SPEKTA
(Jurnal Pengabdian Kepada Masyarakat: Teknologi dan Aplikasi)
ISSN 2723-8016 (online) | 2723-8008 (print)
Vol 4, No. 1, pp. 97-106

Flexible Data Refreshing Architecture for Health Information System Integration

Shokooh Kermanshahani *, Hamid Reza Hamidi
Computer Engineering Department, Imam-Khomeini International University, Qazvin, Iran.
*Corresponding Author: kermanshahani@eng.ikiu.ac.ir

ARTICLE INFO

Article history
Received : March 15, 2023
Revised : April 14, 2023
Accepted : April 17, 2023

Keywords
DataBase Integration;
Schema and Subschema;
Data Warehouse and repository;
System Integration and Implementation.

ABSTRACT

Background: 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.
Contribution: This article proposes a flexible incremental update method for the materialized part of the integration system. It permits us to manage the integration system according to the characteristics of the data sources which can change.
Method: This paper presents a hybrid data integration approach in which the materialized part of the system in mediator is the object indexation structure based on an instance classification of the sources objects which correspond to the global schema. The object identifier of each object in the indexation structure is materialized together with the attributes which are needed for the incremental updating of this indexation (classifying attributes).
Results: The main idea of this paper is to develop a hybrid data integration framework, which represents a new aspect of a hybrid method focusing on flexible data refreshing.
Conclusion: This hybrid approach implements a vertical hybrid approach. It means that at the mediator level, some data of each object are materialized and others are virtual.

INTRODUCTION

In a data integration system, data from different sources are integrated into a global integrated schema which satisfies the needs of users and which is managed by a management system. All heterogeneities are hidden from the user, who queries the global schema as a single database schema [1]. One of the most important challenges for integrating different autonomous

data sources is the heterogeneity which can appear at different levels. The hardware on which two information sources are developed, the network protocols, the software, the data and the query languages may be different. However, the essential and more complicated aspect of heterogeneity is semantic heterogeneity. Semantic heterogeneity characterizes the differences in signification, interpretation or utilization of the same data [2].

1. Health Information Integration

   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 inter- related 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.

   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 [3]. Finding an efficient method to extract useful information is an essential challenge of health informatics, in research as well as in practice.

2. Schema Mapping

   To query the data of several local sources via a global schema, this global schema has to relate to the schemas of the local sources. Different methods have been developed to map several local schemas to a global schema: Global-As-View (GAV), Local-As-View (LAV) and Both-As-View are the most important ones. In a GAV mapping approach, the global schema is defined as a view of the underlying source schemas. In a LAV mapping, the schema of each local source is defined as a view of the global schema, and in a BAV mapping, these two methods are combined [4], [5], [6]. The BAV method can obtain a local schema from the relations of the global schema and vice-versa.

3. Fully Materialized Data Integration

   The global schema of a data integration system can be fully materialized. It means that a new repository is developed by a data management system and a copy of all the data which correspond to the global schema is saved in this repository. Data from local sources are Extracted, Transformed and Loaded (ETL process) to this repository [7], [8], [9], [10]. Query evaluation in such an approach is similar to that of a single database and could have access to a powerful query language and to query optimization. However, in such an approach, the data of local sources cannot be directly accessed and the data repository has to be periodically updated. Therefore, there is always a delay to access the last updated data. In addition, building a new repository and ETL processes are expensive.

4. Fully Virtual Data Integration

   The global schema of a data integration system may be virtual [11], [12], [13]. In this case, all data remains in the local sources and a middleware containing the global schema is developed. This middleware, generally called a mediator, decomposes or reformulates a user query over a set of local sources. It then recomposes the partial responses into a single response for the user. Data and query languages of local sources may be different from those of the middleware

10.12198/spekta.v4i1.7845

Wrappers which contain schema mappings, translate data and queries between the sources and the mediator [14]. With this approach, users have online access to the data of the local sources, but query processing is complicated and time consuming.

5. Hybrid Data Integration

   A third possible solution for data integration is to develop a partially materialized integrated schema. A data integration approach which uses such a global schema is called a Hybrid Approach [15]. Fully materialized and fully virtual data integration approaches obey to different priorities. In a fully materialized approach, the main priority is the query response time, and in fully virtual data integration, data freshness is more important. However, in many data integration scenarios different priorities may be associated with different data, and a tradeoff between query response time and data freshness may be preferred to satisfying only one of these two issues. A flexible approach which permits some data to be materialized and other data to be virtual can satisfy both of these goals. In the existing hybrid approaches the global view is partitioned into materialized and virtual parts. Some objects or relations are chosen to be materialized and others reside in the local sources and will be extracted at query time [2].

6. Efficient Query Processing

   Efficient query processing is one of the most important challenges in data integration. Because of the dynamic nature of the data integration context, new challenges arise to optimize the processing of queries. At the mediator level, the optimizer may not have enough information to decide on a good plan and in addition, at execution time, a source may not respond exactly as it had been considered at optimization time [16].

In a data integration system, two level of query optimization can be considered: global optimization for query plan and local optimization for subqueries that retrieve data from individual data sources [17]. The restriction of the search space for a query, using semantic knowledge at the mediator level, is a solution to provide global optimization for query processing.

The main idea of this paper is to develop a hybrid data integration framework, which represents a new aspect of a hybrid method focusing on flexible data refreshing. Contrary to other projects, 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 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.

METHOD

The existing hybrid approaches provide a rapid access to the materialized data. Other data remain in the local sources and are queried directly from the sources when necessary. As a consequence, only the queries to the materialized part of the system are optimized. A typical example of integration scenarios compatible with such approaches is the integration of geographical data, hotel and tourism information and weather information for a travel agency. In this example, the data of geographical and tourism centers are stable and can be materialized while other information such as weather data change more frequently and are integrated in a virtual manner. Many other data integration scenarios can profit from the tradeoff that a hybrid

approach offers between query response time and data freshness.

RESULTS AND DISCUSSION

The next section reviews the architecture of IXIA an IndeX-based data Integration approach [18]. It provides a query optimization to the integration system. Then we demonstrate the flexibility of data refreshing, according to the needs of the integration applications.

1. The Architecture

   Like a mediator approach, IXIA has a mediator-wrapper architecture, although with some materialization. IXIA has been developed based on the Osiris system in order to take advantage of its object indexation system. Osiris is an object-based database and knowledge base system based on a hierarchy of views where views are similar to concepts defined by logical properties [19], like in a Description Logic approach [20]. Figure 1 shows a presentation of the IXIA architecture.

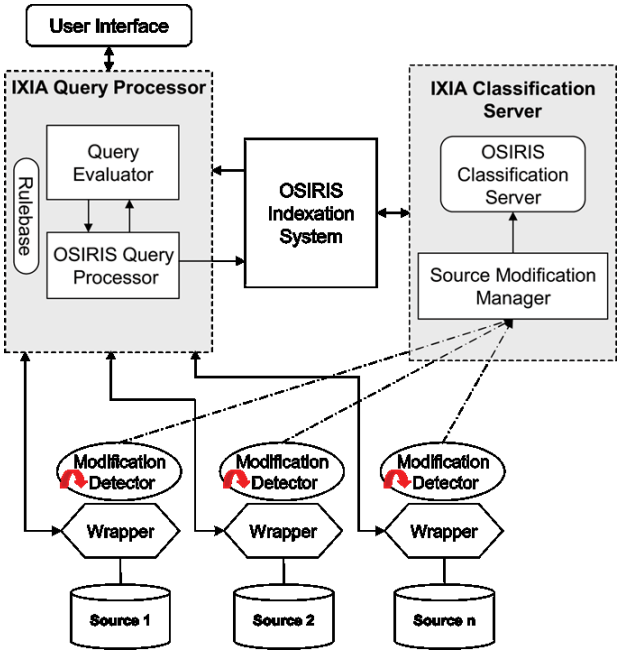

Figure 1. IXIA data integration architecture [18].

The Osiris system implements the P-type data model with two principal objectives: data sharing through the views and automatic verification of integrity constraints. An object is an instance of one and only one P-type, but it can belong to several of its views and change the views it belongs to during its lifetime. Classifying an object into the views of its P-type is a characteristic inherent to this model. This is why the Osiris system, which implements the P-type model, can offer functionalities for decision support, alert management, semantic query optimization, etc. IXIA went deeper into the use of the P-type concept with the purpose of profiting from its object indexation system. As mentioned above, the main materialized part of the IXIA is the indexation structure which is based on the instance classification of Osiris. A direct advantage of this materialization is query optimization for the integration system.

2. Integration Process

   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

10.12198/spekta.v4i1.7845

and 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. 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.

The mappings between the object indexation and data in local sources are made in the wrappers. IXIA 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.

3. The Modules Design

Figure 2 shows a detailed architecture of IXIA. This prototype is an extension of the Osiris database system in order to develop a platform for data integration. Consequently, the architecture of IXIA consists of some modules of Osiris, some extended modules and some new modules. The following subsections describe these modules.

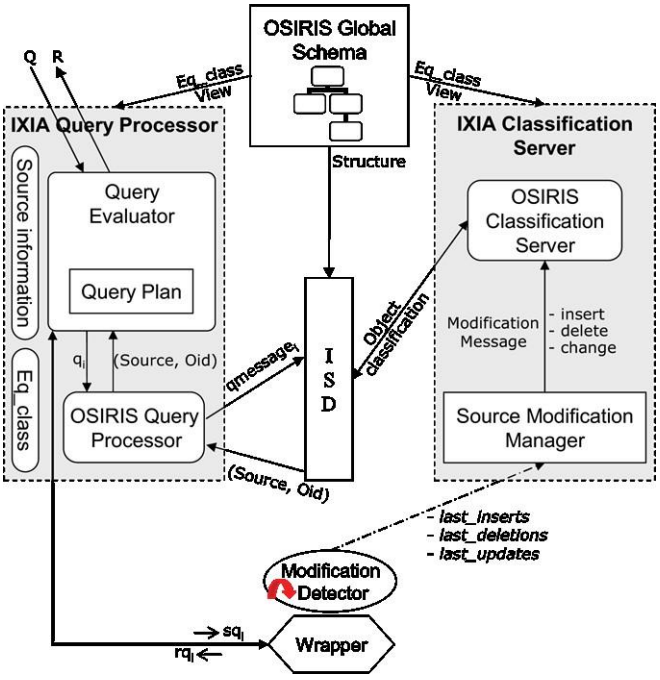

Figure 2. IXIA modules interactions

4. Indexing Structure Descriptor

The Indexation Structure in IXIA is much similar to that of Osiris. The only difference is that IXIA added the sources information to the indexation structure at the mediator level in order to make them unique. Adding sources information to the indexation structure can be done in different ways. In our implementation a source field is added to the object identifier Oid in the indexation module. Object identifiers at the wrappers level are identified through a simple Oid, which is unique in the wrapper of each source. It calls a Local Oid (Loid). The

object identifier in the Indexing Structure Descriptor (ISD) space is:

Goid = Loid + Source-ID (Goid : Global Oid).

5. Wrappers

A wrapper in this architecture, like in all integration architectures, contains the mapping between a source schema and the integrated schema. However, in this approach wrappers contain the assignments between object identifiers Oid in the global schema and the primary-keys in the data sources. In other words, the mapping in this approach, is based on the (Oid, primary-key) correspondence. A study for the mapping of a relational database schema to an Osiris target schema had been done in [21]. In this study, after defining the identifiers assignments, each attribute of the target schema is mapped to one attribute or a function of several attributes of the source schema. In this work, the maintenance of the wrappers when adding, deleting or modifying an object was not considered. The "Modification Detector" of IXIA deals with this task.

Since the mapping between the global schema and source schemas is made by a mapping between the identifiers, adding a new source is independent from the other sources and a novel source can be integrated as the new wrapper be developed. However, some dependencies may exist between data of different local sources, the wrappers develop independently and the dependencies are managed by IXIA Query Evaluator in query processing.

6. Modification Detector

The object indexation information must be refreshed periodically. In order to maintain the indexation data, the modifications of data sources have to be detected. A module called 'Modification Detector" (MD) has been developed for this purpose. A MD module is developed for each source independently from other MDs and the detection process can be executed at different frequencies for different sources. If the classifying attributes of a P-type of the global schema are in two different sources, then to reclassify the objects of this P-type, the corresponding attributes of both sources must be considered. In such case, the MDs remain independent and this question will be dealt with in the "Source Modification Manager". To simplify and according to our motivation application, in the current implementation we consider the situation where all classifying attributes of an object are in a same source. In such case when a real world object is in two different sources, IXIA will create two objects at the mediator level, each for one source, and by joint the source codes to Oids they will be distinguished. Such implementation permits us to verify the consistency of data in different sources.

Studying the data consistency verification can be considered as a future work. In the current IXIA prototype, the MD module has been implemented on top of each wrapper and together they make a single component with two parts. Thus it can be used to add or delete the Oid/Primary- key mapping in the wrapper. The details of the "Modification Detector" and its implementation are explained through the description of the indexation maintenance functionality.

7. Classification Server

The "Classification Server" of IXIA consists of "Osiris Classification Server" (OCS) and "Source Modification Manager" (SMM). It is responsible for object classification in the IXIA mediator. The SMM receives the modification of each source from the MD and prepares a message containing object insertions, object deletions or updated objects to the "Osiris

10.12198/spekta.v4i1.7845

Classification Server". It also adds the source information to the message (it creates a Goid from the Loid received from the MD). The Osiris classification server receives the modification message and makes the classification updates. Figure 3 presents a pseudocode for the Source Modification Manager.

8. Query Processor

Like all query processors in data integration systems, the IXIA query processor receives the user query, performs query decomposition / reformulation and then recomposes partial responses in order to send a response to the user. The IXIA query processor contains two main modules: "Osiris Query Processor" and "Query Evaluator".

Osiris Query Processor module works as in a single Osiris database. For each partial query, received from the query evaluator, it searches in the ISD Space the objects' Oids which satisfy or potentially satisfy that query. However, in an Osiris database, after defining these Oids, the query processor does the verification of the complementary conditions and extracts the requested attributes. In IXIA, since data remain in the sources, this second part is not done by the Osiris Query Processor.

All the tasks which correspond to the data integration query processing are done by the Query Evaluator module. It generates a query plan for the user query. Following this logical query plan, the query evaluator sends partial queries to the Osiris query processor and retrieves the partial response which are the Global Oids (Goids). It is also responsible for preparing the partial queries to the sources and combining the partial responses in order to provide a final response for the user.

```
01. class SourceModificationManager
02. {
03.   OSIRIScServer OSIRIS_cServer      // OSIRIS classification server identifier
04.   SourceModificationManager (OSIRIScServer t){      //constructor
05.       OSIRIS_cServer:=t
06.   }
07.   inserts (ModificationDetector MD, Wrapper wrapper)
08.   { //---- classification of last inserted
09.     Buffer tmp_inserts=new MD.last_inserts      //dump last inserts
10.     MD.I_dumped:=true
11.     while (tmp_inserts has more object) {
12.         extract a local object from tmp_inserts
13.         create a global object based on local object and wrapper
14.         OSIRIS_cServer.insert(global object)
15.     }
16.   }
17.   // ... same implementation for "deletions" and "updates" methods considering:
18.   // ... in line 13. find a global object instead of create a global object
19. } //SourceModificationManager
```

Figure 3. Java-Like pseudocode of the source modification manager.

In this paper comparison of our approach and other existing solutions for data integration, particularly other hybrid approaches, Horizontal Hybrid Approach / Vertical Hybrid Approach. In all the hybrid data integration approaches that were studied in the first section, a horizontal hybrid method is used. This means that in the integrated schema some objects or relations are

implemented virtually and others by a materialized method. All data manipulation in the virtual parts is made in a virtual manner and data of the materialized views are manipulated by materialized paradigms. A user query in such approaches may correspond to materialized or virtual parts or it can be decomposed between virtual and materialized parts. Consequently, the performance of the query processing as well as the data accessing delay are similar to those of fully materialized or fully virtual approaches respectively for materialized and virtual parts of the data integration system.

The hybrid approach that proposed in this paper, implements a vertical hybrid approach. It means that at the mediator level, some data of each object are materialized and others are virtual. The attributes of the objects remain in the local sources and generally data are extracted from the sources at query time. The object identifier of each object in an indexation structure is materialized together with the attributes which are needed for the refreshing of this indexation. This kind of partition of the integration system to the materialized and virtual parts is called a vertical partition, and our approach is a vertical hybrid approach.

In comparison with a fully materialized data integration approach, IXIA does not need to perform a full data migration and the real data remains in the local sources. However the query response time in a fully materialized approach is faster. In our approach the query response time is faster than that of a fully virtual approach; however, in certain situations access to freshly updated data can be delayed. One other advantage of IXIA is that contrary to most data integration approaches, which have a structural global schema, it provides a conceptual schema as the interface to the integrated sources. In other words, the data type used in IXIA can provide an ontology level of information which is inevitable to a health information integration system.

CONCLUSION

This project proposed a data refreshing solution. The modification detector is the core of this solution. It offers data refreshing for different sources, hence for different data can be done in different time periods. This flexibility in IXIA permits us to consider the characteristics of different sources in each data integration application. The arrangement of refreshing for different data sources can be changed by changing the frequencies of different MDs, which is a decision of the system administrator. Adding a novel source to the system is independent from other sources. Because of its special mapping system (object-to-object), when defining a wrapper for a novel source, it can be integrated to the system. The novel source information, however, must be added to the rule base of the query evaluator and the Source Modification Manager.

There are some restrictions and difficulties in the IXIA architecture. The wrappers need two levels of mapping: schema mapping and (Oid, primary-key) mapping. This second kind needs to be updated each time an object is deleted or inserted in a source. When an object is inserted an Oid must be created and assigned to the primary-key. However, this process can also be done by the Modification Detector in order to be optimized.

Finally, it must be considered that like all the approaches which use some materialization in their architecture, we often do not have access to online data and generally there is a slight delay in receiving updated data. Therefore, this approach cannot be applied to applications in which online querying is crucial. IXIA cannot either be used if there are sources which do not give access to the primary-key of data.

10.12198/spekta.v4i1.7845

REFERENCES

[1] L. M. Haas, “Beauty and the Beast: The theory and practice of information integration”, in ICDT, pp. 28-43, 2007

[2] H. Wache, T. Vogele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, S. Hubner, “Ontology-based integration of information | a survey of existing approaches”, in : H. Stuckenschmidt (Ed.), IJCAI{01 Workshop : Ontologies and Information Sharing}, pp. 108-117, 2001.

[3] Andrew Branson, Tamas Hauer, Richard McClatchey, Dmitri Rogulin, Jetendr Shamdasani, “A Data Model for Integrating Heterogeneous Medical Data in the Health-e-Child Project”. 2008.

[4] A. Y. Levy, A. Rajaraman, J. Ordille, “Querying heterogeneous information sources using source descriptions”, in Proceedings of the Twenty second International Conference on Very Large Databases, VLDB Endowment, Saratoga, Calif., Bombay, India, pp. 251-262, 1996.

[5] S. S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. D. Ullman, J. Widom, “The tsimmis project : Integration of heterogeneous information sources”, in IPSJ, pp. 7-18, 1994.

[6] P.J. McBrien and A. Poulovassilis, “Towards Data Visualisation Based on Conceptual Modelling”, ER 2018, Pages 91-99, 2018.

[7] A. Gupta, “Materialized Views: Techniques, Implementations, and Applications”, MIT Press, 1999.

[8] Yi Zhao, Lexin Li, “Multimodal data integration via mediation analysis with high-dimensional exposures and mediators”, Human Brain Mapping. Volume 43, Issue 8, Pages 2519-2533, 2022.

[9] Daneshpour, N. “View Maintenance Expression Improvement in Data Warehouses”, Journal of Modeling in Engineering, 16(55), 101-111. 2018.

[10] L. M. V. Y. V. P. Jarke, M., “Fundamentals of Data Warehouses”, 2nd Edition, Springer, ISBN : 978-3-540-42089-7, 2003.

[11] Li, Y., Feng, A., Li, J., Chen, S., Mumick, S., Halevy, A.Y., Li, V., & Tan, W.C., “Querying subjective data”. The VLDB Journal, 30, 115 – 140, 2020.

[12] Ulrich, H., Kock-Schoppenhauer, A. K., Deppenwiese, N., Gött, R., Kern, J., Lablans, M., Majeed, R. W., Stöhr, M. R., Stausberg, J., Varghese, J., Dugas, M., & Ingenerf, J. “Understanding the Nature of Metadata: Systematic Review”, Journal of medical Internet research, 24(1), 2022.

[13] F. G. V. Lattes, M.-C. Rousset, “The use of CARIN language and algorithms for information integration: The PICSEL sys-tem”, International Journal of Cooperative Information Systems 9 (4), pp. 383-401, 2000.

[14] Gianluca Cima, Federico Croce, Maurizio Lenzerini, “Separability and its Approximations in Ontology-based Data Management”, Semantic Web Journal, 2022.

[15] Ian Harrow, Rama Balakrishnan, Ernesto Jimenez-Ruiz, Simon Jupp, Jane Lomax, Jane Reed, Martin Romacker, Christian Senger, Andrea Splendiani, Jabe Wilson, Peter Woollard, “Ontology mapping for semantically enabled applications”, Drug Discovery Today, Volume 24, Issue 10, Pages 2068-2075, 2019.

[16] Michael Meier, Michael Schmidt, Fang Wei, Georg Lausen, “Semantic query optimization in the presence of types”, Journal of Computer and System Sciences, Volume 79, Issue 6, Pages 937-957, 2013.

[17] Alon Halevy, Cristian Canton-Ferrer, Hao Ma, Umut Ozertem, Patrick Pantel, Marzieh Saeidi, Fabrizio Silvestri, Ves Stoyanov, “Preserving integrity in online social networks”, Communications of the ACM, 66 (2), pp. 92-98, 2022

[18] Michael Meier, Michael Schmidt, Fang Wei, Georg Lausen, “Semantic query optimization in the presence of types”, Journal of Computer and System Sciences, Volume 79, Issue 6, Pages 937-957, 2013.

[19] S. Kermanshahani, H. R. Hamidi, “An Efficient and Adaptive Framework to Access Heterogeneous Health Information Sources”, INFORMATION TECHNOLOGY IN INDUSTRY, Vol 7, No 2, 2019.

[20] A. Simonet, “Types abstraits et bases de donnees: formalisation du concept de partage et analyse statique de contraintes d'integrite”, Phd thesis, Universite Scientique et Medicale de Grenoble, France, 1984.

[21] M. Roger, A. Simonet, M. Simonet, “Bringing together de-scription logics and database in an object oriented model”, in : DEXA '02 : Proceedings of the 13th International Conference on Database and Expert Systems Applications, Springer-Verlag, London, UK, pp. 504-513, 2002.

[22] E. Gay, “Vues osiris sur une base relationnelle”, Memoire d’ingnieur cnam, p-89, CNAM, Centre de Grenoble, France, 1999.

10.12198/spekta.v4i1.7845