Not just top-down and bottom-up: classifying data modelling projects

Rosemary Tomlinson PhD MCMI ChMC

IBM

Contents

Summary 03

Introduction 04

Top-down and Bottom-Up Approaches to Data Modelling 05

A Closer Look at Data Modelling Approaches 07

So What? 13

Conclusion 15

About the Author 16

Summary

Data modelling engagements are most frequently identified as being either ‘top-down’, starting with requirements or, less commonly, ‘bottom-up’ where the data modeller is directed to existing systems as key input for their models. This paper explores the dynamics of these alternative approaches on the data modelling experience and outcomes, adding a further classification of ‘development support’ modelling whereby data models are developed alongside agile development teams. The enhanced model is intended to support data modellers in recognising the implications of the nature of their own projects for their model outcomes.

Introduction

The conventional view of the role of the data modeller from a commercial perspective is as a technical skill. In particular, conceptual and logical data modelling[1] are tried and tested techniques for getting stakeholders and development teams on to the same page and ensuring that data in systems meet the business data requirements.

But those of us working in this area are highly conscious that there is art as well as science in the endeavour and that individuals vary in how they tackle problems in their projects. Some of that is our learning through successful projects, which is then recognised as experience and expertise. But sometimes our approach may not work as successfully as we anticipated in the context of a specific project, and that experience may alter how we go about modelling next time. Many of us work as consultants, either explicitly contracted for through companies like IBM or, in larger companies, assigned from a resource pool into a specific project.

Data modelling engagements differ considerably. Project results and our working environment benefit if we can recognise their nature, identify the associated risks and work with others in the team to mitigate them. One of the joys of the data modelling role (along with that of the business analyst) is the role's position as a conduit between the business and technical worlds. A good data modeller should possess both the personal and analytical skills to chase supplementary information, wherever it resides, to amend and improve their models.

This paper examines the shape of the projects with which we engage in some depth. Not all engagements require or allow a classical top-down approach to modelling, expanding from a conceptual to a logical data model. It proposes a simple classification of projects, describing each category, and then compares them using consistent dimensions. Finally, recommendations are made to show how this classification can be used as a tool against which to reflect how projects can variously impact our data models. This will enable us to engage on our projects in a more proactive and considered manner.

[1] Conceptual and logical data modelling are common industry terms to denote increasing levels of detail in the data modelling. The Conceptual Data Model represents the 'business on a page', a high-level view of 10-15 key subject areas and their important relationships within the enterprise scope under consideration. The Logical Data Model expands this in far greater detail but retains a technology-agnostic viewpoint.

Top-down and Bottom-Up Approaches to Data Modelling

While there is a substantive literature on the technical discipline of data modelling, with many textbooks devoted to sage advice on reusing repeatable patterns and debating appropriate levels of generalisation, there is little discussion as to how the activity should be approached on real-world technical projects. Where a leading practice is presented, the default construct is that of top-down data modelling.

Here it may be recommended that we start with a very high-level conceptual data model with the entire business on a page and then flesh out the business data requirements in a technology-agnostic logical data model. Sometimes this may be just to entity level[2] with primary and foreign keys[3], but if we are aiming to document the data requirements for a system then attributes must also be defined. Finally, the data model is iterated into a physical version, ready for instantiation into the chosen database technology, for example by defining indexes and denormalising[4] for performance.

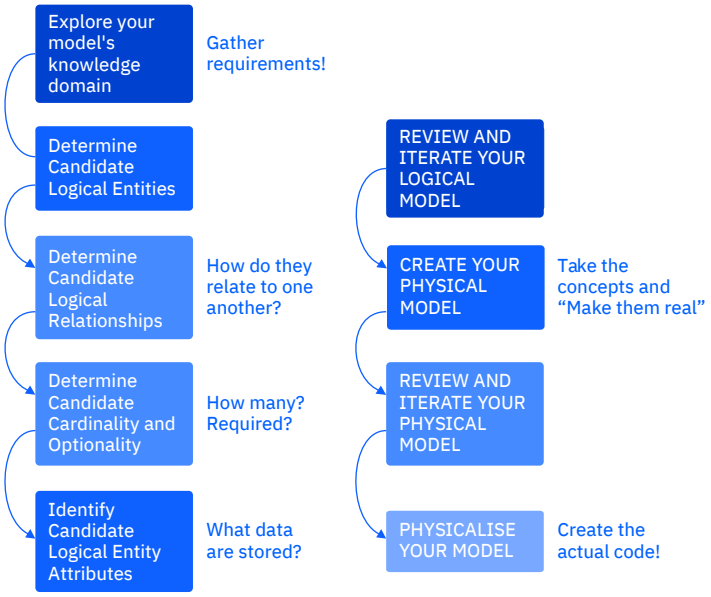
IBM's comprehensive consultant training in data modelling takes this approach. As the course author, Alastair McCullough, puts it "Waterfall methods represent the easiest way to teach new people about data modelling."

[2] Entity. A distinct data object in a model which can be related to other entities. Examples include Address, Sales Order, Customer, Product. An attribute is an individual column within and entity.

[3] Primary and Foreign Key. A primary key is one or more attributes which uniquely identify an entity. A foreign key is a reference in one entity to the primary key from another entity, showing there is a relationship between the two entities. For example, an Address may include a foreign key which is the primary key of the Customer who lives there.

[4] Denormalising. The process of removing relationships between entities and consequently database joins between database tables. This increases database performance as joins increase query time.

High Level Steps involved in Data Modelling — Alastair McCullough



The Data Model Guidelines within IBM's Digital Insight Method discuss both top-down and bottom-up approaches. Top-down approaches begin with business knowledge, usually obtained through process definition or business rule development. Bottom-up approaches aggregate individual data (identified from existing documentation such as reports, screenshots and databases) into groups which are then organised into entities.

More widely, Scott Ambler's Agile Data website works through the initial requirements to produce Domain Model iterations and then a physical data model. John Giles has published the most detailed series of posts discussing why and how to do top-down modelling. His first post presents an instructive review of the debate between top-down and bottom approaches, with his preference leaning towards the former.

Another publicly-available reference, Ronald van Loon's 9 skills you need to become a data modeller blog, also refers to both top-down and bottom-up approaches. This suggests that a top-down approach is usually based on discussions with knowledgeable people, and that a bottom-up approach makes data sharing difficult as models are built without wider reference than the physical models themselves. This is an interesting point, but, as we shall see, it is not uncommon for clients to attempt this with the aim of producing an enterprise-wide logical model.

For many in the data industry, the Wikipedia page on data modeling [sic] will carry some influence and is therefore worth noting. At the time of writing, a diagram on this site shows a development flow from the logical/conceptual to the physical in a top-down manner.

For completion, wider enterprise approaches were also interrogated to see if they had anything to say in this area. The TOGAF® Standard[5] makes no mention but The Zachman Framework for Enterprise Architecture™ [6] is structured from higher concepts to more detail. However, it is always very explicit that it is an ontology/framework which says nothing about the methodology or process around using it.

It would appear from available sources that the top-down approach is the default recommended approach. Logical data models have been built from physical structures, though perhaps less successfully.

[5] TOGAF is a registered trademark of The Open Group.

[6] Published with the permission of John A. Zachman and Zachman International®,Inc - www.zachman.com

A Closer Look at Data Modelling Approaches

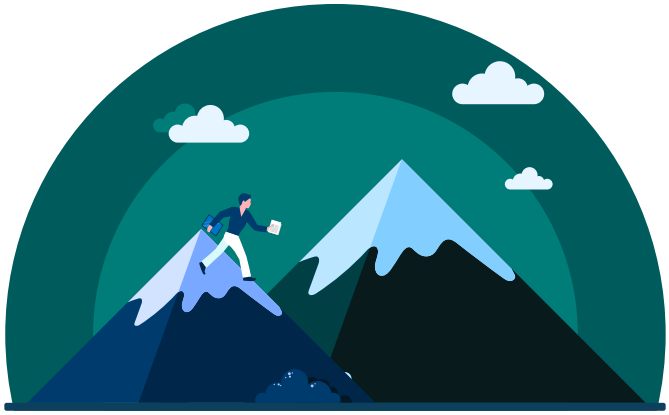
So, while this limited consideration appears to be the industry's default view, it does not chime with the author's experience over numerous projects. In particular, while top-down is generally recommended, these references assume that the data modeller has the luxury of choosing which approach to take; this is not always the case when a programme is up and running at the point of engagement. Indeed, the feel of the project and the nature of the work can differ considerably according to the approach.

From the author's experience, about half the projects undertaken relate to the requirements-based, top-down pattern. Around a quarter of engagements have been bottom-up, focused on deriving logical models from physical systems. However, this author considers that there is possibly a third category, supporting the development process very closely but not in a very top-down fashion. (As an aside, the division bears a striking resemblance to a subtype of our old friend 'People, Process and Technology'!)

So, let us take each of these categories and look at how their nature tends to affect the data modelling process and output across four dimensions:

- **Degree of business focus** - to what extent are business stakeholders likely to engage with the data modelling exercise given all their 'day job' pressures?
- **Potential for interpersonal conflict** - is the approach likely to generate debate and friction between stakeholder perspectives?
- **Ease of data model playback and review** - is the relationship between inputs into the data modelling exercise and its outputs clear when the modeller is presenting the model back to stakeholders?
- **Degree to which reflects current business requirements** - does the data model reflect the true business requirements as understood by the enterprise?

Top-Down, Requirements Based Models



This category reflects the majority view as outlined above. A typical request from such a project to the newly appointed data modeller is to understand the data requirements, delivering conceptual and logical data models in that order. The project itself may be for a new system, or it could be a reaction against an existing system or a previous implementation where it is considered that business needs have not been met, and a step back to requirement gathering is necessary.

In both cases, model development is likely to entail the consideration of any existing requirements and their fresh capture and/or confirmation with business stakeholders, maybe alongside other business analysts, such as process specialists.

Data models are developed with a view to their being physically instantiated, whether hand cranked or used as the requirements against which to evaluate Commercial Off The Shelf ('COTS') packages.

Given a requirement-driven, top-down data modelling engagement, what can we then say is likely about the nature of the modelling?

Degree of business focus

There is a significant incentive for stakeholders to engage fully where the modelling is part of a programme to align requirements and deliver a physical system.

Potential for interpersonal conflict

Considerable negotiations are often required around concepts and definitions to arrive at an enterprise data view across business functions. In some sectors, such as education or business travel, it is very common for the Finance Department to regard the Customer as the Organisation paying, while Marketing will seek to build relationships directly with its employees.

However, since there is explicit and engaged stakeholder input around requirements, it should be relatively straightforward for all to meet and negotiate around the model. And, technically, standard data modelling techniques can handle these nuances.

Ease of playback

Playback sessions are often the most successful in this category as there should be a clear link between the recent input and the model, which facilitates review. The link between the requirements and the model is especially clearest where the engagement starts with a 'blank sheet of paper', engaged stakeholders and recent requirements documents.

Degree to which reflects current business requirements

This approach should certainly reflect requirements as perceived by stakeholders; interviewees are typically keen for the data modelling consultant to understand their world so that the new system can accommodate their business processes.

However, there are two other influencing factors:

Is the strategic direction clear?

The robustness of the process is enhanced where the enterprise's strategic direction has been thought through, for example by using a Strategic Capability Network[7]. Such elaboration helps ground both the process and data-related requirements so that stakeholders' viewpoints can be traced back to enterprise's Value Propositions and tested. Where

[7] The Strategic Capability Network-provides a simple approach to connect conceptual elements of strategy into a model which can be decomposed into 'enablers' such as process, information and technology. Patented by IBM. It is described further and linked to Information Models and Data Models in Hans-Peter Hoidn's paper on Enterprise Architectures.

well-communicated and understood, the overall business strategy can guide the discussion in stakeholder interviews. Requirements elicited in this manner are more likely to be future-proofed, resulting in better solution selection and delivery.

Sadly, even large programmes are frequently technology-driven, for example to rationalise and consolidate 'stovepiped' legacy systems. Here it can be much harder to tie even quite large decisions back to the organisational Vision, which is rarely written with this purpose in mind. There is more room for interpretation and without this collective view, data and process modelling based on detailed stakeholder input can risk replicating the as-is.

How volatile are the true business data requirements?

Any requirements captured in a one-off exercise may be inherently volatile, particularly where they run alongside wider changes in business practice. Detailed requirements' capture may be irrelevant where the whole business is being transformed due to a new business strategy coming into play.

Requirements gathered in the context of a new technology may force changes to business SMEs' data understandings. For example, ERP systems' implementations may impose a different way of viewing the world so the organisation can benefit from their integration capabilities. Stakeholders may take time to adjust, declaring requirements which need to be revisited as their understanding grows. This human reaction takes time to process through. On a more positive note, some of this pain may be limited if high level data models can help stakeholders visualise and compare the before and after as early on as possible.

Even where engaged stakeholders believe they understand the enterprise's direction, relying on this for the data requirements can prove risky. This is because individuals' interpretations are vulnerable to organisational politics and strategic directions can swivel very quickly with changes in senior appointments or revised organisational Vision.

Bottom-Up Data Model Development

In the author's recent experience, bottom-up data modelling engagements are more usual than is commonly supposed, and merit closer analysis.

Some clients may ask for a logical data model based on existing physical systems to give them a holistic picture of their data, typically where their understanding is currently piecemeal, yet they then want, based on this relative paucity of insight, to make major enterprise-wide architectural decisions.

For a critical, possibly undocumented system impacted by major business change, understanding the core conceptual and logical data functionality is a prerequisite to deciding whether it is a sensible starting place to meet the enterprise's new business strategy. Similarly, enterprises created from the integration of several systems with previously 'stovepiped', localised or non-strategic, development may benefit from an encompassing data model from which to educate wider stakeholders. The process can identify synergies and conflicts in data usages between the systems and inform future developments.

Where a requirement-based engagement should include discussions with business stakeholders and a review of requirement documentation, bottom-up model interviews will more typically be with more technical staff. Frontline staff will know their systems, but the work of figuring out and reconciling the underlying requirements driving them will reside more with the modeller.

Investigations here can be fascinating archaeological exercises, disinterring the bones of corporate history. Examining a core system's physical data model allows us to uncover previous requirements which may no longer fit the current business environment. Acquisition of another business or integration with third party software may result in multiple tables performing the same function with only a slight difference in context, for example each table may apply to a particular product brand or geographical area. Entire core tables may be duplicated to accommodate an additional reference data item for a specific purpose to avoid interrupting live service. Even tidying this up in a 'semi-logical' model can provide insight to stakeholders regarding system contents.

An additional tension may be introduced where it is desired (and the modelling activity may be promised to deliver) that the location of data can be broadly identified and then traced in the physical systems as well so that it can be said that the physical data schemas have been mapped and are covered by the data model. A consulting client may want to see this evidenced through, for example, metadata tooling or spreadsheet mappings.

This conflicts with the usual understanding of the logical data model as representing the data requirements and raises issues for which the data modeller needs to agree working principles.

- Should legacy functionality be included? This risks reflecting the as-is situation rather than the strategic model.
- Should only requirements that have actually been built be included? If so, if the system is still being enhanced, a robust procedure is necessary to feedback incremental changes so the model can be updated. Actual changes may not have ended up flowing from any modelling thinking done!
- Should it be hypothetically possible to physicalise the model into any of the systems in the environment?

If the answer to the last point is 'Yes', the models must encompass all the functionality of the considered systems, each of which may have been physicalised based on requirements articulated differently and where the entities have been specified based on differing contexts.

Bottom-up development will therefore tend to result in a more generalised model overall in order to accommodate all the systems, and in doing so, is likely to include functionality such as additional relationships that may not be physicalised in any system.

For example, in an education setting a system may have two tables, one for pupils and one for staff, and another system may have a table for administrators. It may be obvious and more elegant to model these as party roles in the overall enterprise logical model, but in doing so, there is then 'space' for additional roles which are not captured in any system. The data model meets the demands of representing data requirements as represented by the systems, but they are more flexible, futureproofed and 'generous' than the business has instantiated. The desirability of this will vary by audience.

So, given a bottom-up data modelling project, what can we then say is likely about the nature of the modelling?

Degree of business focus

Business focus is likely to be more diffuse than when the engagement is requirement-led as the logical data model is less likely to be explicitly physicalised as part of the project. It can, though, work well as an arena for discussion. A model is far more consumable for a wider audience than, for example, table lists, SQL DDL or JSON scripts.

Potential for interpersonal conflict

There is less likely to be tension here during the model development and review with interviewees as the immediate purpose is not to crystallise and instantiate the requirements, since the physical systems already exist. The purpose is to drive greater awareness to support future change.

The conflict will arise when the enterprise understands the overall position better. When presented back to more senior stakeholders, the exercise may drive out previously unconsidered requirements, revise architectural assumptions about the role the existing systems may play in the new world and impact any data migration strategy.

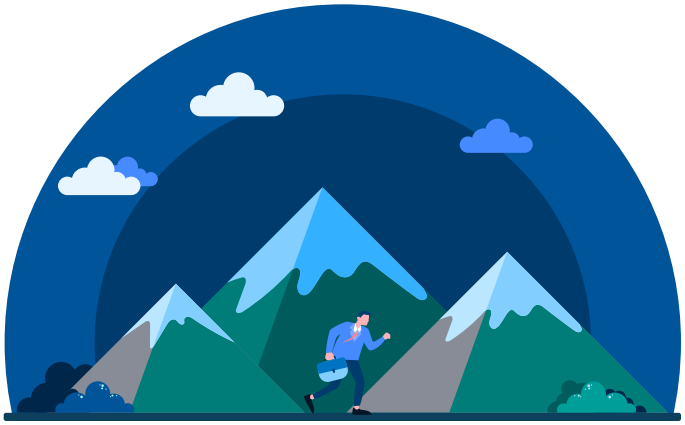
Ease of playback

As the model may be generated as a compromise between systems, it may contain flexibility and generalisations which reflect no one system. This will need explaining to those associated with the systems. In inheriting historic requirements and physical decisions, the model may not reflect the current business requirements or business strategy. This will need explaining to business stakeholders.

Degree to which reflects current business requirements

Analysis of physical systems, especially legacy, will reflect historic requirements as articulated and instantiated by numerous individuals, whose thinking may not be available to the modeller.

Supporting the Development Process in a Big Data Environment



The third context of logical data modelling is explicitly alongside pipeline development teams. These typically use Agile methodologies to incrementally develop a small amount of functionality over short periods of time, often called Sprints. The data modeller may be supporting multiple physical development efforts based on the minimal transformation of source data. The role here is to bring in some alignment between the teams to minimise the multiple accessing of source data and to enable the reuse of data structures.

The Sprint-driven urgency of this approach focuses minds on creating a model that meets immediate software development needs. The logical data modeller may be working very closely with the developers, and without the buffer of a physical database design expert common in previous waterfall or data warehousing approaches. However, this urgency can result in technical debt being incurred and in increased work for later pipelines if they turn out to be insufficiently generalised and the early specific model needs to be reworked. Version control can also become awkward when early stakeholders are happy that their requirements are met and want to keep their specific model. This can be mitigated if data modelling experts are engaged sufficiently early on to put in place some basic conceptual or logical model patterns and if some key business domains are then modelled from a wider perspective early in the development lifecycle.

However, in a pressured environment, the approach can result in modellers suffering from a feeling of continual 'cat-herding.' That is, the speed of development makes it more likely that the more thoughtful logical modelling insights may be ignored, storing up problems for the future. If more time could be allowed, the rich value of the models and the thought behind them could result in more effective implementation. This is particularly the case for the take-on of new data and in supporting incoming data migrations.

It is the author's opinion that this approach of a data modelling project running alongside pipeline development is the hardest to negotiate on a day-to-day basis. What can we then say is likely about the nature of the modelling here?

Degree of business focus

Development teams are strongly focused on short-term business deliverables. However, it can be harder to take a more rounded business view of requirements.

Potential for interpersonal conflict

Stakeholders may have their requirements developed speedily. Friction may arise if delay is required to create a more robust model to fit multiple pipelines when the benefits will be felt by successive pipelines.

Ease of playback

This varies according to the perspective of the development teams compared with that of the wider business. Engagement with the model is likely to be high with the pipeline teams as it helps them to align and develop more efficiently. But the stove-piped nature of business engagement may make it harder to win engagement across pipelines.

Degree to which reflects current business requirements

In the short term, the physical data model, with input from existing wider logical model thinking, will reflect immediate requirements. The difficulty lies in relating models to the wider business priorities over a longer period.

So what?

So why does it matter with which type of data modelling engagement we are detailing? Does the approach change the nature of the end result and, if so, how does that affect the success of our work and our reputations as data modellers?

The differing constraints of each approach will affect the strengths and limitations of the eventual model.

A stakeholder-based, requirements-driven data model can provide a blueprint to deliver the core business capabilities of the organisation. There is a risk, though, of storing up future challenges and technical debt when business capabilities are driven purely by senior stakeholders and requirements are captured as a pure standalone activity. Effective integration and communication between the two is critical. Existing systems knowledge can capture essential detailed requirements which may be otherwise assumed.

Conversely, driving the models from existing systems will provide granularity and an understanding of historic decisions, which is useful for enabling teams' alignment, but will not fully represent the business' current strategy. A logical model based on existing systems may end up 'baggier' and more generalised in order to incorporate all inferred requirements. This is especially pertinent if the embedded knowledge and decision history of the systems have both been dispersed. Sprint-driven development support may often be the closest the logical data modeller gets to actual system development. It can produce the most immediate returns for modelling effort, but without an awareness of the wider environment and some scope for thought, such as modelling some key business subject domains as reusable assets, there is a risk that the data modeller supports the data developers into a tactical dead end based on immediate demands alone.

The approach taken determines the key risks to a successful project for the data modeller. A requirements-driven approach may encounter difficulties at the implementation stage. Only then may some requirements, buried in development history, become apparent. A failure to recognise them may sometimes be blamed on the data modeller. On the other hand, if, for example due to stakeholder availability, the only resources available to support

the work have been existing systems which are being consolidated, the data modeller may then be criticised for propagating existing issues into the shiny new architecture.

Several projects have tasked the author with documenting 'the logical data model' based on existing systems, which is an inherent conflict when the logical data model is understood as 'the data requirements'. Nevertheless, the exercise can perfectly illustrate how unfit for purpose the existing systems are and provide the basis for migration decisions. If we can produce this understanding while the business resolves its wider requirements, generating a truer and more robust logical model for the data requirements subsequently can be a very short exercise.

Recommendations

Finally, the above analysis can be used to generate specific actions for data modellers according to the type of project to which they find themselves assigned. These additional activities, admittedly within other project constraints, will help mitigate the potential deficiencies of each approach.

These can be summarised as follows:

Area	Top-Down	Bottom-Up	Development Support
Focus	Learn as much as is feasible about the existing systems and where they do and do not work	Keep abreast of wider requirements and business strategy	Keep abreast of wider requirements and business strategy
Conflict	Work to get cross functional agreement on requirements	Work to ensure senior stakeholders understand what the model says about the business. Work with them to get agreement to drop defunct functionality and tweak suboptimal functionality	Work to ensure the wider business view is not neglected in each iteration
Playback	Involve systems owners: They may be able to spot important holes in requirements	Additional effort may be required to engage business stakeholders; in illustrating what it says about enterprise requirements (which can be challenged) and the size of the transformation task	Involve business stakeholders interested in the same data domain but not tied to a specific pipeline
Requirements alignment	Check that stakeholders are aligned with wider business direction as well as mapping existing activities into the new world	Additional effort will be required to understand requirements coming from the business if this is not the mandated focus of the work	Additional effort will be required to embed wider business requirements. Wider playbacks and reviews than strictly mandated may assist with this

Conclusion

Data modelling projects come in many shapes, and it is important to recognise that not all allow a classical top-down approach to their development. The categorisation of approaches suggested here provides a simple framework for data modellers to assess the structure of their project and its potential implications for their engagement. Recognising this variety will help data modellers identify and circumvent potential pitfalls in their work.

About the Author

Dr **Rosemary Tomlinson** is a Chartered Management Consultant and IBM Certified Expert Consultant within IBM's Data Platform Services, based in the UK. She has over twenty-five years' cross-industry experience helping clients understand and deliver on their information requirements and many engagements over the last fifteen years have entailed enterprise data modelling.

For more information

See https://www.ibm.com/services/data-analytics

IBM United Kingdom Limited 76/78 Upper Ground
South Bank London SE1 9PZ

Produced in the United Kingdom
September 2021