Logical Data Model Used

The purpose of this document is to show how I use one logical meta-model to view information in a database. In other words, this document will show how I use the meta-model to design database queries. I will describe how I use the logical-meta model as a top-down strategic thinking tool when building queries of information from more than one report.

Report Language Review

In the document "Logical Data Model Replaced" I discussed the idea that each report we bring into the enterprise database can be thought of as a separate context or language. I said words have specific meaning within that report. It is like there is a language surrounding each function. I said there is also a general business language that helps people communicate across functions. I said that to completely communicate that meaning across functions, language context has to be preserved. We preserve the full reports and the full general business model that integrates those reports.

Report Storage

When we bring a report into the database, we store it in tables with keys that are one to one with the meta-model. That connects us to the general business context, plus we also keep the report context with keys that are one to one with line items on the report. Probably the most important thing we do is associate the report with its period or periods.

Report Context

That report line item will have an automatically generated key and all the header information needed to reference the report. That line item will also contain all the classification schemes on the report.

Period Context

We also add a period key. Periods are not dates. History is recorded by periods, not dates. We are recording history and this is a report, not a transaction. Like a navigator, we are reporting position, change in position, activity and direction. There are many estimates involved in change of position, so the period has to be long enough for those estimates to be meaningful. In accounting we call this idea "Periodicity". In accounting we use journal transactions as our input and we produce reports as our output. Here we are using reports (from accounting and all other functions) as our input and we output queries. Many very expensive systems fail to produce desired results and speed by failing to record the period context. Periods, not dates are the correct index to history. Periods are strategic entities, not attributes.

Meta-model Context

The TDC physical database is designed to collect information that will support all business decision making. It is designed to support a model of business science. That business science model was built from an outline of a management accounting, and it includes all the information requirements for business decisions. Like most data models, it is rather stable over time. Even though the techniques of business science are continually changing, the data model has remained very stable.

We have tested it at The Doctors Clinic and over ten years it has worked very well, producing: performance measurement, capital budgeting, operational budget forecasts, and fee scheduling etc. According to the experts, it does things the $6M systems cannot do. They are now saying the problem with the $6M systems is in the data model. They say the data model needs to be expanded. Right now I say the only place in the world with the right physical data model is The Doctors Clinic. But, when I create queries, I use the logical meta-model as my thinking tool. In other words, I think in general business language, not TDC language.

Logical Data Model Used to Think in General Language

I think in the general language. I use the meta-model context to create the queries. I have no control over the many context’s we collect information, but once I have that information associated with the general context, I can use that language to produce my queries. For example: when I am responding to a survey request and their peculiar way of organizing specialties, all I have to do is associate their classification scheme with ours. Once that is done, I am free to think in general business to respond to their request or to copy a previous query and just use the new classification scheme. In creating original query, I used the general language. In many cases, the general language tells me what to look for in the reports I have already collected. I think our reports cover all our functions. With our system we can look back four years at relationships we never considered looking at until today. It is not a new business decision, just new to us. This is what I mean by thinking in the general language.

Top-Down Cross Functional View – The Big Picture

The purpose of this document is to show how I use one logical meta-model to view information in a database. In other words, this document will show how I use the meta-model to design database queries. Thinking in the general language means thinking strategically. It means taking a top-down cross functional view of the business. It means seeing the business generally and specifically from general top down business concepts to specific business relationships. It also means never trying to combine contexts. It means thinking in one language or the other. In other words it means seeing the big picture, keeping it normalized where each word has a specific meaning, and creating meaningful cross functional calculations.