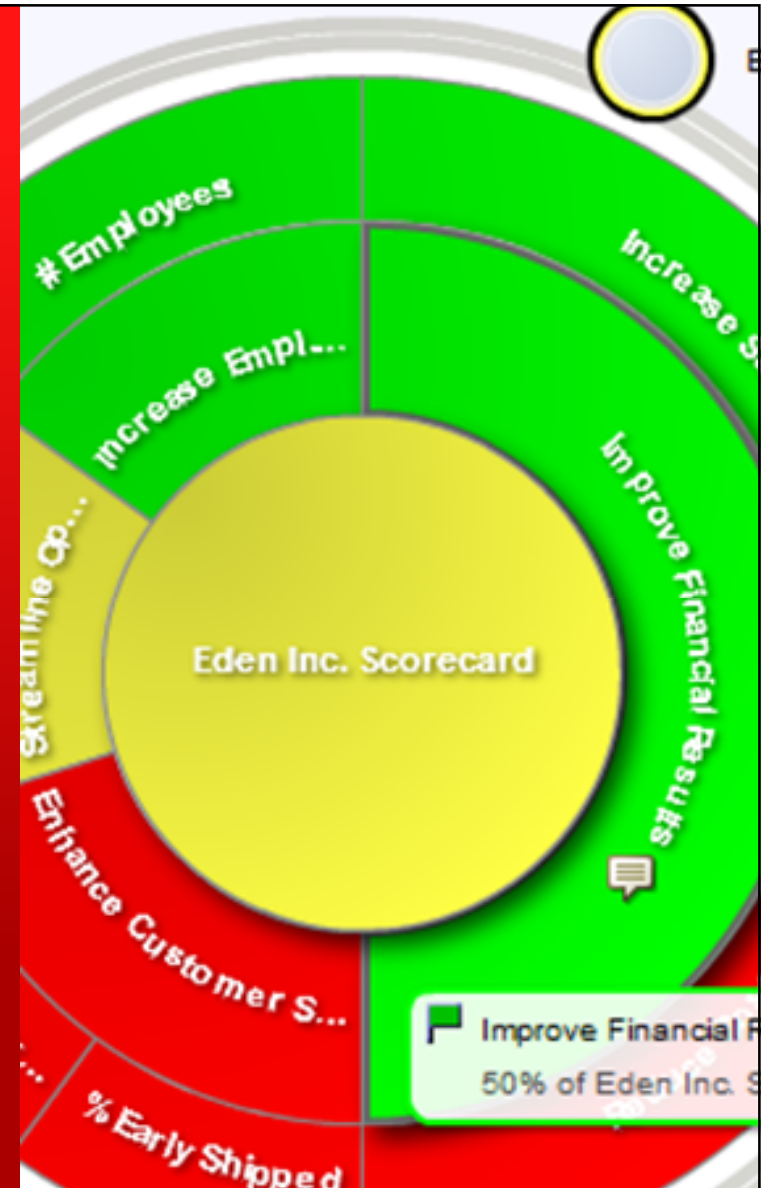


ORACLE[®]

OBIEE: Model First, Build Later

Dan O'Brien



Topics

- Introduction
- BI Design Fundamentals
- Build-First Discourages User Adoption
- Model-First Methodology
- OBIEE Model-First Techniques
- Demo

Introduction

Daniel O'Brien

- Oracle Business Intelligence Pre-Sales in New Zealand
 - Joined in September, 2012
- 15 Years of BI/DW implementation experience (9 Years with OBIEE)
- Most of my experience is in Europe
 - Have an Oracle BI blog www.obieeabc.com
 - NZ Meetup Group <http://www.meetup.com/obi-practitioners-nz/>
- Worn many hats:
 - BI designer, analyst, ETL developer, programmer
 - And User (as investment manager for VC firm)
- BSc & MBA

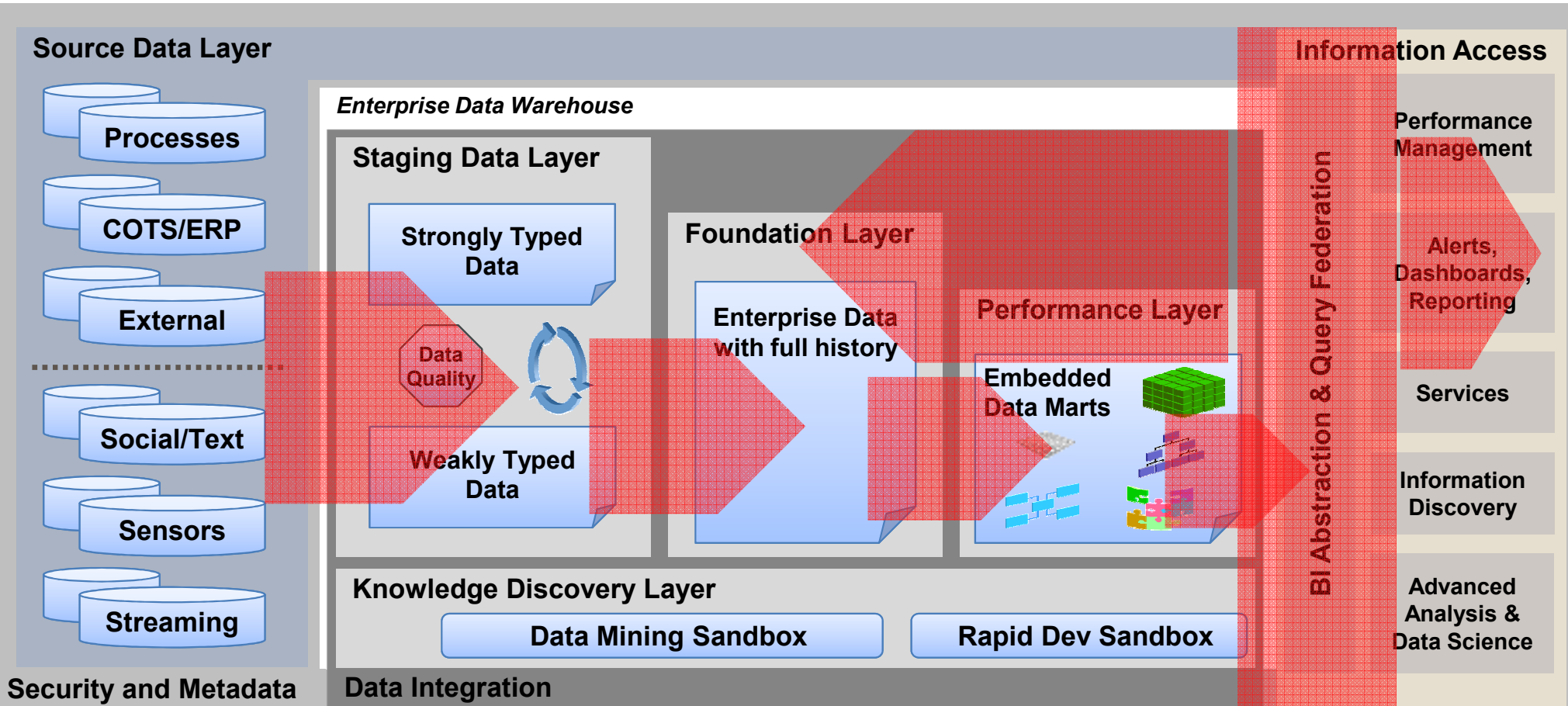
ORACLE

Design Fundamentals

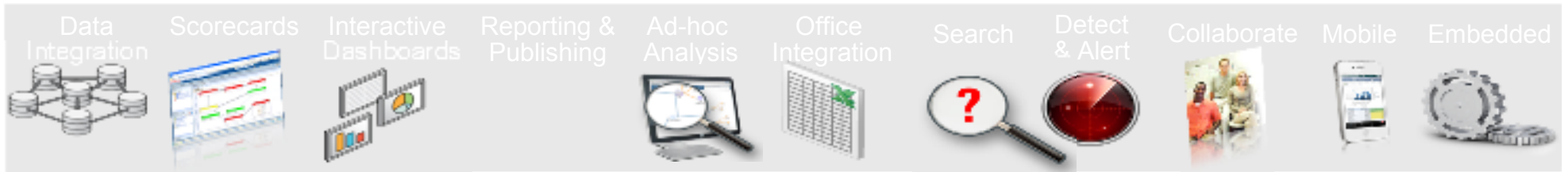
Model First, Build Later

- Business Intelligence has many moving parts
- Two main approaches to Business Intelligence persist
 - Bottom up, traditional warehousing
 - Top down, “user driven” agile
- In most projects, OBIEE sits at the end of a Bottom up approach
- However, OBIEE quite amenable to user driven approach
- **Reverse the traditional flow of data warehouse design**
 - **Make the user pivotal**
 - **Prioritize delivery and user adoption over completeness of solution**

Design Fundamentals: Reference Architecture

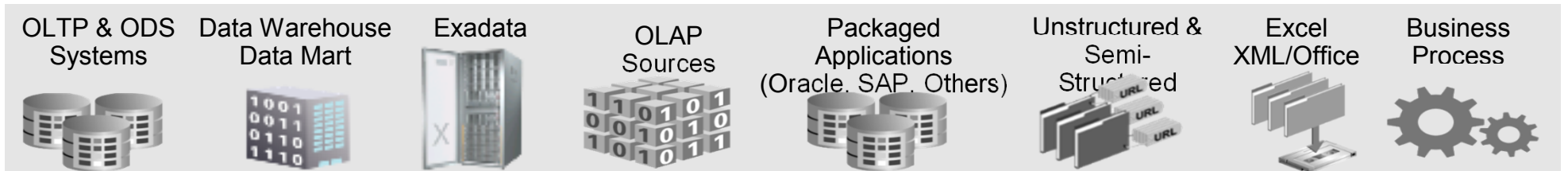


Common Enterprise Information Model



Common Enterprise Information Model

- **Common Metadata Foundation across all Data Sources**
- **Common Security, Access Control, Authorization, Auditing**
- **Common Request Generation and Optimized Data Access Services**
- **Common Clustering, Workload Management, & Deployment**
- **Common Systems & Operational Lifecycle Management**



ORACLE

Time to Business Value

Typical Number of Weeks

New Subject Area Delivery

Analyst

ETL Developer

BI Modeller

Report Writer

Users

Not talking to
Users

7 Weeks Before Users
Get Involved

Weeks

1 2 3 4 5 6 7 8 9 10

User Adoption Suffers

Silent Defection

- Why?
 - Users make do and utilize “back channels” when delivery is slow
 - Business is not static, and highly repeatable - but changing
 - ETL developers usually out of touch with business reality
- Tell-Tale Sign: Large Data Dumps to Excel
- Danger of “Spreadmarts” spreading like the plague
- IT teams have a large “BI Backlog” of requests
- **Bottom-Up OK for static, repeatable but fails for dynamic & ad-hoc**

Model First Development Methodology

ORACLE

Model First Development Methodology: Overview

- Recommended approach for developing Oracle BI repositories
- Driven by business analysis and usage history
- Iterative, top-down approach that focuses on the consolidation and abstraction of core business requirements irrespective of the underlying physical architecture:
 - Build business model first.
 - Integrate with underlying physical architecture.
 - Quickly deploy baseline to end users.
 - Pursue iterative development based on user feedback.

Features of Model First Teams

- Smaller BI Design Team, More Parallelization
- Team is competent in three areas:
 - the business
 - the users
 - the technology
 - focus on all three segments, **multi-disciplinary skills are important**
- Less emphasis on backroom ETL
- More time spent with Users understanding Analytic Applications

Users Involved Early On

New Subject Area Delivery - Model First

Users Involved Week 3

Analyst / Modeller

Users

ETL Developer

Weeks

1 2 3 4 5 6 7 8 9 10

Analyst is also BI Modeller
and Report Writer

Central Tenets of the Model First Development Methodology

- Rapid prototyping
 - Leverage actual subsets or fictitious physical data stores and manageable data volumes to reduce performance issues during development.
- Iterative development and user feedback
 - Leverage prototypes for demos and hands-on sandboxes.
 - Rollout augmented models frequently.
 - Demonstrate responsiveness to feedback.
- Gap analysis
 - Map the business model to actual physical sources.
 - Manage scope and expectations accordingly.
- Gather performance requirements along the way
 - Identify patterns of use, data granularity, user groups, and security constraints.

What is a Business Model?

A business model is a completed matrix that resembles the following graphic, with X denoting key dimensions for a given business process and O denoting minor dimensionality.

		Time	Account	Organization	Product	Geography							
Business processes	Sales	X	O	X	X	X	<table border="1"> <tr><td>X</td><td>Frequently</td></tr> <tr><td>O</td><td>Sometimes</td></tr> <tr><td></td><td>Never</td></tr> </table>	X	Frequently	O	Sometimes		Never
	X	Frequently											
	O	Sometimes											
		Never											
	Forecast	X		X	X	O							
Service	X	X	O	X	X								
Orders	X	X	O	X	O								
Activities	X	X		O	O								

Drill to Levels for More-Detailed Performance Requirements

- Each business process is individually rationalized against the dimensional hierarchies and the user roles to which they apply.

	Time				Organization				Product	Geography				
	Year	Quarter	Month	Day	Level 3 Position	Level 2 Position	Level 1 Position	Level 0 Position	Product Line	SKU	Region	Country	State	City
Sales	X	X	X	O	X	X	X	X	X	O	X	X	O	O
Forecast	X	X			X	X	X	X	X		X	X		

Sales Manager Role

Focus on the Business Model

- Focus on the business model rather than the presentation:
 - Ad hoc reports are typically used once and are not pervasive.
 - Existing reports are useful only when abstracted for their dimension and measure objects.

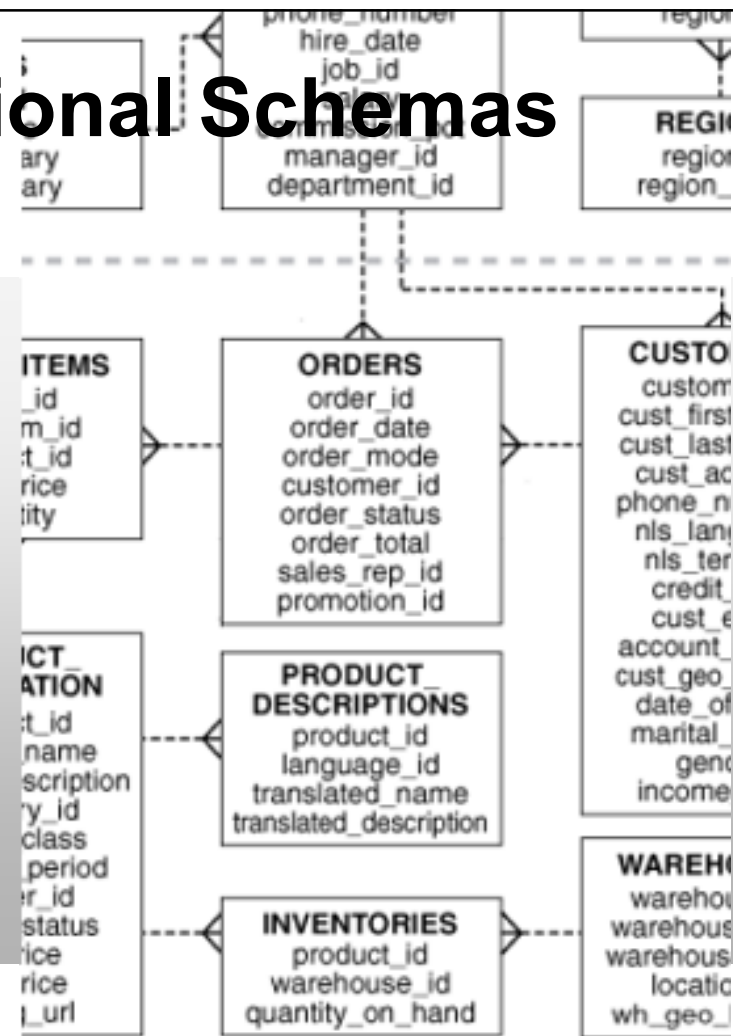
Modelling Techniques

Technique 1: Excel as a Source

- Why?
 - Rapid prototyping
 - Accelerate learning of BI modelling and dashboard design
 - Customers often share demo data in Excel
 - Admin Tool/Excel/BI Server all run on Windows
 - Rewire to Oracle database later/if needed
- Warning: Not production ready. Linux? Query generation + perf.
- Alternative: load XLS into ORCL with SQL Developer
 - But not great if XLS needs mods

Technique 2: Model Transactional Schemas

- Model whatever is available:
 - Source system transactional schemas
 - Foundation layer transactional schemas
 - Staging layer “INMON”-style warehouse
- “Real-Time” ETL
 - Lower cost than cutting ETL mappings
- Useful for *Greenfield projects*



Technique 3: Oracle BI Data Mart Automation

- The Aggregate Persistence Wizard & Summary Advisor automates the aggregate summary process.
 - Use this wizard-driven utility in the Administration Tool to define, populate, store, and map aggregated data stores:
 - Choose the measures that should be aggregated.
 - Choose the dimensions and levels to be aggregated to.
 - Select the data sources in which to physically store the aggregate summaries.
 - Create query performance improvement over normalized, transaction-level physical schemas.

Technique 4: Leverage Oracle BI “Legless” Applications

- Oracle BI Applications are complete, prebuilt BI solutions:
 - The prebuilt Oracle BI Applications repository contains business models that can be mapped to different physical data sources.
 - Value can be realized without Oracle Business Analytic Warehouse and ETL components.
- Redefine the Oracle BI Application Physical layer objects by using the “opaque view” feature:
 - Use `SELECT` statements.
 - Deploy opaque views via BI Server Administrator
 - Materialize as required.



Technique 5: Model with Endeca Information Discovery

Building applications in days, not months

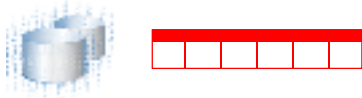
Diverse and changing information integrated and enriched via ETL

Automatically unified in Oracle Endeca Server – no predefined model required

Drag-and-drop application composition in Studio

Interactive search, navigation and visualization for exploration and analysis

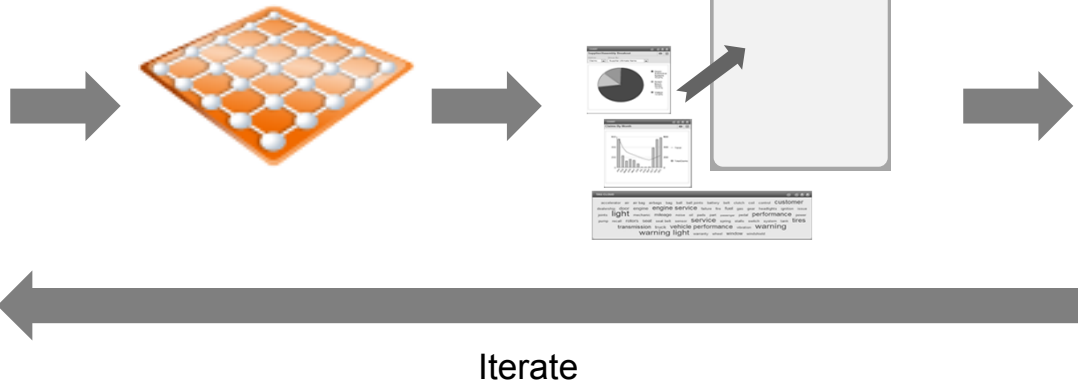
Structured



Semi-Structured



Unstructured



Extend Business Analytics with Unstructured Data

Introducing Oracle Endeca Information Discovery

Oracle Business Intelligence

Best platform for integrated ROLAP and MOLAP

Oracle Endeca Information Discovery

Best platform for Unstructured Analytics



Structured Data

OLTP & ODS Systems

Enterprise Applications (Oracle, SAP, Others)

Data Warehouse & Data Marts



Unstructured Data

Websites

Content Systems, Files, Email

Social Media

Big Data



Summary

Make Users Pivotal

- To Drive Adoption
- Start with Model and Build Later
- Works well for
 - Migration Projects
 - Greenfield Projects
 - Mock-ups and Demos



Demos

Q&A