



Programmentwurf 1

Aufgabenteil E_002

Dokumentation für Wissensbasierte Systeme im Studiengang Angewandte
Informatik

an der Dualen Hochschule Baden-Württemberg Stuttgart

von

Christian Burkard & Florian Kanngiesser

Juni 2012

Matrikelnummer, Kurs
Betreuer

4206853 & 7353550, TITAlA2010
Prof. Dr. Dirk Reichardt

Inhaltsverzeichnis

1	Einführung	1
1.1	Input File Format	1
2	Verfahren	2
2.1	Dempster-Shafer Theorem	2
3	Use Cases	3
4	Programmentwurf	4

1 Einführung

Anders als im Readme beschrieben haben wir uns für einen einzigen File als Input entschieden. Der Gründe dafür waren das im ersten Beispiel nur eine einzige Datei vorlag und wir deshalb unseren ursprünglichen Entwurf darauf ausgelegt hatten. Außerdem spielt es auch keine Rolle ob man ein oder zwei Dateien verwendet.

1.1 Input File Format

Das Input-File ist in 2 Sektionen unterteilt, welche mittels Semikolon voneinander getrennt sind. Die einzelnen Attribute werden mittels Komma separiert. Aus diesem Format sind wir nun in der Lage Primitive und Relationen zu generieren. Zuerst werden die Primitive ausgelesen und in eine geeignete Datenstruktur gespeichert, anschließend die Relationen.

```
1 re;13;0.1
2 ra,9,0.95
3 ra,10,0.9
4 ra,8,0.4
5 ;
6 inside,2,1
7 inside,12,5
8 inside,7,6
```

2 Verfahren

In diesem Kapitel wird das verwendete Verfahren noch einmal theoretisch beleuchtet.

2.1 Dempster-Shafer Theorem

Während des Auslesens des Input Files wird ein HashTable der einzelnen Primitiven gebildet. Sollte über ein Primitiv mehrere Aussagen vorliegen so wird nun zuerst die Gesamtevidenz über dieses Primitiv gebildet. Die Ausgangslage zur Evidenzbestimmung ist die Zuverlässigkeit der Aussage über ein Primitiv (Basismass). Nun wird für jede *inside* Relation die Evidenz mittels der Dempster-Shafer Regel zusammengefasst, also für Objekte mit mehreren Aussagen die Gesamtevidenz und für einzelne die gegebene Zuverlässigkeit.

Sollte es dabei zu einem Konflikt kommen wird zusätzlich noch ein Korrekturfaktor bestimmt. Anschließend wird der Believe über die verschiedenen Straßenschilder gebildet, Grundlage dafür sind die in der Aufgabenstellung beschriebenen Straßenschilder und die im Input File gegebenen Relationen.

Anschließend wird noch bestimmt ob gewisse Schilder übereinander *on top* liegen können. Dazu werden die *ontop* Relationen geprüft und die Objekte die bereits als Schild identifiziert wurden zusammen genommen. Mittels Dempster-Shafer Regel wird nun die Evidenz für die Relation bestimmt und falls vorhanden die Relation ausgegeben, also welches Schild über welchem Schild liegt. Abschließend wird die Plausibilität, ob wir ein Schild finden können zusammen mit dem Believe das wir ein Schild gefunden haben ausgegeben.

3 Use Cases

Um die Anwendung auszuführen muss lediglich der jar File ausgeführt werden mittels: `java -jar StreetSign-Evidence.jar filename`. Dabei ist Filename optional, sollte keine Datei angegeben werden so sucht die Anwendung nach einer Datei namens "testdata2.csv", ist diese Datei nicht vorhanden stürzt die Anwendung ab. Da die Anwendung ihre gesamte Ausgabe über die Konsole erledigt, sollte die Anwendung immer von der Kommandozeile ausgeführt werden.

Man kann der Anwendung immer genau eine Datei mit dem eingangs beschriebenen Format übergeben.

4 Programmentwurf

Auf einer abstrakte Ebene kann das Programm wie folgt beschrieben werden (vgl. Abbildung 4.1): Ein Parser liest die Daten der Classifier über das Bild aus einer CSV Datei aus und speichert die Daten der Relationen in den *Relation* Objekten und die Daten über die Primitive in den *Primitive* Objekten. Der Parser hält eine Liste all dieser Objekte. Die Anwendung holt sich die Daten der Relation Objekte. Jedes Relation Objekt enthält seine zwei zugehörigen Primitive Objekte. Dadurch werden alle Primitive, die nicht Teil einer Relation sind von nun an nicht weiter verwendet. Zunächst berechnet die Relation eine Tabelle mit der akkumulierten Evidenz zu jedem Primitiv. Danach generiert sie eine akkumulierte Tabelle mit der Evidenz aus den beiden Primitiven. Diese wird in einem *Solution* Objekt festgehalten. Nun kann ein *Relation* Objekt mithilfe seines zugehörigen *Solution* Objekt Schilder generieren.

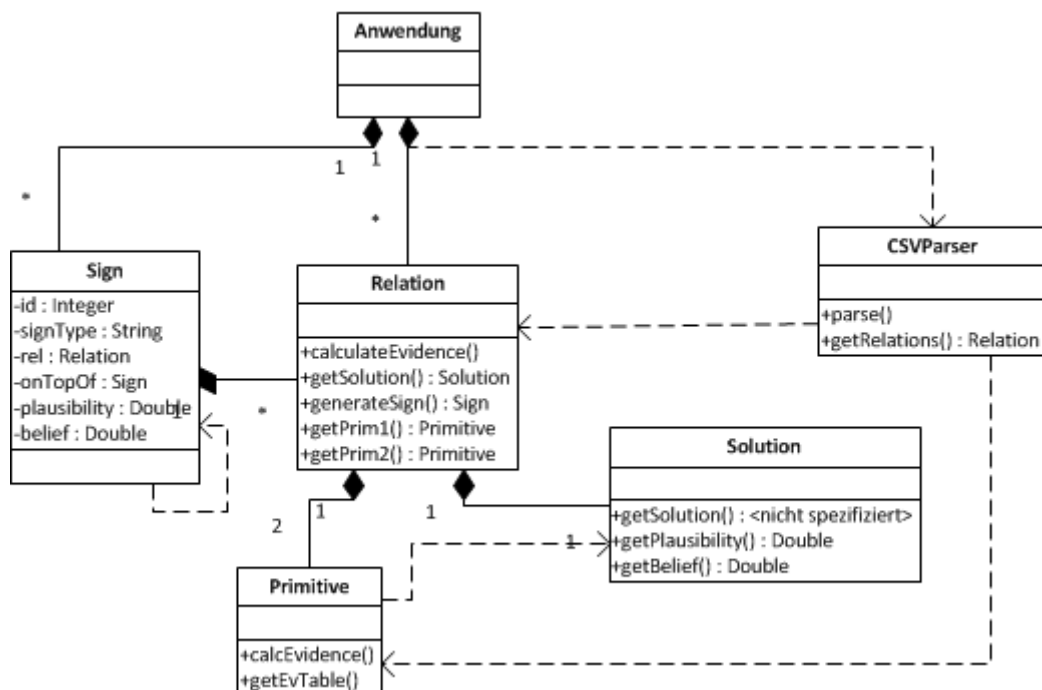


Abbildung 4.1: UML Diagram der StreetSign Evidence Anwendung

Ergibt sich daraus ein Schild, wird dieses in einer Liste gespeichert. Nachdem Relationen auf Schilder überprüft wurden, wird geprüft, ob eine Relation zwei Schilder in eine örtliche Relation zu bringen. Für jede *on top* Relation wird geprüft, ob deren erstes und zweites Primitiv ein gültiges Schild enthält. Wenn diese Bedingung erfüllt wird, merkt sich das örtlich obere Schild, dass und welches Schild sich unter ihm befindet. Am Ende wird über alle Schilder iteriert und ausführlich die bekannten Informationen zu den erkannten Schildern ausgegeben.