A Principled Approach to Data Integration and Reconciliation in Data Warehousing

Author: BALWANT RAI
Organization: Evaltech, Inc.
Evaltech Research Group,
Data Warehousing Practice.
Date: 07/01/2004
Email: erg@evaltech.com

Abstract:
In most of the data warehouse project, we had multiple, diverse sources of data, and integrating them into a single administrative Many information needs were not being met, and the integration provided by a data warehouse was seen as very desirable. This is of immediate benefit to knowledge workers who are now able to formulate their own queries and write their own ad hoc reports. It is also used as the basis for the aggregation and summarization that makes up the multi-dimensional database. Source Integration is one of the core problems in Data Warehousing.
Two critical factors for the design and maintenance of applications requiring Source Integration 1) conceptual modelling of the domain, and 2) reasoning support over the Conceptual representation.

Intellectual Property / Copyright Material
All text and graphics found in this article are the property of the Evaltech, Inc. and cannot be used or duplicated without the express written permission of the corporation through the Office of Evaltech, Inc.
A Principled Approach to Data Integration and Reconciliation in Data Warehousing

Data Integration in Data Warehousing

Abstract

In most of the data warehouse project, we had multiple, diverse sources of data, and integrating them into a single administrative suite. Many information needs were not being met, and the integration provided by a data warehouse was seen as very desirable. This is of immediate benefit to knowledge workers who are now able to formulate their own queries and write their own ad hoc reports. It is also used as the basis for the aggregation and summarization that makes up the multi-dimensional database. Source Integration is one of the core problems in Data Warehousing. Two critical factors for the design and maintenance of applications requiring Source Integration are:

1) Conceptual modelling of the domain, and
2) Reasoning support over the Conceptual representation.

1. Introduction

The most important aspect of a data warehouse is integration. In most of the project the data are from multiple sources. Source can be from db2, oracle, sql server, Informix etc. The main aspect of data warehouse is integrating these sources into a single administrative suite. When data passes from the application-oriented operational environment to the data warehouse, possible inconsistencies and redundancies should be resolved, so that the warehouse is able to provide an integrated and reconciled view of data. There are two basic approaches to the data integration problem, called procedural and declarative.

1) Procedural approach - integration of data with respect to a set of predefined information, without resorting to an explicit notion of integrated data schema.
2) Declarative approach - the goal is to model the data at the sources by means of a suitable language.

We need to construct a unified representation which is similar to the entire data source. This is to be used when querying the global information system.

To understand these approaches, we should know about integration and conceptual modelling. In this section, we will be study the introduction and the approach in designing the conceptual modelling.

Conceptual Modeling

Introduction

Designing the complex systems needed to support a number of core informatics systems problems. The issues at the heart of these problems concern development of a unified conceptual model for the research, and integration of heterogeneous parts of systems developed by different disciplines. The first needs a strategy for merging individual discipline's models into a coherent, multiple discipline model. It must preserve the insights of the individual disciplines, and create a framework for insights in the (new) merged discipline. The second must use general purpose formal models and discipline specific natural models. Integration requires general purpose models, languages, and notation. Scientific depth and insight require leveraging the notations and thought patterns natural to specific disciplines. The notations and thought patterns natural to specific disciplines produce heterogeneous representations, even when encoded in general purpose notations.
Within the software engineering strategy focused on a designer's perspective of development, there are three classes of concerns that must be addressed in creating a system. First, designers need to develop a substantial understanding of a problem to be solved, the domain in which it resides and the community that works in that domain. Second, they must design a system concept, by a creative process whose success appears to be based on insights into the central concepts and activities of the domain and its community. Third, they need to architect and realize that system. The process of developing the understanding and the insight is called system/domain analysis, the development of the system concept and the architecture of the system are called design and realization is called implementation. Intuition and insight in the analysis/design process come from discipline specific understanding and are captured using formal notations. This perspective on development has modeling as its central focus.

A number concept that has emerged in cognitive science over the past decade can help with the domain understanding and system concept design concerns. First, understanding the structure and conceptual metaphor basis of human cognitive models provides a framework for understanding and design. Models, on which understanding and design are based, are captured in a more natural way. Second, there is a cognitive process for developing new meaning for concepts, when they are placed in different contexts. In the new analysis of this process, conceptual model "blends" provide a basis for developing new meaning. An analog of this can support cross discipline and multidiscipline envisioning, which these type of systems need to exhibit. Third, explanation and understanding are related, with explanation reifying an understanding, while from the other direction, explanation develops understanding. This interplay of explanation and understanding provides leveraging during analysis. And fourth, the categorization concept of radial categories leads to a useful characterization of "natural" depictions used in explanations. This characterization places constraints on the amount and type of abstraction used in models derived from those explanations, providing guidelines for analysis.

The interoperation of heterogeneous data types requires transformation of data from its original representation to a standard representation (in a data warehouse architecture) or to a usage representation (in a federated architecture). Valid integration, however, depends on the expertise of scientific curators, understanding the source(s), and scientists who design and perform virtual experiments and analyses with the resulting merged information.

**Central Ideas**

Models underpin system design, perceptual interfaces, and programming interfaces to data sources, analysis programs and other subsystems. The idea here is that development of a unified conceptual model for the research be based on analysis of explanations and accompanying natural depictions. Their capture and analysis is based on analogy to the development of "natural" cognitive models and the blending of models, by an individual, during creative insight.

To address architecture level concerns, general purpose formal and natural modeling languages must be combined. General purpose modeling languages and notations are needed as a basis for automation, but because they are general purpose, they must abstract away any discipline specific intuition and insight. General purpose modeling languages and notations are also formal. But, Scientists understand and explain concepts and issues in their field using notation and language natural to their discipline. Depth and insight require the notations and thought patterns natural to specific disciplines. The models emerging from their explanations and depictions need be combined with general purpose modeling languages and notations, such as XML and UML. The general purpose, formal models are structured by the natural notations for more valid models. The natural notation models are integrated into the general purpose notation models, to retain the discipline's insight into the domain.

The aim of this effort is to develop a mechanism and a methodology for integration of separate discipline's models, based on distilling the inherent structure of each model, blending them to
create the structure for the integrated domain and creating views of this blended structure for each participating discipline. This paper results from work on a number of life sciences projects, taking a systems’ biology approach, which is naturally interdisciplinary, in areas from genomics to ecology. It looks at this type of system from a point of view which uses component based architectures, providing data integration through interfaces. The focus of this work is in integrating ideas about cognitive models and model blending from cognitive science, into a model based

**Approach**

Designers need to develop a substantial understanding of a problem to be solved, the domain in which it resides and the community that works in that domain. They must also develop a system concept, by a creative process whose success appears to be based on insights into the central concepts and activities of the domain and its community, developed while building that understanding. Only after these do they need to architect and realize that system.

The approach is based on using results from cognitive science research, within a framework provided by software engineering, to provide a basis for system design, by capturing the models on which the design is based in a more natural way. This is done by explicit analysis of the natural models of each discipline, capturing the essence of each in a formal model which is then used for system design. The structure and conceptual metaphors used in explanations of the discipline's concepts are captured and analyzed for use in formal models in the system design.

Combine the use of formal and natural models in developing a cross-discipline or multidiscipline understanding of particular domains. This is based on an analysis of the cognitive process which develops new meaning for a concept when placed in a different context, using blended cognitive models and metaphorical mappings. The concept of radial categories characterizing "natural" depictions is used in analyzing explanations, and developing the abstraction used in models derived from those explanations. "Natural" semantic support for creative insight in multidiscipline research is provided by the emergent structure of these blended cognitive models and metaphorical mappings of meanings in the user's discipline. The models developed are used to design the system, the perceptual interfaces provided by systems, and the abstract interfaces to data sources, analysis programs and other subsystems.

**Criteria**

The criteria by which this work is judged address three areas:

1. The models and interfaces to software components must be valid with respect to scientific experiments. The models must accurately reflect the concepts used in the design of experiments. The computer data must reflect the results of the scientific experiments by the disciplines creating the data. The interfaces, both program and perceptual, must portray the data, and inferences from it, in accordance with the models of the disciplines.
2. The models and interfaces must structurally, semantically, and pragmatically conform to each discipline's conceptual models and the blended models. The models and interfaces must address the thought patterns and activities of each discipline.
3. The models and interfaces to software components must resonate with the intuitions and insights of each discipline and support development.

we adopt a declarative approach to integration, and argue that two critical factors for the design and maintenance of applications requiring Information Integration, and in particular integration in Data Warehousing, are the conceptual modelling of the domain, and the possibility
of reasoning over the conceptual representation. The approach we propose is based on building a conceptual representation of both the information sources and the Data Warehouse. An important aspect of the conceptual representation is the explicit specification of the set of interdependencies between objects in the sources and objects in the Data Warehouse. Thus, integration is seen as the process of understanding and representing the relationships between data in the sources and in the Data Warehouse, possibly with some reconciling actions, rather than producing a unified data schema. Moreover, suitable reasoning techniques associated with the conceptual formalism are used to support the designer during the resulting specification process. Specifically, our work provides the following main contributions:

1. We use special class-based logical formalisms, called Description Logics, for the conceptual modelling of both the global domain and the various sources. Since the development of successful Information Integration solutions requires specific modelling features, we propose a new Description Logic, which treats nary relations as first-class citizens. Note that the usual characteristic of many Description Logics to model only unary predicates (concepts) and binary predicates (roles) would represent an intolerable limit in our case.

2. We provide suitable mechanisms for expressing what we call the inter model assertions, i.e. inter-relationships between concepts in different sources. Thus, integration is seen as the incremental process of understanding and representing the relationships between data in the sources, rather than simply producing a unified data schema. The fact that our approach is incremental is also important in amortizing the cost of integration.

3. For an accurate description of the information sources, we incorporate in our logic the possibility of describing the data at the sources in terms of a set of relational structures. Each relational structure is defined as a view over the conceptual representation, thus providing a formal mapping between the description of data and the conceptual representation of the domain.

4. We provide inference procedures for the fundamental reasoning services, namely concept and relation subsumption, and query containment. Indeed, we make use of the first decidability result on query containment for a Description Logic with nary relations. Based on these reasoning methods, we present a methodological framework for Information Integration, which can be applied both in the virtual and in the materialized approach.

2. The Framework
In this approach to Source Integration, three layers can be identified:
1) a conceptual layer, constituted by the Domain Model, including an Enterprise Model and one Source Model for each data source, which provides a conceptual representation of both the information sources and the Data Warehouse;
2) a logical layer constituted by the Source Schemas and the Materialized View Schema, which describes the logical content of source data stores and of the materialized view store, respectively;
3) a physical layer, which consists of the data stores containing the actual data of the sources and the integrated materialized views.

The designer is provided with information on various aspects, including the global concepts relevant for new information requirements, the sources from which a new view can be defined, the correspondences between sources and/or views, and a trace of the integration steps. We describe now the structure of the conceptual and logical layers, which constitute the core of the proposed integration framework. The actual formalism we adopt and the associated reasoning techniques are described in the next section.
Conceptual Layer

The Enterprise Model is a conceptual representation of the global concepts and relationships that are of interest to the Data Warehouse application. It provides a consolidated view of the concepts and relationships that are important to the enterprise, and have been currently analyzed. Such a view is subject to changes and additions as the analysis of the information sources proceeds. The Description Logic formalism we use is general and powerful enough to express the usual database models, such as the Entity-Relationship Model, and the Relational Model. Moreover, there are suitable inferences techniques associated with the formalism, which allow for carrying out several reasoning services on the representation. The formalism is hidden from the user of the DWQ tools who only uses a graphical interface.

For a given information source S, the Source Model of S is a conceptual representation of the data residing in S. Again, our approach does not require a source to be fully analyzed and conceptualized. Source Models are expressed by means of the same formalism used for the Enterprise Model.

The notion of interdependency is a central one in our approach. Since the sources are of interest in the overall architecture, integration does not simply mean producing the Enterprise Model, but rather to be able to establish the correct relationships both between the Source Models and the Enterprise Model, and between the various Source Models. We formalize the notion of interdependency by means of inter model assertions. An inter model assertion states that one object (i.e., class, entity, or relation) belonging to a certain Model (either the Enterprise or a Source Model) is always a subset of an object belonging to another Model. This simple declarative mechanism has been showed to be extremely effective in establishing relationships among different database schemas. We use again a logic-based formalism to express inter model assertions, and the associated inference techniques provide a means to reason about inter dependencies among models.

The logical content of each source S, called the Source Schema, is provided in terms of a set of definitions of relations, each one expressed in terms of a query over the Source Model of S. The logical content of a source represents the structure of data expressed in terms of a logical data model, in the demo the Relational Model. In our framework, such a structure is provided by specifying the connection with the conceptual representation of the source. In other words, the logical content of a source S, or of a portion thereof, is described in terms of a view over the Source Model associated with S (and, therefore, of the Conceptual Data Warehouse Model).

Logical Layer

Our approach requires that each source, besides being conceptualized, is also described in the Source Schema in terms of a logical data model (in our case the Relational Model) which allows for representing the structure of the stored data. Such a structure is specified in terms of a set of relation definitions, each one ex-pressed by means of a view (i.e. a query) over the conceptual representation of the source (i.e. the Source Model). Suitable software components, called wrappers, implement the mapping of physical structures to logical structures.

The Data Warehouse Schema provides a description of the logical content of the materialized views constituting the Data Warehouse. Similarly to the case of the sources, each portion of the Data Warehouse Schema is described in terms of a view over the Conceptual Data Warehouse Model. How a view is actually materialized starting from the data in the sources is specified by means of suitable mechanisms, called mediators.
A Principled Approach to Data Integration and Reconciliation in Data Warehousing

in terms of a query over the Domain Model. A view is actually materialized starting from the data in the sources by means of suitable software components, called mediators

3. The Methodology
The methodology deals with two scenarios,
1) source-driven integration
2) client-driven integration

3.1. Source-Driven Integration
Source-driven integration is triggered when a new source or a new portion of a source is taken into account for integration. The steps to be accomplished in this case are:
1. Source Model construction. The Source Model is constructed capturing the concepts and the relationships of the new source that are critical for the enterprise.
2. Source Model integration. This is done after the construction of source model. This source model is then integrated with the domain model. This integration can lead to changes both to the Source Models, and to the Enterprise Model. The novel part of the methodology is the specification of inter model assertions and the derivation of implicit relationships by exploiting the reasoning techniques. Assertions relating elements in one Source Model and in different Source Models are of importance.
3. Quality analysis. The Quality Factors of the resulting Domain Model are evaluated and a restructuring is accomplished to match the required criteria. This step requires the use of the reasoning techniques associated with our formalisms to check for quality factors such as consistency, redundancy, readability, accessibility, believability.
4. Source Schema construction. The logical view of the new source or a new portion of the source is produced. It is expressed as a collection of queries over the corresponding Source Model. The source schemas are used in order to determine the sources relevant for computing answers to queries, by exploiting the ability to reason about queries.
5. Data Warehouse Schema restructuring. On the basis of the new source, an analysis is carried out on whether the Data Warehouse Schema should be restructured and/or modified in order to better meet quality requirements. Again, the schema is constituted by a set of queries over the Domain Model, and for its restructuring the use of reasoning techniques is crucial. A restructuring of the Data Warehouse Schema may require the design of new mediators.

4.2. Client-Driven Integration
The client-driven design strategy refers to the case when a new query (or a set of queries) posed by a client is considered. The reasoning facilities are exploited to analyse and systematically decompose the query and check whether its components are subsumed by the views defined in the various schemas. The analysis is carried out as follows:
1. By exploiting query containment checking, we verify if and how the answer can be computed from the materialized views stored in the Data Warehouse.
2. In the case where the materialized information is not sufficient, we verify if the answer can be obtained by materializing new concepts represented in the Domain Model. In this case, query containment helps to identify the set of sub queries to be issued on the sources and to extend and/or restructure the Data Warehouse Schema. Different choices can be identified, based on various preference criteria which take into account the above mentioned quality factors.
3. In the case where neither the materialized data nor the concepts in the Domain Model are sufficient, the necessary data should be searched for in new sources, or in new portions of already analysed sources. The new (portions of the) sources are then added to the Domain Model using the source-driven approach, and the process of analysing the query

6 Conclusions
We have described a new approach to data integration and reconciliation in Data Warehousing. The approach is based on the availability of a Conceptual Model of the corporate data, and allows the designer to declaratively specify several types of correspondences between data in different sources. Such correspondences are used by a query rewriting algorithm that supports the task of specifying the correct mediators for the loading of the materialized views of the Data Warehouse. Based on the described methodology, we are currently implementing a design tool within the DWQ project. The tool is based on the Concept Base System, and provides support for both schema and data integration in Data Warehousing.