DATA INTEGRATION AND WAREHOUSING

SEMANTIC INTEGRATION OF SEMI-STRUCTURED AND STRUCTURED DATA SOURCES

submitted to

Boris Glavic

by

Ahad Puthawala (A20345760)

Darpan Patel (A20345898)

Bo Li (A20337971)

TEAM - 17

ABSTRACT

Information from multiple heterogeneous data sources ranging from database integration is to the body in the field challenging questions. In this article, we provide a clever way, information integration, taking into account the semantic heterogeneity. The purpose of this paper is to describe Momis (PICC Environmental multiple information sources) and a plurality of ways to integrate heterogeneous information sources, including queries structured and semi-structured data. We propose to integrate semantic method, which the conceptual framework for each source, the use of a common standard data model and language. Description logic and clustering technology for further discussion. Description logic is basically used to derive a common lexicon (overcome semantic heterogeneity), and clustering techniques are basically used to derive the global mode (for query processing).

1. INTRODUCTION

Sought to provide an integrated access to multiple heterogeneous data sources has become global cooperation and interoperability of information systems a challenging problem. In this case, there are two main fundamental problems. First of all, how to determine whether the source contains semantically related information. Secondly, how to deal with the semantic heterogeneity, to support the integration and unity of the query interface. In order to overcome these complexities, we come up Momis (PICC Environmental multiple information sources) approach to multiple heterogeneous information sources, including the integration and querying structured and semi-structured data. Momis conceptual model based on the following data, or metadata, semantic method sources of information, and information integration following architectural elements[6]:

1) A common object oriented data model, defined according to the ODLI3 language, to describe source schemas for information integration purpose.
2) One or more wrappers, to translate schema descriptions into the common ODLI3 representation.
3) A mediator, and a query processing component, based on two pre-existing tools, namely ARTEMIS and ODB-Tools, to provide I3 architecture for integration and query optimization.

The method consists of a unified extraction and analysis phase and phase. Extraction and analysis phase, a common thesaurus term relationship from the source model, form the basis for identifying different sources using clustering technology semantically similar patterns derived class. Unified stage, cluster semantically similar classes are unified to establish a comprehensive global infrastructure analysts. The purpose of this process is to overcome the absence of a common shared ontology, and to derive a global semi-automatic modes, including all the concepts that belong to the source schema to be integrated. OLCD described using the language of logic plus hierarchical clustering technique (with a complementary target period to allow description language) that allows semi-automatic integration process. Description Logic allows

us to determine the effectiveness of the common lexicon term relationships. Clustering technology allows the desired pattern is unified global automatic identification in different source mode architecture class. In this article, we take into account both information integration of structured and semi-structured data sources. A common thesaurus structure, it has shared the role of the body as the information source. Common vocabulary by analyzing the description source ODLI3 built; by using Description Logic OLCD (with a complementary target language allows description cycle) .The knowledge in the public lexicon, and then use the identification of semantically related information from different sources and in ODLI3 description their integration into the global level. Mapping rules and integrity constraints at the global level to express the relationship between the description and explanation of the source of the definition of integration between the holding. ODB- tools, support OLCD and description logic technology, which allows sources to produce a descriptive analysis of common lexicon, and provides semantic query optimization at the global level, based on the support define mapping rules and integrity constraints.

2. BACKGROUND

In this section, we describe formal background for understanding the architecture of the supporting system and phases of the approach for schema integration

2.1 The ODLI3 language and proposed I3 System Architecture:

In Figure 1, it shows the architecture support system. Structure In this architecture, each of said source is a package called responsible for the translation of data sources into a common ODL_{I3} language translation. In a similar way to convert inquiries made by the wrapper, request from the local OQL_{I3} language performed by a single source. The above package, and I3 mediator, which combines software module integrated and refine the data received from the package. In addition, the mediator queries generated from OQL_{I3} from packaging, formulation of queries from the start of the global pattern. Use descriptive logic technology, we are able to generate translated into automatic mode for a given user query local inquiries. Through the intermediary of a semantic foundation module obtained on description logic elements (ie ODB tools engine) and a cluster member (ie ARTEMIS tool), with a minimum ODL_{I3} interface together[4].

So that S = {S1, S2 ... SN} is a set of integrated models N heterogeneous data sources necessary for. To make it easy to communicate between the source description and intermediary between the engine package, we define a description language called ODL_{I3} language. ODL_{I3} media management system used in the common way to source independent of language. This task will be to translate the package into ODLI3 any particular source language original description language, and add the required mediator, such as the name and type of information source[.

According ODL_{I3}, each source mode Si is, silicone = {text C1 i, ... C2I CMI} collection. Silicon feature CJIϬ a class is the name and attributes. CJI = {ncji, A (CJI)}. Each property ϬA (CJI),

and H = 1,2 ....n, is defined as a pair of ah = {NH, DH}, which is the name of NH and DH, respectively, associated with ah domain[4].

To obtain schema integration, we use ODB- tools, validation and query optimization system based on description logic execution mode.

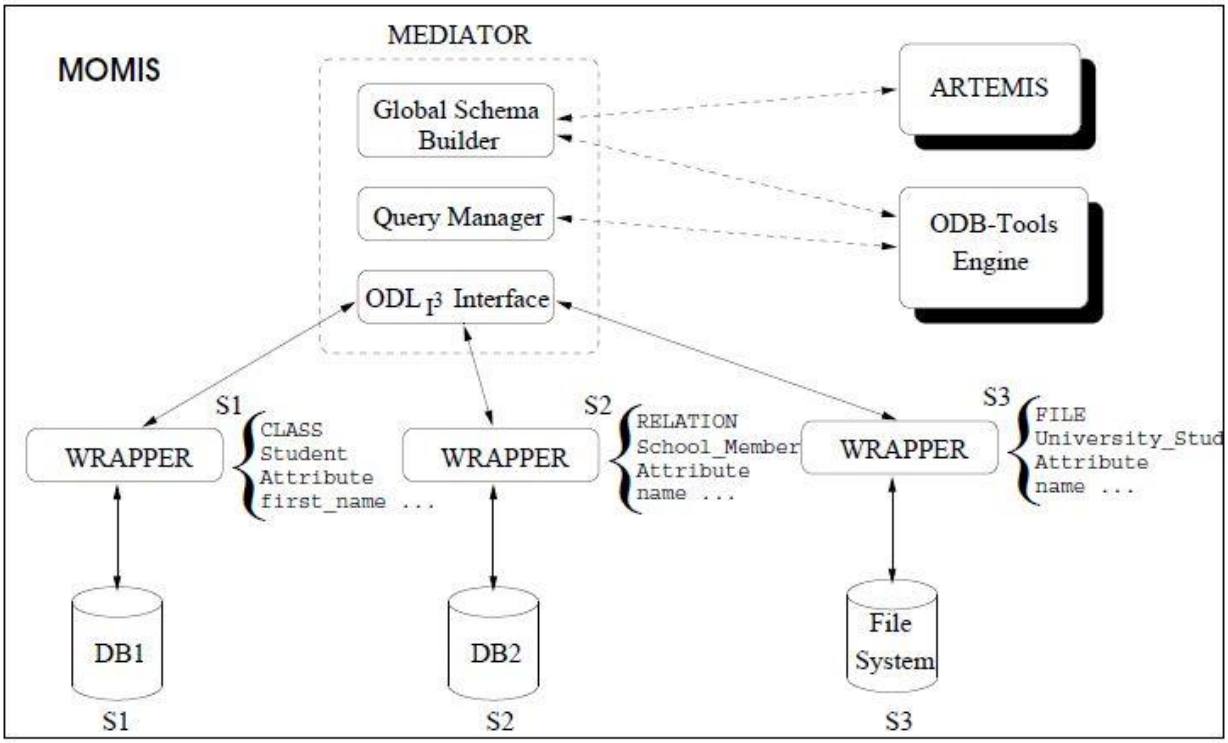

Figure 1: Architecture of the proposed I3 System[1]

2.2 Overview of the Approach

The MOMIS approach to intelligent schema integration is articulated in the following phases:

- Generation of a Common Thesaurus
- Affinity Analysis of ODL_{I3} classes
- Clustering ODL_{I3} classes
- Generation of the mediator Global Schema

1. Generation of a Common Thesaurus

In this step the objective is to generate the Common Thesaurus of schema classes belonging to different ODL_{I3} schemas. By analyzing the structure and context of classes in the schema using OD-tools and the Description Logistics Techniques we can derive Terminological Relationships.

2. Affinity Analysis of ODL_{I3} classes

Affinity analysis is the concept to standardize the relationships that occur between classes. The affinity of two classes is established by affinity coefficients based on the class names and attributes.

3. Clustering ODL_{I3} classes

Classes which have same affinity but belong to different sources are grouped to together in same clusters using clustering techniques like hierarchical clustering techniques. The classes have same or semantic related information that have to be integrated into same cluster.

4. Generation of the mediator Global Schema

The global schema of mediator is constructed by the unification of affinity classes. A class is defined for each cluster, which is representative of all cluster's classes and is defined by the union of their attributes.

2.3 Providing a shared Ontology

In order to illustrate the way our approach works, we will use the following example of integration in the University domain. Researcher uses the Hospital domain as the example to introduce their method of works. The main point of this method is that one department has semi-structured database and another department has relational database which own the same data as the semi-structured database. For integration and query, we consider schema descriptions of the different sources.

The first source is a relational database, University (S1), containing information about the staff and student of the given university. There are five relations: Research_Staff, School_Member, Department, Section and Room. This source is structured database.

The second source Computer_Science (S2) contains information about people who are belonging to the computer science department of the same University. This is an object oriented database and also a semi- structured database. There are six classes: CS_Person, Professor, Student, Division, Location and Course. The stored information in this source on professors and student also gives possibility to retrieve the division of given professor. The Location maintains the division address.

A third source Tax_Postion (S3), derived from the registrar's office. This source stores the information about student's tax_fees.

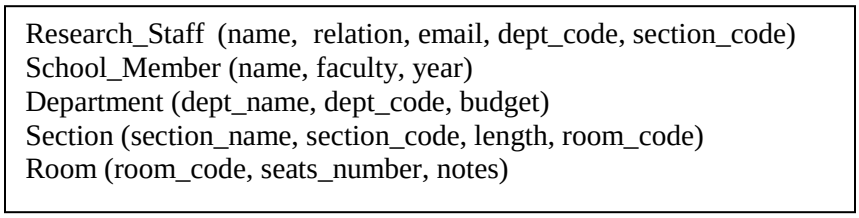

Figure 2: The University Source (S_1)[1]

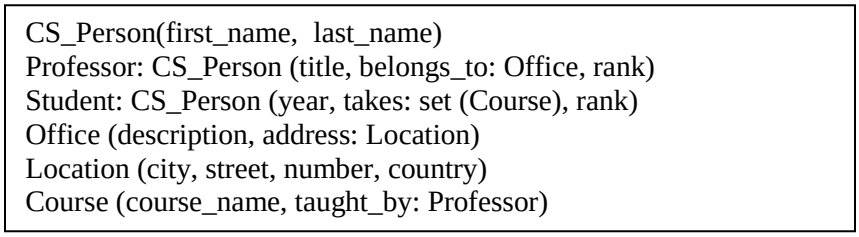

Figure 3: The Computer_Science Source (S_2)[1]

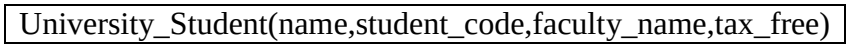

Figure 4: The Tax_Position Source (S_3)[1]

ODLI3 description of the Research_staff and CS_Person [1]

```
interface Research_staff
  (source relational University
    extent Research_Staff
    key name
    foreign_key dept_code, section_code)
{       attribute string name;
        attribute string relation;
        attribute string e_mail;
        attribute integer dept_code;
        attribute integer section_code; };
```

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

3. Generation of Common Thesaurus

The function of Common Thesaurus is to provide the sources of a shared ontology, a dictionary of terminological relationships describing common knowledge about ODLi3 classes and attributes of source schemas is constructed called Common Thesaurus and it has three main relationships will be using for combining database.

- SYN (Synonym-of), also called equivalence. Creating two terms t_i and t_j, with $t_i \neq t_j$, that are considered synonyms. SYN is symmetric, which mean as long as we can prove t_i SYN t_j, we can infer t_j SYN t_i. An example of SYN relationship in our example is <Section SYN Course>.

- BT (Broader Terms) also called hyponymy or generalization. Creating two terms t_i and t_j then t_i has a more general meaning than t_j. BT is not symmetric because the small size group cannot represent the big size group. An example of SYN relationship in our example is <CS_Person BT Student>. The opposite of BT is NT(Narrow Terms): if t_i BT t_j $\rightarrow$ t_j NT t_i

- RT (Related Terms) also called homonymy or positive association. Creating two terms ti and tj which two have a directing relationships and that are generally used together n the same context. RT is symmetric. If ti RT tj, we can infer tj RT ti. An example of RT relationship is <Student RT Course>.

Researcher are using the MOMIS approach to discover the relationships for semi-structure data and with the help of ODB-Tools, they can operate on the progress.

3.1 Automated extraction of relationships

The ODB-Tools automatically extract the set of BT, NT and RT. During the tool transferring ODLI3 into OLCD, it will separately compute BT/NT and RT from generalization hierarchies and aggregation hierarchies. Other RT relationships will be produced by foreign key of relationship. If this is foreign key for both a primary key of original and foreign schema, a BT/NT relationship will be computed. For semi-structured object, ODB-tools extracts RT relationships, due to the nature of relationships defined in the semi-structured data model. Another set of relationships can be automatically extracted exploiting the WordNet lexical system. In this case, synonyms, hypernyms/hyponyms, and related terms can be automatically proposed to the designer, by selecting them according to relationships predefined in the lexical system.

Example 1: Consider the S1 and S2 sources; automatically derived relationships are following:[1]

<Professor RT Office>
<Student RT Course>

<Office RT Location>
<Course RT Professor>
<Research_Staff RT Department>
<Research_Staff RT Section>
<Section RT Room>

Also other relationship computed by WordNet can be given as follows:

<CS_Person BT Professor>
<Student NT CS_Person>

3.2 Integration/Revision of new relationships

After extraction, tool can continue to compute and produce new relationships which can be designed by the user specifically from SYN relationship.

Example 2: New Relationships for classes and attributes[1]

<Research_Staff BT Professor>
<School_member BT Student>
<University_Student BT Student>
<Department BT Office>
<Section SYN Course>
<name BT first_name>
<name BT last_name>
<dept_code BT belongs_to>
<dept_name SYN description>
<section_name SYN course_name>
<faculty SYN faculty_name>

Terminological relationships can connect ODLI3 classes which may contain conflicts relation for the semantics of generalization and equivalence relationships. The terminological relationships transfer the relationship for which SYN to equivalence, BT to generalization and RT to aggregation. Then researcher started creating a new "virtual schema" to solve the contradiction between the structures. A new virtual schema can more clearly indicate the relation diagram produced by ODB-Tools. Next, continuing to produce new relation SYN in to a valid equivalence relationship, researchers define the same structure to both two database classes. Also, for BT relationship, researcher adds the attributes from the generalization class to the specialization class to make the small specialization class has the same data with generalization class. Last part is RT relationship; they create a new aggregation attribute for both two database classes.

3.3 Validation of Relationships

In this part, ODB-Tools is used to validate the terminological relationships by versify the virtual schema. The relationship is valid or invalid is based on the compatibility domains related to attributes. Researcher defined that $a_t = <n_t, d_t>$ and $a_q = <n_q,d_q>$ be two attributes, with n is name and d is domain. Then the ODB-Tools will check that:

$<n_t$ SYN $n_q>$: The relationship is valid if d_t and d_q are equivalent, or if one is a specialization of other;

$<n_t$ BT $n_q>$: The relationship is valid if d_t contains or is equivalent to d_q;

$<n_t$ NT $n_q>$: The relationship is valid if d_t is contains or is equivalent to d_q;

Example 3: According to the thesaurus mentioned in Example 2, ODB tool decide the output validation phase: values of control flags denotes

Value [1]: valid relationship

Value [0]: invalid relationship

<name BT first_name>	[1]
<name BT last_name>	[1]
<dept_code BT belongs_to>	[0]
<dept_name SYN description>	[1]
<section_name SYN course_name>	[1]
<faculty SYN faculty_name>	[1]

3.4 Inference of new relationships

In this part, the ODB-Tool inferences the virtual schema which was created in the revision step into a new generalization and aggregation relation schema.

Example 4: Relationships inferred from explicit relationship of steps 1 to 3.

Inferred relationships

<CS_Person BT Research_Staff>
<CS_Person BT School_membe>
<Section RT Professor>
<Research_Staff RT Course>
<Professor RT Department>
<Professor RT Section>
<Professor RT Course>

<Course RT Room>
<Student RT Section>
<CS_Person BT University_Student>

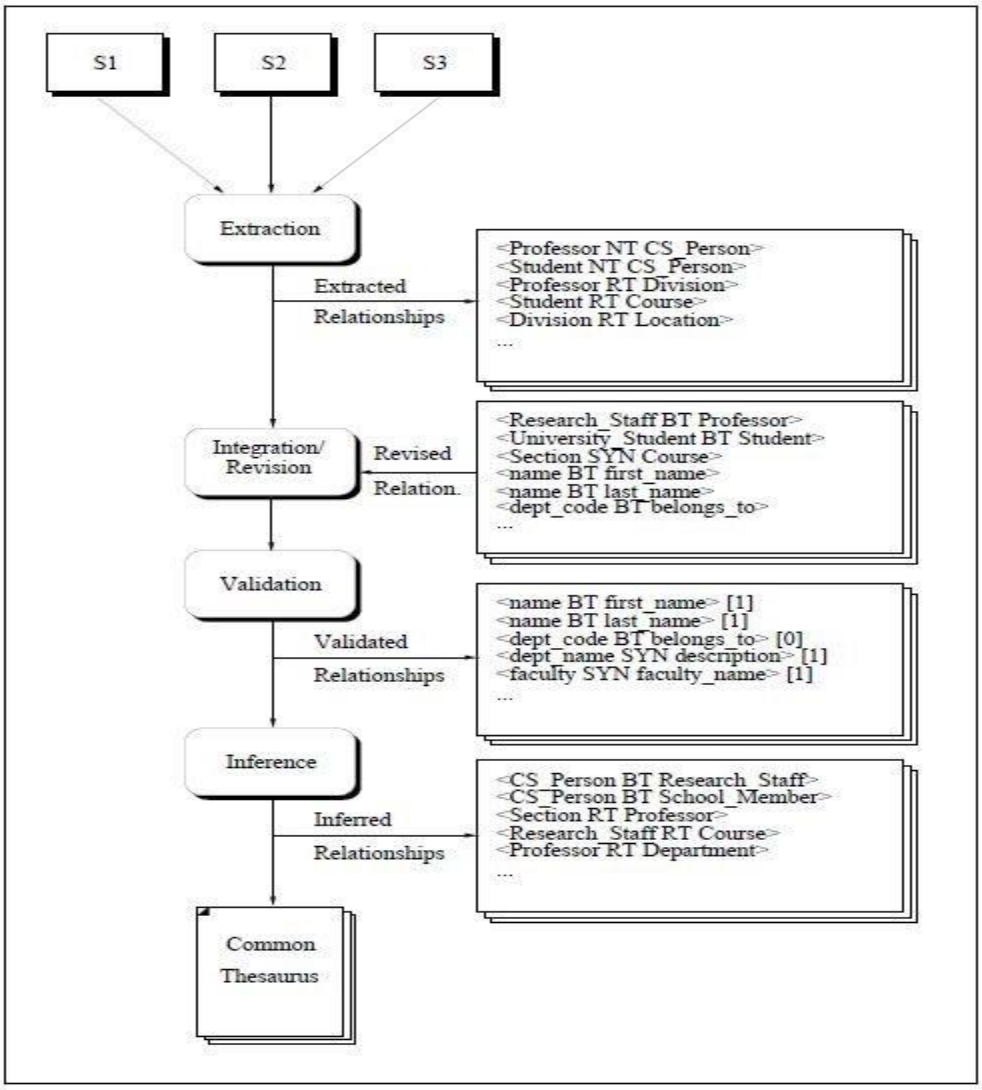

Figure 5: The Generation Process of Common Thesaurus[1]

In Fig.6 Thick Arrows : BT/NT Relationships
Thin Arrows : RT Relationships
Dashed Arrows (------>): Inferred Relationships
Solid Arrows (→): explicitly given Relationships

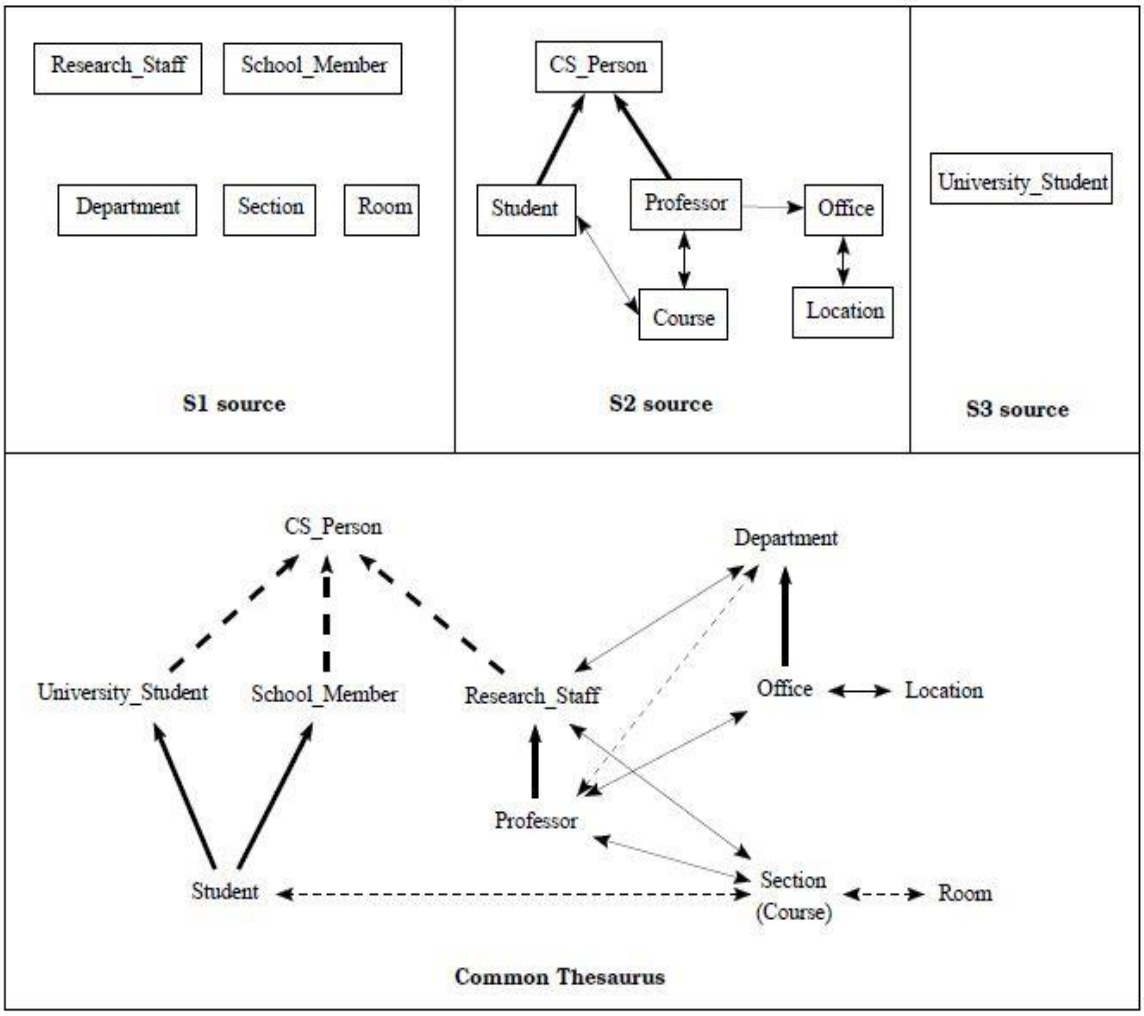

Figure 6: Common Thesaurus for S_1, S_2 and S_3[1]

4. Affinity Analysis of ODL_{I3} classes

To integrate the ODL_{I3} schemas of the different sources in a global schema, we need techniques for identifying the classes that describe the same or semantically related information in different sources schemas. We compare ODL_{I3} classes by means of *affinity coefficients* which allow us to determine the level of semantic relationship between two classes. Comparison of classes with respect to their names (Name Affinity Coefficient) and attributes (Structural Affinity Coefficient), in order to evaluate their level of semantic relationship (Global Affinity Coefficient).

For Relationship, AC (SYN) >= AC (BT/NT) >= AC (RT)

In generalize case AC (SYN) = 1, AC (BT) = AC (NT) = 0.8 and AC (RT) = 0.5

4.1 Name Affinity Coefficient (NAC)

The NAC measures class's affinity with respect to their names. If the two names are connected through oath in the Thesaurus then they have affinity. Their level of affinity is depends upon the length of path, on the type pf relationships involved in this path and on their strengths.

Definition: NAC of two classes c_{ji} and c_{hk} denoted by NA (c_{ji}, chk), where names of the classes are n_{cji} and n_{chk}[1]

$$NA(c_{ji}, c_{hk}) = \begin{cases} 1 & \text{if } n_{c_{ji}} = n_{c_{hk}} \\ \sigma_{\{ji\}1_{\Re}} \cdot \sigma_{12_{\Re}} \cdot \ldots \cdot \sigma_{(m-1)\{hk\}_{\Re}} & \text{if } n_{c_{ji}} \rightarrow^{m} n_{c_{hk}} \text{ AND} \\ & \sigma_{\{ji\}1_{\Re}} \cdot \sigma_{12_{\Re}} \cdot \ldots \cdot \sigma_{(m-1)\{hk\}_{\Re}} \geq \alpha \\ 0 & \text{otherwise} \end{cases}$$

NA () is belongs to [0, 1].
NA () = 0 if a path does not exist between two names in Thesaurus.
NA () = 1 if the name coincide or are synonyms

"The higher the strength of the involved relationships, the greater the affinity of the considered names."

Example: A path between Research_Staff and University_student in Thesaurus. And Relationship is inferred from Researh_Staff →(NT) CS_Person →(BT) University_Student. So, NA (Reasearch_Staff, University_Student) = 0.8*08 = 0.64

4.2 Structural Affinity Coefficient (SAC)

The SAC of two classes c_{ji} and c_{hk} denoted by SA (c_{ji}, chk), is the measure of the affinity of their attributes.[1]

$$SA(c_{ji}, c_{hk}) = \frac{2 \cdot \mid \{(a_t, a_q) \mid a_t \in A(c_{ji}), a_q \in A(c_{hk}), n_t \sim n_q\} \mid}{\mid A(c_{ji}) \mid + \mid A(c_{hk}) \mid} \cdot F_c$$

$$F_c = \frac{\mid \{x \in C \mid flag(x) = 1\} \mid}{\mid C \mid}$$

$$C = \{(a_t, a_q) \mid a_t \in A(c_{ji}), a_q \in A(c_{hk}), \langle a_t \text{ SYN } a_q \rangle \text{ or } \langle a_t \text{ BT } a_q \rangle \text{ or } \langle a_t \text{ NT } a_q \rangle\}$$

where notation $flag(x) = 1$ stands for a positive result and C is the set of validable attribute pairs.

Example: Class Research_Staff in S_1 and class University_Student in S_3 so, SA (Research_Staff, University_Student) = (2*1)/ (5+4) = 0.22.

5. Clustering ODL_{I3} classes

We are employing hierarchical clustering techniques to identify groups of classes having affinity in N sources schemas. It Classify classes into groups at different levels of affinity

The clustering algorithm uses the Global Affinity Coefficient to compute the cluster. The GA () value between the newly defined cluster and each remaining cluster is computed, by keeping the maximum GA () value among the values of every, merged clusters and each reaming cluster in the matrix. The Output of this procedure is Affinity Tree which describe in below Figure.[1]

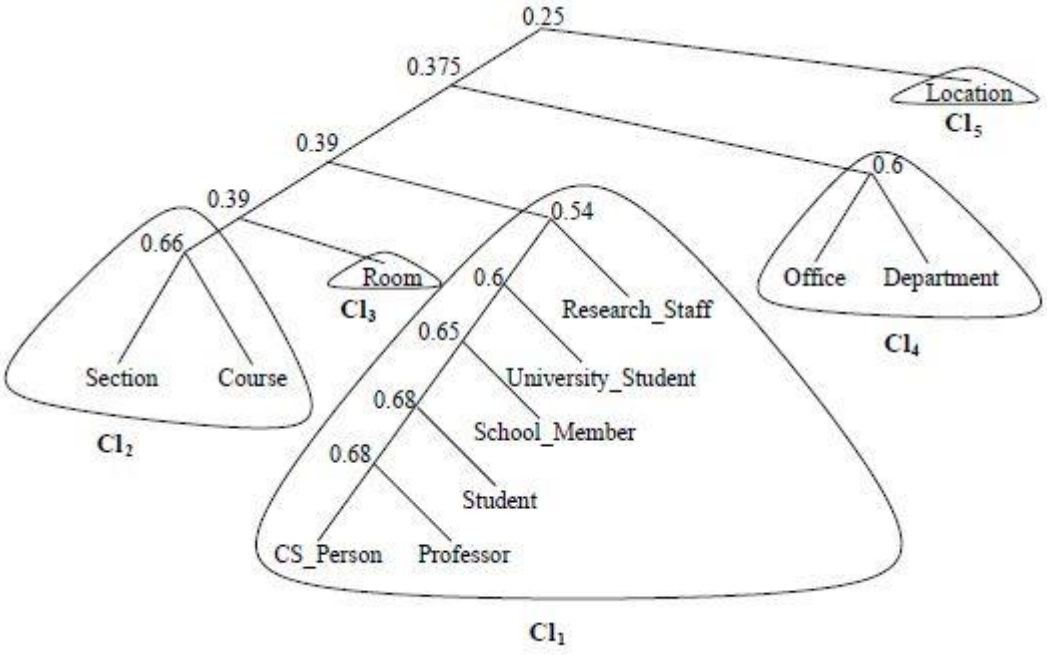

Figure: Affinity tree of S_1, S_2 and S_3

6. Generation of the mediator Global Schema

The mediator view of data stored in the local sources is the process which leads from the cluster generation phase t the definition of mediator global schema. We create a single class, i.e. *global_class*$_i$ (*gc*$_i$) for each cluster, that represents the unified view of all the classes of cluster.

The generation of the gc_i is interactive with the designer. Let consider Cl_i be a cluster in affinity tree. The Global Schema Builder component of MOMIS associates with the gc_i a set of global attributes, corresponding to the union of the attributes of the classes belonging to Cl_i, where the attributes with the family are unified into a global attributes in gc_i.

The generation of the *global_class*$_i$ is in two phases.

- for attributes that have a SYN relationship, only one term is selected as the name for the corresponding global attribute in gc_i
- for attributes that have a BT/NT relationship, a name which is a broader term for all of them is selected and assigned to the corresponding global attribute in gc_i.

For Example, Unification Process for the Cl_1 of Affinity tree is the following set of global attributes:[1]

Cl_1 = (name, rank_title, dept_code, year, takes, relation, email, student_code, tax_ free, section_code, faculty)

Cl_2 = (section_name, section_code, length, room_code, title, rank, description, fund, sector, employee_nr)

The information has to be provided for completing the global class definition. [1]

- The global_class$_i$ name;
- Mappings between the global attributes of global_class$_i$ and corresponding attributes of the classes of Cl_i
    1.) *and* composition : a Broader Term is composed by the union of the Narrowers
    2.) *or* composition : the Broader Term corresponds to the Narrower Terms one at a time;
- default values;
- new attributes to the cluster

Let's consider the example of University_Person. How the mapping rules are created and how to map the attribute n the corresponding attributes of the associated cluster and on possible default/null values defined for it on the cluster. In MOMIS, a mapping table is maintained for each defined global class storing the information on its mapping rules.

In cluster Cl_1 the global attribute "name" of class University.Researh_Staff is mapping with the first_name and last_name as a relationship of BT. Another global attribute "rank" is associated with "rank" for the instances of University, Research_staff and University.School_Member

The mapping rules for global class University_Person are defined below (global schema specification in ODL_{I3}):[1]

```
interface University_Person
(extent Research_Staffers, School_Members, CS_Person
        Professors, Students, University_Students
Key name
{       attribute string name
        mapping_rule (University.Research_Staff.first_name and
                        University.Research_Staff.last_name)
                        (University.School_Member.first_name and
                        University.School_Member.last_name),
                        Computer_Science.CS_Person.name
                        Computer_Science.Student.name
                        Computer_Science.Proofessor.name
                        Tax_Position.University_Student.name;

        attribute string rank
         mapping_rule (University.Research_Staff = 'Professor',
                         University.Schoool_Member = 'Student'
        ……}
```

7. Semantic optimization of global Queries

In above Example there is and global Schema of University_Person which is f Cluster-1's global Schema which consists Research_Staff, University_Student, Schoool_menber, Student, Professor, CS_Person classes

Consider the Source S_2

Default values of attributes for different classes and rules made by ODL_rule

R1: for all X in Student then X.faculty – "Computer Science"
R2: for all X in School_Member then X, rank = 'Student'
R3: for all X in Research_Staff then X.rank = 'professor'

Q -1: Retrieve the names of the student belonging to the "DIW" faculty
We need to use two sub queries that are following:

- select first_name, last_name from School_Member where faculty = "DIW"
- select name from University_Student where faculty = "DIW"

According to the mapping rule we can use the global schema of University_Person because that cluster contains School_Member and University_Student.

After using mapping rules the query look like

select name from University_Person where rank = ‘student’ and faculty = ‘DIW’

So, we can use global schema for query optimization. In above query we used sub query concept so multiple times where clause can be calculated, it takes time and execution of query become slower. Using global schema, the tool uses only one condition so that it may become faster.

8. CRITIQUE

There is still room for improvement of the approach in the direction of reducing the effort of the integration designer. Actually, a manual analysis activity is required to the integration designer, to supply the terminological relationships existing between the different sources not identified by the tool.

Query processing and Optimization process can still be enhanced and corresponding Query Manager Functionalities can be developed further.

If we want a single data, then we need to run the entire process every time.

More data combination will kill the system or lower the efficiency of the system.

9. CONCLUSION

In this paper, we have presented an intelligent approach to schema integration for heterogeneous information sources. It is a semantic approach based on a Description Logics component (ODB-Tools engine) and on an affinity based clustering component (ARTEMIS tool) together with a minimal ODLI3 interface module. In this way, generation of the global schema for the mediator is a semi-automated process. ODLI3 language is available and basic functionalities related to Thesaurus construction and clustering have been implemented.

10. REFERENCE

Most of graph and data is from those paper and reference, we only using those paper as reference to support the idea and using example to introduce our topic.

[1] S. Bergamaschi, S. Castano and M. Vincini. MOMIS : An Intelligent System for the Integration of Semistructured and Structured Data

[2] S. Bergamaschi, S. Castano, S. Montanari ,S. De Capitani di Vimercati, and M. Vincini. A Semantic Approach to Information Integration: the MOMIS project

[3] D.Bneventano, S. Bergamaschi, S. Castano and M. Vincini, R.Guidetti, G.Maalvezzi and M.Melchiori ,Information Integration: the MOMIS Project Demostration

[4] Object Management Group. Object management group. http://www.omg.org/.

[5] S. Bergamaschi, S. Castano and M. Vincini : Semantic Integration of Semistructured and Structured Data Sources