Developing Data Warehouses with Quality in Mind

Yannis Vassiliou

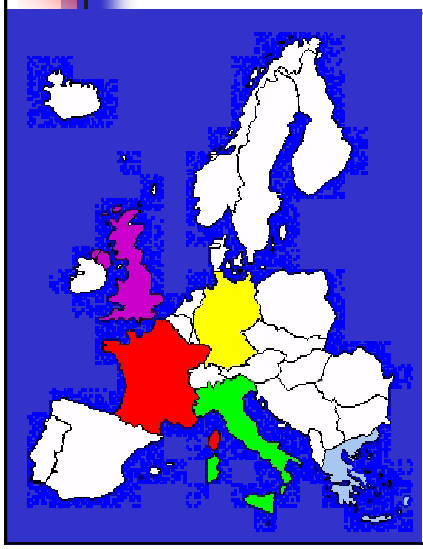
National Technical University of Athens

Workshop on Data Quality

OUTLINE

- Introduction – Motivation
- The Data Warehouse Metadata Framework Developed
  - Architecture, Processes, Quality
  - Models
- Employing the Framework
- Conclusions

Foundations of Data Warehouse Quality- DWQ Project



National Technical University of Athens (NTUA)

Informatik V & Lehr- und Forschungsgebiet Theoretische Informatik (RWTH-Aachen)

Institute National de Recherche en Informatique et en Automatique (INRIA)

Deutsche Forschungszentrum fór kónstliche Intelligenz (DFKI)

University of Rome «La Sapienza» (Uniroma)

Istituto per la Ricerca Scientifica e Tecnologica (IRST)

University of Manchester (UMan)

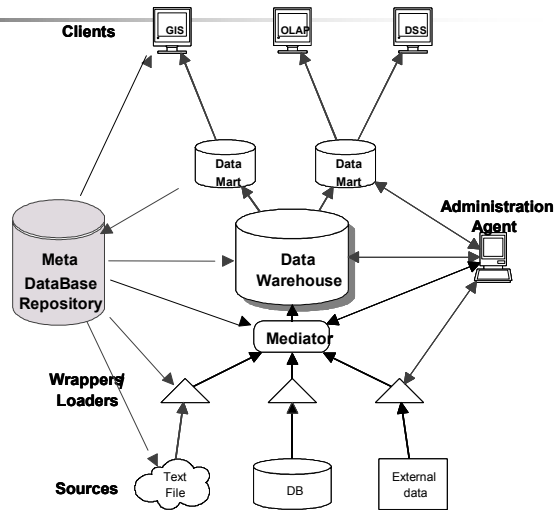
Introduction – Motivation

- Contribute to the systematic understanding and usage of the interplay between QUALITY FACTORS and DESIGN / EVOLUTION OPTIONS in Data Warehousing *(Objective)*
- Develop comprehensive DW Foundations (Framework), Prototype and Evaluate them *(Achievement)*
- Enriched Meta data management facilities with embedded analysis and optimization techniques *(Key Methodology)*

Standard DW Architecture

Examples:
Microsoft Repository
Metadata Interchange Specification (MDIS)

control and manage metadata for OLAP databases.



Standard DW Architecture

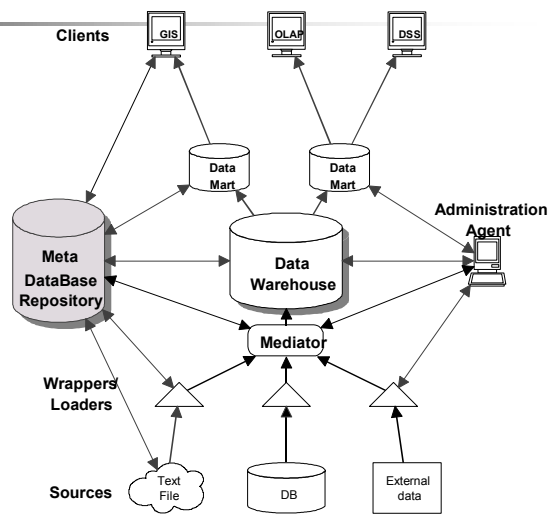
PRACTICAL QUESTIONS not Handled in the Traditional Architecture:

-- *How come the information from the DW is not the same to the one coming from sources?*

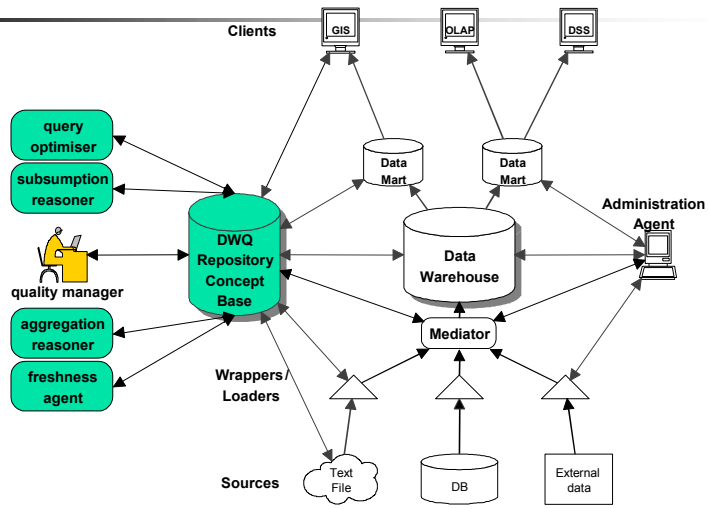
-- *What is the effort required to get in the DW information not currently available?*

-- *If I want 100 % correct data in my DW, how do I design it? how often do I refresh it?*

-- ....



DWQ DW Architecture



A Small Motivating Example

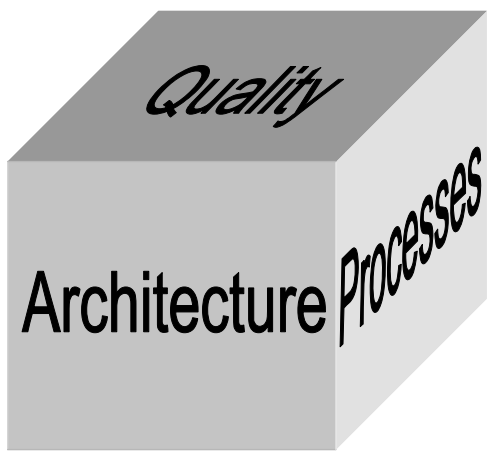
- MINISTRY of HEALTH (Greece)
- Data Warehouse:
  - Sources = COBOL files for all the medical centers in Greece (~2400)
  - Transformation and Cleaning Tasks
- Quality requirements (Goals)

«Achieve 100% completeness and consistency of data»

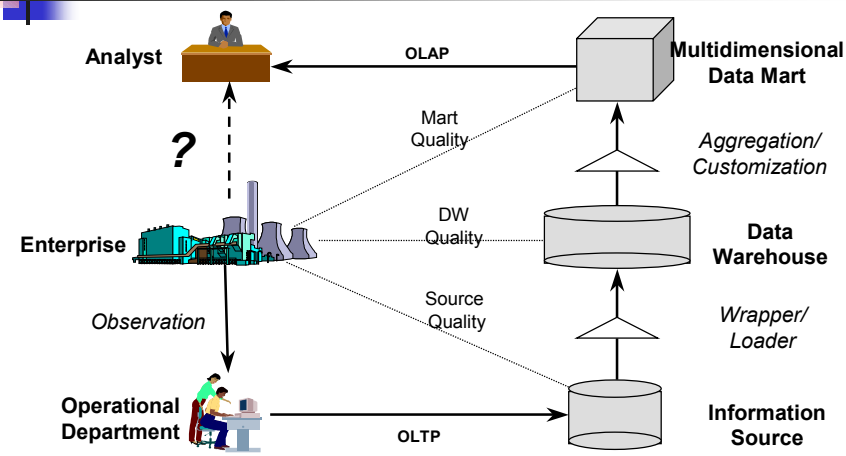
Metadata Framework

- Introduction – Motivation
- *The Data Warehouse Metadata Framework Developed*
  - *Architecture, Processes, Quality*
  - *Models*
- Employing the Framework
- Conclusions

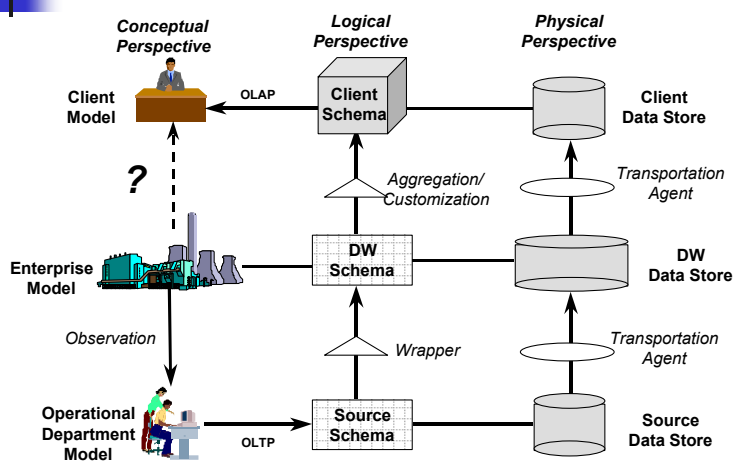
Viewpoints of a DW



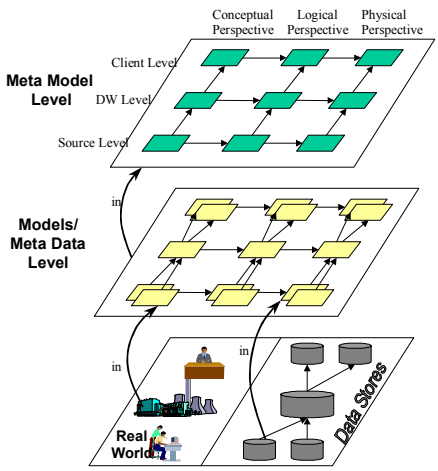
Architecture Model: Step 1
Enterprise Version
of the Traditional DW



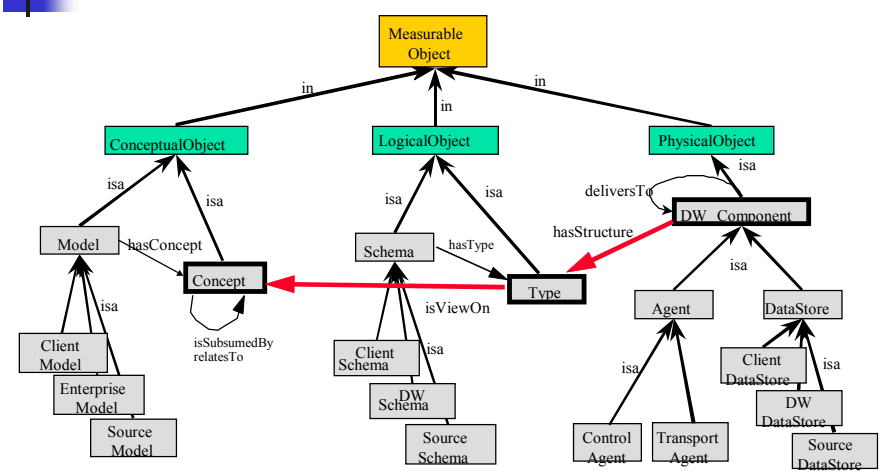
Architecture Model: Step 2
Enterprise Version (Meta level)
Extending the Traditional DW



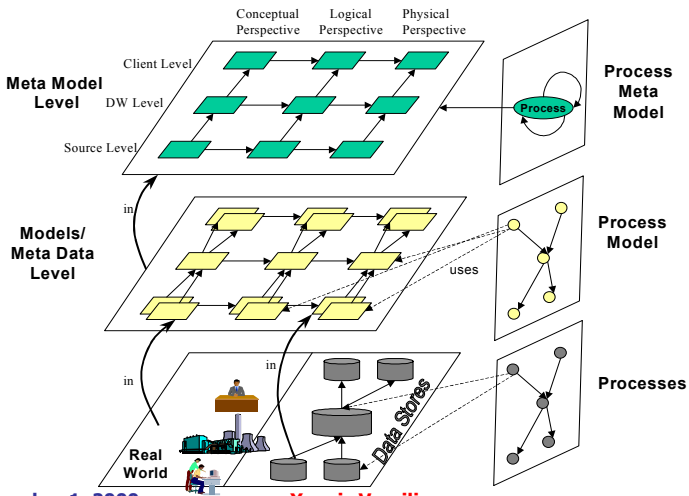
Architecture Model - Instantiation



Architecture Model: Step 3 Structure of the Meta Model as implemented in ConceptBase / Telos



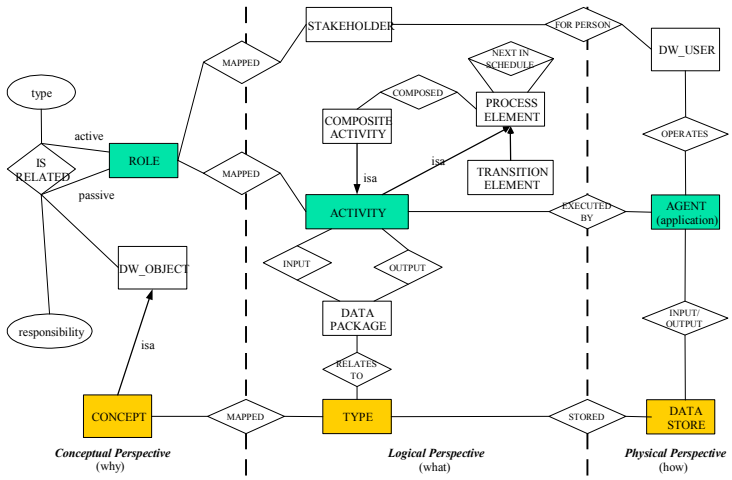
Process Meta Model: Step 1 Capturing the Dynamic Aspects of the Architecture Model (static)



DW Process Meta Model

- Workflow Reference Model (made less abstract to fit in the DW case, e.g.: capture schedules, relationships with data)
- Strategic Dependency Model (conceptual)
- Processes: Cleaning, transformation, transfer, computation
- ROLE – ACTIVITY – AGENT

Process Meta Model: Step 3
DW Operational Process Meta Model



Quality Model

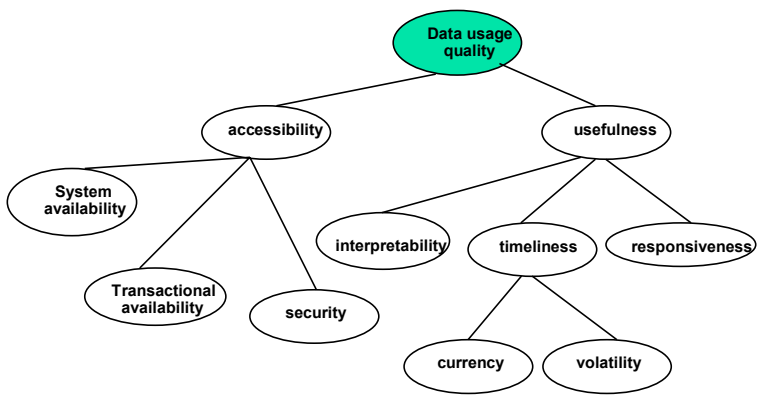
- Quality in a Data Warehouse
  - Quality of Data
  - Quality of Processes
  - Quality of Service

  At all perspectives
- Establishment of Quality aspects (dimensions)
  - Scientific vs. Pragmatic (user defined)

Quality Model

- Concepts:
  - Measurable Object (e.g. logical schema of source)
  - Quality Goal (e.g.,improve availability of source A)
  - Quality Query (decide whether a quality goal is achieved)
  - Quality Dimension (e.g., "availability", "correct")
  - Quality Factor (measurement)
  - Stakeholders (decision makers, designers, administrators, programmers)

Quality Dimensions Example: Data Usage



Quality Factors by Perspective

Conceptual Perspective	*Logical Perspective*	*Physical Perspective*
• Completeness • Redundancy • Consistency • Correctness • Trace ability of Concepts and Models	• Usefulness of schemas • Correctness of mappings • Interpretability of schemas	• Efficiency • Interpretability of schemas • Timeliness of stored data • Maintainability/ Usability of software components

- Questions and metrics for each quality factor ?
- Predictive models of quality impacts and trade-offs ?
- Can the results be mapped back into data warehouse practice ?

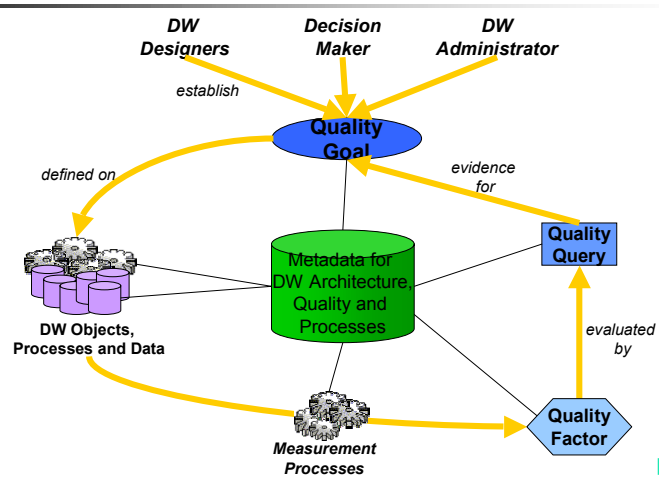
Quality Factors - Metrics

Factor	Methods of measurement	Metrics
Schema quality		
Correctness	final inspection of data warehouse schema for each entity and its corresponding ones in the sources	number of errors in the mapping of the entities
Complete-ness	final inspection of data warehouse schema for useful entities in the sources, not represented in the data warehouse schema	number of useful entities, not present in the data warehouse
Minimality	final inspection of data warehouse schema for undesired redundant information	number of undesired entities in the data warehouse
trace ability	final inspection of data warehouse schema for inability to cover user requirements	number of requirements not covered

Quality Factors – Metrics (Data Usage)

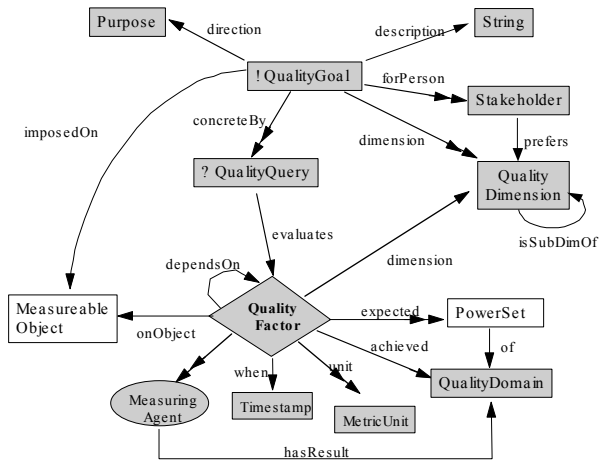
Data Usage Quality	*Logical Perspective*		*Physical Perspective*	
	Schema	Type	Agent	Data Store
Accessibility	Is the schema definition accessible by the users?	Is the type visible and accessible for users?	Is the network sufficient for delivered data?	Is the data store accessible?
Availability	Frequency of updates	Frequency of updates	Response time	Uptime of data store, response time
Security	Level of security (access rights)	Level of security (access rights)	Are there physical access restrictions?	Is the store able to prevent unauthorized access?
Usefulness	Is the schema used by any users?	Is the type used by any users?	Is the data delivered by the agent really used in the destination store?	Is the data in this store queried by a user?
Interpreta-bility	Is the schema understandable?	Is the type understandable?	Is the data delivered understandable?	Is the data stored understandable?

Quality Meta Model Management An Adapted GQM Approach



[Jarke et al., IS99]

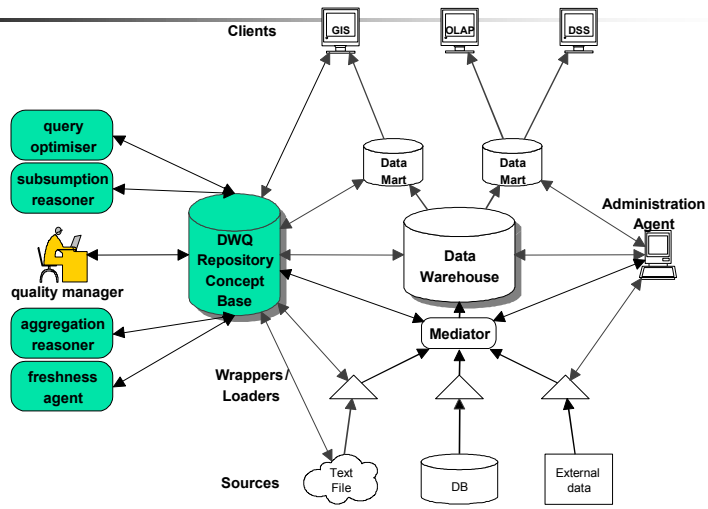
The DWQ Quality Meta Model in ConceptBase



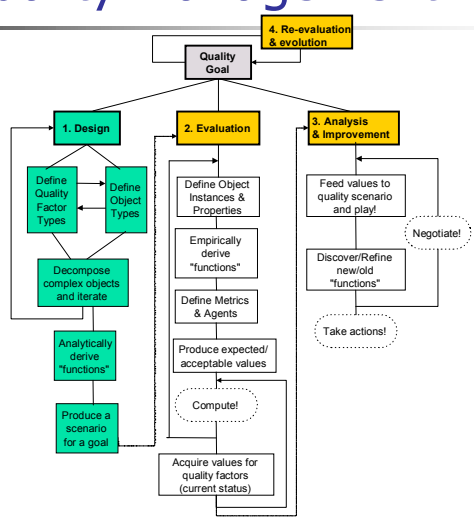
Metadata Management

- Introduction – Motivation
- The Data Warehouse Metadata Framework Developed
  - Architecture, Processes, Quality
  - Models
- *Employing the Framework*
- Conclusions

Employing the Framework -Mapping the Architecture and Models to a Traditional DW

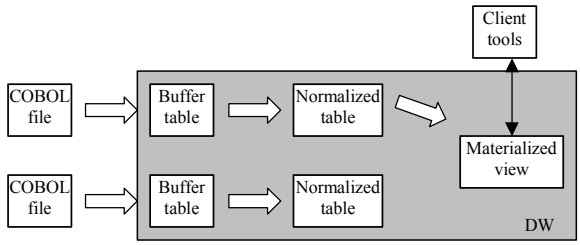


Methodological Approach for Quality Management

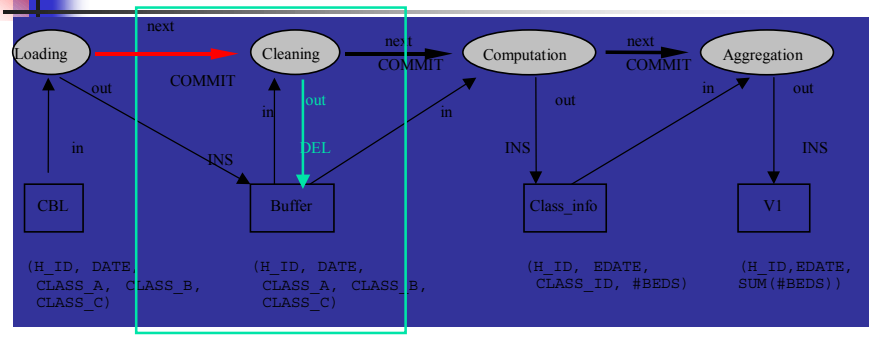


Employing the Framework Running Example

- Ministry of Health Example
- Successful detection of reasons for the *inconsistencies* between DW data and source (legacy system) data



Performing the processes




```
SELECT *
FROM BUFFER B1
WHERE EXISTS (SELECT B2.H_ID, B2.DATE FROM BUFFER B2
                    WHERE B1.H_ID = B2.H_ID AND B1.DATE = B2.DATE
                GROUP BY H_ID,DATE HAVING COUNT(*)> 1)
```

DW Process Quality

Role	Quality Goal	'why?'	*Conceptual*

Achieve 100% consistency of the information to be given to the minister!

Activity	Quality Query	'what?'	*Logical*

Is the propagation activity performing properly?

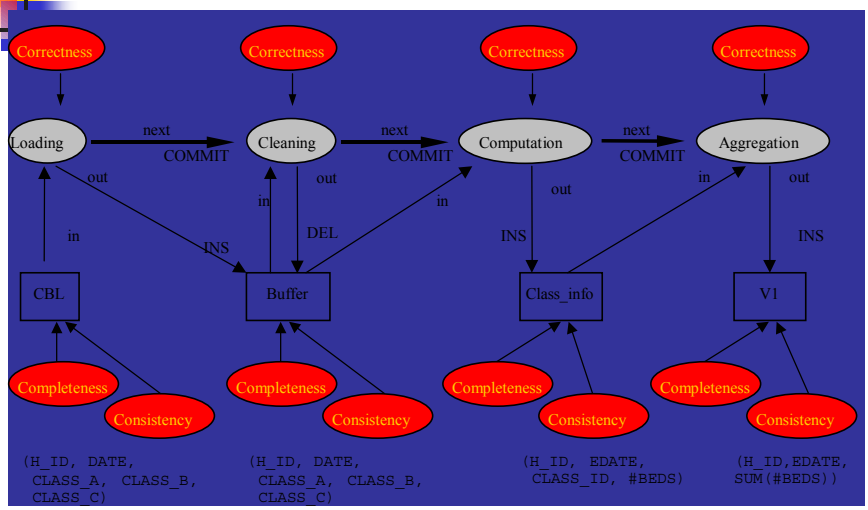
Agent	Quality Factor	'how?'	*Physical*

Correctness of software processes (performed with white box testing)

Analysis of Quality Factors

Quality Dimension	DW objects	Primary Quality Factors	Derived Quality Factors	Design Choices
Consistency	- COBOL Source file - Buffer table - Normalized table - Materialized View - Loading process - Cleaning process - Computation process	- Consistency of a data store - Completeness of a data store - Correctness of an application	- Consistency of a data store - Completeness of a data store	- Data flow - Chosen source files

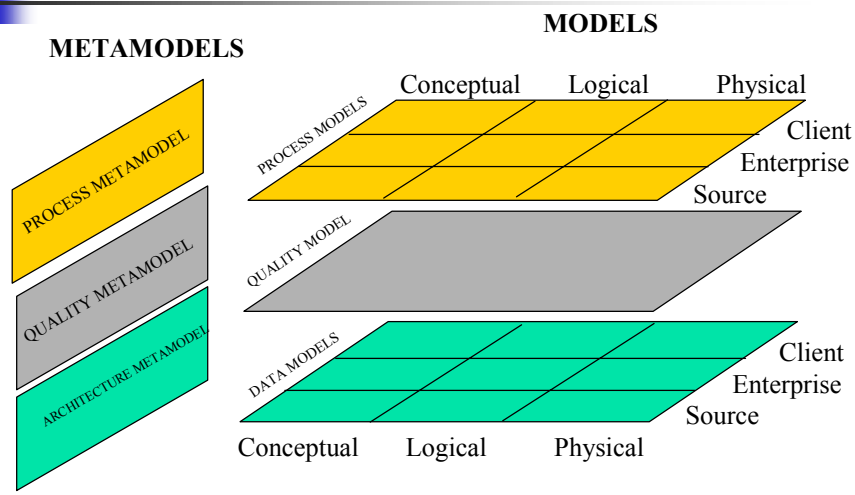
Quality Factors



Conclusions

- Summarizing the Approach
- Formal Results
- Tools Developed
- Several Applications by all partners (TEAM-4, DB / GMI, Telecom Italia, etc.)

Summarizing the DWQ Approach
Architecture, Processes, Quality



Key Formal Results on Quality Impacts

- conceptual: description logic theory and tools for complete reasoning about the relationships between source, enterprise, and client models (Rome, Manchester, Aachen)
- conceptual/logical: containment, satisfiability, and rewriting of queries over views with & without aggregates (DFKI, Rome)
- logical/physical: incremental cost-based optimization of view materializations (Athens)
- physical: detailed impact analysis of replication and refreshment policies (Aachen, INRIA)

Tools Developed