

4 Vorgehen

4.1 Projektphasen

Um bei der Arbeit ein möglichst strukturiertes Vorgehen zu verfolgen, wurden folgende Projektphasen gewählt:

- Erarbeitung und Festhalten der Anforderungen
- Erarbeitung der formalen und technischen Grundlagen
- Modellierung der Ontologie
- Erstellung der Dokumentation zur Wissensmodellierung
- Erarbeitung der praktischen Grundlagen
- Erstellung der abschliessenden Dokumentation

Dabei ist zu sagen, dass die Phasen der Modellierung der Ontologie sowie der Erstellung der Dokumentation zur Wissensmodellierung grösstenteils parallel abliefen bzw. Hand in Hand übergingen. Die während der Modellierung erhaltenen Erkenntnisse konnten direkt in die Dokumentation übernommen werden. Die bei der Erarbeitung des Dokumentes erarbeiteten theoretischen Grundlagen konnten im Gegenzug direkt für die praktische Modellierung genutzt werden.

4.1.1 Anforderungen

In der Projektphase der Anforderungen wurden die Requirments an die Arbeit erarbeitet. Ergebnis dieser Phase ist ein Anforderungsdokument mit den wichtigsten Eckpunkten, siehe Abschnitt 8. Es wurde ursprünglich festgehalten, dass der praktische Teil der Arbeit — die Erarbeitung einer Ontologie anhand einer Wissensdomäne — mittels eines konkreten Anwendungsproblems, der Erlernung der Programmiersprache Prolog, umgesetzt werden soll.

Im Laufe der Arbeit hat sich jedoch gezeigt, dass semantische Netze bzw. Ontologien nicht ein geeignetes Mittel sind um Logiksprachen wie z.B. Prolog abzubilden. Dies wird Kapitel Unterabschnitt 4.1.3 genauer beschrieben.

4.1.2 Formale und technische Grundlagen

Die formalen Grundlagen zur Modellierung der Ontologie sowie zur Erstellung der Dokumentation zur Wissensmodellierung waren teilweise bereits aufgrund des Vorprojektes — BTI7302 – Projekt 2, Details siehe Abschnitt 8 — vorhanden. Die restlichen formalen Grundlagen wurden anhand von [2], [3], [4], [5], [6], [7], [8] und [9] erarbeitet.

Bedingt durch die oben erwähnte Vorarbeit, war das Konzept der technischen Umsetzung bereits gegeben. Es sollte aus zwei Komponenten, einem Backend — in Form einer semantischen Datenbank, zur Verarbeitung von Anfragen an diese — sowie einem Frontend — zur einfachen Handhabung von Abfragen und der Ausgabe von Resultaten — bestehen.

Wie schon in Kapitel 3 erwähnt, fiel die ursprüngliche Wahl für das Backend auf Apache Stanbol ¹, da dies alle Anforderungen zu erfüllen schien. Es stellte sich jedoch im Verlaufe der Arbeit heraus, dass dies nicht da geeignete Produkt für die geplante Umsetzung ist, Details siehe Abschnitt 6.3.

¹<http://stanbol.apache.org>

4.1.3 Modellierung der Ontologie

Vorarbeit fürs Modellierung der Ontologie

Ursprünglich wurde versucht ein Modell der (Logik-) Programmiersprache Prolog zu erstellen. Zuerst geschah dies mittels dem klassischen Ansatz der Modellierung mittels der Modellierungssprache UML, da die Autoren über einen Hintergrund der objektorientierten Programmiersprachen verfügen. Es wurde jedoch schnell klar, dass es nicht ausreicht nur Klassen zu definieren, da eine Ontologie nur Beziehungen zwischen Individuen abbilden kann. Daher muss sie immer über Instanzen bzw. Individuen verfügen.

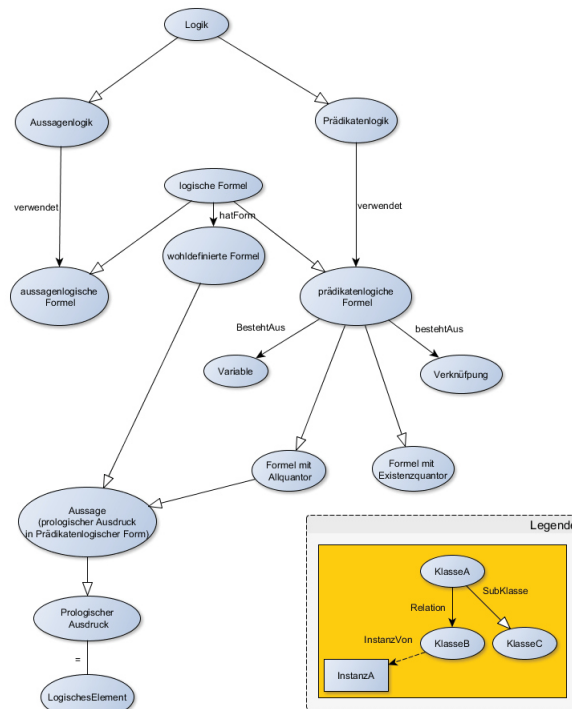


Abbildung 4.1: Vereinfachte Darstellung von Logik, rein mittels Klassen.²

Um auch Relationen abbilden zu können wurde die Modellierung schliesslich mit Individuen erweitert. Dies erlaubte die Definition von Relationen zwischen diesen, was sich als Schritt in die richtige Richtung erweisen sollte.

²Eigene Darstellung mittels yEd.

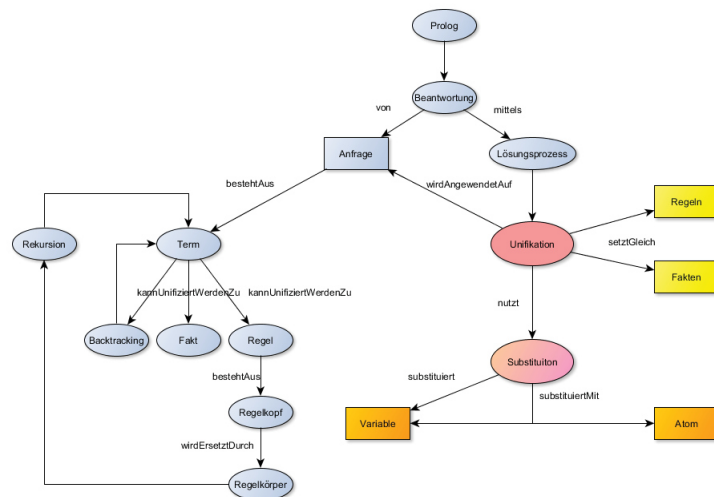


Abbildung 4.2: Vereinfachte Darstellung eines Teils des Lösungsprozesses von Prolog, mittels Klassen, Individuen und Relation.³

Es wurde dann jeweils versucht konkrete Fragen aufgrund der erstellten Ontologie zu beantworten, wodurch Mängel in der Ontologie relativ gut sichtbar wurden. Dies konnten dann schrittweise korrigiert werden, so dass die gewünschten Abfragen doch umgesetzt werden konnten. Details zu den gemachten Abfragen finden sich unter Abschnitt 8.

Schritt 3: Beziehungen des einfachen Token herausfinden

```
SELECT distinct * WHERE {
  ?i a owl:NamedIndividual.
  ?i a ?c.
  ?c ?xx ?z. # Gibt alle Beziehungen aus

  FILTER (regex(str(?i), "atom", "i") && ?c=p1:EinfachesToken)
}
```

i	c	xxx	z
Atom	EinfachesToken	type	Class
Atom	EinfachesToken	subClassOf	SprachElement

Feststellung: Wir sehen, dass einfache Tokens die Prädikate *type* und *subClassOf* mit den Subjekten *Class* bzw. *SprachElement* hat. Dies erlaubt nun die Einschränkung des Queries auf den Teil von Interesse, also *subClassOf*.

Abbildung 4.3: Beispiel einer Abfrage der Ontologie mittels SPARQL.⁴

Schnell wurde klar, dass solch eine Modellierung ins Uferlose gehen kann, woraufhin der Betreuer der Arbeit, Herr Dr. Eckerle, empfahl Literatur über Prolog als Grundlage bzw. Rahmen zu verwenden. Hierbei wurde auf das Buch *Künstliche Intelligenz* von U. Lämmel und J. Cleeve zurückgegriffen [10]. Daraus resultierte eine verbesserte Modellierung der Ontologie von Prolog, mit Klassen, Relationen und Individuen.

³Eigene Darstellung mittels yEd.

⁴Eigene Darstellung mittels Stanford Protégé und Google Docs.

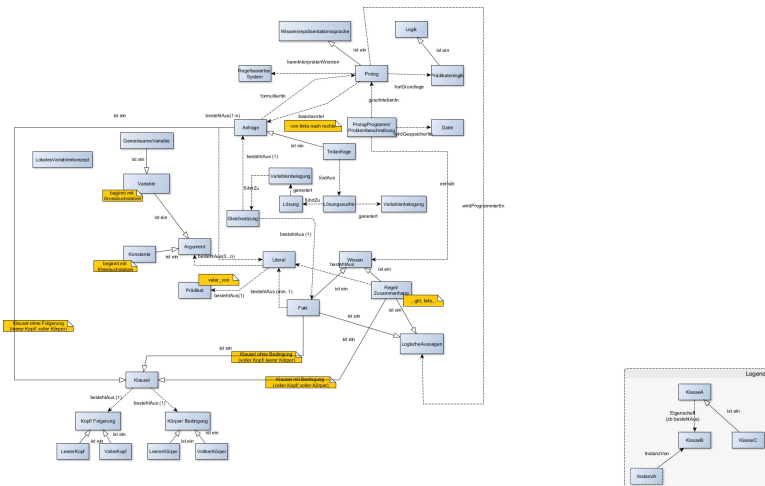


Abbildung 4.4: Vereinfachte Darstellung von Prolog, mit Klassen und Relationen (Individuen wurden der Übersicht halber bewusst weggelassen).⁵

Dies erlaubte jedoch nach wie vor keinen zusätzlichen Gewinn von Mehrwert in Form von Inferenz. Nach diversen Gesprächen mit dem Betreuer der Arbeit, Herrn Dr. Eckerle, stellte sich heraus, dass der Ontologie Regeln fehlen. Es wurde dann versucht mithilfe von Prolog das Konzept von Prolog abzubilden. Die Idee dahinter war, von der objektorientierten Denkweise loszukommen. Für die Autoren war es einfach nachzuvollziehen, dass und wie in Prolog Regeln verwendet werden. In Prolog wird auch schnell deutlich, dass eine Abfrage auf Fakten (ohne Regeln) keinen Mehrwert bringt.

Nach diversen, erfolglosen Versuchen Regeln zu finden, wurde dies zuerst anhand einfacher Modelle, des klassischen des Familienstammbaums, versucht, was schlussendlich auch gelang. Mit diesem simplen Beispiel wurde der Mehrwert einer Wissensmodellierung sogleich erkannt.

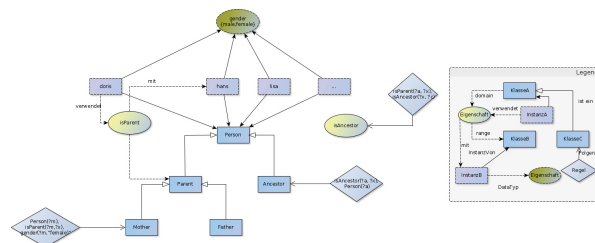


Abbildung 4.5: Darstellung eines einfachen Beispiels, der Abbildung einer Familie, mit Klassen, Individuen, Relationen und Regeln.⁶

Jedoch gelang es auch nach dieser Erfahrung nicht, Regeln für die eigentliche Wissensdomäne zu finden. Die Erkenntnis aus diesem Prozess ist schlussendlich, dass die Abbildung von Prolog in einer Ontologie zwar möglich ist, jedoch eher in lexikalischer Form, was den Vorteil der Inferenz zunichtemacht. So kann beispielsweise gesagt werden, dass Prolog Unifikation auf Anfragen anwendet und dabei Regeln mit Fakten gleichsetzt. Dabei wird Substitution genutzt, welche Variablen mit Variablen und/oder Atomen substituiert. Dies wurde aber Grösstenteils in Form von Fakten abgebildet. Einige wenige Regeln konnten erzeugt werden. Das damit abgebildete Wissen könnte aber auf eine einfachere und intuitivere Art mit Fakten abgebildet werden. Somit ist der Mehrwert der Regel nichtig.

⁵Eigene Darstellung mittels yEd.

⁶Eigene Darstellung mittels yEd.

Dies ergab eine noch sehr simple Ontologie, welche mit diesem semantischen Netz abgebildet werden kann:

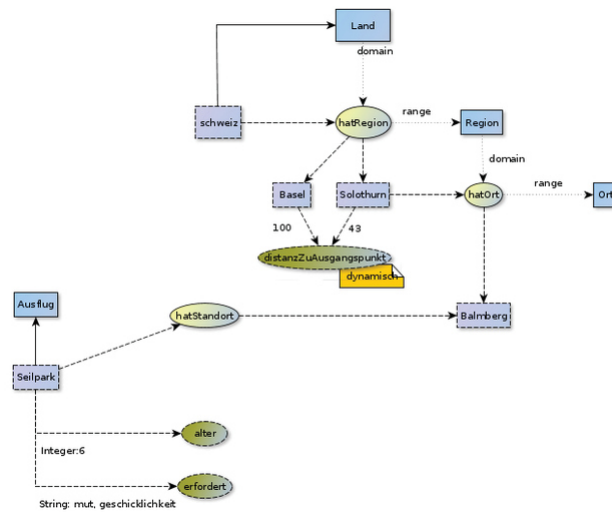


Abbildung 4.7: Semantisches Netz für den Tagesausflug von Familie Muster.⁸

Mit der neue gewählten Wissensdomäne wurde auch der Sinn hinter der Verwendung von Regeln deutlich:

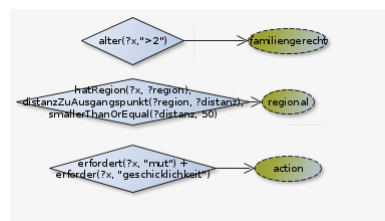


Abbildung 4.8: Regeln zum semantischen Netz für den Tagesausflug von Familie Muster.⁹

Aus der Modellierung konnten die folgenden Kriterien für die Abfrage abgeleitet werden:

- familiengerecht
- action
- regional

Mit der richtigen SPARQL-Abfrage wird der Familie Muster schliesslich der Seilpark in Balmberg vorgeschlagen.

```
SELECT
  *
WHERE {
  ?object :familiengerecht true;
  :regional true;
  :action true.
}
```

Listing 4.2: SPARQL-Abfrage um familiengerechte, regionale und actionreiche Ausflüge zu finden.

⁸Eigene Darstellung mittels yEd.

⁹Eigene Darstellung mittels yEd.

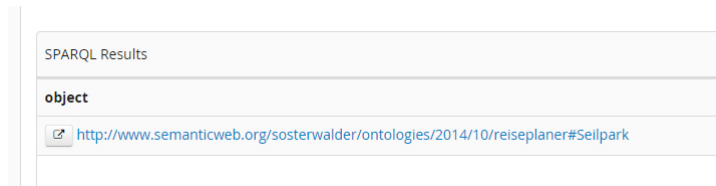


Abbildung 4.9: Ergebnis der Suche von Familie Muster.¹⁰

In diesem ersten, bewusst sehr einfach gehaltenen Anwendungsfall, wird die Herangehensweise der Autoren sichtbar. Im Laufe von weiteren Beispielen wurde die Ontologie stetig erweitert. So haben die Autoren zum Beispiel erst zu einem späteren Zeitpunkt eine Zeiteinheit eingeführt oder gemerkt, dass die Eigenschaft familiengerecht zu oberflächlich und pauschal betrachtet wurde. Diese wurde also dahingehend geändert, dass die Eigenschaft Untereigenschaften beinhaltet, welche unterscheiden in welchem Alter die Kinder sind. Die gesamte Ontologie wird im Kapitel **TODO genauer erläutert**.

4.1.4 Erstellung der Dokumentation zur Wissensmodellierung

Parallel zur Modellierung der Ontologie entstand eine Dokumentation des Vorgehens zur Wissensmodellierung. Diese zeigt exemplarisch auf, wie ein Knowledge-Engineer vorgeht, um eine Problemdomäne systematisch zu modellieren und formalisieren. Wie bereits zuvor erwähnt, wurde als Problemdomäne die Planung von Reisen gewählt. Die Dokumentation findet sich unter Abschnitt 8.

Aufbau des Tutorials

Im Gegensatz zu herkömmlichen Tutorials enthält das in dieser Arbeit erstellte einen grossen Theorieanteil. Aus diesem Grund wurde das Dokument in drei Aspekte aufgeteilt.

Da es sich hierbei um eine wissenschaftliche Arbeit handelt, haben die Autoren entschieden in einem erste Teil theoretischen Hintergrundwissen zur Wissensmodellierung bereitzustellen. So wird zum Beispiel erläutert, was Expertensysteme sind, wie diese graphisch dargestellt werden können und welche Schreibweisen respektive Sprachen verwendet werden um eine Ontologie abzubilden und darauf Abfragen zu stellen.



Durch diese Eule wird die zweite Herangehensweise im Dokument eingeleitet. In diesem Teil liegt der Fokus auf praktischen Tipps der Autoren, auf Erfahrungen, welche gemacht wurden und welche mit Mehrwissen hätten vermieden werden können. Die Autoren versuchen dem Leser so die Arbeit zu vereinfachen.



Der letzte Teil des Dokuments, welcher durch den Elefanten gekennzeichnet ist, verkörpert ein Tutorial im klassischen Sinne. Einem pragmatisch veranlagten Leser ist es so möglich, durch Folgen des Elefanten, innerhalb von kurzer Zeit ein simples Beispiel eines Expertensystems aufzubauen.

4.1.5 Erarbeitung der praktischen Grundlagen

Die Erarbeitung der praktischen Grundlagen war die Vorarbeit für die Erstellung des Tutorial-Dokuments. Dementsprechend wurden die gewonnen Erkenntnisse darin eingearbeitet.

¹⁰Eigene Darstellung mittels yEd.