Report on the ISOcat project

Marc Kemps-Snijders

Menzo Windhouwer

Peter Wittenburg

Sue Ellen Wright

Overview

1. DCR implementation
2. DCR organization
3. DCR output formats and applications
4. DCR neighbourhood planning

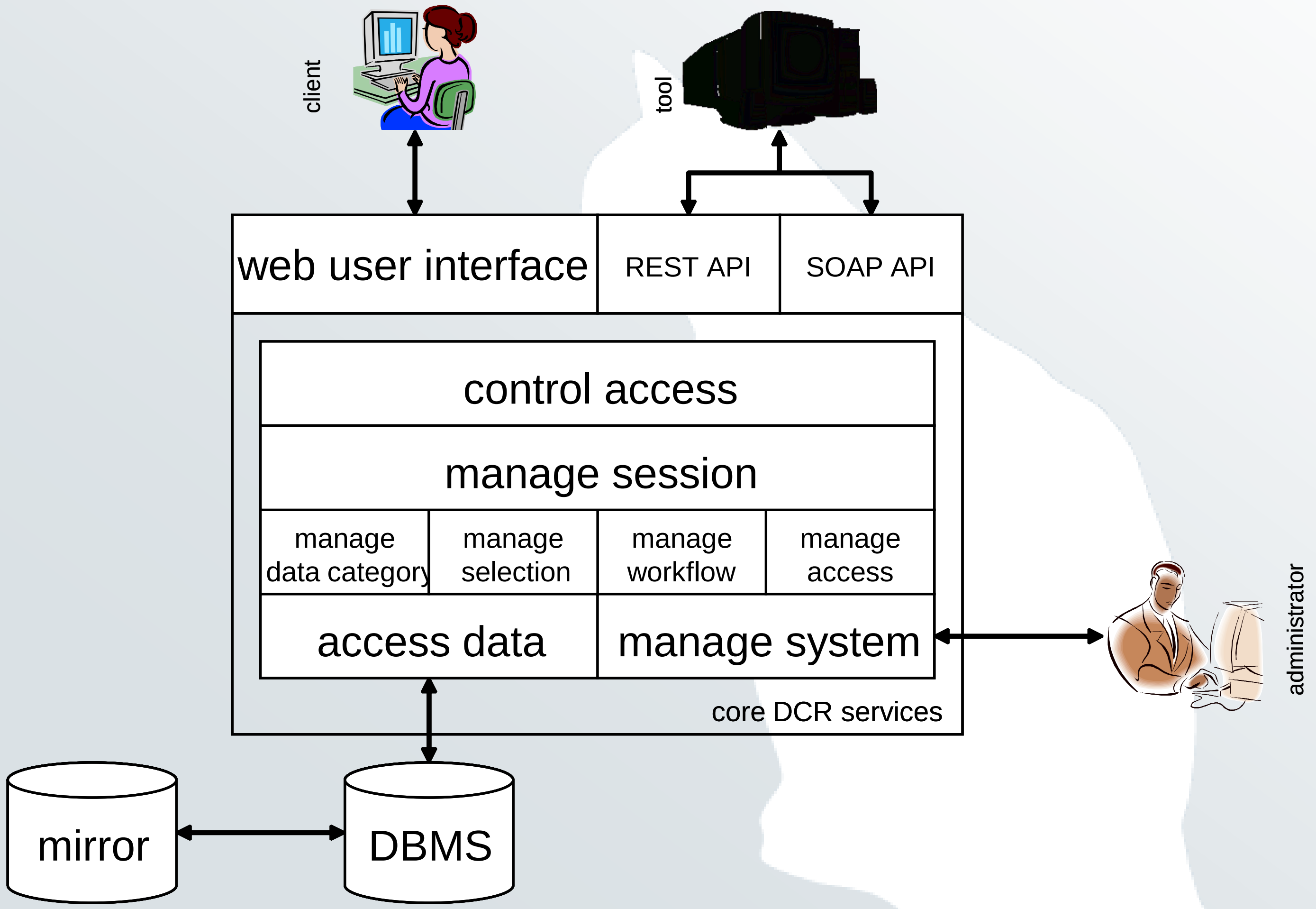
Overview

1. DCR implementation
   - ISOcat introduction
   - Demonstration & tutorial
   - Planning
2. DCR organization
3. DCR output formats and applications
4. DCR neighbourhood planning

ISOcat introduction

- ISOcat is
  - the reference implementation of ISO 12620:2009
  - the DCR implementation to be used by TC37
  - the successor to SYNTAX
- ISOcat provides
  - a state-of-the-art web user interface
  - a RESTful Application Programming Interface

ISOcat system architecture



ISOcat demonstration

What is missing?

- Standardization workflow
- Share DC(S)s
  - DC(S) locking mechanism
- Commenting DC(S)s
  - embed existing forum service
- Search private DCs
  - generate list of all accessible DCs for a user
- Advanced query interface
  - also to be used to implement a user’s own checking rules
- Storing user preferences
- …

- Focus was/is on functionality, not yet on performance

ISOcat

ISOcat Planning-1

1. Fix SYNTAX import process
2. Actual migration from SYNTAX to ISOcat
   - Warn users and shut down SYNTAX
   - Get fresh data dump from SYNTAX
   - Actual data cleanup starts.
3. TDGs use ISOcat
   - TDG's must validate existing data categories
   - Add missing functionality (e.g., sharing DC(S)s)
   - Report and fix any bugs
4. Supporting TDGs and implementing the standardization workflow
   - Finish ISO 12620:2009
5. Open up ISOcat to the general public

ISOcat Planning-2

6. Initiate TDG reappointment process via SC 3 secretariat
7. Reappoint TDG members
   - Dependent on people signing up via the survey
8. Embedding ISOcat in its neighbourhood
   - Mirror sites
   - Open source environment
   - Other registries

ISOcat tutorial

- Caveats:
  - Firefox 3 has been the most stable, however, is poorly interactive with larger profiles/DCSs
  - In the past Internet Explorer 7 became instable after some time, however, I didn't experience the problem for a while now
  - Don't edit one Data Category concurrently (by logging in twice), as this may lead to lost updates
  - Concurrently generating large DCIF documents has been reported to be problematic (but we may try …)
- Problems or ideas:
  - Help facility will be setup in the form of a forum
  - File a bug report/feature request at:

    http://sourceforge.net/tracker/?group_id=244572

ISOcat tutorial

Discussion

- What functionality do you miss?

Overview

1. DCR implementation
2. DCR organization
   - TDG organization
   - Guidelines for DC specifications
   - Procedures for reviewing existing DC specifications
   - Procedures for adding new DC specifications
3. DCR output formats and applications
4. DCR neighbourhood planning

TDGs and activities

TDG 1: Metadata

TDG 2: Morphosyntax

TDG 3: Semantic Content Representation

Activity 1: Discourse Relations

Activity 2: Dialogue Acts

Activity 3: Referential Structures and Links

Activity 4: Logico-semantic Relations

Activity 5: Temporal Entities and Relations

Activity 6: Semantic Roles and Argument Structures

TDG 4: Syntax

TDG 5: Machine Readable Dictionary

TDG 6: Language Resource Ontology

TDG 7: Lexicography

TDG 8: Language Codes

TDG 9: Terminology

Activity 1: General Principles

Activity 2: Concept Modeling

Activity 3: ISO Terminology Entries

Activity 4: Benchmarking Terminology

Activity 5: Terminology Management

Activity 6: TBX

Activity 7: TBX-Basic

Activity 8: Other TBX/TMLs

Activity 9: Geneter

Activity 10: TMS

TDG 11: Multilingual Information Management

TDG 12: Lexical Resources

TDG 13: Lexical Semantics

TDG 14: Source Identification

TDGs and activities in ISOcat

- In ISOcat each TDG has been created
- Each TDG owns a profile with the same name
- For each Activity we can create
  - An (ad-hoc) group of experts
  - An (public) DCS (owned by the TDG)
  - An profile related to the TDG

ISOcat

TDG Authorization

- Current TDGs have been officially created by resolutions passed in their respective SC plenaries.
- Theoretically, TDGs could also be created at the TC level, although 12620 does not explicitly provide for this.
- Current TDGs shall be reconfirmed reconstituted after the Tilburg meetings.
- Members will be officially reappointed.

Ramifications

- TDG chairs *SHALL (MUST, HAVE TO!)* fill in a description for their TDGs in the TDG survey.
- Individuals need to indicate their continued interest in working with their assigned TDG so that their SCs will reappoint them.

Guidelines for DC specifications

- English "self name" and mnemonic identifier
  - DCR Guidelines
  - Set XML rules
  - XML best practices for names
- Definitions
  - ISO 704 best practices for writing rigorous definitions
  - ISO 12620 presentational style as compared to ISO 704 terminology style
  - Defining data category concepts
  - Avoiding tautologies within definitions and with respect to data element names
  - Coordinating definitions for shallow concept systems (closed DCs + their value domains)
  - Finding coordinate data categories in other TDGs and proposing harmonization strategies

ISOcat

Procedures for reviewing existing DCs

- Select small DCSs grouping closely related DC specifications together (such as a closed DC + the simple DCs in its value domain).
- Review the DC names to ensure that they following proper naming rules and guidelines. Enter the name in the English Language Section.
- Provide the obligatory **+note** in the English Language Section.

Reviewing existing DCs-2

- Check the definitions for:
  - Proper definition form
  - Consistency among simple DC definitions for simple DCs dependent on the same closed DC
  - Absence of internal tautology or repetition of terms from the DC name
  - Consistency with definitions for the same basic DC defined by other TDGs
  - Possibilities for harmonization among similar DC specifications

Switching from SYNTAX to ISOcat

- Import issues
  - Sometimes the type of the DC has to be guessed.
    - Lack of explicit field code for DC type in SYNTAX data
    - A DC typed as open may actually be a closed or a simple DC
  - Some “bogus” DC specs need to be weeded out.
- Cleanup process
  - Check result:
    - Most of the time the now mandatory English note is missing
    - Demote some of these errors?
  - Standardized DCs can’t be edited:
    - Reassign them to an expert, and fast track them later through the standardization process using change requests?

Discussion

- How do you envision the switch from SYNTAX to ISOcat?

Overview

1. DCR implementation
2. DCR organization
3. DCR output formats and applications
   - Embedding DC persistent identifiers (PIDs) in schemata and other resources
   - DCS-based templates for schemata/resources:
     - XML Schema/Relax NG, RDF(S)/OWL, ...
     - DCIF-based stylesheets: constructing ISOcat XSLT stylesheets plug-ins to generate other schema/resource templates, e.g. TBX-based templates
4. DCR neighbourhood planning

DC Persistent Identifiers

- The DCR provides ‘cool URIs’ to the data category specifications

  http://www.isocat.org/datcat/DC-1708

  For more information on cool URIs, see http://www.w3.org/Addressing/
- The Registration Authority of ISO 12620.2009 guarantees the persistence of these URIs.
- The non-mnemonic syntax of the URIs was chosen to meet the requirements of PID frameworks, and to prevent ‘semantic rot’.
- The ‘DC-’ prefix is used for private DCs, while the ‘ISO-DC-’ prefix is used for standardized DCs.

ISOcat

Data Category PIDs

- To be able to leverage the power of the DCR, linguistic resources should now be annotated with these DC PIDs.
  - In general the PIDs will be embedded in the schema of the resource.
  - The desired result is to ensure server-side resolution of the PID and delivery of the actual content of the referenced DC specification.

Embedding DC PIDs – built in

- Some schema languages have built-in facilities to embed the PIDs
  - ODD

```
<elementSpec ident="pos">
  <equiv name="partOfSpeech"
         uri="http://www.isocat.org/dc/ISO-DC-1345"/>
    <!-- additional specifications here -->
</elementSpec>
```

  - XCS (only complex DCs)

```
<datCatSet>
  <termNoteSpec name="animacy
      datcatId="http://www.isocat.org/dc/ISO-DC-78">
    <contents datatype="picklist" forTermComp="yes">
       animate inanimate otherAnimacy
    </contents>
  </termNoteSpec>
</datCatSet>
```

Embedding DC PIDs – DC Reference

- The DC Reference XML vocabulary can be used to annotate schemas or resources without built in facilities:
  - Relax NG:

```
<element name="identifier"
       dcr:datcat="http://www.isocat.org/datcat/DC-8">
  <data type="string"/>
</element>
```

  - XML Schema:

```
<xs:element name="identifier">
  <xs:annotation>
    <xs:appinfo>
      <dcr:datcat pid="http://www.isocat.org/datcat/DC-8"/>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
```

ISOcat

Embedding DC PIDs - RDF

- RDF has its own DC Reference statement:

```
<rdf:Property rdf:about="http://www.isocat.org/ns/dcr.rdf#datcat">
  <rdfs:subPropertyOf
    rdf:resource="http://www.w3.org/2002/07/owl#sameAs"/>
</rdf:Property>
```

- To be used to annotate a RDF resource:

```
<rdf:Description rdf:about="http://example.com/app/myId">
  <dcr:datcat rdf:resource="http://www.isocat.org/datcat/DC-8"/>
  <rdfs:label xml:lang="en">Identifier</rdfs:label>
</rdf:Description>
```

Note: no choice has been made yet for the resource to be a RDF class or property

DCS-based templates

- To encourage the embedding of DC PIDs the DCR supports various export templates which can be instantiated for a specific DCS:
  - DCIF (implemented)
  - Basic RDF (implemented)
  - Relax NG (planned)
  - XML Schema (planned)
  - OWL (planned)
  - XCS (planned)
  - ODD (planned)
  - …
- Notice: in most cases this will result in a template which the user can download and has to complete further by putting the data categories in their application specific context
- Later ISOcat may support uploading an annotated schema and check its validity against the DCR (as long as the used patterns are recognizable)

Alternative RDF/OWL patterns

```
:headword
        dcr:datcat <http://www.isocat.org/datcat/DC-258> ;
        rdfs:label "head word"@en ;
        rdfs:comment "A lemma heading a dictionary entry."@en ;
        rdfs:label "lemma"@nl ;
        rdfs:comment "Het eerste woord van een artikel in een
                        woordenboek."@nl .

:partOfSpeech
        dcr:datcat <http://www.isocat.org/datcat/DC-396> ;
        rdfs:label "part of speech"@en ;
        rdfs:comment "A category assigned to a word based on its grammatical and
                semantic properties."@en .
```

DCs become either a class or property:

```
:headword a rdfs:Class .

:partOfSpeech a rdf:Property ;
        rdfs:domain :headword .
```

DCs become classes:

```
:headword a rdfs:Class .

:partOfSpeech a rdf:Class.

:hasPartOfSpeech a rdf:Property ;
        rdfs:domain :headword
        rdfs:range :partOfSpeech .

:noun a partOfSpeech .
```

DCIF-based plug-ins

- The DCS export formats are based on the DCIF export of a DCS
- Some export formats may require the user to make some choices between various possible patterns:
  - OWL: will the DC be a property or a class?
  - OWL: how will the value domain of a complex DC be mapped?
  - XCS: on which level should the DC appear?
  - XSD/RNG: which name in which language to use for a value (simple DC)
  - …
- A plug-in system is under development to support this, which will allow to store these choices together with a DCS
  - global and local (DC specific) properties
  - XSLT 1 or 2 stylesheets stored in ISOcat or accessed remotely
  - remote procedure call
  - …

Requirements for DCR plug-ins

- Generated templates should faithfully represent the relationships in the DCR:
  - between complex and simple DCs (value domains)
  - (optionally?) between simple DCs (is-a relationships)
- When possible also constraints should be supported
  - embed constraints in a fitting rule language
    - OWL plug-in: SWRL
    - RNG: schematron
    - …

Discussion

- What export formats do you miss?

Overview

1. DCR implementation
2. DCR organization
3. DCR output formats and applications
4. DCR neighbourhood planning
   - Mirrors of the TC37 DCR
   - Separate DCR instances (e.g., TBX meta-data categories)
   - Other types of registries (e.g., relation registries)

Mirrors of the TC37 DCR

- Several instances of ISOcat TC37 instance will be (virtually) sharing the same database
- Mirrors intend to be created at
  1. MPI - The Netherlands
  2. KAIST - Korea
  3. MITRE – US (?)
  4. BRANDEIS – US (?)
- The idea is that databases will be coupled using a PostgreSQL replication mechanism, e.g., Slony-I

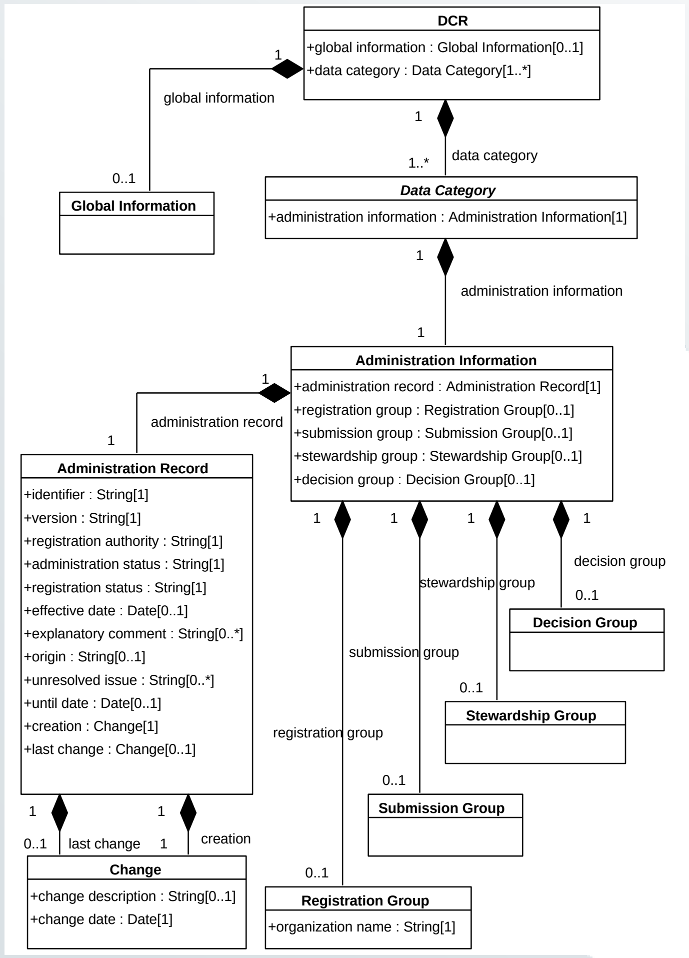
Separate DCR instances

- ISOcat is open source and will be available on sourceforge
  - see http://sourceforge.net/projects/isocat/
- Using the software other DCR instances can thus be created
  - for other domains
  - for ‘meta data categories’
  - …

‘meta data categories’

- Definition of a data category:
  - result of the specification of a given data field
- In a data model you’ve ‘containers’ which contain ‘data fields’
  - UML: classes contain attributes
  - Relational databases: tables contain fields
  - Data-centric XML documents: inner nodes and leaf nodes
  - TBX: …
- Can you create data categories for the ‘containers’?
  - Are those complex data categories?
  - Are they open/closed/constrained?
  - If so what would be their data type?
  - Or do we need a separate data category type?
  - May some data categories function in some applications as ‘containers’ in others as ‘data fields’?
- Will people expect data categories for the ‘containers’?
- Do we keep the TC37 DCR ‘pure’?
  - And store the ‘container’ concepts in the Relation Registry …

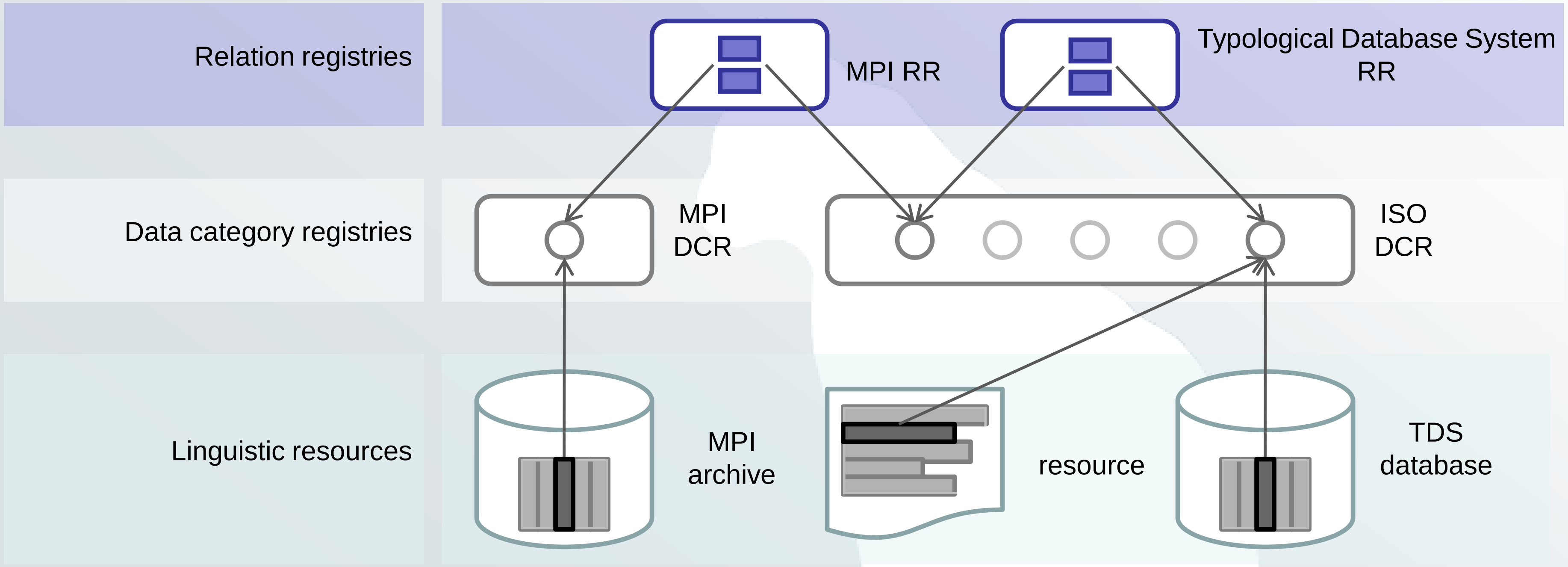
UML diagram



DCR
+global information : Global Information[0..1]
+data category : Data Category[1..*]
global information
1
0..1
Global Information
data category
1
1..*
Data Category
+administration information : Administration Information[1]
administration information
1
1
Administration Information
+administration record : Administration Record[1]
+registration group : Registration Group[0..1]
+submission group : Submission Group[0..1]
+stewardship group : Stewardship Group[0..1]
+decision group : Decision Group[0..1]
administration record
1
1
Administration Record
+identifier : String[1]
+version : String[1]
+registration authority : String[1]
+administration status : String[1]
+registration status : String[1]
+effective date : Date[0..1]
+explanatory comment : String[0..*]
+origin : String[0..1]
+unresolved issue : String[0..*]
+until date : Date[0..1]
+creation : Change[1]
+last change : Change[0..1]
last change
1
0..1
creation
1
1
Change
+change description : String[0..1]
+change date : Date[1]
decision group
1
0..1
Decision Group
stewardship group
1
0..1
Stewardship Group
submission group
1
0..1
Submission Group
registration group
1
0..1
Registration Group
+organization name : String[1]

January 8, 2009

Other registry types



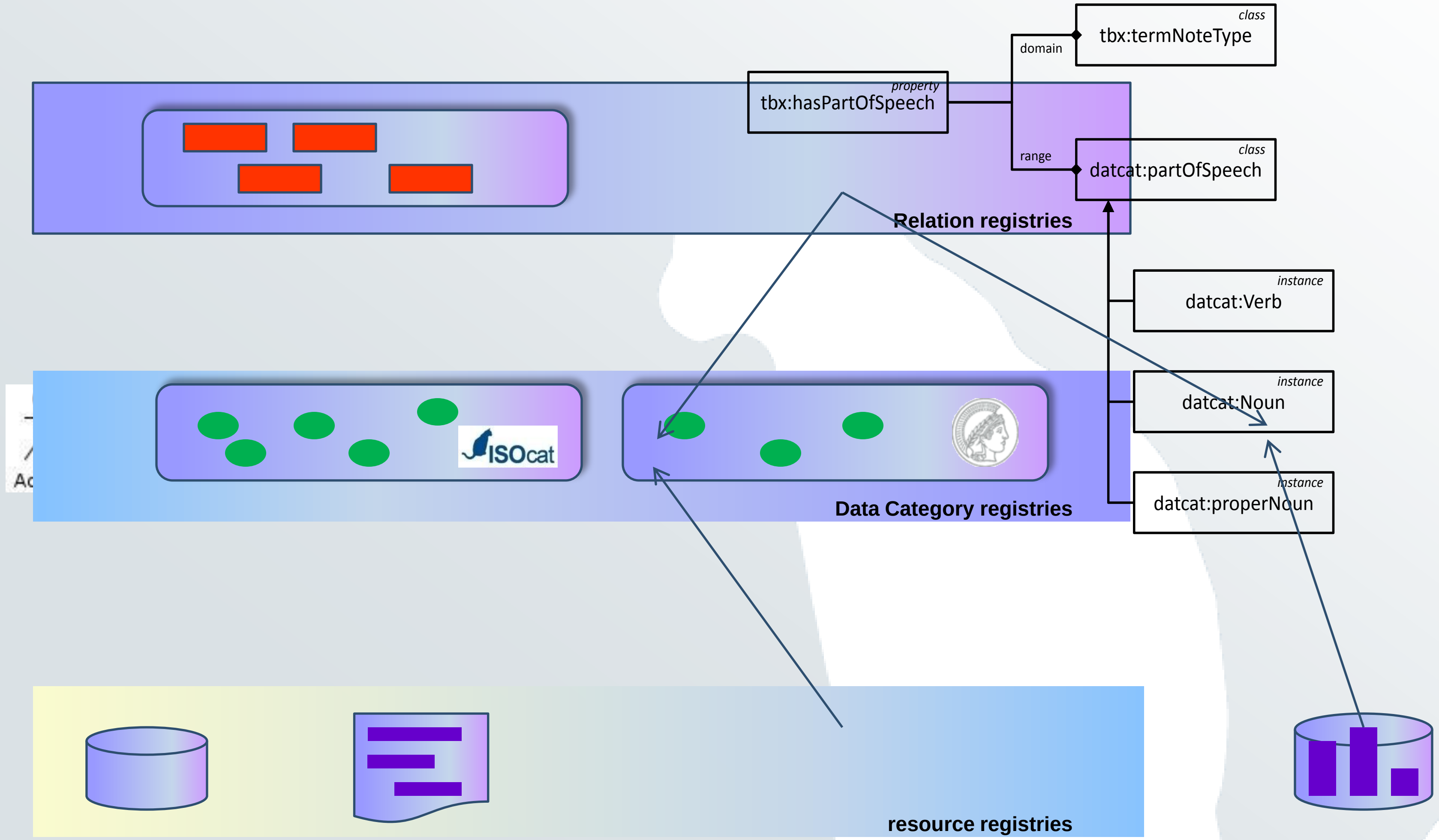
Relation registry

- A relation registry contains relationships between two or more data categories
- These relationships can be stored in various ways:
  - (fuzzy) equivalence
  - resource schemas
  - taxonomies
  - ontologies
  - …
- The registry can be populated manually, but also through some (machine learning) algorithm
- Registries may have different levels of trust
- The more semantic context the relationship encodes, the more effectively it can be utilized to determine semantic overlap

Utilizing the registry network

- If there is a set of common APIs an agent can traverse the network to identify semantic overlap, or help an user to understand a resource
  - A researcher finds an interesting resource in the MPI archive, and asks the agent to find similar resources. The agent crawls the network:
    1. The set of MPI DCR DCs related to the MPI resource
    2. A RR provides equivalence of some of these DCs with DCs from the TC37 DCR
    3. A cluster of the TC37 DCs appear in a common semantic context specified in the TDS RR
    4. Resources within this context in the TDS thus have a high chance of being of interest to the researcher

Registry network



Discussion

- How do you envision the (interaction in the) neighbourhood in which the DCR will operate?