User Study: Gathering Intented Match Results

Database Group, Stanford University
{melnik}db.stanford.edu

Feb 16, 2001

This user study attempts to collect various intended match results for a set of schema matching problems. General remarks:

1. The information provided about the source and target schemas is intentionally vague. Imagine a plausible scenario and try to map elements in both schemas according to the scenario you have in mind.
2. You don't have to match every element on the left and every one on the right, partial mappings are fine (if consistent with the scenario you have in mind).
3. $m : n$ correspondences between schema elements are welcome.
4. No mapping expressions are required.
5. The elements in the left and right schemas are numbered. Please fill out the table following every problem as shown in the example below.

When you are finished, please return your results to my office (438). Thanks a lot!

Example

This example shows schematically two XML schemas.

```
1  Cust                                    a  Customer
2    C#                                    b    CustID
3    CName                                 c    Company
4    FirstName                             d    Contact
5    LastName                              e    Phone
```

Many possible match results are conceivable for the schemas. Two of them are depicted below:

Left element(s)	Right element(s)
1	a
2	b
3	c
4,5	d

Left element(s)	Right element(s)
2	b
3	c,d

1 Problem: matching of XML schemas

```
1  <Schema name="Schema 1"
       xmlns="urn:microsoft-com:xml-data">
2  <ElementType name="AccountOwner">
3      <element type="Name"/>
4      <element type="Address"/>
5      <element type="Birthdate"/>
6      <element type="TaxExempt"/>
   </ElementType>
7  <ElementType name="Address">
8      <element type="Street"/>
9      <element type="City"/>
10     <element type="State"/>
11    <element type="ZIP"/>
   </ElementType>
</Schema>
```

```
a  <Schema name="Schema 2"
       xmlns="urn:microsoft-com:xml-data">
b  <ElementType name="Customer">
c      <element type="Cname"/>
d      <element type="CAddress"/>
e      <element type="CPhone"/>
   </ElementType>
f  <ElementType name="CustomerAddress">
g      <element type="Street"/>
h      <element type="City"/>
i      <element type="USState"/>
j      <element type="PostalCode"/>
   </ElementType>
</Schema>
```

Left element(s)	Right element(s)

2 Problem: matching of XML schemas

```
1  HOUSE
2    ADDRESS
3    COUNTY
4    PRICE
5    DESCRIPTION
6    CONTACT-INFO
7      OFFICE-INFO
8        OFFICE-NAME
9        OFFICE-PHONE
10     AGENT-INFO
11       AGENT-NAME
12       AGENT-PHONE
```

```
a  listing
b    location
c    area
d    price
e    comments
f    contact
g      agent
h        name
i        office
j      brokerage
k        name
l        phone
m    house-style
```

Left element(s)	Right element(s)

3 Problem: matching of XML schemas

```
1  <E n="Library">
2    <E n="Item">
3      <e n="ISBN"/>
4      <e n="Author"/>
5      <e n="Title"/>
6      <e n="Year"/>
    </E>
7    <E n="Author">
8      <e n="FirstName"/>
9      <e n="LastName"/>
     </E>
10   <E n="BorrowedItems">
11     <e n="Item"/>
12     <e n="Borrower"/>
     </E>
13   <E n="Borrower">
14     <e n="FirstName"/>
15     <e n="LastName"/>
  </E>
</E>
```

```
a  <E n="Collection">
b    <E n="Document">
c      <e n="Identifier"/>
d      <e n="Creator"/>
e      <e n="Contributor"/>
f      <e n="Publisher"/>
g      <e n="Title"/>
h      <e n="Year"/>
    </E>
i    <E n="Creator">
j      <e n="Name"/>
     </E>
k    <E n="Name">
l       <e n="first"/>
m       <e n="last"/>
     </E>
n    <E n="Publisher">
o      <e n="Address"/>
p      <e n="Name"/>
     </E>
   </E>
```

Left element(s)	Right element(s)

4 Problem: matching XML schemas using data instances

In this problem, XML tags in both schemas need to be matched given two instances of schemas. The numbering enumerates all different tags on the left and on the right. Remember, you are matching the tag names, the particular instance values provide the hints for the matching process.

```
1  <amazon>
2    <item>
3      <title>Sony DCR-PC100
             Digital HandyCam Camcorder</title>
5      <listPrice>1899.99</listPrice>
6      <ourPrice>1699.00</ourPrice>
7      <youSave>200.00</youSave>
8      <review>
9         <avgReview>4.5</avgReview>
10        <numOfReviews>20</numOfReviews>
       </review>
11     <availability>On Order;
          usually ships within 1-2
          weeks</availability>
12     <features>
13        <zoom>10x optical zoom</zoom>
          <zoom>120x digital zoom</zoom>
14        <lcd>2.5inch LCD</lcd>
15        <other>4 MB Memory Stick included</other>
       </features>
     </item>
   </amazon>
```

```
a  <yahoo>
b    <productInfo>
c      <id>Sony DCR-PC100</id>
d      <merchantPrice>1799.94</merchantPrice>
e      <rating>
f         <userRating>3.5</userRating>
g         <userReviews>7</userReviews>
       </rating>
h      <description>
i         <LCDScreenSize>2.5in</LCDScreenSize>
j         <opticalZoom>10x</opticalZoom>
k         <special>4MB Memory Stick</special>
       </description>
     </productInfo>
   </yahoo>
```

Left element(s)	Right element(s)

5 Problem: matching XML schemas using data instances

Same problem as the previous one: XML tags in both schemas need to be matched given two instances of schemas. The numbering enumerates all different tags on the left and on the right. Remember, you are matching the tag names, the particular instance values provide the hints for the matching process.

```
1  <db1>
2    <Faculty>
3      <SSN>234-56-7890</SSN>
4      <Facu_Name>Richie Solomon</Facu_Name>
5      <Salary>170000</Salary>
     </Faculty>
6    <Student>
7      <Stud_ID>7206362</Stud_ID>
8      <Stud_Name>Teresa Lista</Stud_Name>
9      <Stipend>23000</Stipend>
10     <Tel>408-973 0110</Tel>
     </Student>
   </db1>
```

```
a  <db2>
b    <Personnel>
c      <ID>234-56-7890</ID>
d      <Name>Solomon, Richie</Name>
e      <Address>Sand Hill Road, Menlo Park, CA</Address>
f      <W_phone>(408) 495 8423</W_phone>
g      <H_phone>(650) 923 4193</H_phone>
     </Personnel>
     <Personnel>
      <ID>7206362</ID>
      <Name>Lista, Teresa</Name>
      <Address>Cotton St, Palo Alto, CA</Address>
      <W_phone>(408) 973 0110</W_phone>
      <H_phone>(650) 198 2424</H_phone>
    </Personnel>
  </db2>
```

Left element(s)	Right element(s)

6 Problem: matching of XML catalogues using data instances

In this problem, catalogue entries in both schemas need to be matched given two instances of schemas. The numbering enumerates all different catalog categories on the left and on the right.

```
  <yahoo>
1   <cat id="Home">
2     <cat id="Electronics and Photography">
3       <cat id="Television and Video">
4         <cat id="Camcorders">
5           <cat id="By Format - DV">
              <product name="SONY DCR-PC100"/>
            </cat>
          </cat>
        </cat>
6       <cat id="Photography">
7         <cat id="Brands - Polaroid">
            <product name="POLAROID PDC 3000"/>
          </cat>
        </cat>
      </cat>
8     <cat id="Movies">
9       <cat id="Comedy">
10        <cat id="Parody">
11          <cat id="Science Fiction">
             <product name="Mars Attacks!"/>
            </cat>
          </cat>
12        <cat id="Satire">
            <product name="The Graduate"/>
          </cat>
        </cat>
      </cat>
    </cat>
  </yahoo>
```

```
  <epinions>
a   <cat id="Home">
b     <cat id="Electronics">
c       <cat id="Video">
d         <cat id="Camcorders">
            <product name="Sony DCR-PC100"/>
          </cat>
        </cat>
e       <cat id="Photo">
f         <cat id="Cameras">
            <product name="Minolta Maxxum 9"/>
          </cat>
        </cat>
      </cat>
g     <cat id="Arts and Entertainment">
h       <cat id="Movies">
i         <cat id="Video">
            <product name="The Graduate"/>
            <product name="Mars Attacks!"/>
          </cat>
        </cat>
      </cat>
    </cat>
  </epinions>
```

Left element(s)	Right element(s)

7 Problem: matching of relational schemas

The numbering enumerates tables and columns in both schemas.

```
1  CREATE TABLE Personnel (
2    Pno int,
3    Pname string,
4    Dept string,
5    Born date,
     UNIQUE perskey (Pno)
   );
```

```
a  CREATE TABLE Employee (
b    EmpNo int PRIMARY KEY,
c    EmpName varchar(50),
d    DeptNo int REFERENCES Department,
e    Salary dec(15,2),
f    Birthdate date
   );

g  CREATE TABLE Department (
h    DeptNo int PRIMARY KEY,
i    DeptName varchar(70)
   );
```

Left element(s)	Right element(s)

8 Problem: matching of relational schemas

Table on the right presents a previous version of the schema shown on the left. The left schema is the evolved schema. Match the new version of the schema onto the old one!

```
1  CREATE TABLE Address (
2    Id int PRIMARY KEY,
3    Street string,
4    City string,
5    PostalCode int
   );

6  CREATE TABLE Professor (
7    Id int PRIMARY KEY,
8    Name string,          # name
9    Sal double,           # salary
10   addr int              # address
   );

11 CREATE TABLE Student (
12   Name string,          # name
13   GPA double,           # grade point avg
14   Yr int                # year of studies
   );

15 CREATE TABLE PayRate (
16   Rank int PRIMARY KEY, # project rank
17   HrRate double         # hourly pay rate
   );

18 CREATE TABLE WorksOn (
19   Name string,          # name of student
20   Proj string,          # project name
21   Hrs int,              # hours spent
22   ProjRank int          # project rank
   );
```

```
a  CREATE TABLE Professor (
b    Id int PRIMARY KEY,
c    Name string,
d    Salary double,
e    Address string
   );

f  CREATE TABLE Student (
g    Name string,
h    GradePointAverage double,
i    Year int
   );

j  CREATE TABLE WorksOn (
k    StudentName string,
l    Project string,
m    Expenses double
   );
```

Left element(s)	Right element(s)

9 Problem: matching of relational schemas

Left schema is the same as in the previous example. It deals with professors, students, and provides the information about who worked on which project for how long. Moreover, the schema contains information about payment of professors and students.

Right schema captures general personnel information.

Hints for interpretations of schemas:

- `WorksOn(ProjRank)` may or may not be foreign key for `PayRate(Rank)`
- `WorksOn(Name)` may or may not be foreign key for `Professor(Name)` or `Student(Name)`
- `Student(Yr)` may or may not be foreign key of `PayRate(Rank)`
- Pay rate of a student may or may not depend on the year and/or his/hers grades

```
1  CREATE TABLE Address (
2    Id int PRIMARY KEY,
3    Street string,
4    City string,
5    PostalCode int
   );

6  CREATE TABLE Professor (
7    Id int PRIMARY KEY,
8    Name string,          # name
9    Sal double,           # salary
10   addr int              # address
   );

11 CREATE TABLE Student (
12   Name string,          # name
13   GPA double,           # grade point avg
14   Yr int                # year of studies
   );

15 CREATE TABLE PayRate (
16   Rank int PRIMARY KEY, # project rank
17   HrRate double         # hourly pay rate
   );

18 CREATE TABLE WorksOn (
19   Name string,          # name of student
20   Proj string,          # project name
21   Hrs int,              # hours spent
22   ProjRank int          # project rank
   );
```

```
a  CREATE TABLE Personnel (
b    Id int PRIMARY KEY,
c    Name string,
d    Sal double,          # salary
e    Addr string          # address
   );
```

Left element(s)	Right element(s)