Introduction

With the ease of information exchange through the spread of communication technologies, the value of information to an enterprise has become evident. Concise, accurate and up-to-date information is considered a key asset that must be maintained carefully. This has been recognized by many large companies and public authorities who have appointed a chief information officer (CIO) in charge of planning, developing and managing information assets and interweaving them into a copious landscape of enterprise knowledge. Information integration across application boundaries or even across company boundaries is a topic of interest to many enterprises.

Currently Europe’s corporations spend over 10 billion Euro in dealing with these problems. According to studies, up to 30%\(^1\) of future IT spending will go into Enterprise Application Integration (EAI). There are various driving forces for the increasing needs of integration in existing applications (Fensel 2003): company mergers require integration of software infrastructures; new software systems need to be integrated with legacy systems; there are no integrated, optimal solutions for all the needs of a company; new protocols and standards continue to emerge and companies

\(^1\)David Sink cited in InformationWeek: \(//www.informationweek.com/story/IWK20020524S0005, \text{InfoWorld}\) January 2002 survey of 500 IT leaders: \(//www.infoworld.com/article/02/02/22/020225feintro1.html\)
need to be compliant with these standards in order to enable cooperation with other companies. Consequently, disparate information, spread over various sources, in various formats, and with non-matching semantics, is a major obstacle for enterprises in using their information to its full potential.

Many organizations discovered that, even if they use exactly the same hardware and software components during data interchanges, differences in the way database or XML schemas are defined at the syntactic and semantic level, no matter how small, can still hamper information sharing and reuse.

When a new application is introduced to an enterprise, the application will typically have to use some data that is already existing somewhere within the organization. It is, however, often hard to find out whether this data is actually already present and, if it is present, it is even harder to obtain its exact location and to understand its precise meaning (semantics). This is because existing databases are often created from the point of view of a particular application and not from a broader company-wide point of view, which makes reuse across different applications difficult. Owing to those difficulties, developers tend to create new data sources that have some semantic overlap with existing sources. For a particular application this might be an efficient solution – from a company-wide perspective, however, it is the worst solution.

How does such an uneven information landscape evolve? Even a practised CIO cannot avoid conflicting requirements during the development of applications inside an organization. The reasons are manifold; some examples are:

- Changing market needs and thus changing requirements for an application domain may entail the extension and adaptation of information structures. If these changes are not done with care they may cause legacy risks in information structures. Old structures may live alongside new ones, because not all application parts are adapted.

- Optimizing internal business processes may require the integration of two independently developed in-house applications. Coming from different domains these applications may use different terminologies, different information structures and, even worse, deviating semantics. A simple integration strategy may cause redundancies or inconsistencies in the overall information landscape.

- Off-the-shelf applications independently developed by third parties must be integrated. There is typically only a very limited chance to influence the underlying concepts.
External requirements may force adaptations – for example, legal requirements, interfaces to other organizations and so on.

Situations as listed before can arise even in small scenarios. The information integration challenges can be identified on various levels:

- Information may be represented in incompatible terminology – for example, one application uses a table named customer, another application uses a table named client.
- Information is structured according to different paradigms – for example, flat tables in relational databases, spreadsheets or hierarchical trees in XML.
- Finally the semantics may differ – for example customers in one data source may be named partners in another and combined up with dealers in the same source.

1.1 FINDING A WAY OUT OF THE DILEMMA

Only a rigorous information integration approach can avoid such problems. Optimally, information integration is done in parallel with the evolution of the information sources. However, in general, strategic information integration is not employed before severe problems arise.

The most obvious, but weakest, option is to build ad-hoc import/export bridges that interface between two applications. In most cases, these bridges are one-way, which means that information is transported from one application to the other, but not necessarily the other way around. These bridges are typically incorporated into either the exporting or importing application. Thus knowledge about the bridges is only maintained and coordinated locally.

A more structural solution is to develop a general mapping of concepts between two applications. These (one-way or two-way) mappings can be documented and maintained on a longer term. However they have a major disadvantage: the number of point-to-point mappings that need to be maintained (at least for the core concepts like, for example, customers or products), increases more or less with $O(n^2)$ (the square number of application domains).

Another option is to base the mappings on a central repository containing a canonical data model. The central repository defines all concepts and
their interrelations in an application-independent way. The main advantage of such a solution is that the repository can be developed and maintained as a central store of information on top of the individual applications. For each application there must be individual (two-way) mappings that define the relationship to the concepts in the repository and hence also the relationship to other application domains.

This last option is used in practice by data warehouses in management information systems (MIS): a common information model is developed. All relevant information from different information sources is regularly imported into a central database, based on the common model, and merged in order to gain higher-value, more compact, information.

Data warehouses face the problem of bringing disparate information together into a joint model. Most often this model is based on the relational data model. However data warehouses in general implement a one-way street: information is imported into a data warehouse, but no information gets back to the information sources. The data in a data warehouse is not available for further manipulation, but only for further aggregation and analysis.

More general solutions are needed to support the documentation, usage and maintenance of such a central repository. This is where ontologies come into play. The purpose of an ontology is to define a shared conceptualization. This means that an ontology can help to define an enterprise-wide common understanding of concepts. Having been a research topic since the late 1990s, ontologies nowadays provide a large set of features to cope with problems arising from information integration. Ontologies allow the definition of real-world semantics in a way that makes them both processable by machines and understandable by humans. Ontologies can provide support for handling disparate terminologies, such as, for example, synonym and homonym handling. They can also provide sophisticated mechanisms to define a complex knowledge model. Since ontologies abstract from the physical representation, they can mediate between different storage concepts. Last but not least, ontologies provide a flexible way to specify semantics to a set of data sources.

The purpose of this book is to introduce the concepts, the methodology and the tools needed to build such an central ontology. Also it will discuss and contrast them with other methods for information integration.
INTRODUCTION

1.2 THE BACKGROUND TO THIS BOOK

The results presented in this book were developed within the joint EU-funded Corporate Ontology Grid (COG) project. It was carried out by three companies: Fiat richerca (Italy), the research department of the automotive manufacturer Fiat, Unicorn (Israel), a company providing tools and consultancy for semantic data management for large enterprises, and LogicDIS (Greece), a leading developer of business applications. The Institute of Computer Science at the University of Innsbruck provided scientific advice in the application of ontologies.

In the COG project we investigated the problems in semantic heterogeneity between data sources in an enterprise and how to overcome them by semantic integration of the sources using a central information model (i.e., an ontology). We built the information model using existing applications, data sources (assets) and input from domain experts. Precise mappings between each data asset and the central model ensure a well-understood meaning for the concepts in each asset. These mappings enable application architects to discover the location of information in data sources, spread throughout the enterprise. Furthermore, because the mappings are created on a detailed level, they enable automatic query translation and the automatic generation of transformations between different sources.

The aim of the COG project was to create such an information architecture and to combine existing real live data sources. Production Systems at Fiat were used as the sources for a thorough and in-depth study of the potential applications of ontology technology for semantic information integration in industrial enterprises.

1.3 THE STRUCTURE OF THE BOOK

In this book we present the approach, the methods applied and the lessons learned during the COG project. It is structured in a series of chapters that discuss the various aspects of information integration through a centrally managed ontology.

2www.cogproject.org, sponsored by the EU under contract number IST-2001-38491.
1.3.1 Data modelling and ontologies

The information sources in an enterprise can be viewed from different perspectives: an internal, technology-oriented perspective, an application-oriented perspective or a conceptual, integration-oriented perspective. This was already recognized with the three-schema architecture (Tsichritzis and Klug 1978) for database management systems. In this chapter we use this architecture to discuss how traditional database management systems view the world. We contrast this view with an introduction to ontologies. Ontologies are applied in a wide range of application fields from linguistic thesauruses to application fields such as domain modelling, meta data annotations and automated reasoning. We present ontologies and their modelling constructs and show how they can be applied to form a central repository to which individual data sources can be mapped.

Ontological modelling is the basis for semantic information management (SIM), which provides an architecture and a methodology for developing such a central ontology. We show the layers of the architecture together with the steps proposed to build such an ontology.

1.3.2 Information integration with relational databases and XML

The support for data and information integration has become an important requirement, which has to be answered by major database technology vendors and integration tool providers. Various types of architectures, such as message-oriented middleware or so-called integration hubs have been provided. This chapter introduces different approaches for the integration of relational databases or XML-based document structures, such as low-level file transfer, up to sophisticated transaction-oriented synchronization mechanisms. It discusses technical solutions for data integration such as database sharing or federation. XML as an established standard format for exchange of documents is also considered a key technology for data integration. In particular its support for data transformations has generated a market for integration tools.

1.3.3 The show case

Throughout the rest of the book we will refer to examples taken from the show case used in the COG project. This show case originated from real data sources from the automotive industry and exhibited the typical problems
found when working with heterogeneous data sources. It was used to verify the concepts and to gain experience in dealing with real-life industrial problems.

1.3.4 Semantic information integration

In this chapter we present the approach to semantic information integration which was applied in the COG project. We describe how the semantic information architecture along with the Unicorn Workbench tool were applied in the project to solve the information integration problem. We use the semantic information architecture methodology and the Unicorn Workbench tool to create an information model (an ontology) based on data schemas taken from the showcase. We map these data schemas to the information model in order to make the meaning of the concepts in the data schemas explicit and to relate them to each other, thereby creating an information architecture that provides a unified view of the data sources in the organization. We furthermore provide an extensive survey of other efforts in semantic information integration and a comparison with our approach in the COG project.

1.3.5 Data source queries

In subsequent chapters we describe the conceptual ontology query language developed by Unicorn, look into the querying support provided by the Unicorn Workbench and describe the way the querying architecture is built in the COG project. This is completed by a review of comparable approaches in the literature.

1.3.6 Generating transformations

Here we describe how transformations can be generated and used. EAI systems try to overcome heterogeneity problems between different schemas. Nevertheless, current EAI approaches require significant time and cost and provide solutions that do not scale. Developers have to create transformation programs manually to deal with the heterogeneous data sources in a company. This solution may provide satisfactory results for a limited number of schemas. However, the number of transformations increases with the square of the number of schemas, hampering the scalability of the solution.
Maintainability problems also appear, as changes in the data schemas require manual rewriting of the transformation code to adapt the transformations to the changes in the schemas.

A survey on current approaches to define transformations in EAI is also presented and compared to our approach.

1.3.7 Best Practices and Methodologies

This chapter focuses on best practices and lessons learned during the COG project and the methodology used in integrating the heterogenous data sources of Centro Ricerche Fiat. Another important aspect of this chapter is the best practices learned concerning the multilingual challenges of the COG project.

The reader will find a glossary of frequently used terms from the COG project at the end of the book, in order to make it easier to look up the definition of these terms.