Data Solution
Design Patterns
by Roelant Vos

Training Overview

Practical training with ready-to-use patterns to architect, implement and fully automate your data solution.

Contents - day 1

The first training day is focused on the essential concepts and architecture for the data solution, and what the overall objectives are for working with data.

There are various ways to design, model, data solutions.

These are more than just technical solution alternatives. It is important to have a clear understanding of their underlying ideas, and how they impact the overall solution architecture.

The training content includes a refresher on intra-systems integration, implementation approaches, data behaviour and modelling for business process alignment, as well as the entities and constructs used in data solution modelling.

To start focusing on automation and code generation, the mechanism for capturing design metadata is also covered on the first day.

At the end of the day, we will have covered the main architecture building blocks ('patterns') for modelling, designing, and implementing a flexible data solution and how to capture this in design metadata.

Forenoon sessions (0830 – 1230):

- Session 1 – Introductions
- Session 2 – Pattern-based design

  Break – 15 min – Morning tea (10:45-11:00)

- Session 3 – Data solution architecture(s)

  Lunch break (1230 – 1330)

Afternoon sessions (1330 – 1700):

- Session 4 – Data staging concepts

  Break – 15 min – Afternoon tea (15:00-15:15)

- Session 5 – Modelling concepts
- Session 6 – Metadata and code generation

Contents - day 2

The 2nd training day covers the key areas of the *integration layer*. - the heart of the solution.

We'll start with an overview of the Core Business Concept ('CBC') pattern, the Natural Business Relationship ('NBR') pattern, and delve into advanced topics of managing contextual data and the complexities around time-variance.

At this stage, the main entity types are available in the data solution, and the focus shifts to orchestration approaches - and how to embed these in a process control framework.

The control framework is an essential component to guarantee reliable information delivery and ties in to the conceptual and technical implications of parallel loading and managing of consistency of data.

The collection of patterns then be delivered using process automation, as part of the implementation of release management and 'DevOps'.

Forenoon sessions (0830 – 1230):

- Session 1 – Core Business Concept pattern
- Session 2 – Natural Business Relationship pattern

Break – 15 min – Morning tea (10:45-11:00)

- Session 3 – Context pattern - part 1 (introduction)

Lunch break (1230 – 1330)

Afternoon sessions (1330 – 1700):

- Session 4 – Context pattern - part 2 (historization, handling time-variant data)
- Session 5 – Control framework

Break – 15 min – Afternoon tea (15:00-15:15)

- Session 6 – Testing
- Session 7 – Technical considerations
- Session 8 – Orchestration, workflows, and parallelism
- Session 9 – DevOps and versioning

Contents - day 3

Day 3 is focused on the delivery of information for consumption. This means investigating the transition from the data solution integration layer to various forms of data delivery ('marts').

As part of the information delivery, the application of business logic and technical considerations are covered.

Data delivery requires a transformation of the timelines that are used, including different ways to consider historised data. The *bitemporal* approach is a key component of this step, and requires a significant amount of time on day 3

When all the parts of the solution are finally in place, we review the end-to-end solution, and to what extent the patterns supported by automation simplify its delivery.

Forenoon sessions (0830 – 1230):

- Session 1 – Temporality concepts

  Break – 15 min – Morning tea (10:45-11:00)

- Session 2 – Data delivery for consumption

  Lunch break (1230 – 1330)

Afternoon sessions (1330 – 1700):

- Session 3 – Application of business logic

  Break – 15 min – Afternoon tea (15:00-15:15)

- Session 4 – Completing the solution

Prerequisites

- Sufficient understanding of English, as the course language is English
- Understanding of data engineering, for example Data Warehouse and ETL development
- Good understanding of SQL (e.g. joining tables, using window functions)
- Basic scripting / programming awareness (e.g. C#, Python)
- Familiarity with data modelling for Data Warehousing (e.g. CIF, Kimball / Dimensional, Ensemble techniques such as Data Vault, Anchor)

Session modules

Pattern-based design

The direction to move away from manually creating data integration logic, towards a more pattern-based approach directed by the information model, is called Model Driven Design or Pattern Based Design..

Fundamentally this is about cultivating a mindset of flexibility in design by leveraging modular patterns and supporting technologies.

This session provides an introduction of the thinking behind data logistics generation and automation.

- Required components for model-driven Design / pattern-based design
- Styles of data logistics generation
- Guiding principles and requirements
- Family of hybrid modelling techniques
- Overview of the modelling concepts and core entity types
- Overview of core implementation patterns

Data solution architecture

This session takes a step back and looks at the overall design.

Now that there is a foundational understanding of the modelling approach and its intent, the end-to-end architecture can be explained. This will provide a reference point for the advanced topics, and explores design considerations, especially related to the interactions between difference concepts. What needs to be done where, and what are the impacts of certain design decisions?

The following topics are covered:

- Overview of the layers and areas in a data solution architecture
- Architecture options and considerations
- Back-room and front-room operations
- Combining multiple platforms and technologies in a design
- Specifics and requirements of each area in the architecture
- Separation of concerns

Data staging concepts

Getting the data into the data solution is one of the most complex areas in the architecture. This session covers the fundamental concepts that need to be included in any design and explores the impacts on the subsequent layers in the architecture.

- Understanding different data staging approaches (patterns)
- Implications of date/time stamping, where and how to capture the Load Date / Time Stamp – and other dates
- Key requirements for a Staging Layer
- Persistent Staging Area (PSA) considerations
- Preparing to be near-real-time
- Supporting parallel processing
- Change Data Capture (CDC)

Design metadata

With the source and target models available, focus can be placed on the design (mapping) metadata itself. What metadata is required, where is it located and how should you store it?

Regardless whether you are developing a solution yourself or use commercial Data Warehouse Automation software the metadata is the same.

This session explains in detail what needs to be provided and creates an understanding how this metadata fits into the various data logistics patterns.

The focus of this session is to provide the following:

- Overview of the required patterns
- Overview of the required metadata

Code generation

The design metadata captures what data needs to go where, why, and how. Defining a template will drive what will be done to deliver this.

This is a practically-oriented session which includes 'hands-on' exercises.

- Defining code generation templates
- Merging design metadata with templates to generate data solution code

Core Business Concepts pattern

The Core Business Concept (CBC) entities are the single most defining aspect of a data solution, and heavily influence subsequent design decisions.

They are also the most straight-forward patterns to implement from a development perspective. Given the critical nature of the role CBCs play, it is important to ensure the implementation of the data logistics 'just works'.

To achieve this the following considerations are discussed:

- Pattern, structure and implementation
- Parallism
- Technical types of Business Keys
- Key distribution approaches
- Metadata and code generation

Natural Business Relationship pattern

You could look at Core Business Concepts as if they were the 'joints' of the model. In that case, the Natural Business Relationships would be the 'bones'.

Natural Business Relationships manage relationships between business concepts and govern the granularity and the 'Unit of Work' within the model. They require design and implementation choices specifically geared towards this.

This session covers the concepts that are specifically relevant to Natural Business Relationship implementation:

- Pattern, structure and implementation
- Recursive and clustering mechanisms
- Degenerate attributes
- Metadata for code generation

Context pattern

The context entities provide the details ('describe') for Core Business Concepts and Natural Business Relationships. This is also where 'time variance' is managed, where the data changes are captured in time.

The content of this session is geared towards defining a flexible approach for 'tracking changes' and covers concepts such as:

- Pattern, structure, and implementation
- Redundancy
- Row (record) condensing, either considered independent or as part of CDC
- Change merging
- Column scope
- End-dating
- Zero records

- Multi-Active (multi-variant) approaches

Technical considerations

As with any solution it is important to understand how the technology can be configured to support a robust and scalable application. Correctly configured database functionality such as indexing, partitioning and parallelism has a big impact on the effectivity of the data solution.

This session will discuss considerations related to technologies such as:

- Compression
- Partitioning
- Indexing
- Filtering
- Referential Integrity
- Error handling

Control framework

Most organisations have a data logistics control framework in place, and every Data Warehouse Automation software includes one.

This session explains why a data logistics control framework is essential for a reliable data delivery, not only simply for auditing but as an integral part to ensure consistency of delivered information.

The following topics are covered in this session:

- Transaction isolation at application level – how can a data solution guarantee consistency?
- Examples of logical grouping for execution of load processes (options and considerations)
- Rollback and recovery

Testing

Having a reusable, generic library of tests is valuable for quality development as well as preventing regression using DevOps.

We will cover:

- Defining a testing framework
- Creating test cases.
- Applying data checks without judgment

Orchestration, workflows, and parallelism

There are various ways how parallel processing can be implemented. It is the intent to load data as soon as it is available, and this session explains what this means from a design and implementation perspective.

This session will discuss considerations related to technologies such as:

- Batching processes versus independent execution
- Parallelism, considerations and impacts on the solution design
- Process redundancy
- Referential Integrity in a parallel loading environment

DevOps and versioning

This session focuses on how to organise and automate the workflow of activities to physically create the data integration processes and deploy them to a target environment.

Adequate release management has a significant impact on system reliability and the ability to quickly refactor or even virtualise. In this session the various required components to develop the data solution are automatically combined into a delivery.

The following topics are covered:

- The format and integration of metadata
- Automation of data logistics (code) generation
- Automating a process workflow
- DevOps organisation

Temporality concepts

Combining multiple time-variant tables into a single delivery is a key requirement to deliver data to marts. Handling potentially multiple timelines into a single delivery is the main focus point in this session.

From a technical point of view the *point-in-time* approach is explained in detail. This can provide a method to manage performance in delivering the Presentation Layer.

By pushing down the complexities of managing time-variant data into a helper table the performance implications of certain design decisions can be balanced out.

The session covers the following topics:

- Time-variance concepts – 'date math'
- How to join time-variant entities
- Timing issues and how to resolve these
- PIT pattern, structure and implementation
- Stacked versus continuous PIT approaches (temporal considerations)
- PIT concepts for data virtualisation
- Impacts of load order on PIT information

Data delivery for consumption

The presentation layer is (very broadly) defined as anything that is fit-for-purpose. This is on purpose, because the intent of data solution is to support the organisation in the clarification of requirements over time – as opposed to the requirement of having everything available upfront.

- Dimension pattern(s), structure and implementation
- Fact table pattern(s), structure and implementation
- Types of history
- Handling timing issues
- Switching time perspectives – how to match the business expectations?

Application of business logic

Data is not (necessarily) available in the source systems in the way it is ideally made available through the data solution. This means that it cannot be loaded directly and requires alternative patterns to be able to process the data.

Correctly applying the patterns in a clear architecture that *separates concerns* goes a long way in simplifying working with data. With this in place, you have a flexible solution that can automatically be refactored or adjusted in workshops with subject matter experts (SMEs) – while retaining auditability.

The regular entities can be reused to support deriving information (by applying business logic) to provide alternative perspectives of the data available in the data solution.

This session discusses the following items:

- Applying business logic, the 'front-room' versus the 'back-room'
- How to record and manage business logic
- Handling competing interpretations of data
- Technical options and considerations for implementation of transformations
- Driving keys
- Impacts of date/time selection on derived tables