

Data Modelling is NOT JUST for DBMS - Part 2 (of 2)

In part 1 of this article Chris Bradley looked into the history of data modelling and the benefits of modelling for DBMS development. In part 2 he discusses how data modelling is relevant in today's IT landscape



Recap

In part 1 of this article we looked into the background of data modelling and some of the history associated with it. We also touched upon the criticism modelling receives and the baggage it unfortunately attracts in far too many organisations.

We also examined the typical benefits that are expected when modelling is associated with DBMS development, and took a deep-dive into the role of modelling for DBMS development

Finally, we posed the question:

“What needs to change to make modelling more relevant to today's environments?”

In part 2 we'll be discussing:

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

Modelling for the “new” technologies

I feel I must make a confession here. The technologies are not really all that new! It's just that “traditionally” Data modelling has not been seen as being relevant to these areas. To break out of this “modelling is one trick pony” view we need to show how and why data modelling IS relevant for today's varied IT landscape.

Thus we must show that it's relevant for the “new” technologies such as:

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

ERP packages

As data architects, when faced with projects that are embarking upon the introduction of a major ERP package, have you ever heard the cry:

“We don't need a data model – the package has it all”?

But, does it?

Is data part of your business requirement? Of course it is. So just how do you know whether the package meets your overall business data requirements? You did assess the data component when doing your fitness for purposes evaluation didn't you?

A data model will assist in both package configuration and fitness for purpose evaluation

How can you assess that the ERP package has compatible data structures, definitions and meanings as your legacy systems?

Again a good data model will assist this.

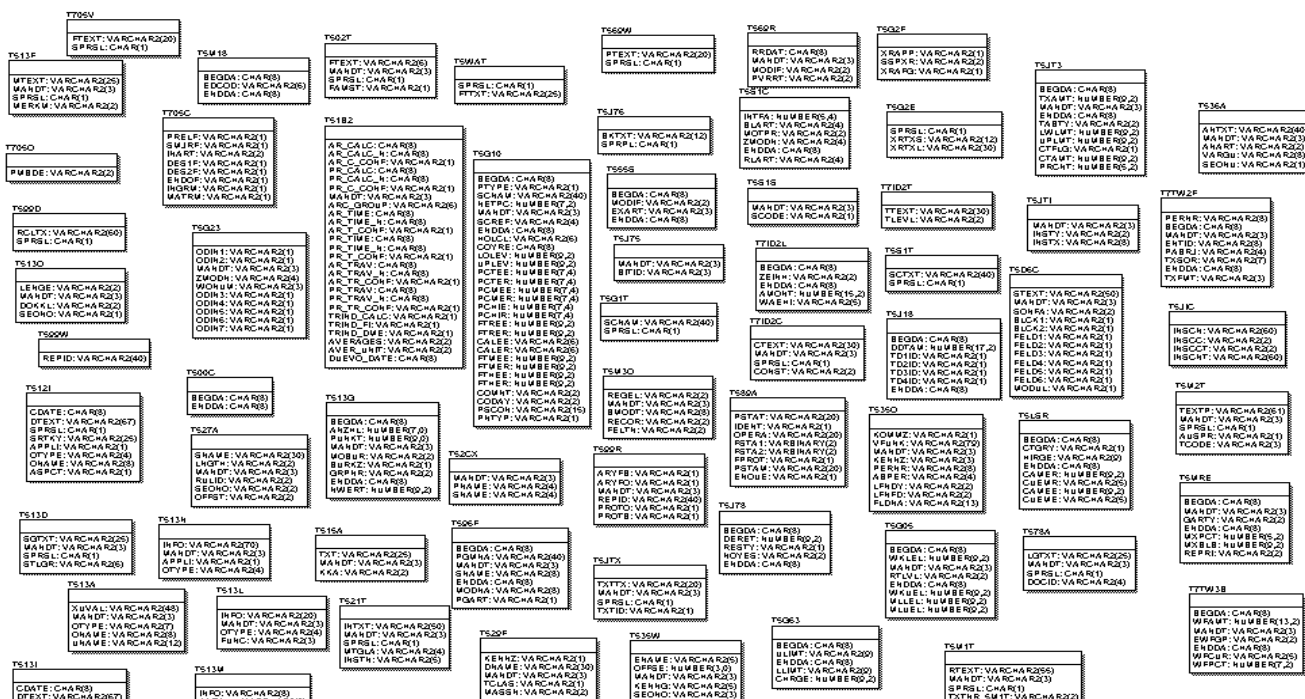
What about data integration, legacy data take on and master data integration – how can these readily be accomplished?

You guessed it – a data model can help here too.

The critics say that modelling isn't needed for ERP packages. But that's because they are wedded to the old-world view that modelling is only used for DBMS development. It's not. In this case, when implementing ERP systems, the model will NOT be required to generate a DBMS from, however for all of the other aspects described above it IS invaluable.

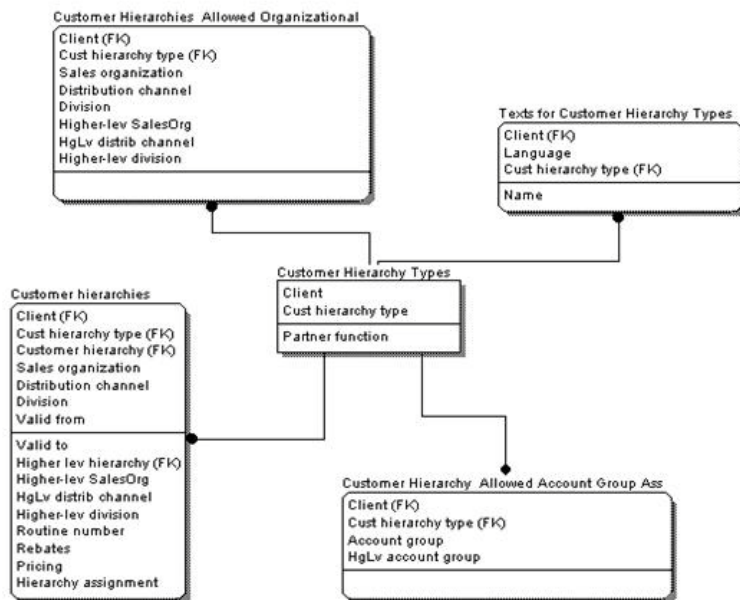
So what's the problem? Why can't we just point our favourite data modelling tool at the underlying DBMS of the package? Simply put, for the most part the problem is that Database System Catalog does not hold useful metadata. Several well known ERP systems do not hold any Primary Key (PK) or Foreign Key (FK) constraints in the Database itself (see figure 1).

Figure 1: Part of an ERP reverse engineered directly from the DBMS



It's their application layer that holds all of this, and it's their proprietary ERP DD which holds the 'Logical View' of the data. What we really need is to be able to get the ERP metadata into a useful format such as that shown in figure 2 below.

Figure 2: Useful model from an ERP



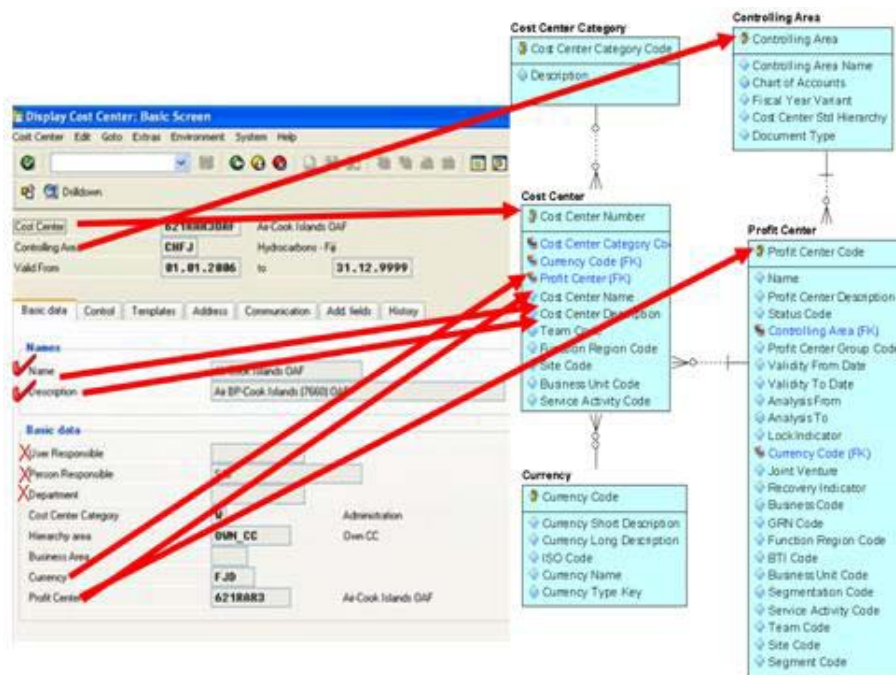
validate a model created from SAPHIR by examining key screen items such as shown in Figure 3.

So in summary, why do we need to bother undertaking data modelling when implementing an ERP system?

- 1) It's for requirements gathering. If your business data is part of your requirement, you need to model them.
- 2) It's for a fit for purpose evaluation. Surely you must have evaluated the suitability of the package before deciding to implement it?
- 3) It's for configuration. Using models as a communication vehicle to demonstrate use case is invaluable. From these the many options in the ERP system can be examined and then configured with confidence.
- 4) It's for legacy data migration and take on.
- 5) For master data alignment. The ERP may have its own master data sets. You can use the model to ensure correct alignment of these with your corporate master data initiative. Don't fall into the trap of letting the tail wag the dog!
- 6) Fundamentally, this is the key one. It's all about ensuring that your ERP data can integrate within your overall Information Architecture

How can we do that? Well there isn't space in this article to go into the detail, and much of it varies from ERP to ERP. However with for example SAP, there is a metadata extraction facility independently available called SAPHIR. Additionally, you can also

Figure 4: Validating an ERP model from transaction screens



Modeling For The Business". When we look at this real example we see data such as:

Title, Author(s), ISBN, Price, Publisher, Amazon URL and so on.

Looking at the authors, (myself, Steve & Donna) there is also some information (on the back cover) relating to each of us.

We can develop a model to represent this "real world" data and show it in an Entity Relational format. Typically these ER models can represent real world data pretty accurately.

Figure 5 shows an example ER model for the book authoring data subject area. A book can be authored by many writers (in this case, me, Steve and Donna). However, a writer can author many books (e.g. Steve has also written "Data Modeling Made Simple"). So as we see in figure 5 we've added an intersection entity (Book Authorship) to resolve the many to many relationship.

SOA and XML

I don't intend to give a detailed exposition on the subject of SOA, however it's worth reminding ourselves of the fundamental components in the architecture.

The Bus in SOA is a "conceptual" construct, which helps to get us away from point to point thinking. An approach for integrating applications via a bus is using Message Oriented Middleware (MOM).

Message Broker is a dispatcher of messages and comes in many varieties. The broker operates upon a queue of messages within the routing table.

Adapters are where the different technology worlds are translated, e.g. UNIX, Windows, OS/390 and so on.

Fundamentally, SOA is built upon a message based set of interactions, i.e. all interaction between components is through messages. These are generally XML messages, so it is true to say that XML is at the core of SoA.

But there is a potential problem.

XML is a hierarchical structure, but the real world of data is not.

Let's illustrate this with a real world example – a book. Looking at figure 4, we see that this book is entitled "Data

Now, when we want to use data in this model within an XML message, we have to turn the model into a hierarchic XML representation. Thus we need to decide whether to make Book the parent of Book Authorship or to choose Writer.

In figure 6 below, the resultant XML model has been created after choosing Book as the parent.

Figure 3: Book example

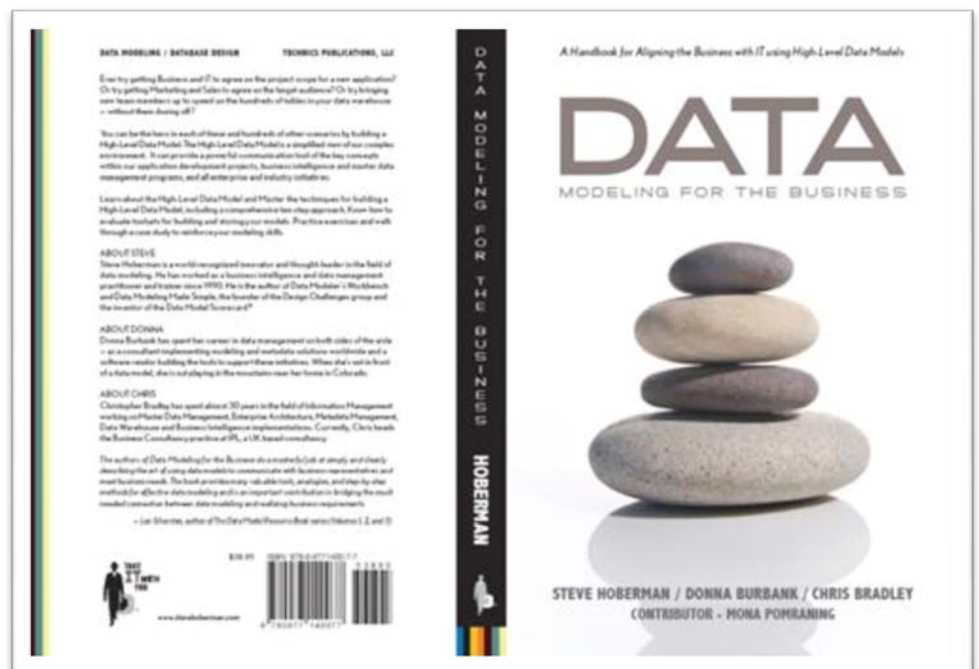


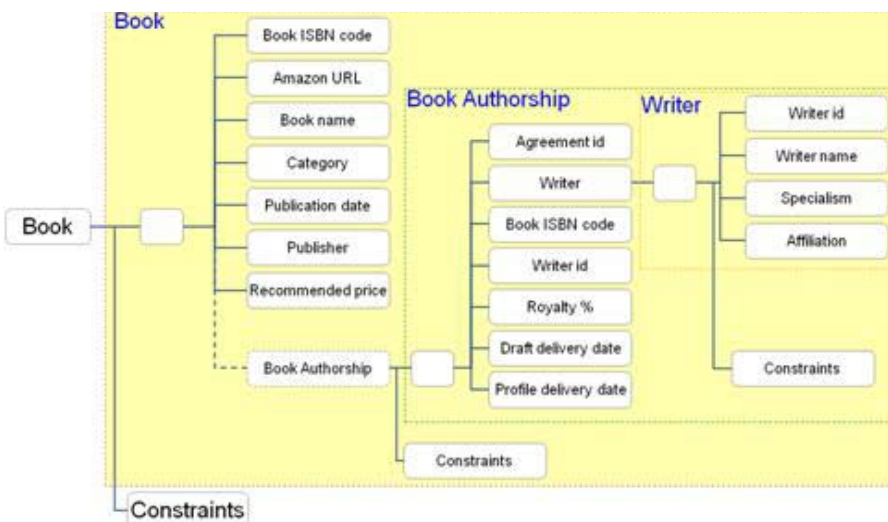
Figure 5: Book example ER model



Whilst simplistic (for the sake of the example), the XML model in figure 6 now represents the XML schema were going to use. Within our SOA based system, we may have a transaction which utilises an XML message called “Book Details”. How the XML message has been created from the XML schema and is utilised (in the message queue) in our SOA is shown in Figure 7 below.

So clearly, data modelling IS a key component required in an SOA implementation.

Figure 6: Book XML model



Business Intelligence

When looking at Business Intelligence and Data Warehouses, we are trying to ensure that the data utilised by the business for their queries and reports is reliable.

In order to accomplish this, not only do we need to manage the data that the business utilises, but also the metadata. We all know by now that much of this metadata is contained within the data models.

So, what are the main reasons for managing this model metadata?

1. **Reduce Cost:** In addition to all the other points below, the goal here is to reduce the overall cost of managing a significant part of the IT infrastructure. Managing metadata helps automate processes, reduce costly mistakes of

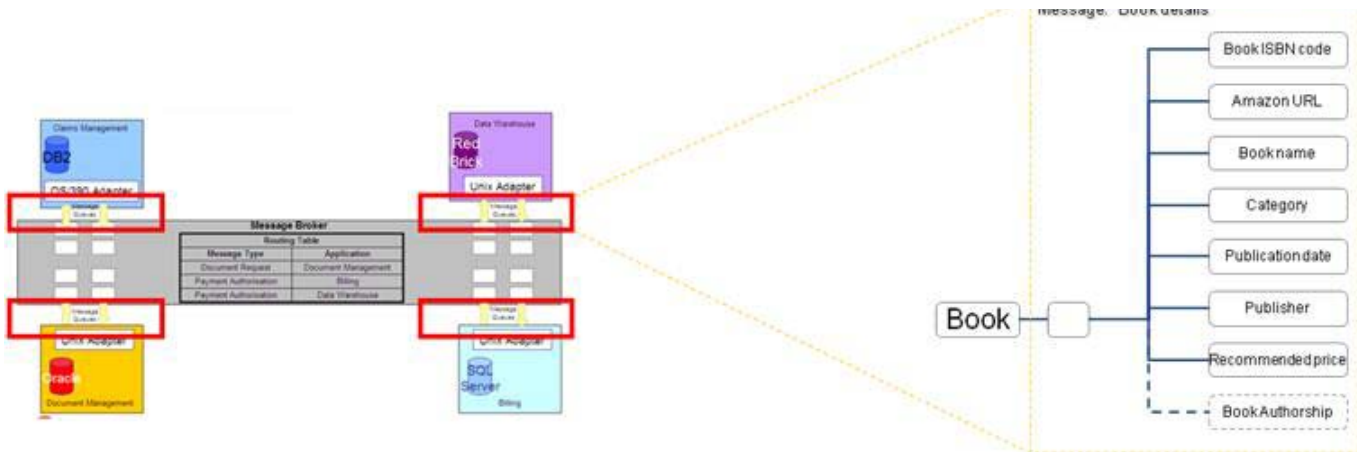
creating redundant/non-conformant data, and reduce the length of time to change systems according to business needs.

2. **Higher Data Quality:** Without proper management, the same type of data may be managed differently in the places it is used and degrade its quality/accuracy.
3. **Simplified Integration:** If data is understood and standardized, it reduces the need for complex and expensive coding and scripting to transform and massage data during integration.
4. **Asset Inventory:** Managing the knowledge about where data lives and what you store is critical for eliminating redundant creation.
5. **Reporting:** Creating a standard definition of data types and making it easy for the enterprise to find will reduce cost in application development (e.g. time to research and create new objects) as well as facilitate a general understanding of the enterprise’s data.
6. **Regulatory Compliance:** Without metadata management, you are not complying with regulations. Bottom line: An audit trail of data, starting with its whereabouts, is critical to complying with government mandates.

The top 5 benefits from managing this model metadata for reporting are:

- #5 **Data Structure Quality.** Models ensure that the business design of a data architecture is appropriately mapped to the logical design, providing comprehensive documentation on both sides.

Figure 7: Book details XML message



- #4 Data Consistency. By having standardized nomenclature for all data – including domains, sizing, and documentation formats – the risk of data redundancy or misalignment is greatly reduced.
- #3 Data Advocacy. Models help to emphasize the critical nature of data within the organization, indicating direction of data strategy and tying data architecture to overall enterprise architecture plans, and ultimately to the business’s objectives.
- #2 Data Reuse. Models, and encapsulation of the metadata underpinning data structures, ensure that data is easily identified and is leveraged correctly in the first place, speeding incremental tasks through reuse and minimizing the accidental building of redundant structures to manage the same content.
- #1 Data Knowledge. Models, combined with an efficient modelling practice, enable the effective communication of metadata throughout an organization, and ensure all stakeholders are in agreement on the most fundamental requirement: the data

- Relationships between business entities are explicit (e.g. Product determines Brand determines Manufacturer)
- Tightly coupled to current business model

Features of a dimensional model

- “Star Schema” (or snowflake or even star flake)
- Optimised for reporting
- Business entities are de-normalised
- More data redundancy to support faster query performance
- Relationships between business entities are implicit (it’s evident that a Product has a Brand and Manufacturer, but the nature of the relationship between these entities is not immediately obvious)
- Loosely coupled to business model – changes to the business model can often be accommodated via graceful changes without invalidating existing data or applications.

So, where should I start if I want to develop a model that is suitable for reporting?

- Firstly, start with the existing data landscape. There should be an existing conceptual or logical data model for the area under consideration. Hopefully there will also be a physical model of the source system too.
- Examine the data model of the source system (ER Model)
- Identify the facts and their level of granularity
- Identify the dimensions and their position within hierarchies
- Design the dimensional model

ER Models vs. Dimensional models for reporting

Much has been written previously about the appropriateness of ER vs. Dimensional models for BI and Data Warehousing.

To dispel any myths it’s worth looking at the key features of each type of model:

Features of an ER model

- Optimised for transactional processing (arrival of new data)
- Normalised – typically in 3rd (or 5th normal form)
- Designed for low redundancy of data

- Define mappings and transformations from fields in the source system (ER Model) to fields in the dimensional model
 - Hierarchies map to dimension tables (sometimes after applying a lookup)
 - Transaction figures map to measures in FACT tables (sometimes after applying some aggregation or other calculation)

Data lineage

Don't forget data lineage – it's applicable to many aspects, and now with regulatory compliance requirements in many sectors this is now a statutory need.

In BI and DW applications, mappings and transformations determine how each field in the Dimensional Model is derived. The derivations could actually drive the ETL process. In lineage, like BI the metadata is vital!

What is the problem?

Fundamentally we need to be able to help business users to answer questions or concerns raised such as:

That figure doesn't look right! Where does it come from?

How can we prove to the auditor that financial data has been handled correctly?

Not only do we need to help our primary customers (the business folks), but also we need to be able to help IT staff to answer questions such as:

I need to integrate the data supplied from your system with the data in my system. How can I understand where your data has come from and what it means?

And finally, we need to be able to help systems to answer questions such as:

When a piece of source data is updated, which items in the Data Warehouse will need to be recalculated?

So why does data lineage matter?

We aim to have an increased understanding of where data comes from and how it is used, which will lead to increased confidence in the accuracy of data

The knowledge of how data is transformed is itself valuable intellectual property that should be retained within a business, and very importantly it is absolutely necessary for compliance with the Basel II Accord and Sarbanes-Oxley Act (SOX): SOX requires that lineage & transformation of financial data is recorded as it flows through business systems.

Two key aspects of Data Lineage

Transformations:

What has been done to the data?

Business Processes:

Which business processes can be applied to the data?

What type of actions do those processes perform (Create, Read, Update, Delete)?

Audit Trail – who has supplied, accessed, updated, approved and deleted the data and when?

Which processes have acted on the data?

So where do I need Data Lineage?

For the design of ETL processes, the creation of Dimensional Models, the transforming data to XML (typically from ER) and for workflow design.

Demonstrating benefits

As I mentioned in part 1 of this article, we constantly need to demonstrate the benefits accruing from data modelling. Nobody owes us a living, and no matter how important WE believe the place of modelling to be, it is beholdant upon us to demonstrate (and sell) the benefits within our organisations.

So just how can you gain traction, budget and Executive buy-in? Here's a few tips:

1. Be Visible about the program:
 - a. Identify key decision-makers in your organization and update them on your project and its value to the organization
 - b. Focus on the most important data that is crucial to the business first! Publish that and get buy in before moving on. (e.g. start small with a core set of data)
2. Monitor the progress of your project and show its value:
3. Define deliverables, goals and key performance indicators (KPIs)
4. Start small—focus on core data that is highly visible in the organization. Don't try to "boil the ocean" initially.
5. Track and Promote progress that is made

6. Measure Metrics where possible
 - a. "Hard data" is easy (for example # data elements, #end users, money saved, etc.)
 - b. "Softer data" is important as well (data quality, improved decision-making, etc.) Anecdotal examples help with business/executive users e.g. "Did you realize we were using the wrong calculation for Total Revenue?" (based on data definitions)

Remember, soft skills are becoming critically important for Information professionals, and whilst you might not like it, the hard facts are that part of YOUR job nowadays IS marketing.

The greatest change required

As Information Professionals, we need to break away from the "you must read my detailed data model" mentality and make the appropriate model 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 repurpose the information they present to be suitable for the audience: Don't show a business user a data model!

Information should update 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 modelling play a real part in our business then it's up to us to demonstrate and communicate the benefits that are realized. Remember, Data modelling isn't a belief system, just because you "get it" don't assume that the next person does.

So what can we do?

1. Provide Information to users in their "Language"
 - Repurpose information into various tools: BI, ETL, DDL, etc.
 - Publish to the Web
 - Exploit collaboration tools / SharePoint / Wiki and so on. What about a Company Information Management Twitter channel?
 - Business users like Excel, Word, Web tools, so make the relevant data available to them in these formats.
2. Document Metadata
 - Data in Context (by Organization, Project, etc.)
 - Data with Definitions
3. Provide the Right Amount of Information
 - Don't overwhelm with too much information. For business users, terms and definitions might be enough.

- Cater to your audience. Don't show DDL to a business user or business definitions to a DBA.
4. Market, Market, Market!
 - Provide Visibility to your project.
 - Talk to teams in the organization that are looking for assistance
 - Provide short-term results with a subset of information, and then move on.
 5. Be aware of the differences in behaviour & motivations of different types of users, for example a DBA is typically:
 - Cautious
 - Analytical
 - Structured
 - Doesn't like to talk
 - "Just let me code!"
 -

However a Data Architect is:

- Analytical
- Structured
- Passionate
- "Big Picture" focused
- Likes to Talk
- "Let me tell you about my data model!"

And a Business Executive is:

- Results-Oriented
- "Big Picture" focused
- Has little Time
- "How is this going to help me?"
- "I don't care about your data model."
- "I don't have time."

As Information professionals we've got to get these softer skills baked into ourselves and our colleagues. Some of the key things as a profession we can do are to:

- Develop Interpersonal skills
- Avoid methodology wars & notation bigots. Please don't air discussions about Barker vs. IE vs. UML class diagrams in front of business users. Yes, sadly enough I have seen this done!
- Remember, nobody owes us a living, so we must constantly demonstrate benefits. As data professionals we constantly need to fight for their existence
- Examine professional certification (CDMP / BCS etc). This shows we are serious about our profession.

What needs to stay the same?

So having highlighted the areas that need to change in order to make modelling more relevant to our business colleagues, and the information environments of today, are there any things that should stay the same?

Yes indeed.

We must keep the disciplines and best practices that have existed in the modelling community for many years. These can be categorised into 3 major areas as follows:

1) Modelling rigour:

Development of Conceptual, Logical and Physical Data models with good lineage and object re-use.

Structures created in the most appropriate normal form (typically 3rd normal form);

Good and consistent data definitions, for all components of the data model.

2) Standards & Governance

These cover standards for both development and usage of information models, including aspects of data quality.

Data Governance including ownership, stewardship and operational control of the data.

3) Object reuse via a common repository

Not only used for data modelling, the metadata that is captured whilst developing Conceptual, Logical and Physical Data models is of immense use for many aspects of the business. Interestingly, several organisations are now beginning to use this metadata as the basis of their Business Data Dictionaries.

The key here is holding the metadata in a common, repository and reusing the objects where appropriate.

So, in the course of these two articles we have examined many aspects of data modelling, starting with its history, its use in DBMS development and firmly refuting the criticism that it's only appropriate for DBMS development.

But as data professionals, it's up to us to make the biggest change necessary to make it appropriate to the new technologies and business environments of today. We need to grasp the nettle and engage effectively within our businesses.

Go to it

~♦~

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.