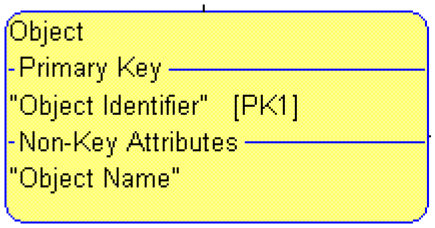
Object Relational Data Storage for Metadata

This document makes an effort to describe a common storage structure for metadata. In general, metadata is represented as having three parts: the existence of the metadata itself (as an object), that object's attributes and the relationship of this object to its attributes and other objects. In this paper, a database schema is discussed in order to accommodate this simple and loose structure. The benefit of this type of design is simple data storage structure. The drawbacks of this type of design are: complex queries that change per the context of the retrieval of metadata and loose control over content structure. Both benefits and drawbacks are discussed as well as an example implementation.

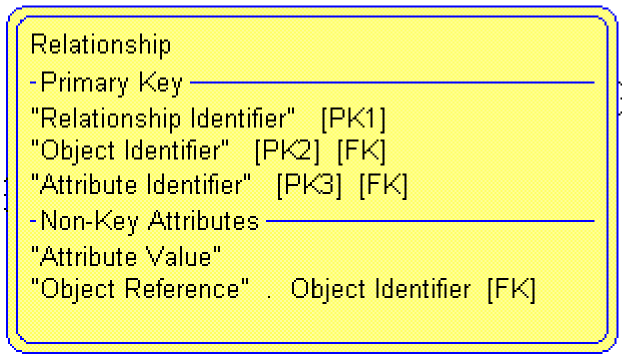
Conceptual Design

As mentioned, the storage of metadata being presented is of three parts: the objects of metadata, the parts that describe the metadata and the relationships of objects to attributes and other objects. This can be conceptually realized as three data entities: objects, attributes, and relationships.

The object entity is identified by an object identifier unique to each object. Each object also carries with it an alias, or unique name to identify that object. The attribute entity is described the same way, only using an attribute identifier and name.



The relationship entity is an associative entity building links between the object and attribute entity, as well as links between the object entity and itself. Each relationship is identified by a unique identifier. Each relationship will also have two foreign key identifiers, one each for the object in the relationship and the attribute of the relationship. Each relationship must be composed of an object and an attribute.


Relationship
-Primary Key
"Relationship Identifier" [PK1]
"Object Identifier" [PK2] [FK]
"Attribute Identifier" [PK3] [FK]
-Non-Key Attributes
"Attribute Value"
"Object Reference" . Object Identifier [FK]

Relationships can be of two types: a reference relationship or a value relationship. A reference relationship means that the meaning of the attribute is expressed via another object, and so the reference points to another object. It is possible to have an object reference itself if it is meaningful to do so. A value relationship is on that the attribute being described simply contains a string value, or text.

Benefits

Since the storage structure of data is left to the methods that store it, the data model is left simple and has no context. The context of the stored data is realized via the retrial method and the data's existence in the context of its use and not in the storage engine.

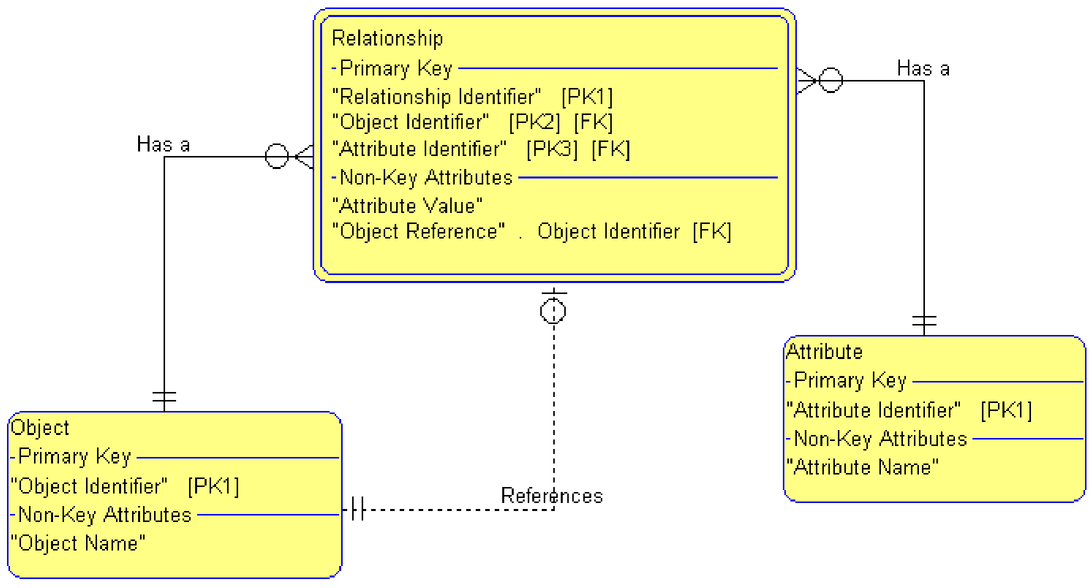
The schema also provides the ability to accommodate varying logical data model schemas to be stored in a flattened perspective. This way, the data is stored in a meaningful and descriptive way specific to the users of that information.

Drawbacks

The simple storage method requires special queries to extract and insert data into the storage engine. Since the data stored meaningful to the source of the data, retrieval methods will either need to be overly generic or very specific to an area. These mechanisms can be provided via an API.

Another drawback is the loose constraints of the storage of data. Since data can be described any way possible, it is likely that duplicate information will exist, as well as badly defined objects, attributes and relationships.

Entity Relation Diagram


Relationship
-Primary Key
"Relationship Identifier" [PK1]
"Object Identifier" [PK2] [FK]
"Attribute Identifier" [PK3] [FK]
-Non-Key Attributes
"Attribute Value"
"Object Reference" . Object Identifier [FK]
Has a
Has a
Object
-Primary Key
"Object Identifier" [PK1]
-Non-Key Attributes
"Object Name"
References
Attribute
-Primary Key
"Attribute Identifier" [PK1]
-Non-Key Attributes
"Attribute Name"

Metawarehouse (Entity Relation Subject Area) System Architect Tue Apr 03, 2007 10:17
Comment
Conceptual ER of a metadata warehouse schema.

Enterprise Example of Concepts

The current enterprise data model is an example of how data can be stored in the metadata repository. The model provides the scheme to store information relative to the enterprise. For example, the enterprise model contains entities that are conceptual entities. A conceptual entity is a type of entity, and so the concept of a conceptual entity and members of that type can all be stored as metadata. Specifically we can store an object in the metadata as “Entity”. We can create an attribute in the metadata as “Type”. We now have to relate the “Type” attribute to the “Entity” object via the relationship. The object of the relationship is the “Entity” and the attribute of the relationship is “Type”. There are two options that reflect the ability to handle the different types of data. The relationship can either be a reference or a value. The different kinds of “Types” to be stored can be as objects themselves, or as simply a value. If as a value, the value of the relationship is simply set as “Conceptual”. If as a reference, then a new object is created as “Conceptual”, and the relationship references the “Conceptual” object.

Summary

These ideas expressed are not new or innovative; this concept is used in both the Ab Initio EME and System Architect Repository. This concept is also a standard metadata concept. It provides context free storage of information, and is not limited to any type of metadata. While this increases the complexity of understanding the information out of context, it does provide a central location for all information.