

Jérôme Euzenat and Pavel Shvaiko

Ontology matching

Solutions to problems

(First edition)

Springer

D

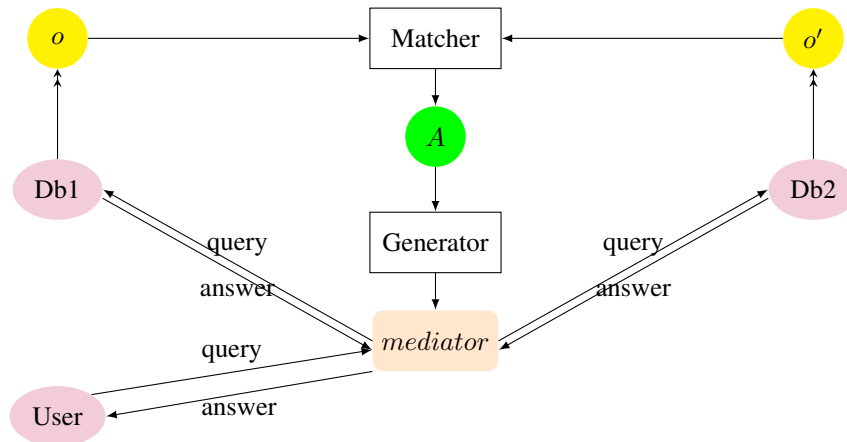
Solutions

We provide sketches of the solutions to the exercises of Appendix C. Most computations are made with the help of Alignment API (version 4.1). The book web site¹ provides scripts and Java classes for using the Alignment API in order to solve some of the questions.

D.1 Applications

Solution D.1 (Application definition).

1. A possible architecture for the application is provided by the following drawing:



2. The requirements of the application for ontology matching appears to be:
 - instances: YES, there are instances available.

¹ <http://book.ontologymatching.org>

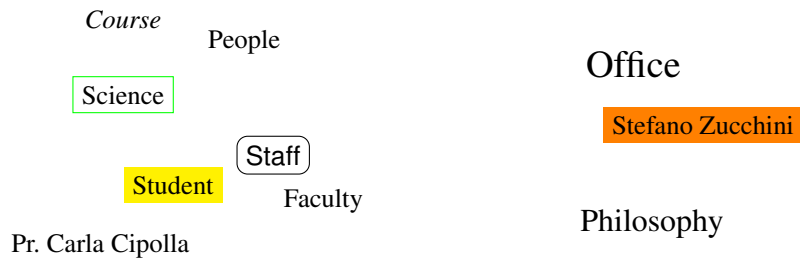
- run-time: No, matching is possible at design time. However, since it is said that the database keeps evolving, there is still come alignment maintenance to be carried out.
- automatic: NO, automatic matching is not strictly required and it may be wise to have results checked before putting them in production.
- correct: YES, the alignment must be correct. It would be particularly embarrassing, that it provides incorrect information.
- complete: Yes, the more complete the results, the best. However, having incomplete matching is not critical, this means that some information will be missing in the unified view.
- operation: the operation to be carried out is query mediation because the databases have to remain independent.

D.2 The matching problem

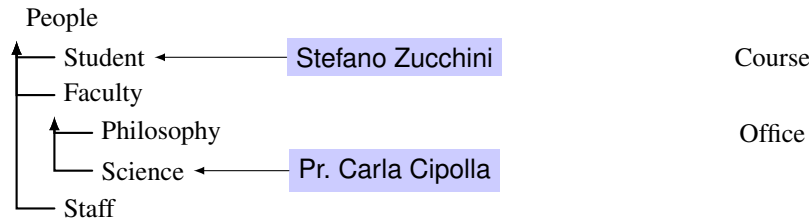
Solution D.2 (Ontology representation). Let o be a first data source to be integrated. o is described in English as follows:

- People are divided among Students, Faculty and Staff. Faculty is further divided depending on its departments and sub-departments, e.g., Philosophy, Science.
- People are characterised by their firstname and lastname which are strings, id which is a uri and birthdate which is a date.
- Students will attend Courses which are taught by Faculty and Faculty people have an Office as a room.
- Pr. Carla Cipolla is a visiting professor in Computer science and Stefano Zucchini is a PhD Student.

1. Provide a description for o as a folksonomy:



2. Provide a description for o as a directory:



3. Here is a description of *o* as an XML schema:

```

<schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.w3.org/2001/XMLSchema">

  <complexType name="People">
    <sequence>
      <element name="firstname" type="xsd:string" minOccurs="1" maxOccurs="1"/>
      <element name="lastname" type="xsd:string" minOccurs="1" maxOccurs="1"/>
      <element name="birthdate" type="xsd:date" minOccurs="1" maxOccurs="1"/>
      <element name="id" type="xsd:anyURI" minOccurs="1" maxOccurs="1"/>
      <element name="type" type="xsd:string"/>
    </sequence>
  </complexType>

  <complexType name="Student">
    <complexContent>
      <extension base="People">
        <sequence>
          <element name="attends" type="Course"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <complexType name="Faculty">
    <complexContent>
      <extension base="People">
        <sequence>
          <element name="office" type="Room" maxOccurs="1"/>
          <element name="department" type="Department" minOccurs="1"
            maxOccurs="1"/>
          <element name="teaches" type="Course"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <complexType name="Philosophy">
    <complexContent>
      <extension base="Faculty">
      </extension>
    </complexContent>
  </complexType>

  <complexType name="Science">
    <complexContent>
      <extension base="Faculty">
      </extension>
    </complexContent>
  </complexType>

  <complexType name="Course">
    <sequence>
      <element name="title" type="xsd:string" minOccurs="1" maxOccurs="1"/>
      <element name="room" type="Room" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>

```

4 Appendix D: Solutions

```
</sequence>
</complexType>

<complexType name="Departement">
  <sequence>
    <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>

<complexType name="Room">
  <sequence>
    <element name="name" type="xsd:string" minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>

</schema>
```

The two instances would be described by the following XML file:

```
<?xml version="1.0" ?>

<Student>
  <firstname>Stefano</firstname>
  <lastname>Zucchini</lastname>
  <birthdate>16/04/1987</birthdate>
  <id>1178</id>
  <type>PhD student</type>
</Student>

<Science>
  <firstname>Carla</firstname>
  <lastname>Cipolla</lastname>
  <birthdate>13/06/1977</birthdate>
  <id>1998</id>
  <type>Visiting professor</type>
  <departement>
    <Department><name>Computer science</name></Department>
  </departement>
</Science>
</xml>
```

4. Here is a description of *o* as an SQL relational database schema:

```
CREATE TABLE People (
  type VARCHAR(20),
  department VARCHAR(40),
  firstname VARCHAR(30),
  lastname VARCHAR(30),
  birthdate DATE,
  id URI,
  office INT,
  PRIMARY KEY (firstname lastname birthdate id)
);

CREATE TABLE Department (
  name VARCHAR(40),
  PRIMARY KEY (name)
);

CREATE TABLE Room (
  number INT,
  PRIMARY KEY (number)
);

CREATE TABLE Course (
  title VARCHAR(100),
  id VARCHAR(10),
```

```

        room INT,
        PRIMARY KEY (id)
    );

    CREATE TABLE teaches (
        course INT,
        firstname VARCHAR(30),
        lastname VARCHAR(30),
        birthdate DATE,
        id URI
    );

    CREATE TABLE attends (
        course INT,
        firstname VARCHAR(30),
        lastname VARCHAR(30),
        birthdate DATE,
        id URI
    );

    INSERT INTO People (firstname, lastname, birthdate, id, type, department)
    VALUES ('Carla', 'Cipolla', '13/06/1977', 1998, 'Visiting professor',
            'Computer science');

    INSERT INTO People (firstname, lastname, birthdate, id, type)
    VALUES ('Stefano', 'Zucchini', '16/04/1987', 1178, 'PhD student');
    
```

5. Figure D.1 is a description for o as a UML class diagram. Instances do not appear there.

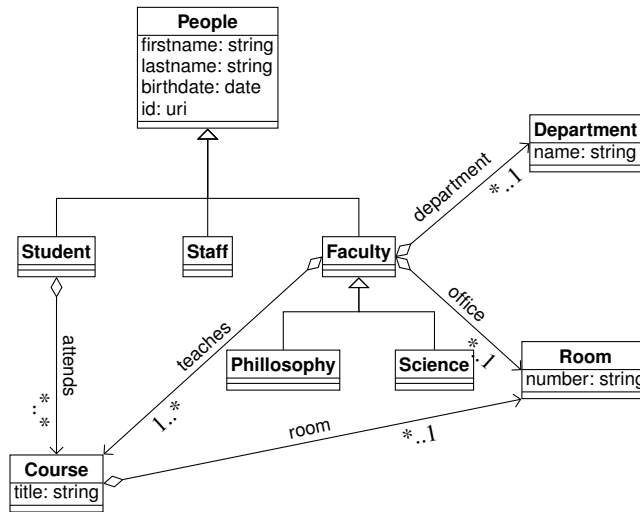
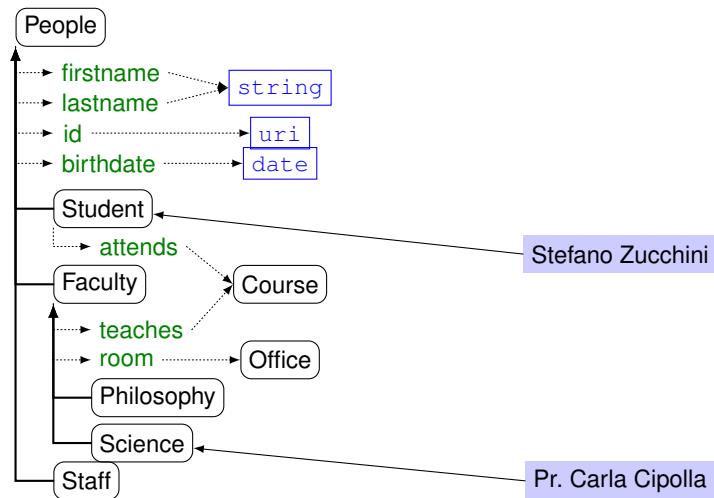


Fig. D.1. Representation of o as a UML class diagram.

6. Fig. D.2 is a description of o as an ontology.

Solution D.3 (Ontology semantics). Assume that the left hand side ontology of Fig. C.1 is denoted as o .

Fig. D.2. Representation of *o* as an ontology.

1. *o* can be expressed in OWL as follows:

```

<owl:Class rdf:about="#People">
  <rdfs:subClassOf
    <owl:Restriction>
      <owl:onProperty rdf:resource="#firstname"/>
      <owl:allValuesFrom rdf:resources="xsd:string"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf
    <owl:Restriction>
      <owl:onProperty rdf:resource="#lastname"/>
      <owl:allValuesFrom rdf:resources="xsd:string"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf
    <owl:Restriction>
      <owl:onProperty rdf:resource="#id"/>
      <owl:allValuesFrom rdf:resources="xsd:uri"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf
    <owl:Restriction>
      <owl:onProperty rdf:resource="#birthdate"/>
      <owl:allValuesFrom rdf:resources="xsd:date"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Student">
  <rdfs:subClassOf rdf:resource="#People"/>
  <rdfs:subClassOf
    <owl:Restriction>
      <owl:onProperty rdf:resource="#attends"/>
      <owl:allValuesFrom rdf:resources="#Course"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Faculty">

```



```

<rdfs:subClassOf rdf:resource="#People"/>
<rdfs:subClassOf
  <owl:Restriction>
    <owl:onProperty rdf:resource="#teaches"/>
    <owl:allValuesFrom rdf:resources="#Course"/>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf
  <owl:Restriction>
    <owl:onProperty rdf:resource="#room"/>
    <owl:allValuesFrom rdf:resources="#Office"/>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Science">
  <rdfs:subClassOf rdf:resource="#Faculty"/>
</owl:Class>
<owl:Class rdf:about="#Computer science">
  <rdfs:subClassOf rdf:resource="#Science"/>
</owl:Class>
<owl:Class rdf:about="#Biology">
  <rdfs:subClassOf rdf:resource="#Science"/>
</owl:Class>
<owl:Class rdf:about="#Philosophy">
  <rdfs:subClassOf rdf:resource="#Faculty"/>
</owl:Class>
<owl:Class rdf:about="#Boxology">
  <rdfs:subClassOf rdf:resource="#Faculty"/>
</owl:Class>
<owl:Class rdf:about="#Staff">
  <rdfs:subClassOf rdf:resource="#People"/>
</owl:Class>
<owl:Student rdf:about="#Stefano Zucchini" />
<owl:Computer Science rdf:about="#Pr. Carla Cipolla" />

```

2. *o* can be expressed in the simplified description logic format as follows:

People $\leq \forall$ firstname.string	People $\leq \forall$ lastname.string
People $\leq \forall$ id.uri	People $\leq \forall$ birthdate.date
Student \leq People	Student $\leq \forall$ attends.Course
Faculty \leq People	
Faculty $\leq \forall$ teaches.Course	Faculty $\leq \forall$ room.Office
Science \leq Faculty	Computer science \leq Science
Biology \leq Science	Philosophy \leq Faculty
Boxology \leq Faculty	Staff \leq People
Stefano Zucchini \in Student	Pr. Carla Cipolla \in Computer science

3. The models of *o* are those interpretations *m* over a domain *D* such that:

$$\begin{aligned}
m(\text{teaches}) &\subseteq D \times D \\
m(\text{attends}) &\subseteq D \times D \\
m(\text{People}) &\subseteq D \\
&\cap \{x \in D; \forall y; \langle x, y \rangle \in m(\text{firstname}), y \in \mathbb{S}\} \\
&\cap \{x \in D; \forall y; \langle x, y \rangle \in m(\text{lastname}), y \in \mathbb{S}\} \\
&\cap \{x \in D; \forall y; \langle x, y \rangle \in m(\text{id}), y \in \mathbb{U}\} \\
&\cap \{x \in D; \forall y; \langle x, y \rangle \in m(\text{birthdate}), y \in \mathbb{D}\} \\
m(\text{Student}) &\subseteq m(\text{People}) \\
&\cap \{x \in D; \forall y; \langle x, y \rangle \in m(\text{attends}), y \in m(\text{Course})\} \\
m(\text{Faculty}) &\subseteq m(\text{People}) \\
&\cap \{x \in D; \forall y; \langle x, y \rangle \in m(\text{teaches}), y \in m(\text{Course})\} \\
&\cap \{x \in D; \forall y; \langle x, y \rangle \in m(\text{room}), y \in m(\text{Office})\} \\
m(\text{Science}) &\subseteq m(\text{Faculty}) \\
m(\text{Computer science}) &\subseteq m(\text{Science}) \\
m(\text{Biology}) &\subseteq m(\text{Science}) \\
m(\text{Philosophy}) &\subseteq m(\text{Faculty}) \\
m(\text{Boxology}) &\subseteq m(\text{Faculty}) \\
m(\text{Staff}) &\subseteq m(\text{People}) \\
m(\text{Stefano Zucchini}) &\in m(\text{Student}) \\
m(\text{Pr. Carla Cipolla}) &\in m(\text{Computer science})
\end{aligned}$$

4. $o \models \text{Philosophy} \leq \text{People}$ because for any model m of o , $m(\text{Philosophy}) \subseteq m(\text{Faculty})$ and $m(\text{Faculty}) \subseteq m(\text{People})$, hence $m(\text{Philosophy}) \subseteq m(\text{People})$.
5. $o \not\models \text{teaches} \leq \text{attends}$ because the interpretation I with one course c , one student s and one faculty f such that $I(\text{People}) = \{s, t\}$, $I(\text{Course}) = \{c\}$, $I(\text{Student}) = \{s\}$, $I(\text{Computer science}) = I(\text{Science}) = I(\text{Faculty}) = \{f\}$, $I(\text{Stefano Zucchini}) = s$, $I(\text{Pr. Carla Cipolla}) = f$, $I(\text{attends}) = \{\langle s, c \rangle\}$, $I(\text{teaches}) = \{\langle f, c \rangle\}$ all other terms having an empty interpretation, is a model of the ontology and it does not satisfy the assertion.

D.3 Classification

Solution D.4 (Kinds of techniques). It seems that all the techniques considered in our classifications can be used:

- terminological: because the two ontologies are expressed in the same language. However, the poor overlap between the two sets of terms can be the sign of a poor overlap between the ontologies or require lexicon-based methods;
- structural: because there are constraints on properties, but we need anchors;

- extensional: because there are instances, this is invaluable, especially that they are common;
- semantic: also needs anchors.

D.4 Basic techniques

Solution D.5 (Name-based distance computation). Given the ontologies o and o' of Problem C.1 as illustrated in Fig. C.1.

1. Here are the distances or similarity between all the labels occurring in Fig. C.1 obtained with the OntoSim library (version 2.1)²
 - substring distance:

	Course	Science	Faculty	Student	Staff	Office	Boxology	People	Biology	Philosophy	ComputerScience
Assistant	0.87	0.88	0.88	0.75	0.57	0.87	1.00	1.00	0.88	0.89	0.92
Reference	0.87	0.50	0.88	0.75	0.86	0.73	1.00	0.87	1.00	1.00	0.67
AssistantProfessor	0.92	0.92	0.92	0.84	0.74	0.83	0.92	0.92	0.92	0.86	0.94
VisitingProfessor	0.91	0.92	0.92	0.92	0.91	0.83	0.92	0.91	0.92	0.85	0.94
ResearchAssistant	0.83	0.92	0.92	0.83	0.73	0.91	1.00	0.91	0.92	0.93	0.94
TeachingAssistant	0.91	0.92	0.83	0.83	0.73	0.91	0.92	0.91	0.92	0.85	0.88
Professor	0.87	0.88	0.88	0.88	0.86	0.73	0.88	0.87	0.88	0.79	0.92
Room	0.80	1.00	1.00	1.00	1.00	0.80	0.83	0.80	0.82	0.86	0.79
PhDStudent	0.88	0.76	0.88	0.18	0.73	0.88	1.00	0.88	1.00	0.80	0.84
Lecture	0.69	0.86	0.86	0.71	0.83	0.85	0.87	0.69	0.86	0.88	0.91
FullProfessor	0.89	0.90	0.80	0.90	0.89	0.79	0.90	0.89	0.90	0.83	0.93
AssociateProfessor	0.92	0.84	0.92	0.92	0.91	0.83	0.92	0.92	0.92	0.86	0.88
Staff	0.82	0.83	0.83	0.67	0.00	0.64	1.00	1.00	1.00	0.87	0.90

² <http://ontosim.gforge.inria.fr>

- 3-gram distance:

	Course	Science	Faculty	Student	Staff	Office	Boxology	People	Biology	Philosophy	ComputerScience
Assistant	1.00	1.00	1.00	1.00	0.80	1.00	1.00	1.00	1.00	1.00	1.00
Reference	1.00	0.67	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.80
AssistantProfessor	1.00	1.00	1.00	1.00	0.89	1.00	1.00	1.00	1.00	1.00	1.00
VisitingProfessor	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ResearchAssistant	1.00	1.00	1.00	1.00	0.89	1.00	1.00	1.00	1.00	1.00	1.00
TeachingAssistant	1.00	1.00	1.00	1.00	0.89	1.00	1.00	1.00	1.00	1.00	1.00
Professor	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Room	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
PhDStudent	1.00	1.00	1.00	0.23	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Lecture	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
FullProfessor	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
AssociateProfessor	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Staff	1.00	1.00	1.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00

There

is not much in common between the trigrams of these labels: trigrams are more effective on longer texts.

- edit distance:

	Course	Science	Faculty	Student	Staff	Office	Boxology	People	Biology	Philosophy	ComputerScience
Assistant	0.89	0.78	0.89	0.67	0.67	0.89	1.00	1.00	0.89	0.90	0.87
Reference	0.78	0.56	0.89	0.78	1.00	0.67	1.00	0.78	1.00	1.00	0.60
AssistantProfessor	0.89	0.78	0.94	0.83	0.78	0.89	0.94	0.89	0.89	0.89	0.89
VisitingProfessor	0.88	0.82	1.00	0.82	0.82	0.88	0.94	0.88	0.88	0.88	0.94
ResearchAssistant	0.88	0.82	0.82	0.82	0.82	0.94	1.00	0.94	0.94	0.88	0.88
TeachingAssistant	0.88	0.88	0.82	0.82	0.82	0.94	0.94	0.94	0.94	0.82	0.88
Professor	0.78	0.89	1.00	0.89	0.89	0.78	0.89	0.78	0.89	0.80	0.87
Room	0.83	1.00	1.00	1.00	1.00	1.00	0.75	0.83	0.71	0.80	0.93
PhDStudent	0.90	0.80	0.90	0.30	0.80	0.90	1.00	0.80	1.00	0.80	0.80
Lecture	0.57	0.86	0.86	0.86	0.86	0.86	1.00	0.71	1.00	0.90	0.80
FullProfessor	0.85	0.92	0.92	0.92	0.92	0.85	0.92	0.85	0.85	0.85	0.93
AssociateProfessor	0.83	0.72	0.89	0.83	0.83	0.89	0.89	0.83	0.89	0.89	0.78
Staff	1.00	0.86	1.00	0.71	0.00	1.00	1.00	1.00	1.00	0.90	0.93

– Jaro–Winckler measure:

	Course	Science	Faculty	Student	Staff	Office	Boxology	People	Biology	Philosophy	ComputerScience
Assistant	0.72	0.75	0.68	0.67	0.78	0.74	1.00	1.00	0.75	0.89	0.83
Reference	0.84	0.70	0.75	0.78	0.80	0.66	1.00	0.64	1.00	1.00	0.74
AssistantProfessor	0.74	0.80	0.71	0.73	0.80	0.76	1.00	1.00	0.76	0.81	0.76
VisitingProfessor	0.75	0.86	0.76	0.81	1.00	0.73	0.76	1.00	0.81	0.75	0.73
ResearchAssistant	0.86	0.84	1.00	0.72	1.00	0.91	1.00	0.71	1.00	0.71	0.75
TeachingAssistant	1.00	0.84	0.72	0.82	0.70	0.91	0.76	0.75	0.76	0.72	0.79
Professor	0.76	0.82	1.00	0.75	0.80	0.70	0.75	0.70	0.70	0.66	0.72
Room	0.67	1.00	1.00	1.00	1.00	0.67	0.65	0.67	0.64	0.98	0.72
PhDStudent	0.77	0.93	0.90	0.68	1.00	0.75	1.00	0.79	1.00	0.68	0.73
Lecture	0.55	0.80	0.76	0.65	1.00	0.68	1.00	0.68	0.97	0.98	0.75
FullProfessor	0.69	0.76	0.65	0.76	1.00	0.80	0.75	0.77	0.80	0.74	0.72
AssociateProfessor	0.90	0.79	0.68	0.81	0.80	0.79	0.85	0.74	0.83	0.79	0.71
Staff	1.00	0.73	0.73	0.66	0.40	0.69	1.00	1.00	1.00	1.00	1.00

2. For computing linguistic measures, we have used WordNet 3.0, JWNL 1.4, Lucene 3.0.2 through the OntoSim library and the Alignment API 4.1:

– cosynonymy distance:

	Course	Science	Faculty	Student	Staff	Office	Boxology	People	Biology	Philosophy	ComputerScience
Assistant	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Reference	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
AssistantProfessor	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
VisitingProfessor	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ResearchAssistant	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
TeachingAssistant	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Professor	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Room	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
PhDStudent	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Lecture	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
FullProfessor	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
AssociateProfessor	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Staff	1.00	1.00	0.89	1.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00

This distance is not very useful because few synonyms are in the classes.

- gloss overlap similarity (as a distance):

	Course	Science	Faculty	Student	Staff	Office	Boxology	People	Biology	Philosophy	ComputerScience
Assistant	0.98	1.00	1.00	0.93	0.94	0.98	1.00	1.00	1.00	0.97	1.00
Reference	0.96	0.98	1.00	0.97	0.98	0.96	1.00	0.99	0.99	0.97	1.00
AssistantProfessor	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
VisitingProfessor	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ResearchAssistant	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
TeachingAssistant	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Professor	0.98	1.00	0.94	0.92	0.98	1.00	1.00	0.96	1.00	1.00	1.00
Room	0.94	1.00	0.97	0.94	0.97	1.00	1.00	0.97	0.97	0.97	1.00
PhDStudent	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Lecture	1.00	0.96	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.97	1.00
FullProfessor	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
AssociateProfessor	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Staff	0.92	1.00	0.92	0.98	0.00	0.97	1.00	0.97	1.00	0.97	1.00

- Wu–Palmer similarity (as a distance):

	Course	Science	Faculty	Student	Staff	Office	Boxology	People	Biology	Philosophy	ComputerScience
Assistant	0.50	1.00	1.00	0.40	0.57	0.42	1.00	1.00	0.50	1.00	1.00
Reference	0.12	0.43	0.50	0.52	0.40	0.36	1.00	0.75	0.43	0.38	1.00
AssistantProfessor	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
VisitingProfessor	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
ResearchAssistant	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
TeachingAssistant	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Professor	0.58	1.00	1.00	0.54	0.65	0.48	1.00	1.00	0.57	1.00	1.00
Room	0.20	0.78	0.25	0.47	0.33	0.22	1.00	0.50	0.55	0.80	1.00
PhDStudent	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Lecture	0.29	0.67	0.67	1.00	0.67	0.29	1.00	0.80	0.62	0.69	1.00
FullProfessor	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
AssociateProfessor	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Staff	0.29	0.80	0.00	0.44	0.00	0.09	1.00	0.64	0.52	0.82	1.00

Solution D.6 (Extensional distance computation). Consider the two ontologies o and o' of Problem C.1 as illustrated in Fig. C.1; assume that the two following tables of Problem C.6 specify data instances for o and o' , respectively.

1. In order to use the extensions of classes for matching o and o' , it is first necessary to produce record linkage because there are no “common extension” or easy way to identify objects.

2. It is clear that `firstname` and `lastname` in the first table have to be compared with `name` in the second one. An easy approximation would be to check if the `lastname` is a suffix for `name`, this joined with the test that `firstname` is also a prefix (of length 1) for `name` should provide a good guess. Unfortunately, it does not help to distinguish between Paola and Pierluigi Pomodoro. It is also not enough for identifying Federico di Guava since the name has not be written in the same way. So, further normalisation of the names should certainly be performed (identifying first and last names in the second table and comparing them to the first one). So, one good record linkage technique for these tables would be:

- a) `name2` is `name` with the first term (alternatively all terms but the last) acronymed, i.e., replaced by their first letter followed with a dot ".";
- b) `name3` is `name2` with all capital letter but the first in each names preceded by a space " ";
- c) `name'2` is the concatenation of `firstname` and `lastname`;
- d) compute edit distance between `name3` and `name'2`.

3. This distance will provide the following results:

	128	1998	1678	1178	88	445	1664	unknown	178
fdg	0.90	0.70	0.90	0.90	0.70	0.90	0.80	0.00	0.91
gc	0.50	0.70	0.70	0.90	0.70	0.00	0.70	0.90	0.64
sz	0.80	0.90	0.90	0.00	0.90	0.90	0.90	0.90	0.82
plp	0.60	0.80	0.00	0.90	0.70	0.70	0.80	0.90	0.64
cc	0.80	0.00	0.80	0.90	0.78	0.70	0.56	0.70	0.73
pc	0.00	0.80	0.60	0.80	0.80	0.50	0.70	0.90	0.73
ms	0.64	0.91	0.82	0.64	0.91	0.73	0.64	0.91	0.82
pp	0.60	0.80	0.00	0.90	0.70	0.70	0.80	0.90	0.64
dm	0.91	0.82	0.91	0.82	0.73	0.82	0.73	0.73	0.64

4. The substring similarity yields:

	128	1998	1678	1178	88	445	1664	unknown	178
fdg	0.92	0.91	0.92	0.92	0.83	0.83	0.91	0.58	0.84
gc	0.85	0.84	0.92	0.92	0.92	0.38	0.84	0.92	0.85
sz	0.92	0.91	0.92	0.33	0.91	0.92	0.91	0.92	0.92
plp	0.92	0.92	0.38	0.92	0.84	0.92	0.92	0.85	0.93
cc	0.71	0.30	0.90	0.90	0.90	0.81	0.80	0.90	0.91
pc	0.27	0.81	0.91	0.91	0.90	0.73	0.81	0.91	0.83
ms	0.83	0.91	0.91	0.83	0.91	0.74	0.82	0.91	0.83
pp	0.91	0.81	0.27	1.00	0.90	0.82	0.71	0.91	0.91
dm	0.92	0.84	0.85	0.85	0.92	0.92	0.84	0.92	0.78

5. Single linkage applied to the results of the first method comparing individuals (Answer 3) provides:

	Course	Science	Faculty	Student	Staff	Office	Boxology	People	Biology	Philosophy	ComputerScience
Assistant	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Reference	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
AssistantProfessor	1.00	0.00	0.00	0.70	0.64	1.00	1.00	0.64	0.60	0.00	0.00
VisitingProfessor	1.00	0.00	0.60	0.56	0.64	1.00	1.00	0.56	0.00	0.60	0.00
ResearchAssistant	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
TeachingAssistant	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Professor	1.00	0.00	0.00	0.56	0.64	1.00	1.00	0.56	0.00	0.00	0.00
Room	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
PhDStudent	1.00	0.73	0.64	0.00	0.64	1.00	1.00	0.00	0.73	0.64	0.73
Lecture	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
FullProfessor	1.00	0.00	0.90	0.80	0.91	1.00	1.00	0.80	0.70	0.90	0.00
AssociateProfessor	1.00	0.00	0.60	0.80	0.64	1.00	1.00	0.64	0.00	0.60	0.70
Staff	1.00	0.73	0.64	0.00	0.64	1.00	1.00	0.00	0.73	0.64	0.73

and the substring comparison yields:

	Course	Science	Faculty	Student	Staff	Office	Boxology	People	Biology	Philosophy	ComputerScience
Assistant	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Reference	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
AssistantProfessor	1.00	0.38	0.27	0.81	0.83	1.00	1.00	0.81	0.90	0.27	0.38
VisitingProfessor	1.00	0.30	0.71	0.80	0.91	1.00	1.00	0.80	0.38	0.71	0.30
ResearchAssistant	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
TeachingAssistant	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Professor	1.00	0.27	0.27	0.71	0.83	1.00	1.00	0.71	0.27	0.27	0.30
Room	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
PhDStudent	1.00	0.74	0.83	0.33	0.78	1.00	1.00	0.33	0.85	0.83	0.74
Lecture	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
FullProfessor	1.00	0.58	0.92	0.91	0.84	1.00	1.00	0.84	0.83	0.92	0.58
AssociateProfessor	1.00	0.27	0.91	0.71	0.91	1.00	1.00	0.71	0.27	0.91	0.81
Staff	1.00	0.74	0.83	0.33	0.78	1.00	1.00	0.33	0.85	0.83	0.74

D.5 Strategies

Solution D.7 (Measure aggregation). Consider the distances between ontology entities in o and o' given by (i) edit distance computed on their names (see Problem D.5,

3rd item), and (ii) the distance computed with the single linkage measure applied to the substring distance (see Problem D.6, 3).

1. The aggregation with the $\max(x + y - 1, 0)$ triangular norm is:

	Course	Science	Faculty	Student	Staff	Office	Boxology	People	Biology	Philosophy	ComputerScience
Assistant	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Reference	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
AssistantProfessor	1.00	0.78	0.94	0.83	0.83	1.00	1.00	0.89	0.90	0.89	0.89
VisitingProfessor	1.00	0.82	1.00	0.82	0.91	1.00	1.00	0.88	0.88	0.88	0.94
ResearchAssistant	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
TeachingAssistant	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Professor	1.00	0.89	1.00	0.89	0.89	1.00	1.00	0.78	0.89	0.80	0.87
Room	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
PhDStudent	1.00	0.80	0.90	0.33	0.80	1.00	1.00	0.80	1.00	0.83	0.80
Lecture	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
FullProfessor	1.00	0.92	0.92	0.92	0.92	1.00	1.00	0.85	0.85	0.92	0.93
AssociateProfessor	1.00	0.72	0.91	0.83	0.91	1.00	1.00	0.83	0.89	0.91	0.81
Staff	1.00	0.86	1.00	0.71	0.78	1.00	1.00	1.00	1.00	0.90	0.93

2. The aggregation with the weighted product, such that $2/3$ is the weight for the former distance and $1/3$ is the weight for the latter one, is:

	Course	Science	Faculty	Student	Staff	Office	Boxology	People	Biology	Philosophy	ComputerScience
Assistant	0.20	0.17	0.20	0.15	0.15	0.20	0.22	0.22	0.20	0.20	0.19
Reference	0.17	0.12	0.20	0.17	0.22	0.15	0.22	0.17	0.22	0.22	0.13
AssistantProfessor	0.20	0.07	0.06	0.15	0.14	0.20	0.21	0.16	0.18	0.05	0.08
VisitingProfessor	0.20	0.05	0.16	0.15	0.17	0.20	0.21	0.16	0.08	0.14	0.06
ResearchAssistant	0.20	0.18	0.18	0.18	0.18	0.21	0.22	0.21	0.21	0.20	0.20
TeachingAssistant	0.20	0.20	0.18	0.18	0.18	0.21	0.21	0.21	0.21	0.18	0.20
Professor	0.17	0.05	0.06	0.14	0.16	0.17	0.20	0.12	0.05	0.05	0.06
Room	0.19	0.22	0.22	0.22	0.22	0.22	0.17	0.19	0.16	0.18	0.21
PhDStudent	0.20	0.13	0.17	0.02	0.14	0.20	0.22	0.06	0.19	0.15	0.13
Lecture	0.13	0.19	0.19	0.19	0.19	0.19	0.22	0.16	0.22	0.20	0.18
FullProfessor	0.19	0.12	0.19	0.19	0.17	0.19	0.21	0.16	0.16	0.17	0.12
AssociateProfessor	0.19	0.04	0.18	0.13	0.17	0.20	0.20	0.13	0.05	0.18	0.14
Staff	0.22	0.14	0.18	0.05	0.00	0.22	0.22	0.07	0.19	0.17	0.15

3. The aggregation with the weighted sum, such that $2/3$ is the weight for the former distance and $1/3$ is the weight for the latter one, is:

	Course	Science	Faculty	Student	Staff	Office	Boxology	People	Biology	Philosophy	ComputerScience
Assistant	0.93	0.85	0.93	0.78	0.78	0.93	1.00	1.00	0.93	0.93	0.91
Reference	0.85	0.70	0.93	0.85	1.00	0.78	1.00	0.85	1.00	1.00	0.73
AssistantProfessor	0.93	0.65	0.72	0.83	0.79	0.93	0.96	0.86	0.89	0.68	0.72
VisitingProfessor	0.92	0.65	0.90	0.82	0.85	0.92	0.96	0.85	0.72	0.83	0.73
ResearchAssistant	0.92	0.88	0.88	0.88	0.88	0.96	1.00	0.96	0.96	0.92	0.92
TeachingAssistant	0.92	0.92	0.88	0.88	0.88	0.96	0.96	0.96	0.96	0.88	0.92
Professor	0.85	0.68	0.76	0.83	0.87	0.85	0.93	0.76	0.68	0.62	0.68
Room	0.89	1.00	1.00	1.00	1.00	1.00	0.83	0.89	0.81	0.87	0.96
PhDStudent	0.93	0.78	0.88	0.31	0.79	0.93	1.00	0.64	0.95	0.81	0.78
Lecture	0.71	0.90	0.90	0.90	0.90	0.90	1.00	0.81	1.00	0.93	0.87
FullProfessor	0.90	0.81	0.92	0.92	0.90	0.90	0.95	0.84	0.84	0.87	0.82
AssociateProfessor	0.89	0.57	0.90	0.79	0.86	0.93	0.93	0.79	0.68	0.90	0.79
Staff	1.00	0.82	0.94	0.59	0.26	1.00	1.00	0.78	0.95	0.88	0.87

4. The aggregation with the ordered weighted average, such that $2/3$ is the weight for the former distance and $1/3$ is the weight for the latter one, is:

	Course	Science	Faculty	Student	Staff	Office	Boxology	People	Biology	Philosophy	ComputerScience
Assistant	0.96	0.93	0.96	0.89	0.89	0.96	1.00	1.00	0.96	0.97	0.96
Reference	0.93	0.85	0.96	0.93	1.00	0.89	1.00	0.93	1.00	1.00	0.87
AssistantProfessor	0.96	0.65	0.72	0.83	0.81	0.96	0.98	0.86	0.90	0.68	0.72
VisitingProfessor	0.96	0.65	0.90	0.82	0.88	0.96	0.98	0.85	0.72	0.83	0.73
ResearchAssistant	0.96	0.94	0.94	0.94	0.94	0.98	1.00	0.98	0.98	0.96	0.96
TeachingAssistant	0.96	0.96	0.94	0.94	0.94	0.98	0.98	0.98	0.98	0.94	0.96
Professor	0.93	0.68	0.76	0.83	0.87	0.93	0.96	0.76	0.68	0.62	0.68
Room	0.94	1.00	1.00	1.00	1.00	1.00	0.92	0.94	0.90	0.93	0.98
PhDStudent	0.97	0.78	0.88	0.32	0.79	0.97	1.00	0.64	0.95	0.82	0.78
Lecture	0.86	0.95	0.95	0.95	0.95	0.95	1.00	0.90	1.00	0.97	0.93
FullProfessor	0.95	0.81	0.92	0.92	0.90	0.95	0.97	0.84	0.84	0.89	0.82
AssociateProfessor	0.94	0.57	0.90	0.79	0.89	0.96	0.96	0.79	0.68	0.90	0.80
Staff	1.00	0.82	0.94	0.59	0.52	1.00	1.00	0.78	0.95	0.88	0.87

In summary, it seems that the two measures either complement each other or agree. The two former aggregations heavily depend on the higher or the lower aggregated value, while the two latter aggregation are rather an average between these values. In spite of the different techniques used, they provide similar results.

Solution D.8 (Thresholds). The correspondences resulting from the application of a .6 threshold to the similarity of Problem C.8 are:

1. for a hard threshold of .6 are (11):

PhDStudent = .92 Student	Staff = .65 Student
PhDStudent = .65 Philosophy	Staff = .64 Science
PhDStudent = .64 People	Staff = .63 Staff
Professor = .62 People	Assistant = .62 Science
Room = .62 Office	Professor = .60 Faculty
PhDStudent = .60 Science	

3. for a proportional threshold of .6 are those over the best value (.92) multiplied by .6, i.e., those over .552, that is the previous ones plus 2 correspondences (+2=13):

Assistant = .58 Faculty	Staff = .56 People
-------------------------	--------------------

2. for a delta threshold of .6 are those over the best value (.92) minus .6, i.e., those over .32, that is the previous ones plus 17 correspondences (+17=30):

Professor = .55 Staff	PhDStudent = .55 Staff
PhDStudent = .52 Boxology	Assistant = .46 Philosophy
PhDStudent = .45 Faculty	Professor = .44 Philosophy
Assistant = .44 Student	Assistant = .43 Staff
Professor = .40 Science	Assistant = .40 People
Professor = .36 Student	Professor = .36 Office
Lecture = .34 Philosophy	PhDStudent = .34 Office
Assistant = .33 Boxology	Staff = .33 Faculty
PhDStudent = .33 Course	

4. for a percentage threshold of .6 are the 60% best correspondences, i.e., the 37 best correspondences, that is the previous ones plus 7 correspondences (+7=37 because there are three correspondence at .23):

Professor = .32 Boxology	Assistant = .32 Course
Reference = .28 Boxology	Reference = .26 Philosophy
Lecture = .26 Faculty	Reference = .25 Science
Reference = .23 People	... or any of the other at .23

Solution D.9 (Alignment extraction). Consider the two ontologies o and o' of Problem C.1 as illustrated in Fig. C.1. Assume that the similarity between their entities is expressed by the similarity table of Problem C.8.

1. The alignment extracted with the greedy algorithm, is:

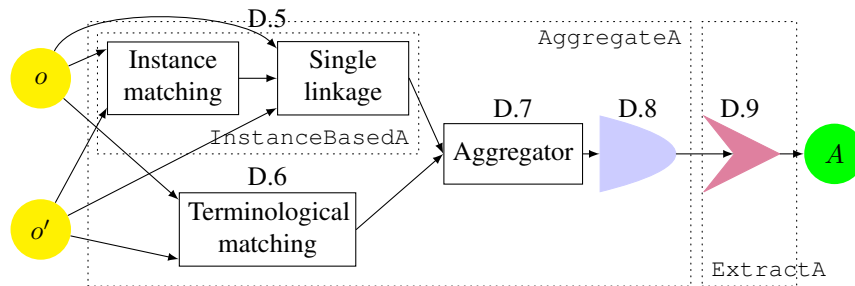
PhDStudent = .92	Student	Staff = .64	Science
Professor = .62	People	Room = .62	Office
Assistant = .58	Faculty	Lecture = .34	Philosophy
Reference = .28	Boxology		

- The alignment provided by the greedy algorithm is stable;
- The weights of the alignment extracted by the greedy algorithms sum to 4.0. The maximum weight alignment, given below, has weights summing to 4.03:

PhDStudent = .92	Student	Staff = .63	Staff
Professor = .62	People	Room = .62	Office
Assistant = .62	Science	Lecture = .34	Philosophy
Reference = .28	Boxology		

Solution D.10 (Composing matchers). Consider the application described in Problem D.1.

1. The matching architecture developed so far would be:



The advantage of this architecture is that it allows to test many approaches for terminological matching (D.5), instance-based matching (D.6), aggregation (D.7), applying threshold (D.8) or extracting (D.9) the alignment.

- The figure above provides the name of the classes, available on the book web site, which implements the surrounding dotted composition. The implementation in a new `Alignment` class corresponds to composing the `AggregateAlignment` and `ExtractAlignment` processes. As implemented in the `ComposedAlignment` class, this can be achieved through:

```
public void align( Alignment alignment, Properties params )
    throws AlignmentException {
    // Call AggregateAlignment
    matcher.align( alignment, params );

    // Call ExtractAlignment... that only do extraction from matcher
    extractor.align( matcher, params );

    // Extract alignment
    for ( Cell c : extractor ) { addCell( c ); }
}
```

D.6 Evaluation of matching systems

Solution D.11 (Precision and recall computation). The precision, recall, F-measure, overall and Hamming distance alignments A_1 , A_2 and A_3 and R is given in the following table:

Alignment	precision	recall	F-measure	overall	Hamming distance
R	1.	1.	1.	1.	1.
A_1	.33	.33	.33	-.33	.2
A_2	.66	.66	.66	.33	.5
A_3	.33	.33	.33	-.33	.2

Hence A_2 seems to be the best alignment. The results for A_3 are different from those given by the Alignment API because, we have considered here that the correspondence `Student = PhDStudent` is not correct because of the relation which should have been \geq , though the Alignment API only considers the entities and thus consider the correspondence correct (all values are .5).

Solution D.12 (Application specific evaluation). We have three systems and only one application. This application, given our previous answers speed (runtime=No) and automation (automatic=NO) are not important, precision is very important (correct=YES), recall moderately important (complete=Yes). Hence, we would assign weights 1 to speed and automation, 5 to precision, 4 to recall, which, once normalised yield .1, .5 and .4. Since we have no information about speed, it is not taken into account and all systems are equally automated.

	A_1	A_2	A_3
Automatic	1.	1.	1.
Precision	.33	.66	.33
Automatic Precision			
Recall	.33	.66	.33
Our application	.1	.5	.4
	.27	.63	.27

The results are very similar to those provided by F-measure.

D.7 Representing alignments

Solution D.13 (Representation generation). Consider R as described in Fig. C.2.

1. R in OWL as given by the Alignment API (see Solution D.16, Question 1, for the actual API output):

```
<?xml version='1.0' encoding='utf-8' standalone='no'?>
<!DOCTYPE rdf:RDF [
  <ENTITY admin "http://book.ontologymatching.org/exercise/admin.owl#" >
  <ENTITY lab "http://book.ontologymatching.org/exercise/lab.owl#" >
  >
<rdf:RDF
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

```

```

xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"

<owl:Ontology rdf:about="">
  <rdfs:comment>OWL Alignment rendering</rdfs:comment>
  <rdfs:comment>(altered from OWLXiomsRendererVisitor
                output)</rdfs:comment>
</owl:Ontology>

<owl:Class rdf:about="&admin;Office">
  <owl:equivalentClass rdf:resource="&lab;Room"/>
</owl:Class>
<owl:Class rdf:about="&lab;PhDStudent">
  <rdfs:subClassOf rdf:resource="&admin;Student"/>
</owl:Class>
<owl:ObjectProperty rdf:about="&admin;room">
  <owl:equivalentProperty rdf:resource="&lab;office"/>
</owl:ObjectProperty>
<owl:Class rdf:about="&admin;Faculty">
  <owl:equivalentClass rdf:resource="&lab;FullProfessor"/>
</owl:Class>
<owl:ObjectProperty rdf:about="&admin;teaches">
  <owl:equivalentProperty rdf:resource="&lab;teaches"/>
</owl:ObjectProperty>
<owl:Class rdf:about="&admin;Course">
  <owl:equivalentClass rdf:resource="&lab;Lecture"/>
</owl:Class>
</rdf:RDF>

```

2. *R* in C-OWL as given by the Alignment API:

```

<?xml version='1.0' encoding='utf-8' standalone='no' ?>
<!DOCTYPE rdf:RDF [
  <!ENTITY admin "http://book.ontologymatching.org/exercise/admin.owl#" >
  <!ENTITY lab "http://book.ontologymatching.org/exercise/lab.owl#" >
]
>
<rdf:RDF
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:cowl="http://www.itc.it/cowl#"
  xml:base="http://www.itc.it/cowl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"

  <cowl:Mapping rdf:ID="">
    <cowl:sourceOntology>
      <owl:Ontology
        rdf:about="http://book.ontologymatching.org/exercise/admin.owl"/>
      </owl:sourceOntology>
    <cowl:targetOntology>
      <owl:Ontology
        rdf:about="http://book.ontologymatching.org/exercise/lab.owl"/>
      </owl:targetOntology>
    <cowl:bridgeRule>
      <cowl:Equivalent>
        <cowl:source>
          <owl:Class rdf:about="&admin;Office"/>
        </owl:source>
        <cowl:target>
          <owl:Class rdf:about="&lab;Room"/>
        </owl:target>
      </owl:Equivalent>
    </owl:bridgeRule>
    <cowl:bridgeRule>
      <cowl:Onto>
        <cowl:source>
          <owl:Class rdf:about="&admin;Student"/>
        </owl:source>

```

```

    <owl:target>
      <owl:Class rdf:about="&lab;PhDStudent"/>
    </owl:target>
  </owl:Onto>
</owl:bridgeRule>
<owl:bridgeRule>
  <owl:Equivalent>
    <owl:source>
      <owl:ObjectProperty rdf:about="&admin;room"/>
    </owl:source>
    <owl:target>
      <owl:ObjectProperty rdf:about="&lab;office"/>
    </owl:target>
  </owl:Equivalent>
</owl:bridgeRule>
<owl:bridgeRule>
  <owl:Equivalent>
    <owl:source>
      <owl:Class rdf:about="&admin;Faculty"/>
    </owl:source>
    <owl:target>
      <owl:Class rdf:about="&lab;FullProfessor"/>
    </owl:target>
  </owl:Equivalent>
</owl:bridgeRule>
<owl:bridgeRule>
  <owl:Equivalent>
    <owl:source>
      <owl:ObjectProperty rdf:about="&admin;teaches"/>
    </owl:source>
    <owl:target>
      <owl:ObjectProperty rdf:about="&lab;teaches"/>
    </owl:target>
  </owl:Equivalent>
</owl:bridgeRule>
<owl:bridgeRule>
  <owl:Equivalent>
    <owl:source>
      <owl:Class rdf:about="&admin;Course"/>
    </owl:source>
    <owl:target>
      <owl:Class rdf:about="&lab;Lecture"/>
    </owl:target>
  </owl:Equivalent>
</owl:bridgeRule>
</owl:Mapping>
</rdf:RDF>

```

3. *R* in SWRL as given by the Alignment API:

```

<?xml version='1.0' encoding='utf-8' standalone='no'?>
<!DOCTYPE swrlx:Ontology [
  <!ENTITY admin "http://book.ontologymatching.org/exercise/admin.owl#" >
  <!ENTITY lab "http://book.ontologymatching.org/exercise/lab.owl#" >
]
>

<swrlx:Ontology swrlx:name="generatedA1"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:swrlx="http://www.w3.org/2003/11/swrlx#"
  xmlns:owlx="http://www.w3.org/2003/05/owl-xml"
  xmlns:ruleml="http://www.w3.org/2003/11/ruleml#">
  <owlx:Imports
    rdf:resource="http://book.ontologymatching.org/exercise/admin.owl"/>

  <ruleml:imp>
    <ruleml:_body>
      <swrl:classAtom>
        <owlx:Class owlx:name="&admin;Office"/>

```

```

        <ruleml:var>x</ruleml:var>
    </swrl:classAtom>
</ruleml:_body>
<ruleml:_head>
    <swrlx:classAtom>
        <owlx:Class owlx:name="&lab;Room"/>
        <ruleml:var>x</ruleml:var>
    </swrl:classAtom>
</ruleml:_head>
</ruleml:imp>

<ruleml:imp>
    <ruleml:_body>
        <swrl:individualPropertyAtom swrlx:property="&admin;room"/>
        <ruleml:var>x</ruleml:var>
        <ruleml:var>y</ruleml:var>
    </swrl:individualPropertyAtom>
    </ruleml:_body>
    <ruleml:_head>
        <swrl:individualPropertyAtom swrlx:property="&lab;office"/>
        <ruleml:var>x</ruleml:var>
        <ruleml:var>y</ruleml:var>
    </swrl:individualPropertyAtom>
    </ruleml:_head>
</ruleml:imp>

<ruleml:imp>
    <ruleml:_body>
        <swrl:classAtom>
            <owlx:Class owlx:name="&admin;Faculty"/>
            <ruleml:var>x</ruleml:var>
        </swrl:classAtom>
    </ruleml:_body>
    <ruleml:_head>
        <swrlx:classAtom>
            <owlx:Class owlx:name="&lab;FullProfessor"/>
            <ruleml:var>x</ruleml:var>
        </swrl:classAtom>
    </ruleml:_head>
</ruleml:imp>

<ruleml:imp>
    <ruleml:_body>
        <swrl:individualPropertyAtom swrlx:property="&admin;teaches"/>
        <ruleml:var>x</ruleml:var>
        <ruleml:var>y</ruleml:var>
    </swrl:individualPropertyAtom>
    </ruleml:_body>
    <ruleml:_head>
        <swrl:individualPropertyAtom swrlx:property="&lab;teaches"/>
        <ruleml:var>x</ruleml:var>
        <ruleml:var>y</ruleml:var>
    </swrl:individualPropertyAtom>
    </ruleml:_head>
</ruleml:imp>

<ruleml:imp>
    <ruleml:_body>
        <swrl:classAtom>
            <owlx:Class owlx:name="&admin;Course"/>
            <ruleml:var>x</ruleml:var>
        </swrl:classAtom>
    </ruleml:_body>
    <ruleml:_head>
        <swrlx:classAtom>
            <owlx:Class owlx:name="&lab;Lecture"/>
            <ruleml:var>x</ruleml:var>
        </swrl:classAtom>
    </ruleml:_head>
</ruleml:imp>

```



```

    </swrl:classAtom>
  </ruleml:_head>
</ruleml:imp>

</swrlx:Ontology>

```

4. *R* in the SEKT mapping language as given by the Alignment API:

```

MappingDocument (<"")
  source (<"http://book.ontologymatching.org/exercise/admin.owl">)
  target (<"http://book.ontologymatching.org/exercise/lab.owl">)
  classMapping( <"#s075689">
    bidirectional
    <"&admin;Office">
    <"http://book.ontologymatching.org/exercise/lab.owl#Room">
  )

  classMapping( <"#s035024">
    unidirectional
    <"http://book.ontologymatching.org/exercise/admin.owl#Student">
    <"http://book.ontologymatching.org/exercise/lab.owl#PhDStudent">
  )

  attributeMapping( <"#s074544">
    bidirectional
    <"http://book.ontologymatching.org/exercise/admin.owl#room">
    <"http://book.ontologymatching.org/exercise/lab.owl#office">
  )

  classMapping( <"#s026252">
    bidirectional
    <"http://book.ontologymatching.org/exercise/admin.owl#Faculty">
    <"http://book.ontologymatching.org/exercise/lab.owl#FullProfessor">
  )

  attributeMapping( <"#s006240">
    bidirectional
    <"http://book.ontologymatching.org/exercise/admin.owl#teaches">
    <"http://book.ontologymatching.org/exercise/lab.owl#teaches">
  )

  classMapping( <"#s016467">
    bidirectional
    <"http://book.ontologymatching.org/exercise/admin.owl#Course">
    <"http://book.ontologymatching.org/exercise/lab.owl#Lecture">
  )

)

```

5. *R* in SKOS as given by the Alignment API:

```

<?xml version='1.0' encoding='utf-8' standalone='no' ?>
<!DOCTYPE rdf:RDF [
  <!ENTITY admin "http://book.ontologymatching.org/exercise/admin.owl#" >
  <!ENTITY lab "http://book.ontologymatching.org/exercise/lab.owl#" >
]
>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:skos="http://www.w3.org/2008/05/skos#"
  <skos:Concept rdf:about="&admin;Office">
    <skos:exactMatch rdf:resource="&lab;Room"/>
  </skos:Concept>

  <skos:Concept rdf:about="&admin;room">
    <skos:exactMatch rdf:resource="&lab;office"/>
  </skos:Concept>

```

```

<skos:Concept rdf:about="&admin;Student">
  <skos:broadMatch rdf:resource="&lab;PhDStudent"/>
</skos:Concept>

<skos:Concept rdf:about="&admin;Course">
  <skos:exactMatch rdf:resource="&lab;Lecture"/>
</skos:Concept>

<skos:Concept rdf:about="&admin;teaches">
  <skos:exactMatch rdf:resource="&lab;teaches"/>
</skos:Concept>

<skos:Concept rdf:about="&admin;Faculty">
  <skos:exactMatch rdf:resource="&lab;FullProfessor"/>
</skos:Concept>

</rdf:RDF>

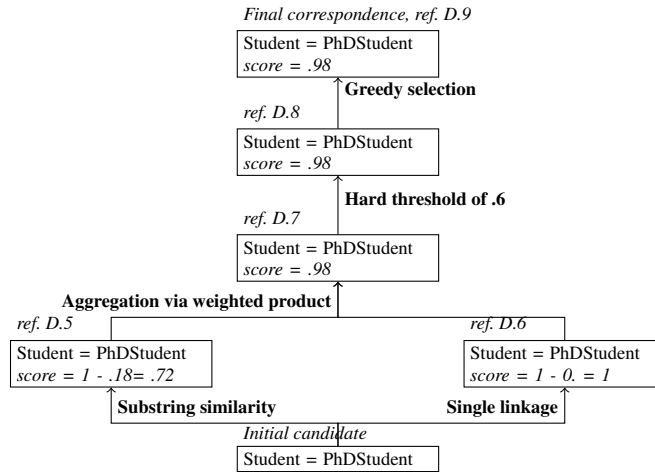
```

6. The discussion on the advantages of alignment formats is rather generic. The OWL representation provides a merge between the two ontologies. Once loaded, this OWL does not distinguish an alignment and two ontologies, but only one ontology. The other representations maintain the two ontologies and represent the alignments as mappings (C-OWL, SEKT or SKOS) or rules (SWRL). SWRL and C-OWL use oriented mappings (from source to target) while the others are not oriented. OWL, C-OWL and SWRL are immediately operational for reasoning and expressed in a formally defined language, while SEKT and SKOS require a processor. With respect to the RDF format, all these representations loss information such as the metadata on the alignments or the confidence measures.

D.8 Explaining alignments

Solution D.14 (Alignment explanation). Consider the ontologies of Fig. C.2 and the architecture and a solution devised for Problem D.10.

1. A decision flow for the correspondence $\text{Student} = \text{PhDStudent}$ is provided below. Following the discussion of Solution D.11, the relation of this correspondence is not specific enough, and hence, it is an incorrect one. The above presented decision flow would be that of a matching system able to compute only equivalence correspondences.
2. However, a system able to compute subsumption relations, e.g., through a common suffix rule, hence being able to generate the $\text{Student} \geq \text{PhDStudent}$ as an initial candidate, would manage to deliver it as the final and correct correspondence with the same explanation flow.
3. Both the terminological substring matcher and the instance-based single linkage matcher agree that there is a relation between entities Student and PhDStudent by assigning them a high similarity, which was further reinforced through an aggregation via weighted product. This correspondence, with the similarity of .98, passed the hard threshold of .6 and was also retained by the greedy alignment extraction algorithm. Hence, it was delivered as the final one. The (in)correctness depends of the ability of a matching system to compute a specialisation relation.



D.9 Processing alignments

Solution D.15 (Merging ontologies).

1. The merge between o and o' according to R can be expressed with OWL import by:

```

<rdf:RDF xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#">

  <owl:Ontology rdf:about="">
    <rdfs:comment>Matched ontologies</rdfs:comment>
    <owl:imports
      rdf:resource="http://book.ontologymatching.org/exercise/admin.owl"/>
    <owl:imports
      rdf:resource="http://book.ontologymatching.org/exercise/lab.owl"/>
  </owl:Ontology>

  <owl:Class
    rdf:about="http://book.ontologymatching.org/exercise/admin.owl#Office">
    <owl:equivalentClass
      rdf:resource="http://book.ontologymatching.org/exercise/lab.owl#Room"/>
  </owl:Class>
  <owl:Class
    rdf:about="http://book.ontologymatching.org/exercise/lab.owl#PhDStudent">
    <rdfs:subClassOf
      rdf:resource="http://book.ontologymatching.org/exercise/admin.owl#Student"/>
  </owl:Class>
  <owl:ObjectProperty
    rdf:about="http://book.ontologymatching.org/exercise/admin.owl#room">
    <owl:equivalentProperty
      rdf:resource="http://book.ontologymatching.org/exercise/lab.owl#office"/>
  </owl:ObjectProperty>
  <owl:Class
    rdf:about="http://book.ontologymatching.org/exercise/admin.owl#Faculty">
    <owl:equivalentClass rdf:resource="
      http://book.ontologymatching.org/exercise/lab.owl#FullProfessor"/>
  </owl:Class>

```

```

<owl:ObjectProperty
  rdf:about="http://book.ontologymatching.org/exercise/admin.owl#teaches">
  <owl:equivalentProperty
    rdf:resource="http://book.ontologymatching.org/exercise/lab.owl#teaches"/>
  </owl:ObjectProperty>
<owl:Class
  rdf:about="http://book.ontologymatching.org/exercise/admin.owl#Course">
  <owl:equivalentClass
    rdf:resource="http://book.ontologymatching.org/exercise/lab.owl#Lecture"/>
  </owl:Class>
</rdf:RDF>

```

This is what is indeed rendered by the `OWLAxiomsRendererVisitor` delivered with the Alignment API.

- It can also be expressed as an integrated OWL ontology like the one of Figure D.3, in which `Faculty` replaces `FullProfessor`, `Staff` in o has been renamed `Staff#2`, `Room` replaces `Office`, `Lecture` replaces `Course`, `office` replaces `room` and both `teaches` have been unified.

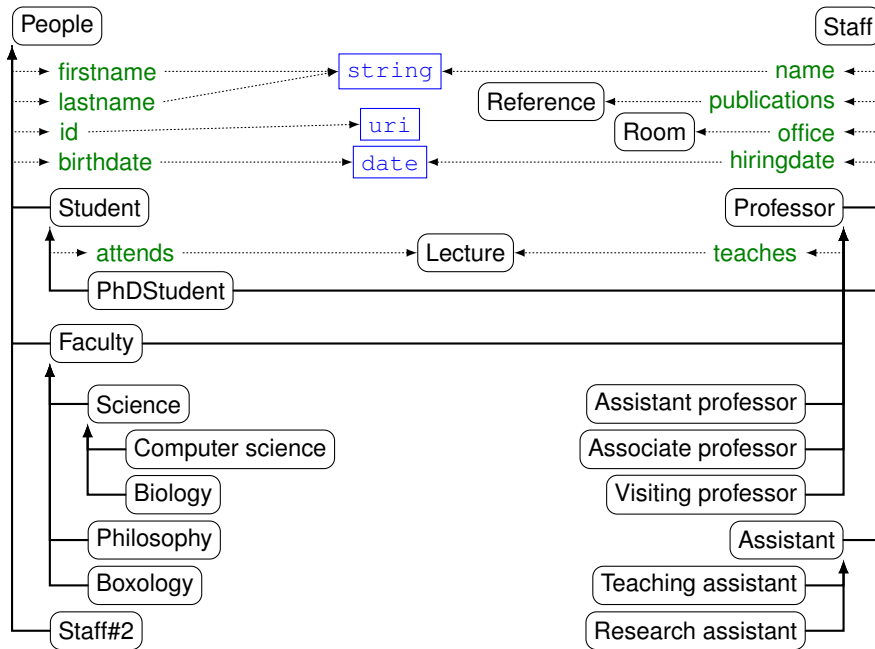


Fig. D.3. Merge of the ontology o and o' with regard to the reference alignment R .

Solution D.16 (Data translation). Assume that we want to transform the data instances from ontology o into instances of o' . Consider the data instances in the first table of Problem C.6.

1. The transformation of data instances in the first table of Problem C.6 with regard to the reference alignment R would lead to the creation of 6 Full professor with no name, no hiring date and no office³.
2. There may be two ways to perform the translation using the functions of the Alignment API. If the data is or can be expressed in XML, then the Alignment API is able to generate an XSLT transformation thanks to its `XSLTRendererVisitor` serialiser. This transformation may then be applied to the XML data. Otherwise, it is necessary to use the alignment from Java for performing the adequate actions. Here is the core of a program that does this with the Alignment API and Jena:

```

public void transform( OntModel onto1, OntModel onto2, Alignment al ) {
    Hashtable<String,Individual> ht = new Hashtable<String,Individual>();
    URI base = al.getOntology2URI();
    // For each correspondence involving a class, transform individuals
    for ( Cell c : al ) {
        OntClass cl = onto1.getOntClass( c.getObject1AsURI( al ).toString() );
        if ( cl != null && ( c.getRelation().getRelation().equals("=")
            || c.getRelation().getRelation().equals("<") ) ) {
            OntClass cl2 = onto2.getOntClass(
                c.getObject2AsURI( al ).toString() );
            for ( Individual i : listInstances( cl ) ) {
                String id = base.getScheme()+base.getSchemeSpecificPart()
                    +URLEncoder.encode( i.getLabel(""), "UTF-8" );
                Individual o = onto2.createIndividual( id, cl2 );
                ht.put( i.getURI(), o );
                System.out.println(o.getURI()+" rdf:type "+o.getOntClass()+" .");
            }
        }
    }
    // For each correspondence involving a property, transform statements
    for ( Cell c : al ) {
        OntProperty pr =
            onto1.getOntProperty( c.getObject1AsURI( al ).toString() );
        if ( pr != null ) {
            OntProperty pr2 =
                onto2.getOntProperty( c.getObject2AsURI( al ).toString() );
            StmtIterator stmtit =
                onto1.listStatements( (Resource)null, pr, (RDFNode)null );
            for ( ; stmtit.hasNext(); ) {
                Statement st = stmtit.nextStatement();
                Individual subj = (Individual)ht.get( st.getSubject().getURI() );
                if ( subj != null ) {
                    RDFNode obj = st.getObject();
                    if ( obj.isResource() ) {
                        obj = (Individual)ht.get( obj.asResource().getURI() );
                    }
                    if ( obj != null && !obj.isAnon() ) { // do not translate
                        onto2.createStatement( subj, pr2, obj );
                        System.out.println( subj+" "+pr2+" "+obj+" .");
                    }
                }
            }
        }
    }
}

```

A working example is given in the exercise web site.

Solution D.17 (Mediation).

³ It would have been better to ask for transferring from o' to o , to have the rooms instead of the ids in o 's table. Even this would only allow to transfer PhD and Full professors.

1. The translation of the query ‘SELECT ?x.room WHERE Faculty(?x)’ expressed with respect to ontology o involves finding the correspondences for attribute room and class Faculty. These are respectively office and FullProfessor. Hence the translated query is ‘SELECT ?x.office WHERE FullProfessor(?x)’, Rewritten in terms of SPARQL, the query is:

```
SELECT ?y
WHERE { ?x rdf:type Faculty AND. ?x office ?y }
```

and its translation is:

```
SELECT ?y
WHERE { ?x rdf:type FullProfessor . ?x room ?y }
```

2. When evaluated against the second table, the answer will be $\{\langle B12 \rangle\}$ because the only instance of the class FullProfessor in the second ontology is Federico di Guava whose office is B12 (or rather the room identified by B12).
3. A mediator able to perform the transformations of any query q based on the reference alignment R will have to transform the query with regard to R ($q' = TransformQuery(q, R)$), it will then have to evaluate the result ($r' = Eval(q')$) and to return the results by translating them with regard to the converse of R ($r = Translate(r', Invert(R))$). Hence, the full mediator would be implemented by:

$$Translate(Eval(TransformQuery(q, R)), Invert(R))$$