A Framework for Building Resilient Data Warehouses using a Mandala Topology Architecture

Michel JACQUES
Information Assembly SPRL, Brussels, Belgium

First European Business Intelligence Summer School (eBISS 2011)
July 3 - 8, 2011 Paris, France

Introduction

Constructing ETL work for a DW project is a complex affair, one that requires planning. This framework uses a DW architecture based on a Mandala Topology. At its core is an agile, step-by-step approach to identify ETL work units:

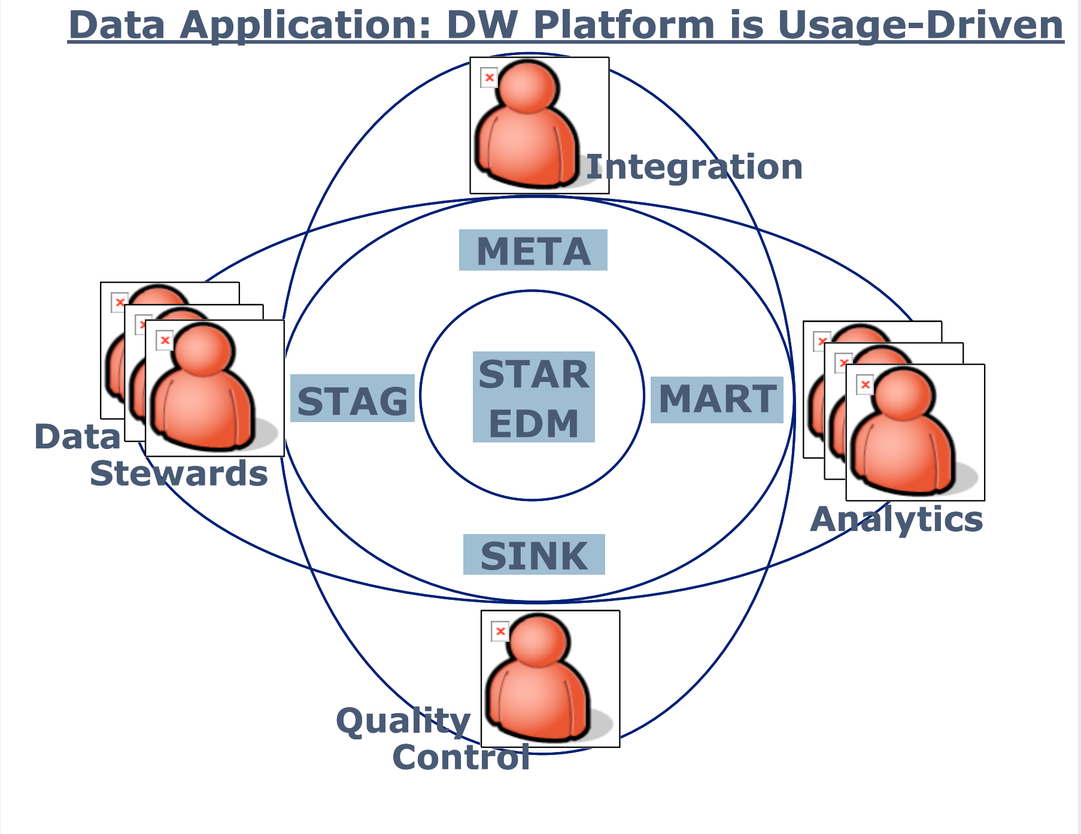
1. Identifying the external users
2. Positioning main data flows,
3. Decomposing a flow internal layers
4. Incorporating them within the DW application platform (Mandala)
5. Extending conceptual EDM with a functional level of detail
6. Classifying data model entities and their dependencies
7. Combining the Functional ER model with Topic Areas & Tiers
8. Enumerate the ETL work units in matrix format

The method focuses on the topological relationship between all the DW artifacts, in order to comprehensive improve planning & design of DW.

1.Users Perspectives & Interfaces

Each **user** type has a different **perspective** and **role** towards the DW. Existing users include: data stewards, analytical end-users, quality control masters, DW administrators & integration managers. Each user has his own set of information requirements and accesses the DW via a **specific interface** (STAG, META, MART, or SINK). The STAR can only be accessed indirectly via the other interfaces to ensure **data integrity & security** and **prevents dependencies** caused by different user demands. The users are part of an iterative feedback loop improving future data content and quality.

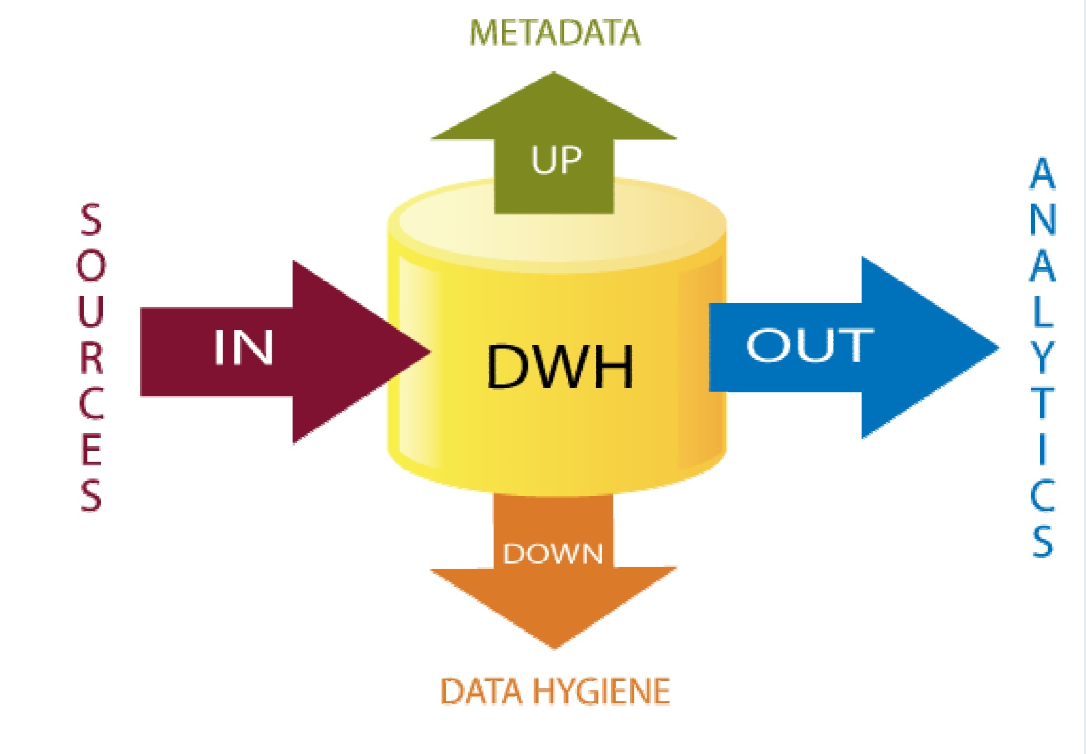
Data Application: DW Platform is Usage-Driven



Hence a multi-perspectives/faceted data warehouse architecture acting as a crossway between these end-users is comprehensive and non-discriminatory at an organizational level.

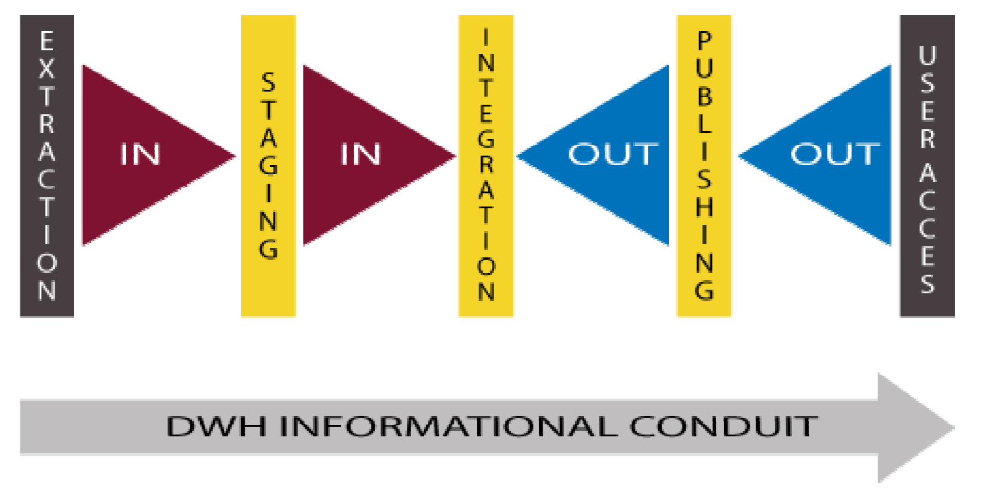
2.The Four Major Streams of Data

Source data is in an unrefined state (to various degrees) that must have its **data elements** differentiated into a DW core data model (IN) in order to able to recombine them effectively afterwards for analytics (OUT). During this transition, two other types of data elements are produced: metadata (UP) and erroneous data (DOWN) streams. **No data should be lost in this closed system**. Thus the position & direction of a data stream determines its purpose, the means, and its destination. This naturally leads to asymmetries between streams, which must be accounted for in the ETL design..



"Just about every [DW] process has side effects; but they can be deliberate and sustaining instead of unintentional and pernicious…and we can also be inspired by it to design some positive side effects to our own enterprises instead of focusing exclusively on a single end." (p.80)
Ref: W. McDonough and M. Braungart, *Craddle to Craddle*, North Point Press, NY, 2002

3.The Five ETL Layers & Chirality



The purpose of an **ETL** is to increase the **integration** & **analytical capability** of sourced data. An ETL data path consists of 5 layers, each conducting a different set of data transformations. The first two "IN" stages integrate the data to enable **multiple interpretations**, while the last two "OUT" stages specialise the data such that it becomes **fit-for-purpose**. This mirror-effect is referred to as ETL **chirality**.

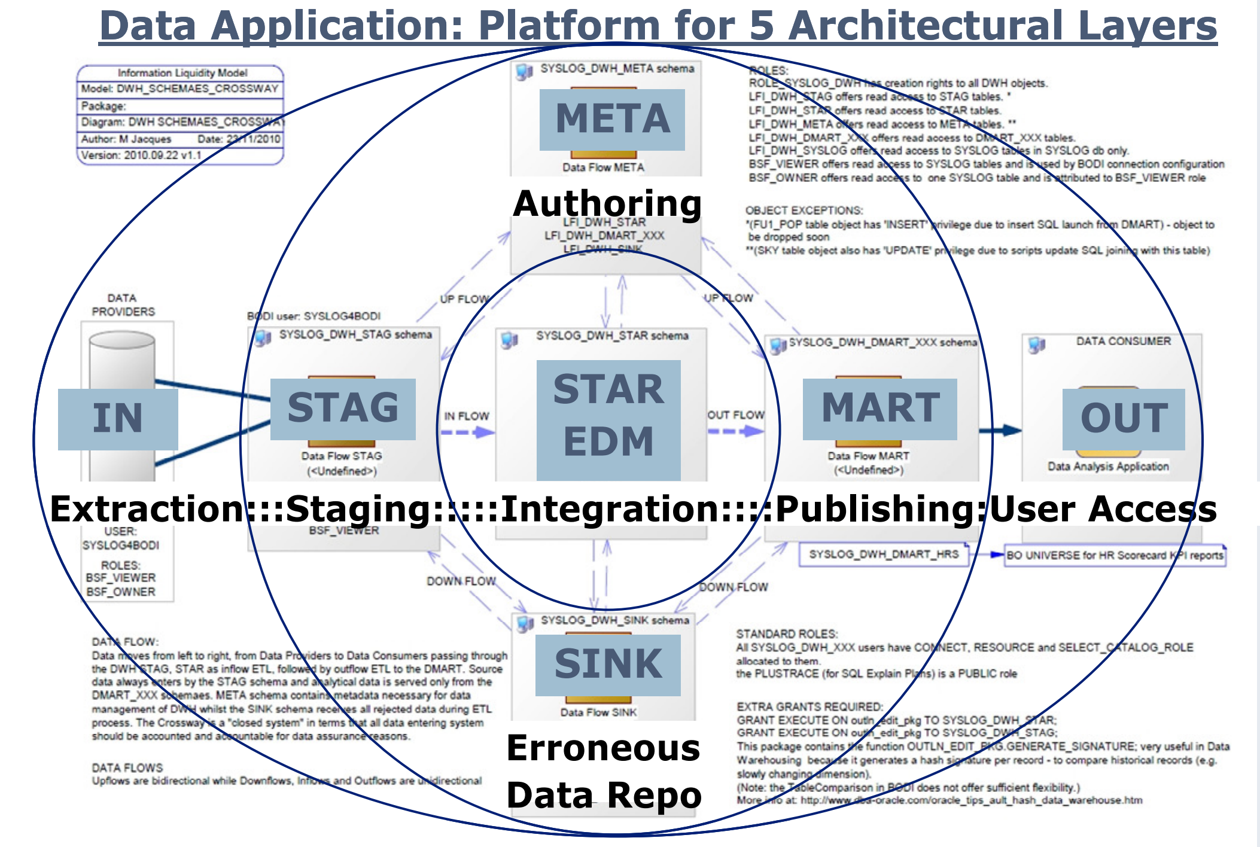
The following are important elements when selecting the most appropriate data modelling technique to use: a) the degree of **convergence** built into the data during ETL; b) the number of unique **pathways** in the dimensional model; c) increased data flow **resilience** by decreased data reliance; and d) ability for **decoupling** of model components.

Data disturbances will occur either from external sources in an unexpected, subtle or extreme manner, which requires a faster data recovery by minimising data reload to only what is relevant. Resilience also involves cyclic transfer of information across data applications reinforcing each system data quality and monitoring usage.

4.DW Mandala Topology Architecture

The Buddhist Mandala metaphor helps visualize an **architecture** where topological relationships between DW **components** including: data modeling, ETL data flows, and surrounding actors & applications; functionally **interact**. The architecture is similar to a road crossway that gives essential **context** and **movement** to data and operations. The context is composed of 5 distinct locations (STAG, META, SINK, MART, & STAR) that provide a **clear logical structure** determining data, flows, modeling patterns, access and security, user groups, and integration methods. Moreover, locations enable data persistence facilitating data recovery and transformations.

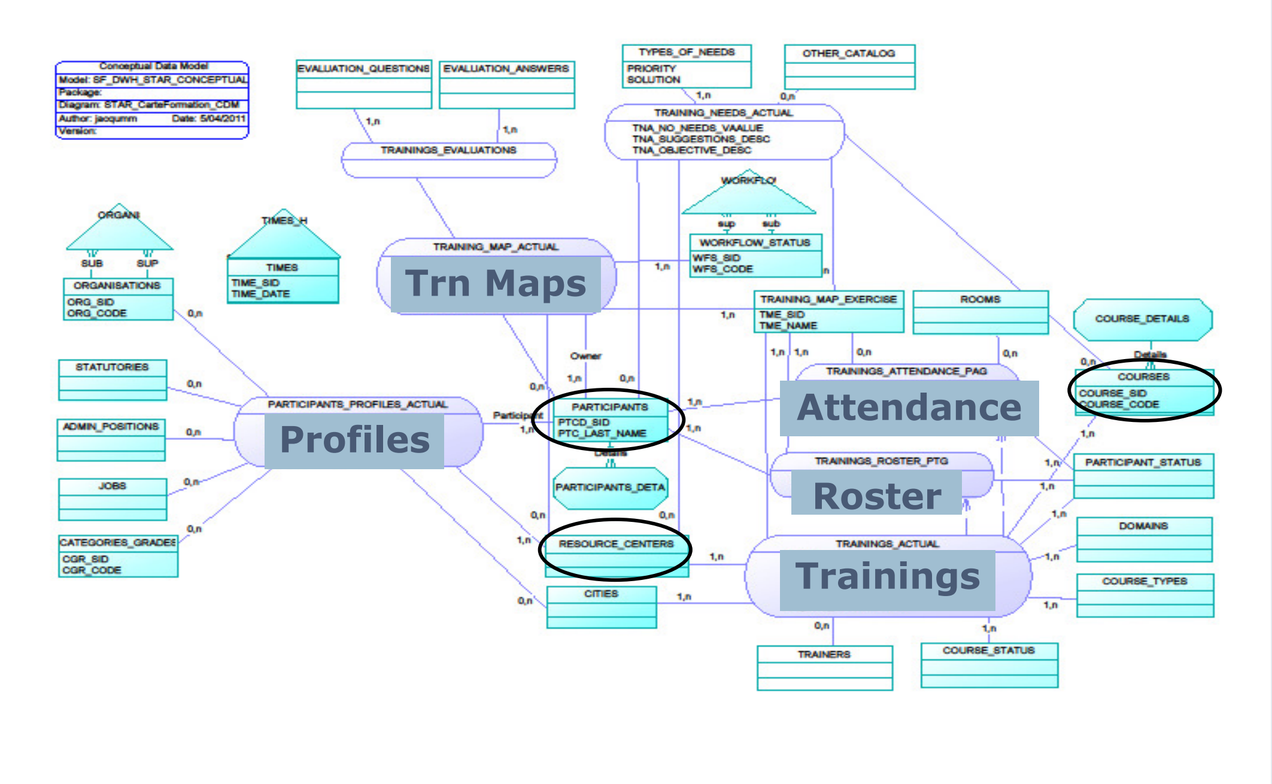
Data Application: Platform for 5 Architectural Layers


META
Authoring
IN
STAG
STAR
EDM
MART
OUT
Extraction:::Staging:::::Integration::::Publishing:User Access
SINK
Erroneous
Data Repo

5.Functional ER Data Model

The conceptual model is business-oriented, while the logical model is focused on content and application. The **functional data model**, proposed herein, places itself in the gap between the two. It extends the number of entity types from the initial fact and dimension with new types for holding hierarchies, dim. ids, details & associations. This improves history-keeping and makes the core data model more resilient while decoupling the associated data flows.

Enterprise Data Model: Conceptual Model



There is a need to extend our "vocabulary of forms" when modelling. This involves adding functional features so as to harmonise "form with function" and thus achieve a greater **decoupling of DW artefacts, whilst maintaining data cohesion.**

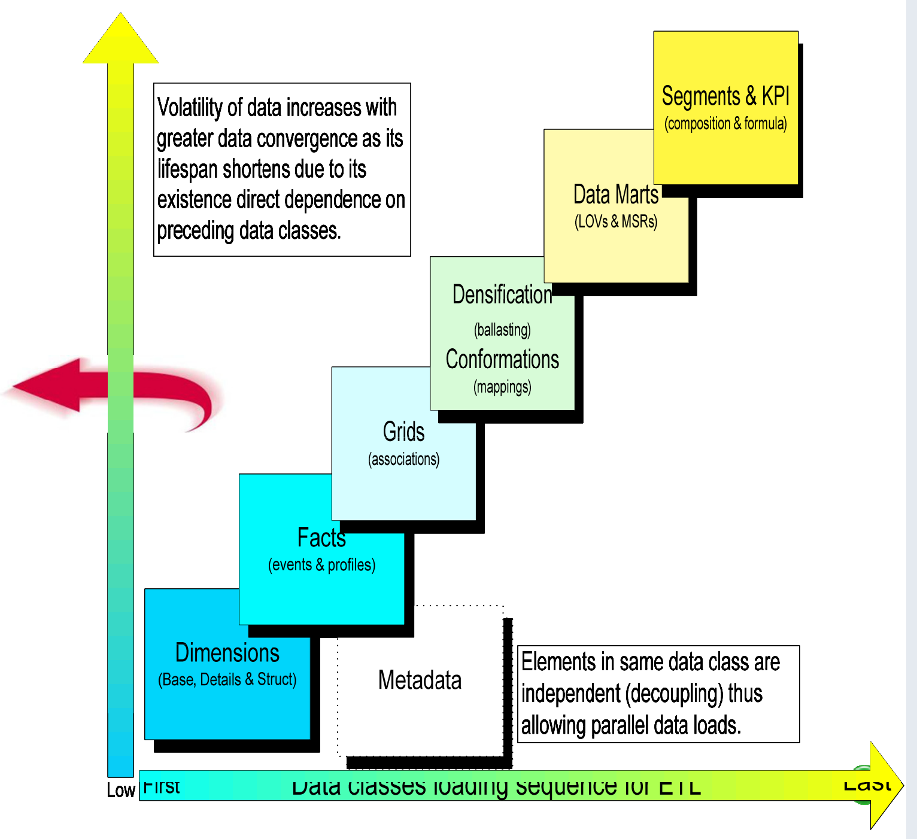
6.Volatility of Functional Entities

This functional data model applies additional data entity **classes** giving it increased ETL **flexibility**. The classes follow a step-wise approach whereby they become increasingly **volatile** (data susceptibility to change). Since data in lower classes is used to derive new data in upper classes, the data volatility in lower class will be less than that in upper classes. Volatility determines which data flows are performed in parallel or in sequence. This principle drastically reduces ETL execution and development time.

ETL Decomposition Process: Classes as Tiers Revisited

DEPENDENCY LAYERS as TIERS

DATA CLASSES	TIERS
Metadata (static)	T00
Dimensions base, details & struct)	T01
Facts (events & profiles)	T02
Grids (associations)	T03
Densification & Conformations (ballasting & mappings)	T04
Data Marts (lov, hier & msr)	T05 T06 T07
Segments & KPI (compositions & formula)	T08 T09
Topic Area Promotion	T99



Volatility of data increases with greater data convergence as its lifespan shortens due to its existence direct dependence on preceding data classes.

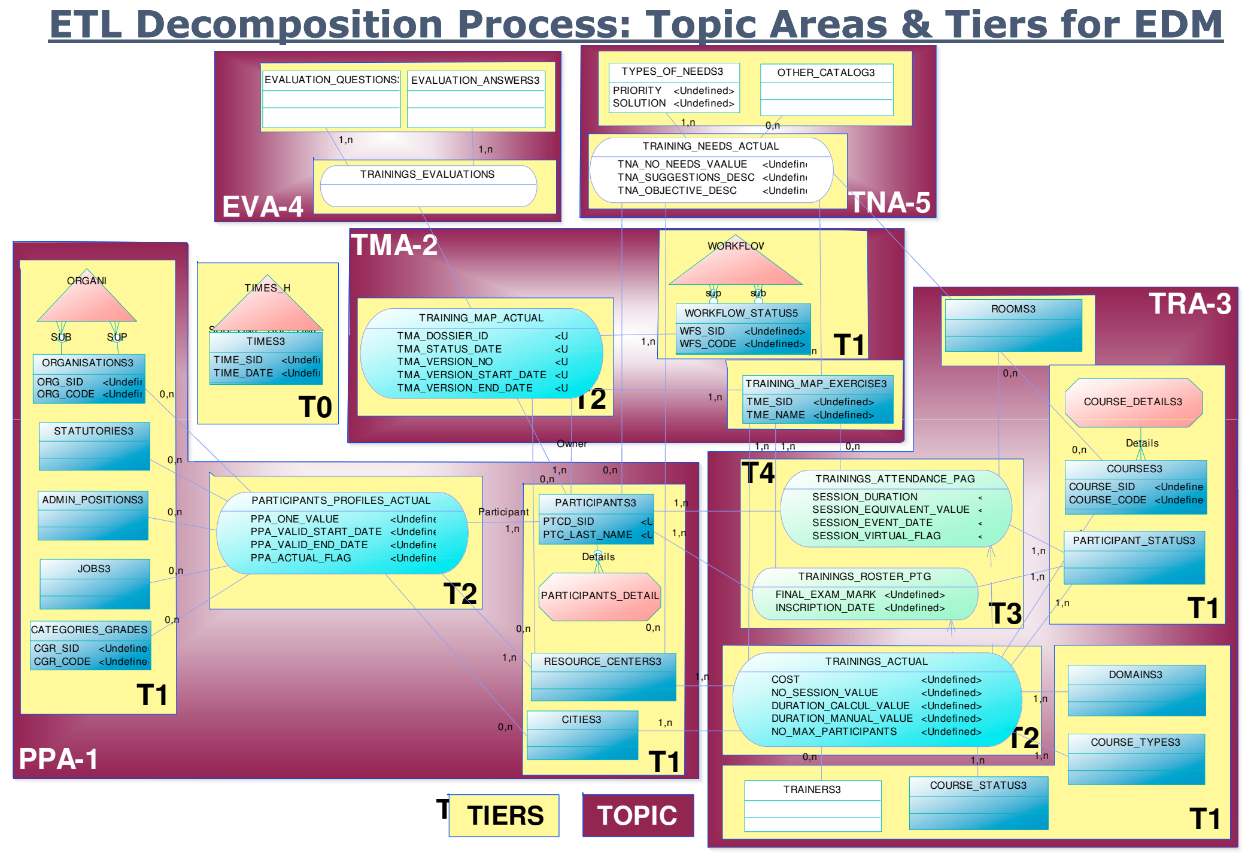
Elements in same data class are independent (decoupling) thus allowing parallel data loads.

The 6 classes are mapped to 9 tiers that maintain volatility.

7. Agile DW Meets Functional ER Model

The ETL decomposition process requires an **entity** having both a class and a theme. For each entity the process allocates: a) a **tier** corresponding to a class; b) a **topic area** corresponding to a theme; and c) a **priority** corresponding to prevailing business needs. An entity defines the work unit in which it is contained. A work unit is the smallest functional artefact determining **granularity** of resource allocation**.** Regrouping work units is as follows: Work unit >> Module >> Topic Area >> Enterprise Data Model (EDM).

ETL Decomposition Process: Topic Areas & Tiers for EDM



Concept of Tiers and Topic Areas is borrowed from R. Hughes' book: *The Agile Data Warehousing*, iUniverse Inc, Bloomington, 2008

8.Work Matrix from Method & Conclusion

Development progress follows an iterative and mostly top-down approach: one topic area, one module, one tier at a time, although units within same tier can be developed in parallel. Once there is a Reference Model (built prototype) for a Tier work unit, estimates for all units can be based on the reference model and adjusted according to variable difficulty factors. Once completed & tested, work units modules are then promoted to next dev. environment.

ETL Decomposition Process: Work Units and Modules

PROGRESS FLOW by DATA CLASSES, TIERS & REFERENCE MODELS, TOPIC AREAS, MODULES and WORK UNITS

DATA CLASSES	TIERS REF.MODEL		PPA : UNIT		TMA : UNIT		EXPERTISE
Metadata (static)	T00	T0-S: static	Meta	PPA-Module1	Meta	TMA-M1	Data Modeler
Dimensions (base, details & struct)	T01	T1-B: base T1-D: detail T1-S: struct	PAD ORG PTC … PTCD ORGS		WFS TME WFSS		ETL Developer * inflow
Facts (events & profiles)	T02	T2-E: event T2-P: profile	PPA		TMA		
Grids (associations)	T03	T3-A: assoc.					
Densification & Conformations ((ballasting & mappings)	T04	T4-B: ballast T4-M: map		PPA-M2	PTC	TMA-M2	* inflow * outflow
Data Marts (lov, hier, & msr)	T05 T06 T07	T5-L: lov T6-H: hier T7-M: msr	PADL ORGL PTCL … ORGH PPAM	PPA-M3	WFSL TMAM	TMA-M3	* outflow ETL Developer
Segments & KPI (compositions & formula)	T08 T09	T7-C: compo. T8-F: formula	POP	PPA-M4	CF	TMA-M4	BI Developer
Topic Area Promotion	T99	T99-DP: deployment process	DEV ACC STRES PROD	PPA-M5	STRES	TMA-M5	Release Engineer

Conclusion: The advantages of implementing such a topological architecture include: greater scalability of additional data themes, enhanced performance of data flows, increased resilience of decoupled artifacts, sturdier quality control, and lower operating and development costs. The framework provides a comprehensive, reproducible, and proven DW architecture solution.

Ecole Centrale Paris, Châtenay-Malabry