Hybrid Approach for Health Information Integration

Shokooh Kermanshahani
Assistant Professor
Computer Engineering Department
Imam-Khomeini International University
Qazvin, Iran, 3414896818
Email: kermanshahani@eng.ikiu.ac.ir

Hamid Reza Hamidi
Assistant Professor
Computer Engineering Department
Imam-Khomeini International University
Qazvin, Iran, 3414896818
Email: hamidreza.hamidi@eng.ikiu.ac.ir

Abstract— **Having a consistent and unified view of heterogeneous distributed medical information sources is an inevitable need of health informatics. Integrating medical information of patients or about a disease, a treatment or side effects of a drug, etc, is very useful to help medical education, to achieve medical research goals and to provide the computer-based decision support systems. Data integration is one of the main responses of the computer community to satisfy these new needs and several technologies have been developed for this purpose in the research domain as well as in practice that generally propose a fully materialized or fully virtual method. We present, IXIA, a semi-materialized data integration framework, which represents a new aspect of a hybrid method. Our framework is developed based on an object-oriented indexing model and a flexible data refreshing method.**

Keywords-Data integration; Views; Mediators; DataWarehouse; Health Information.

I. INTRODUCTION

One of the most important challenges in health informatics is to have a consistent integrated view of information about each person, either patient or expert. During their life, people refer to different medical centers in different times. By the way, they collect several distributed but interrelated data. Extract useful information from these data is an essential need of a physician while consulting a patient. With increasing growth in the use of computers, today most medical centers including hospitals, clinics, analyzing laboratories, etc, have their own autonomous sources such as documents, data sources and database systems with a large spectrum of heterogeneity from hardware to conceptual and logical data schemas. Health informatics need to integrate these sources to create a unified view and handle unified data [27].

In addition integrating medical information itself is an essential step towards providing the level of personalization required in the next generation of healthcare provision and in order to provide the computer-based decision support systems for medical uses [27].

Finding an efficient method to extract useful information is an essential challenge of health informatics, in research as well as in practice. Among different domains related to this concept, Data Mining for instance, this paper tries to shed more light on information integration.

Information integration is the problem of combining and querying data from several autonomous and heterogeneous sources in a homogeneous fashion. Achieving this goal is also important economically because in the absence of shared resources, new applications would have to be developed for all new needs. Additionally in many cases such as the Web, developing a new application that contains all the necessary tasks and information seems unattainable or very complicated.

There are several approaches in the various research projects which try to develop a data integration system. See for example Tsimmis[2], IM[3], picsel[4], Momis[5,6], Ibis[7], AmosII[8], OBSERVER[9], INFORMIX[12], Automed[10,11].

Broadly speaking, these approaches are classified into materialized and virtual approaches [16]. In the first approach, a materialized integrated view is created and maintained, and a physical repository is created. In the virtual approach, only the sources contain the real data and a global schema provides a coherent integrated virtual view of the underlying data sources [17, 18, 19].

Data warehouses, which are the main (semantic) materialized data integration approach, provide users with a powerful query language that can be used to combine, analyze and otherwise manipulate their data. Data transformation and the translation process are made in the data warehouse [13]. Query processing in a data warehouse is fast and efficient but a solution based on warehousing alone is cost effective for live data integration because of the loading process and its maintenance and implementation costs.

In the mediation (virtual) approach, the real data stays in the local sources and one or several virtual schemas are defined which are the interface between users and the integration system [14]. The users pose their queries through this virtual schema.

The materialized approach may be appropriate if the modifications in the local sources are not frequent and a very fast query response time is needed, whereas the virtual approach is better if the local sources are frequently updated. "The virtual and materialized approaches represent two ends of a vast spectrum of possibilities" [15].

In this paper we present a method to health information integration based on IXIA [28, 29], a semi-materialized framework for a hybrid data integration system, which can offer a trade-off between a fully materialized and a fully virtual data integration approach, according to the needs of the integration application.

Two important aspects on which we are going to discuss are the efficiency of query answering in comparison with a mediation approach and the flexibility in data refreshing in comparison with a materialized data integration system.

A. Query Processing

Depending on the method of schema transformation (Local As View [3], Global As View [2] and Both As View [10]), query decomposition or query reformulation is made in a virtual approach. Query processing is done at the source level. Then a mediator composes the partial responses with respect to data reconciliation (or consistency) and sends a unified response to the user. These processes are time-consuming. There are a wide variety of mediation methods, using different data models, schema translation methods, query evaluation, data translation, etc. In most mediation approaches, the procedures involved in the query evaluation process are executed at the time of the query. In some of them (see Tsimmis [2] as an example), there is a kind of pre-computing using predefined queries. However, in such approaches one cannot request novel queries. In semi-materialized data integration approaches, certain procedures associated with query evaluation are executed in an offline manner, thus reducing the query response time. We call such procedures, pre-procedures. In other words, in our approach the problem of analysis and processing a query is boiled down to the problem of object indexing and data refreshing for this indexation.

B. Data Refreshing

Contrary with fully materialized data integration, in semi-materialized methods all the attributes do not migrate to a repository. We save only the Oid-primary key correspondence and some of the attributes which are named classifying attributes. Data refreshing is made separately for each local source and it can also be made with different frequencies. In many cases the refreshing delay after information modification can be shortened so that it can be considered as an on-line access. The advantages of this approach are discussed in section V.

The semi-materialized framework is developed based on the OSIRIS platform, which is an object-based database and knowledge base system where views play a primitive role. In section II we briefly present the related efforts. We present our semi-materialized integration approach in section III and we discuss its advantages in section IV. Some conclusions and perspectives are presented in section V.

II. Related Work

The hybrid approach to data integration was introduced with the Squirrel project [15]. Its focus was on the relational or object-oriented data sources. It develops a general and flexible mediator approach which allows a relation in the integrated view to be fully materialized, fully virtual or partially materialized (i.e., some of the attributes are materialized and others are virtual). It makes an incremental update for (partially) materialized relations in the integrated view.

Another effort for a hybrid approach develops a framework for warehousing the web contents [20]. Its objective is the integration of Data Warehouse (DW) data with the required web-based information. It selects a set of nonvolatile or frequently queried web data to materialize in the DW and some other queries can be answered using web data that is not materialized, or using a combination of them. An ontology engine makes the mapping rules between web data and attributes in Data Integration Systems (DIS).

A hybrid approach for ontology integration has also discussed in [21]. The main difference between this approach and [20] is that in this approach the data sources can be ontologies, web data, or any form of structural or semi-structural data sources.

Contrary to these approaches, in our approach all of the attributes and relations remain in the local sources and are retrieved at query time. The materialized level of this approach lies in the indexation information. We materialize the link between object identifiers (Oid) in the integrated view (or: in the global schema) and the data identifiers (primary keys in relational sources, Oids in an object-oriented source) in the local sources. Some of the attributes which are effective in indexation refreshing are also materialized.

The two main differences between our semi-materialized framework and the approaches described above are query optimization for all of the queries and the flexibility in data refreshing at the indexing level that can be made with different frequencies for different data sources.

III. OSIRIS

As a hybrid approach, the main objective of IXIA is to make a trade-off between query response time and data freshness in a data integration application including medical information integration systems [28, 29]. IXIA, however proposes a new hybrid methodology in comparison with the existing hybrid data integration approaches such as [15, 20, 21]. Since the data model of IXIA is based on OSIRIS [22, 23], in this section we first review the OSIRIS data model and query evaluation and then we present IXIA architecture.

OSIRIS is a view-based database and knowledge base model where views are similar to concepts defined by logical properties, similar Description Logic approaches presented in TIMC-IMAG laboratory by Ana and Michel Simonet [22].

A. P-types and Views

The main concept of the OSIRIS model is the P-type concept, which supports the specification of viewpoints on a domain [23]. For example, DOCTOR and PATIENT can be viewpoints of the P-type PERSON in the hospital domain, To specify a P-type one first gives its minimal (root) view then its other views by simple or multiple specialization. When specializing a view new attributes and assertions (logical constraints) may be added. The minimal view is the root of the hierarchy of views of a P-type. Thus, in OSIRIS a P-type is defined from its views, which are object-preserving [24]. Such a top-down approach is contrary to that of relational systems where views are defined as restrictions of a set of existing relations, and may themselves be used as relations in order to define other views.

The type of a P-type is derived from the views declarations (including the minimal view). The type PERSON contains all the attributes and methods which appear in its views. The domain of an attribute in the type PERSON is the union of its domains in the views where it is declared.

To express that a person may be seen as a patient, a doctor, a staff member, one will create the views PERSON, PATIENT, STAFF, … as subtypes of the P-type PERSON (see Figure 1). The set of interest of the minimal view PERSON is identical to that of the P-type PERSON. The domain of another view is a subset of the domain of the view it specializes, or of the intersection of the domains of the views it specializes in case of multiple specialization.

In OSIRIS, the P-type is given the name of its minimal view. All the objects of a P-type are models of its minimal view. Access to an object under a viewpoint provides access to the attributes of the viewpoint. Thus accessing an object from the minimal view only provides the attributes of the minimal view while accessing it in the viewpoint of the P-type gives access to the whole set of attributes of the type..

An object belongs to one P-type, PERSON for example, if and only if it satisfies the requirements of its minimal view. This means that its assertions are valid. An object can belong to only one P-type, which means that P-types are disjoint concepts if considered in a DL perspective.

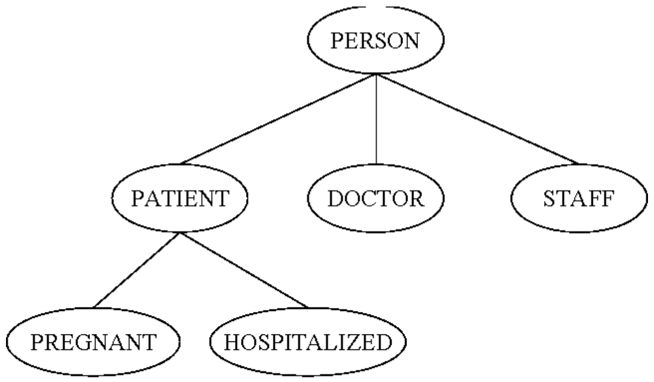

Figure 1. Graph of P-type PERSON.

We present the main features of the P-type description language through a very simple example. As figure 1, the modeled universe is that of persons and services of a healthcare center. Persons may be PATIENT, DOCTOR, HOSPITALIZED, etc, or some of them simultaneously. A given person is a model of the minimal view and may belong to none, any or several other views.

```
view PERSON      -- Minimal view of P-type PERSON
 attr     Code: int
          Name : P_NAME;  -- P_NAME is declared elsewhere
          Sex : CHAR;
          BirthDay : DATE;
          Radiology : STRING;
          followedBy: setof DOCTOR; -- view of PERSON
 key      Code      -- External key ; not mandatory
 methods
          -- other functions specification
 assertions
 -- Domain Constraints
          Sex in { "f", "m" };
          0 ≤ Age ≤120;
          Radiology in  { "allowed", "not allowed", "to verify"};
end; -- Note that the minimal view automatically contains a private attribute  Oid : tOid.
```

```
view      PATIENT : PERSON-- PATIENT specializes PERSON
 attr     ConsultationDate : DATE ;
          ConsultationReason: STRING;
          Pregnant in  { "yes", "no" };
 -- Inter-Attribute Dependencies
          Pregnant = "yes" ➔ radiology= "not allowed";
end;
```

```
view      HOSPITALIZED : PATIENT
 attr     Service : SERVICE;-- a view of another P-TYPE
          ArrivalDate: Date;
end ;
```

```
view DOCTOR : PERSON
 attr     Specialty : setof STRING in {"cardiologist", "nephrologist", "diabetologist", rheumatologist"};
          YearsOfExperience: INT;
end ;
```

B. *Classification Space*

Most innovative features of the system come from the use of a classification space, which is distinct from the original set of users' views.

The classification space is a partitioning of the object space into equivalence classes named Eq-classes, according to the relation "have the same truth values according to the (entire set of) Domain Predicates of the type". As a consequence all objects of a given Eq-class are models of the same assertions (Domain Constraints and Inter-Attribute Dependencies) [23].

In a P-type T, one considers for each attribute Ai the set PT(Ai) of predicates over Ai which appear in the assertions (Domain Constraints and Inter-Attribute Dependencies) of the views of T. Elementary predicates in these constraints are of the form Ai ϵ Dik where Dik is a subset of the domain of definition Δi of Ai. A predicate Ai ϵ Dik defines a partitioning of Δi into two (disjoint) subdomains: Dik and Δi - Dik. The product of all the partitions defined by the predicates of PT(Ai) constitutes a partition of Δi whose blocks dij, called Stable Sub Domains (SSDs), have the following property: stability of an attribute. When the value of an attribute Ai of an object varies within the same stable subdomain dij, Ai continues to satisfy the same set of predicates of PT(Ai) [25].

Considering the above definition of the P-type PERSON, and considering only the predicates on the attributes Age, Pregnant and Sex, we obtain the following partitioning of the attributes:

```
SSDs of Age:  d11 = [0, 18[,
    d12 = [18, 27], d13 = ]27, 65[,
    d14 = [65,140]
SSDs of Sex:   d21 = {"f"}, d22 = {"m"}
SSDs of Pregnant:  d31 = {“yes”},
    d32 = {"no”}
```

This partition can be extended to the space of objects (which is restricted here to the three dimensions considered) and constitutes the classification space of the P-type.

Each element of the classification space is called an Eq-class. It is represented by a tuple with *n* elements, where *n* is the number of classifying attributes of the P-type.

Let SDS_{Attr1}, SDS_{Attr2}, …, SDS_{AttrN} be the set of the stable subdomains of the attributes ATTR1, ATTR2, …, ATTRn of the P-type T respectively.

$$\text{ClassificationSpaceT} = \{ \langle d_{1i}, d_{2j}, \ldots, d_{nk} \rangle \mid d_{1i} \in SSD_{Attr1}, d_{2j} \in SSD_{Attr2}, \ldots d_{nk} \in SSD_{AttrN} \}$$

Partitioning the object space into Eq-classes is central to the implementation of the OSIRIS system. Although the actual partition is not represented in its totality (its size is exponential to the number of classifying attributes) it underlies most runtime processes such as object classification, view classification (subsumption), integrity checking and object indexing.

C. *Indexing Structure Descriptor*

An indexing structure called ISD (Indexing Structure Descriptor) is defined for each P-type [23]. Its main components are:

- A vector of SSDs representing one or more Eq-classes indexing a set of objects. Two values have been added to represent the unknown and undefined (null) states of an attribute.
- A vector of views that provides the status (Valid, Invalid or Potential) of each view of the P-type for the set of objects indexed by this ISD.
- A reference to the actual set of objects of the ISD
- The total number of objects indexed by the ISD

An object, even if only partially known, belongs to one and only one ISD, as an ISD denotes all its possible Eq-classes. The sets of Eq-classes corresponding to different ISDs may not be disjoint in the case of incompletely known objects. Only actual ISDs, i.e., ISDs containing actual objects, are represented. When an unknown attribute becomes known the corresponding object changes its ISD (a new ISD is created if necessary). When an attribute changes its value it remains within the same ISD iff none of its attributes has changed its SSD.

D. *Query evaluation in OSIRIS*

Queries are evaluated in three steps:

1. Determination of the ISDs corresponding to the query when rewritten in terms of the SSDs of its attributes.
2. Determination of the ISDs indexing objects that are valid for the query.
3. Projection of the resulting objects onto the attributes of interest of the query.

IV. IXIA GLOBAL ARCHITECTURE

Like a mediator approach, IXIA has a mediator-wrapper architecture, although with some materialization. IXIA has been developed based on OSIRIS knowledge based system in order to take advantage of its object indexation system. IXIA materializes the indexation structure. A direct advantage of this materialization is query optimization for the integration system.

Figure 2 shows a global presentation of the IXIA architecture. Only the relevant modules of OSIRIS are shown here. We briefly describe this architecture in the following two subsections:

A. *Indexation and Indexation maintenance*

After defining the integrated schema (an OSIRIS schema), the classification server makes a first object indexation for all the sources objects which correspond to the global schema and then sends the indexation data to be saved in the OSIRIS indexation module.

The indexation data are then incrementally updated by the classification server. The "Modification Detector" modules detect if there is some updating in the sources which results in updating the indexation data from the last indexation maintenance. The modification detector of each source functions independently and can be executed with different frequencies.

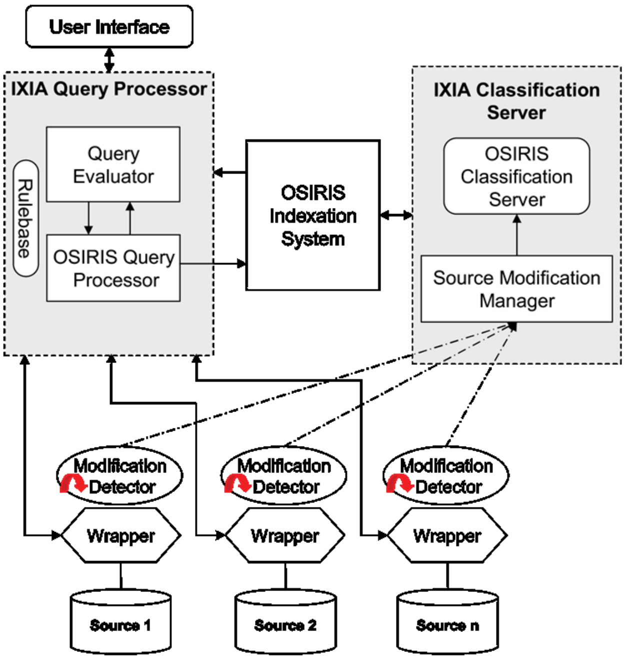

Figure 2: IXIA architecture.

Updating information obtained from the modification detectors is sent to the "Source Modification Manager" module of the IXIA classification server. This module adds the source information and prepares the "indexation repairing message" for the "OSIRIS Classification Server", which does the indexation maintenance just as in a single OSIRIS database.

We note that mapping between the object indexation and data in local sources are made in the wrappers. We save the (Oid, Primary-key) correspondence between the OSIRIS objects of the global schema and the data in sources. Wrappers also do the mapping between the local sources' schemas and the OSIRIS Global Schema.

B. *Indexation and Indexation maintenance*

Like OSIRIS, query processing in IXIA is done using the indexation information and provides a query optimization.

As we explained in section 1, depending on the method of schema transformation (LAV [3], GAV [2] or BAV [10]), a query decomposition / reformulation is made in a virtual approach. These processes are time-consuming. In most mediation approaches, the procedures involved in the query evaluation process are executed at query time.

In IXIA, some procedures associated with query evaluation are executed in an off-line manner, thus reducing the query response time. We call these procedures pre-procedures, and they consist of the indexation process and maintenance. In other words, in IXIA the problem of analyzing and processing a query is transformed into the problem of object indexing, refreshing this indexation, and searching the response objects' attributes in the sources. The query decomposition and the generation of the execution plan are done by the "Query Evaluator" module of the IXIA query processor. The partial queries are sent to the OSIRIS Query Processor to find the satisfied objects using the object indexation system. Re-composition of partial responses into a final response is also done by IXIA Query Evaluator.

Because of the hierarchical structure of views, the search space is reduced at mediator level which implies a query processing optimization in this approach.

C. *Example of modification process*

We consider a simple relational local source as follows:

S1:

Patient (code, Pkey, name, sex, birthday, address, comment);

Hospitalized (code, entry date, coming out date, comment);

Pregnant (code, prevision delivery date, sonography result);

Doctor (code, Dkey, sex, name, specialty, sector);

Staff (code, Skey, name, sex, birth date, sector, address);

Test (Tcode, description, comment);

Considering the global schema of integration view of figure 1, the modification detector procedures are as follows:

```
If      new entry   in  Patient, Doctor, Staff
{       Insert an Oid;
        Retrieve all corresponding attributes
        // (using wrapper mappings);
        Send attributes to classification server;
        // (classification server will classify this object)
}
For (Oid = 1 to n (the number of objects for s1))
{       Select classifying attributes;
        For (j = 1 to m (number of the classifying attributes))
        {
            Compare new value with last value
            // (last value in Data Store);
            If the last value and new value are  in different SDS
            {
                    ImFlage( Indexation Modification Flag) = 1;
                    Classifying Att[Oid, j] = change;
            } // if
         } // for
         If  ImFlag == 1
         {
            Send Oid to classification server;
            Send all of Classifying attributes where
                    Att[Oid, j] == change to classification server;
         }
}
```

V. Conclusion

In this paper, we have presented a hybrid approach to data integration. Like a mediation approach, it has a mediator-wrapper architecture but some information is materialized at the mediator level. Contrary to the other hybrid methods [15, 20, 21], we do not choose some relations or some queries to be materialized and let the others virtual. Instead, we use a materialized object indexation at the mediator level which reduces the response time for all the queries of the Data Integration System. The materialized indexing information is refreshed using a Modification Detector over each source. Data refreshing at the indexation level can be made with different frequencies for different sources. This approach makes possible the addition of a new source independently from the other local sources.

Providing an ontology level of information, based on the hierarchical structure of views, IXIA can be used in health information integration systems where a conceptual presentation of integrated schema is crucially needed.

Acknowledgment

The authors wish to thank A. Simonet and M. Simonet from TIMC-IMAG laboratory in France and Imam-Khomeini International University. This work was sup-ported in part by a grant from Qazvin OMC project.

References

[1] Alon Halevy, Anand Rajaraman, Joann Ordille, Data integration: the teenage years, VLDB '06: Proceedings of the 32nd international conference on Very large data bases}, pp. 9-16, 2006, Seoul, Korea.

[2] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom. "The TSIMMIS Project: Integration of Heterogeneous Information Sources". In Proceedings of IPSJ Conference, pp. 7-18, Tokyo, Japan, October 1994.

[3] Alon Y. Levy, Anand Rajaraman, Joann J. Ordille, Querying Heterogeneous Information Sources Using Source Descriptions, Proceedings of the Twenty-second International Conference on Very Large Databases, pp. 251-262, 1996, Bombay, India

[4] Francois Goasdoue Veronique Lattes, Marie-Christine Rousset, The Use of {CARIN} Language and Algorithms for Information Integration: The {PICSEL} System, International Journal of Cooperative Information Systems, V9, N4, pp. 383-401

[5] D. Beneventano, S. Bergamaschi, S. Castano, A. Corni, R. Guidetti, G. Malvezzi, M. Melchiori, M. Vincini:, Information Integration: the MOMIS Project Demonstration, ACM SIGMOD Digital Symposium Collection 2000.

[6] D. Beneventano, S. Bergamaschi, The MOMIS Methodology for Integrating Heterogeneous Data Sources, IFIP World Computer Congress. Toulouse France, 22-27 August 2004.

[7] Andrea Calì, Diego Calvanese, De Giacomo, Giuseppe, Maurizio Lenzerini, Paolo Naggar, Fabio VernacotolaIBIS, Semantic Data Integration at Work, Proc. of the 15th Int. Conf. on Advanced Information Systems Engineering (CAiSE 2003), pp. 79-94, Lecture Notes in Computer Science.

[8] T. RISCH and V. JOSIFOVSKI and T. KATCHAOUNOV, Functional data integration in a distributed mediator system, In Functional Approach to Computing with Data, Springer, 2003

[9] Eduardo Mena, Vipul Kashyap, Amit P. Sheth, Arantza Illarramendi, OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies, Conference on Cooperative Information Systems, pp. 14-25, 1996

[10] E. Jasper, N. Tong, P. McBrien, A. Poulovassilis, View generation and optimisation in the AutoMed data integration framework, 2003.

[11] Michael Boyd, Sasivimol Kittivoravitkul, Charalambos Lazanitis, Peter McBrien, Nikos Rizopoulos: AutoMed: A BAV Data Integration System for Heterogeneous Data Sources. CAiSE 2004: 82-97

[12] Nicola Leone, Gianluigi Greco, Giovambattista Ianni, Vincenzino Lio, Giorgio Terracina, Thomas Eiter, Wolfgang Faber, Michael Fink, Georg Gottlob, Riccardo Rosati, Domenico Lembo, Maurizio Lenzerini, Marco Ruzzi, Edyta Kalka, Bartosz Nowicki, Witold Staniszkis, The INFOMIX system for advanced integration of incomplete and inconsistent data, SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data, 2005, 1-59593-060-4, pp. 915-917, Baltimore, Maryland.

[13] Laura M. Haas, Eileen Tien Lin, Mary Tork Roth: Data integration through database federation. IBM Systems Journal 41(4): 578-596 (2002)

[14] Gio Wiederhold, Mediators in the Architecture of Future Information Systems. IEEE Computer 25(3): 38-49 (1992)

[15] Richard Hull and Gang Zhou, A framework for supporting data integration using the materialized and virtual approaches, pp. 481-492, 1996.

[16] Hull, R. "Managing Semantic Heterogeneity in Databases: A Theoretical Perspective". PODS'97, May 12-14, Tuscon Arizona

[17] W.H.Inman, "Building the Data Warehouse", Joha Wiley and Sons, 3rd edition, 2002

[18] A. HALEVY, "Data Integration: The Teenage Years", VLDB 2006.

[19] Shokoh Kermanshahani, Houda Ahmad , Ana Simonet, Michel Simonet, A view based architecture for a multi-level semantic mediator, IKE 2008, LasVegas USA.

[20] Yan Zhu, A Framework for Warehousing the Web Contents, ICSC'99, LNCS 1749, 83-92, 1999.

[21] A. Alasoud, V. Haarslev, N. Shiri, A Hybrid Approach for Ontology Integration, VLDB 2005, Trndheim.

[22] M. Roger, A. Simonet, M. Simonet, "Bringing Together Description Logics and Databases in an Object-Oriented Model", DEXA 2002, Database and Expert System Applications, Toulouse, Sept. 2002

[23] A. Simonet, M. Simonet, "Classement d'instance et Evaluation des Requêtes en OSIRIS", in BDA'96 : Bases de Données Avancées, Cassis, France, Août 1996, pp 273-288.

[24] M. H. Scholl, C. Laasch, M. Tresch, Updatable Views in Object-Oriented Databases, Proc. 2nd DOOD conf., pp 187-198, Dec. 1991

[25] D.Stanat, D. McAllister, Discrete Mathematics in Computer Science, Prentice Hall, 1977.

[26] AHMAD, H.,KERMANSHAHANI, S., Simonet, A., simonet, M.: A View-Based Approach to the Integration of Structured and Semi-Structured Data, IEEE International Baltic Conference on Databases and Information Systems – Communication of Baltic DBIS 2006.

[27] Andrew Branson, Tamas Hauer, Richard McClatchey, Dmitri Rogulin, Jetendr Shamdasani: A Data Model for Integrating Heterogeneous Medical Data in the Health-e-Child Project. CoRR abs/0812.2874 (2008)

[28] Shokooh Kermanshahani, Hamid Reza Hamidi: A Data Integration Approach Based on Indexation. International Journal of Computer Science and Network Security, 16 VOL.17 No.7, July 2017.

[29] Shokoh Kermanshahani, Semi-Materialized framework: A Hybrid Approach to Data Integration. CSTST '2008, Proceeding of the 5th international conference on Soft computing, pages 600-606, Cergy-Pontoise Francd. ACM, New York, NY, USA.