The growth of corporate databases has led to ever-larger corporate data models. While an enterprise data model for a medium-sized manufacturing company may contain more than 300 entities (Scheer, 1994), the data model of the Petrochemical Open Software Foundation (1994) has more than 800 entities and the data model underlying the SAP R/3 standard software contains more than 2000 entities. Large modeling projects suffer from a weakness of current modeling methodologies: there is a lack of effective methods and tools for model validation and integration (Bubenko & Wangler, 1992). One of the core problems is the meaning associated with the elements of a data model (Loucopoulos, 1992); it is rarely made explicit. Instead, large modeling projects depend on highly skilled data administrators to interpret and evaluate models. They also need to spend much time for defining and enforcing naming conventions which aim to reduce semantic ambiguity (Department of Defense, 1993). To overcome these weaknesses, we present an approach which uses natural language processing to leverage meaning for the integration and validation of data models. We also present a prototype tool which proves that our approach is feasible with today’s technology.

Although data modeling has been an area of intensive research, there is a lack of operational procedures for measuring model quality and for integrating large-scale models. Progress has been limited because the meaning associated with the individual elements of a model needs to be taken into account. This meaning rarely is made explicit and therefore not directly available for validation and integration procedures. In this paper we show that natural language processing techniques based on a custom-built dictionary can be used to interpret data models and leverage meaning for validation and integration procedures. We describe a general-purpose dictionary which contains syntactic and semantic word categories for 23,000 English words. We show how the syntactic and semantic information in the dictionary can be used to detect semantic inconsistencies, reject inconsistent naming, identify unregistered abbreviations and detect synonym candidates. To prove feasibility of our approach, we describe a prototype tool which has been implemented on a standard personal computer.

The growth of corporate databases has led to ever-larger corporate data models. While an enterprise data model for a medium-sized manufacturing company may contain more than 300 entities (Scheer, 1994), the data model of the Petrochemical Open Software Foundation (1994) has more than 800 entities and the data model underlying the SAP R/3 standard software contains more than 2000 entities. Large modeling projects suffer from a weakness of current modeling methodologies: there is a lack of effective methods and tools for model validation and integration (Bubenko & Wangler, 1992). One of the core problems is the meaning associated with the elements of a data model (Loucopoulos, 1992); it is rarely made explicit. Instead, large modeling projects depend on highly skilled data administrators to interpret and evaluate models. They also need to spend much time for defining and enforcing naming conventions which aim to reduce semantic ambiguity (Department of Defense, 1993). To overcome these weaknesses, we present an approach which uses natural language processing to leverage meaning for the integration and validation of data models. We also present a prototype tool which proves that our approach is feasible with today’s technology.

As Figure 1 shows, the problems of validation and integration consist of several sub-problems which have received different amounts of attention in previous research. The goal of model validation is to ensure that a model adequately represents the underlying domain and is consistent with the purpose of the model (Bubenko & Wangler, 1992; Blum, 1994). Model validation can be broken down into three components: The purpose of verification (internal validation) is to determine whether the model taken by itself is without detectable flaws and whether it adheres to the syntactic and diagrammatic rules of the modeling methodology. External validation is concerned with the context of a model. Its goal is to determine whether a schema is an adequate representation of the domain of interest which meets the needs of the user.
Indirect validation is an often-neglected third approach, which derives insights about the quality of a model by evaluating the modeling process and the modeling capabilities of the authors rather than the model itself.

Much progress has been made in verification with the exception of naming analysis, which is addressed below. Many methods have been developed that test a schema for syntactical correctness, for minimality or for other structural properties (Batini, Ceri & Navathe, 1992, 140-143). Syntax-level error detection algorithms check, for example, whether any relationship in a schema is linked to less than two entities or whether a generalization hierarchy includes cycles (Hars 1994, pp.207-220). These algorithms are well understood and have been incorporated in many commercial modeling tools (Bubenko and Wangler 1992). Much less progress has been made in external validation (Blum, 1991). Techniques that determine whether a schema is an adequate representation of the domain of interest primarily rely on expert review. Another approach for external validation is the comparison of the model with other sources of information. A model, for example, can be compared with a reference model for the domain. Forms or other written documents from the domain may be used to check whether the information relevant in that form has been accurately represented in the model. A third approach is the development of a prototype and the solicitation of feedback from users based on the prototype (Loucopoulos, 1992). Much of the problem of external validation, however, is a semantic problem: are the entities, relationships and attributes adequate representations of domain of interest? So far, little progress has been made in this issue.

Similar problems have been encountered for the integration of data models. The integration of data across diverse databases and applications is one of the success factors for large information systems projects. It is a prerequisite for coordinating diverse units of an organization (Goodhue, Wybo & Kirsch 1992). It ensures that data have the same meaning and use across time and across users (Martin 1986). To achieve data integration, the conceptual schemas of all databases need to be merged into a single conceptual schema. Many researchers have worked on this issue and much progress has been made in the Eighties as the overviews given in Wagner (1989) and Batini, Lenzerini & Navathe (1986) show. The integration problem has been decomposed into the problems of identifying similarities between different schemas, detecting and removing inconsistencies, merging models, and completing possible information gaps in the integrated schema through additional data modeling (Batini & Lenzerini 1983).

Although much progress has been made in the removal of inconsistencies, the detection of similarities still poses many difficulties (Kamel, 1995). It has become clear that the detection of similarities requires knowledge about the concepts involved that is not made explicit in the structures of a schema (Creasy & Ellis, 1993). Currently effective domain-independent algorithms for detecting synonyms are not available. As a result, today’s modeling tools only provide rudimentary schema integration functionalities. In the absence of tool support for synonym detection, the main approach for avoiding the proliferation of unrecognized synonyms in medium-sized and large development projects is the costly definition and administration of detailed naming conventions.

A key reason why current integration and validation approaches are not effective is the high emphasis on structure in current modeling methodologies (Loucopoulos, 1992) and the disregard of content. Although Entity-Relationship (ER) Models often claim to be “semantic data models,” neither methods nor current tools explicitly evaluate or act on meaning.

Figure 2 illustrates this problem. All three data models
Related Content

Mark up Languages and Electronic Commerce
www.igi-global.com/chapter/markup-languages-electronic-commerce/30271?camid=4v1a

Designing Document SQL (DSQL): An Accessible yet Comprehensive Ad-Hoc Querying Frontend for Query
www.igi-global.com/article/designing-document-sql-dsql/37211?camid=4v1a

Predicting Software Abnormal State by using Classification Algorithm
www.igi-global.com/article/predicting-software-abnormal-state-by-using-classification-algorithm/165162?camid=4v1a

The Quality of Data Representations Developed by NonExpert Designers: An Experimental Study
www.igi-global.com/article/quality-data-representations-developed-nonexpert/51125?camid=4v1a