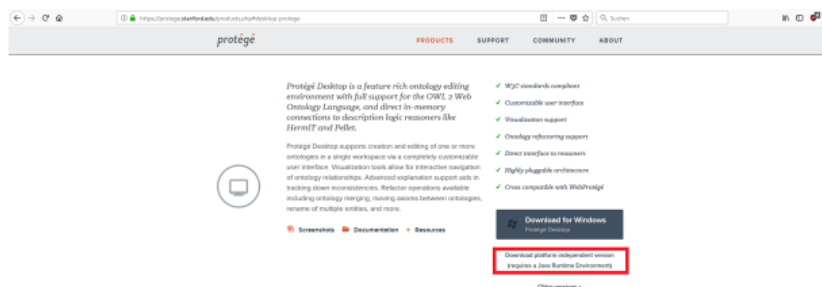
## Protégé: Installation (Windows)

3. Über den „Download for Windows“-Button den Download starten.



## Protégé: Installation (Independent)

3.1 Über den „Download platform independent version“-Button den Download starten.



## Protégé: Installation

4. Die Registrierung ermöglicht es Protégé, Ihre Nutzerdaten zur weiteren Optimierung von Protégé zu verwenden.

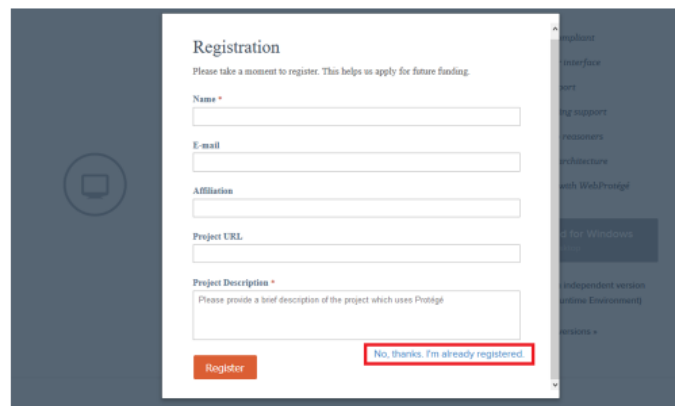
Zusätzlich werden Sie über Änderungen und Updates per E-Mail informiert.

Dieser Schritt kann alternativ übersprungen werden.



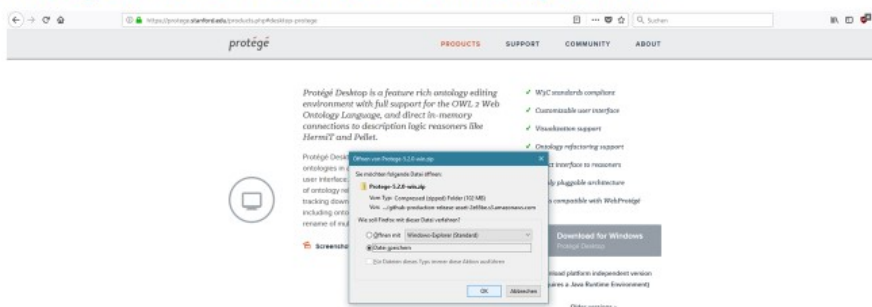


## Protégé: Installation



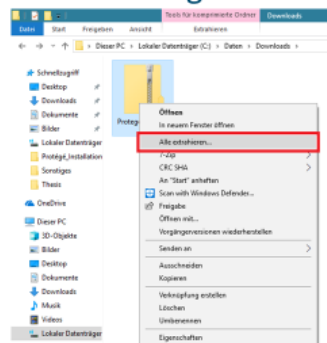
## Protégé: Installation

### 5. Protégé an einem Ort nach Wahl abspeichern



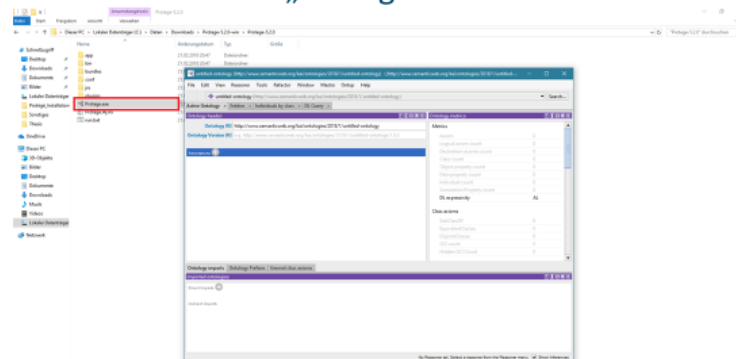
## Protégé: Installation

### 6. In entsprechenden Ordner navigieren und Protégé extrahieren



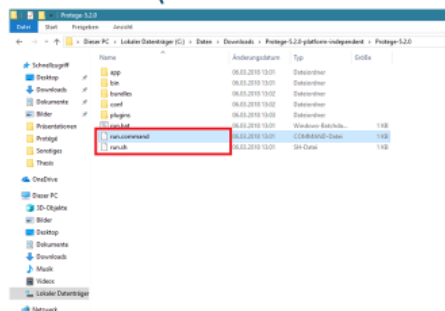
# Protégé: Installation (Windows)

## 7. In extrahiertem Ordner „Protege.exe“ ausführen



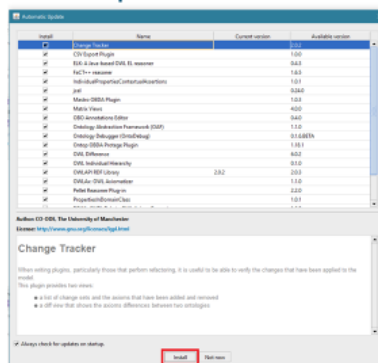
# Protégé: Installation (Independent)

## 7.1 Protégé unter Linux mit „run.sh“ und unter Mac OS mit „run.command“ ausführen (Java 8 oder neuer benötigt)



# Protégé: Installation

## 8. Sämtliche verfügbaren Updates auswählen und installieren



## Protégé: Installation - Fehlerbehandlung

*FaCT++ requires platform specific libraries which cannot be found!*

Für diesen Fehler folgen Sie bitte den weiteren Anweisungen auf den folgenden Seiten zur Behebung des Fehlers.

imbi



## Protégé: Installation - Fehlerbehandlung

1. Protégé beenden
2. Im Ordner „plugins“ im Installationsordner die Datei  
*uk.ac.manchester.cs.owl.factplusplus-x.y.z.jar*  
löschen. x.y.z steht dabei für die Version von FaCT++

imbi



## Protégé: Installation - Fehlerbehandlung

3. Unter  
*C:\Users\Username\AppData\Local\Temp\*  
alle Dateien beginnend mit „ProtegeCache“  
löschen.
4. Protégé neustarten

imbi



## Anhang C: Protégé: Kernkonzepte



UNIVERSITÄTS  
KLINIKUM  
HEIDELBERG

# Protégé: Kernkonzepte

Kai Nico Groß

Institut für Medizinische Biometrie und Informatik  
Sektion Medizinische Informatik



## Inhaltsverzeichnis

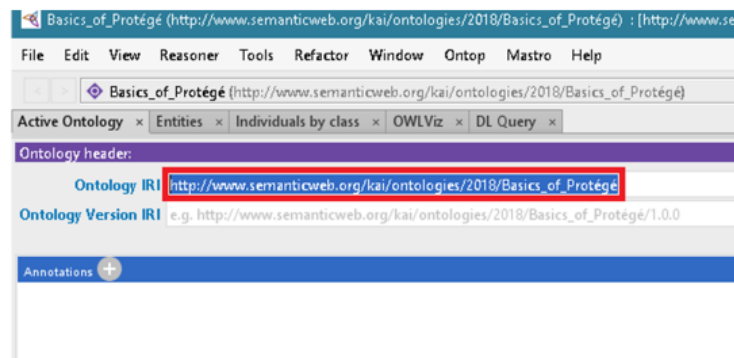
- Anlegen und Speichern eines Projektes
- Klassen und Subklassen
- Object Properties
- Data Properties
- Übungsaufgabe



## Anlegen und Speichern eines Projektes

- Nach dem Öffnen von Protégé kann in dem Feld „Ontology IRI“ der Name festgelegt werden.
- Gleichzeitig dient die dort verwendete Adresse als Ort der Veröffentlichung, sofern das Projekt nach Abschluss veröffentlicht werden soll.

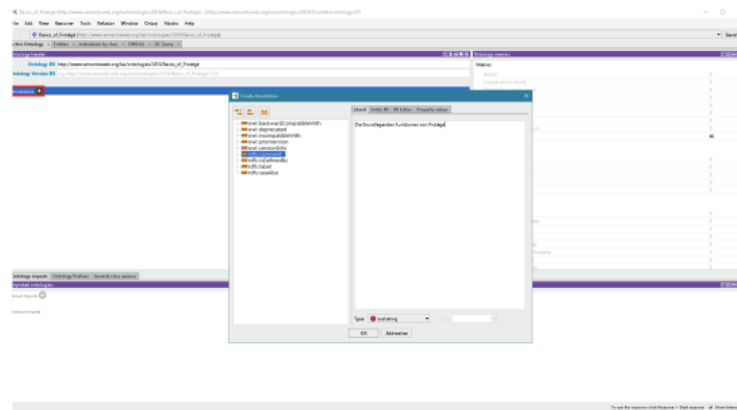
## Anlegen eines Projektes



## Hinzufügen von Kommentaren

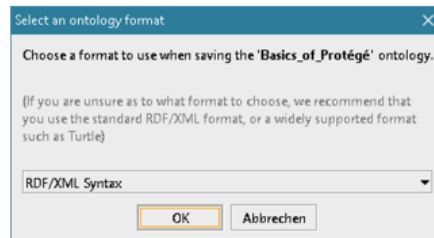
- Über den „+“-Button bei Annotation können Kommentare hinzugefügt werden.
- Dazu muss die Annotation als Comment gewählt werden und der „Type“ sollte als String ausgewählt werden.

## Hinzufügen von Kommentaren



## Speichern eines Projektes

- Projekte in Protégé sollten stets im RDF/XML-Syntax abgespeichert werden.



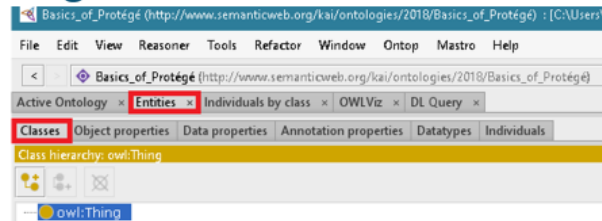
Notiz:

XML (Extensible Markup Language): Annotierung und Strukturierung von Daten und Dokumenten verwendet

RDF (Resource Description Framework): Ressourcen durch Eigenschaften zu beschreiben und Werte zuordnen

## Klassen in Protégé

- Um in Protégé Klassen anzulegen muss über den Reiter „Entities“ zu dem Subreiter „Classes“ gewechselt werden.



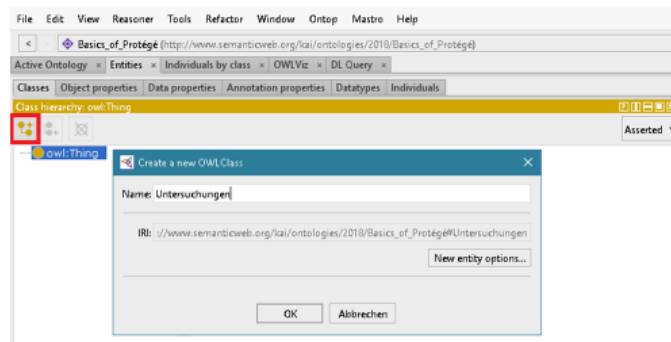
## Klassen in Protégé

- Alle Klassen in Protégé sind Subklassen von „owl:Thing“
- Über Subklassen können Klasseneigenschaften vererbt werden.

## Das Erstellen von Klassen in Protégé

- Um eine einzelne Klasse in Protégé zu erstellen muss zuerst „owl:Thing“ ausgewählt sein
- Dann kann über den „Add Subclass“-Button eine neue Klasse erstellt werden.

## Das Erstellen von Klassen in Protégé



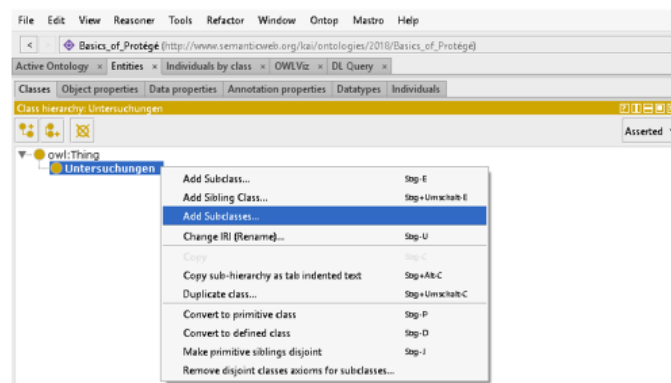
## Das Erstellen von Klassen in Protégé

- Alternativ können auch mehrere Subklassen gleichzeitig erstellt werden.

  1. Dazu muss der entsprechende Ort ausgewählt werden .
  2. Per Rechtsklick gelangt man an die Option „Add Subclasses...“.



## Das Erstellen von Klassen in Protégé



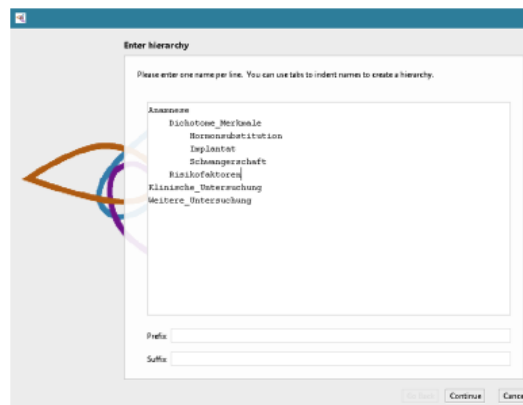


## Das Erstellen von Klassen in Protégé

- Dabei kann eine ganze Klassenhierarchie erstellt werden.
- Dazu muss die entsprechende Struktur über Tab-Einschübe erstellt werden.



## Das Erstellen von Klassen in Protégé



## Das Erstellen von Klassen in Protégé

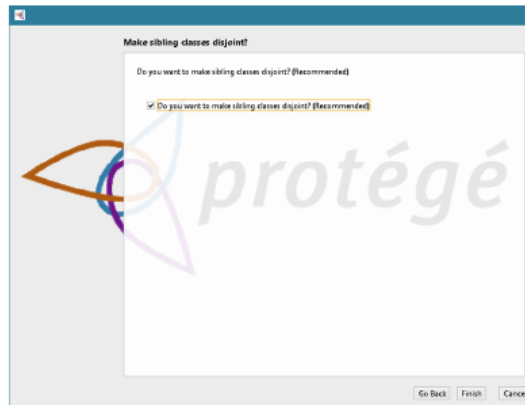
- Im Anschluss müssen die eben erstellten Klassen nun „disjoint“ werden.
- Das Disjoinen dient dem Zweck, dass mehrere Klassen nicht die selbe Instanz teilen können.
- Protégé erleichtert oder übernimmt das Disjoinen automatisch.



Notiz:

Apfel und Banane sind beides Subklassen von Obst. Wenn man nun ein Stück Obst hat, dann kann dieses entweder ein Apfel oder eine Banane sein, aber nicht Beides.

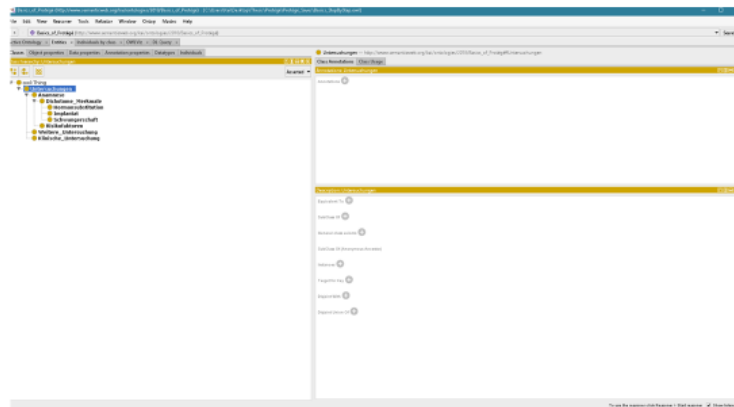
## Das Erstellen von Klassen in Protégé



imbi

UK  
HD

## Das Erstellen von Klassen in Protégé



imbi

UK  
HD

## Aufgabe 1

### Erstellen Sie folgende Klassen:

Zu den klinischen Untersuchungen gehören die Inspektion der Mamillen sowie der Mamillen-haut. Bei der Untersuchung der Mamillen ist zwischen benigner und maligner Qualität des Sekrets zu unterscheiden und die Seitigkeit des Sekrets zu bestimmen.

imbi



## Aufgabe 1

Bei der Untersuchung der Mamillenhaut ist der Hautbefund und die Nicht-Therapierbarkeit zu überprüfen. Weiterhin gehören zu den klinischen Untersuchungen die Palpation mit einem Tastbefund, dessen Ergebnis benigne oder maligne sein kann, und die klinische Untersuchung zu Schmerzen.

imbi



## Aufgabe 1

Zu den weiteren Untersuchungen gehören die Feinnadelpunktion, Kernspin, Mammographie und Ultraschall.

Es soll eine Klasse „Patient“ entstehen mit der Subklasse „Alter“, in der unterschieden wird, ob die Patientin unter 31 oder 31 und älter ist.

imbi



## Aufgabe 1

Die Klasse Diagnose erhält Untersuchungsindikationen zu jeder der „weiteren Untersuchungen“ sowie zu „keiner Indikation“.  
Die Klasse „Diagnostische Beziehung“ besteht aus der heuristischen Beziehung und überdeckender Beziehung.

imbi



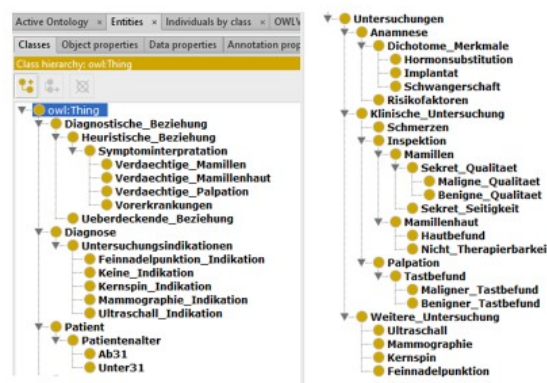
## Aufgabe 1

Die heuristischen Beziehungen bestehen aus den Symptominterpretationen von verdächtigen Mamillen, verdächtiger Mamillenhaut, verdächtiger Palpation und Vorerkrankungen.

imbi



## Aufgabe 1: Lösung



imbi



## Object Properties

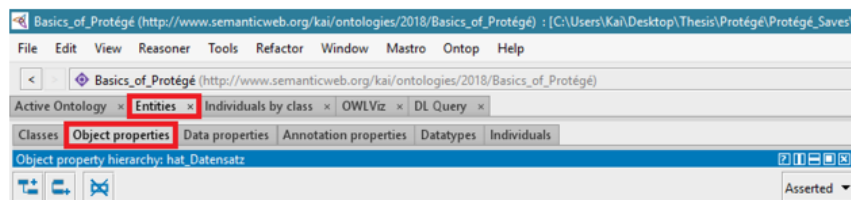
- Object Properties beschreiben in Protégé die Verbindungen zwischen Klassen und deren Instanzen.

imbi



## Object Properties

- Zum Erstellen von Object Properties muss zuerst auf den Tab „Object properties“ unter „Entities“ navigiert werden.



imbi



## Object Properties

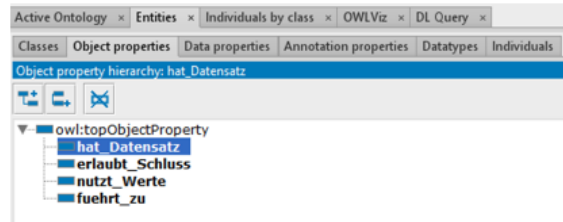
- Wie zuvor bei den Klassen, erbt auch bei den Object Properties alles von einer obersten Property, der „owl:topObjectProperty“.
- Object Properties können äquivalent zu den Klassen erzeugt werden.

imbi



## Object Properties

- Die Object Properties „hat\_Datensatz“, „erlaubt\_Schluss“, „fuehrt\_zu“ und „nutzt\_Werte“ sollen erstellt werden.



imbi

UK  
HD

## Object Properties

- Die Object Properties müssen mit den entsprechenden Klassen verknüpft werden. Dazu gibt es die Domain(s) und die Range(s).
- Die Domain gibt an welche Klasse die Object Property verwendet, wohingegen die Range angibt welche Klasse verwendet wird.

imbi

UK  
HD

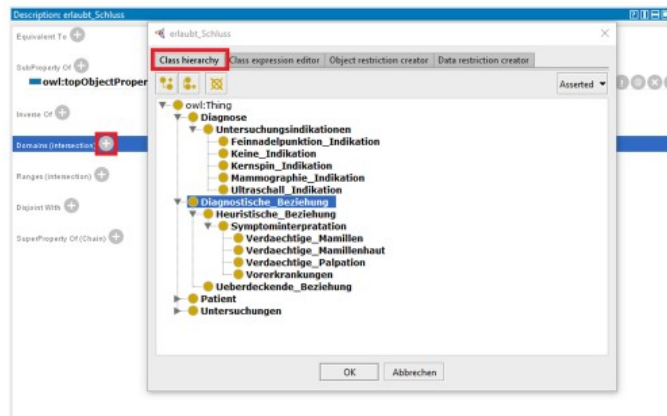
## Object Properties

- Eine Diagnostische Beziehung erlaubt einen Schluss einer Diagnose.
- Über den „+“-Butten bei Domain kann über den „Class hierachy“-Tab die Klasse „Diagnostische\_Beziehung“ als Domain für „erlaubt\_Schluss“ festgelegt werden.

imbi

UK  
HD

## Object Properties



imbi

UK  
HD

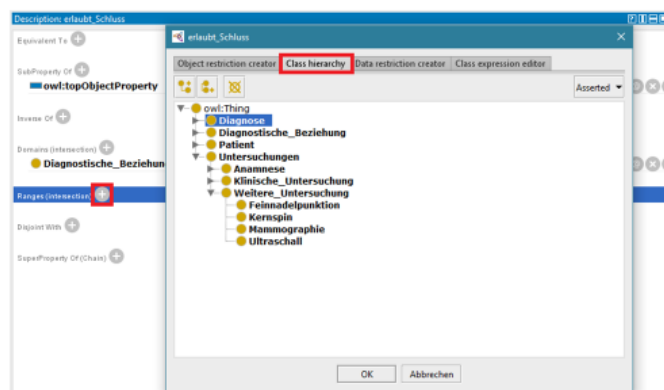
## Object Properties

- Über den „+“-Butten bei Range kann über den „Class hierarchy“-Tab die Klasse „Diagnostische\_Bezeichnung“ als Range festgelegt werden.

imbi

UK  
HD

## Object Properties



imbi

UK  
HD

## Aufgabe 2

- Es sollen folgende Object Properties erstellt werden:
  - Ein Patient hat einen Datensatz, der Untersuchungen, Diagnosen und diagnostische Beziehungen enthält
  - Eine diagnostische Beziehung nutzt die Werte der Untersuchung(en)

imbi



## Aufgabe 2

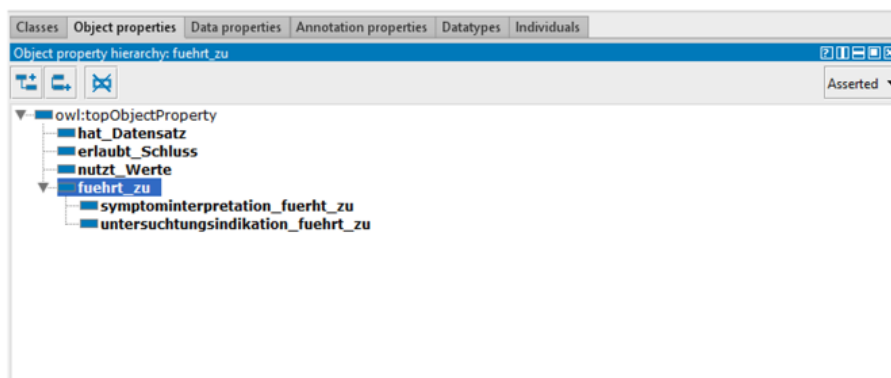
- Symptominterpretation führt zu einer klinischen Untersuchung
- Ein Untersuchungsindikator führt zu einer weiteren Untersuchung

Beide könnten bspw. eine Subproperty von „führt\_zu“ sein.

imbi



## Aufgabe 2: Lösung



imbi





## Data Properties

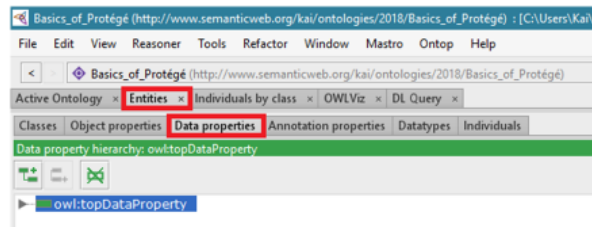
- Data Properties in Protégé beschreiben Beziehungen zwischen Instanzen und Datenwerten.

imbi



## Data Properties

- Zum Erstellen von Data Properties muss zuerst zu dem Tab „Data properties“ unter „Entities“ navigiert werden.



imbi



## Data Properties

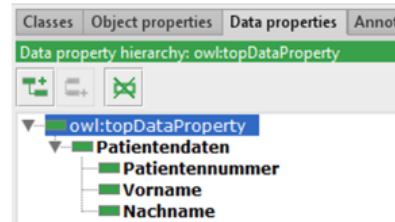
- Wie zuvor bei den Klassen, erbt auch bei den Data Properties alles von einer obersten Data Property, der „owl:topDataProperty“
- Data Properties können äquivalent zu den Klassen und Object Properties erstellt werden.

imbi



## Data Properties

- Es sollen als Data Properties Patientendaten angelegt werden, bestehend aus einer Patientennummer, Vor- und Nachname.



imbi

UK  
HD

## Data Properties

- Wie bei den Object Properties beschreibt die Domain hier, welche Klasse Bezug auf die Daten nimmt.
- Mit der Range wird angegeben in welchem Datentyp entsprechende Daten vorhanden sein müssen.

imbi

UK  
HD

## Data Properties

- Durch die Domain „Patient“ der Klasse Patientendaten haben alle Subklassen ebenfalls dieselbe Domain, sofern keine andere eingetragen wird.

imbi

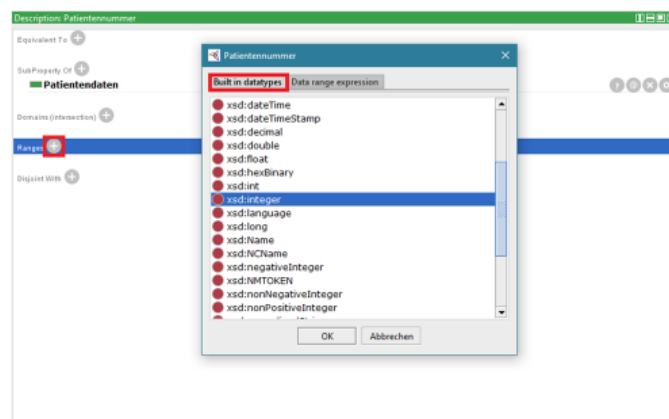
UK  
HD

## Data Properties

- Um den Datentyp einer Dataproperty festzulegen kann über den „+“-Button bei Range zu dem Tab „Built in datatypes“ navigiert werden.
- Hier kann der gewünschte Datentyp festgesetzt werden.



## Data Properties

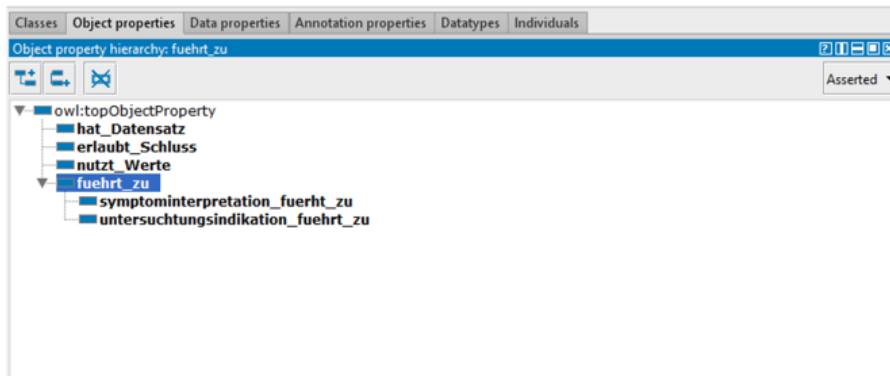


## Data Properties

- Für die Patientennummer soll ein Integer sowie für Vor-/Nachname ein String gesetzt werden.



## Aufgabe 2: Lösung



imbi

UK  
HD

## Aufgabe 3

Ein Ort besitzt eine Bevölkerung in Form einer Zahl und kann unterschieden werden in ein Land, Staat oder eine Stadt. Dabei ist ein Staat ein Teil von einem Land, eine Stadt befindet sich in einem Land und kann aber auch eine Hauptstadt des Landes sein. Ein Land kann an ein anderes Land grenzen.

imbi

UK  
HD

## Anhang D: Protégé: Vertiefung



UNIVERSITÄTS  
KLINIKUM  
HEIDELBERG

# Protégé: Vertiefung

Kai Nico Groß

Institut für Medizinische Biometrie und Informatik  
Sektion Medizinische Informatik



## Inhaltsverzeichnis

- Property Characteristics
- Property Chains
- Class Expressions
- Object Restrictions
- Reasoner
- Visualisierung
- Exportierung als Java Code

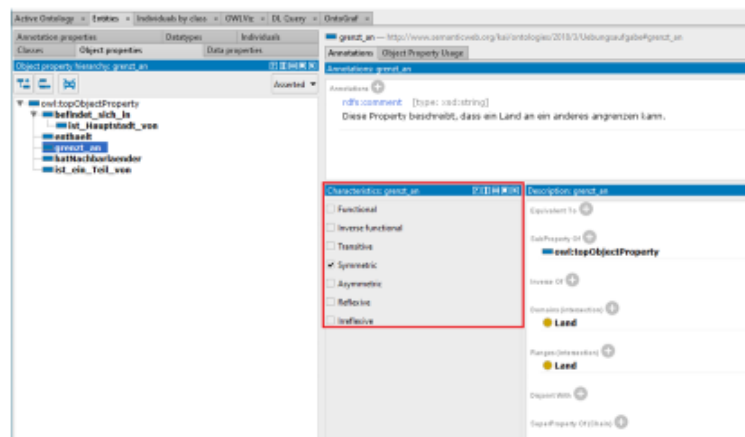


## Property Characteristics

- In Protégé stehen den Properties verschiedene Optionen bereit, um diese genauer definieren und beschreiben zu können.
- Dies ist möglich durch das Setzen eines Hakens in das entsprechende Kästchen der ausgewählten Property



## Property Characteristics



## Property Characteristics

- Object Properties Characteristics: Functional, Inverse functional, Transitive, Symmetric, Asymmetric, Reflexive, Irreflexive
- Data Properties Characteristics: Functional



## Property Characteristics: Functional

- Die Property kann für jedes Individuum höchstens einen Wert besitzen. D.h. es gibt für das Individuum höchstens eine ausgehende Beziehung der Property.
- Wenn mehrere Individuen als Werte dieser Property angegeben sind, so werden alle als dasselbe Objekt betrachtet.



## Property Characteristics: Inverse Functional

- Es gibt für das Individuum höchstens eine eingehende Property.  
Bei mehrere eingehenden Werten, werden diese als das gleiche Objekt bezeichnet.



Notiz:

Die ausgewählte Property ist inverse functional -> die inverse Property (explizit deklariert oder nicht) ist functional.

## Property Characteristics: Transitive

- Transitive Properties ermöglichen das Überspringen von Individuen.  
Wenn X mit Y in Verbindung steht und Y mit Z in Verbindung steht, besteht auch eine Verbindung zwischen X und Z.

imbi



## Property Characteristics: Symmetric

- Die Property hat sich selbst als Inverse.  
Wenn Individuum X mit Individuum Y in Verbindung steht, so steht auch Y mit X in Verbindung.

imbi



## Property Characteristics: Asymmetric

- Die Property kann sich nicht selbst als Inverse haben.  
Wenn Individuum X mit Individuum Y in Verbindung steht, so steht Y nicht mit X über die gleiche Property in Verbindung.

imbi





## Property Characteristics: Reflexive

- Die Characteristic Reflexive führt dazu, dass einzelne Individuen über diese Property mit sich selbst in Verbindung stehen.

imbi



## Property Characteristics: Irreflexive

- Die Characteristic Reflexive führt dazu, dass einzelne Individuen über diese Property nicht mit sich selbst in Verbindung stehen können.

imbi



## Property Chains

- Property Chains werden verwendet um Properties durch bereits bestehende Properties zu beschreiben.
- Dabei werden die verwendeten Properties durch den Operator „o“ miteinander verknüpft.

imbi



## Property Chains

- Zur Erstellung von Property Chains, muss bei der ausgewählten Property auf den „+“-Button bei „SuperProperty Of(Chain)“ geklickt werden, um die genauen Properties mit dem o-Operator verknüpft werden.

imbi



## Property Chains

- Beispiel:  
Die Property „hatGroßvater“ lässt sich durch die existierende Property „hatVater“ beschreiben, indem sie hintereinander mit sich selbst verkettet wird:
- $\text{hatGroßvater} = \text{hatVater} \circ \text{hatVater}$

imbi



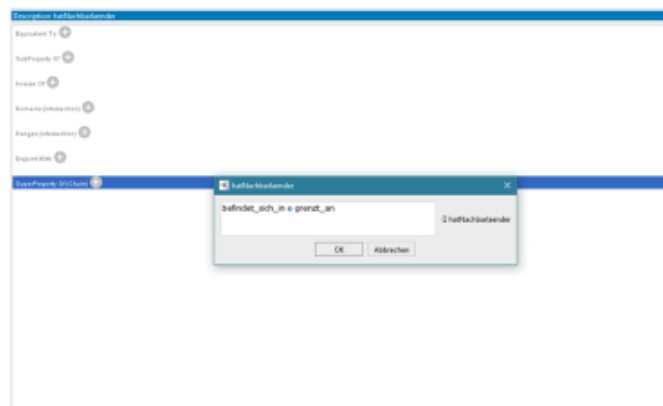
## Aufgabe 1

Erstellen sie die Property „hatNachbarlaender“ durch eine PropertyChain um eine Stadt mit den Nachbarländern des Landes zu verbinden, in dem sich die Stadt befindet.

imbi



## Aufgabe 1: Lösung



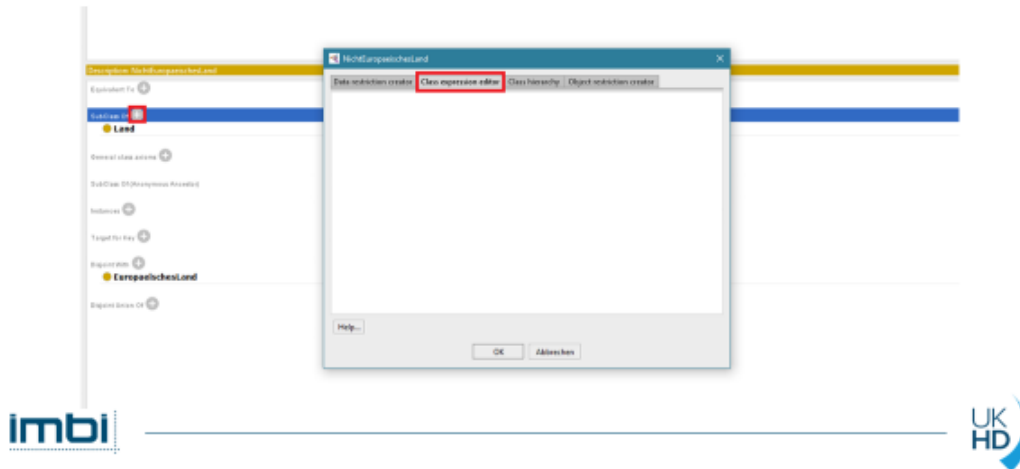
## Class Expressions

- Class Expressions werden genutzt um Individuen mit gemeinsamen Merkmalen zu beschreiben. Dazu werden Klassennamen, als auch die Class Expressions verwendet.
- Class Expressions können beliebig geschachtelt werden um entsprechende Aussagen zu treffen.

## Class Expressions

- Um eine Class Expression zu erstellen, muss bei entsprechender Klasse auf den „+“-Button bei „SubClass Of“ geklickt und zu dem Tab „Class Expressions“ navigiert werden.

## Class Expressions



## Class Expressions

- Zu den Class Expressions gehören folgende sich selbst erklärende Operatoren:
  - and
  - or
  - not



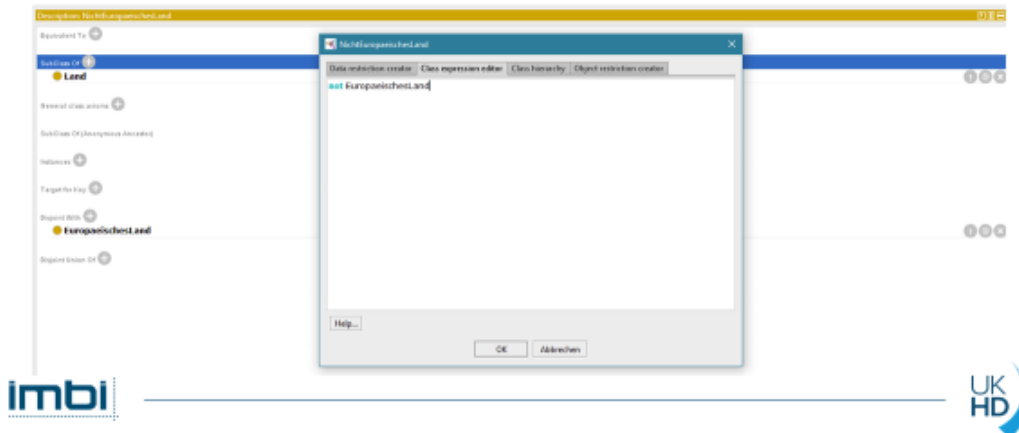
## Aufgabe 2

Erstellen Sie die Unterklassen „EuropaeischesLand“, „NichtEuropaeischesLand“ und verwenden Sie eine Class Expression um „NichtEuropaeischesLand“ zu definieren.

Erstellen Sie die Klasse Gegend und beschreiben Sie diese durch Länder oder Staaten.



## Aufgabe 2: Lösung



### Object Restrictions

- Object Restrictions werden genauso wie Class Expressions genutzt um Individuen genauer zu beschreiben.
- Die Object Restrictions verwenden nicht nur Klassen, sondern auch Properties und den benötigten RestrictionType.

### Object Restrictions

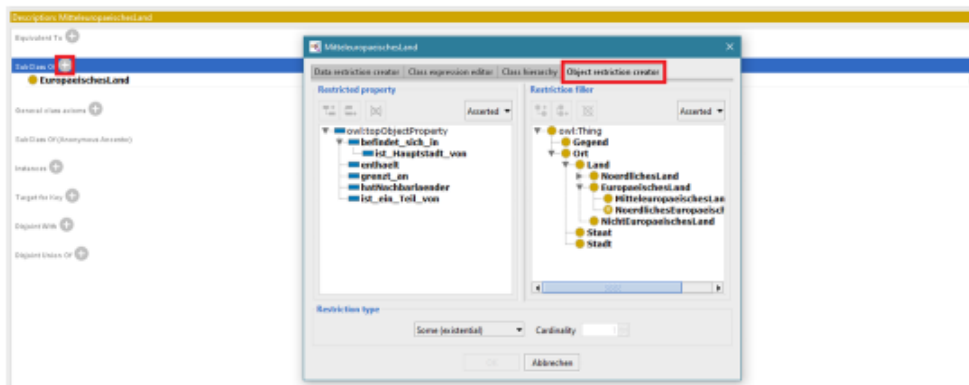
- Folgende RestrictionTypes gibt es:
  - some
  - value
  - only
  - min
  - max
  - exactly

## Object Restrictions

- Um eine Object Restriction zu erstellen, muss bei entsprechender Klasse auf den „+“-Button bei „SubClass Of“ geklickt und zu dem Tab „Object Restriction Creator“ navigiert werden.



## Object Restrictions



## Object Restrictions

- Alternativ kann die Object Restriction auch unter den Class Expressions von Hand eingetippt werden. Dies ist besonders nützlich wenn eine Klasse statt eines Individuum z.B. als Value benötigt wird.

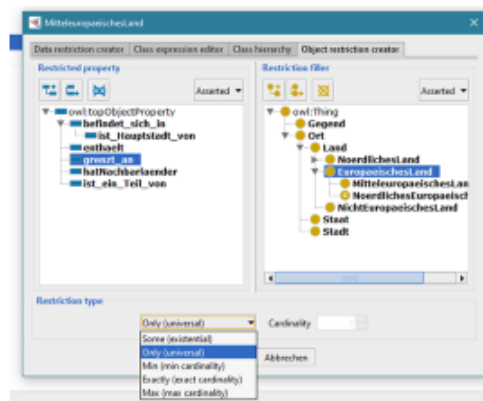


## Object Restrictions

- Beispiel:  
Es soll die Klasse „MittleuropaeischesLand“ durch bereits bestehende Klassen und Properties beschrieben werden.  
Ein „MittleuropaeischesLand“ ist ein eu. Land, welches nur an eu. Länder grenzt.



## Object Restrictions

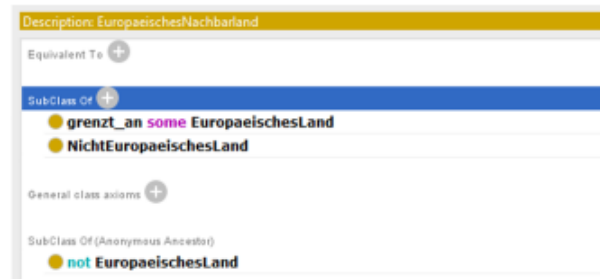


## Aufgabe 3

Erstellen Sie die Klasse „EuropaeschesNachbarland“ und ergänzen Sie die Beschreibung mit Hilfe der Object Restriction.



## Aufgabe 3: Lösung



### Reasoner

- Reasoner sind Programme zur Überprüfung von logischen Konsequenzen, die aus Fakten oder Axiomen folgen.
- Reasoner werden in Protégé verwendet, um die Ontologie auf Fehler zu überprüfen.

### Reasoner

- Bekannte OpenSource Reasoner sind zB. Pellet (implementiert in Java), Fact++ (implementiert in C++) sowie Hermit (ebenfalls in Java).
- Diese Reasoner sind alle in Protégé entweder direkt oder als Plugin verfügbar.

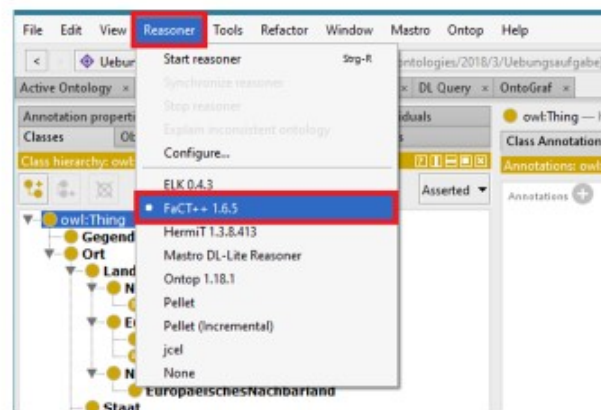


## Reasoner

- Zum Verwenden eines Reasoners, muss zuerst dieser ausgewählt werden.
- Dies geschieht in Protégé unter dem entsprechenden Tab „Reasoner“.



## Reasoner



## Reasoner

- Im selben Tab kann der Reasoner dann gestartet und wieder gestoppt werden.
- Ein gestarter Reasoner wird unten rechts in Protégé als „active“ angezeigt.
- Bei einer Interferenz schreitet der Reasoner ein und zeigt das Problem auf.



## Visualisierung

- Die Visualisierung kann genutzt werden um die Zusammenhänge einer Ontologie graphisch darzustellen und diese überprüfen und veranschaulichen zu können.

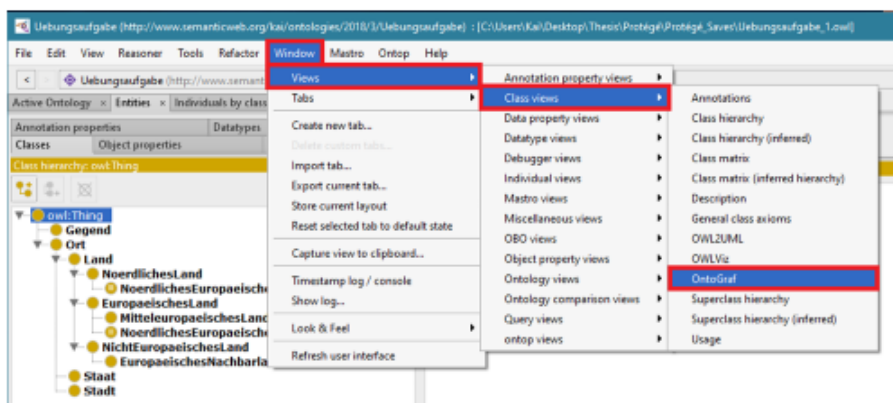


## Visualisierung

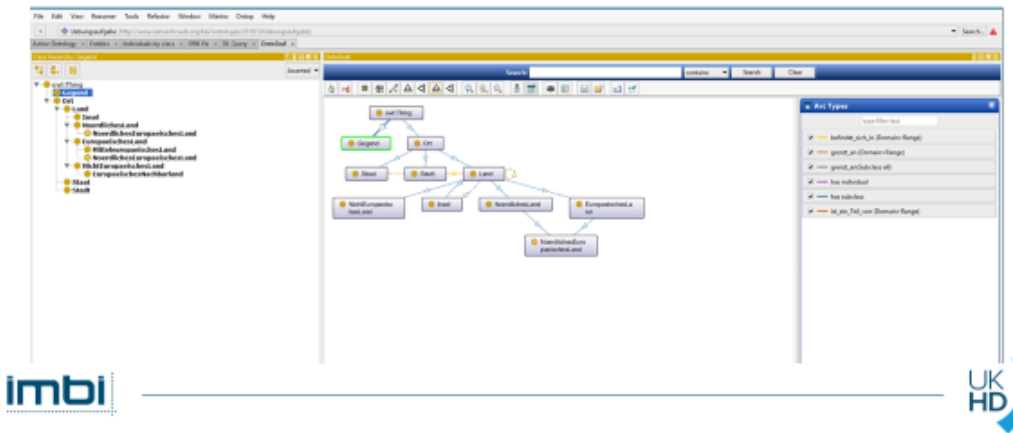
- Um eine Visualisierung auszuwählen folgt man dem Tab „Window“ über „Views“ zu den „Class views“. Hier kann ein bereits vorhandenes Plugin die Visualisierung übernehmen.
- Im folgenden Beispiel wird „OntoGraf“ verwendet.



## Visualisierung



## Visualisierung

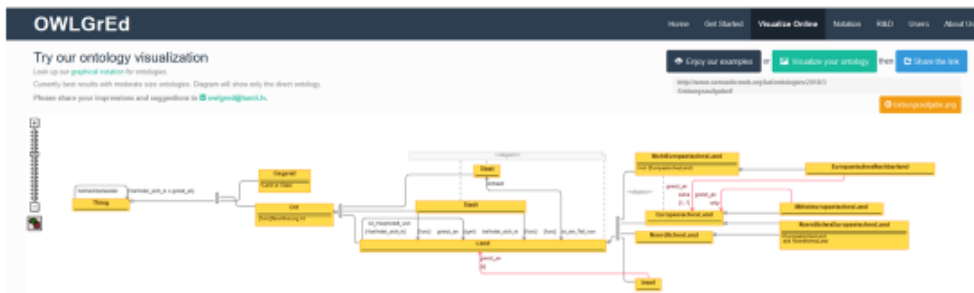


## Visualisierung

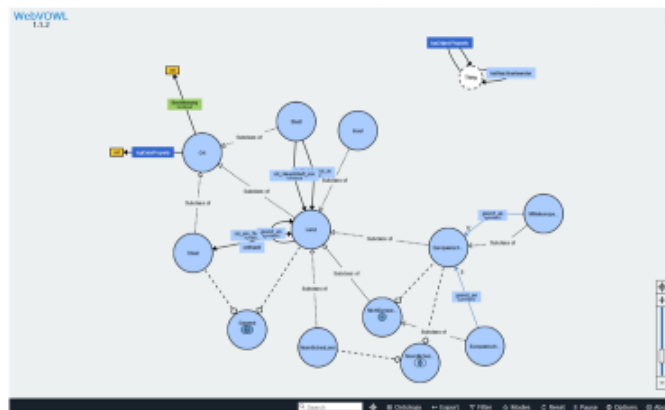
- Alternativ zu den Plugins in Protégé, bieten diverse Visualisierungs-Programme auch Web-Versionen zur Visualisierung von Ontologien an.



## Visualisierung: [owlgred.lumii.lv/online\\_visualization](http://owlgred.lumii.lv/online_visualization)



## Visualisierung: [www.visualdataweb.de/webvowl/](http://www.visualdataweb.de/webvowl/)



imbi

UK  
HD

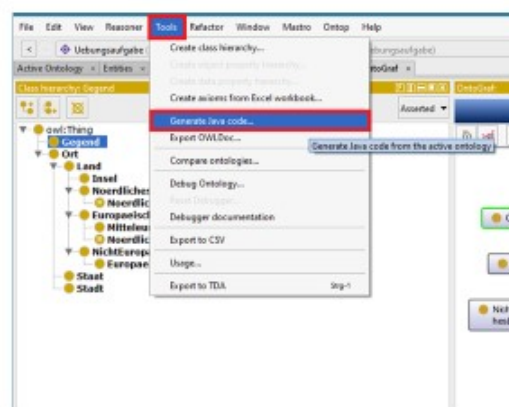
## Exportierung als Java-Code

- Protégé bietet die Möglichkeit an, die erstellte Ontologie als Java-Code zu generieren und gegebenenfalls die Ontologie als Grundlage für ein Programm zu verwenden.
- Dazu wird unter dem Tab „Tools“ die Option „Generate Java code...“ aufgerufen.

imbi

UK  
HD

## Exportierung als Java-Code



imbi

UK  
HD

## Exportierung als Java-Code

- Daraufhin fragt Protégé nach Zielordner und Dateinamen.
- Nach der Bestätigung wird der Java-Code generiert und im Zielordner abgelegt.



## Exportierung als Java-Code

Name	Änderungsdatum	Typ	Größe
impl	29.04.2018 19:44	Dateiordner	
EuropaeischesLand.java	29.04.2018 19:44	JAVA-Datei	4 KB
EuropaeischesNachbarland.java	29.04.2018 19:44	JAVA-Datei	4 KB
Gegend.java	29.04.2018 19:44	JAVA-Datei	1 KB
Insel.java	29.04.2018 19:44	JAVA-Datei	4 KB
Land.java	29.04.2018 19:44	JAVA-Datei	4 KB
MitteleuropaeischesLand.java	29.04.2018 19:44	JAVA-Datei	4 KB
NichtEuropaeischesLand.java	29.04.2018 19:44	JAVA-Datei	4 KB
NoerdlichesEuropaeischesLand.java	29.04.2018 19:44	JAVA-Datei	4 KB
NoerdlichesLand.java	29.04.2018 19:44	JAVA-Datei	4 KB
Ort.java	29.04.2018 19:44	JAVA-Datei	3 KB
Staat.java	29.04.2018 19:44	JAVA-Datei	4 KB
Stadt.java	29.04.2018 19:44	JAVA-Datei	4 KB
Test.java	29.04.2018 19:44	JAVA-Datei	18 KB
Vocabulary.java	29.04.2018 19:44	JAVA-Datei	11 KB



## Anhang E: ProKEt: Allgemeines



UNIVERSITÄTS  
KLINIKUM  
HEIDELBERG

# ProKEt: Allgemeines

Kai Nico Groß

Institut für Medizinische Biometrie und Informatik  
Sektion Medizinische Informatik



## Inhaltsverzeichnis

- ProKEt
  - Verwendung
  - Vorbereitung
  - Hochladen und Ausführen von Wissensbasen



## Verwendung

- ProKEt ist ein Programm zur Verwendung von Wissensbasen in tabellarischer Form.
- Es ermöglicht das Herleiten von Lösungen aus Wissenssammlungen durch das regelbasierte Expertensystem

imbi



## Vorbereitung

- Im Folgenden wird das Programm KnowOF verwendet, welches auf dem ProKEtkern arbeitet.
- Freundlicherweise hat die Uni Würzburg den Zugang hierzu erteilt.

imbi



## Vorbereitung

- Nur Wissensbasen im Excelformat .xlsx verwendbar
- Um die Wissensbasen entsprechend zu erstellen, wird ein Programm benötigt, welches dieses Speicherformat ermöglicht.

imbi



Notiz:

Programme die in der Lage sind Tabellen als .xlsx zu speichern: MS Office, Open Office, Numbers for Mac, Google Sheets

## Vorbereitung

- KnowOF ist eine webbasierte Anwendung.
- Zur Verwendung wird eine aktive Internetverbindung und ein Browser benötigt.
- Getestete kompatible Browser sind:
  - Chrome, Firefox, Internet Explorer, Opera, Safari

imbi



## Vorbereitung

- Um KnowOF aufzurufen verwenden Sie folgenden Link:  
[http://winc031.informatik.uni-wuerzburg.de:8080/testing\\_master/knowOfStatic/](http://winc031.informatik.uni-wuerzburg.de:8080/testing_master/knowOfStatic/)

imbi





## Vorbereitung

- Sämtliche Inhalte dieser und folgender Präsentationen kann in der ProKEt-Dokumentation nachgeschlagen werden:
- <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/Einfuehrung>



## Vorbereitung

Willkommen zur Wissenserwerbsumgebung KnowOF

Wissensbasis hochladen	
Wissensbasis wählen:	<input type="button" value="Durchsuchen..."/> keine Datei ausgewählt
Hochladen wählen:	<input type="button" value="Durchsuchen..."/> keine Datei ausgewählt
<input type="button" value="Hochladen"/>	<input type="button" value="Hochladen mit Standardlayout laden"/>
<input type="button" value="Dienst anzeigen"/>	<input type="button" value="Wissensbasis mit anderem Layout laden"/>
	<input type="button" value="Wissensbasis übertragen"/>

Gruppen verwalten	
Wissensbasen verwalten	Gruppen Verwaltung anzeigen
Keine Wissensbasen in dieser Gruppe	Bitte wählen Sie eine Wissensbasis
Wissensbasen	Aktionen



## Hochladen und Ausführen von Wissensbasen

- Im Bereich „Wissensbasis hochladen“ unter „Wissensbasis wählen“ mit Hilfe des Buttons „Durchsuchen...“ die gewünschte Wissensbasis im .xlsx-Format auswählen.
- Mit Hilfe des „Hochladen“-Buttons wird die Datei hochgeladen.



# Hochladen und Ausführen von Wissensbasen

Willkommen zur Wissenserwerbsumgebung KnowOF

The screenshot shows the 'Wissensbasis hochladen' section of the KnowOF interface. It includes a search bar for 'Wissensbasis wählen' with the text 'Beispiel\_Computerspielsucht.xlsx' and a 'Durchsuchen...' button. Below it, the 'Mediendatei wählen' section shows 'Keine Datei ausgewählt.' and a 'Durchsuchen...' button. The 'Hochladen' button is highlighted with a red box. Other buttons include 'Wissensbasis mit Standardlayout testen', 'Wissensbasis mit anderem Layout testen', and 'Wissensbasis übernehmen'. Below this, there are sections for 'Gruppen verwalten' and 'Wissensbasen verwalten'.

# Hochladen und Ausführen von Wissensbasen

- Nach erfolgreichem Hochladen, gibt KnowOF die Meldung, dass der Upload erfolgreich war.
- Bei einem Fehler kann man nach dem Aufrufen des „Bericht ansehen“-Buttons die Fehlerquellen der Tabelle einsehen.

The screenshot shows a success message at the top: 'Upload Erfolgreich' and 'Verifikation ist fehlgeschlagen. Genaueres dazu im Bericht'. Below this, the 'Wissensbasis hochladen' section is visible, with the 'Bericht ansehen' button highlighted in red. The rest of the interface is the same as in the previous screenshot.

# Hochladen und Ausführen von Wissensbasen

Willkommen zur Wissenserwerbsumgebung KnowOF

The screenshot shows the 'Wissensbasis hochladen' section of the KnowOF interface. It includes a search bar for 'Wissensbasis wählen' with the text 'Beispiel\_Computerspielsucht.xlsx' and a 'Durchsuchen...' button. Below it, the 'Mediendatei wählen' section shows 'Keine Datei ausgewählt.' and a 'Durchsuchen...' button. The 'Hochladen' button is visible, and the 'Bericht ansehen' button is highlighted with a red box. Other buttons include 'Wissensbasis mit Standardlayout testen', 'Wissensbasis mit anderem Layout testen', and 'Wissensbasis übernehmen'. Below this, there are sections for 'Gruppen verwalten' and 'Wissensbasen verwalten'.

# Hochladen und Ausführen von Wissensbasen

## Fehlerliste

- Fehler in Zelle C12

	A	B	C	D	E	F
1	Wenn	ID	Typ	Fragen-Antworten	Aktion	Layout
2						
3			U	Computerspielrecht	1.0	
4		CS	JN	Spielen Sie Computerspiele?		
5	CS = Yes		OC	Wie häufig spielen Sie?		dropdown
6			A	weniger als 1h pro Tag		
7			A	1h bis 3h pro Tag		
8			A	mehr als 3h pro Tag		
9				D-Header		Suchverhalten
10				D-SSP		x
11				D-SP		1.0
				U	Absichtlicher Fehler	

## Detaillierte Fehlerliste

- Fehler in Zelle C12: Hierarchy violation, actual value 'U' is none of the expected ['d-ab', 'd-bf', 'd-fb', 'd-fs', 'd-header', 'd-qc', 'd-qn', 'd-qnn', 'd-sp', 'd-ssp', 'd-typ'] children



## Anhang F: ProKEt: Spalten



UNIVERSITÄTS  
KLINIKUM  
HEIDELBERG

# ProKEt: Spalten

Kai Nico Groß

Institut für Medizinische Biometrie und Informatik  
Sektion Medizinische Informatik



## Inhaltsverzeichnis

- Standardtypen:
  - Typ
  - Fragen-Antworten
  - ID
  - Wenn
  - Aktion
  - Layout
- Regeln
  - RH
  - H
  - HN



Das in dieser Präsentation verwendete Beispiel ist frei erfunden und dient nur dem Zweck der Veranschaulichung und liegt keinem existierenden gültigen Leitfaden zu Grunde.

imbi



## Spalten

- Standardtypen beschreiben, was in welcher Spalte der Tabelle stehen darf, damit diese richtig vom System verwendet werden kann.

imbi



## Spalten

- Die Spalten der Tabelle können in ProKEt in beliebiger Reihenfolge sortiert und platziert werden.
- Aus Gründen der Übersicht wird empfohlen, die folgende Sortierung zu verwenden.

imbi



## Beispiel

Wenn	ID	Typ	Fragen-Antworten	Aktion	Layout
		U	Computerspielsucht	1	
	CS	JN	Spielen Sie Computerspiele?		
CS = Yes		OC	Wie häufig spielen Sie?		
		A	weniger als 1h pro Tag		
		A	1h bis 3h pro Tag		
		A	mehr als 3h pro Tag		

imbi



## Typ

- Dies ist die wichtigste Spalte bei der Nutzung von ProKEt/KnowOF.
- Hier wird über den Zeilentyp definiert, wie die entsprechende Zeile auszulesen und zu behandeln ist.

imbi



## Typ

- In Zeile 3 bspw. wird mit dem Typ „U“ eine Untersuchung eingeleitet, während in Zeile 6 eine Antwort mit dem Typ „A“ gesetzt wird.
- Weitere Typen werden im späteren Verlauf genauer betrachtet.

imbi



## Fragen-Antworten

- In dieser Spalte werden alle möglichen Fragen und Antworten aufgelistet.
- Hier steht der größte Teil des dem Benutzer angezeigten Textes.

imbi



## Fragen-Antworten

▼ 1. Computerspielsucht

**Spielen Sie Computerspiele?**

Yes

No

Unknown

imbi



Notiz:

In der deutschen Version von ProKEt wird nur bei der „Date“-Antwortmöglichkeit „Tag/Monat/Jahr“ angegeben, alles weitere bleibt auf englisch (z.B. yes/no/unknown)

## ID

- In dieser Spalte kann einer Frage oder einer Antwort eine ID hinzugefügt werden, um im späteren Verlauf leichter darauf zugreifen zu können.

imbi



## ID

- Der Ja-Nein-Frage „Spielen Sie Computerspiele?“ wird die ID „CS“ zugeordnet.

Wenn	ID	Typ	Fragen-Antworten	Aktion	Layout
		U	Computerspielsucht	1	
	CS	JN	Spielen Sie Computerspiele?		

imbi



## Wenn

- In dieser Spalte werden Bedingungen eingetragen, die erfüllt sein müssen, um z.B. Fragen oder Untersuchungen sichtbar zu machen.

imbi





## Wenn

- Um die OneChoice Frage „Wie häufig spielen Sie?“ sehen zu können, muss die vorherige Frage „Spielen Sie Computerspiele?“ mit „Ja“ beantwortet sein.

Wenn	ID	Typ	Fragen-Antworten	Aktion	Layout
		U	Computerspielsucht		1
	CS	JN	Spielen Sie Computerspiele?		
CS = Yes		OC	Wie häufig spielen Sie?		



Notiz:

Bei einer One-Choice Frage (OC) kann nur eine der folgenden Antworten (vom Typ A) ausgewählt werden.

## Wenn

▼ 1. Computerspielsucht

**Spielen Sie Computerspiele?**

Yes

No

Unknown

**Wie häufig spielen Sie?**

weniger als 1h pro Tag

1h bis 3h pro Tag

mehr als 3h pro Tag

Unknown



## Aktion

- Die Aktionsspalte legt fest, welche Untersuchungen „U“ angezeigt werden, ohne dass Bedingungen erfüllt sein müssen.
- Die Nummerierung beginnt mit „1“.
- Die Reihenfolge der Tabelle, nicht die Nummerierung, legt die Anzeigenfolge fest.

imbi



## Aktion

Wenn	ID	Typ	Fragen-Antworten	Aktion	Layout
		U	Computerspielsucht	1	
	CS	JN	Spielen Sie Computerspiele?		

▼ 1. Computerspielsucht

**Spielen Sie Computerspiele?**

Yes

No

Unknown

**Wie häufig spielen Sie?**

weniger als 1h pro Tag

1h bis 3h pro Tag

mehr als 3h pro Tag

Unknown

imbi



## Aufgabe 1

- Erstellen Sie die weitere Untersuchung „Drogenabhängigkeit“ mit der Ja-Nein-Frage „Nehmen Sie Drogen?“.
- Weisen Sie eine Aktionsnummer zu, damit die Untersuchung angezeigt werden kann.

imbi



## Aufgabe 1:Lösung

1. Computerspielsucht

**Spielen Sie Computerspiele?**

- Yes
- No
- Unknown

2. Drogenabhängigkeit

**Nehmen Sie Drogen?**

- Yes
- No
- Unknown

imbi



### Layout

- In der Layout-Spalte können Informationen zu den jeweiligen Typen angegeben werden, um das Layout anzupassen.
- Dabei gibt es einen Layoutbefehl nur für den Typ „OC“ (OneChoice) und eine Reihe von Layout-Befehlen für den ganzen Dialog.

imbi



### Layout: OC

- Bei dem Typ „OC“ gibt es die Möglichkeit, die Antworten in einem Dropdownmenü anzeigen zu lassen.
- Dazu wird in der Layoutspalte bei einem OC-Typ „dropdown“ gesetzt.

imbi



## Layout: OC

Wenn	ID	Typ	Fragen-Antworten	Aktion	Layout
		U	Computerspielsucht		1
	CS	JN	Spielen Sie Computerspiele?		
CS = Yes		OC	Wie häufig spielen Sie?		dropdown
		A	weniger als 1h pro Tag		
		A	1h bis 3h pro Tag		
		A	mehr als 3h pro Tag		

The screenshot shows a user interface for a survey. It features a section titled '1. Computerspielsucht'. Underneath, there are two questions. The first is 'Spielen Sie Computerspiele?' with radio buttons for 'Yes', 'No', and 'Unknown'. The second is 'Wie häufig spielen Sie?' with a dropdown menu. The dropdown menu is open, showing three options: 'weniger als 1h pro Tag', '1h bis 3h pro Tag', and 'mehr als 3h pro Tag'. The first option is currently selected.

## Layout: Dialog

- Layoutbefehle, die für den gesamten Dialog gelten, müssen als Block am Ende einer Wissensbasis aufgelistet werden.
- Die Typen D-Header, D-SP und D-SSP werden betrachtet. Weitere Typen sind für die grundlegende Verwendung nicht von Bedeutung.

## Layout: D-Header

- Um der gesamten Wissensbefragung eine Überschrift geben zu können, kann der „D-Header“-Typ verwendet werden.
- Dazu wird in der Typ-Spalte „D-Header“ und in der Layout-Spalte die gewünschte Überschrift gesetzt.

## Layout: D-SP

- Dieser Typ ermöglicht es in der Layout-Spalte anzugeben, wie viele Spalten ein Dialog nutzen soll.
- Mögliche Werte sind 1 bis 4.

imbi



## Layout: D-SSP

- „D-SSP“ ermöglicht das Anzeigen des SolutionPanels. In diesem Panel werden hergeleitete Diagnosen angezeigt.
- Um das SolutionPanel anzeigen zu lassen, muss in der Typ-Spalte „D-SSP“ und in der „Layout“-Spalte ein „x“ gesetzt werden.

imbi



## Layout: Dialog

Wenn	ID	Typ	Fragen-Antworten	Aktion	Layout
		U	Computerspielsucht	1	
	CS	JN	Spielen Sie Computerspiele?		
CS = Yes		OC	Wie häufig spielen Sie?		dropdown
		A	weniger als 1h pro Tag		
		A	1h bis 3h pro Tag		
		A	mehr als 3h pro Tag		
		D-Header			Suchtverhalten
		D-SSP			x
		D-SP			1

imbi



## Layout: Dialog



## Regeln

- Regeln werden verwendet, um aus gegebenen Antworten auf Lösungen/Diagnosen zu schließen oder diese auszuschließen.
- Regeln sind für den Nutzer von ProKEt nicht sichtbar. Die Ergebnisse dieser Regeln werden im SolutionPanel angezeigt.

## RH

- RH-Spalten werden verwendet um Lösungen herzuleiten.
- Dazu wird eine Reihe von Und-Verknüpfungen mehrerer Antworten verwendet.
- In der RH-Spalte werden für die Fragen die Antwortmöglichkeiten angegeben.

## RH

- Nur wenn der Nutzer alle Fragen jeweilig beantwortet hat, wird die Lösung als anerkannt betrachtet und im SolutionPanel angezeigt.

imbi



## RH

- In der „Lzeile“ der RH-Regel steht der Name der Lösung, während in der „LB“-Zeile der Wert dieser Lösung zugewiesen wird.
- Genauerer zu den beiden Typen „LZeile“ und „LB“ wird zu einem späteren Zeitpunkt erläutert.

imbi



## RH

Wenn	ID	Typ	Fragen-Antworten	Aktion	Layout	RH
		LZeile				Test-Lösung
		LB				p7
		U	Untersuchung1		1	
		JN	Ja-Nein-Frage 1			j
		JN	Ja-Nein-Frage 2			n
		MC	Wählen Sie aus den folgenden Möglichkeiten			
		A	Antwort 1			x
		A	Antwort 2			x
		A	Antwort 3			
		Num	Schreiben sie eine Zahl!			
		OC	Wählen sie eine der folgenden Farben:			
		A	Blau			
		A	Grün			
		A	Rot			x
		D-SSP			x	

imbi



## RH

The screenshot displays a survey form titled "1. Untersuchung1" with two columns of questions. The left column, "Ja-Nein-Frage 1", has radio buttons for "Yes" (selected), "No", and "Unknown". Below it is a section "Wählen Sie aus den folgenden Möglichkeiten" with checkboxes for "Antwort 1" (selected), "Antwort 2", "Antwort 3", and "Unknown". The right column, "Ja-Nein-Frage 2", has radio buttons for "Yes", "No" (selected), and "Unknown". Below it is a section "Schreiben sie eine Zahl!" with a text input field and an "Unknown" radio button. At the bottom, there is a section "Wählen sie eine der folgenden Farben:" with radio buttons for "Blau", "Grün", "Rot" (selected), and "Unknown". To the right, a "Solutions" panel shows a green plus icon and the text "Test-Lösung (999)".

## H (Heuristisch)

- In H-Spalten werden einer Lösung Punkte gegeben, die abhängig von den vom Nutzer gegebenen Antworten sind.
- Bei ausreichender Punktzahl gilt die Lösung als bestätigt, wobei eine ausreichend negative Punktzahl diese widerlegt.

## H (Heuristisch)

- Der Name der Lösung befindet sich in der LZeile der H-Spalte.
- In der H-Spalte werden bei den Antworten Punkte vergeben.



## H (Heuristisch)

Wenn	ID	Typ	Fragen-Antworten	Aktion	Layout	H
		Lzeile				Suchtverhalten
		U	Computerspielsucht		1	
	CS	JN	Spielen Sie Computerspiele?			
CS = Yes		OC	Wie häufig spielen Sie?			
		A	weniger als 1h pro Tag			
		A	1h bis 3h pro Tag			p3
		A	mehr als 3h pro Tag			p4
		U	Drogenabhängigkeit	2		
	D	JN	Nehmen Sie Drogen?			p5
D=Yes		OC	Wie häufig nehmen Sie Drogen?		dropdown	
		A	Weniger als 4 mal Jährlich			n4
		A	Alle 5 bis 11 Wochen			n1
		A	Alle 2 bis 4 Wochen			p3
		A	Häufiger als alle 2 Wochen			p6
		D-SSP			x	




## HN

- HN-Spalten unterscheiden sich von H-Spalten nur darin, dass Punkte bei einer „Ja-Nein-Frage“ mit der Antwort „Nein“ oder einer nicht gegebenen Antwort einer Multiple-Choice- oder One-Choice-Frage trotzdem angerechnet werden.




## Anhang G: ProKEt: Typen



UNIVERSITÄTS  
KLINIKUM  
HEIDELBERG

# ProKEt: Typen

Kai Nico Groß

Institut für Medizinische Biometrie und Informatik  
Sektion Medizinische Informatik



## Inhaltsverzeichnis

- Allgemein
- Typen:
  - Fragen-Typen
  - Erklärungs-Typen
  - Layout-Typen (wurden bereits behandelt)
  - Sonstige Typen



## Allgemein

- Typen werden in der Typ-Spalte festgelegt und sind die wichtigsten Eigenschaften der Tabelle.
- Sie geben an, wie die Zeile interpretiert werden muss.

imbi



## Fragen-Typen

- Fragen bilden den Kern einer Wissenseingabetabelle und müssen in der ganzen Wissensbasis eindeutig sein.
- Fragen, die mit „-I“ enden sollen nicht vom Nutzer beantwortet werden, sondern von ProKEt/KnowOF selbst.

imbi



Notiz:

„-I“ steht für „-Indirekt“

## Fragen-Typen

- Fragen-Typen:
  - Date
  - Num
  - JN
  - Num-I
  - JNU
  - OC
  - MC
  - Text



## Fragen-Typen: Date

- Eine Frage mit dem „Date“-Typ erfordert ein Datum für die Antwort.
- In der Layout-Spalte kann „YearOnly“ angegeben werden, damit der Nutzer nur das Jahr angeben muss.



## Fragen-Typen: Date

Wenn	ID	Typ	Fragen-Antworten	Aktion	Layout
		U	Körperlicher Zustand		1
		Date	Geburtsdatum		

▼ 1. Körperlicher Zustand

---

**Geburtsdatum**

Day    Month    Year



## Fragen-Typen: JN (Ja-Nein)

- JN definiert eine Frage, welche mit „Ja“ oder „Nein“ beantwortet werden kann.

Wenn	ID	Typ	Fragen-Antworten	Aktion	Layout
		U	Körperlicher Zustand		1
		Date	Geburtsdatum		
		JN	Rauchen Sie?		




## Fragen-Typen: JNU (Ja-Nein-Unbekannt)

- JNU definiert eine Frage, welche mit „Ja“, „Nein“ oder „Unbekannt“ beantwortet werden kann.

Wenn	ID	Typ	Fragen-Antworten	Aktion	Layout
		U	Körperlicher Zustand		1
		Date	Geburtsdatum		
		JN	Rauchen Sie?		
	VE	JNU	Sind Vorerkrankungen bekannt?		




Notiz:

Aufgrund der Servereinstellung wird „unknown“ bei jeder Frage als mögliche Antwort angezeigt.

## Fragen-Typen: MC (MultipleChoice)

- MC definiert eine Frage, bei welcher mehrere Antwortmöglichkeiten (oder auch keine) gewählt werden können.
- Auf eine MC-Frage müssen mindestens zwei mögliche Antworten (A-Typ) folgen.

imbi



## Fragen-Typen: MC (MultipleChoice)

Wenn	ID	Typ	Fragen-Antworten	Aktion	Layout
		U	Körperlicher Zustand		1
		Date	Geburtsdatum		
		JN	Rauchen Sie?		
	VE	JNU	Sind Vorerkrankungen bekannt?		
VE=yes		Text	Was für Vorerkrankungen?		
		MC	Körperverzierungen?		
		A	Tätowierung(en)		
		A	Piercing(s)		
		A	Keine		

imbi



## Fragen-Typen: Num (Numerisch)

- Der „Num“-Type definiert eine Frage, deren Antwort eine Zahl sein muss.
- Nach den Erklärungs-Elementen können bei Num-Fragen folgende optionale Elemente in genau dieser Reihenfolge zur besseren Definierung folgen.

imbi



## Fragen-Typen: Num (Numerisch)

### 1. Einheit:

– Definiert die Einheit des Zahlenwerts (z.B. km, g,...).

### 2. Min-Wert:

– Definiert den minimalen zulässigen Wert.

### 3. Max-Wert:

– Definiert den maximalen zulässigen Wert.

imbi



## Fragen-Typen: Num (Numerisch)

Wenn	ID	Typ	Fragen-Antworten	Aktion	Layout
		U	Körperlicher Zustand		1
		Date	Geburtsdatum		
		JN	Rauchen Sie?		
	VE	JNU	Sind Vorerkrankungen bekannt?		
VE=yes		Text	Was für Vorerkrankungen?		
		MC	Körperverzierungen?		
		A	Tätowierung(en)		
		A	Piercing(s)		
		A	Keine		
	K	Num	Körpergröße		
		Einheit	m		
		Min-Wert		0	
		Max-Wert		5	

imbi



## Fragen-Typen: Num-I (Numerisch-Indirekt)

- Unterscheidet sich vom „Num“-Typ darin, dass der Antwortwert vom System berechnet wird.
- Die Berechnung der Antwort erfolgt durch einen RF-Typ.

imbi



## Fragen-Typen: OC (OneChoice)

- OC-Fragen unterscheiden sich zu MC-Fragen nur darin, dass exakt eine einzige Antwort ausgewählt werden kann.
- Wie zuvor erwähnt kann in der Layout-Spalte für OC-Fragen der Layout-Typ „dropdown“ gesetzt werden.




## Fragen-Typen: OC (OneChoice)

Wenn	ID	Typ	Fragen-Antworten	Aktion	Layout
		U	Körperlicher Zustand		1
		K	Überspringen der bisherigen Daten		
		OC	Augenfarbe		dropdown
		A	Blau		
		A	Braun		
		A	Grau		
		A	Grün		




## Fragen-Typen: Text

- Eine Frage des Typs „Text“ ermöglicht dem Nutzer die Eingabe eines beliebigen Textes.

Wenn	ID	Typ	Fragen-Antworten	Aktion	Layout
		U	Körperlicher Zustand		1
		K	Überspringen der bisherigen Daten		
		OC	Augenfarbe		dropdown
		A	Blau		
		A	Braun		
		A	Grau		
		A	Grün		
		Text	Name		
		Text	Vorname		






Notiz:

Maximale Anzahl an Zeichen: 4683

## Gesamtes Beispiel

1. Körperlicher Zustand		
<b>Geburtsdatum</b> Day Month Year [ ] [ ] [ ] <input type="radio"/> Unknown	<b>Rauchen Sie?</b> <input type="radio"/> Yes <input type="radio"/> No <input type="radio"/> Unknown	<b>Sind Vorerkrankungen bekannt?</b> <input type="radio"/> Yes <input type="radio"/> No <input type="radio"/> Unknown
<b>Körperverzierungen?</b> <input type="checkbox"/> Tätowierung(en) <input type="checkbox"/> Piercing(s) <input type="checkbox"/> Keine <input type="radio"/> Unknown	<b>Körpergröße</b> [ ] m <input type="radio"/> Unknown	<b>Gewicht</b> [ ] kg <input type="radio"/> Unknown
<b>Augenfarbe</b> Please select... <input type="radio"/> Unknown	<b>Name</b> [ ] <input type="radio"/> Unknown	<b>Vorname</b> [ ] <input type="radio"/> Unknown



## Aufgabe 1

- Erstellen Sie für die Untersuchung „Körperlicher Zustand“ folgende zusätzlichen Fragen:
  - Trinken Sie Alkohol, wenn Ja, 1x pro Monat/Woche/Tag
  - Taillenumfang: mindestens 0 cm



Haarfarbe: Blond, Braun, Rot oder Schwarz



## Aufgabe 1:Lösung

Wenn	ID	Typ	Fragen-Antworten	Aktion	Layout
	ALK	JN	Trinken Sie Alkohol?		
ALK=yes		OC	Wie oft trinken Sie Alkohol?		
		A	1 mal pro Monat		
		A	1 mal pro Woche		
		A	täglich		
		Num	Taillenumfang		
		Einheit	cm		
		Min-Wert		0	
		OC	Haarfarbe		
		A	Blond		
		A	Braun		
		A	Rot		
		A	Schwarz		



## Aufgabe 1: Lösung

▼ 1. Körperlicher Zustand

<b>Geburtsdatum</b> Day Month Year <input type="text"/> - <input type="text"/> - <input type="text"/> <input type="radio"/> Unknown	<b>Rauchen Sie?</b> <input type="radio"/> Yes <input type="radio"/> No <input type="radio"/> Unknown	<b>Sind Vorerkrankungen bekannt?</b> <input type="radio"/> Yes <input type="radio"/> No <input type="radio"/> Unknown
<b>Körperverzierungen?</b> <input type="checkbox"/> Tätowierung(en) <input type="checkbox"/> Piercing(s) <input type="checkbox"/> Keine <input type="radio"/> Unknown	<b>Körpergröße</b> <input type="text"/> m <input type="radio"/> Unknown	<b>Gewicht</b> <input type="text"/> kg <input type="radio"/> Unknown
<b>Augenfarbe</b> Please select... <input type="radio"/> Unknown	<b>Name</b> <input type="text"/> <input type="radio"/> Unknown	<b>Vorname</b> <input type="text"/> <input type="radio"/> Unknown
<b>Trinken Sie Alkohol?</b> <input type="radio"/> Yes <input type="radio"/> No <input type="radio"/> Unknown	<b>Taillenumfang</b> <input type="text"/> cm <input type="radio"/> Unknown	<b>Haarfarbe</b> <input type="radio"/> Blond <input type="radio"/> Braun <input type="radio"/> Rot <input type="radio"/> Schwarz <input type="radio"/> Unknown



## Erklärungs-Typen

- Erklärungs-Typen bieten die Möglichkeit, verschiedene Elemente zur weiteren Erklärung von Fragen-, A- und U-Typen zu verwenden.
- Die Anzahl der Erklärungselemente ist beliebig, solange sie alle hintereinander stehen.



## Erklärungs-Typen

1. EB (Erklärungsbild)
2. EH (Erklärungs-HTML-Text)
3. EL (Erklärungs-Link)
4. ET (Erklärungstext)

imbi



### Erklärungs-Typen: EB (Erklärungsbild)

- Durch den EB-Typ kann eine Bilddatei angezeigt werden.
- Inhalt der Fragen-Antworten-Spalte ist der Dateiname des Bildes.
- Das Bild muss in der Mediendatei als .zip Datei hochgeladen werden.

imbi



### Erklärungs-Typen: EB (Erklärungsbild)

- ProKEt skaliert das Bild nicht, sondern versucht es in exakter Auflösung anzuzeigen.

imbi



## Erklärungs-Typen: EB (Erklärungsbild)

Wissensbasis hochladen:

Wissensbasis wählen:  Beispiel\_Typen\_V3.xlsx

Mediendatei wählen:  Bier.zip

imbi



## Erklärungs-Typen: EH (Erkl.-HTML-Text)

- Mit dem „EH“-Typ kann ein HTML formatierter Text angezeigt werden.
- Dabei wird die gesamte Eingabe im Fragen-Antworten-Feld als HTML interpretiert.

imbi



## Erklärungs-Typen: EL (Erklärungs-Link)

- Durch den „EL“-Typ kann dem Nutzer ein anklickbarer Link aus der Fragen-Antworten-Spalte angezeigt werden.

imbi



## Erklärungs-Typen: ET (Erklärungstext)

- Mit dem Typ „ET“ kann ein unformatierter Text angezeigt werden, um das Verständnis der Frage, z.B. mit einem Beispiel, zu verbessern.



### Beispiel

Wenn	ID	Typ	Fragen-Antworten	Aktion	Layout
		U	Körperlicher Zustand		1
	ALK	JN	Trinken Sie Alkohol?		
		EB	Bier.jpg		
		EH	<span style="background-color: black; color: white;">Don't drink and drive</span>		
		EL	<a href="https://www.bier.de/">https://www.bier.de/</a>		
		ET	Auch Bier ist Alkohol!		
ALK=yes		OC	Wie oft trinken Sie Alkohol?		
		A	1 mal pro Monat		
		A	1 mal pro Woche		
		A	täglich		



### Beispiel


X

Don't drink and drive <https://www.bier.de/> Auch Bier ist Alkohol!

**Trinken Sie Alkohol?** ?

Yes

No

Unknown

**Taillenumfang**

cm

Unknown



## Sonstige Typen

- U (Untersuchung)
- A (Antwort)
- K und Kom (Kommentar)
- LZeile (Lösungszeile)
- L (Lösung)
- LB (Lösungsbewertung)
- RF-Regeln

imbi



## Sonstige Typen: U (Untersuchung)

- Eine Zeile des Typs „U“ dient der Untergliederung des Fragebogens.
- In der Aktionsspalte wird angegeben, ob die Untersuchung bereits zu Beginn des Dialogs angezeigt werden soll oder erst nachdem sie hergeleitet wurde.

imbi



## Sonstige Typen: A (Antwort)

- Eine Zeile mit einem „A“-Typ beschreibt eine Antwort auf eine vorangehende Frage.
- Es können Erklärungstypen dazu verwendet werden.

imbi



## Sonstige Typen: K und Kom (Kommentar)

- Mit den Typen „K“ und „Kom“ lässt sich eine Zeile definieren, die nicht ausgelesen wird, um Kommentare, zu benachbarten Zeilen, der Tabelle einfügen zu können.



## Sonstige Typen: K und Kom (Kommentar)

Wenn	ID	Typ	Fragen-Antworten	Aktion	Layout
		U	Körperlicher Zustand		1
		Text	Vorname		
		K	Folgender Abschnitt behandelt das Alkoholverhalten des Patienten		
ALK		JN	Trinken Sie Alkohol?		
		EB	Bier.jpg		
		EH	<span style="background-color: black;">Don't drink and drive</span>		
		EL	https://www.bier.de/		
		ET	Auch Bier ist Alkohol!		
ALK=yes		OC	Wie oft trinken Sie Alkohol?		
		A	1 mal pro Monat		
		A	1 mal pro Woche		
		A	täglich		
		Kom	Ende des Abschnitts Alkoholverhalten		
		Num	Taillienumfang		
		Einheit	cm		
		Min-Wert		0	



## Sonstige Typen: LZeile (Lösungszeile)

- In einer Zeile des Typs „LZeile“ können Lösungen/Diagnosen angegeben werden.
- Diese Lösungen/Diagnosen werden dann durch die *RH*, *H* und *HN*-Spalten hergeleitet.

Wenn	ID	Typ	Fragen-Antworten	Aktion	Layout	H	H
		LZeile				Suchtverhalten	Gesund und Fit
		LB					
		U	Körperlicher Zustand		1		
		Date	Geburtsdatum				



## Sonstige Typen: L (Lösung)

- Mit dem „L“-Typ können existierende Lösungen aus der LZeile referenziert werden.
- Dabei wird in der Fragen-Antworten-Spalte der Name, der gewünschten Lösung, gesetzt.
- L-Typen werden nach den Fragen-Antworten-Typen aufgelistet.



## Sonstige Typen: L (Lösung)

Wenn	ID	Typ	Fragen-Antworten	Aktion	Layout	H	H
		LZeile				Suchtverhalten	Gesund und Fit
		LB					
		U	Körperlicher Zustand	1			
		Date	Geburtsdatum				
		JIN	Rauchen Sie?			p5	n5
	VE	JINU	Sind Vorerkrankungen bekannt?				n3
VE=yes		Text	Was für Vorerkrankungen?				
	ALK	JIN	Trinken Sie Alkohol?				
		EB	Bier.jpg				
		EH	<span style="background-color: black; color: black;">Don't drink and drive</span>				
		EL	<a href="https://www.bier.de/">https://www.bier.de/</a>				
		ET	Auch Bier ist Alkohol!				
ALK=yes		OC	Wie oft trinken Sie Alkohol?				
		A	3 mal pro Monat			p2	
		A	3 mal pro Woche			p5	n3
		A	täglich			p6	n7
	S	JIN	Treiben Sie Sport				
S=yes		OC	Wie oft treiben Sie Sport?				
		A	Weniger als 3 mal pro Woche				p4
		A	3 bis 3 mal pro Woche				p6
		A	täglich				n7
		L	Suchtverhalten				n7
	X			BMI = ((G/K)/K)			
	D-SP					3	
	D-SSP				x		



## Sonstige Typen: LB (Lösungsbewertung)

- Durch diese Zeile können für verschiedene Regeln Bewertungen festgelegt werden, die eine Lösung/Diagnose bekommt.





## Sonstige Typen: LB (Lösungsbewertung)

### Mögliche Bewertungen:

Name	Entspricht	Name	entspricht
p7	999	n1	-2
p6	80	n2	-5
p5	40	n3	-10
p4	20	n4	-20
p3	10	n5	-40
p2	5	n6	-80
p1	2	n7	-999



## Sonstige Typen: LB (Lösungsbewertung)

- Je höher der Wert ist, desto sicherer ist die Diagnose.
- Bei einer positiven Bewertung von p6 und höher gilt die Diagnose als bestätigt
- Bei einer negativen Bewertung von n6 oder weniger gilt die Diagnose als widerlegt.



Notiz:

Zu Alle Punkte einer Lösung (positive wie auch negative) werden miteinander verrechnet.

Z.B. 2 mal p5 ergibt ebenfalls eine bestätigte Lösung.

## Sonstige Typen: LB (Lösungsbewertung)

Wenn	ID	Typ	Fragen-Antworten	Aktion	Layout	H	H
		L-Zeile				Suchtverhalten	Gesund und Fit
		LB					
		J	Körperlicher Zustand	1			
		Date	Geburtsdatum				
		JN	Rauchen Sie?			p5	n5
	VE	JNU	Sind Vorerkrankungen bekannt?				n3
VE=yes		Text	Was für Vorerkrankungen?				
	ALK	JN	Trinken Sie Alkohol?				
		FB	Bier.jpg				
		EH	<span style="background-color: black;">Don't drink and drive</span>				
		EL	https://www.bier.de/				
		ET	Auch Bier ist Alkohohl!				
	ALK=yes	OC	Wie oft trinken Sie Alkohol?				
		A	1 mal pro Monat			p2	
		A	1 mal pro Woche			p5	n1
		A	täglich			p6	n7
	S	JN	Treiben Sie Sport				
S=yes		OC	Wie oft treiben Sie Sport?				
		A	Weniger als 1 mal pro Woche				
		A	1 bis 3 mal pro Woche				p4
		A	täglich				p6
		L	Suchtverhalten				n7
		D-SP		3			
		D-SSP		x			



## Sonstige Typen: LB (Lösungsbewertung)

**1. Körperlicher Zustand**

**Geburtsdatum**  
 Day: 1, Month: 1, Year: 1995  
 Unknown

**Rauchen Sie?**  
 Yes  
 No  
 Unknown

**Sind Vorerkrankungen bekannt?**  
 Yes  
 No  
 Unknown

**Körperverletzungen?**  
 Tätowierung(en)  
 Piercing(s)  
 Keine  
 Unknown

**Körpergröße**  
 1,80  
 m  
 Unknown

**Gewicht**  
 80  
 kg  
 Unknown

**Trinken Sie Alkohol?**  
 Yes  
 No  
 Unknown

**Wie oft trinken Sie Alkohol?**  
 1 mal pro Monat  
 1 mal pro Woche  
 täglich  
 Unknown

**Treiben Sie Sport**  
 Yes  
 No  
 Unknown

**Wie oft treiben Sie Sport?**  
 Weniger als 1 mal pro Woche  
 1 bis 3 mal pro Woche  
 täglich  
 Unknown

**Solutions**

- Suchtverhalten (120)
- Gesund und Fit (-999)



## Sonstige Typen: RF-Regeln

- Durch „RF“-Typen lassen sich selbst Regeln schreiben.
- Die Bedingungen werden in die Fragen-Antworten-Spalte geschrieben.
- Die ausgeführte Aktion wird in die Aktion-Spalte geschrieben.



## Sonstige Typen: RF-Regeln

- Die Antwort der zuvor erstellten „Num-I“-Frage wird in der RF-Zeile berechnet.
- Zu beachten ist, dass die Berechnung und das Ergebnis dem Benutzer nicht gezeigt werden.

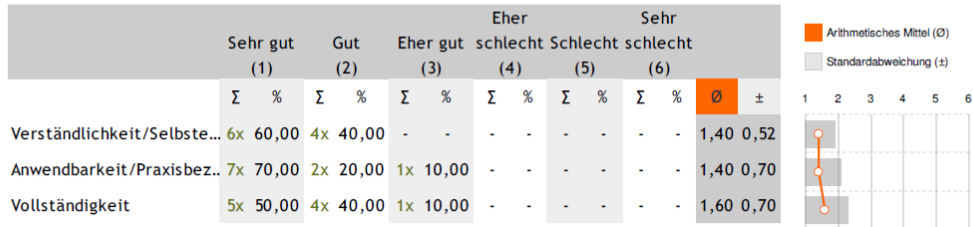
Wenn	ID	Typ	Fragen-Antworten	Aktion	Layout	H	H
	K	Num	Körpergröße				
	G	Num	Gewicht				
		Num-I	BMI				
		RF	Gewicht > 0 AND Körpergröße > 0	$BMI = ((G / K) / K)$			

# Anhang H: Rohdaten der Umfrage zur Erwartungsmessung

## Feedback des Präsentationskonzeptes

1. Wie würden Sie die Präsentation "Ontologie und Protégé: Installation" anhand der folgenden Punkte bewerten? \*

Anzahl Teilnehmer: 10



2. Sonstige Bemerkungen/Verbesserungsvorschläge?

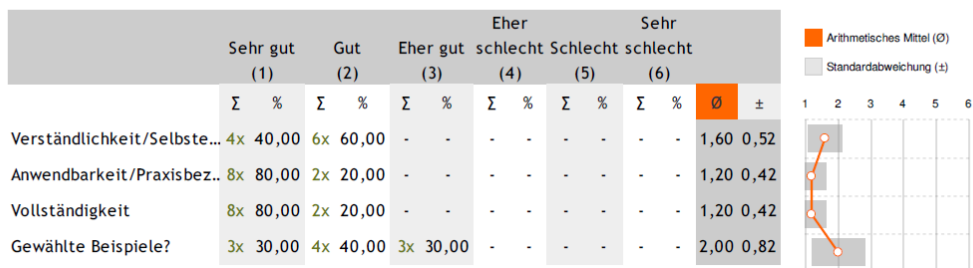
Anzahl Teilnehmer: 3

- Formatierung der Bilder(Folie Protégé: Installation Abschnitt 8)  
Folien nummerieren oft hilfreich
- Ich bin natürlich nicht sehr in der Materie vertieft. Allerdings denke ich, dass das verwendete Vokabular für einen Studenten, der in dem Bereich bereits Erfahrungen hat, die Folien sehr gut versteht. Die einzelnen Themenbereiche sind sehr gut erklärt.
- Ich würde statt der super ausführlichen Installationsanleitung noch Ontologie ein wenig genauer erklären. Zumindest für mich wars eher unverständlich, aber ich bin auch seit Jahren aus der Informatik raus :P

Dieser Schritt kann alternativ einfach übersprungen werden.  
"einfach" raus streichen.

3. Wie würden Sie die Präsentation "Protégé: Kernkonzepte" anhand der folgenden Punkte bewerten? \*

Anzahl Teilnehmer: 10



4. Sonstige Bemerkungen/Verbesserungsvorschläge?

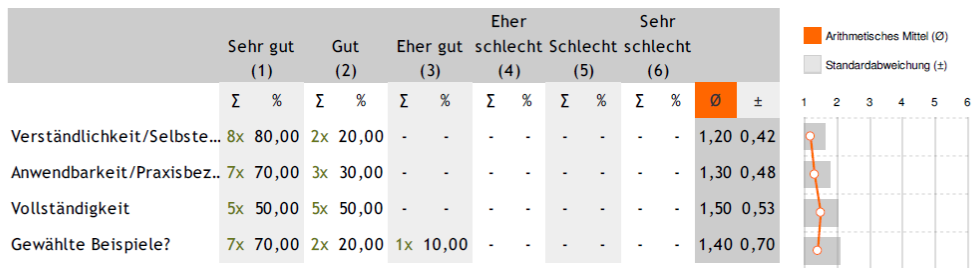
Anzahl Teilnehmer: 5

- erster Aufgabentext sehr erschlagend  
sehr detaillierte Schritte  
nachvollziehbare Beispiele
- Ich verstehe die Aufgabe "Ein Ort besitzt eine Bevölkerung ... .. kann an ein anderes Land grenzen." nicht wirklich. Was man machen genau machen soll.
- Aufgabe zu Klassen sehr schwierig für jemanden ohne entsprechendes Fachwissen
- Einzelne Funktionen sind gut beschrieben und mit Screenshots zur besseren Übersicht/Einblick versehen.
- S. 17 und 18 schwer erkennbar, was gemeint ist. Für eine Präsentation okay, allerdings erscheint mir das ganze mehr ein Tutorial zu sein und dafür würde ich die relevanten Informationen noch mehr hervorheben, vergrößern, farbig einkreisen, etc.

S19 und 20 und S47 läuft der Text unten in die Fußzeile rein.

5. Wie würden Sie die Präsentation "Protégé: Vertiefung" anhand der folgenden Punkte bewerten? \*

Anzahl Teilnehmer: 10



6. Sonstige Bemerkungen/Verbesserungsvorschläge?

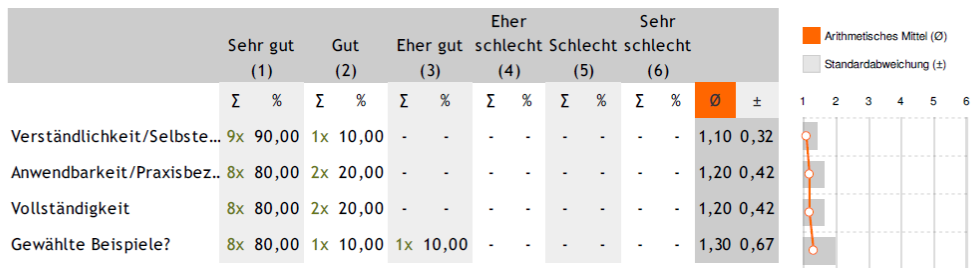
Anzahl Teilnehmer: 1

- S6, S25, S39 Text läuft in Fußzeile

S44 Bild hängt zu sehr im Titel drin

7. Wie würden Sie die Präsentation "ProKt: Allgemeines" anhand der folgenden Punkte bewerten? \*

Anzahl Teilnehmer: 10

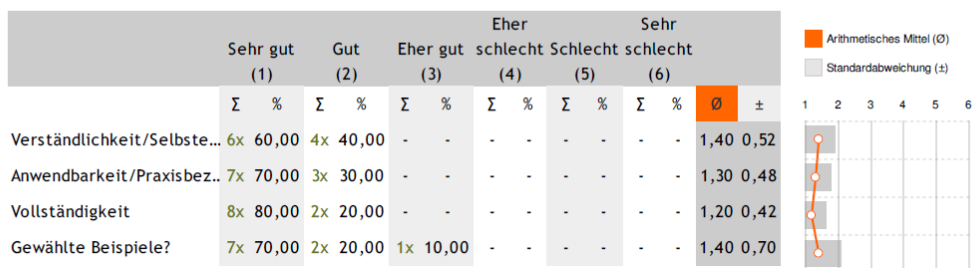


8. Sonstige Bemerkungen/Verbesserungsvorschläge?

Anzahl Teilnehmer: 0

9. Wie würden Sie die Präsentation "ProKt: Spalten" anhand der folgenden Punkte bewerten? \*

Anzahl Teilnehmer: 10

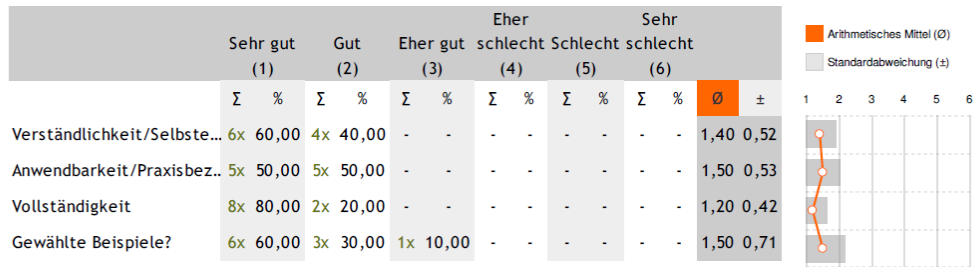


10. Sonstige Bemerkungen/Verbesserungsvorschläge?

Anzahl Teilnehmer: 0

11. Wie würden Sie die Präsentation "ProKEt: Typen" anhand der folgenden Punkte bewerten? \*

Anzahl Teilnehmer: 10



12. Sonstige Bemerkungen/Verbesserungsvorschläge?

Anzahl Teilnehmer: 1

- Es ist gern gesehen wenn am Ende eines Kapitels, bzw. dieser zwei Kapitel Protege und ProKEt, als letztes eine ganz kurze Zusammenfassung, auf ein zwei Folien verfasst wird, was man alles gelernt hat, nur zur kleinen Auffrischung.

## Literaturverzeichnis

- 1 <http://www.philolex.de/ontologi.htm> Last visited: 30.07.2018
- 2 <https://www.philosophie.uni-muenchen.de/fakultaet/schwerpunkte/ontologie/index.html> Last visited: 30.07.2018
- 3 <https://gi.de/informatiklexikon/ontologien/> Last visited: 30.07.2018
- 4 T.R.Gruber. A Translation Approach to Portable Ontologies. Knowledge Acquisition Band 5, Nr. 2, Academic Press, 1993, S. 199–220
- 5 N. Guarino, D. Oberle und S. Staab. What Is an Ontology?  
<https://userpages.uni-koblenz.de/~staab/Research/Publications/2009/handbookEdition2/what-is-an-ontology.pdf> Last visited: 30.07.2018
- 6 <https://protege.stanford.edu/about.php> Last visited: 30.07.2018
- 7 <https://protege.stanford.edu/> Last visited: 30.07.2018
- 8 Bruce G. Buchanan, Edward H. Shortliffe. Rule-Based Expert Systems. Addison-Wesley Publishing Company, 1984
- 9 Bruce G. Buchanan, Edward H. Shortliffe. Rule-Based Expert Systems. Addison-Wesley Publishing Company, 1984, S.3
- 10 Bruce G. Buchanan, Edward H. Shortliffe. Rule-Based Expert Systems. Addison-Wesley Publishing Company, 1984, Kapitel 14, S. 295ff
- 11 <https://pdfs.semanticscholar.org/211b/5110fd22bf23d54aa55d4a342eef676f2cf2.pdf> Last visited: 30.07.2018
- 12 <https://sourceforge.net/projects/proket/> Last visited: 30.07.2018
- 13 <http://www.is.informatik.uni-wuerzburg.de/en/research/proket-dialog-generator/> Last visited: 30.07.2018
- 14 <http://protege-project.136.n4.nabble.com/Best-way-to-manually-uninstall-Protege-ver-5-0-5-beta-17-td4667841.html#a4667900> Last visited: 30.07.2018
- 15 [https://protegewiki.stanford.edu/wiki/Protege4Views#Annotations\\_views](https://protegewiki.stanford.edu/wiki/Protege4Views#Annotations_views) Last visited: 30.07.2018
- 16 <http://protegeproject.github.io/protege/views/class-hierarchy/> Last visited: 30.07.2018
- 17 [https://www.w3.org/2007/OWL/wiki/Primer#Class\\_Disjointness](https://www.w3.org/2007/OWL/wiki/Primer#Class_Disjointness) Last visited: 30.07.2018

- 18 <http://protegeproject.github.io/protege/views/object-property-hierarchy/> Last visited: 30.07.2018
- 19 [https://www.w3.org/2007/OWL/wiki/Primer#Object\\_Properties](https://www.w3.org/2007/OWL/wiki/Primer#Object_Properties) Last visited: 30.07.2018
- 20 <http://protegeproject.github.io/protege/views/data-property-hierarchy/> Last visited: 30.07.2018
- 21 [https://protegewiki.stanford.edu/wiki/Protege4Views#Data\\_property\\_views](https://protegewiki.stanford.edu/wiki/Protege4Views#Data_property_views) Last visited: 30.07.2018
- 22 <http://protegeproject.github.io/protege/views/object-property-characteristics/> Last visited: 30.07.2018
- 23 [https://www.w3.org/2007/OWL/wiki/Primer#Property\\_Characteristics](https://www.w3.org/2007/OWL/wiki/Primer#Property_Characteristics) Last visited: 30.07.2018
- 24 [https://www.w3.org/2007/OWL/wiki/Primer#Property\\_Chains](https://www.w3.org/2007/OWL/wiki/Primer#Property_Chains) Last visited: 30.07.2018
- 25 <http://protegeproject.github.io/protege/class-expression-syntax/> Last visited: 30.07.2018
- 26 [https://protegewiki.stanford.edu/wiki/Using\\_Reasoners](https://protegewiki.stanford.edu/wiki/Using_Reasoners) Last visited: 30.07.2018
- 27 <http://protegeproject.github.io/protege/menus/> Last visited: 30.07.2018
- 28 [http://its-wiki.no/images/9/9e/Presentation\\_Java\\_Export.pdf](http://its-wiki.no/images/9/9e/Presentation_Java_Export.pdf) Last visited: 30.07.2018
- 29 [https://protege.stanford.edu/conference/2006/submissions/slides/AppDevelopmentTutorial\\_Part1.pdf](https://protege.stanford.edu/conference/2006/submissions/slides/AppDevelopmentTutorial_Part1.pdf) S.29ff Last visited: 30.07.2018
- 30 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/columns/Typ> Last visited: 30.07.2018
- 31 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/columns/Fragen-Antworten> Last visited: 30.07.2018
- 32 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/columns/ID> Last visited: 30.07.2018
- 33 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/columns/Wenn> Last visited: 30.07.2018



- 34 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/columns/Aktion>  
Last visited: 30.07.2018
- 35 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/columns/Layout>  
Last visited: 30.07.2018
- 36 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/OC> Last visited:  
30.07.2018
- 37 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/D-Header> Last vis-  
ited: 30.07.2018
- 38 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/D-SP> Last visited:  
30.07.2018
- 39 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/D-SSP> Last visited:  
30.07.2018
- 40 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/columns/RH> Last  
visited: 30.07.2018
- 41 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/columns/H> Last vis-  
ited: 30.07.2018
- 42 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/columns/HN> Last  
visited: 30.07.2018
- 43 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/Frage> Last visited:  
30.07.2018
- 44 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/Date> Last visited:  
30.07.2018
- 45 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/JN> Last visited:  
30.07.2018
- 46 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/JNU> Last visited:  
30.07.2018
- 47 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/MC> Last visited:  
30.07.2018
- 48 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/Num> Last visited:  
30.07.2018
- 49 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/Num-I> Last visited:  
30.07.2018

- 
- 50 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/OC> Last visited:  
30.07.2018
- 51 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/Text> Last visited:  
30.07.2018
- 52 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/EX> Last visited:  
30.07.2018
- 53 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/EB> Last visited:  
30.07.2018
- 54 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/EH> Last visited:  
30.07.2018
- 55 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/EL> Last visited:  
30.07.2018
- 56 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/ET> Last visited:  
30.07.2018
- 57 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/K> Last visited:  
30.07.2018
- 58 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/Kom> Last visited:  
30.07.2018
- 59 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/Lzeile> Last visited:  
30.07.2018
- 60 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/L> Last visited:  
30.07.2018
- 61 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/LB> Last visited:  
30.07.2018
- 62 <http://winc031.informatik.uni-wuerzburg.de/kbdocs/#/doc/RF> Last visited:  
30.07.2018

## Erklärung

Name: Groß Vorname: Kai Nico  
Matrikel-Nr.: 185320 Studiengang: MIB

Hiermit versichere ich, Kai Nico Groß, dass ich die Arbeit selbständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen wurden, sind in jedem Fall unter Angabe der Quelle kenntlich gemacht. Die Arbeit ist noch nicht veröffentlicht oder in anderer Form als Prüfungsleistung vorgelegt worden.

---

Ort, Datum

---

Kai Nico Groß