

Scenario dopo la riunione del 15 ottobre 2002

A seguito della riunione si è provato a delineare un nuovo scenario.

1. Intervento di trasformazione del Common Thesaurus.

Dopo la fase di clusterizzazione si potrebbe pensare di intervenire sul Common Thesaurus al fine di mettere in evidenza le relazioni terminologiche esistenti tra i cluster. All'interno del CT, infatti, anche dopo la fase di calcolo dei cluster, le informazioni relative alle relazioni tra i nomi delle classi o degli attributi, mantengono il riferimento alle sorgenti locali. Questo impedisce di rendere esplicite le relazioni che si formano dopo la creazione dei cluster.

L'intervento sul CT, potrebbe analizzare le relazioni in esso contenute e dividere quelle intra-cluster da quelle inter-cluster. Le relazioni inter-cluster potrebbero essere modificate in modo da rendere esplicita la relazione tra i cluster coinvolti. In altre parole si potrebbe modificare il riferimento alle classi locali con i cluster in cui sono mappati.

Proviamo a spiegare con un esempio.

Supponiamo di avere le sorgenti *S1* e *S2* così composte:

S1

B1 ($y_{11}, \dots, y_{1n}$)

C1 ($x_{11}, \dots, x_{1n}$)

S2

B2 ($y_{21}, \dots, y_{2n}$)

C2 ($x_{21}, \dots, x_{2n}$)

All'interno del *Common Thesaurus* potrebbero esserci tra le altre le seguenti relazioni:

S1.C1.x ₁₁ SYN S2.B2.y ₂₃
S1.C1.x ₁₃ SYN S1.B1.y ₁₃

I cluster potrebbero essere così formati (tra parentesi ci sono gli attributi globali):

CL1 ($f_{11}, \dots, f_{1n}$)

S1.C1

S2.C2

Mapping Table

$f_{11} \Leftrightarrow S1.C1.x_{11}$

...

CL2 ($f_{21}, \dots, f_{2n}$)

S1.B1

S2.B2

Mapping Table

...

$f_{22} \Leftrightarrow S2.B2.y_{23}$

...

Analizzando il Common Thesaurus si osserva che la prima relazione è inter-cluster, mentre la seconda è intra-cluster. L'intervento che si propone, porterebbe a modificare il CT nel seguente modo:

CL1.f₁₁ SYN CL2.f₂₂.

Questa modifica consentirebbe di conoscere le relazioni esistenti tra i cluster.

2. Integrazione della nuova sorgente.

Il processo di integrazione potrebbe vedere impegnati la nuova sorgente, la GVV e il nuovo Common Thesaurus modificato come spiegato al punto precedente (eventualmente anche le Mapping Table delle classi globali e il precedente CT). La nuova proposta si basa sull'osservazione che lo schema globale è la sintesi di tutta la conoscenza portata dalle varie sorgenti locali quindi risulta essere il migliore candidato per il confronto. Probabilmente anche in questo nuovo scenario sono necessari un set di indici o coefficienti per aiutare il progettista a valutare quando conviene ripetere da capo il processo di integrazione.

Continuando con l'esempio precedente si potrebbe volere inserire la sorgente S3:

S3

B3 (y₃₁, ..., y_{3n})

C3 (x₃₁, ..., x_{3n})

D3 (z₃₁, ..., z_{3n})

Il confronto verrebbe condotto con:

GVV

CL1 (f₁₁, ..., f_{1n})

CL2 (f₂₁, ..., f_{2n})

A questo punto bisogna studiare quale procedimento eseguire per integrare la nuova sorgente nello schema.

Una possibilità potrebbe essere quella di confrontare in modo diretto lo schema della nuova sorgente con la GVV. Per far questo bisognerebbe inserire nel CT (modificato e relativo solamente alla GVV) le relazioni (RT) tra gli attributi e le classi di appartenenza sia della GVV che della nuova sorgente, ad esempio:

CL1 RT f₁₁

...

C3 RT x₃₁

...

Il passo successivo potrebbe essere un nuovo calcolo dei cluster per capire dove le classi della nuova sorgente si vanno a mappare.

Esempio di riferimento.

Il seguente esempio è stato preso in parte dalla tesi di Marri Elisa e in parte dal file XML sRoma.xml.

Sorgente UNIVERSITY (*UNI*) (University)

Research_Staff (name, e_mail, dept_code, section_code)

School_Member (name, faculty, year)

Department (dept_name, dept_code, budget)

Section (section_name, section_code, length, room_code)

Room(room_code, seats_number, notes)

Sorgente COMPUTER_SCIENCE (*CS*) (Computer_Science)

CS_Person (first_name, last_name)

Professor:CS_Person (belongs_to: Office, rank)

Student:CS_Person (year, takes:set<Course>, rank)

Office (description, address:Location)

Location (city, street, number, county)

Course (course_name, taught_by:Professor)

Sorgente TAX POSITION (*TP*) (tax_position_xml)

Student (name, student_code, faculty_name, tax_fee)

ListOfStudent (Student:set<Student>)

Sorgente UNIVERSITY (UND) (University)

```
interface Research_Staff
( source relational University
  extent Research_Staff
  key (name)
  foreign_key (dept_code)
  references Department (dept_code);
  foreign_key (section_code)
  references Section (section_code)
{ attribute string name;
  attribute string e_mail;
  attribute integer dept_code;
  attribute integer section_code; }
```

```
interface School_Member
( source relational University
  extent School_Member
  keys (name))
{attribute string name;
  attribute string faculty;
  attribute integer year; };
```

```
interface Department
( source relational University
  extent Departments
  key dept_code )
{ attribute string dept_name;
  attribute integer dept_code;
  attribute integer budget;};
```

```
interface Section
( source relational University
  extent Section
  key section_code
  foreign_key (room_code)
  references Room (room_code))
{ attribute string section_name;
  attribute integer section_code;
  attribute integer length;
  attribute integer room_code; };
```

```
interface Room
( source relational University
  extent Room
  key room_code )
{ attribute integer room_code;
  attribute integer seats_number;
  attribute string notes; };
```

Sorgente COMPUTER_SCIENCE (CS) (Computer_Science)

```
interface CS_Person
( source object Computer_Science
  extent CS_Persons
  key (first_name, last_name))
{ attribute string first_name;
  attribute string last_name;};
```

```
interface Professor : CS_Person
( source object Computer_Science
  extent Professors )
{attribute Office belong_to;
  attribute string rank;};
```

```
interface Student : CS_Person
( source object Computer_Science
  extent Students )
{ attribute signed long year;
  attribute set<Course> takes;
  attribute string rank;};
```

```
interface Office
( source object Computer_Science
  extent Offices
  key description )
{ attribute string description;
  attribute Location address;};
```

```
interface Location
( source object Computer_Science
  extent Locations
  keys city, street, county, number)
{ attribute string city;
  attribute string street;
  attribute string county;
  attribute signed long number; };
```

```
interface Course
( source object Computer_Science
  extent Courses
  key course_name )
{ attribute string course_name;
  attribute Professor taught_by; };
```

Sorgente TAX POSITION (TP) (tax_position_xml)

```
interface Student
( source semistructured tax_Position
  extent Students
  key student_code )
{ attribute string name;
  attribute integer student_code;
  attribute string faculty_name;
  attribute integer tax_fee; };
```

```
interface ListOfStudent
( source semistructured
  tax_position_xml
  extent ListOfStudent)
attribute set <Student>
Student;
```

Estrazione relazioni intra-schema $\Rightarrow$ SIMA

University.Department RT University.Research_Staff
University.Section RT University.Research_Staff
University.Room RT University.Section
Computer_Science.Student NT Computer_Science.CS_Person
Computer_Science.Professor NT Computer_Science.CS_Person
tax_position_xml.Student RT tax_position_xml.ListOfStudent
Computer_Science.Course RT Computer_Science.Student
Computer_Science.Office RT Computer_Science.Professor
Computer_Science.Location RT Computer_Science.Office
Computer_Science.Professor RT Computer_Science.Course

Estrazione di relazioni inter-schema $\Rightarrow$ SLIM

Inserimento di relazioni da parte del progettista

University.Room SYN Computer_Science.Location
University.Room SYN Computer_Science.Office

Validazione relazioni tra attributi $\Rightarrow$ SIMB

Computer_Science.CS_Person.first_name NT tax_position_xml.Student.name	[1]
Computer_Science.CS_Person.last_name NT tax_position_xml.Student.name	[1]
Computer_Science.Course.course_name SYN University.Research_Staff.section_name	[1]
University.Research_Staff.dept_code BT Computer_Science.Professor.belongs_to	[0]
University.School_Member.faculty SYN tax_position_xml.Student.faculty_name	[1]

Validazione delle relazioni tra classi ed inferenza di nuove relazioni intensionali

tax_position_xml.ListOfStudent RT University.School_Member

Computer_Science.Course RT University.School_Member

Computer_Science.Professor RT University.Room

Computer_Science.Student RT University.Section

Computer_Science.Professor RT University.Section

Computer_Science.Course RT tax_position_xml.Student

tax_position_xml.ListOfStudent RT Computer_Science.Student

Computer_Science.Professor RT University.Location

Relazioni del Thesaurus

Computer_Science.Student.rank	SYN	Computer_Science.Professor.rank
Computer_Science.Location	BT	Computer_Science.Location.city
Computer_Science.Location	BT	Computer_Science.Office.address
Computer_Science.CS_Person.first_name	NT	tax_position_xml.Student.name
Computer_Science.CS_Person.first_name	NT	University.Research_Staff.name
Computer_Science.CS_Person.first_name	NT	University.School_Member.name
Computer_Science.CS_Person.first_name	RT	Computer_Science.CS_Person.last_name
University.Room	SYN	University.Section.room_code
University.Section.room_code	SYN	University.Room.room_code
University.Room	SYN	University.Room.room_code
University.Department	SYN	Computer_Science.Professor.belongs_to
University.Department	SYN	University.Research_Staff.dept_code
University.Department	SYN	University.Department.dept_name
Computer_Science.Professor	SYN	Computer_Science.Course.taught_by
Computer_Science.CS_Person	BT	Computer_Science.Course.taught_by
Computer_Science.Professor	RT	tax_position_xml.Student.faculty_name
Computer_Science.Professor	RT	University.School_Member.faculty
Computer_Science.Student	SYN	tax_position_xml.Student
Computer_Science.Student	SYN	tax_position_xml.Student.student_code
Computer_Science.Student	SYN	University.School_Member
tax_position_xml.Student	SYN	tax_position_xml.Student.student_code
tax_position_xml.Student	SYN	University.School_Member
University.School_Member	SYN	tax_position_xml.Student.student_code
tax_position_xml.Student	NT	Computer_Science.CS_Person
Computer_Science.CS_Person	BT	tax_position_xml.Student.student_code
University.School_Member	NT	Computer_Science.CS_Person
Computer_Science.Student.year	SYN	University.School_Member.year
University.Research_Staff	NT	Computer_Science.CS_Person
Computer_Science.Location	BT	Computer_Science.Location.county
Computer_Science.CS_Person.last_name	NT	tax_position_xml.Student.name
Computer_Science.CS_Person.last_name	NT	University.Research_Staff.name

Computer_Science.CS_Person.last_name	NT	University.School_Member.name
tax_position_xml.Student.faculty_name	SYN	University.School_Member.faculty
Computer_Science.Course	SYN	Computer_Science.Student.takes
University.Section	SYN	Computer_Science.Student.takes
Computer_Science.Course	SYN	Computer_Science.Course.course_name
Computer_Science.Course	SYN	University.Research_Staff.section_code
Computer_Science.Course	SYN	University.Section
Computer_Science.Course	SYN	University.Section.section_name
University.Section	SYN	Computer_Science.Course.course_name
Computer_Science.Course.course_name	SYN	University.Section.section_name
University.Section	SYN	University.Research_Staff.section_code
University.Section	SYN	University.Section.section_name
tax_position_xml.Student.name	SYN	University.Research_Staff.name
tax_position_xml.Student.name	SYN	University.School_Member.name
University.Research_Staff.name	SYN	University.School_Member.name
University.Department	RT	University.Research_Staff
University.Section	RT	University.Research_Staff
University.Room	RT	University.Section
Computer_Science.Student	NT	Computer_Science.CS_Person
Computer_Science.Professor	NT	Computer_Science.CS_Person
tax_position_xml.Student	RT	tax_position_xml.ListOfStudent
Computer_Science.Course	RT	Computer_Science.Student
Computer_Science.Location	RT	Computer_Science.Office
Computer_Science.Professor	RT	Computer_Science.Course
Computer_Science.Office	RT	Computer_Science.Professor
Computer_Science.Course	RT	tax_position_xml.Student
tax_position_xml.ListOfStudent	RT	Computer_Science.Student
University.Research_Staff	RT	Computer_Science.Course
University.Room	RT	Computer_Science.Course
Computer_Science.Course	RT	University.School_Member
tax_position_xml.ListOfStudent	RT	University.School_Member
Computer_Science.Professor	RT	University.Section
Computer_Science.Student	RT	University.Section

Generazione dei Cluster

Cluster 1 – Department

University.Department

dept_code
dept_name
budget

Cluster 2 – Course

Computer_Science.Course
University.Section

course_name
length
room_code
section_code
taught_by

Cluster 3 – ListOfStudent

tax_position_xml.ListOfStudent

student

Cluster 4 – Locations

Computer_Science.Location

county
city
street
number

Cluster 5 – Persons

Computer_Science.CS_Person
Computer_Science.Professor
Computer_Science.Student
University.Research_Staff
University.School_Member
tax_position_xml.Student

name
rank
e_mail
faculty
dept_code
belongs_to
section_code
student_code
takes
tax_fee
title
year

Cluster 6 – Room

University.Room
Computer_Science.Office

address
code
description
notes
seats_number

Schema Algoritmo

Obiettivo: Splittare il Thesaurus. Modificare e inserire le relazioni inter-cluster in una nuova struttura dati simile al Thesaurus.

Passi dell'algoritmo.

- Al termine della fase di Clustering (o meglio di TUNIM)
- Recupero le relazioni dal Thesaurus
- Recupero le informazioni sui cluster o sulle classi globali
- Per ogni relazione presente nel Thesaurus
 - o Controllo se è intra o inter cluster
 - Nel caso inter-cluster
 - Modifico il PATH NAME riferendolo al cluster (o alla classe globale)
 - Inserisco la nuova relazione nella struttura dati.

Osservazioni:

? Forse è meglio compiere l'analisi dopo TUNIM perché alcune relazioni coinvolgono attributi interni ai cluster e quindi occorre fare riferimento al nome del cluster e ai suoi attributi per trasformare la relazione del Thesaurus.

? Forse conviene lavorare direttamente sulle GlobalClasses e sulle MappingTable per essere sicuri del risultato. Forse è anche più coerente con l'obiettivo.

? Creare una nuova classe che rappresenti il nuovo tipo di relazione che si vuole inserire nel Thesaurus. Ad esempio: GlobalRel, GlobAttrRel, GAttributeRel, sul modello di InterfaceRel, IntAttributeRel, AttributeRel).

? Valutare se togliere dal CT le relazioni modificate o se semplicemente aggiungere al Thesaurus quelle modificate.

? Valutare se modificare anche le relazioni intra-cluster che coinvolgono attributi globali differenti.

? Non si potrebbe sottoporre la GVV al processo di estrazione delle relazioni intra-schema (SIMA) per raggiungere l'obiettivo.

? Invece di creare nuove classi per contenere le nuove relazioni si potrebbe inserirle nel CT con il PATH NAME riferito alla GVV:

(NOMESCHEMAGLOBALE).(NOMECLASSEGLOBALE).(NOMEATTRIBUTOGLOBALE)
che corrisponde all'attuale
(Source).(Interface).(Attribute)

? Forse non serve mettere il nome dello schema globale perché a questo punto è sempre lo stesso, cioè quello dello schema integrato.

? Quando si analizzano le relazioni del CT si può trovare che più di una relazione si modifica nella stessa relazione riferita alle classi globali. Controllare che non ci siano duplicati.

? Tipi di nuove relazioni:

1- CLASSE GLOBALE – CLASSE GLOBALE

UNI.Room RT UNI.Section

GVV.Room RT GVV.Course

UNI.Research_Staff RT CS.Course

GVV.Person RT GVV.Course

2- CLASSE GLOBALE – ATTRIBUTO GLOBALE

CS.Location BT CS.Office.address

GVV.Location BT GVV.Room.address

(Coinvolge due classi globali distinte)

CS.Professor RT UNI.School_Member.faculty

GVV.Person RT GVV.Person.faculty

UNI.Department SYN UNI.Department.dept_name

GVV.Department SYN GVV.Department.dept_name

(Le ultime due coinvolgono la stessa classe globale)

3- ATTRIBUTO GLOBALE – ATTRIBUTO GLOBALE

UNI.Section.room_code SYN UNI.Room.room_code

GVV.Course.room_code SYN GVV.Room.code

(Coinvolge due classi globali distinte)

? Il caso di relazioni tra attributi globali della stessa classe globale o sono banali e pressoché inutili o sono dovuti a un mapping errato.