An Effective Database Model for System & Information Integration

Ashwin Perti, Shivank Sharma, Shivank Garg, Vritul Kumar, Yash Kumar

1Associate Professor, Project Guide
2,3,4,5 Students of ABES Engineering College, Ghaziabad

ABSTRACT

Most organizations are moving away from point-to-point integration in a bid to be more agile and to reduce integration costs. Whether at the stage of using simple messages to talk to web services or a fully governed SOA approach, the key to achieving these goals of agility and reuse is to solve the challenge of consistent data. This paper proposes Canonical Model to be established and adopted, ideally at an enterprise level or at the very least at a domain level to manage diversity of data sources without point-to-point integration.

Keywords – Canonical Model, heterogeneity, data sources, data integration

1. Introduction

The development of internet and rise of the information age has made data handling the most fundamental of all organization processes. With more and more information sources available, managing and accessing all of these via a unified and robust interface has led to extensive research in the field of data integration. The data may not be of the same data type, format and detail, which creates the need for integrating the data into a common format before it can be successfully accessed by user via any DBMS tool.

1.1 Data Integration as a concept

Data integration involves combining data from different heterogeneous sources and providing users with a unified view of this data[1]. This process becomes significant in a variety of situations, which include both commercial (when two similar companies need to merge their databases) and scientific (combining research results from different bioinformatics repositories, for example) domains. Need for data integration appears with increasing frequency as the volume and the emerging trend to share existing data explodes. This is one of the biggest concerns of industries dealing with voluminous data and is one of the most researched problems amongst database pundits.

For example, consider a situation where an organization acquires another and inherits an entire IT department that may not be compatible with its own. In order to succeed and stay competitive, the organization should employ processes that consume less efforts and resources and make data system and information interoperable among all of its subsidiaries.

Figure 6: Inputs for reliable data generation.

In today’s world, volumes of data grow exponentially in all realms from personal data to enterprise and global data. Thus, it is extremely important to be able to understand data sets and organize them. Advanced studies in data integration, migration, synchronization, business intelligence etc. help minimize these problems.
2. Data Integration

Data integration aims to provide a uniform integrated access to multiple heterogeneous information sources designed independently and having strictly related contents. However, the integrated view, constructed by integrating the information provided by the different data sources by means of a specified integration strategy could potentially contain inconsistent data; that is, it can violate some of the constraints defined on the data. [2]

![Diagram of data integration process]

**Figure 2:** Steps involved in data integration.[1]

2.1 Data Transformation

Heterogeneous data sources implies heterogeneous schemata. In order to achieve unified information, schematic heterogeneities must be bridged. Data from the data sources is transformed to conform to the global schema of the unified data. Schema mapping is used to identify and handle the schematic diversity by assuming a given target schema and transforming the input data source schema accordingly.

2.2 Duplicate Detection

The next step of the process is duplicate detection which aims to identify and remove the multiple representation of the same real-world object. The process involves establishing a comparative measure for similarity detection. The two main challenges are effectiveness and efficiency.

As the data sources are different, unique identifiers may vary. A complete set of attributes which together can uniquely identify an object must be set as a threshold. Efficiency is challenged at this point as the data set are often too large so a large attribute set causes computation delay.

2.3 Completeness vs Conciseness

Heterogeneous data integration further raises two challenges. A complete data integration may include a comprehensive view but may also contain anomalies in the form of missing data. As the data sources are different, not all the sources may have the same extent of data collection of their objects.

Selecting only the common attribute sets will present a concise view but may omit integral information and may present an inconsistent output.[1]

3. Canonical Data Model

Canonical Model is a design pattern used to communicate between different data formats. As a form of Enterprise Application Integration, it is intended to reduce costs and standardize on agreed data definitions associated with integrating business systems. A Canonical Model is any model that is canonical in nature, i.e. a model which is in the simplest form possible based on a standard, common view within a given context.

3.1 What is Canonical?

Creating your “canonical” view of what an entity looks like and implementing it within your middleware can help insulate you from changes in the future. If you build an orchestration which directly uses an external schema, you are more susceptible to changes to that schema. If your orchestration handles a canonical message however, you may only have to update the transformation and continue to publish the canonical message to the MessageBox without further changes to any subscribing orchestrations. This really shows its worth if you have multiple subscribers to a single published message as you may only have a single change to make.

Fig. 3 describes the structure of a canonical model where the ‘canonical layer’ acts as the intermediary system between the database and the user.
3.2 Goal of Canonical Data Model
The goal of Canonical Model is to provide a dictionary of reusable common objects and definitions at an enterprise or business-domain level to enhance system interoperability. A Canonical Data Model allows developers and business users to discuss the integration solution in terms of the company's business domain, not a specific package implementation. For example, packaged applications may represent the common concept of a customer in many different internal formats, such as 'account', 'payer', and 'contact'. Defining a Canonical Data Model is often the first step to resolving cases of semantic dissonance between applications.
Canonical schemas also help developers understand business entities and provide consistency across solutions and integrations as they offer the opportunity for reuse. If you’ve defined what your company considers to be an “Invoice”, the same schema applies to all integration solutions. They can use meaningful names for elements and attributes rather than system or domain specific ones which may be alien to your organization or someone not used to the sending platform (SAP schemas are a great example!). Referencing plays an important part also, and canonical schemas can reference other ones (a Canonical Invoice schema might reference a Canonical Address schema for example).
A canonical data model is much like any other data model in a DBMS tool. Only the way you design it is potentially different because you may have to take account of many different data consumers and data providers. It can be used as a standard to translate data into a common format suitable for loading into a Data Warehouse. It allows us to show Many-to-Many Relationships in a very concise and economical style.

4. Illustration
Consider two distinct databases representing inventories of albums and books respectively. The databases are schematically unique from each other but are comparable as they carry about the same tasks such as managing multiple suppliers and storage locations of products on sale.
The database design involves three tables, Albums, Wholeseller and storage. The intent behind the design is to achieve basic functional behavior of a typical inventory management such as the storage location, supplier information and keeping a track of the number of items of each product available. The database diagram may not represent a typical design used in the industry as it has only been constructed for demonstrational purpose.

Figure 5: Database diagram for organization 2 which deals in books.

As it is evidently clear that the second database has an extra table which is used to merge map the information in the products, supplier and storage tables (descriptive names). The design does not permit the two databases to be mutually complementary to a common business logic until one or both of them are altered. The process of redesigning can be cumbersome and even be high on cost. But if an alternative be proposed to handle the distinctiveness of the two databases, both effort and cost may be saved.

The process involves designing a middle layer to handle the specific databases. A simplistic design is described in Fig. 6.
The table describes a part of what forms a base on which the canonical layer can be constructed. The columns represent the tables in the specific databases, prefixed by ‘AttributeIn’ which can be used as an identifier. The second column defines the generalized names of the comparable specific attribute entities. The specific definition is given in the respective columns under the tables, according to their occurrence. The nomenclature used is 
<db_name>.dbo.<table_name>~column_name, where ‘dbo’ is in accordance with MS SQL Server 2012 and ‘~’ is a separator.
This system ensures that the business logic requires only the general entity recognition and the specific reference can be dynamically loaded in the canonical layer. In other words, single business logic can cater to multiple repositories.

5. Conclusion
The paper proposes an alternate solution to conventional system integration approach where either one of the integration system is used as a focal point or a new design is used to represent a unified model. A very basic canonical design may involve the metadata mapping of the specific databases which efficiently define the relationship between the underlying databases.
The system is a direct implementation of a mediated query system [3]. It is very similar to a meta-search system but the only difference is that the underlying databases has to be heterogeneous.
Despite growing number of advantages, canonical data model characteristics are not implemented while designing databases for organizations, as ease of integration upon mergers and acquisitions is a priority for the organization although it does leads to serious consequences.
The model works only in the situation where the underlying systems are comparable and will only add to the complexity of data integration if it is used on other scenarios. It requires an up to date mapping in the layer. If the mapping is inefficient, the model fails. This requires knowledge of data dictionaries and metadata handling which implies an expert capabilities of database management.

References