# A MAS Based ETL Approach for Complex Data

O. Boussaid, F. Bentayeb, J. Darmont

Abstract : In a data warehousing process, the phase of data integration is crucial. Many methods for data integration have been published in the literature. With the development of Internet, the availability of various types of data (images, texts, sounds, videos, databases...) has increased rendering their structuring more difficult. These data that are structured or unstructured are named "complex data". In this paper we propose a new approach for complex data integration, based on a Multi-Agent System (MAS) associated with the data warehousing approach. Our objective is to take advantage of the MAS that are a set of agents to perform the phase of complex data integration. Indeed, we consider the different tasks of the data integration process as services offered by agents. To validate this approach, we have developed a multi-agent system for complex data integration.

*Keywords*: data integration, complex data, ETL, Multi-Agent System, services, agents.

# 1 Introduction

The data warehousing and OLAP (On-Line Analytical Processing) technologies [Inm96, Kim96] are now considered mature in management applications, especially when data are numerical. With the development of the Internet, the availability of various types of data (images, texts, sounds, videos, databases...) has increased. These data that may be structured or unstructured are named "complex data". Structuring and exploiting these data becomes difficult and requires the use of efficient techniques and powerful tools to facilitate their integration into a data warehouse. Its consists in Extracting and Transforming complex data before they are Loaded in the data warehouse (ETL process).

In this paper, we propose a new approach for complex data integration, based on a Multi-Agent System (MAS). Our approach consists in physically integrating complex data into a relational database, named ODS (Operating Data Storage) considered here as a buffer ahead of the data warehouse. We are then interested in extracting, transforming and loading complex data into the ODS.

The aim of this paper is to take advantages of the Multi-Agent Systems that are intelligent programs, composed of a set of

agents, each one offering a set of services, to perform the complex data integration. Indeed, we can assimilate the different tasks of the integration process, which is technically difficult, to services carried out by agents.
*Data extraction*: This task is performed by the agent in charge to extract data characteristics from complex data. The obtained characteristics are then transmitted to the agent responsible for data structuring.
*Data structuring*: To perform this task, the responsible agent deals with the organization of the data according to a well-defined data model. Then, the model is transmitted to the agent responsible for data storage.
*Data storage*: This task is performed by the agent that feeds the database with the data by using the model supplied by the data structuring agent.

In order to validate this approach, we have developed a MAS for complex data integration. This system is composed of a set of intelligent agents offering the different services that are necessary to the integration process of complex data, and is based on an evolutionary architecture that offers a great flexibility. Indeed, this system allows to update the existing services or to add/create new agents.

This paper is organized as follows: section 2 presents a state of the art concerning the data integration approaches and agent technology. In section 3 we expose the issue of complex data integration and our approach. We give in section 4 some advantages of MAS and show how SMA are adapted to carry out this approach via our proposed architecture. Finally, we conclude this paper and we present research perspectives in section 5.

# 2 State of the art

We present in this section an overview of the techniques our proposal relies on, namely those regarding data integration and the ETL process.

Nowadays, two main and opposed approaches are used to perform data integration over the Web.

In the mediator-based approach [Rou02], the different data remain located at their original sources. The user is provided an abstract view of the data, which represents distributed and heterogeneous data as if they were stored in a centralized and homogeneous system. The user's queries are executed through a mediator-wrapper system [GLR00]. A mediator reformulates queries according to the content of the various accessible data sources. A wrapper is data source-specific, and extracts the selected data from the target source. The major interest of this approach is its flexibility, since mediators are able to reformulate and/or approximate queries to better satisfy the user. However, when the data sources are updated, modified data are lost, which is not pertinent in a decision support context where historicity is important.

On the opposite, in the data warehouse approach [Inm96, Kim96], all the data from the various data sources are centralized in a new database, the data warehouse. The multidimensional data model of a data warehouse is analysis-oriented: data represent indicators (measures) that can be observed according to axes of analysis (dimensions). A data warehouse actually characterizes and is optimized for one given analysis context. In a data warehouse context, data integration corresponds to the ETL

process that accesses to, cleans and transforms the heterogeneous data before they are loaded in the data warehouse. This approach supports the dating of data and is tailored for analysis. However, refreshing a data warehouse is a complex and time-consuming task that implies running a whole ETL process again each time an update is required.

The classical ETL process, as its name hints, proceeds in three steps [Kim96]. The first *extraction* phase includes understanding and reading the data source, and copying the necessary data in a buffer called the preparation zone. Then, the second *transformation* phase proceeds in several successive steps: clean the data from the preparation zone (syntactic errors, domain conflicts, etc.); discard some useless data fields; combine the data sources (by matching keys, for instance); create new keys for dimensional records to avoid using keys that are specific to data sources; and build aggregates to optimize the more frequent queries. In this phase, meta data are essential to store the transformation rules and various correspondences. Eventually, the third *loading* phase stores the prepared data into multidimensional structures (data warehouse or data marts). It also usually includes an indexing phase to optimize later accesses.

An agent software is a classical program that is qualified as "intelligent". Intelligent agents are used in many fields such as the networks, on-board technologies, human learning,... An intelligent agent is supposed to have the following intrinsic characteristics: *intuitive* - it must be ready to take initiatives and to achieve the actions that are assigned to him; *reactive* - it must listen to the actions of its environment and act in consequence; *sociable* - it must be able to communicate with other agents and/or users[Klu01]. Moreover, agents may be mobile and can independently move through an acceptor network in order to perform various tasks. A Multi-Agent System designates a collection of actors that communicate with each other [SZ96]. Each actor is able to offer specific services and has a well-defined goal. This introduce the concept of service: each agent is able to perform several tasks, in an autonomous way and communicates the results to a receiving actor (human or software). The MAS must respect the programming standards defined by the FIPA (Foundation for Intelligent Physical Agents) [fip02].

# 3 A MAS based approach For Complex data ETL

## 3.1 Complex data integration approach

Data integration corresponds to ETL phase in the data warehousing process. To achieve the complex data integration, the traditional approach of ETL is not adapted. We presented in this paper our approach to accomplish the extracting, transforming and loading process on the complex data in an original way.

For integrating complex data captured from a web into a Decision Support Database such as a data warehouse, we propose a modelling process to achieve this goal. We first designed a conceptual UML

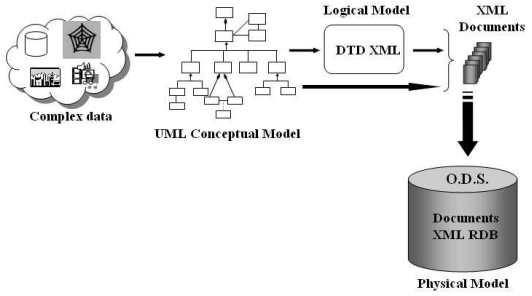


Figure 1: A Classical Modelling Process for Complex Data Integration

model for a complex object representing a superclass of all the types of complex data we consider (text, multimedia documents, relational views from databases)[DBB02b]. The UML conceptual model is then directly translated into an XML schema (DTD or XML-Schema), which we view as a logical model. The last step in our (classical) modelling process is the production of a physical model in the form of XML documents which are stored in relational database. We consider this database as an ODS (Operational Data Storage), which is a data repository that is typically used in a traditional ETL process before the data warehouse proper is constituted. Our objective is not only to store data, but also to truly prepare them for analysis.

## 3.2 A MAS-based prototype

Note that the complex data integration is more complex than a mere ETL task. It requires a succession of tasks to perform the complex data integration. Indeed, we can assimilate the different tasks of the integration process, which is technically difficult, to services may be carried out by agents. To accomplish this aim, we have developed a MAS based prototype, we present its architecture in Fig 2. It is based on a platform of generic agents and we have instantiated five agents offering services allowing the complex data integration. The purpose of this collection of agents is to perform several tasks. Each agent is able to offer specific services and has a well-defined goal.

The first main agent created in our prototype, is *MenuAgent*, who pilots the system, supervises agent migrations and indexes the accessible sites from the platform. Some others default pilot agents help in the management of the agents and provide an interface for the agent development platform.

The essential of the work of integration process is achieved as services about collecting, structuring, generating and storing data, provided by the remaining agents which we present in the next section.

To develop our prototype, we have build a platform using JADE version 2.61 [jad02] and Java language [jav02], that is portable across agent programming platforms.

## 3.3 A Complex Data ETL

*Extracting*

Recall that our modelling approach corresponds to the complex data integration processus. The conceptual level permits the user to select the data and to establish its analysis goals. The *Extraction* phase is thus made by *DataAgent* which collects the data concerning the documents. this is consist on extracting the attributes of the complex object that has been selected by the user. A particular treatment is

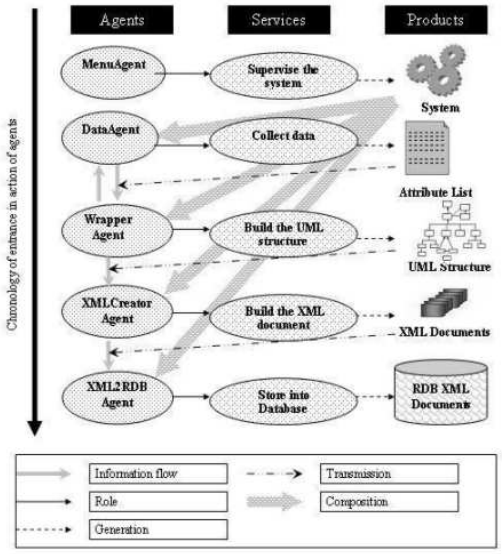


Figure 2: A MAS Based ETL Architecture for Complex Data Integration

applied depending on the subdocument class (image, sound, etc.) of the UML class diagram, since each subdocument class bears different attributes. *DataAgent* uses three ways to extract the actual data: (1) it communicates with the user through graphical interfaces allowing a manual capture of data ; (2) it uses the standard Java methods and packages; (3) it uses the ad-hoc automatic extraction algorithms [DBB+02a]. Our objective is to progressively reduce the number of manually-captured attributes and to add new attributes that would be useful for later analysis and that could be obtained with data mining techniques. This work is completed with *WrapperAgent* that instantiates the UML structure based on the data supplied by *DataAgent*.

*Transforming*

The logical level coincides with the *Transforming* phase. The UML conceptual model is then directly translated by the*XMLCreator* agent into an XML schema (DTD or XML-Schema), which we view as a logical model. XML is the format of choice for both storing and describing the data. The schema indeed represents the meta data. XML is also very interesting because of its flexibility and extensibility, while allowing straight mapping into a more conventional database if strong structuring and retrieval efficiency are needed for analysis purposes.

*Loading*

The last level in our (classical) modelling process corresponds with the *Loading* phase and consists on the production of a physical model in the form of XML documents and their loading into a relational database. This is achieved by *XMLCreator* and *XML2RDBAgent* agents. The *XMLCreator*'s principle is to parse the XML schema recursively, fetching the elements it describes, and to write them into the output XML document, along with the associated values extracted from the original data, on the fly. Missing values are currently treated by inserting an empty element, but strategies could be devised to solve this problem, either by prompting the user or automatically. The XML documents obtained with the help of *XMLCreator* are mapped into a relational database by *XML2RDBAgent*. It operates in two steps. First, a DTD parser exploits our logical model (XML schema) to build a relational schema, i.e., a set of tables in which any valid XML document

(regarding our DTD) can be mapped. To achieve this goal, we mainly used the techniques proposed by [eA00b, ea00a]. Note that our DTD parser is a generic tool: it can operate on any DTD. It takes into account all the XML element types we need, e.g., elements with +, *, or ? multiplicity, element lists, selections, etc. The last and easiest step consists in loading the valid XML documents into the previously build relational structure.

# 4 Justification

The variety of data types (images, texts, sounds, videos, databases...) increases the complexity of data. It is necessary to structure them in a not classical way. Because data are complex, they necessitate more information. Furthermore, it is important to consider this information and to represent it in the form of meta data. Then, the choice of the XML formalism is fully justified. Since our proposal is based on the classical modelling process, it allows the user to determine what are his/her analysis objectives, to select how to represent data and how to store them into a database. It constitutes a whole process permitting to carry out the complex data integration. Such an objective is also the one of the ETL process.

Our proposed process necessitate several tasks which must repetitively made. These tasks are not necessarily sequential and are assimilated to services offered by well-defined agents in a system intended to achieve such integration process. With a view to reach this goal, we have developed a MAS-based prototype that is based upon a flexible and evolutive architecture on which we can updated services, and even create new agents to consider data refreshing, analysis and so on.

# 5 Conclusion and Perspectives

In this paper, we proposed a new approach for complex data integrating based on both data warehouse technology and multi-agent systems. This approach is based on a flexible and evolutive architecture on which we can add, remove or modify services, and even create new agents. We developed then, a MAS-based prototype that allows this integration with respect to these following three steps of the ETL process: Two agents named respectively, *DataAgent* and *WrapperAgent* are charged to model the input complex data into UML classes; the *XMLCreator* agent translates UML classes into XML documents that are mapped in relational database with the *XML2RDBAgent*. Moreover, note that the different agents that compose our system are mobile and the services they propose coincide with the Extraction, Transformation (ETL) and Loading tasks of the data warehousing process.

We plan to extend the services offered by our MAS-based prototype, especially extracting data from their sources and analyzing them. For example, the *DataAgent* can converse with online search engines and exploit their answers. In the other hand, we can create new agents in charge to model data in the multidimensional way and apply analysis methods like OLAP or data mining.

# REFERENCES


[DBB+02a] J. Darmont, O. Boussaid, F. Bentayeb, S. Rabaseda, and Y. Zellouf. *Multimedia Systems and Applications*, volume 22, chapter Web multiform data structuring for warehousing, pages 9–27. Kluwer Academic Publishers, 2002.

[DBB02b] J. Darmont, O. Boussaid, and F. Bentayeb. Warehousing web data. In *4th International Conference on Information Integration and Web-based Applications and Services (iiWAS'02)*, pages 148–152, Bandung, Indonesia, 2002.

[ea00a] G. Kappel et al. X-ray - towards integrating xml and relational database systems. In *19th International Conference on Conceptual Modeling*, pages 339–353, 2000.

[eA00b] R. Anderson et Al. *Professional XML Databases*. Wrox Press, 2000.

[fip02] Fipa web site. http://www.fipa.org, 2002.

[GLR00] F. Goasdoué, V. Lattès, and M.C. Rousset. The use of carin language and algorithms for information integration: the picsel system. *International Journal of Cooperative Information Systems (IJCIS)*, 9(4):383–401, 2000.

[Inm96] W.H. Inmon. *Building the data Data Warehouse*. John Wiley & Sons, USA, 2 edition, 1996.

[jad02] Jade web site. http://sharon.cselt.it/projects/jade/, 2002.

[jav02] Java's official website. http://www.sun.java.com, 2002.

[Kim96] R. Kimball. *The data warehouse toolkit*. John Wiley, 1996.

[Klu01] M. Klusch. Information agent technology for the internet: A survey. *Journal on Data and Knowledge Engineering, Special Issue on Intelligent Information Integration, D. Fensel (Ed.)*, 36(3), 2001.

[Rou02] M.C. Rousset. Knowledge representation for information integration. In *International Symposium on Methodologies for Intelligent Systemes (ISMIS)*, 2002.

[SZ96] K. Sykara and D. Zeng. Coordination of multiple intelligent software agents. *International Journal of Cooperative Information Systems*, pages 1–31, 1996.