The Open Source release of the MOMIS Data Integration System

Sonia Bergamaschi\textsuperscript{1}, Domenico Beneventano\textsuperscript{1}, Entela Kazazi\textsuperscript{2}, Mirko Orsini\textsuperscript{2}, Laura Po\textsuperscript{1}, Serena Sorrentino\textsuperscript{1}

\textsuperscript{1} DII, Università di Modena e Reggio Emilia
Via Vignolese 905, 41125 Modena - Italy
\{sonia.bergamaschi,domenico.beneventano,laura.po,serena.sorrentino\}@unimore.it

\textsuperscript{2} Data River s.r.l.,
Via Vignolese 905, 41125 Modena - Italy
\{entela.kazazi,mirko.orsini\}@datariver.it

Abstract. MOMIS (Mediator EnvirOnment for Multiple Information Sources) is an Open Source Data Integration system able to aggregate data coming from heterogeneous data sources (structured and semi-structured) in a semi-automatic way. DataRiver\textsuperscript{3} is a Spin-Off of the University of Modena and Reggio Emilia that has re-engineered the MOMIS system, and released its Open Source version both for commercial and academic use. The MOMIS system has been extended with a set of features to minimize the integration process costs, exploiting the semantics of the data sources and optimizing each integration phase. The Open Source MOMIS system have been successfully applied in several industrial sectors: Medical, Agro-food, Tourism, Textile, Mechanical, Logistics. This paper describes the features of the Open Source MOMIS system and how it is able to address real data integration challenges.

1 Introduction

In the past 15 years the problem of data integration has been largely discussed by the research community [6]. With the increasing volumes of data and the growing need to share information, data integration has become a fundamental process in the field of business applications.

The size of the market for data integration tools has been estimated at approximately $1.35 billion as of the end of 2009. A projected five-year compound annual rate of approximately 9.4% will yield a market of approximately $2.1 billion by 2014 [8].

“Customers are seeking low-cost, good enough data integration capabilities”(as pointed out in [8]), for this reason to be competitive compared to big vendors MOMIS is distributed as an Open Source tool. On the other hand an open source version might benefit from the open source community that can develop and make available extensions of the system.

\textsuperscript{3} http://www.datariver.it/
The goal of the Open Source MOMIS system is the minimization of the integration process costs. A data integration project is often developed by integration designers that have a partial or null knowledge of the data sources and of the application domain. Even if the designers have a good knowledge of the application domain, often they are not skilled on the techniques to integrate the data stored in the data sources. In traditional Data Integration Systems, designers have to manually built the integrated schema, defining all the mappings between each global class/attribute and the corresponding local classes/attributes on the local data sources, thus the integration process requires several days/weeks depending on the size of the integration project. Another drawback is due to the fact that designers can see the global result of the integration only at the end of the overall integration process, and it is only at that time that they can refine mappings in order to improve the integrated schema.

To overcome these problems, a first result of integration is semi-automatically derived by MOMIS and proposed to the designer in few minutes; the designer can then improve this integration result, through an iterative refinement process and a set of features (described in details in Section 3). The main features proposed in MOMIS are: (1) a GUI that facilitates the integration process, (2) a set of explore and preview tools that allow the designer to preview the integration result during each integration phase, (3) the possibility to create different unified views to explore the global result of the integration process of the underlying data sources, (4) a suite of tools for semantic annotation of data sources w.r.t. a common lexical reference; these tools allow to import/export the local source annotations, and permit to extend the lexical reference itself with domain glossaries, (5) a preview of the query plan that allows the designer to visualize, for each executed global query, the set of queries that compose the query plan.

MOMIS development started in 1997 and the research activity continued within several national and international projects through the years. MOMIS was exploited in several scenarios, e.g., for the integration of molecular and phenotypic data sources and the development of an integrated information system for cereals breeders in the CEREALAB project and for the integration of several tourism web sites and the development of a Tourism Vertical Web Portal in the WISDOM project. Moreover, the Open Source MOMIS system has been used on real-data sets to integrate clinical data of patients. This work has been conducted in the Olive Tree Project for sharing data about Cancer registries of different countries in the Mediterranean Sea.

The paper is organized as follows: Section 2 presents the MOMIS system, by describing its architecture and by identifying the main phases of the data integration process. In Section 3, we describe the features that have been introduced in the Open Source version of the tool. In Section 4, the web site, documentation, tutorials and community of the Open Source MOMIS system are presented. In the end, Section 5 sketches out the future development directions.

---

4 http://www.cerealab.unimore.it/
5 http://www.dbgroup.unimo.it/wisdom/
2 The Data Integration Process and Architecture

In this section, we will briefly present the MOMIS architecture and the main phases of the data integration process. A full and detailed description of MOMIS is out of our scope and can be found in [5, 2, 4].

MOMIS builds a unified schema, called Global Schema (GS), of several (heterogeneous) data sources (also called local sources), and allows users to formulate queries on it. It follows a Global-As-View (GAV) approach for the definition of mappings between the GS and local schemas: the GS is expressed in terms of the local schemas. MOMIS performs data integration following a virtual approach that preserves the autonomy and security of the original data sources. The GS generation process is composed by four main phases:

1. **Local Source Upload**: (see Figure 1-1) the integrator designer exploits the wrapper tool \(^6\) (see Figure 2) to logically extract the schema of each local source and convert it into the common language ODL\(_I\)\(^7\).

2. **Local Source Annotation**: (see Figure 1-2) the designer is asked to annotate the local sources, i.e. to associate to each class and attribute names (in the following also called terms) one or more meanings w.r.t. a common lexical reference, that in our case is the lexical database WordNet \([7]\). WordNet is a thesaurus for the English language, that groups terms (called lemmas in WordNet terminology) into sets of synonyms called synsets, provides short

---

\(^6\) The type of sources supported at present are MySQL Server, MS SQL Server, Oracle, DB2, PostgreSQL database and JDBC-ODBC sources.

\(^7\) ODL\(_I\), is an object-oriented language, with an underlying Description Logic, deriving from the standard ODMG.
definitions (called gloss), and connects the synsets through a wide network of semantic relationships.

The designer can manually select a base form and the appropriate WordNet meaning(s) (i.e. synset(s)) for each term and/or perform automatic annotation (see Section 3). Moreover, in the MOMIS Open Source version, the designer can extend WordNet with Domain Glossaries (see Section 3). The Local Source Annotation phase is performed by the Global Schema Designer tool (see Figure 2).

3. **Semantic Relationships Extraction:** (see Figure 1-3) starting from the annotated local schemas, MOMIS derives a set of intra and inter-schema semantic relationships in the form of: synonyms (SYN), broader terms/narrower terms (BT/NT) and related terms (RT) relationships. The set of semantic relationships is incrementally built by adding: structural relationships (deriving from the structure of each schema), lexical relationships (deriving from the element annotations, by exploiting the WordNet semantic network), designer-supplied relationships (representing specific domain knowledge) and inferred relationships (deriving from Description Logics equivalence and subsumption computation). The Semantic Relationship Extraction phase is performed by the Global Schema Designer tool (see Figure 2).

4. **GS generation:** starting from the discovered semantic relationships and the local sources schemas, MOMIS generates a GS consisting of a set of global classes, plus *Mapping Tables* which contain the mappings to connect the global attributes of each global class with the local sources attributes. The GS generation is a process where classes describing the same or semantically related concepts in different sources are identified and clustered into the same global class (see Figure 1-4).
The designer may interactively refine and complete the proposed integration result through the GUI provided by the Global Schema Designer tool. In particular, she/he can: modify the proposed global classes and mappings; select the appropriate Join Function for each global class; define Transformation Functions in order to transform the local attribute values into the corresponding global attribute values; and solve possible data conflicts through the definition of Resolution Functions (applied to each global attribute to obtain, starting from the values computed by the Transformation Functions the corresponding value of the global attribute).

Finally, once obtained the desired integration result, a user can pose queries on the GS by using the Query Manager tool (see Figure 2). As MOMIS follows a GAV approach, the query processing is performed by means of query unfolding [3]. The query unfolding process generates for each global query (i.e. a query on the GS) a Query Plan composed by a set of queries:

- a set of local queries that have to be executed on the local sources simultaneously by means of wrapper,
- a mapping query for merging the partial results (defined by means of the join function),
- a final query to apply the resolution functions and residual clauses.

In the Open Source version of MOMIS, we implemented the Query Manager Web Service which allows to integrate MOMIS with other applications (e.g. Business Intelligence solutions). Moreover, a user-friendly Web Application (see Figure 2) has been implemented to guide an end-user, without experience on data integration solutions, to easily compose and execute query on the integrated schema.

3 Features

The MOMIS system has been extended with a set of features and components to address several important data integration challenges and speed up data integration projects:

**Data Integration project.** During the integration process the designer has to face different problems. When she/he starts the integration, even if she/he knows the domain of the underlying data sources, she/he doesn’t know which is the final appropriate integrated schema (Global Schema). The Open Source MOMIS system introduces the concept of Data Integration project, where each project can be composed by several alternative Global Schemas that represent different views of the set of the underlying data sources. The creation of a new project is performed by following few steps: first of all, the designer creates the new project, then uploads the local sources and starts the creation of a new GS by editing each section that compose the integration process; once completed the GS the designer can:
– pose query on the GS created
– upload other sources
– create a new GS on the local sources (or a subset of the local sources)

The Global Schemas can be easily imported/exported from a Data Integration project to other projects.

User-friendly and flexible GUI. Despite advances in both commercial and research tools, data integration solutions are still too hard to use and building an integrated schema may require many staff days and even months. A user-friendly GUI is a critical factor to ensure the success on the market of commercial products. In order to guide the designer through the integration process, a very intuitive, flexible and user-friendly interface has been designed for the MOMIS system. MOMIS has been developed as an Eclipse Rich Client Platform\(^8\) (RCP) application that allows developers to use the IBM’s open source popular Eclipse platform\(^9\) to create flexible and extensible desktop applications. All system components are built as plug-ins of the Eclipse development environment, which supports also an easy incorporation of new tools. As shown in

---
\(^8\)http://wiki.eclipse.org/Rich_Client_Platform
\(^9\)http://www.eclipse.org/
In Figure 3, the MOMIS GUI is divided into three main sections: Source Explorer, Global Schema Explorer, and Global Schema Designer.

**Annotation Tools.** The annotation phase is one of the most critical and expensive steps because it deeply affects the subsequent phases.

Usually, integration projects involve large data sources, with hundreds of tables and attributes, coming from a particular domain of interest (e.g., medical, biological, tourism . . .). The manual annotation of each data source element is unacceptable and the semantic heterogeneity reconciliation must be performed on the basis of a domain-specific lexical reference, also called domain glossary.

Thus, a toolset has been developed to optimize the annotation phase and help the designer during the extension of the lexical reference:

- **Annotation Importer:** the reuse of previous annotations between different GS is an important feature. For this reason, a tool for easily importing source annotations from a GS to another GS has been developed.
- **WordNet Extender:** the WordNet Extender [1] tool enables the extension of the lexical reference with domain glossaries. Extending WordNet has the aim to maintain the semantics of the local schemas as much as possible without
having partial loss of knowledge (if the lexical reference does not contain a satisfactory meaning for a data source element, the designer may annotate it with a similar one) or total loss of semantics (if the designer doesn’t annotate the element at all). An intuitive GUI (see Figure 4) for the extension of the lexical reference guides the designer to perform step-by-step operations such as providing new terms (lemmas), definitions for new concepts (glosses) and building relationships between added concepts and the pre-existing ones. In order to optimize the annotation phase and increase the annotation accuracy, we implemented an automatic annotation algorithm with stemming and stop words removal functionalities. If the lexical reference has been extended, the automatic annotation algorithm associates to each data source element the more recent meaning of the domain glossary, else, it associates to the data source element the most frequent meaning present in WordNet.

- **Hypernym Graph Viewer**: to help the designer to build sound relationships between added synsets and the pre-existing ones we implemented the Hypernym Graph Viewer tool (see Figure 5). The hypernym relationship is a WordNet semantic relationship that connects two synsets where the first generalize the second (e.g. name is a hypernym of first_name); the opposite of hypernym is the hyponym relationship (e.g. first_name is a hyponym of name). The designer can visualize in an interactive graph the hypernym relationships of a specific lemma (more precisely of the set of synsets associated to the lemma) or of a specific synset. The designer can find synsets by keyword search, and focus on a specific synset to view the set of its hypernyms.
In Figure 4 and in Figure 5 we present an example; a designer has to integrate data coming from tourism data sources, in particular data that refers to hotels and camps. The designer does not find satisfactory the meaning associated to hotel. From the Hypernym Graph Viewer she/he can notice that the synset associated to hotel is a hyponym of the synset “a structure that has a roof and walls”. The designer creates a new synset for the lemma "hotel" by introducing the gloss “a lodging that provides accommodation, meals and other services for paying guests”. Then, she/he relates hotel with the synset associated to the lemma "living accommodations" and defines a new hypernym relationship.

- **Lexical shared Repository**: once the lexical reference has been extended the designer can export the domain glossary and reuse it in other projects. WordNet is distributed as a set of data files. The WordNet internal organization has been extrapolated and all the terms, definitions and relationships are stored in an embedded relational database. We have chosen the HyperSQL\(^{10}\) DBMS, a lightweight and Open Source DBMS written in Java. The HyperSQL WordNet database has been embedded and is distributed within the MOMIS system, so that no configuration is required. If an extended lexical reference has to be shared by an organization, a shared repository can be created by using the Open Source MySQL\(^{11}\) DBMS Server to store data. The MOMIS system can be configured to use this shared lexical reference and so, the designer can exploit already defined domain glossaries.

**Data Preview functionalities.** Information Integration solutions require both data management and subject matter expertise. A Data Integration project is often developed by designers that have a partial knowledge of the domain of the data sources. By using the data preview tool the designer can explore the local data source instances and make a preview of the partial integration results during each integration phase. Usually the designer has to deal with large data sources, with hundreds of tables and attributes, and probably some tables/attributes are not significant at the global level view (e.g. an auto-increment identifier). During the source upload phase the designer can explore the tables/attributes of the local sources and select only the ones that are going to be involved in the integration process. As described the annotation phase is one of the most critical and hardly step because it deeply affect the subsequent phases. Even if a set of tools has been developed to optimize the annotation phase, the table and attribute names often are abbreviation, company codes or are not informative of the instances that they contain. So a data preview tool is helpful to choose the right meaning of a term on the lexical reference and so maintain as much as possible the semantics of the local schemas. The GS generated automatically by the system can be refined interactively via a set of implemented editors that help the designer during the definition of Join, Transformation and Resolution Functions. Through the data preview tool she/he can explore the content of global

\(^{10}\) http://hsqldb.org/

\(^{11}\) http://www.mysql.com/
classes and attributes. Accordingly to the information that she/he obtained from the preview she/he can define: how to change the proposed mappings; which is the appropriate Join Function for each global class; which Transformation and Resolution Function should be applied to each local and global attribute.

**Query Plan Viewer.** A relational DBMS gives support to the Query Manager (QMDB) for the fusion of partial results that are stored in temporary tables. We have chosen as DBMS for the Query Manager, HyperSQL, so the installation of the MOMIS system doesn’t need any configuration at all. Through the Query Plan Viewer (see Figure 6), the designer for each executed global query can visualize the set of queries that compose the Query plan, and can make a preview of the data contained in the temporary tables created on the QMDB (see Figure 6), so the query execution is completely transparent to the designer.

4 The MOMIS Toolkit

MOMIS is an Open Source Software released under the GNU General Public License (GPLv2), which allows use, modification and incorporation into Open Source products. We encourage both developers and researchers to download the version 1.1 of the software (from http://www.datariver.it), and to contribute to
the future development of the MOMIS system. The developer documentation is available with the source code. Together with the version 1.1, DataRiver published on the website a detailed user manual and a set of video tutorials to learn quickly how to integrate data sources with the MOMIS system.

MOMIS can serve as an open research platform, providing many useful components that can be extended by developers and researchers. We invite both developers and researchers to discuss specific issues, ideas and design with the DataRiver team. Please contact the DataRiver team if you are interested in contributing to the MOMIS system.

5 Future Work

The Roadmap of the Open Source MOMIS system includes improvements about Provenance, Automatic annotation, Object identification and Collaboration:

1. Provenance (or lineage) describes where data came from and how it was derived. It provides valuable information that can be exploited for many purposes, ranging from statistical resumes presented to the end-user, to more complex applications such as data cleaning (identifying and correcting data errors);

2. Advanced Automatic Annotation techniques will be included for a faster integration process: combination of several annotation methods, also probabilistic methods; abbreviations and acronyms expanded by using the information provided by the schemata and abbreviation dictionaries; compound nouns (composed of more words) automatically interpreted and annotated on the basis of their constituents;

3. Object Identification techniques (also known as record linkage or duplicate detection) identifies instantiation of the same object in different sources. The current technique (exact matching) will be extended introducing advanced methods based on similarity measures;

4. Collaboration environment to enable real-time collaboration between integration designers during each phase of the integration process will be developed. Teams of integration designers will be able to share domain glossaries, annotations, integrated schemas, and whole integration projects, reducing the cost of data integration projects.

6 Acknowledgment

We would like to thank all DataRiver members and collaborators that took part in the re-engineering of the MOMIS system: Alberto Corni, Antonio Sala, Andrea Prandini, Silvano Pancaldi, Daniele Miselli, Sara Quattrini, Enrico Calanchi, Matteo Di Gioia, Fabrizio Orlandi.
References