Logical Data Model Shared

The purpose of this document is to show how I use one logical meta-model to support four business model platforms: the Function; Enterprise; Team and Object platforms. This document will try to address the many uses of this one multimedia dictionary across these business model platforms.

Business Model Platforms

Each business model platform provides a unique view a business that I use in my work as a business modeler and decision support analyst. I am an accountant who is an expert in the information requirements for business decisions. I use databases to convert reports created by functional systems into decision support information. In doing that, I use each of these business model platforms; and my logical meta-model dictionary is the tool I use to control my process.

Functional Databases

The overall view of functional databases would be a continuous network of workflow. Functional databases are process (input output) oriented databases supporting current operations of a business including current planning and reporting. They are workflow oriented. With them you can understand workflow by single-threading any cost or decision from start to finish.

For efficiency, these functional databases share information, new customers or suppliers or employees discovered by one function are then automatically available to other functions that come into contact with those objects. Some people call this sharing an enterprise database; but even though data is shared, the basic architecture is still a network of functions.

The logical meta-model supports functions by supplying technology independent information architecture (system requirements) for that whole workflow network. It shows what goes in and what comes out each system in the network. With this meta-model you can plug-and-play new systems with simple data conversions.

Enterprise Databases

Relationships characterize the enterprise databases. Business relationships are the basic architecture of these enterprise databases. The meta-model defines these relationships.

Enterprise databases are organized around the roles objects play in the business. These roles or relationships are what keeps order and protects the meaning of the information. This database includes all the internal and external reports decision makers use to make decisions. Business relationships are used to keep the context of those reports and to combine report information without combining report context.

The enterprise database is broad and shallow. It is broad because it addresses all the information collected from all the reports. It is shallow because there are just two levels: the strategic entity and its strategic class entity.

This enterprise platform is where we store the reports and relationships that produce calculations for a specific decision. We can support those decisions with queries to the enterprise database, but a better way is to use the Team Database to define the problem.

Team Databases

Team Databases are tall and skinny. They use the information from the functional and enterprise databases. The functional category schemes are already available for use in building strategies for each team or each purpose or defining each problem.

They are skinny because they focus on a specific team or purpose.

They are tall because they take advantage of association type or many-to-many filtering of the classifications defined by the functions and their reports.

The logical meta-model is the key to this filtering. This advanced filtering is very important because the decision maker fine tunes his decision by choosing just the right level of information from each report in making his decision. He also chooses just the right levels for his rates and factors. With the meta-model in the Team Database, that fine tuning is a filtering process of defining the problem by selecting filters from a drop down list of object roles previously defined.

Object Databases

Object databases are object oriented. Every name we create in the other databases is automatically an object and a class of an object. For every object we have collected role information that is available to that object. The object oriented database is an object view of all the information we have collected. For example if an object has been used by several functions, you will see object information from each function attached to that object.

Object databases are most useful when we interact with objects. For example; when we schedule meetings we want to bring together people objects with facility objects, equipment objects and information objects. You want just what you need when you need it. The logical meta-model multimedia dictionary gives you the ability to define and schedule your meeting. You use the meta-model to pull into your meeting the specific objects you need.

Logical Data Model Sharing Summary

The purpose of this document is to show how I use one logical meta-model to support four database platforms. This document addresses the many uses of this multimedia dictionary across these platforms. It supports the functional platform by supplying technology independent system requirements for the whole workflow network. For the enterprise platform, it is the key to pulling cross-functional information together. For the Team Database Platform, it is the key to planning and fine tuning for all kinds of decision making. Finally for the Object Platform, it provides the multi-media lists for planning and scheduling objects. In summary, one logical data model supports all these big picture views of a business model, and it works for all business.