INF FAKULTÄT FÜR INFORMATIK

Clustering-Ontologie

Implementierung eines Java-Programms

Andy Koch, Stephan Besecke, Lars Grotehenne

Inhalt

- Anforderungen
- Aufgabenverteilung / Workflow
- Programm–Architektur
- Clustering–Turtle–File
- Funktionalitäten
- Anfragen & Hilfsanfragen
- Erweiterbarkeit & Verbesserungideen
- Live–Demo

Anforderungen

- Java–Konsolenprogramm zur Verarbeitung eines RDF–Turtle–Files und Sparql–Abfragen
- Erstellung eines RDF–Turtle–Files für Clustering Algorithmen4

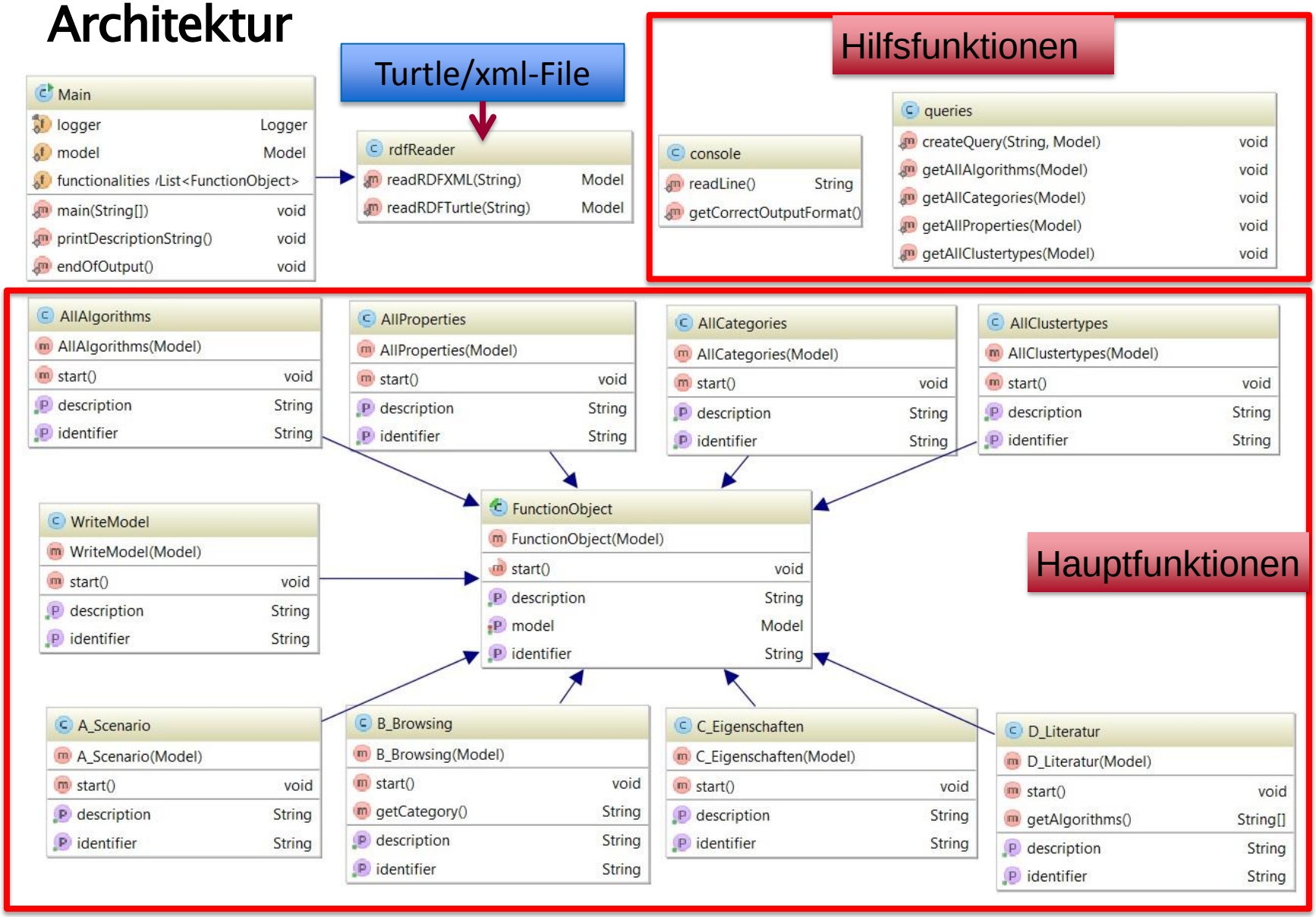
Bewältigung verschiedener Aufgaben

- A: Ausgabe von Clusteringalgorithmen zu einer Datenmenge mit bestimmten Eigenschaften (Szenario)
- B: Ausgabe von Clusteringalgorithmen zu einer bestimmten Kategorie
- C: Ausgabe von Clusteringalgorithmen zu bestimmten Eigenschaften
- D: Ausgabe von Papern zu ausgewählten Clusteringalgorithmen

Aufgabenverteilung / Workflow

1. Architektur & Implementierung Java-Programm (Lars)
2. Datensammlung Clustering (Stephan)
3. Clustering-Turtle-File (Stephan)
4. Erstellung erster Anfragen (Andy)
5. Optimierung Turtle-File (Andy & Lars)
6. Optimierung Anfragen & Hilfsanfragen (Lars)
7. Tests (Andy, Stephan, Lars)

Architektur



Clustering-Turtle-File (Auszüge): Algorithmen

```
#algorithms
:alg1    a                :Algorithm;
         rdfs:label       "DBScan";
         :has_category    :Density-based;
         :described_in    :paper1;
         :properties      [
                          :has_property    (:handle_noise);
                          :speed           (:normal);
                          :take_parameter (:neighborhood_size);
                          :can_cluster     (:encircled
                                            :not_convex
                                            :diff_dense);
                          ].
```

Clustering-Turtle-File (Auszüge): Eigenschaften

```
#properties and values
:take_parameter     a            rdf:predicate,
                                 :Algo-property;
                    dc:title     "take_parameter";
                    :has_values ( :neighborhood_size :bandwidth :num_clusters
                                  :damping :sample_preference :linkage_type :distance ).

:can_cluster        a            rdf:predicate,
                                 :Algo-property;
                    dc:title     "can_cluster";
                    :has_values ( :encircled :not_convex :diff_dense :bridget ).

:has_property       a            rdf:predicate,
                                 :Algo-property;
                    dc:title     "has_property";
                    :has_values  ( :handle_noise :multi_dim ).

:speed              a            rdf:predicate,
                                 :Algo-property;
                    dc:title     "speed";
                    :has_values ( :very_fast :fast :normal :slow :very_slow ).
```

Clustering-Turtle-File (Auszüge): Kategorien

```
#categories
:Density-based      a           :category;
                    dc:title    "Density-based".
:Delauney-based     a           :category;
                    dc:title    "Delauney-based".
:Centroid-based     a           :category;
                    dc:title    "Centroid-based".
:Connectivity-based a           :category;
                    dc:title    "Connectivity-based".
:Hierarchical       a          :category;
                    a           :Connectivity-based; # synonym to connectivity...
                    dc:title    "Hierarchical".
:Distribution-based a           :category;
                    dc:title    "Distribution-based".
:Subspace-based     a           :category;
                    dc:title    "Subspace-based".
```

Clustering-Turtle-File (Auszüge): Literatur

```
#paper
:paper1     a               :Paper;
            dc:source       "https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf";
            dc:title        "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise".
            #dc:description  "Abstract sehr lang...".
            # year          1996

:paper2     a               :Paper;
            dc:source       "http://link.springer.com/chapter/10.1007/978-3-319-09259-1_6";
            dc:title        "Density Based Clustering: Alternatives to DBSCAN".

:paper3     a               :Paper;
            dc:source       "http://www.loria.fr/~berger/Enseignement/Master2/Exposes/meanShiftCluster.pdf";
            dc:title        "Mean Shift, Mode Seeking, and Clustering".
            #dc:description  "Ein langer Text".
            # year          1995
```

Funktionalitäten

- Modell ausgeben (Turtle, XML, N-Triple, N3, JSON, RDF JSON)
- Grundlegende Ausgaben:
  - Kategorien, Algorithmen, Eigenschaften, Clustertypen
- A: Scenario: Auswahl von einem oder mehreren Clustertypen
  -> Ausgabe aller passenden Algorithmen
- B: Browsing: Auswahl einer Kategorie
  -> Ausgabe aller passenden Algorithmen
- C: Eigenschaften: Auswahl mehrerer Eigenschaften und deren Werte
  -> Ausgabe aller passenden Algorithmen
- D: Literatur: Auswahl von einem oder mehreren Algorithmen
  -> Ausgabe aller zugehörigen Paper

Hilfsanfragen

Gibt alle Clustertypen zurück

```
"PREFIX dc: <http://purl.org/dc/elements/1.1/>" +
        "PREFIX : <http://cluster.info#>" +
        "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>" +
        "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>" +
        "SELECT ?Clustertyp " +
        "WHERE {" +
        ":can_cluster :has_values/rdf:rest*/rdf:first ?Clustertyp." +
        "}";
```

Gibt alle Kategorien zurück

```
"PREFIX dc: <http://purl.org/dc/elements/1.1/>" +
        "PREFIX : <http://cluster.info#>" +
        "SELECT ?Kategorie " +
        "WHERE {" +
        "       ?a a :category." +
        "?a dc:title ?Kategorie. " +
        "       }";
```

Hilfsanfragen

Gibt alle Eigenschaften & Werte zurück

```
"PREFIX dc: <http://purl.org/dc/elements/1.1/>" +
"PREFIX : <http://cluster.info#>" +
"PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>" +
"PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>" +
    "SELECT ?Eigenschaft ?Wert " +
    "WHERE {" +
      "?property a :Algo-property." +
      "?property dc:title ?Eigenschaft." +
      "?property :has_values/rdf:rest*/rdf:first ?Wert." +
    "}";
```

Gibt alle Algorithmen zurück

```
"PREFIX dc: <http://purl.org/dc/elements/1.1/>" +
      "PREFIX : <http://cluster.info#>" +
      "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>" +
      "SELECT ?Algorithmus " +
      "WHERE {" +
      "?algo rdfs:label ?Algorithmus." +
      "}";
```

A: Algorithmen nach Szenarien / Clustertypen

```
queries.getAllClustertypes(model);
System.out.println("Einen oder mehrere Clustertypen eingeben, getrennt" +
        " mit Komma, ohne Leerzeichen, z.B. 'encircled,not_convex'");
String input = console.readLine();
String types[] = input.split(",");
String filter = "";
for(String type : types) {
    filter = filter + " :" + type;
}
    String queryString =
            "PREFIX dc: <http://purl.org/dc/elements/1.1/>" +
            "PREFIX : <http://cluster.info#>" +
            "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>" +
            "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>" +
            "PREFIX  list: <http://jena.hpl.hp.com/ARQ/list#>" +
            "SELECT ?Algorithmus " +
            "WHERE {" +
                "?algo rdfs:label ?Algorithmus." +
                "?algo :properties ?props." +
                "?props :can_cluster ?list." +
                 "filter not exists {" +
                    "values ?value { "+filter+" }" +
                    "filter not exists {" +
                        "?list rdf:rest*/rdf:first ?value" +
                    "}" +
                "}" +
            "}";

    queries.createQuery(queryString, model);
```

B: Algorithmen nach Kategorien

```
String category = getCategory();

String queryString =
        "PREFIX dc: <http://purl.org/dc/elements/1.1/>" +
        "PREFIX : <http://cluster.info#>" +
        "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>" +
        "SELECT ?Algorithmus " +
                "WHERE {" +
                "?algo :has_category ?category." +
                "?category dc:title ?categoryname." +
                "FILTER(?categoryname = '"+category+"')." +
                "?algo rdfs:label ?Algorithmus." +
                "}";

queries.createQuery(queryString, model);
```

C: Algorithmen nach Eigenschaften und Werten

```
queries.getAllProperties(model);
System.out.println("Bitte Eigenschaften und Werte angeben, z.B.: 'speed=normal,has_property=handle_noise'");

String properties = console.readLine();
String propertyArray[] = properties.split(",");

String queryString =
        "PREFIX dc: <http://purl.org/dc/elements/1.1/>" +
                "PREFIX : <http://cluster.info#>" +
                "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>" +
                "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>" +
                "PREFIX  list: <http://jena.hpl.hp.com/ARQ/list#>" +
                "SELECT ?Algorithmus " +
                "WHERE {" +
                "?algo rdfs:label ?Algorithmus." +
                "?algo :properties ?props.";


for(int i=0; i<propertyArray.length; i++) {
    String propValue[] = propertyArray[i].split("=");

    queryString=queryString +
            "{?props :"+propValue[0]+" ?list"+i+"." +
            "filter not exists {" +
            "values ?value { :"+propValue[1]+ " }" +
            "filter not exists {" +
            "?list"+i+" rdf:rest*/rdf:first ?value" +
            "}" +
            "}}";
}
queryString=queryString+"}";
queries.createQuery(queryString, model);
```

D: Paper zu einem oder mehreren Algorithmen

```
String[] algorithms = queries.getAlgorithms(model);

String filterOptions = "";
if(algorithms.length > 0) {
    filterOptions = "?Algorithmus = '"+algorithms[0]+"'";
    for(int i=1; i<algorithms.length; i++) {
        filterOptions = filterOptions + " || ?Algorithmus = '"+algorithms[i]+"'";
    }
}
String queryString =
        "PREFIX dc: <http://purl.org/dc/elements/1.1/>" +
        "PREFIX : <http://cluster.info#>" +
        "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>" +
            "SELECT ?Algorithmus ?Paper " +
            "WHERE {" +
               "?algo rdfs:label ?Algorithmus." +
               "FILTER("+filterOptions+")." +
               "?algo :described_in ?paper." +
               "?paper dc:title ?Paper." +
            "}";

queries.createQuery(queryString, model);
```

Erweiterbarkeit

Folgendes unterstützt das Programm und die Anfragen

- Problemloses hinzufügen eigener Funktionen / Anfragen (Erben vom FunctionObject)
- Hinzufügen von Algorithmen, Papern, Eigenschaften & Werten, Kategorien, Clustertypen
- Auswechseln des RDF–Turtle–Files, wahlweise auch ersetzen mit RDF–XML–File

Verbesserungsideen

- Usability
  - Auswahlmöglichkeiten
  - Groß–Kleinschreibung, Rechtschreibung
  - User–Interface
  - Struktur
  - Sicherheitsabfragen
- Kategorien verschachtelt (z.B. Density– & Delauney–Based)
- Algorithmen zu mehreren Kategorien finden
- Abfragen auslagern –> Programm das sich ein Turtle–File und zugehörige Abfragen lädt

Live-Demo