8-15-1997

Integrating Distributed Data Over Their Semantic Identity

Nektarios Georgalas
UMIST, georgala@co.umist.ac.uk

Integrating Distributed Data Over Their Semantic Identity

Nektarios Georgalas, Information Systems Group, Department of Computation, UMIST, PO BOX 88, Sackville Street, M60 1QD, Manchester UK
tel:0044-161-2003744, fax:0044-161-2003364, email:georgala@co.umist.ac.uk

Abstract

In this paper, we present a method that integrates distributed data managed by a variety of DBMSs. We focus on an infrastructure designed to assist business modelling and requirements analysis. ERA method that describes the enterprise's structure provides metamodels which are stored in the SIS semantic repository. The infrastructure suggested uses the semantics of the database information which are acquired by the application of a database re-engineering method and depicted in a EER model. We explain the framework and we describe the structure of our approach and the mechanisms that accomplish the data integration.

Introduction

Enterprises nowadays are taking advantage of the Internet and Intranet technology and seek methods to integrate their Information systems and databases. It is observed that strategically significant information of the company is disparate within the distributed data stores and is often unexploited with critical consequences. As such crucial issues are spawned: How we keep track of this information, how we bridge the gap of incompatible legacy systems that manage it to embrace new technology and how it can be obtained each time that is needed in order to assist Requirements Engineering process.

Our research group is currently involved with issues within the scope of the generic problem of business modelling. The up to date deliverables are assembled in the ERA (Enterprise Requirements Analysis) methodology [1]. This method supports a set of metamodels which describe an Enterprise structure from different aspects. Key persons process the metamodels provided in order to capture the present enterprise goals, how they are achieved, and how goals and operations have to be changed for the enterprise benifit. For analyzing such a problem, the operational Enterprise data is critical in terms of illustrating the present organization status and therefore we need to incorporate it to the entire process. A direct access to it is necessary.

Framework

ERA metamodels are in a higher abstraction level than models that represent the current Enterprise schema. This conceptual schema describes the entites involved with the business functions as well as the associations among them. The schema instantiates the metamodels recommended by ERA methodology. The raw operational data that reside as simple records in database tables instantiate the Enterprise schema. Different DBMSs may be assigned to the data management and can differ on the data model they adopt. For the last decade, the relational model seems to dominate [8] .

The management of the models is supported by a special semantic repository called SIS (Semantic Indexing System) and an associated network platform. SIS is an object oriented database and provides a hierarchy of abstraction layers. Objects that reside on a higher level are instantiated by objects of the immediate lower level and so on, till we reach the lowest level of plain individual items. This hierarchy hosts the objects of ERA metamodels at the *Meta-level* M followed by their instances that comprise the Enterprise's conceptual schema on the *Simple-level* S, and finally the simple tokens (raw data) on the lowest *Token* layer. The classes are introduced in the repository using the TELOS language ([2], [3])and can be invoked by using the QI (Query Interpreter) [4]. Both TELOS and QI have been especially designed to support the whole system.

The last level of SIS can be populated by some of the real data to accomodate the ERA methodology. Nevertheless, not all of the information that is dispersed among a company's databases can be transfered on the Token level as that would cause overloading and redundancy. Instead, channels to the data's original locations should be established since they have a major importance for ERA function. Our approach grants the paths to this data and assists in overcoming problems caused by the variety of DBMSs vendors or the data distribution.

Approach

Our approach tends to preserve SIS as the "heart" of the entire framework. To comply with this requirement the disparate information needs to be represented in the repository via an object model. The SIS semantic models follow the EER (Extended Entity Relationship) principles and consequently it is necessary to extract an EER schema from the data of the business relational databases. This model will supplement the enterprise schema that is already stored on S-Level. Hence we distinguish two parts of the S-level information regarding the enterprise:

- certain knowledge describing the organization from the ERA point of view which we will hereinafter call *SIS enterprise Schema*.
- the enterprise schema as it is developed and pictured in the business's legacy system which we will call *Legacy Schema*.

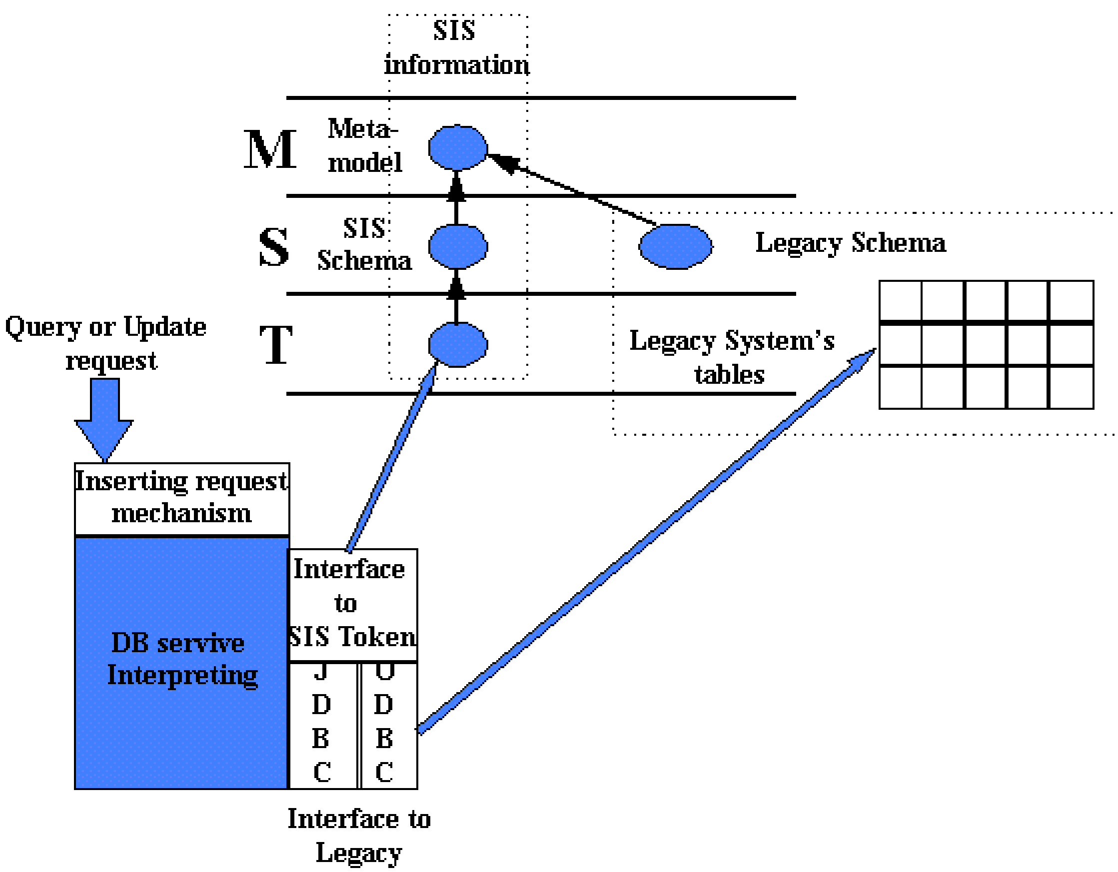

Figure 1
Concepts of SIS and legacy system access

The SIS schema is produced from the application of the ERA methodology to the specific organization, and encompasses classes that instantiate the ERA metamodels of M-level. The Legacy schema is added on the

S-layer after Reverse Engineering the business's databases, in order to depict the real physical data interrelation (entities, relationships etc.). It is obvious that the union of both sets gives the unified *Enterprise Schema*. It is necessary to draw this line for the following reasons:

- A query that is released by a user may need to invoke and combine mixed information from either the SIS or the legacy databases. That is because tokens that instantiate the SIS schema classes reside in the repository, whereas tokens that instantiate the Legacy schema are stored in the legacy enterprise system. As such it is imperative, in order for the query to be adressed, to follow either the path to SIS or to the legacy database.
- The legacy schema is produced by re-engineering the databases. Several methods have been proposed in the literature regarding the issue of how to extract EER models from relational data ([5], [6], [7]). On the other hand, the SIS schema has cropped up as a result of the ERA approach.

The distinction between SIS and Legacy schema is not strict as new classes that populate the S-level, after the re-engineering, could instantiate the ERA metamodel superclasses. Thus, the right links have to be established with classes of the upper (M) level. This presumes that the user assigned to this job is immersed in the ERA methodology's principles in order to interprete accurately the Legacy schema's classes in relation to the metamodel classes.

It is also important to construct a platform that could grant access to the Legacy databases. The purpose is to consolidate a channel to the raw information via which the user can query the operational data regardless of their location. Right now there is a client server architecture implemented for the access of SIS. The idea is to extend the already available client with additional software components so that the new route to the legacy database is open for queries and updates.

Figure 1 depicts how the integration of the legacy system and SIS is accomplished. In the dotted rectangle the SIS schema and the information pertinent to the ERA metamodels are contained. The hierarchy declares the instantiation of superclasses by semantically lower level subclasses. The part that describes the entities of the entire enterprise schema and their interrelation resides on S-layer. The raw data on the other hand are not placed on the T-layer but still remain in the enterprise repositories. Their conceptual weight though is the same as information of T-layer (illustrated in Fig.1 by putting a figure of a relational table aside to the T-layer objects). Therefore, a clear separation between T-objects and relational records, that bear the same level of abstraction, is made. Moving upwards to S, we encounter the whole enterprise schema. Classes comprise instances of the ERA metamodels (M-level) but we need to distinguish between classes that are instantiated on T-layer and those that represent the equivalent to T relational enterprise entities. The latter case's objects are produced by extending the business schema following a reverse engineeing method.

The rest of figure 1 illustrates the mechanism that materializes the user-databases interaction. Queries are of the following kinds:

- View SIS or relational database data
- Update SIS or relational database data

Below the insert mechanism, we find the "Db Service interpreting" module which analyses the users' query. It encompasses the component that recognizes the query subparts that need to be redirected accordingly to the interfaces responsible to interact with the respective databases (SIS or Legacy). The interface for SIS is available and has been developed in JAVA whereas JDBC and ODBC API are adopted as the interface for the relational databases. JAVA employment advances the entire system as it benefits from advantages like platform independence, security, and robustness. The use of JDBC and ODBC standards broadens the scope of database accessibility granting independence of vendors.

Conclusions

Our approach attempts to collect the disparate Enterprise data residing in distributed locations by integrating it in the semantic repository. The database information is represented by an EER model that is extracted via reverse engineering and then introduced to SIS. As a result, a unified Enterprise schema is available to the stakeholders who are interested in analysing requirements and use CASE tools that support ERA methodology. An extension to the system that accesses SIS is provided in order to establish the links with the distributed databases.

References

[1] Loucopoulos P., Kavakli E., Louridas P., Prekas N., Tzanaki A., "The Enterprise Requirements Analysis (ERA) Approach", Information Systems Engineering Group, Dept of Computation, UMIST, Manchester, October 1996

[2] Mylopoulos J., "Conceptual Modeling and Telos" in "Conceptual Modeling, Databases and CASE" of Loucopoulos P. and Zicari R., Wiley Publications, 1992

[3] Dorr M., Klimathianakis P., Theodorakis M., " SIS- Data Entry Language User's Manual. SIB Static Analyzer and Class Management Runtime System Version 1.3", November 1995

[4] Dadouris C., Dorr M., Kizlaridou S., Prekas N., "SIS-Query Interpreter: An Interactive program to Use PQI functions", Institute of Computer Science, Foundation for Research and Technology, Crete (Greece), 1995

[5] Chiang R.H.L., Barron T.M., Stoney V.C., "Reverse Engineering of Relational Databases: Extraction of an EER Model from a relational Database", Data and Knowledge Engineering, Vol. 12, No 2, 1994

[6] Aiken P.H., "Data reverse engineering: slaying the legacy dragon", New York, McGraw-Hill, 1996

[7] Johanesson P., Kalman K., " A Method for translating relational Schemas into Conceptual Schemas", Proceedings of the 8th International Conference on E-R Approach, Toronto, 1989

[8] Keller A.M., "Migrating from Relational to Object Technology", Stanford University, URL: http://www.persistence.com/persistence/pagethree.pages/migrating.html