

Data modelling is NOT JUST for DBMS's. Part 1 (of 2)

Data Modelling has been around for over 30 years. So, why is it that in many organisations the benefits of data modelling still need to be "sold" and in others the big benefits simply fail to be delivered?. Chris Bradley looks at some of the common misconceptions.



Introduction

"It's Data Modelling Jim, but not as we know it"

Could these be the words of Chief (data) Engineer Scotty on the StarShip Enterprise? Enterprise Data Modelling that is!

The problem is that "Data modelling" has, in too many companies received a lot of bad press. Have you heard any of these?

- "It gets in the way",
- "It takes too much time",
- "What's the point of it";
- "It's not relevant to today's systems landscape".
- "I don't need to do modelling, the package has it all"

How many of these have you heard? And they are just the polite comments!

Yet when Data modelling first came onto the radar in the 1970's the potential was enormous: We'd get benefits of:

- "a single consistent definition of data"
- "master data records of reference"
- "reduced development time"
- "improved data quality"
- "impact analysis"

"We don't need to develop a data model – our package does it all!"

to name but a few.

Would corporations want to realise these benefits? You bet, it's a no-brainer.

So then, why is it that now, here we are, 30+ years on and we see in many organisations that the benefits of data modelling still need to be "sold" and in others the big benefits simply fail to be delivered?. What's happened, what needs to change?

As with most things a look back into the past is a good place to start.

Background and History

Looking back into the history of data management; we see a number of key eras.

1950's – 70's: IT was starting to enter the world of commerce and later on in this period we saw the introduction of the first database management systems such as IMS, IDMS and TOTAL. Who can remember the DBMS that could be implemented entirely on tapes?*** Remember the cost of disc storage at the time was exceptionally high. The concept of "database" operations came into being and the early mentions of "corporate central databases" appeared. ** It was IMS HISAM if you really want to know.

1970 – 1990: Data was "discovered". Early mentions of managing data "as an asset" were seen and the concepts of Data requirements analysis and Data Modelling were introduced.

1990 – 2000: The "Enterprise" became flavour of the decade. We saw Enterprise data management coordination, Enterprise data integration, Enterprise data stewardship, Enterprise data use. An important change began to happen in this period, in that there was a dawning realisation that "technology" wasn't the answer to many of the data issues. Data Governance started to be talked about seriously

2000 and beyond: Data quality, Data modelling as a Service, Security & Compliance, Services oriented Architecture (SOA), Governance (still) and Alignment with the Business were / are the data management challenges of this period.

And all of this has to be undertaken in these rapidly changing times when we have a "new" view of Information: Web 2.0, Blogs, Mashups, Anyone can create data! At the same time we have a greater dependence on "packaged" or COTS applications such as the major ERPs. Also there's more and more use of SOA, XML, Business intelligence and less traditional "bespoke" development.

Notice I sneaked in "mashups" (or web application hybrid) there? See the Wiki article [http://en.wikipedia.org/wiki/Mashup_\(web_application_hybrid\)](http://en.wikipedia.org/wiki/Mashup_(web_application_hybrid)) for more on mashups. There are many powerful facilities available now that enable you to create your own mashups. Make

no mistake, these are now becoming the “cottage industry” IT applications of this decade. Remember the home grown departmental Excel macros of the 90's and onwards that became “critical” to parts of the business? Well mashups are doing the same thing now. But just who is looking at the data definitions, standards, applicability etc? Certainly not the data management group – because they don't know these things are being built in departmental silos, and anyway the “data team” is pigeon holed as being only involved in DBMS development.

So that leads us on to examine the belief that many people have (too many unfortunately) that Data Modelling is only for DBMS development. Why is that?

Modelling for DBMS development

In its early days data modelling WAS primarily aimed at DBMS development. We'll have a look at the two main techniques in a moment.

Just to illustrate we can look at 4 typical roles:

The Enterprise data customer:

This might be at Director or CxO level. The accuracy of data is critical, they are reports users, and the data “products” we as data professionals produce are key to serving the needs of this level of user.

“Doing this modelling stuff takes too much time – we just want to get on and build our system”

The Data Architect:

This person knows the business and its rules. He/she manages data knowledge and defines the conceptual direction and requirements for capturing of data.

The DBA:

This person is production oriented, manages data storage and the performance of databases. He also plans and manages data movement strategies and plays a major part in data architecture by working with architects to help optimise and implement their designs in databases.

The developer DBA:

This role works closely with the development teams and is focused on DBMS development. They frequently move and transform data often writing scripts and ETL to accomplish this.

Data models (more accurately the metadata) were (and are) seen as the glue or the lingua franca for integrating IT roles through the DBMS development lifecycle. All of the roles above depend on metadata from at least one of the other roles.

What then are the steps for developing DBMS's using models? This could be the subject of a huge paper, but I'll try and summarise it simply here:

There are two “main” approaches to creating DBMS's from models: One is the “top down” or “to-be” approach and the other is termed the “bottom-up” or “as-is” approach.

Top Down (to-be) approach

Step 1:

Document the business requirement and agree on high-level scope. The output is typically some form of Business Requirements Document (BRD).

Step 2:

Create a more detailed business requirement document with subscriber data requirements, business process and business rules.

Step 3:

Understand and document the business keys, attributes and definitions from business subject matter experts. From this create and continually refine a logical data model.

Step 4:

Verify the logical data model with the stakeholders. Walk a number of major use cases through the model. Apply technical design rules, known volumetric and performance criteria and create a first cut physical data model.

Step 5:

Refine the physical design with DBA support and implement the DBMS using the refined physical model.

This approach has the great advantage that the “new” or “to-be” business and data requirements are foremost. However, it doesn't take account of any existing systems.

Bottom up (as is) approach

The primary purpose of the Bottom-up or as-Is approach is to create a model of an existing system into which the new requirements can be added. Frequently, the bottom-up approach is used because a model of the as-is system simply doesn't exist often because it's evolved and /or the design staff have moved on.

The steps in this approach are:

Step 1:

Reverse engineer the database schema from the system that is already implemented. From this you will have the database catalog, table, column, index names etc. Of course these will all be in “IT” language without any business definitions.

Step 2:

Profile the real data by browsing and analyzing the data from the tables. Scan through the ETLs to find out any hidden relationships and constraints.

Step 3:

Find out foreign key relationships between tables, from IT subject matter experts and verify the findings. The typical output here is a refined physical model.

Step 4:

Document the meanings of columns, and tables from IT subject matter experts

Step 5:

Try to understand the business meanings of probable attributes and entities that may be candidates for logical data model. From here the result is a "near logical" model.

A third way is a hybrid of these frequently called the "middle out" approach.

Now all of these uses of models described so far, and the history of data modeling, and if we're honest until very recently much of the literature from the data modeling tool vendors leaves us with the assumption that data modeling is just for DBMS's.

WRONG

What needs to change?

The use and benefit of Data modeling is considerably greater than just its "one trick pony" current press would suggest. To make data modeling relevant for today's IT landscape we must show that it's relevant for the "new" technologies such as:

- ERP packages;
- SOA & XML
- Business Intelligence
- Data Lineage

We also need to break away from the "you must read my detailed data model" mentality and make the information available in a format users can readily understand. This for example means that Data Architects need to recognize the different motivations of their users and re-purpose for the audience: Don't show a business user a data model!

Information should updated instantaneously, and we must make it easy for users to give feedback, after all you'll achieve common definitions quicker that way.

We need to recognize the real world commercial climate that we're working in and break away from arcane academic arguments about notations methodologies and the like. If we want to have Data modeling play a real part in our business then it's up to us to demonstrate and communicate the real benefits that are realized. Remember, Data modeling isn't a belief system, just because you "get it" don't assume that the next person does.

Part 2:

So how can we accomplish this?

In part 2 we'll be discussing:

- Modelling for the "new" technologies
- Demonstrating benefits
- The greatest change required
- What needs to stay the same?



Chris Bradley has 30 years of Information Management experience. He is a recognised thought-leader in the Information Management field, is a published author (Data modelling for the Business- A handbook for aligning the Business with IT using high level data models), a regular columnist for the BeyeNETWORK and an International conference speaker. Chris also sits on industry advisory panels. Chris leads IPL's Business Consulting practice and can be contacted at BusinessConsulting@IPL.com.