Knowledge Management for Electronic Commerce applications

Keywords:

Electronic Commerce,
Knowledge management,
Intelligent Integration Of Information,
Virtual catalogs

Acknowledgements

Preface

Index of Figures

Foreword

This work summarizes the activities developed during the Ph. D studies in Information Engineering.

It is organized in two parts.

The *first part* describes the Knowledge Management Systems and their applications to Electronic Commerce. In particular, a technical and organizational overview about the most critical issues concerning the Electronic Commerce applications is presented. This part is the result of a two years long research carried out in cooperation with Professor Enrico Scarso within the interdisciplinary – ICT and business organization – MIUR project "*Il Commercio Eletronico: nuove opprtunità e nuovi mercati per le PMI*".

The *second part* introduces the Intelligent Integration of Information (I^3) research topic and presents the MOMIS system approach for I^3. It outlines the theory underlying the MOMIS prototype and focuses on the generation of virtual catalogs issues in the electronic commerce environment exploiting the SI-Designer component. A new MOMIS architecture, based on XML Web Service, is finally proposed. The new architecture not only aims at addressing specific virtual catalogs' issues, but it also lead to a general improvement of the MOMIS system.

1 Introduction

It is a widely spreading opinion in the academic literature that managing knowledge strongly differs from managing information, and consequently that installing a sophisticated information and communication technology (ICT) infrastructure does not end all the knowledge management (KM) activities. On the contrary, according to various KM scholars [47],[35],[55],[59], equating knowledge management to information management (IM) is one of the worst mistakes that firms can make [23],[60][1].

This does not mean, however, that there are no relations between ICTs, knowledge and knowledge management, and especially for two reasons.

First, the ability of these technologies to rapidly and efficiently elaborate, store and transfer a great amount of information within and among organisations, practically without time and space limits, may be of great help in the knowledge creation and sharing processes[2]. The role of ICTs in supporting knowledge management is widely highlighted by [1], who analyse the contribution given by these technologies in the four KM processes of knowledge creation, storing and retrieval, distribution and application.

Second, it must be recalled that a two-way relation between knowledge and information may be identified, given that on the one hand knowledge creation often relies on available information, on the other hand the development of relevant information requires the application of context-specific knowledge [55]. What is more, data are created from information by putting information into a

[1] On this point, [47] affirms that "the great trap in knowledge management is using information management tools and concepts to design knowledge management systems".

[2] According to [38] ICTs, and specifically the Internet, may be particularly useful in exchanging information between organisations in that: 1. they may be effective in lowering at least some temporal and physical barriers; 2. they may facilitate the access to (and the retrieval of) information bases storing data; 3. they may help locate the various elements relevant to the information sharing process.

predefined data structure that completely defines its meaning[3], and this is done by someone who uses his/her domain knowledge [61]. On this point, [23] argue the existence of a knowledge-information cycle, compounded by four consecutive steps: information creation, information use, knowledge creation and knowledge use.

Hence, there is no doubt that information management systems (and IT) may assist in creating new knowledge [23], and that one of the most challenging KM activities is to implement IT-based productivity improvement in information management. Nevertheless, we need to more deeply discuss the ability (and the limits[4]) of ICTs in supporting KM processes, especially when different organisations are involved and knowledge is exchanged through the Internet, e.g. during web-based transactions.

As a point of fact, the exchange and management of knowledge and information flowing inside the Internet still remain an open issue. Many and different problems exist, as the results of the interdisciplinary (ICT and business organization) research - reported in [59] - states; the most important ones are:

- the difficult retrieval of really useful information. The vast volume of data, news and facts available on the Internet coming from several different sources produces the *information overload* phenomenon [59], that is a condition of excessive information that turns out in a substantial lack of valuable information;
- the extreme diversity of the information flowing inside the web, often arranged according to heterogeneous technical solutions, may hinder the access to the data and the actual interaction between trading partners;

[3] An example is the storage of data in a semantically well-defined database.

[4] [45] underlines the existence of three key myths affecting KM (IT-based) technologies, from which the firms has to beware of. They regard the fact that these are able to "automatically": deliver the right information to the right person at the right time; store human intelligence and experience; distribute human intelligence.

– the security policies required by a permanent connection to the Internet. The protection devices implemented to preserve the local networks from malicious access attempts leads often to a lack of integration between EC applications and internal information system preventing firms to actually streamline their business processes.

The present work deals with the heterogeneity (format, language, semantics, and so on) of data and information coming from different sources, and the effects of this heterogeneity on electronic transactions. It refers to the issue of the management of web content, and the question concerning the "virtual" aggregation of contents drawn from various web-sites.

More Specifically, we addressed the information overload and diversity issues by exploiting the MOMIS (Mediator envirOnment for integration of Multiple Information Sources) system. The MOMIS follows an Intelligent Information of Information approach (I^3) to create a single virtual view of multiple, heterogeneous information sources. In particular we experimented the SI-Designer component of the MOMIS system for creating virtual catalogs in the Electronic Commerce environment. SI-Designer is a semi-automatic software tool tailored to assist the designer in the creation of an integrated virtual view of all the data source involved in the integration process. The results of this experimentation are widely documented in [10],[11],[12].

The work on the MOMIS' SI-Designer component, indeed allowed us investigate the relationships existing between Electronic Commerce Technologies (ECTs), knowledge and knowledge management. Furthermore, the cooperation on these topics with Professor Enrico Scarso yielded relevant outcomes in the fields of business organization and Knowledge Management documented in [14] and [15]

The experimental results on SI-Designer suggested a development of the original MOMIS architecture. In particular, the opportunities offered by the XML Web Services appeared to be ideal for overcoming some emerged issues

concerning security policies and data sources' description. XML Web Services are self-contained, modular applications that can be described, published, located and invoked over a network, generally, the World Wide Web. The exploitation of XML Web Services led to a proposal for a new architecture which will be described along with its implementation.

The work is organised as follows. In the next section the role of ICT in knowledge management and the different phases of a commercial transaction are described. This is followed by the results of the research about the most significant technical and organizational issues involved by ECTs. In the fourth section the MOMIS system and its integration process are introduced. After that, the integration process performed by the SI-Designer component is analysed according to a KM perspective. In chapter five the XML Web Services-based architecture for the MOMIS system is presented. Finally, some concluding remarks about the framework's new features and the relationship between knowledge management and electronic commerce are outlined.

2 Knowledge Management Systems and Electronic Commerce information needs

Knowledge Management Systems refer to a class of information systems suited to manage organizational knowledge. That is they are IT-based systems developed to support the processes of knowledge creation, storage/retrieval, transfer and application: examples ranges from finding an expert using online directories to learning about customer needs and behaviour by analyzing transaction data. Therefore it is almost impossible to identify a single role for IT in knowledge management [3].

2.1 Knowledge and Knowledge Management

Defining knowledge has been a crucial question since the classic philosophers' era. Rather than explore the different meanings of the term knowledge, we are interested in the views of knowledge as referred in the Information Technology literature: this will enable us to focus on the technical solutions proposed in this work from a knowledge management perspective.

The definitions of knowledge given by various ICT scholars ([62],[44]) seems to classify data, information and knowledge. Nevertheless, the attempts to establish a hierarchy among data, information and knowledge rarely survive a deep evaluation [3].

What is often unclear is when information becomes knowledge. Consistently with[61],[1]) we assume that information is converted to knowledge when it is processed in the mind of individuals and knowledge becomes information once it is articulated and presented in form of text, graphics, words or other symbolic forms.

The main implication of this definition, from an IT point of view, is that systems designed to support knowledge management should be tailored to enable users to assign meanings to information and to capture some of their knowledge in information and data.

As a not-shared knowledge has a limited value for any organization, the major challenge for a knowledge management system is to capture and integrate knowledge rather than create it. In [1] the ability to integrate and apply specialized knowledge of organizational members is deemed fundamental for ability of firms to create and sustain competitive advantages. Hence in absence of an explicit strategy to integrate knowledge, in organizations the communication and the information sharing have only a random effect at best.

2.2 Knowledge Management Systems

As the number of a firm's valuable information sources continuously grows, organizing and conceptualizing knowledge becomes problematic. Organizations, furthermore, are becoming more geographically dispersed. Thus a knowledge management system must provide a rich set of features to address those broad requirements.

Adopting the Bowman's classification of Knowledge Management Systems (KMS in the remainder) [25] two basic features are text search and retrieval and taxonomy definition.

The text search feature should be an enhancement of a search engine. Everyone has experienced the case where the results of a search are so voluminous as to be useless. A search engine should help users narrow down the items to be reviewed by ranking the items according to their relevance to the search expression. Relevance ranking of search results is a feature that takes on increased significance as the quantity of the information increases. Techniques for ranking documents include counts of the keywords, inferences, etc. For organizations operating in particularly volatile environments, the emergence of

new terms is an ongoing challenge. The ease with which new terms can be defined and associated with existing materials can be an important for the maintenance of the system

The definition of knowledge maps, or taxonomies, to group information into categories relevant to the organization is another fundamental feature of a KMS. Once the materials are linked to the knowledge categories, this feature becomes an alternative retrieval method that supplements the text search In some cases, users may be uncomfortable identifying keywords and may wish to start a search by looking at the predefined knowledge topics. This can reduce the number of irrelevant materials that are often identified by the search engines. Definition of the knowledge categories can be accomplished by human designers or through the use of software tools that analyse a collection of materials and suggest a classification scheme based on the observed content. These tools utilize techniques such as word occurrence, noun phrase extraction, and inferencing to develop recommended taxonomies.

The current state of the automatic classification technology is such that it is advisable for humans to review and refine the recommended classification scheme.

Maintenance of the knowledge taxonomy scheme can become a significant challenge. New categories that are important to an organization may emerge and others will decline in today's rapidly changing environment. The ability to add, merge, and delete topical categories is critical to maintaining a current taxonomy. Furthermore, it is important that the repository technology provide the ability to link existing materials to the new categories, regardless of whether the new terms appear in the documents. Automatic reclassification becomes critical as the load of information increases. Furthermore the need of retrieving information from legacy systems is not infrequent. Thus the effectiveness of a KMS rely even on its ability to interact with existing databases, file servers and document management systems.

2.3 Electronic Commerce applications and KMS

According to the given definition of KMS, the electronic commerce platforms support the process of knowledge management by organizing products and/or services, providing advanced search features and supporting business partners across the different phases of a business transaction.

2.3.1 Business transactions

With the aim of understanding the information needs brought about by EC, the various phases which constitute a business transaction are here described, together with the possible role played by ICTs to fulfil such needs. In doing this, we refer to the model suggested by Gebauer and Scharl ([33])[5], who single out the following four stages:

information: buyers and sellers reach out the world in search of information needed to find the seller/buyer and contact it. This phase comprises both searching for a particular source of information and finding required information. Buyers may locate information sources, use them to scan product listings, obtain offerings from potential suppliers, and gather additional information about products, vendors, or transaction-specific requirements. Sellers, instead, may create information, store and make it the most accessible and clear possible. Different EC applications are available to support these activities, such as electronic catalogs, search tools, configuration support for complex purchases, workflow routing for approval, and so on.

negotiation: once in contact, buyers and sellers exchange information about product/services offered, prices, delivery time, exact identity of the traders,

[5] The literature is full of models describing the transaction processes (a brief review is in Gebauer and Scharl, 1999), which however differ among them only in a few details.

payment systems, and all the other information required to define a "trusted" transaction. When a firm is not capable of running its own site, third party exchanges may be needed [53]. Likewise, certificate authorities (either public or private) are required for the trusted identification of trading partners. IT support negotiations in various ways, by providing transaction information and decision support, by identifying new bargaining options, and by providing additional information, such as the volume of previous business, supplier performance, or spending patterns.

settlement: a completed transaction requires: delivery and other logistics, payment, and fiscal performance. Banks and credit card networks play a key role, as well as the related services for payment security. Of course, logistics implies physical activities, although the Internet may be vital for faster co-ordination. The settlement phase may include some form of mutual performance monitoring. In this phase the focus is on execution and efficiency. ICTs to support transaction settlement include EDI systems, and various tools to process orders internally and between transaction partners, facilitate order tracking, and support payment processes.

after-sales and transactions analysis: help-on-line and similar services may be important in case of some goods; furthermore, statistic reports and economic monitoring of transactions may help to adjust offerings and prices. Capturing, storing, and managing data are vital at this point. Similar to the first phase, it is mainly information that is being exchanged between buyer and seller. The electronic support of after-sales activities ranges from simple electronic mail services to automated helpdesks and sophisticated electronic maintenance manuals. Data warehousing applications support the storing, accessing and processing of large amounts of data concerning past transactions.

Even though information plays a role in every phase of the business transaction (as testified by the possible use of ICTs in all of them), without doubt

the first one exhibits a predominant information content[6]. In particular the information phase requires [33]:

- a fast access to a broad range of information (such as lists of firms, product listings and other additional information);
- a strong search functionality and navigational aids (able to distinguish between useful information and garbage);
- user interfaces that are easy to operate.

Typically, EC platforms try to satisfy these requirements by means of directories of buyers and/or suppliers and electronic product catalogs based on classification schemes and providing some form of content management and search tools.

2.3.2 Electronic Commerce and contents management

Although many attempts have been made to establish commonly accepted standards to categorise and list businesses, products and services[7], efficient solutions has not yet developed. As a result, the integration and management of catalog content still involve so much hand work that such activities have became one of the key success factors for the providers of such systems and services.

It must be underlined that content management strongly differs from the simple aggregation and display of the collected information. It, in fact, requires that information has processed with the aim of making it really useful (i.e. valuable for the end-user), and this in turn calls for deep knowledge of the possible utilisation of what has been elaborated. Furthermore, accommodating multiple content sources in a single one[8] is a complex task: a not coherent aggregation, in fact, can generate inconsistent and wrong information. To avoid

[6] This is also the reason why this phase is most frequently accomplished by EC applications

[7] It is sufficient to remember here some of the most important initiatives aimed to develop overall standard for EC, as UDDI (Universal Discovery Description and Integration) project promoted by Ariba, IBM and Microsoft to create a universal business "phone book"; RosettaNet that defines the data properties, messaging, transaction services and so on for the high-technology industry; BizTalk started by Microsoft is a set of guidelines for how to publish schemes in XML.

this, data need to be classified, cleaned, and commerce-enabled though a combination of software tools, processes and experts of the field [54].

In short, all the activities (and applications) aiming to extract and integrate information from heterogeneous (i.e. not directly comparable) sources and to search and locate useful information inside them, may be rightly regarded as critical to EC. The point here is that all these activities require a deep knowledge of the specific business context to be effectively done and, by generating new information, generate new knowledge: this is the reason why they may be rightly regarded as KM activities.

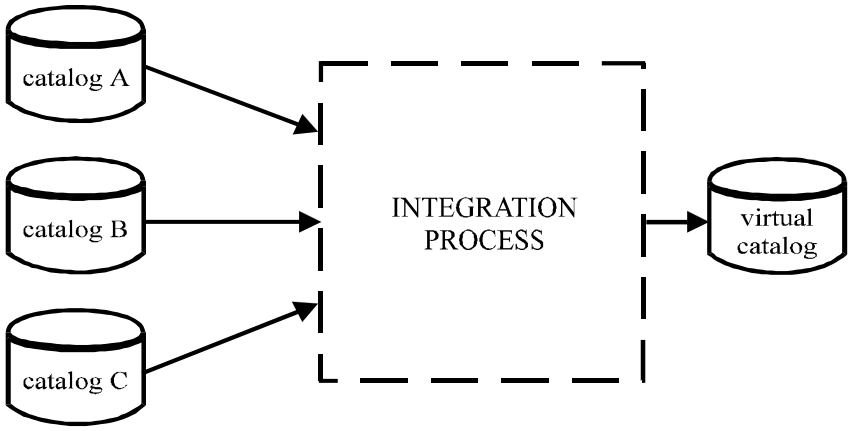

Figure 1- The creation of a virtual catalog from multiple heterogeneous sources

For the same reason, in chapter 4 a framework able to support the integration among different information sources to construct a unique "virtual catalog" (Figure 1) will be presented and discussed according to a KM perspective. In particular, it will be examined which parts of the integration process can be automatically done by the software applications, and which others still need the

[8] An "electronic catalog" must contain content that users can understand, search and maintain. A comprehensive, consistent and complete content enable users to quickly find, identify and verify products and services they are looking for [54].

involvement of a field expert. The aim is to discuss if and how the process performed by the framework can be usefully interpreted following such a point of view, and which (theoretical and practical) indications can be derived from this kind of analysis.

3 Technical approaches to Electronic Commerce: known issues

Given the ability of technical infrastructures to heavily influence the strategies and the organization of businesses, this section aims at reviewing the major issues of the electronic commerce platforms and in particular:

- to individuate technical back-logs preventing a widespread diffusion of the electronic commerce
- to give indications about the most important technical features that both e-commerce platforms and applications should provide in order to be really useful for business
- to indicate the newest trend in the e-commerce technologies.

3.1 Electronic commerce definition

In order to consistently define the technical features and, more in general, the technologies to be analysed, a clear definition of e-commerce must be given.

Both academic and technical literatures bring a variety of definitions for the electronic commerce. Thus the term electronic commerce is used with reference to many concepts that are not rarely very different from each other.

Holsapple and Singh [39] have recently collected and analysed a relevant set of definitions. On the basis of the approach followed in studying electronic commerce they indicate five classes (perspectives):

- *Trading view*: electronic commerce is defined as a set of computer-based tools designed to support business transactions. Definitions of such a kind usually underline the concept that is most commonly referred when thinking about electronic commerce. This perspective is based on the idea of business transactions and on the different existing

ways to accomplish a transaction over a computer network. Viewing electronic commerce under this perspective means focusing on the different kinds of purchases/sales allowed over a network, on those phases of a business transaction that can be executed through electronic tools, on the behaviours of the economical agents acting online.

- *Information exchange view*: electronic commerce is defined as that set of information exchange processes which underlies any business transaction. The starting point is that the materials' flow involved in any business transaction has a secondary role with respect to the role played by information that enables, supports and governs the flow itself. Information not only goes along with the transaction but it also precede and follows the transaction. Definitions of such a kind consider any business as an information business. Therefore information is not merely a support element for the value chain, but it is a basic source of value.
- *Activity view*: focuses on those activities that can be accomplished through electronic commerce technologies. This perspective moves from the assertion that electronic commerce applications may affect a variety of activities both inside and outside a firm.
- *Effects view*: it set the focus on the effects brought by the adoption of an electronic commerce solutions. Definitions of such a kind highlight how electronic commerce may affect the structure of costs in a firm as well as the relationships with customers and the business process reengineering. The topical point, under this perspective, are the results expected after the adoption of an electronic commerce application and how those results can be achieved.
- *Value chain view*: this perspective aims at individuating the role played by the new technologies and information and how them can be exploited to create a competitive advantage. Definitions of such a kind highlight how electronic commerce applications may lead to new,

previously unexplored, organizational solutions. New technologies are considered as fundamental to coordinate the different components of the value chain.

Starting from that analysis and considering that some of the proposed definitions not rarely focus on complementary faces of the electronic commerce, the authors introduce an integrated definition to synthesize the different perspectives:

Electronic commerce is a way to achieve business objectives that exploits new technologies to exchange the information needed by activities within the value chain and allows the decision making that informs the activities themselves.

This first definition is then enhanced by introducing the concept of knowledge management (proposed along the previous chapter). Knowledge management becomes a key topic when talking about electronic commerce, since electronic commerce is not only a mere information exchange, but it creates value only when used as an effective tool to manage knowledge.

Finally, the proposed definition is the following:

Electronic commerce is a way to achieve business objectives that exploits new technologies to manage knowledge in order to enable both the execution of activities within the value chain and the decision making that informs the activities themselves.

The adopted definition, which will be used in the remainder, basically implies two advantages. First, this definition introduces a broader view about the electronic commerce which affects much more than the activities involved by a business transaction. This view is consistent with the trend that replaces the term “electronic commerce” with the wider “electronic business” including all those business activities that are supported by the information and communication

technologies (ICT). Second, this definition connects two different business scopes sharing a variety of common resources, first of all knowledge as key to any competitive advantage.

In other words, adopting the given definition means, on one hand, taking into account all the technologies that can be exploited to manage the knowledge needed to plan, coordinate, run and check all the value chain activities and to take decisions about the activities themselves. On the other hand it means considering as electronic business applications those based on technologies enabled to manage both internal and external organizational processes, not necessarily connected to trading tasks.

3.2 Technical approaches to electronic commerce: major issues

Electronic commerce application experienced a variety of technical and conceptual issues that heavily influenced their diffusion in the recent past. In the following a list of the major issues is presented.

Information available over the Internet is often excessive as well as lacking, hard to find, useless, not rarely incomprehensible. This is a crucial issues especially when considering the new frontier of electronic commerce: the collaborative commerce [53]. This new paradigm focuses on the activities that precede and follow a business transaction rather than investigating the transaction itself. Collaborative commerce widens the concept of electronic commerce introducing a new model based on the sharing of a broader set of information. Under this perspective, the possibility of managing unstructured or semi-structured information (documents, contracts, etc.) becomes a critical factor in the diffusion of electronic commerce application, since only the 20% of information managed by firms can be considered as structured information. Problems in accessing and exchanging non-structured information cause electronic commerce applications to be basically adopted in indirect goods procurements only.

Widening the collaboration scopes, in fact, means enabling trading partners to easily exchange documents of different types, stored in different formats and in different places.

The contents of Internet sites and portals is another major issues that results not disconnected from the previously presented information overload problem.

Content management (catalogues building and maintenance for example) turned out as a very time consuming task. Activities of such a kind requires, on one hand, to be supported by dedicated software applications, but they rely, on the other hand, on a "manual" review from experts operators. The presence of a "domain expert" is required by the wide majority of marketplaces and e-business applications. The domain expert is responsible for defining taxonomies, choosing the proper terms to be used, maintaining and updating the application each time that new contents have to be inserted or obsolete information must be deleted. A typical content management issue is making out potential trading partner lists. That is creating precise lists of products and business running any electronic commerce solution.

Several basic features, such as online payments, are insufficient or even missing. In many cases applications seem to promise much more than they really offer. With respect to some electronic commerce implementations, it is more direct and easy to use a phone call than to access the offered online services. The lack of suitable software tools, not only affects payments but also – and seriously – search features. The incapability to take into account semantics, make the search results useless especially when accessed by inexperienced users.

Scalability, that is the opportunity to gradually implement electronic commerce solutions reducing their organizational impact, is not rarely insufficient.

Scalability can be analysed under two dimensions: transactional and functional:

Transactional scalability affects the number of executed transaction and involves the number of trading partners.

Functional scalability concerns the number and the complexity of activities and processes supported by the new technologies and, more in general, the overall automation of the business processes.

One of the main issues obstructing the automation of business is the lack of an effective interoperability between different electronic commerce applications. Due to this lack of interoperability companies have to implement and to maintain many different systems (to address, for example, the needs of different customers).

A complete, effective integration between electronic commerce applications and companies' legacy systems is far from being a stable reality. A marketplaces assessment by Crimson Consulting (2000) over a sample of 700 showed that about ten of them were completely integrated with back-end systems and enterprise resource planning software inside the company.

This is a main question since only a real integration with the internal information systems can lead to a dramatic reduction of manual operations during information and processes exchange between firms. Increasing the automation of information flow would lead to an optimization of the order's cycle, to a reduction of the stocks' capacity and to a better business process reengineering.

Efforts on this topic are becoming more and more relevant. Several ICT firms are specializing in enterprise application integration (EAI) solutions. Notice that the lack of a standard format for data exchange is probably the first issue when approaching integration projects.

The investment needs for starting up electronic commerce solutions are still very onerous. Literature brings many reports estimating the costs of a completely operational electronic commerce platform. Since electronic commerce

implementations may strongly differ from each other, it is not simple to single out a precise and unambiguous indication about the evaluation of costs[9].

AMR Research, for example, states that the costs for creating a private electronic commerce platform (not a mere supply chain hub, rather an electronic commerce system for the "widened company") may range from 50 to 100 million dollars. IDAPTA underlines that the implementation of an industry-wide exchange (that is a platform designed for a marketplace of an entire industrial segment) may require an investment from 5 up to 125 million dollars depending on the complexity of the exchanged goods. The composition of costs shows a 15.2% rate for business process reengineering and supply chain management activities, a 13% rate due to supply-side solutions and a 10.9% rate for integrating existing back-end systems. The distribution of costs may change among different implementations. In particular, business process reengineering and integration becomes less expansive as the product complexity decreases.

In general this rating highlights that the cost of the software platform is fairly insignificant with respect to the sum of the other costs.

Due to technical issues and to the business process reorganization required by the adoption of the new technologies, implementation times of electronic commerce platforms are generally long. As the proposed costs rating shows, activating electronic commerce solutions causes a variety of organizational issues unlikely to be overcome in short times.

The proposed overview is certainly a not exhaustive list of technical issues for electronic commerce applications, but we believe that it holds the most interesting problems and the ones that should be more carefully investigated. The dimension for some of the proposed issues is such that they could not only be faced under a technical perspective, but they also have economical and organizational implications.

[9] Information about costs come from an article published by *eMarketer* may, 25 2001 (www.emarketer.com)

3.3 XML and electronic commerce

A complete analysis of the technical issues involved by electronic commerce applications should take into account the opportunity offered by the diffusion of XML.

Some of the introduced issues, in fact, could find in XML the basis for a future resolution.

Shortly, XML is a meta-language designed to structure and mark up documents independently from the application that use them, thus enabling reusability, flexibility and adaptability to complex applications.

Since no semantics is predefined in XML, it's up to the document's author to define the meaning for each mark up symbol, i.e. to define the specific language each time.

In the following we will focus on the potentialities and on the limits of XML in the development of electronic commerce applications. For technical aspects about XML refer to the World Wide Web Consortium (www.w3c.org/XML).

The key of XML success is probably its versatility:

- Formats are fairly easy to comprehend. It is possible to develop gradually more sophisticated application by enhancing and refining semantics step by step.
- Formats are self-describing. The developer is allowed to chose the proper term for each concept
- Documents are platform-independent. XML is decoupled from both operating systems and applications
- XML is web oriented

These features make XML the ideal choice for:

- Creating structured documents

- Separating the production from the use of data
- Managing very complex structured information (whereas the adoption of a relational database would heavily increase the complexity)
- Evaluating information within semi-structured documents

Translating this versatility into software features, XML is well suited, in electronic commerce applications for:

- *Electronic publishing*: since presentation is separated from content it's easy to edit a document and then publish it using different supports (web browser, mobile phones, TV devices)
- *Web searching*: metadata in XML can be used as pointers to the needed information
- *Data exchange*: XML allow information exchange and application integration reducing costs of development efforts, as with is needed is basically an agreement on mark up symbols.

The trend toward an intensive adoption of XML is shared by both software vendors and end users.

Software vendors are interested in XML since it helps in making easy the development of middleware applications reducing development costs and contributing to standardize the architecture of medium layers' applications. Data interoperability between heterogeneous applications is achieved by leveraging the platform independence. Moreover XML is supported by a variety of hardware devices – including wireless systems – and all major software development tools are already XML enabled.

End users take advantage from the simplicity, the extensibility and the openness of XML. XML allows to easily manage data output from companies' legacy systems, empowers data comparisons and aggregations and puts the basis for a universal format for data exchange in heterogeneous environments. XML dramatically reduces the cost of data access infrastructures as they converge to a standard format.

About the limits currently attributed to XML, the lack of a real standard seems to be the most relevant one. Open issues are represented by security and by the poor manageability offered by the existing development tools for the different components of XML language suite (XSL, XML Schema, XPath, etc.).

Even if XML is not a mature technology, a widespread diffusion seems to be inevitable. The starting point will probably be the integration of data within companies since the lack of a standard typically does not affect internal information.

In the next chapter, a section will specifically investigate the treatment of XML data sources.

3.4 Standardization problems

XML has been diffused so quickly that it is currently considered an indispensable tool for electronic commerce application developers. Nevertheless, XML is far from being considered a standard language for electronic commerce. The openness and the versatility of XML allow documents editor to define industry-wide dictionaries, specific languages, proprietary terms for mark up. A recent report from Accenture [52] highlighted the existence of about 500 XML dictionaries. The growth of these specific languages, causes serious standardization and interoperability issues.

An efficient and effective system for managing business transactions over the Web requires a clear communication and a mutual comprehension between trading partners. In particular, all the linguistic conventions active in the information exchanges, that are usually implicit in direct human interactions, must be clearly fixed when applied to electronic transactions. This settlement basically requires the definition of a standard language.

The problem of the identification of a standard language can generally be addressed through two opposite approaches. The first way is to agree on a unique standard language which should be able to support the widest range of situation. The main issue, in this case, is to fail in defining the standard language.

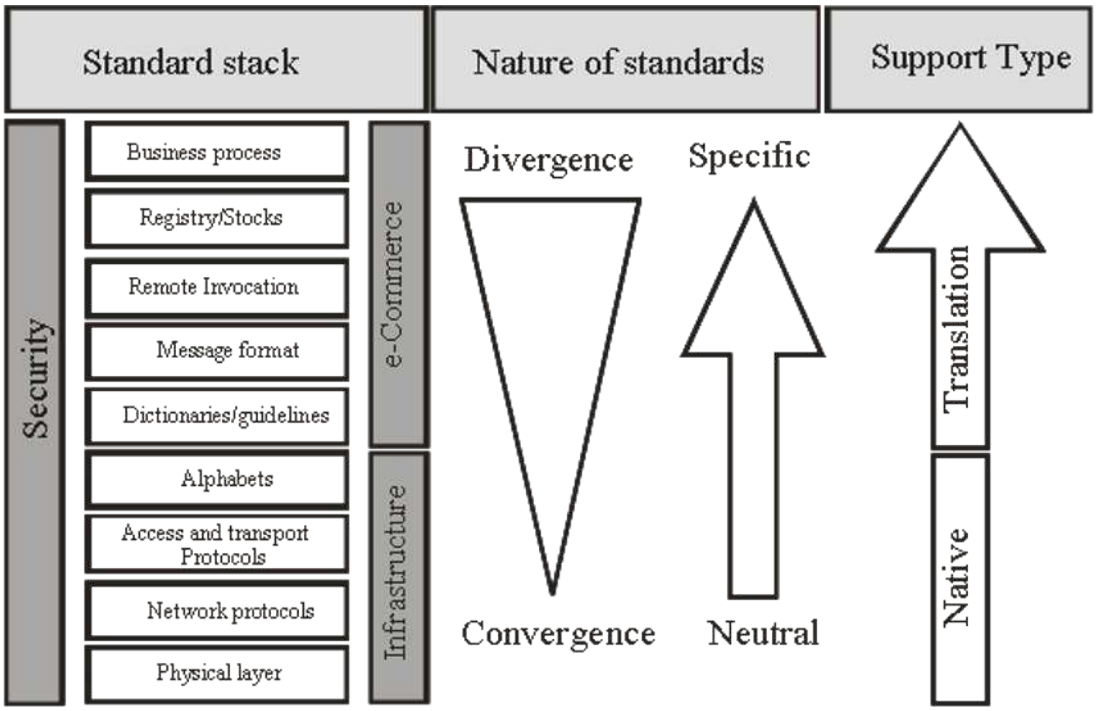

Figure 2 – Standardization levels in electronic commerce

The second way is the coexistence of multiple simpler standards partially overlapped to each other. In other words, the choice is between a unique, native standard and a collection of standards supported by translation features to make these different standards interact and communicate.

Even this choice is not unique since the standardization can be achieved at several levels. The lower levels concern the physical infrastructure (i.e. technical issues only), the higher level affect the electronic commerce applications (i.e. organizational problems). In particular:

- The physical layer affects the network and the manages bits over the network

- Network protocols states the rules for data transmission
- Access and Transport protocols regulate the transfer of big portions of data (i.e. files, tables etc)
- Alphabets specify the characters used by the adopted dictionary
- Dictionaries hold the meaning of each used term. For example RosettaNet, ebXML and others developed proprietary dictionaries.
- Message structure describes the rule to be followed in message editing
- Data guidelines indicates which elements can be represented and if different data sets can coexist or not.
- Remote invocation states the way for an application to invoke a method of another application running on another machine. This is a crucial point in distributed computing
- Stores and Registers manage big amount of data
- Finally the standardization may affect business processes and workflows, since it is possible, for example, to agree on the procedures required by the execution of certain processes.

Figure 2 shows that the standardization problem covers different scopes. The first is strictly technical and cover the programming language used by the applications. The second refers to the communication language, that is the meaning of each used word. The third scope deals with the definition of data. The fourth covers processes, i.e. the action taken by the actors of a business process. Finally the last scope involves the security, the integrity and the accessibility of the shared information.

The higher the level the more diverging are the standards since the influence of the particular business process becomes more relevant. The natural solution is to adopt native standard at the lower levels while choosing multiple different standards at the higher levels. The dominant trend, in fact, is to develop industry wide standards for the higher levels that rely on a common base (that is a shared standard at the lower levels) and provide translation features whereas a common lover level standard is not available.

Among different standards the SOAP protocol will be analysed in the next chapters

3.5 Interoperability and Integration

Interoperability between applications and integration of heterogeneous systems are key topics for electronic commerce.

Interoperability may be defined as the pool of technical features provided by connected systems that allows a certain end to end service to be delivered in a consistent and predictable way [65].

Interoperability is required for a electronic commerce application to succeed, since it allows multiple actors (that may differ from each other for technical infrastructure, business processes, type of processed information) to automatically interact in order to exchange data, run business transactions, manage shared process, etc. Interoperability affects multiple layers - as well as standards – ranging from physical networks and communication protocols to document formats. The highest interoperability level would allow different company to automatically take part to a single business process.

Interoperability heavily depends on the standardization level of the involved applications. Obviously, in presence of shared standards, interoperability between application become far easier. Therefore the adoption of open standards increases the overall level of interoperability.

It is important to underline that standards could be not sufficient to achieve a complete interoperability if the legacy systems or the enterprise resource planning software involved have been released long before the introduction of those standards.

Under these conditions as well as when a common standard can not be established, the integration of heterogeneous systems appear to be the only chance.

Integration has become more and more relevant inside the electronic commerce application developers community, since a new class of applications, called EAI (Enterprise Application Integration), has been recently introduced.

Integration is a very complex task and affects different sides of the technical infrastructure. A difference between "operational/applicative" Integration and "informative" integration can be introduced.

The former allows two heterogeneous systems to share data acting on the format of exchanged message, on the transport protocols and even on communication interfaces. The latter is required since every single system holds its own representation of the reality it works on. This representation heavily influences the communication with external systems. Therefore the informative integration is typically more complex than the operational, as it involves the domain of the application, the meanings and the interpretation of data and, in turn, linguistic and social topics.

A distinction between internal and external integration may be introduced. The internal integration aims at integrating heterogeneous application within the same company, while the external exploits the World Wide Web to connect and integrate heterogeneous applications among different companies.

The principles underlying both internal and external integration are basically the same. Facing external integration generally means dealing with bigger issues of informative integration, thus the complexity and the dimension of the problem in external integration is typically far greater than in internal integration.

In general, three ways to address conflicts between heterogeneous applications for electronic commerce can be individuated:

- Completely rewriting existing applications: it is a costly solutions and it could be compromised by the evolution of electronic commerce standards

- Abandoning the integration projects, thus renouncing to the possibilities offered by electronic commerce applications such as the business processes automation
- Adopting integration software which in relatively short times and with small changes to the existing systems allows to establish a connection between older and electronic commerce applications

Each of the proposes solutions offers advantages and disadvantages. The integration currently appear to be the most promising.

Integration is flexible, since it allows to add o remove elements at any time and scalable, as it can be performed in an incremental way.

The opportunity of connecting existing systems relies on the availability of software designed to automate the integration process as much as possible. Software of such a kind, usually referred as middleware, act as an interpreter between different IT systems.

A major issues is the data interpretation. The different data representations generally obstruct the integration process, since the logics underlying data representation in legacy systems is still oriented to memory saving and to the optimization of computation times, rather than to the possibility of accessing information from outside the applications. Thus, the integration process may require heavy manual operations [36] as:

- ERP applications usually manage a huge number of transactions which must be continuously notified to the connected systems
- Each new release of these applications not rarely causes relevant updates to the data definitions.

This issue gets more complicated in case of external integrations. A cooperation between multiple company, in fact, requires an agreement about processes across companies' boundaries and, in turn, a definition of a standard for

business processes. The higher the number of involved companies and of industrial sectors the more complex the integration process.

When the number of industrial sectors gets higher, integration can be achieved by introducing multiple standards (one for each sector) and connecting different sectors through translation hubs.

As previously stated, informative integration is the most complex task, since the information to be integrated could be heterogeneous from both a structural and semantic point of view.

Virtual catalogs represent a typical electronic commerce application that has to face with informative integration.

Virtual catalogs are on line catalogs holding information about products or services from multiple vendors..

The limit of this approach is that it still requires a manual revision from a domain expert in order to eliminate all the semantic ambiguities that rise when different data representation and interpretation models are put together. Further, several ICT companies in the United States are specialising in translation software designed to export data from a native format to open formats. Many content management companies propose solutions to automate existing information processing.

The MOMIS system (Mediator environment for Multiple Information Sources,) is a semiautomatic framework for integration of heterogeneous data sources. MOMIS as well as other available integration frameworks requires a human integration designer to revise the integration process and, in particular, to remove ambiguities between terms and concepts.

The MOMIS system, as well as virtual catalogs, will be analysed in the next chapter

3.6 Toward Modularity: component based architectures

The limited flexibility and the poor scalability of electronic commerce applications along with the incompleteness of features are still open issues.

Companies selecting electronic commerce solutions have to face two alternatives:

- Referring to multiple vendors to pick up the best from each one. This choice allows to generally fulfil all functional requirements but opens wide integration issues
- Adopting a solution from a single vendor which fulfil the minimum requirements.

The first choice, note as best in class or best of breed, involve the implementation of multiple tools. In general each tool address a single problem with encouraging results. Nevertheless the cost of integration of the tools may represent up to the 70% of the overall cost [1]. Considering the cost of the support services for each tool, this solution appear to be too much costly in the majority of cases.

The second solution involve the adoption of an application suite. Referring to a single vendor resets the need for integration between different applications. Nevertheless, it is not rare that the solutions proposed by a single vendor are not sufficient to fulfil all the specification in a effective way. Further, a single suite involve generally long times to activate all the features, heavy maintenance and update costs.

Finally integration problems rise whereas the company needs to set up collaborative commerce solutions with other firms. It is almost impossible that different companies in a supply chain – for example – run the same application suite.

Technical evolution seems to overcome these problems, by leveraging component based architectures.

These architectures require to break down different features into software packages developed through open standard languages. These packages can then be easily integrated into a modular framework.

Each component can be separately maintained and updated. Components are designed to directly interact with the logical layer of other component based applications. Components can be distributed in different servers over a network. The features of a single component can be easily extended to address specific business processes' needs.

Component bases architectures should contribute to a standardization process since they leverage a common lower level standard. Basically the technology involved are Component Object Model, JAVA RMI, CORBA. Recently XML Web Services seem to be the more promising technical choice, since they allow existing application to collaborate without relevant changes to their original structure. Further, XML Web Services overcome some security and performance issues common to the other technologies.

4 Intelligent Integration of Information

Developing intelligent tools for the integration of information extracted from multiple heterogeneous sources is a challenging issue to effectively exploit the numerous sources available online in global, Internetbased information systems. Main problems to be faced are related to the identification of semantically related information (that is, information related to the same real world concept in different sources), and to semantic heterogeneity. In fact, information sources available online in global information systems already exist and have been developed independently. Consequently, semantic heterogeneity can arise for the aspects related to terminology, structure, and context of the information, and has to be properly dealt with in order to effectively use the information available at the sources.

Integration and reconciliation of data coming from heterogeneous sources is a hot research topic in databases. Several contributions appeared in the recent literature, including methods, techniques and tools for integrating and querying heterogeneous databases. The integration of semistructured and unstructured data sources presents new problems and challenges: in this case, the heterogeneity concern not only the semantics of data, but also the degree by which the structure of data is explicitly represented in the sources. The significant growing of semi-structured data sources (document, texts, etc.) calls for the design of methods and techniques for this new type of data integration. Thus, the typical problems of integration should be addressed in the light of these new requirements.

The goal of information extraction and integration techniques is to construct synthesized, uniform descriptions (i.e. a global virtual view) of the information of multiple heterogeneous sources, to provide the user with a uniform query interface against the sources independent from the location and heterogeneity of the data at the sources. Any integration system that allows for a mechanisms of querying a global virtual view must contain a module for the reformulation of queries in terms of data stored in the sources and for the optimization of global querying

process. This problem is known in the literature as query rewriting and query answering using views, and has been studied very actively in the recent years. Moreover, to meet the requirements of global, Internetbased information systems, it is important to develop toolbased techniques, to make information extraction and integration activities semiautomatic and scalable as much as possible.

In this work, we focus on capturing and reasoning about semantic aspects of schema descriptions of heterogeneous information sources for supporting integration. Both semistructured and structured data sources are taken into account [21].

We experimented intelligent, toolsupported techniques to information extraction and integration with reference to the electronic commerce enivironment; an objectoriented language has been exploited, called ODL_{I^3}, derived from the standard ODMG, with the underlying Description Logics OLCD (Object Language with Complements allowing Descriptive cycles) [18],[22], derived from the KLONE family [66]. Information extraction has the goal of representing source schemas in ODL_{I^3}. In case of semistructured information sources, information extraction produces also object patterns, to be used as schema information for the source to generate the corresponding ODL_{I^3} description. ODL_{I^3} descriptions of the information sources are exploited to set a shared ontology for the sources, in form of a Common Thesaurus, by exploiting the OLCD Description Logics inference capabilities. Information integration has the goal of producing global, integrated ODL_{I^3} descriptions of the sources. It is performed in a semiautomatic way, by exploiting ODL_{I^3} descriptions of source schemas and by combining clustering techniques and the OLCD Description Logics on the Common Thesaurus. Furthermore, the WordNet lexical system [48] is used to automatically extract intersources terminological relationships. Mapping rules are defined at the global level to express the relationships holding between obtained ODL_{I^3} integrated descriptions and ODL_{I^3} sources descriptions, respectively.

The experimentation of the described techniques within an Electronic Commerce scenario heavily relied on the MOMIS (Mediator envirOnment for Multiple Information Sources) system, conceived as a joint collaboration between University of Milano and University of Modena and Reggio. In particular, the generation of virtual catalogs through the SI-Designer software component of the MOMIS has been investigated. The SI-Designer component will be widely described in the following. For a documentation of the experimental results refer to [10],[11],[12]

4.1 Virtual catalogs

As previously affirmed, catalogs are a key component in B2B EC environments. Catalogs can be organized as individual company catalogs or they can participate in a multicatalog framework. In the second case, from a buyer point of view, it is very crucial to have a single interface to search for products, that is a uniform view of data coming from different catalogs and a unique query language. On the other hand, from a supplier point of view, it is important to guarantee both the uniqueness of their catalogs and the participation in a multi-catalog framework. Virtual catalogs [8] are conceived as instruments that dynamically retrieve information from multiple catalogs and present data in a unified manner without physically storing the data. Trading partners, instead of having to interact with multiple heterogeneous catalogs, can interact in a uniform way with a single virtual catalog.

The problems that have to be faced in creating a virtual catalog are mainly due to structural heterogeneity (i.e. different platforms/data management systems), as well as to the lack of a common ontology, causing semantic differences between information sources. Moreover these semantic differences can cause different kinds of conflicts, ranging from simple contradictions in the use of terms (when different terms are used by different source to indicate the

same concept), to structural conflicts (when different primitives are used to represent the same information).

The MOMIS project aims to integrate data coming from structured and semi-structured data sources that is, it can be regarded as a framework able to support the creation of a virtual catalog by combining content from multiple suppliers.

4.2 Overview of the MOMIS System

Like other integration projects [6],[56], MOMIS follows a “semantic approach" to information integration based on the conceptual schema, or metadata, of the information sources, and on the I^3 architecture [40] (see Figure 3); for a detailed description of the MOMIS system see [16] available at http://www.dbgroup.unimo.it/Momis.

Mediator data integration systems usually follow this architecture: each data source provides a schema and a mediated (global) virtual schema of all the sources is obtained manually or semi-automatically, for a particular integration application. The mediated schema has a set of mapping descriptions that specify the semantic mapping between the mediated schema and the sources schema. The data integration system uses these mapping descriptions to reformulate a user query into queries over the source schemata.

Unlike previous mentioned approaches, mapping descriptions obtained as a result of the semi-automatic integration process of the MOMIS system [10],[17] include extensional intra/interschema knowledge which represents fundamental knowledge for a correct and complete schema integration [57]. For instance, if there are two classes Person in two different sources, then these classes may contain instances corresponding to the same real-world object or may refer to disjoint sets of real-world objects.

The system architecture is composed of functional elements that communicates using the CORBA [51] standard. Data are manages by exploiting a

common data model, ODM_{I^3}. This model is defined according to the ODL_{I^3} language, to describe source schemas for integration purposes. ODM_{I^3} and ODL_{I^3} have been defined as subset of the corresponding ones in ODMG, following the proposal for a standard mediator language developed by the I^3/POB working group [26]. ODL_{I^3} extends ODL standard by introducing new constructors (intensional/extensional relationships) to support the semantic integration process.

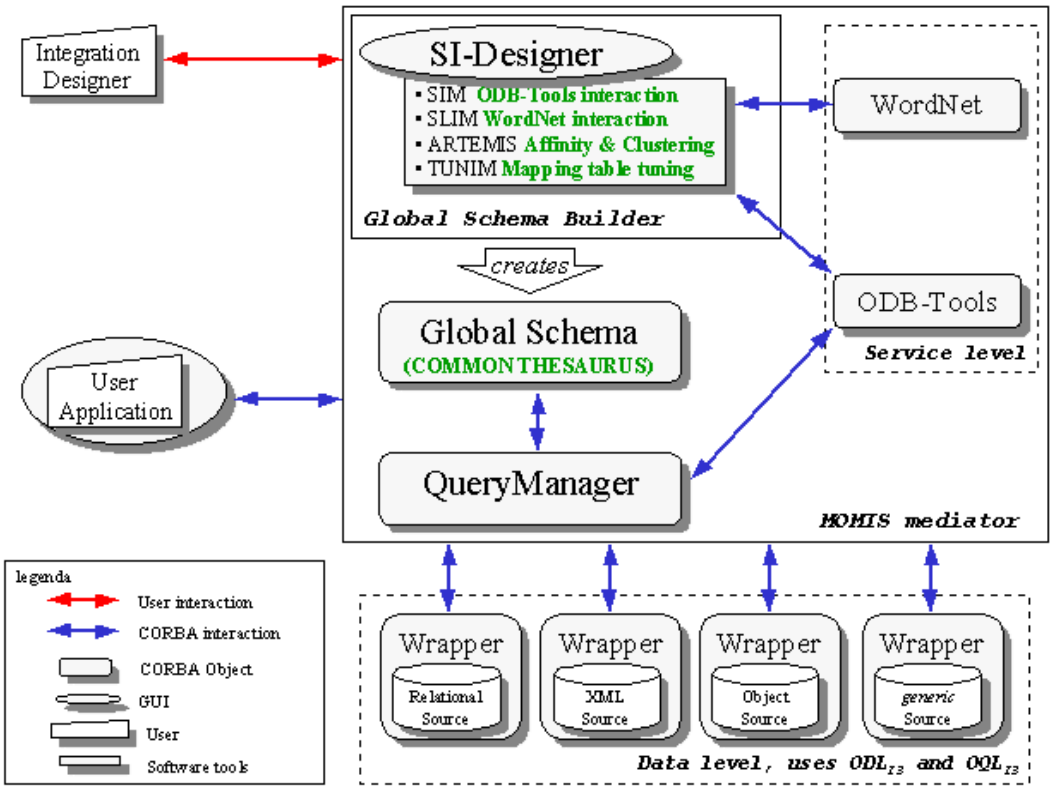

Figure 3 – The MOMIS Architecture

To interact with specific local source, Momis uses a Wrapper, that has to be placed over each source. There is a wrapper for each source has to be integrated. This wrapper translates metadata descriptions of the source into the common ODL_{I^3} representation, and translates (reformulates) a global query expressed in

the OQL_{I^3} [10] query language into a query expressed in the source language. The wrapper has to export query result data set to the global view in order to build a reconciled view over the heterogeneous sources involved.

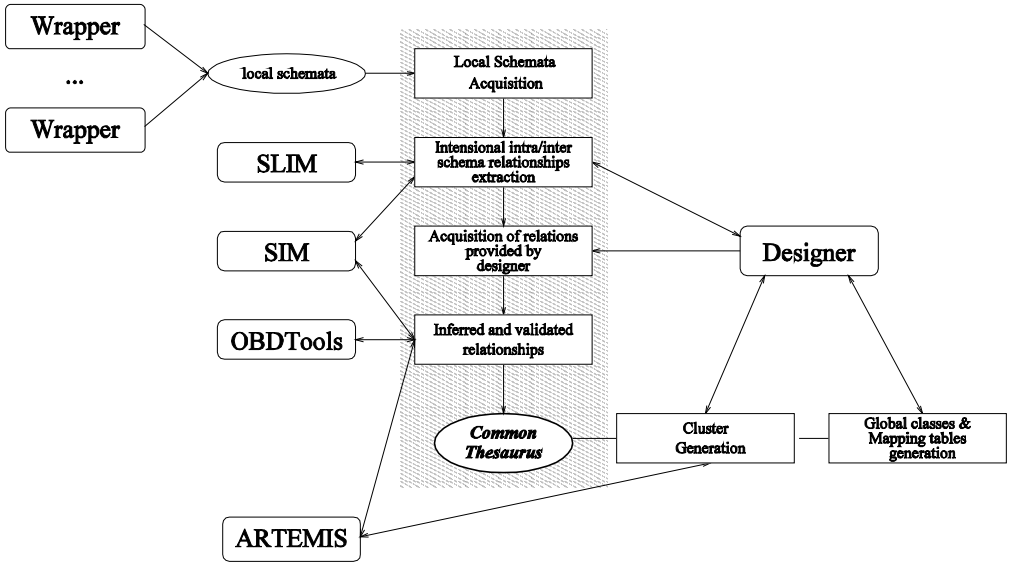

Figure 4 - The SI-Designer architecture

The core of the MOMIS system is the Mediator. It is composed of two modules: the Global Schema Builder (GSB) and the Query Manager (QM). The GSB module processes and integrates ODL_{I^3} descriptions received from wrappers to derive the mediated schema. The QM module performs query processing and optimization. The QM generates OQL_{I^3} queries to be sent to wrappers starting from each query posed by the user on the mediated schema. QM automatically generates the translation of the query into a corresponding set of sub-queries for the sources and synthesize a unified global answer for the user.

ODB-Tools Engine, is the OLCD *Description Logics* [6],[26] based tool performing schema validation and query optimization [10],[17],[40].

The integration designer is supported in the supervision of the integration process by the SI-Designer software component. Sources integration is based on

[10] OQL_{I^3} is a subset of OQL-ODMG

the individuation of an ontology shared by each source; the ontology is represented as a set of terminological relationships called *Common Thesaurus*.

As shown in Figure 3, *GSB* is composed by four modules:

- SIM (*Source Integrator Module*): extracts intra-schema relationships starting from a relational, object and semistructured source. Moreover this module performs the "semantic validation" of relationships and infers new relationships by exploiting ODB-Tools capabilities.
- SLIM (*Sources Lexical Integrator Module*) extracts inter-schema relationships between names and attributes of ODL_{I^3} classes of different sources, exploiting the WordNet lexical system.
- ATREMIS performs the affinity-based clustering on the ODL_{I^3} descriptions.
- TUNIM manages mapping between local and global attributes.

SI-Designer (Figure 4) provides the designer with a graphical interface to interact with SIM, SLIM, ARTEMIS and TUNIM modules showing the extracted relationships and supporting him in the *Common Thesaurus* construction.

4.3 The ARTEMIS tool environment

The GSB module of MOMIS relies on the ARTEMIS component. ARTEMIS is a tool environment for the integration of heterogeneous databases [67],[68], and for data sources over the Web [69],[70]. The ARTEMIS functionalities exploited in the framework of MOMIS are those for discovering semantically related elements in source schemas by performing the affinity-based clustering on the ODL_{I^3} descriptions. ARTEMIS provides MOMIS GSB with groups (i.e., clusters) of ODL_{I^3} classes. Clusters are then considered by GSB module for constructing elements in mediated schemas.

A high-level architecture of ARTEMIS is reported in Figure 5 and is realized as a set of modules that communicate among them according to the standard

CORBA. ARTEMIS affinity evaluation module performs schema matching by establishing semantic relationships between ODL_{I^3} classes of different sources. For this purpose, ARTEMIS evaluates different affinity coefficients, as described in what follows, taking into account both intensional and extensional interschema properties featuring the sources to be integrated. The Clustering module is responsible for grouping ODL_{I^3} classes based on the affinity evaluation results, putting together in the same cluster those that denote the same real world concept in the different sources.

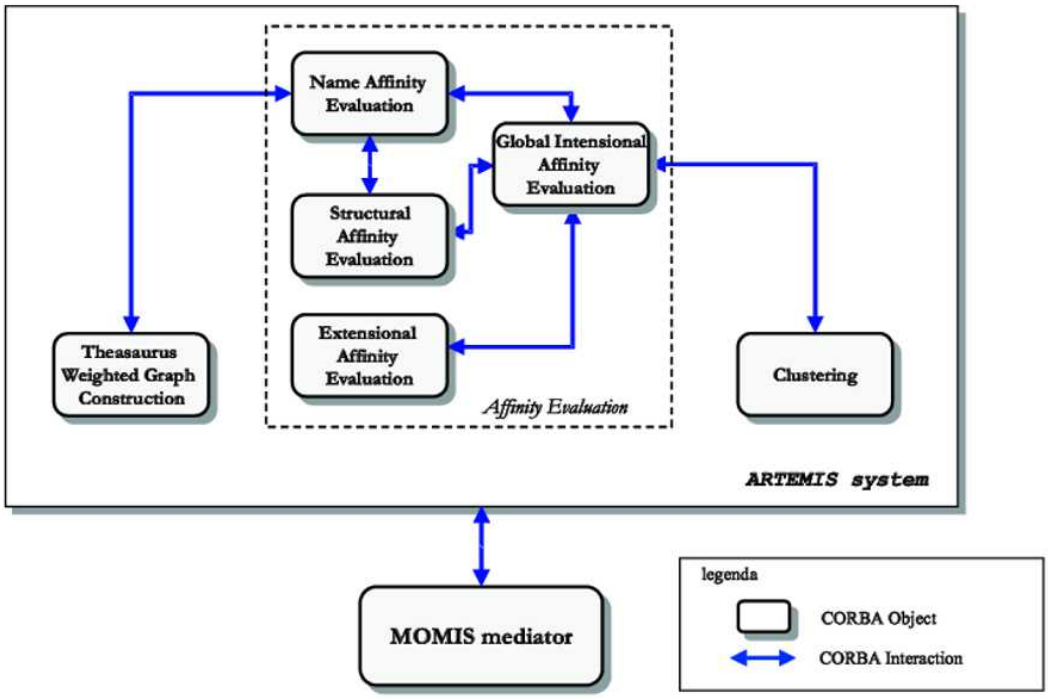

Figure 5 – The ARTEMIS architecture

4.4 Overview of the approach

One of the main tasks of mediators is to fuse information from heterogeneous information sources. This may involve, for example, removing redundancies and resolving inconsistencies in favour of the most reliable source. Mediators play the central role in information integration, and one of their most important task is to perform object fusion. This involves grouping together information (from the same or different sources) about the same real-world entity. In doing this fusion, the mediator may also “refine" the information by removing redundancies, resolving inconsistencies between sources in favour of the most reliable source, and so on.

A mediator may also have to avoid accessing to a particular source if, on the basis of extensional interschema knowledge, another involved source includes the information of such a source, or will provide an empty answer or if another source provides similar information at a lower cost (either financial or computational).

From a theoretical point of view, solving a user (mediated) query, i.e. giving a single unified answer w.r.t. multiple sources, implies to face two main problems: query reformulation/optimization and object fusion.

The techniques proposed in MOMIS rely on the availability of integration knowledge, whose semantics is given in terms of description logics. Integration knowledge is expressed in terms of local sources schemata, a virtual mediated schema and its mapping descriptions (i.e. semantic mappings w.r.t. the underlying sources both at the intensional and extensional level), extensional intra/interschema knowledge. Extensional knowledge is exploited to detect extensionally overlapping classes and to discover implicit join criteria among classes, thus allowing to achieve the optimized query reformulation and object fusion goals. In particular, starting from the method developed in [57], MOMIS exploits the “base extension" approach in order to face the reformulation/optimization problem of a mediated query and, on the basis of mapping descriptions, we develop a semi-automatic method to discover implicit join rules among classes in order to face the object fusion problem. The

techniques are under development in the MOMIS system but can be applied, in general, to data integration systems including extensional intra/interschema knowledge.

4.5 Wrapping of source schemas

In this section, we describe wrapping techniques developed in MOMIS. Understanding the wrapping system, specially in the case of XML sources, is important to highlight the benefits introduced by the XML Web Services-based architecture proposed in chapter 5.

The main goal of an information mediator system is to permit an integrated information accessibility from heterogeneous information sources, located in different sites and stored in different architectural platform. During this integration process many problems coming from structural and implementation heterogeneity (including for example differences in hardware platforms, DBMS, data languages) and from the semantic heterogeneity, when different names are employed to represent the same information (naming conflicts) or when different modeling constructs are used in different sources to represent the same kind of information.

To manage the implementation heterogeneity, a mediator system typically encapsules each sources by a *wrapper*, that logically converts the underlying data structure to a common information model. Therefore the wrapper architecture and interfaces are crucial, because wrappers are the focal point for managing the diversity of data sources.

In particular, the main tasks of a wrapper component are:

- during the integration process, the wrapper translates the schema of the source into the common data model of the mediator;
- during the query processing process, the wrapper converts queries posed over the common data model into requests suitable for the

source, and it converts data returned by the source into the common data model.

For conventional structured information sources (e.g. relational databases, object oriented databases), schema description is always available and can be directly translated into the selected common data model.

For example, for flat files and object oriented databases wrappers perform a syntactic translation, while for the relational databases the translation is based on *transformation rule-sets*, as described in [23] for relational to ODMG schema conversion. For semistructured information sources (e.g., Web data sources), schema description is generally not directly available in the sources. In fact, a basic characteristic of semistructured data is that they are “self-describing". This means that the information generally associated with the schema is specified directly within data [26].

In this section we describe the wrapper strategy and implementation within the MOMIS system.

Two kinds of wrapper are available: the former is a wrapper to manage relational databases, founded on JDBC technology [32] to connect and to query the sources. The latter is a wrapper to manage semistructured data files.

In this way we are also able to integrate XML sources. We are trying to improve the capabilities of our wrapper in order to manage HTML files, and other XML based languages.

4.5.1 Wrapping relational sources

Within the MOMIS system, the wrapper that manages relational databases is implemented with the JDBC technology and lets the mediator system access virtually any relational data source by using the interfaces provided by the Java programming language. In fact, the JDBC technology permits the use of the standard interface SQL to query a wide range of relational databases. In this way, the schema structure of the source is available from a standard language.

The translation into ODM_{I^3}, on the basis of the ODM_{I^3} syntax (see Appendix A) and of the schema definition is performed by the wrapper as follows. Given a relation of a relational source, translation involves the following steps:

i. an ODM_{I^3} class name corresponds to the relation name
ii. for each relation attribute an attribute is defined in the corresponding ODM_{I^3} class.

Furthermore, attribute domains are extracted. Our approach aims to obtain source schemas by adopting standard technologies. Nevertheless each wrapper has to be customized in relation of the specific DBMS the mediator system needs to be connected. In particular, different DBMS platforms do not store information about primary and foreign key in a standard way and using standard catalogs. For this reason, a customized wrapper for each database platform we want to interact to is available.

4.5.2 Wrapping semistructured sources

Research on semistructured data aims at extending database management techniques to data with irregular, unknown or often changing structure. To integrate this sources, two issues are to be faced: the former is an automatic way to derive and explicitly represent the schema of the source. The latter is to communicate to the mediator when and how a source was modified, and to develop methodologies to modify automatically the global schema. Moreover, in order to integrate semistructured information sources, we have to take into account that several data models represent this kind of source have been proposed in literature. In particular the OEM model (Object Exchange Model) may be thought as the de facto model to represent semistructured data. It was originally introduced for the TSIMMIS data integration project [37]. In [] they are proposed methodologies to extract structures from semistructured data files. But, in order to query this kind of sources, you have to consider that no standard languages to query semistructured information sources were developed. The work to realize a

language to perform queries over XML files is developing at the Word Wide Web Consortium (now only working drafts have been issued).

According to the models proposed in literature for semistructured information sources [7, 42], MOMIS represents semistructured information sources as rooted, labelled graph with the semistructured data (e.g., an image or free-form text) as nodes and labels on edges.

A semistructured object (object, for short) can be viewed as a triple of the form $\langle id; label; value \rangle$ where *id* is the object identifier, *label* is a string describing what the object represents, and *value* is the value, that can be atomic or complex. The atomic value can be integer, real, string, image, while the complex value is a set of semistructured objects, that is, a set of pairs (*id, label*).

A complex object can be thought as the parent of all the objects that form its value (children objects). A given object can have one or more parents. We denote the fact that an object *so′* is a child object of another object *so* by $so \rightarrow so'$ and use notation *label*(*so*) to denote the label of *so*. In semistructured data models, labels are descriptive as much as possible. Generally, the same label is assigned to all objects describing the same concept in a given source.

To represent the schema of a semistructured source *S*, we introduce the notion of *object pattern*. All objects *so* of *S* are partitioned into disjoint sets, denoted set_l, such that all objects belonging to the same set have the same label *l*. An object pattern is then extracted from each set to represent all the objects in the set. Formally, an object pattern is defined as follows:

[**Object pattern**] Let set_l be a set of objects in a semistructured source S having the same label *l*. The *object pattern* of set_l is a pair of the form $\langle l, A \rangle$, where *l* is the label of the objects belonging to set_l, and $A = \bigcup label(so')$ such that there exists at least one object $so \in set_l$ with $so \rightarrow so'$.

From this definition, an object pattern is representative of all different objects that describe the same concept in a given semistructured source. In particular, *l* denotes the concept and set *A* the properties (or attributes) characterizing the concept in the source. Since semistructured objects can be heterogeneous, labels

in A can be defined only for some of the objects in set_l, but not for all. We call such kind of labels “optional" and denote them with symbol “?".

An object pattern description follows an *open world semantics* typical of the Description Logics approach [66]. Objects conforming to a pattern share a common minimal structure represented by non optional properties, but can have further additional (i.e., optional) properties.

In this way, objects in a semistructured data source can evolve and add new properties, but they will be retrieved as valid instances of the corresponding object pattern when processing a query.

According to this data model a wrapper to manage XML files has been developed. This wrapper aims to map a data model of XML file into the corresponding OEM model. This wrapper could be thought as the core of further extensions that aim to manage XML based files as RDF files or XHTML files.

4.5.3 A wrapper for XML files

As previously affirmed the XML language is a W3C recommendation and arises as a language to describe information sources by using an universal format. One of the main goals of this standard is to exchange les across the Internet. Contrary to HTML, that aims to exchange les in a "human-readable" way, XML language aims to exchange les across the Internet in a "machine-readable" way.

In comparison with HTML, the fashion data-oriented of XML it is synthesizable as follows [58]:

- The user may define personal tag names at will.
- The structure of the XML file may include tags nested to any level.
- A XML file may include a description of its structure for use by applications that need to perform structural validation. This definition may be DTD (Document Type Definition) as well as XML Schema.

In this way a XML file may be thought as self-describing like a semistructured data source.

The main analogies may be summarized as follows:

- OEM attribute → XML tag

- OEM object → XML element
- Atomic value of an OEM attribute (string, real, image, ...) → PCDATA value

By using this mapping it is possible to use the XML data model to describe semistructured information sources. This mapping implies some critical aspects that have to be analyzed [58].

the order: strictly, a XML le may be thought as representing an ordered data model. An element in a DTD row is described as containing other elements with a prefixed order. The concept of order is normally not applied in semistructured data file. To foresee an element order management could be useful to obtain specific information during the query process. Actually our wrapper describes data in an unordered way.

the attributes: The XML tags may contain attributes (called ATTLIST) which are semantically w.r.t. OEM attributes. For example the following extract of XML file uses an XML attribute:

```
<Student country="Italy">
<name>Philip</name>
</Student>
```

Each tag may contain several couples attribute-value. Our wrapper translates this structure into an OEM compliant structure without lack of semantics. This translation generates a new tag for each couple attribute-value as follows:

```
<Student>
<name>Philip</name>
<country>Italy</country>
</Student>
```

the references/links: Our wrapper does not support link inter files. XLINK, the W3C recommendation which allows elements to be inserted into XML

documents in order to create and describe links between resources, is not supported in our software. Nevertheless ID-IDREF(S) couples, defined in XML language to express references intra-schema, are supported. In order to manage this kind of information our wrappers have to interact with the user. In fact, according to the XML syntax, we do not have to explicitly express which tag an IDREF attribute refers to. To avoid its loss during the translation process, the user is asked for some further information by a graphical interface.

4.6 Running example

In order to illustrate how the MOMIS approach works, we will use the following example of integration in the Car manufacturing catalogs, involving two different data-sources that collect information about vehicles[11]. The first data-source is the Volkswagen database (VW), a relational database storing information from that manufacturer. The second data-source is the FIAT catalog, containing semi-structured XML information about cars from the Italian car factory Both database schemata have been built by analyzing the web site of the corresponding company.

```
Vehicle(name, length, width, height)
Motor(cod_m, type, compression_ratio, KW, lubrication, emission)
Fuel_Consumption(name, cod m, drive_trains, city_km_ l,
highway_km_ l)
Model(name, cod_m, tires, steering, price)
```

Figure 6 - The VW database

The VW database is a structured data source based on relations (such as “Vehicle”) and attributes (“name”); it could be implemented through any of the relational database management systems (RDBMS) available on the market. The

[11] In order to clearly describe the integration process we choose a B2C example. A B2B example, in fact, typically involves a more specific knowledge about terms and concepts belonging to the domain even if it doesn’t change the system’s approach to the integration issue.

FIAT catalog is a XML schema: it describe the structure of data stored in a single XML file (i.e. a text file composed according to the XML syntax).

```
.<!ELEMENT fiat(car*)>
<!ELEMENT car(name,engine,dimensions,tires,performance,price)>
<!ELEMENT engine(name,cylinders?,layout?,capacity_cc?,
compression_ratio power_kw,fuel_system)>
<!ELEMENT dimensions(length,width,heigth,luggage_capacity)>
<!ELEMENT performance (urban_consumption,combined_consumption,speed)>
<!ELEMENT name (#pcdata)>
```

Figure 7 - The FIAT catalog

4.7 The ODL_I^3 language: intensional and extensional knowledge representation

For a semantically rich representation of source schemas and object patterns associated with information sources to be integrated we introduced an object-oriented language, called ODL_{I^3} very close to the *ODL* language [28],[40]. In the following we report the main features of ODL_{I^3} related to intensional relationships and extensional relationships.

Intensional relationships. These are *terminological relationships* expressing intra and interschema knowledge for the source schemas. Intensional relationships are binary relationships defined between classes and attributes, and are specified by considering class/attribute names, called terms. The following kinds of relationships can be specified in ODL_{I^3}

- SYN (Synonym-of), defined between two terms t_i and t_j , with $t_i \neq t_j$, that are considered synonyms in every considered source (i.e., t_i and t_j can be indifferently used in every source to denote a certain concept).
- BT (Broader Terms), or hypernymy, defined between two terms t_i and t_j such as t_i has a broader, more general meaning than t_j. BT relationship is not symmetric. The opposite of BT is NT (Narrower Terms), or hyponymy.

- RT (Related Terms), or positive association, defined between two terms t_i and t_j that are generally used together in the same context in the considered sources.

An intensional relationships has no implications for the extension/compatibility of the structure (domain) of the two involved classes (attributes). Consequently, our notion of intensional relationships is different from the one proposed by Catarci and Lenzerini [27], where an intensional relationships has some extensional import.

Extensional relationships. Intensional relationships SYN, BT and NT between two classes C_1 and C_2 may be "strengthened" by establishing that they are also extensional relationships [27]. This operation is performed both manually, with a graphical interface the designer can introduce new relationships, and automatically, by using structural relationships previously defined. The system checks if the extensional relationships introduced are each other consistent. The following extensional relationships can be defined in ODL_{I^3} :

C_1 SYN_{ext} C_2 : this means that the instances of C_1 are the same of C_2 .

C_1 BT_{ext} C_2 : this means that the instances of C_1 are a superset of the instances of C_2.

C_1 NT_{ext} C_2 : this means that the instances of C_1 are a subset of the instances of C_2 .

C_1 $DISJ_{ext}$ C_2 : this means that the instances of C_1 are disjoint from the instances of C_2.

In contrast with [57] we do not introduce an overlap relationship as we assume a default overlap relationships among two classes if no extensional relationship is specified.

Moreover, extensional relationships "constraint" the structure of the two classes C 1 and C 2 .

If an extensional relationship C_1 NT_{ext} C_2 is issued, we have that:

- strict inheritance between C_1 and C_2 is enforced for the common attributes;
- both C_1 and C_2 may have further attributes as we adopt usual description logics semantics (i.e. open world semantics).

Extensional relationships can be partially automatically extracted and partially explicitly declared by the integration designer.

Extensional rules can be expressed in ODL_{I^3} by using integrity constraint rules as follows:

An extensional relationship C_1 SYN_{ext} C_2 is equivalent to two ISA relationship C_1 ISA C_2 and C_2 ISA C_1 plus an intensional relationship C_1 SYN C_2.

An extensional relationship C_1 NT_{ext} C_2 is equivalent to an ISA relationship C_1 ISA C_2 plus an intensional relationship C_1 NT C_2 .

An extensional relationship C_1 BT_{ext} C_2 is equivalent to an ISA relationship C_2 ISA C_1 plus an intensional relationship C_1 BT C_2.

The ISA relationship can be expressed in the ODL_{I^3} language by the following rule:

```
extrule <RuleName> forall x in C1
  then X in C2
```

Furthermore, to describe the disjointness relationship we use the following relationship:

```
extrule <RuleName> forall x in (C1 AND C2 )
then X in BOTTOM
```

4.8 Generation of a Common Thesaurus

The *Common Thesaurus* is a set of terminological intensional and extensional relationships, describing inter-schema knowledge about classes and attributes of sources schemas; it provides a reference on which to base the identification of

classes candidate to integration and subsequent derivation of their global representation. In the Common Thesaurus, we express inter-schema knowledge in form of terminological relationships (SYN, BT, NT, and RT) and extensional relationships (SYN_{ext}, BT_{ext}, NT_{ext}) between classes and/or attribute names.

The Common Thesaurus is constructed through an incremental process during which relationships are added in the following order:

schema-derived relationships: Terminological and extensional relationships holding at intra-schema level. These relationships are extracted analyzing each ODL_{I^3} schema separately. In particular, intraschema RT relationships are extracted from the specification of foreign keys in relational source schemas. When a foreign key is also a primary key both in the original and in the referenced relation, a BT/ NT relationship is extracted. At this stage SI-Designer exploits the SIM module for extracting the intra-schema relationships For example,

```
[VW.Model RT VW.vehicle]
[VW.Model RT VW.motor]
[fiat.engine RT fiat.car ]
```

lexical-derived relationships: Terminological relationships holding at inter-schema level are extracted analyzing different sources ODL_{I^3} schemas together through the interaction between SI-Designer and the SLIM module.

In the next section lexical relationships will be examined as their extraction is one of the most interesting topic with reference to the previously presented integration issues in Electronic Commerce application.

The most significant lexical relationships are the following:

```
[fiat.car SYN VW.vehicle]
[fiat.engine.compression ratio SYN VW.motor.compression ratio]
[fiat.dimension BT VW.vehicle.width]
```

designer-supplied relationships: Terminological and extensional relationships supplied directly by the designer, to capture specific domain knowledge about the

source schemas. Consider the VW source, in which the model entity can be considered as a specialization of the vehicle entity. This relationship can not be automatically extracted using both the lexical and the structural approaches, hence we supplied the following relationship:

```
[VW.Model NT fiat.car]
```

This is a crucial operation, because the new relationships are forced to belong to the Common Thesaurus and thus used to generate the global integrated schema. This means that, if a nonsense or wrong relationship is inserted, the subsequent integration process can produce a wrong global schema. SI-Designer help the designer in detecting wrong relationships by performing a *Relationships validation* step with ODB-Tools[12].

inferred relationships: Terminological and extensional new relationships, holding at intra-schema level, inferred by exploiting inference capabilities of ODB-Tools. In the examined domain ODB-Tools infers the following relationships:

```
[VW.Model RT fiat.dimensions]
[VW.Model NT fiat.engine]
```
[VW.motor NT fiat.car]

All these relationships are added to the Common Thesaurus and thus considered in the subsequent phase of construction of Global Schema. For a more detailed description of the above described process see [16].

Terminological relationships defined in each step hold at the intensional level by definition. Furthermore, in each of the above step the designer may "strengthen" a terminological relationships SYN, BT and NT between two classes C_1 and C_2 by establishing that they hold also at the extensional level, thus defining also an extensional relationship. The specification of an extensional relationship, on one hand, implies the insertion of a corresponding intensional relationship in

[12] ODB Tools is a description logics [1,7] based tool performing both relationships validation and queries optimization.

the Common Thesaurus and, on the other hand, enable subsumption computation (i.e. inferred relationships) and consistency checking between two classes the C_1 and C_2.

4.9 Lexical-derived inter-schema relationships

The extraction of these relationships is based upon the lexical relations holding between classes and attributes names, deriving from the mining of used words. This is a kind of knowledge which is not based on the rules of a data definition language but derives from the name assigned by the designer. It is a designer's task to assign descriptive/meaningful names or, at least, correctly interpretable names. An interpretation uncertainty is therefore inherent to the language ambiguity; Bates [9] writes *"the probability of two persons using the same term in describing the same thing is less than 20%"*.

Anyway, knowledge associated with schema names is an opportunity that must be exploited to extract relationships. As it is almost impossible to carry out this task manually when the number and dimensions of schema grows, it was decided to experiment the use of the WordNet [48] lexical system to extract and propose to the designer intensional inter-schema relationships.

4.9.1 The WordNet database

WordNet is a lexical database which was developed by the Princeton University [34],[48] Cognitive science Laboratory.

WordNet is inspired by current psycholinguistic human lexical memory connected theories and it is regarded as the most important researcher's available resource in the fields of computational linguistics, textual analysis and other related areas. The lexical WordNet database, in the current 1.6 version has 64089 lemma which are organized in 99757 synonym sets (*synset*).

The starting point of lexical semantics is the assertion of the existence of a conventional association between the words form (i.e. the way in which they are

pronounced or written) and the concept/meaning they express; such association is of the many-to-many kind, giving rise to the following properties:

Synonymy: property of a concept/meaning which can be expressed with two or more words. A synonyms group is named *synset*. Note that one and only *synset* exists for each concept/meaning. Later a *synset* will be indicated with *S*, while ***S*** will indicate the *synset* set.

Polysemy: property of a single word having two or more meanings.

The correspondence between the words form and their meaning is synthesized in the so called *Lexical Matrix* ***M***, in which the words meaning are reported in rows (hence each row represents a *synset*) and columns represent the words form (form/base lemma).

Each matrix element is a(*entry*), *e* = (*f, m*) definition, where *f* is the *base form* and *m* (*meaning*) is the meaning counter; for example (`address`, `2`) refers to the address where a person or an organization can be found; while (`address`, `1`) refers to a computer address in the informatics sphere. From here on the base form and the meaning of an element *e* = (*f, m*) will be respectively indicated with *e.f* and *e.m*. An element of the matrix may be *null* or *indefinite*.

As only one ***M*** row is associated to a *synset*, from here on we will use $S \in \boldsymbol{S}$ as a ***M*** row indicator. In other words the non null elements of the ***M***[*s*] row, represent each and every *s* element. In the same way, as only one ***M*** column is associated to a base form, from here on we will use the base forms as ***M*** columns index.

4.9.2 Semantic relationships between schema terms

With the concept of *term* we associate a definition to each class or attribute name. A *term* is formed by the *t = (n, e)* couple, where *n* indicates a class or attribute name, and *e* indicates a definition.

A class or attribute name *n* are qualified as follows a class name is qualified by the name of the source schema to whom the class belongs (`source_name.class name`), an attribute name is moreover qualified with the name of the class to whom it belongs (`source_name.class_name.attribute name`).

The classes and attributes names set is indicated by **N**; the set of words in **N** is indicated by ***I*.** The relation between *synset* defined in WordNet are the starting point to define semantic relations between words. Various relations are obtainable with the WordNet database; some of them are between single words others are between *synset*. In this context we will use the following relations between *synset*:

Synonymy, **Hypernymy**, **Hyponymy**, **Olonymy**, **Meronymy**, **Correlation**[13]

As hyponymy and meronymy are inverse relations to hypernymy and olonymy, respectively, the set of relations between *synset* is the following:

$$W = \{\mathbf{S}_{\text{ynonymy}}, \mathbf{H}_{\text{ypernymy}}, \mathbf{O}_{\text{lonomy}}, \mathbf{C}_{\text{orrelation}}\}$$

Given the *synset* $\boldsymbol{S}$ set and the **W** relations set, The function $\phi : S \times W \rightarrow 2^s$ is inserted giving for each *synset s* the set of *synset* associated through the $r \in W$ relation:

$$\phi(s,r) = \{s' \mid s' \in \boldsymbol{S},\ r \in W, \langle s' r s \rangle\}$$

Given a *synset* $\boldsymbol{S}$ set and a $\boldsymbol{I}$ set of words, the function $H : S \rightarrow 2^{\boldsymbol{I}}$ defined associating, on the basis of the lexical matrix, a set of words to a given *synset* :

$$H(s) = \{t = (n,e) \mid n \in \boldsymbol{N},\ \boldsymbol{M}[\mathrm{s}][t.e.f] = t.e\}$$

We can hence obtain the relations between the words using the relations existing between the *synset* that contain those words. Given a set of words **I** the set of relations between words $R, R \subseteq \boldsymbol{I} \times W \times \boldsymbol{I}$ is defined as follows:

$$R = \{\langle t_i r t_j \rangle \mid r \in W, t_i, t_j \in \boldsymbol{I}, \exists s : t_i \in H(s), t_j \in \phi(s,r), t_i \neq t_j\}$$

The relations deriving from are proposed as semantic relations to be inserted in the *Common Thesaurus* according to the following correspondence:

Synonymy : corresponds to a SYN relation.

Hypernymy : corresponds to a BT relation.

[13] Correlation is a relation linking two synsets to the same hypernym, i.e. the same father

Olonymy : corresponds to a RT relation.

Correlation : corresponds to a RT relation.

On the basis of these considerations, an algorithm has been developed which having as input the terms related to the schemata to be integrated, outputs the detected semantic relations:

$$\text{Input } \mathbf{I} = \{ t_i \mid t_i, n \in \mathbf{N}\}$$

$$\text{Output } R = \{\langle t_i r t_j \rangle, r \in \{\text{SYN}, \text{BT}, \text{RT}\}\}$$

The algorithm is presented at [49]. We will now consider the use of the SLIM (*Sources Lexical Integrator Module*) tool, that extracts inter-schema relationships between names and attributes of ODL_{I^3} classes of different sources, exploiting the WordNet lexical system.

Starting from the schema to be integrated, the designer must fix the set. Given a name the associated words must be

chosen. This choice involves two steps:

1. **Base form choice.** The designer is supported in such a choice by the system which gives him the base form (word form) using the WordNet morphologic processor. If a base form is not found, or there is an ambiguity, or it is not satisfactory, the designer can directly introduce it at any time.
2. **Meaning choice.** The designer can relate a name to one, more than one, or no meaning. The choice of not relating a name to any meaning can be made for various reasons: (**a**) the concept is too complex and it can not be expressed with one word; (**b**) it belongs to the *tops*, i.e. to the generic concepts, therefore it would be related to the whole; (**c**) it is a substitute key, therefore it doesn't add any knowledge; (**d**) it is used as *foreign key*, therefore this relation has already been used during the extraction of relations from the schema structure.

In such a choice, the designer is supported by the tool which gives him, for a given name, its hypernymy hierarchy. Figure 9 shows the hypernymy hierarchy of

`engine`; in this case, meanings 1 and 2 have to be both selected as they are similar.

The designer selects one or more meanings from those found in WordNet starting from the base form chosen at step 1. Therefore, all the words that are related to the same name, share the same base form. For example, all the 15 meanings that WordNet relates to the `[name]` base form are obtained (see Figure 8). Selecting them all, i.e. considering 15 words for the `[car.name]` attribute, we could obtain "wrong" results, which are not suitable within the examined context. Some of them are shown in the following:

[fiat.car.name SYN vw.fuel_consumption.name]

[fiat.car.name SYN fiat.engine.name]

The annotation activity, i.e. the task of associating each term to its correct meaning, could be significantly time consuming for the integration designer since he must process all the terms of all the local schemata assigning a word form and a correct meaning to each of them.

Note that some of this relationships can look quite strange but they are true in some particular context. The problem, hence, is now resolving the meaning ambiguity so that a context-suitable couple (base form, meaning counter) can be supplied to WordNet for each concept of a source. To help the designer in the choice of the "right" meaning, for each couple (base form,meaning counter), a syntactic category (names - **N**, verbs - **V**, adjectives -**Aj**, Adverbs - **Av**) is indicated (see Figure 8).

This semi-automatic approach reduces the complexity of the designer task, in fact, a "*difficult*" problem (i.e. is finding the relations between all words), is divided in many "*easy*" ones, choosing each terms' meaning from a list. In practice this is an 80/20 problem, that is 80% of the words is worked out in the 20% of the time, just the time for reading the definitions, while the remaining 20% occupies the 80% of the time, because the choice is between very similar meanings. To speed up the 80% part a "cache" of the already selected couple

(base form, meaning counter) is used (see Figure 8: the symbol denotes the meaning already chosen by the designer for the `name` concept).

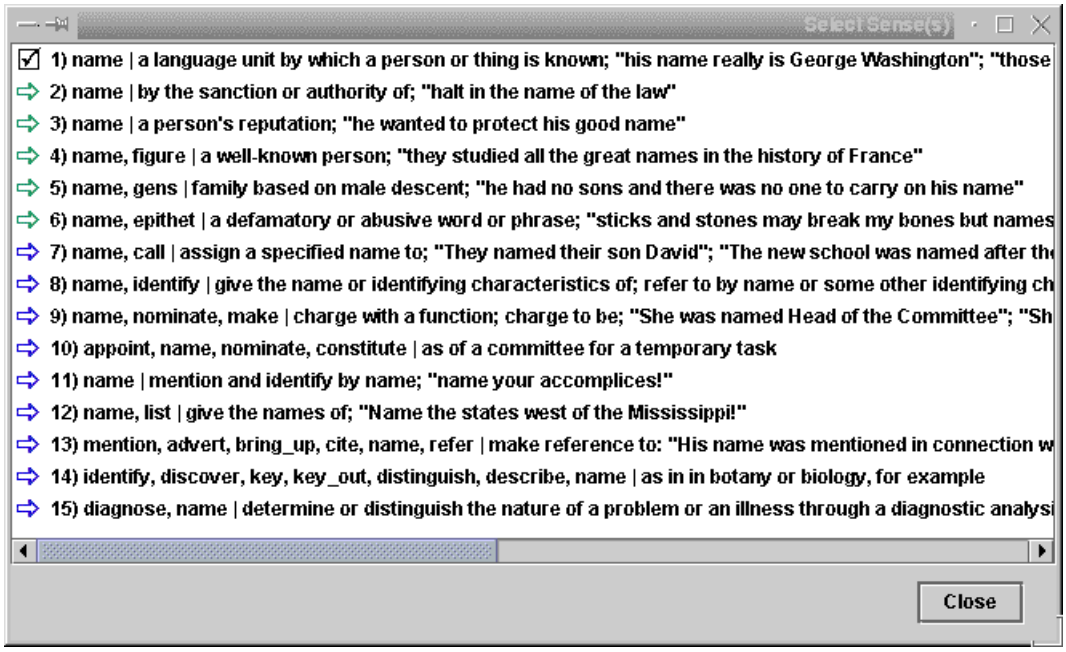

Figure 8 - Meanings of the term "name"

Furthermore, the system can show the generalization hierarchy of the meanings in order to help the designer in the most difficult choices. For example, (see Figure 9) in the case of `engine`: we see that " engine#2" inherits only from "causal_agent..." and "engine#3" from "wheeled_vehicle" and "concern railway contest," whereas "engine#1" inherits also from "machine" and "motor#1." Thus we select "engine#1".

At the end of this phase, the system shows the relationships derived by using WordNet (see Figure 10). The designer may delete any of the showed relationships and add new ones.

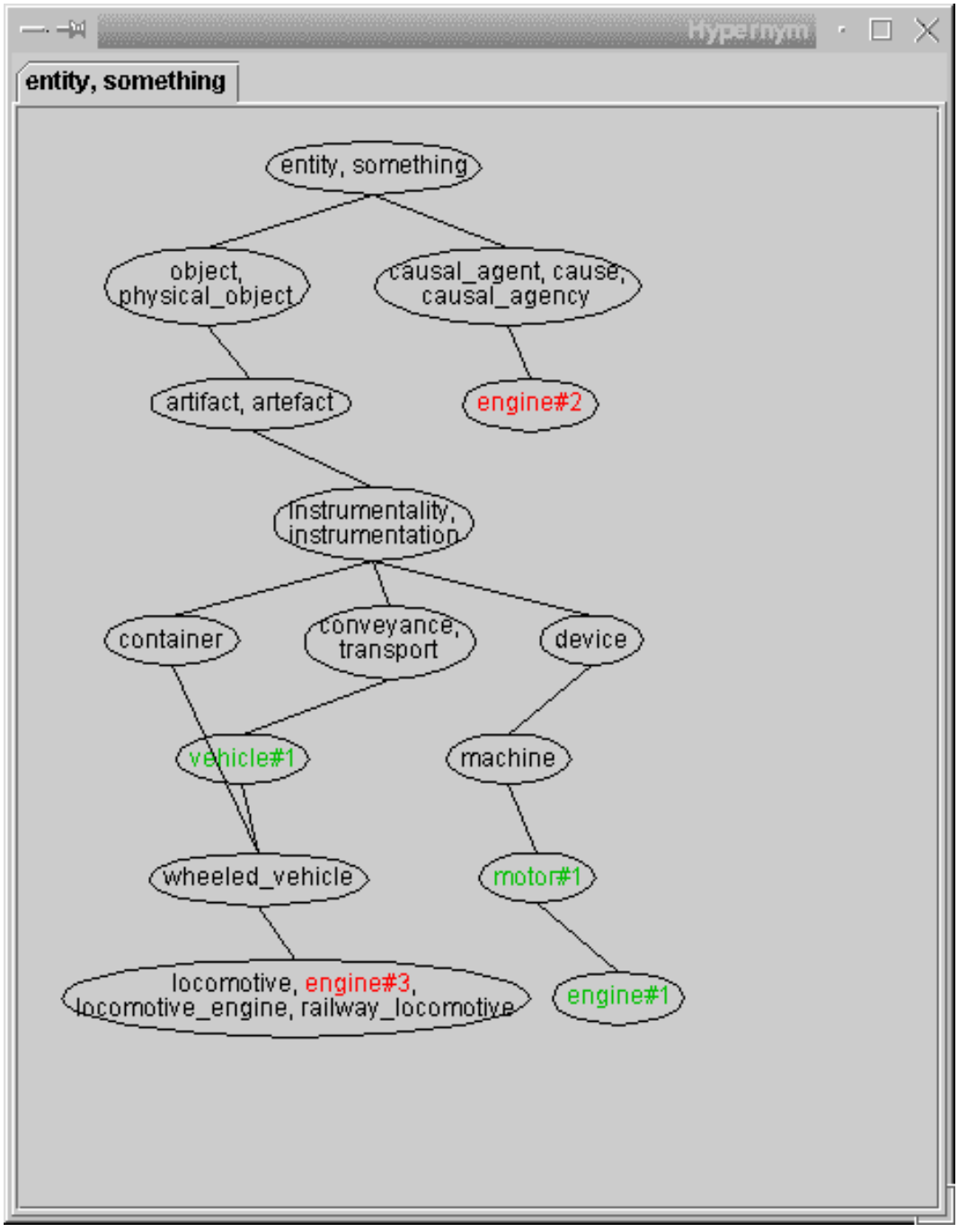

Figure 9 - Hypernymy hierarchy of `engine`

4.10 Clusters generation

To identify all the ODL_{I^3} classes having affinity in the considered source schemas, ARTEMIS uses a hierarchical clustering technique, which classifies classes into groups at different levels of affinity, forming a tree.

- the leaves represent all the local classes: adjacent leaves represent classes with high affinity, while leaves far apart from each other represent classes with low affinity;
- each node represents a clustering level and is associated to the affinity coefficient between the sub-trees (clusters) it joins.

Within SI-Designer, the designer, at any iteration, can insert a threshold value which is used by ARTEMIS to build clusters: each cluster is made by all the classes belonging to a sub-tree having at the root node a coefficient which is higher than the threshold value. Figure 11 shows the global classes that ARTEMIS identified

4.11 Generation of the global attributes and mapping tables

For each cluster, SI-Designer creates a set of global attributes and, for each of them, it determines the correspondence with the *local attributes* (i.e. those of the classes belonging to the cluster to which the global class corresponds). In some cases, the correspondence is unique while in other cases the tool identifies different kinds of correspondences but can't solve their ambiguity: in this case the tool asks the designer to choose the right one. The tool builds the global attributes set to be associated to a cluster in two phases:

1. Union of the attributes of all the classes belonging to the cluster
2. Fusion of the "similar" attributes.

The union of the classes attributes consists in a mere collection of each class attributes in a single set. Such attributes set is a redundant set.

In the second phase (Fusion of the "similar" attributes), SI-Designer tries to eliminate these redundancies considering the relationships within the *Common*

Thesaurus. The fusion process is automatic for the attributes which are associated by validated relationships while it is not always automatic when their relationships are not validated.

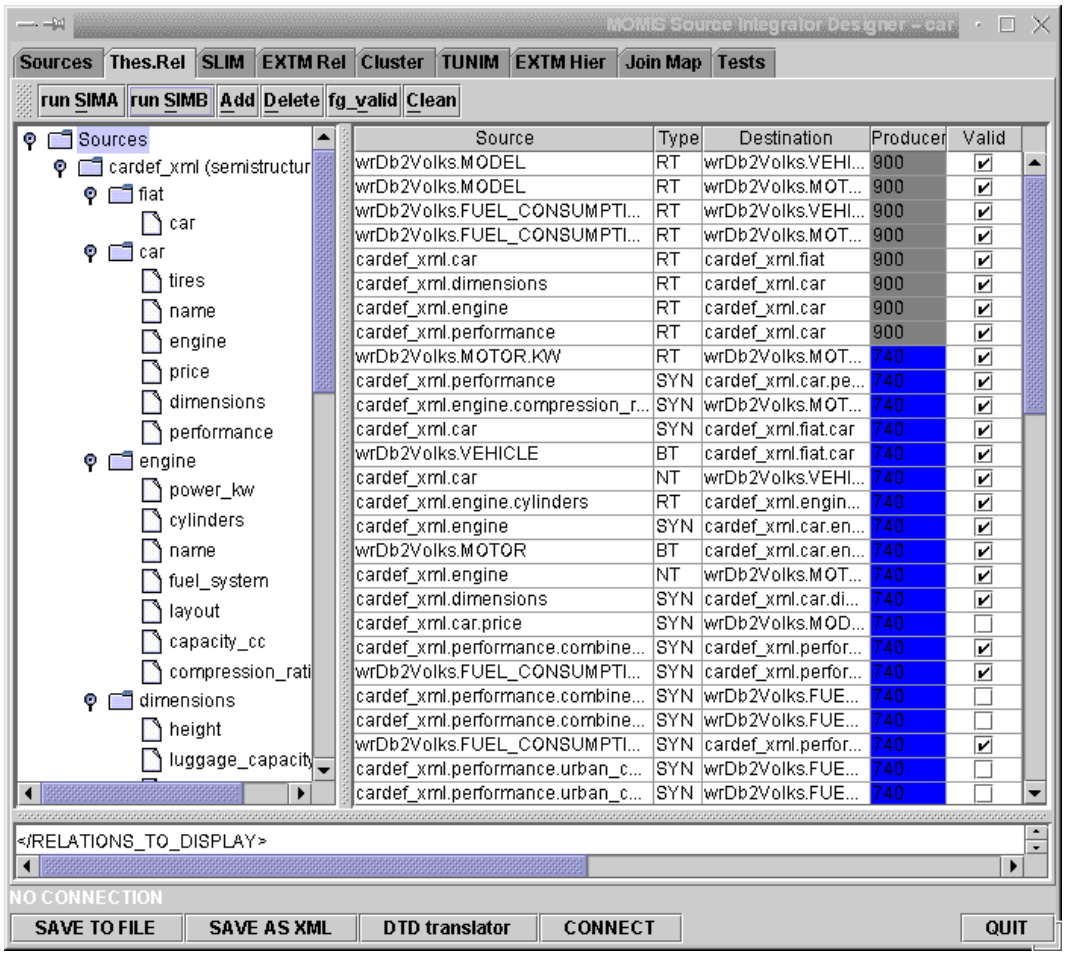

Figure 10 - Relationships within the Common Thesaurus

In particular, the system operates in the following way:

- **Attributes associated in validated relationships.**

  For these attributes the fusion is always automatic:

  - To each of the attributes connected by SYN relationships SI-Designer will connect one only global attribute: the domain and local attributes are the same and the name can be chosen by the designer between those proposed by SI-Designer or explicitly introduced.

- The attributes connected by NT relationships are treated by SI-Designer substituting them with a global attribute having the same name and the domain of the generalization attribute.

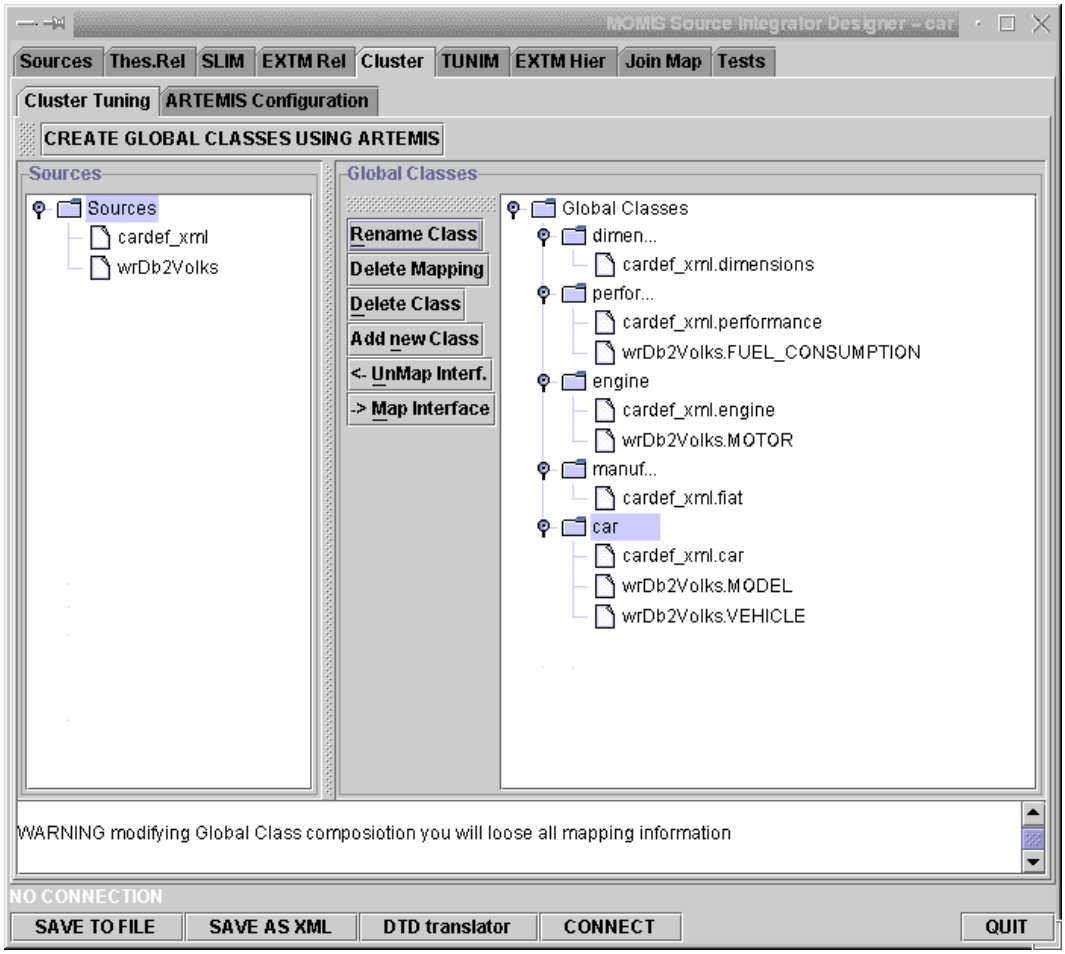

Figure 11 The Artemis module calculates the clusters that compose the Global Virtual View.

- **Attributes associated in non validated relationships.**

  Common Thesaurus relationships that do not passed validations belong to this category: SI-Designer can automatically find a global attribute only in a limited set of cases: it's up to the designer to add global attributes needed to complete the integration. The automatic individuation of a global attribute is only performed in this case, if the attributes in the relationships have the following requirements:

1. they are linked by SYN or BT relationship;
2. related classes belong to the same cluster;
3. they represents aggregation hierarchy (complex attributes or foreign key);

Once the global attribute set has been found, the designer can extend it to represent further local sources information: this case often occurs when some information is stored in a local source as a metadata.

While creating global attributes, SI-Designer builds also a *mapping-table* It is a $MT[CL][AG]$ table where CL represents the set of the local classes which belong to the cluster referred by the mapping-table, and AG represents the global attributes set built by MOMIS. Let C be the name of a local class, A the name of a global attribute and $AL$the name of a local attribute; each element $MT[C][A]$ of the table can assume the following values:

⇒ AL with $AL \in C$

This value is added when:

- The A global attribute refers to the information stored in local attribute.
- specialization relationship link attributes belonging to different classes.

⇒ AL_1 ***and,*** AL_2 ***and ...*** AL_n ***and*** with $AL_i \in C$

This is used when the value of the A attribute represents the linking of the values assumed by a set of attributes belonging to the same local class C.

⇒ ***case of*** AL $cost_1$: AL_1 $cost_2$: AL_2 ... $cost_n$: AL_n

where $AL, AL_i \in C$ $i = 1,...,n$ are $cost_i$, $i = 1,...,n$ constants.

This situation occurs when the A global attribute can assume one value in a set of AL_i local attributes belonging to the same class and the value choice passes through a third attribute, from the same class AL_i, which act as a selector.

⇒ *constant*

In this case the *A* global attribute value doesn't refer to any *C* local class attribute. *A* value is set by the designer according to the meaning set for the global attribute.

⇒ *null*

In this case *A* global attribute, while accessing the *C* local class doesn't get any value.

SI-Designer creates a set of global classes and a *mapping-table* for each global class. Furthermore, it provides the designer with an interface that allows a complete view of all the global classes (names and attributes), including the mapping-tables, class names setting and *mapping table* editing.

Finally, the ODL_{I^3} description of the Global Schema is obtained.

4.12 MOMIS from a KM perspective

The integration process supported by MOMIS can be viewed from a KM perspective. To do so, we have re-conceptualized the mediation process using the well known Nonaka knowledge creation model [50] (for a more detailed description about this topic refer to[13]): the result is described Figure 12.

The starting point is the input catalogs. They offer information that, coming from different sources, is not directly comparable. Resolving the syntactic and semantics inconsistencies existing between the various sources is the main goal of the MOMIS integration process, that can be subdivided into two sub-processes.

The first, automatically performed, is conducted by software components such as wrappers, WordNet and so on. Using the Nonaka terminology this process can be regarded as a *knowledge combination* process, since it applies the explicit knowledge embedded in the software tools to the explicit knowledge contained in the catalogs to produce new explicit knowledge[14]. Such new information is

[14]Nonaka and Takeuchi [50] precisely state that reconfiguration of existing information through sorting, adding, combining and categorising of explicit knowledge can lead to new

organized in a different way (as far as the logical structure, the semantics and the syntax are concerned) from the initial one. Nevertheless, it still includes mistakes and inconsistencies that make it useless.

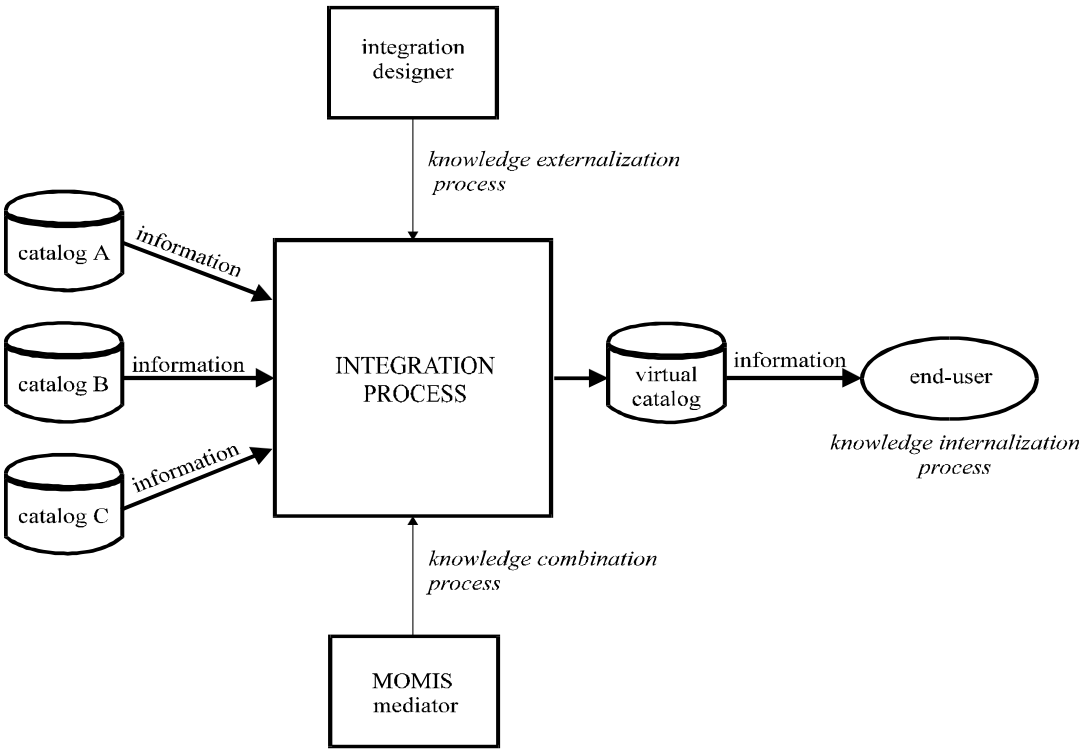

Figure 12 - A KM View of the MOMIS' integration process

The second entails the decisive contribution of the integration designer. It can be considered as a *knowledge externalisation* process, since the designer applies his tacit knowledge about the business domain to the information generated in the previous phase, to create new information, that will form the virtual catalog, i.e. the outcome of the integration process.

The last step concerns the search from the virtual catalog and the utilization of the retrieved information by the end-user. Also this process can be read according to the Nonaka taxonomy. In particular it can be considered a *knowledge*

knowledge.

internalisation process, since the end-user applies his tacit knowledge on the information contained in the virtual catalog to take business decision.

The whole mediation process creates new knowledge and, consequently, produces business value. The amount of the generated knowledge (and value) can be in first approximation estimated as the difference between the efficiency and efficacy[15] of the end-user decision made on the basis of the various catalogs separately taken, and that made starting from the unique virtual catalog.

The value added by the integration process depends on both the software components and the integration designer. The latter, in particular, must know not only the meaning of the terms contained in the catalogs, but also (and more) how the end-user employs the retrieved information. In other words, in order to give a full contribution to the integration process, the designer has to deeply know the business context in which information is used.

[15] In very short, efficiency is related to time and efforts spent in retrieving workable information; efficacy, instead, concerns the "quality" of the decision taken on the basis of the retrieved information.

5 A SOAP-enabled system for information integration

5.1 Web services at a glance

It is a largely widespread opinion that Web Services will be the fundamental building blocks in the move to distributed computing on the Internet. In fact, enterprises are moving their existing applications to the Web, and consequently a complete infrastructure to manage the specific issue introduced by the open platform is needed [30].

Several definitions of Web Services have been provided. In our opinion, a Web Services may be thought of as self-contained, modular applications that can be described, published, located and invoked over a network, generally, the World Wide Web. Essentially Web Service architecture may describe three roles [43], [41]:

- **Service provider**. From a business perspective, this is the owner of the service. From an architectural perspective, this is the platform that hosts access to the service.
- **Service requestor**. From a business perspective, this is the business that requires certain functions to be satisfied. From an architectural perspective, this is the application that is looking for and invoking or initiating an interaction with a service. The service requestor role can be played by a browser driven by a person or a program without a user interface, for example another Web Service

- **Service registry**. This is a searchable registry where service providers publish their service descriptions. Service requestors find services and obtain binding information (in the service descriptions) for services during development for static binding or during execution for dynamic binding. For statically bound service requestors, the service registry is an optional role in the architecture, because a service provider can send the description directly to service requestors. Likewise, service requestors can obtain a service description from other sources besides a service registry, such as a local file, FTP site, Web site, Advertisement and Discovery of Services (ADS) or Discovery of Web Services (DISCO).

Existing applications can be integrated more rapidly, easily and less expensively since Web Services reduce at the minimum what is absolutely required for interoperability. Integration occurs at a higher level in the protocol stack, based on messages centred more on service semantics and less on network protocol semantics, thus enabling real platform and language independence. These characteristics are ideal for connecting business functions across the Web both between enterprises and within enterprises. They provide a unifying programming model so that application integration inside and outside the enterprise can be done with a common approach, leveraging a common infrastructure. The integration and application of Web Services can be done in an incremental manner, by using existing languages and platforms and by adopting existing legacy applications. Previous platforms and architectures relying on distributed computing (CORBA, DCOM, Java RMI) have yielded systems where the coupling between various components is too tight to be effective for low-overhead, every time and everywhere B2B applications over the Internet. These approaches require too

much agreement and shared context among business systems from different organizations to be reliable for open, e-business cross-platform.

5.2 The SOAP approach

SOAP is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of serializing rules for expressing instances of application-defined data types, and a convention for representing remote procedure calls and responses.

All SOAP messages are encoded using XML. A major design goal for SOAP is simplicity and extensibility. This means that there are several features from traditional messaging systems and distributed object systems that are not part of the core SOAP specification. SOAP defines a message-processing model but does not itself define any application semantics, such as programming model or implementation-specific semantics.

The SOAP specification defines also the relationships between HTTP messages and SOAP. This HTTP binding is important because HTTP is supported by almost all modern operating systems. The HTTP binding is optional, but almost all SOAP implementations support it. The HTTP transport binding for SOAP makes it attractive for industrial uses. Since most organizations are familiar with HTTP and already have it incorporated into their network infrastructure, SOAP fits right in without the complex changes to the network or firewalls that many other protocols require.

One of the most relevant uses of SOAP is to enable XML Web services. An XML Web service is a function that is exposed through a SOAP interface so that other SOAP-based application on the Web can call it to access the service.

WSDL (Web Services Description Language) [64] is an XML format for describing network services as a set of endpoints operating on messages

containing either document-oriented or procedure-oriented information. As communications protocols and message formats are standardized in the web community, it becomes increasingly possible and important to be able to describe the communications in some structured way. WSDL addresses this need by defining an XML grammar for describing network services as collections of communication endpoints capable of exchanging messages. WSDL service definitions provide documentation for distributed systems and serve as a recipe for automating the details involved in applications communication. A WSDL file is an XML document that describes a set of SOAP messages, and how the messages are exchanged. Since WSDL is XML, it is readable and editable, but in most cases, it is generated and consumed by software. SOAP introduces the following advantages w.r.t. the communication mechanism used by the CORBA architecture:

- While IIOP, ORPC, are binary protocols, SOAP is a text-based protocol. Using XML for data encoding gives SOAP some unique capabilities. For example, due to the readability of a XML file, it is much easier to debug applications based on SOAP than a binary stream. Vice-versa the SOAP protocol it is not optimized to transfer huge data sources.
- Due to the communications among the different SOAP machines uses the HTTP protocol, no further configuration are needed in order to overcome firewalls and others protections.
- Because it is based on a vendor-agnostic technology, namely XML, HTTP, and Simple Mail Transfer Protocol (SMTP), SOAP appeals to all vendors.

5.3 Software at work

As the MOMIS is a distributed system, it requires a set of component to work properly to obtain the global schema.

First the wrappers must be running to get the ODL_{I^3} format of the local schemas. One wrapper is needed for each data source to be integrated. A parser within the wrapper translates the conceptual description of the data source (relational, object oriented, XML, etc.) into the corresponding ODL_{I^3} description.

The wrappers are also web services provider exposing two standard methods: the getDescription method which returns the ODL_{I^3} schema as a characters string and the getAnnotatedDescription method which returns both the ODL_{I^3} description and the annotation of the local schema.

A running wrapper performs the translation of the conceptual schema of the local source in unattended mode. The translation is then available for the mediator through the getDescription method.

The mediator provides a web service as well, since the WordNet dictionary is actually accessed through a set of methods which are exposed by the mediator.

For a local annotation, the wrapper must provide a soap client to invoke those methods and, in turn, to find out the meanings within the WordNet's lexicon. Unlike the translation into ODL_{I^3}, the annotation is a user driven task.

When the local annotation has been accomplished, the user issues the annotated schema. The information becomes available through the getAnnotatedDescription method which joins the terms of the ODL_{I^3} description with the address of their meanings within the WordNet database.

The mediator will then evaluate the lexical relationships that holds between the returned WordNet addresses and insert them into the common thesaurus.

5.4 The XML Web Services' role

Enabling XML Web Services within the original MOMIS architecture [cite] means, on one hand, extending the MOMIS capabilities through the benefits brought by a web services-based architecture (i.e. real platform independence, low

overhead service integration), but, on the other hand, it causes a significant evolution in the integration process.

This extended architecture decouples the common thesaurus generation task introducing a new actor in the integration scenario: the wrapper domain specialist. (WDS).

The WDS acts on a single data source. Once the wrapper has generated the ODL_{I^3} description of the local schema, the WDS uses the soap client located on the wrapper associated with the data source to access the WordNet web service running at mediator level. The WordNet web service allows the WDS to perform a precise annotation of the local schema by assigning the correct WordNet meaning to each term within the local schema.

The WDS differs from the very integration designer since he is supposed to supply a detailed knowledge about a specific data source rather than a global experience about the whole integration domain.

The common thesaurus generation resumes after each local schema description has been translated into ODL_{I^3} and, if possible, annotated by the WDS.

One of the most relevant advantages coming from the introduction of the WDS is releasing the integration designer from annotating each local schema. The uniqueness of the WordNet database, in addition, prevents ambiguity even in presence of many data sources (i.e. many different WDSs). The generation of the common thesaurus becomes a more rapid since the integration designer works, in general, on predetermined meanings.

Another major becomes valuable in presence of non meaningful terms within the local schema. In theses cases the direct experience of a WSD is fundamental to correctly convert abbreviations, acronyms and conventional words into meaningful terms.

The wrappers, within a web services-based architecture, are service providers exposing both the ODL_{I^3} description and the annotation of the local schema.

Nevertheless the wrapper must include a SOAP compliant module acting as Web Services requestor in order to access the WordNet Web Service and to provide the annotation of the local schema. A graphical user interface is also required to allow the WSD to easily browse the meanings available in WordNet and to assign them to the terms of the local schema.

That is, a generic wrapper becomes a more complex software with respect to the wrappers described in the earlier MOMIS versions [cite].

The introduction of the web services within the original I^3 architecture involves a trade-off between the complexity and the versatility of the wrappers with reference to optional implementation of a SOAP client module for the local annotation of the schema. The more complex wrapper application, the wider is the contribution to the whole integration process.

A lightweight wrapper designed to convert the local schema into the ODL_{I^3} format and to publish it as web service would be perfectly compliant with the rest of the system. It would be a straightforward solution (considering the variety of web services publishing tools available) well tailored for meaningful data source.

5.5 Running Example

Let us introduce the following example to illustrate the exploitation of the new architecture. Let us suppose that an industrial group had to run financial statistics about two controlled companies called CompA and CompB. The CompA company evaluates its financial performances by directly accessing the information about the invoices of a given year. This information is stored in a relational database. The CompB company creates year by year a XML file with data that could be useful in evaluating statistics.

The CompA's database	The CompB's XML schema
DOCH (ID, DT, DD, CID) DOCR (DID,RD,IID,Q) ITM (ID, DSC, PR, UM) CST(ID, NM, USA)	<?xml version="1.0" encoding="utf-8"?> <xs:schema targetNamespace=http://tempuri.org/fnSchema.xsd xmlns:xs="http://www.w3.org/2001/XMLSchema"> <xs:complexType name="InvoiceStat"> <xs:sequence> <xs:element name="ItemID" type="xs: integer "/>

	<xs:element name="ItemDesc" type="xs: string "/> <xs:element name="CustomerID" type="xs: integer "/> <xs:element name="CustName" type="xs: string "/> <xs:element name="Date" type="xs: date "/> <xs:element name="Price" type="xs:float"/> <xs:element name="Quantity" type="xs:float"/> <xs:element name="UnitMeas" type="xs:float"/> </xs:sequence> </xs:complexType> </xs:schema>

Figure 13 – CompA's database and CompB's XML Schema

This over-simplified example aims to illustrate the benefits coming from the exploitation of the distributed knowledge provided by the WDS rather than a complex integration scenario. The proposed integration domain requires a wrapper for relational databases and another one for XML sources. As the CompB file includes only meaningful terms, a Wrapper's Domain Specialist is not required for the local schema annotation. Therefore, a simple ODL*I3* parser for XML source files would be the recommended wrapper for the CompB source. The CompA database's schema holds plenty of acronyms and conventional words. Thus, only a local expert would be able to explain the meaning of each term, or, in other words, to annotate the local schema. A local annotation is typically very effective under those conditions. That is, let us focus on the CompA wrapper.

The considered wrapper basically performs two steps. First it establishes a connection to the CompA's schema and builds the corresponding ODL_{I^3} description on the basis of a fixed set of translation rules. The translated schema is stored in an environment variable of the wrapper as an XML string. This values remains unchanged unless the wrapper is stopped and restarted.

When the mediator retrieves the ODL_{I^3} schema through the *getDescription* web method, the web services inside the wrapper has only to read the environment variable and to assign it as return value of the method.

The second main purpose of the wrapper is the support to the local annotation. To enable a local expert (WDS) to locally annotate the schema, the wrapper links up with the WordNet dictionary which is only stored at mediator level.

The annotation includes two steps: the base form choice and the meaning choice. The former requires the WDS to select the word form from the list of suitable base forms supplied by the WordNet morphologic processor. This is accomplished by invoking the *checkWord* web method on the mediator, which returns an acknowledgment only if the word, has been found within WordNet. If a base form is not found — as we could expect in the CompA case — the WDS can directly introduce it. In our example, the "customer" base form would replace the CST term. Likewise, the WDS introduces meaningful base forms for each term of the local schema.

```
interface customer {
        attribute   integer  code;
        attribute   string   name;
        attribute   string   address;
}
```

Figure 14 - The CompA's ODL$_I^3$ interface for the CST relation

The Figure 14 - The CompA's ODLI3 interface for the CST relation illustrates the ODL_{I^3} format for the CST class, once that the correct forms have been selected by the WDS. The latter is the meaning choice. The designer can relate a name to one, more than one, or no meaning. The choice of not relating a name to any meaning can be made for different reasons: the concept is too complex and it cannot be expressed with one word; it belongs to the tops, i.e. to generic concepts.

The meanings are retrieved by invoking the *getSense* web method exposed by the mediator. When called for a base form, the function puts into a bidimensional array of values all the meanings associated to the form and the corresponding logical address within the WordNet's data files. The WDS is then allowed to

choose on or more meanings. The wrapper stores, for each base form in the local schema, only the logical address of the meaning indicated by the WDS. In the CompA case, the WDS associates the first meaning to the name attribute of the customer class: it turns the wrapper keeps the value corresponding logical address in the running version of WordNet (in this case 12548).

At the end of the annotation the WDS is required to save his choices and to issue the schema. From this point on, the mediator is allowed to invoke the *getAnnotatedDescription* on the wrapper to exploit the local annotation. Notice that any semantic ambiguity between concepts among different schemas will be removed by the annotation since a third party-supplied morphology (i.e. the WordNet dictionary) is trustfully shared by each source.

As the WDS selects one or more meanings from those found in WordNet starting from the chosen base form, all the words that are related to the same name, share the same base form.

That is the WDS experience becomes fundamental to make the correct choice. The annotation activity could be significantly long in terms of time even for the WDS. The semi-automatic approach supplied by the wrapper reduces the complexity of the task, in fact, a "difficult" problem (i.e. finding the relations between all words), is divided in many "easy" ones, choosing each terms' meaning from a list. Furthermore the wrapper provides a graphical representation of the generalization hierarchy of the meanings in order to help the WDS in the most difficult choices.

Once that the mediator has gathered the local schemas the common thesaurus generation starts. Lexical relationships are extracted in the following order:

Schema-derived relationships: extracted by analysing each ODL_{I^3} schema separately.

```
CompA.DOCR RT CompA.DOCH
CompA.CST RT CompA.DOCH
CompA.ITM RT CompA.DOCR
```

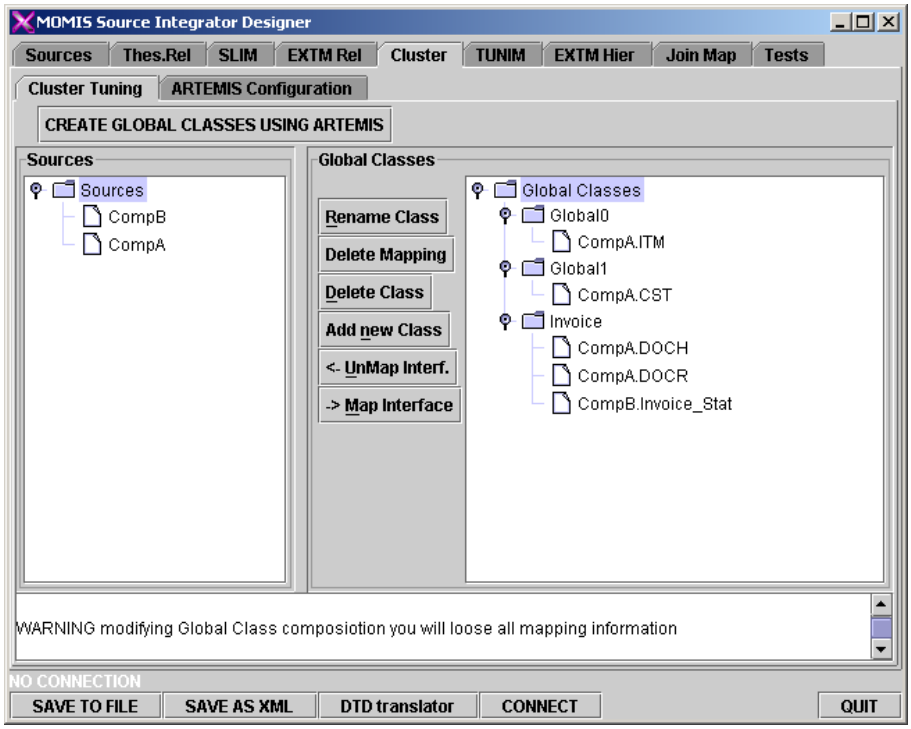

Figure 15 - Global classes

Lexicon derived relationship: extracted exploiting the lexical relationships existing between terms in WordNet. If annotated schemas are gathered from the wrappers (see CompA case) the mediator has only to run the extraction algorithm. Otherwise (see CompB source in our example) the integration designer must first annotate the local schema and then extract the relationships

`CompB.Invoice_Stat NT CompA.DOCH → (Invoice NT Document)`

`CompA.DOCR NT CompA.DOCH → (line NT head)`

`CompB.Invoice_Stat.Date SYN CompA.DOCH.Date`

Inferred relationships: holding at intra-schema level, are inferred by exploiting the inferencing capabilities of ODB-Tools.

`CompA.CST RT CompB.Invoice_Stat`

`CompA.ITM RT CompB.Invoice_Stat`

All these relationships are added to the Common Thesaurus and thus considered in the subsequent phase of construction of Global Schema. The Figure 15 shows the global classes obtained for the considered integration domain.

The mapping table in Figure 16 illustrates the correspondence between the global attributes and the attributes of the local schemas.

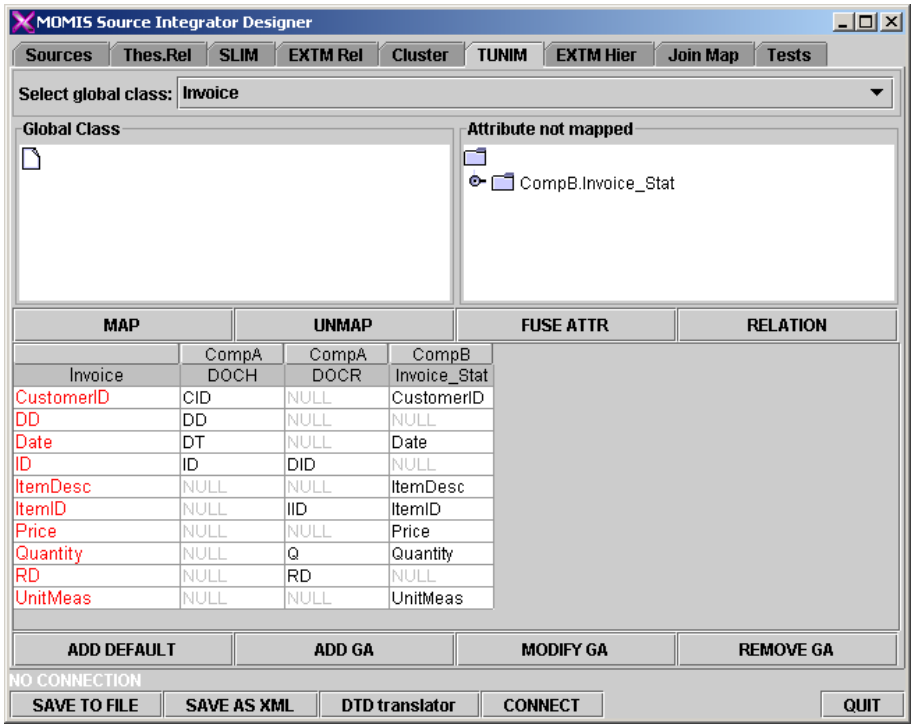

Figure 16 - Mapping tables

Concluding remarks

6 References

[1] Aberdeen Group, 2001, *Component-Based Architectures: Time to Migrate the Enterprise Application Portfolio*, October, www.aberdeen.com

[2] Alavi M., Leidner D.E., 1999, "Knowledge Management Systems: Issues, Challenges, and Benefits". *Communication of the Association for Information Systems*, vol. 1, art. 7,.

[3] Alavi M., Leidner D.E., 2001, "Knowledge Management and Knowledge Management Systems: Conceptual Foundation and Research Issues", *MIS Quarterly*, vol. 25, n. 1, 107-136

[4] Ang A., Tang N., 2001, "The Relationship between Knowledge Management and E-business", *Managing Knowledge: Conversations and Critiques*, Leicester, April 10-11

[5] Archer N., Gebauer J., 2001, "B2B Applications to Support Business Transaction: Overview and management Considerations", in Warketing M. (ed.), *Business to Business Electronic Commerce*, Idea Group Publishing, forthcoming

[6] Arens Y., Chee C.Y., Hsu C., and Knoblock C. A.. Retrieving and integrating data from multiple information sources. *Int. Journal of Intelligent and Cooperative Information Systems*, 2(2):127–158, 1993.

[7] Ariba, 2000, *The role of a Network in B2B Commerce*, www.ariba.com

[8] Arthur M. Keller. Smart catalogs and virtual catalogs. In International Conference on Frontiers of Electronic Commerce, Oct 1995.

[9] Bates M. Subject access in online catalogs: A design model. *Journal of the American Society for Information Science*,11:357–376, 1986.

[10] Benetti I., Beneventano D., Bergamaschi S., Corni A., Guerra F., and Malvezzi G.. SI-Designer: a tool for intelligent integration of information. *International Conference on System Sciences (HICSS 2001)*, January 2001.

[11] Benetti I., Beneventano D., Bergamaschi S., Guerra F., and Vincini M. SI-Designer: an intelligent tool for e-commerce. *IJCAI-01 Workshop on e-business and the intelligent web*. Seattle USA 2001

[12] Benetti I., Beneventano D., Bergamaschi S., Guerra F., Vincini M., *An Information Integration Framework for e-commerce*, IEEE Intelligent Systems Magazine, (January/February 2002).

[13] Benetti I., Bergamaschi S., Scarso E. "Managing knowledge thorough electronic commerce applications: a framework for integrating information coming from heterogeneous sources", *International Journal of electronic business*. To be published.

[14] Benetti I., Scarso E "Commercio Elettronico e piccole e medie imprese: opportunità e vincoli tecnologici" In *Atti XIII Riunione Scientifica AiIG - Impresa e Competizione Knowledge-Based*, vol. 1, 183-204.

[15] Benetti I., Scarso E "Architetture e applicazioni informatiche di commercio elettronico: una ricognizione sullo stato e le traiettorie di sviluppo delle tecnologie", *Programma di Ricerca Rcientifica di Rilevante Interesse Nazionale* "Il commercio elettronico: nove tecnologie e nuovi mercati per le pmi". Rapporto tecnico. Dicembre 2001

[16] Beneventano D., Bergamaschi S., Castano S., and Vincini M. Semantic integration of heterogeneous information sources. *Journal of Data and Knowledge Engineering*, 36(3):215–249, 2001.

[17] Beneventano D., Bergamaschi S., Castano S., Corni A., Guidetti R., Malvezzi G., Melchiori M., and Vincini M., '*Information integration: The MOMIS project demonstration*, Proceedings of 26th International Conference on Very Large Data Bases, September 10-14, 2000, Cairo, Egypt, pages 611–614. Morgan Kaufmann, 2000.

[18] Beneventano D., Bergamaschi S., Lodi S. and Sartori C. "Consistency Checking in Complex Object database Schemata with Integrity Constraints" in *IEEE Transactions on Data and Knowledge Engineering*, vol. 10, pag. 576-598. 1988

[19] Beneventano D., Bergamaschi S., Sartori C., and Vincini M. *ODB-Tools: A Description Logics Based Tool For Schema Validation and Semantic Query Optimization in Object Oriented Databases*, Proceedings of IEEE Int. Conference on Data Engineering (ICDE-97), Birmingam UK 1997.

[20] Beneventano D., Bergamaschi S., Sartori C., and Vincini M. ODB-QOPTIMIZER: A tool for semantic query optimization in oodb. *In Int. Conference on Data Engineering - ICDE97*, 1997.

[21] Bergamaschi S., Castano S., and Vincini M. Semantic integration of semistructured and structured data sources. *SIGMOD Records, 28(1)*, March 1999.

[22] Bergamaschi S., Nebel B. "Acquisition and Validation of Complex Object Database Schemata Supporting Multiple Inheritance", *Journal of Applied Intelligence*, Kluwer Academic Publishers, Boston, vol. 4 1994, pp. 185-203.

[23] Blumentritt R., Johnston R., 1999, "Towards a Strategy for Knowledge Management", *Technology Analysis & Strategic Management*, vol. 11, n. 3, 287-300

[24] Bolisani E., Scarso E., 1999, "Information technology management: a knowledge-based perspective", *Technovation*, vol. 19, 209-219

[25] Bowman B. J., 2002 "Building Knowledge management Systems". Information Systems Management. Summer 2002

[26] Buneman P., Raschid L. and Ullman J. Mediator languages - a proposal for a standard, April 1996.

[27] Catarci T. and Lenzerini M. Representing and using inter-schema knowledge in cooperative information systems. *Journal of Intelligent and Co-operative Information Systems*, 2(4):375–398, 1993.

[28] Cattell R. G. G., editor. The Object Database Standard: ODMG 3.0. Morgan Kaufmann Publishers, San Mateo, CA, 2000.

[29] Chiu C., 2000, "Reengineering Information Systems With XML", *Information Systems Management*, Fall, 40-54

[30] Curbera F., Duftler M., Khalaf R., Nagy W., Mukhi N., Weerawarana S., *Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI*, IEEE Internet Computer, March-April 2002, pp. 86-93.

[31] Deloitte Research, 2000, *The Future of B2B. A New Genesis*, www.dc.com/deloitte_research

[32] Fisher Maydene. JDBC Database access http://java.sun.com/docs/books/tutrial/jdbc/index.html

[33] Gebauer J., Scharl A., 1999, "Between Flexibility and Automation: An Evaluation of Web Technology from a Business Process Perspective", *Journal of Computer Mediated Communication*, vol. 5, n. 2, www.ascusc.org/jcmc

[34] Gilarranz J., Gonzalo J., and Verdejo F. Using the eurowordnet multilingual semantic database. *In Proc. of AAAI-96 Spring Symposium Cross-Language Text and Speech Retrieval*, 1996.

[35] Gupta A.K., Govindarajan V., 2000, "Knowledge Management's Social Dimension: Lessons From Nucor Steel", *Sloan Management Review*, Fall, 71-80

[36] Hammer K., 2001, "Almost Perfect: Where Middleware and XML May Fail to Deliver", eAI Journal, June, 12-16

[37] Haustein S., *Semantic Web Languages: RDF vs. SOAP Serialisation*, Proceedings of the Second International Workshop on the Semantic Web - SemWeb'2001, Hongkong, China, May 1, 2001.

[38] Hendriks P., 1999, "Why Share Knowledge? The Influence of ICT on the Motivation for Knowledge Sharing", *Knowledge and Process Management*, vol. 6, n. 9 , 91-100

[39] Holsapple C.W., Singh M., 2000, "Electronic Commerce: From a Definitional Taxonomy Toward a Knowledge-Management View", *Journal of Organizational Computing and Electronic Commerce*, vol. 10, n. 3, 149-170

[40] Hull R. and King R. et al. ARPA I^3 reference architecture, 1995. Available at http://www.isse.gmu.edu/I3 Arch/index.html.

[41] IBM Web Service Architecture Team, Web Services architecture overview – the next stage of evolution for e-business, September 2000.

[42] Kocharekar R., 2001, "K-commerce: knowledge-based commerce architecture with convergence of E-commerce and Knowledge management", *Information Systems Management*, Spring, 30-35

[43] Kreger, *Web Services Conceptual Architecture (WSCA 1.0)*, May 2001 – IBM Software group - http://www-3.ibm.com/software/solutions/webservices/pdf/WSCA.pdf.

[44] Maglitta, J. 1995 "Smarten up!", ComputerWorld, 29(23), pp. 84-86

[45] Malhotra Y., 2000, "Knowledge Management for E-Business Performance: Advancing Information Strategy to "Internet Time"", *Information Strategy: The Executive's Journal*, vol. 16, n. 4, 5-16

[46] Markus M.L., 2001, "Toward a Theory of Knowledge Reuse: Types of Knowledge Reuse Siuations and factors in reuse Success", *Journal of Management Information Systems*

[47] McDermott R., 1999, "Why Information Technology Inspired But Cannot Deliver Knowledge Management", C*alifornia Management Review*, vol. 41, n. 4, Summer, 103-117

[48] Miller A.G. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995. Object management group. http://www.omg.org/, 2000.

[49] MOMIS Staff Available at htto://www.dbgrouop.unimo.it/

[50] Nonaka I, Takeuchi H, 1995, *The Knowledge Creating Company*, Oxford University Press, New York

[51] Object Management Group http://www.omg.org

[52] Paolillo E., 2001, "Usare XML per le applicazioni di Commercio Elettronico", in Pozzi P., Casagni M., De Sabbata P., Vitali F. (a cura di), *Commercio Elettronico e XML. Scenari, tecnologie, applicazioni*, Angeli, Milano, 81-101

[53] Philips C., Meeker M., 2000, *The B2B Internet Report*, Morgan Stanley Dean Witter

[54] Requisite Technology, 2000, *A Catalog Content Management White Paper*, www.requisite.com

[55] Roberts J., 2000, "From Know-how to Show-how? Questioning the Role of Information and Communication Technologies in Knowledge Transfer", *Technology Analysis & Strategic Management*, vol. 12, n. 4, 429-443

[56] Roth M.T. and. Scharz P Don't scrap it, wrap it! a wrapper architecture for legacy data sources. In *Proc. of the 23rd Int. Conf. on Very Large Databases, Athens*, Greece, 1997.

[57] Schmitt I. and Turker C. An Incremental Approach to Schema Integration by Rening Extensional Relationships. In G. Gardarin, J. French, N. Pissinou, K. Makki, and L. Bougamin, editors, *Proc. of the 7th ACM CIKM Int. Conf. on Information and Knowledge Management*, November 3{7, 1998, Bethesda, Maryland, USA, pages 322{330, New York, 1998. ACM Press.

[58] Suciu D. Semistructured data and XML In Proc. of *International Conference of Foundations of Data Organization* (FODO'98) Kobe, japan 1998

[59] Teece D.J., 2000, "Strategies for Managing Knowledge Assets: the Role of Firm Structure and Industrial Context", *Long Range Planning*, vol. 33, 35-54

[60] Tiwana A., 2000, *The Knowledge Management Toolkit*, Prentice Hall, Upper Saddle River, NJ

[61] Tuomi I., 2000, "Data Is More Than Knowledge: Implications of the Reversed knowledge Hierarchy for Knowledge Management and Organizational Memory", *Journal of Management Information Systems*, vol. 16, n. 3, 103-117

[62] Vance, D. M. "Information, Knowledge and Wisdom: The Epistemic Hierarchy and Computer-Based Information System", In *Proceedings of the Third Americas Conference on Information Systems*, B. Perkins and I Vessey (eds.), Indianapolis, IN, August 1997

[63] W3C, *Simple Object Access Protocol (SOAP) 1.2*, W3C Working Draft - 26 June 2002.

[64] W3C, Web Services Description Language (WSDL) 1.1, W3C Note 15 March 2001.

[65] Williams H., Whalley J., Li F., 2000, "Interoperability and Electronic Commerce: A New Policy Framework for Evaluation Strategic Options", *Journal of Computer Mediated Communication*, vol. 5, n. 3

[66] Woods W.A., Shnmotze J.C. "The kl-one family" In F.W. Lehman (ed.), special issue of *Computers & Mathematics with Applications* vol. 23 No. 2-9 1989

[67] S. Castano and V. De Antonellis. A schema Analysis and Reconciliation Tool Environment for Heterogeneous Databases. In Proc. of IDEAS'99 Int. Database Engineering and Applications Symposium, Montreal, Canada, August 1999.

[68] Castano S., De Antonellis V., and De Capitani di Vimercati S.. Global viewing of heterogeneous data sources. Transactions on Data and Knowledge Engineering, 13(2), 2001.

[69] Castano S., De Antonellis V., and De Capitani di Vimercati S. An xml-based interorganizational knowledge mediation scheme to support b2b solutions. In Proc. of 9th IFIP 2.6 Working Conference on Data Semantics (DS-9), Hong Kong, April 2001.

[70] Castano S., De Antonellis V., De Capitani di Vimercati S. and Melchiori M.. An xml-based integration scheme for web datasources. ISI Journal-Special Issue on Data Reengineering for the Web, 6(1), 2001.