Electronic Medical Records

Mohammed F. Audi, Rania A. Hodhod, and Abdel-Badeeh M. Salem

Abstract—**Electronic Medical Record (EMR) relational database is considered to be a major component of any medical care information system. A major problem for researchers in medical informatics is finding the best way to use these databases to extract valued useful information to and about the patient's diseases and treatments. Integrating different EMR databases is a great achievement that will improve health care systems. This paper presents an AI approach to extract generic EMR from different resources and transfer them to clinical cases. The utilized approach is based on retrieving different relationships between patients' different data tables (files) and automatically generating EMRs in XML format, then building frame based medical cases to form a case repository that can be used in medical diagnostic systems.**

Index Terms - **Electronic medical records, frame knowledge representation, relational databases, XML.**

I. INTRODUCTION

ELECTRONIC Medical Record (EMR) relational database is considered to be a major component of any medical care information system. EMR can be defined as a collection of electronic health information about patients. One of the major challenges that physicians are facing is how they can have valued information that can help them to gain greater insight about their patients. Different modern hospitals are using different system for medical records [4].

A doctor's EMR in the office is supposed to enable connection with outside sources of patient data, other clinicians using the same or different EMRs. The desire to connect a clinician with the local system holding all patient data from different resources is an important goal. Health Information Exchange (HIE) in which different large institutions could connect hospitals and academic centers could exchange information with each other is difficult because of different EMR structure of the different medical systems they have [5]. This raises the need for a useful representation of EMR that enables fast and accurate access to knowledge and understanding of the content.

EMR relational databases are collection of patients' data items that are organized as a set of formally-described tables

Mohamed M. Audi, Jr., is a graduate of computer science, Ain Shams University, Cairo, Egypt, (e-mail: mo.fekry@gmail.com).

Rania A. Hodhod, Dr., is an assistant professor of computer science, Ain Shams University, Cairo, Egypt, (e-mail: rania_adel@hotmail.com).

Abdel-Badeeh M. Salem, Prof. Dr., is a professor of computer science, Ain Shams University, Cairo, Egypt, (e-mail: asalem@eun.eg).

from which data can be processed or assembled in many different ways without having to reorganize the database tables. The standard user and application program interface to a relational database is the Structured Query Language (SQL). SQL statements are used both for interactive queries for information from a relational database and for gathering data for reports. In addition to being relatively easy to create and access, a relational database has the important advantage of being easy to extend [2].

An analysis of large and complex systems such as environmental systems linked to socio-economic systems requires several simulation models which makes it so expensive to be applied. These simulation models must be able to interface with each other in the conceptual level which is not applicable and there will be some overlap in their applications domains. Simulation models are normally generating large amounts of data, which need to be explored or mined for the analysis and possible decision process like (data integration) [6].

A large variety of approaches have been proposed in the literature for performing data source integration. Many of them are embedded in more complex systems managing the interoperability and the cooperation of data sources characterized by heterogeneous representation formats. As a consequence, frequently, a data source integration approach is implemented as a module of a more general system [7]. TSIMMIS exploits the self-describing Object Exchange Model (OEM) to represent data sources into consideration. The semantic knowledge is encoded as a set of rules in the Mediator Specification Language (MSL); this enforces source integration at the mediator level. The exploitation of OEM and MSL allows TSIMMIS to integrate heterogeneous and semi-structured data sources [8]. Clio uses database middleware systems as transformation engines for translating data from a source scheme to a target one. In particular, Clio handles both scheme and data transformations within the same integration task. In order to carry out its activity, Clio exploits object-extended SQL functionalities at both the wrapper and the middleware level [9]. LSD exploits machine learning techniques to match a new data source against a previously determined global scheme. In particular, sources which LSD operates upon are XML DTDs. LSD exploits some base learners using different instance level matching schemes; these are trained to assign tags of a mediated scheme to data instances of a source scheme. A Meta learner is used for combining the predictions of each of the base learners [10]. SKAT uses first order logic rules to express match and

mismatch relationships, as well as to derive new matches. The user initially provides application-specific match and mismatch relationships and then validates generated matches [11]. GARLIC exploits the object oriented language GDL for describing the local sources within complex wrapper architecture; the global scheme is obtained by manually unifying the local sources by means of the so-called Garlic Complex Objects. [12].

Recent fashion in knowledge representation languages is the XML usage. XML is a new and powerful technique for internet development. It's a method of defining structure data in a text file. This tends to make the output of these knowledge representation languages easy for programs to parse, at the expense of human readability. XML strength lies in its simplicity to represent data and knowledge [2]. On the other side, providing a convenient structure for object representation can be attained through Frames knowledge representation as they are useful for representing commonsense knowledge, and allow nodes to have structures they can be regarded as three-dimensional representations of knowledge [1]. Frames provide the means to constructing an efficient case repository.

The existing work considers the presence of medical relational databases to extract the needed patient records information. To the extent of our knowledge, there is no single system that handles both the preparation stage of these patients' records through retrieving patient data from different sources with different data structures, and the transfer of the retrieved information to a generalized XML model as part of frame-based representation. This paper proposes a new approach that attempts the above two stages; extraction, and transformation and then the utilization of the generated EPR in a medical diagnostic system.

II. ELECTRONIC MEDICAL RECORDS

The Electronic Medical Record (EMR) is a file kept on a computer containing information about the patient's health. Previously, patient records were kept as hard copies in physical files. The movement of physical files towards the electronic forms allows physicians to query, transfer and handle patients' information in an easy way. An electronic record is created for each service a patient receives from clinical departments, such as radiology, laboratory, or pharmacy, or as a result of administrative action (e.g., creating a claim) [1]. Storing and transferring patient information electronically reduces clinical errors and improves patient safety as well as allowing clinicians to communicate more quickly and accurately and identifying relevant information in an easy way [5].

EMR can be viewed as a clarification of the physician's problem-solving strategy as it contains a problem situation and its physician solution (action). Information contained can be divided into three main parts:

1) *Problem situation*: the state of problem description

documented by the physician.
2) *Solution*: the physician solution given as diagnosis and treatment.
3) *Outcome*: the state resulted when solution is applied on the problem stated before

From the above EMR properties, EMR is seen as quite similar to a medical case content. Accordingly, EMR can be viewed as an abstraction of a clinical case 'Problem Solving' knowledge system. Table 1 shows the similarity between EMR and a clinical case [1].

TABLE I
AN ELECTRONIC PATIENT RECORD VERSUS A CLINICAL CASE

	EMR contents	Medical case contents
Problem	- Personal information - Admission information - Nursing documentation	- Age - Gender - Vital signs - Temperature, pulse, blood pressure - Reports
Solution	- Clinician requests - Progress notes	- Diagnosis - Procedures
Out-come	- Physician examination - Diagnosis reports	- Assessment - Progress status

III. EMR RELATIONAL DATABASE

EMR relational database is a set of tables containing data fitting into predefined categories about the patients. Each table (sometimes called relation) contains one or more data categories in columns. Each row contains unique instance of data for the categories defined by the columns. For example, a typical patient database would include a table that describes a patient with columns for name, address, phone number ….etc. Another table would describe a disease: disease, patient, date. A clinician who uses the database could obtain a view of the database that fits his needs. For example, see all patients who have certain disease after a specific date or see a summary report.

A relation is defined as a set of tuples that have the same the attributes. A tuple represents an object and information about the object. Objects are typically physical objects or concepts. A relation is usually described as a table, which is organized into rows and columns. All the data referenced by an attribute are in the same domain and conform to the same constraints [5].

The relational model specifies that the tuples of a relation have no specific order and that the tuples, in turn, impose no order on the attributes. Fig. 1 shows the relational database terminology.
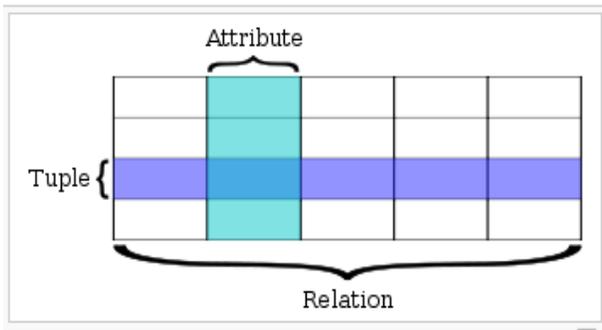
Fig.1 - Relational database terminology.

EMRs normally store their data in a relational database or hierarchical database in "transactional" form. The transactional form includes all information need to conduct the health care enterprise, including "internal" data of little interest to the end consumer/clinician (internal date-time stamps, update codes, workstation origin codes, incremental data updates, etc.) [1].

In some circumstances, there is a case to be made for extracting key clinical data (extraction), cleaning up the data (transformation), and writing (loading) the data into a database specifically designed to ease data analysis; this operation is called data cleansing. This process may be repeated across multiple databases, providing uniform, concept-compatible data in "normalized" form. By performing this process and paring down the quantity of data, the reliability of analysis is enhanced and summarized data becomes available to be extracted and manipulated easily. Fig. 2 shows the above processes [5].
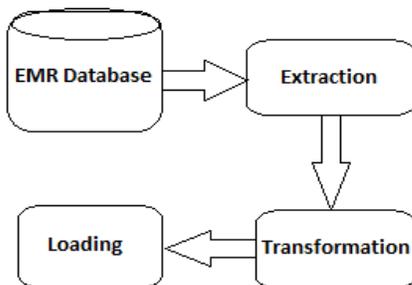

Fig.2 - Data cleansing processes

IV. FRAME KNOWLEDGE REPRESENTATION USING XML

Knowledge representation (KR) is aiming to encoding knowledge in an easy way to facilitate inferences from it. There are three main KR techniques as summarized in. Table 2 Based on the table classification, the best one is the one that uses structural representation (Frames) to represent medical records. The frames advantages can be summarized as following:
1) support representing a structured collection of data (cases).

2) have properties and values like cases.
3) can be linked through their properties (the concept of inheritance).
4) structured nature makes them easier to be extended.
5) can have default values for their properties.
6) can contain multiple methods that can operate on data stored in frames.
7) are not independence (no shared values).

Frame can be described as a network of nodes and relations. A frame represents an object or a concept as a collection of attributes (slots), potentially having values. When a frame is being used, the slots' values can be altered to make the frame corresponds to a particular situation.

Both slot values and slots may themselves be frames. In fact, the most basic kind of facet a slot can have is the value facet. The value facet is the facet of a slot used to hold the data for the slot.

The frame system state can be represented as F: I2 → S, where I - a set of identifiers, S – set of slots of the form <v, d, {Qi}, {Dj}, {Ck}> that include current slot value v ∈ T , default slot value d ∈ T, set of query procedures {Qi } and set of daemons. Query procedures Qi are expressions constructed according to some defined syntax, and daemons are represented by functions that change system state Dj: W → W. A set of slot values T can be of arbitrary structure [5].

TABLE II.
KNOWLEDGE REPRESENTATION TECHNIQUES

	Production Rules	Semantic Net	Frames
Data abstraction:			
Multiple Methods	No	Yes	Yes
Defined properties	No	Yes	Yes
Operate on data	Yes	No	Yes
Inheritance	Yes	Yes	Yes
Classification	Yes	Yes	Yes
Independence	No	Yes	No
Expressive	Yes	No	No

The frame database is closest to the object data model. Each frame database supports four tables. The general conceptual schema of the frame database is presented on Fig. 3.
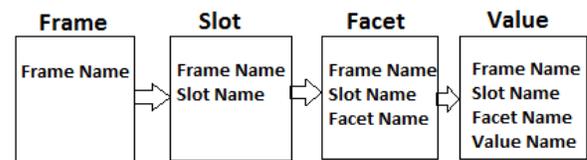

Fig.3 - Frame database schema structure

V. PROPOSED METHODOLOGY

This research aims to highlight one of the most important

challenges that doctors face to have valuable information about their patients, which consequently aid them to provide better service. Patients may have more than one medical record in different institutes that use different medical care systems. The desire to connect the clinician to only one shared system that holds all the patient's medical records has been increased as a must step to improve medical care services. This paper aims firstly, to shed light on the importance of medical records and the need to improve the medical services provided to the patient. Secondly, it aims to extract frame-based clinical cases (EMR) from different databases systems having different structure to form one main case repository.
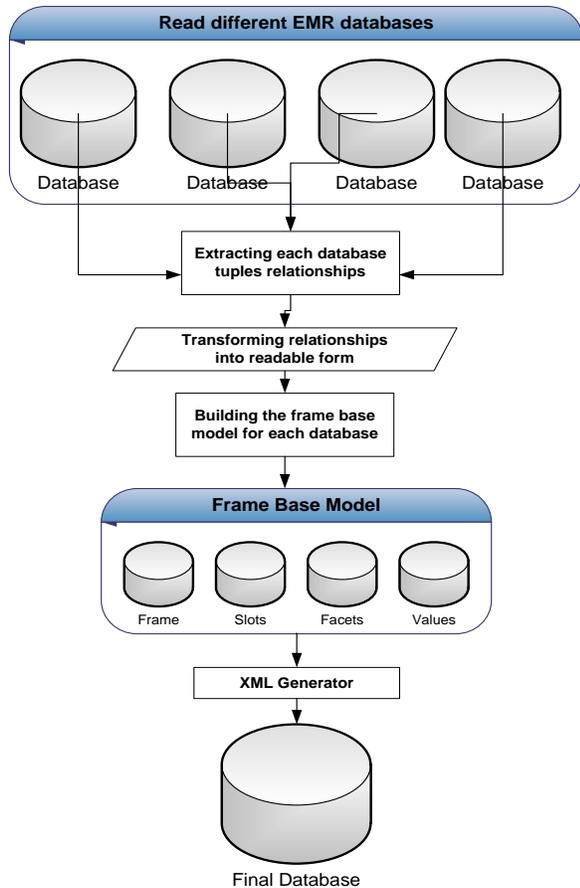

Fig.4 - Architecture for the proposed methodology

Having one case repository for medical records that is located in a single system is a major achievement in health information field. However, the basic problem faced by many medical records vendor is of data integration. The idea is to extract data from multiple different platforms and store it in a uniform mode. This would be of great benefit to HIEs within which different large institutions, academic centers, community doctors, and clinical laboratories can exchange information with each other.

The proposed solution is to make use of the various repositories of electronic patient records. As not all the repositories are well known structure, we intend to find out the internal structure of each repository and then make use of this information to extract medical records to form a general cases

repository. Architecture in Fig. 4 represents the developed methodology. It works as follows:

Phase I. Accessing and extracting different medical database tuples relationships regardless their different structure.

In this phase, each database attributes must be analyzed. Each attribute's characteristics must be defined, resulting in the definition of all database tuples schema. The schema of database tuples includes four attributes: Schema Names, Tables Names, Attributes Names, and Relationships types. This valuable information about the databases makes the databases clear to be manipulated in a correct way.

This phase is responsible for the following activities:

1) Read all database tables.
2) Identify the tables' relationships.
3) Retrieve schema, names, attributes and relationships names with each other.

Fig. 5 shows an example of two related tables and retrieved information

PATIENTS	
Patient_ID	name
1	A
2	B
3	C
.	.
.	.
.	.

PROBLEMS	
Sick_ID	disease
1	Pressure
1	Brain
2	Stomach
2	Malaria
2	Cancer
.	.

RETRIEVED INFORMATION

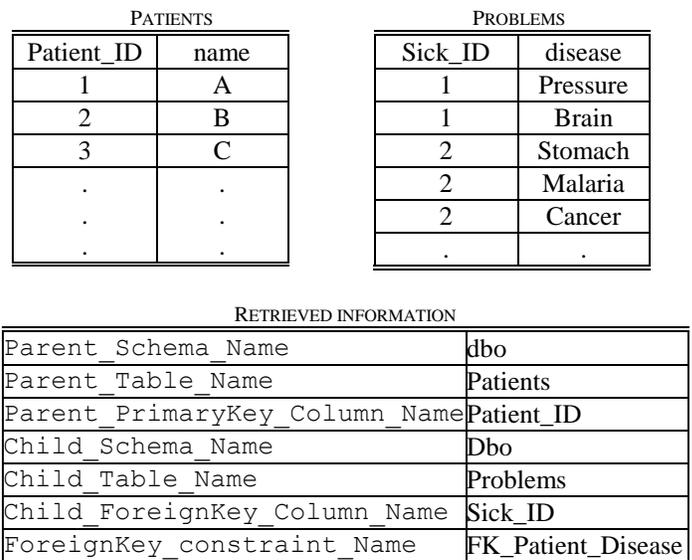|	
Parent_Schema_Name	dbo
Parent_Table_Name	Patients
Parent_PrimaryKey_Column_Name	Patient_ID
Child_Schema_Name	Dbo
Child_Table_Name	Problems
Child_ForeignKey_Column_Name	Sick_ID
ForeignKey_constraint_Name	FK_Patient_Disease

Fig.5 - Two tables related together via different keys names, ERP diagram and retrieved information about their relationship.

This phase may include using an input dictionary to unify relationships attributes names. Fig. 6 shows how we use the dictionary services to match attributes together in order to build a readable form of relationship that can be processed.

Having a data dictionary is a powerful documentation tool for recording the semantics of each attributes and mapping attributes to each other.
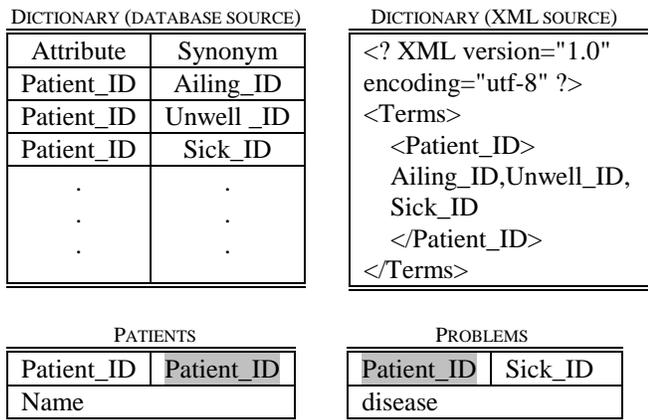
DICTIONARY (DATABASE SOURCE)	
Attribute	Synonym
Patient_ID	Ailing_ID
Patient_ID	Unwell _ID
Patient_ID	Sick_ID
.	.
.	.
.	.

```
DICTIONARY (XML SOURCE)

<? XML version="1.0"
encoding="utf-8" ?>
<Terms>
   <Patient_ID>
   Ailing_ID,Unwell_ID,
   Sick_ID
   </Patient_ID>
</Terms>
```

PATIENTS	
Patient_ID	Patient_ID
Name	

PROBLEMS	
Patient_ID	Sick_ID
disease	

Fig.6 - Two dictionary sources for terms synonyms and the resulting tables after unifying the keys names.

Phase II. Building the frame-based model for each database.

Frame architecture is based on knowledge representation that is classified hierarchy with inheritance relation. Building such model is a critical step because of different structure and complex relationship in different databases.

<! ELEMENT FRAME (Slot +) >

<! ATTLIST FRAME FRAMENAME #REQUIRED >

< ! ELEMENT Slot (Facet +) >

< ! ATTLIST Slot

slotName CDATA #REQUIRED

frameName CDATA #REQUIRED >

< ! ELEMENT Facet (Value +) >

< !ATTLIST Facet

facetName CDATA #REQUIRED

slotName CDATA #REQUIRED

frameName CDATA #REQUIRED >

< !ELEMENT Value EMPTY >

< !ATTLIST Value

valueName CDATA #REQUIRED

< ! ELEMENT Patient ID (Patient ID, Name, Disease)>

Fig.7 - DTD sample

~~The main steps of this process are~~

1) Transforming complex relationship types into simple ones
2) Creating a tree structure by cycles breaking and parent conflicts resolution; ex: when a child table has more than

one parents ones.
3) Specifying the sequence or choosing between the children of a same father.
4) Creating XML Data Definition Types (DTD) from the relational schema of the frame database. Fig. 7 is an example of DTD.

Phase III. Using XML generator to generate frame-based xml files.

By using the database schema, the XML generator grabs data from relational database and builds a dynamic XML documents. Fig. 8 represents the XML generator algorithm
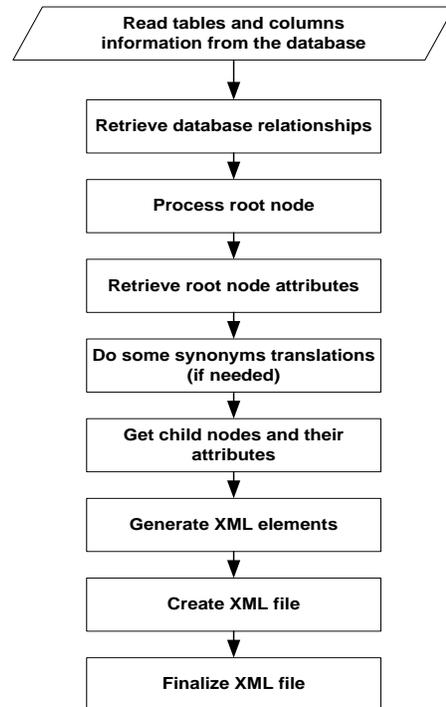

Fig.8 - XML generator algorithm

The main steps of the XML generator algorithm are the following:
1) Get tables and columns information
2) Retrieve tables relationships
3) Define the default namespaces and create a symbolic root element.
4) Define a set of first-level entities which have to be modeled as direct sub elements of the root element.
5) For each root element, loop to get its attributes' names.
6) For each attribute in root element list, get the attributes values.
7) Append values to attributes.
8) Finalizing the generation process of XML file by adding the required data.

The XML generator fetches data from the database and creates an XML document using the existing frame schema by

constructing a query to set fields values for each attributes in the XML file. Fig. 9 shows a sample of generated XML file.

```
<? XML version="1.0"?>
<Frame>
        FrameName = "Patient">
        <Patient_ID>1</Patient_ID>
        <Name>A</Name>
        <Disease>
          <Value>
            Value Name = "Pressure" \>
          <Value>
            Value Name = "Brain" \>
        </Disease>
</Frame>
```

Fig.9 - A sample of the XML document

Phase IV. Building AA integrated database.
   This step is still under development with the intension to using genetic algorithms.

VI. EVALUATION

   The architecture presented in this paper provides the means to collect EMRs of a single patient from different medical care systems databases having different structures and transform them to a unified format that can be used by a general repository. The algorithm was tested on three virtual databases that have different formats and managed to retrieve, transform and save the retrieved EMRs in the new desired format.

Peculiarities of our approach:
1) Many of integration approaches proposed in the literature have been designed for carrying out the integration of predefined well-known structured data sources. On the contrary, the approach we have presented in this paper is capable of handling heterogeneous information sources as it's working on unknown structured data sources. Our approach is capable of handling a large variety of information sources formats.
2) Another interesting peculiarity of our approach is the capability of handling unknown terms and values by using its imbedded dictionary that enables our approach to handle any key-term by getting its synonym. The dictionary itself is considered an advantage because of its dynamically to be filled with any medical terms. Different dictionary formats (XML and database) makes it easy for end-users to fill it in an easy way.
3) The proposed approach is characterized also by of handling null and incomplete data by allowing default values to fill empty slots in frame-based XML medical records.

Bottlenecks that may affect the algorithm:

1) The more dictionary filling process, the more accuracy we have in extracting different terms from the data sources.
2) Many difficulties are encountered when managing unstructured data sources, as it is difficult to get the relationships that these data sources are built on.

VII. CONCLUSION

   This paper presents an AI approach that aims to integrate different medical care systems databases. This approach consists of five main steps. (1) Accessing and extracting different medical databases tuples relationships regardless their different structures, by analyzing each database attribute and each attribute's properties. (2) Transforming retrieved relationships into a readable form, by analyzing the synonym relations between attributes. (3) Building a frame-base model for each database, by analyzing the different relationships and defining a DTD. (4) Using an XML-generator to generate frame-based XML cases. XML proves to provide a simple and clear way of representing proper cases. (5) Building the case-base which is considered a very critical task. This step is still under development with the intension to using genetic algorithms.

   Further research is still needed to implement the proposed XML model based on different resources rather than relational database. Further effort will be aimed at building this dynamic XML generator into a knowledge-based information system.

REFERENCES

[1] A.L. Rector, W.A. Nolan, S. Kay, Foundations for an electronic medical record, Methods of Information in Medicine, 30, 86-179, 1991.
[2] M. Marinov, Using XML to Represent Knolwdge by Frames, 2004.
[3] R.C. Chakraborty, Artificial Intelligence – Knowledge Representation – issues, predicate logic, rules, http://www.myreaders.info, Feb. 02, 2008.
[4] O.S. Kemkar, P.B. Dahikar, Data Analysis of EMR, in Distributed Database with respect to today's E-Health APPS, 2012.
[5] J. Fong, S.M. Huang, a Frame Model Approach for Expert and Database System Integration, 1999.
[6] A. Jolma, A.E. Rizzoli, A review of interoperability techniques for models, data, and knowledge in environmental software, International Congress on Modeling and Simulation, Volume 4, pp. 1745--1750. 2003.
[7] L. Pontieri, D. Ursino, E. Zumpano, An approach for the extensional integration of data sources with heterogeneous representation formats, 2002.
[8] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. Ullman, V. Vassalos, J. Widom, The TSIMMIS approach to mediation: Data models and languages, Journal of Intelligent Information Systems 8 (1997) 117–132.
[9] L.M. Haas, R.J. Miller, B. Niswonger, M.T. Roth, P.M. Schwarz, E.L. Wimmers, Transforming heterogeneous data with database middleware: Beyond integration, IEEE Data Engineering Bulletin 22 (1) 1999.
[10] A. Doan, P. Domingos, A. Halevy, Reconciling schemas of disparate data sources: A machine-learning approach, in: Proceedings of the International Conference on Management of Data 2001), Santa Barbara, California, USA, 2001.
[11] P. Mitra, G. Wiederhold, J. Jannink, Semi-automatic integration of knowledge sources, in: Proceedings of Fusion 99, Sunnyvale, California, USA, 1999.
[12] M.T. Roth, P.M. Schwarz, a wrapper architecture for legacy data sources, in: Proceedings of the International Conference on Very Large Data Bases (VLDB, 1997), Athens, Greece, 1997.