ORE: An Iterative Approach to the Design and Evolution of Multi-Dimensional Schemas

Petar Jovanovic[1], Oscar Romero[1], Alkis Simitsis[2], and Alberto Abelló[1]

1: Universitat Politecnica de Catalunya, BarcelonaTech
2: HP Labs, Palo Alto

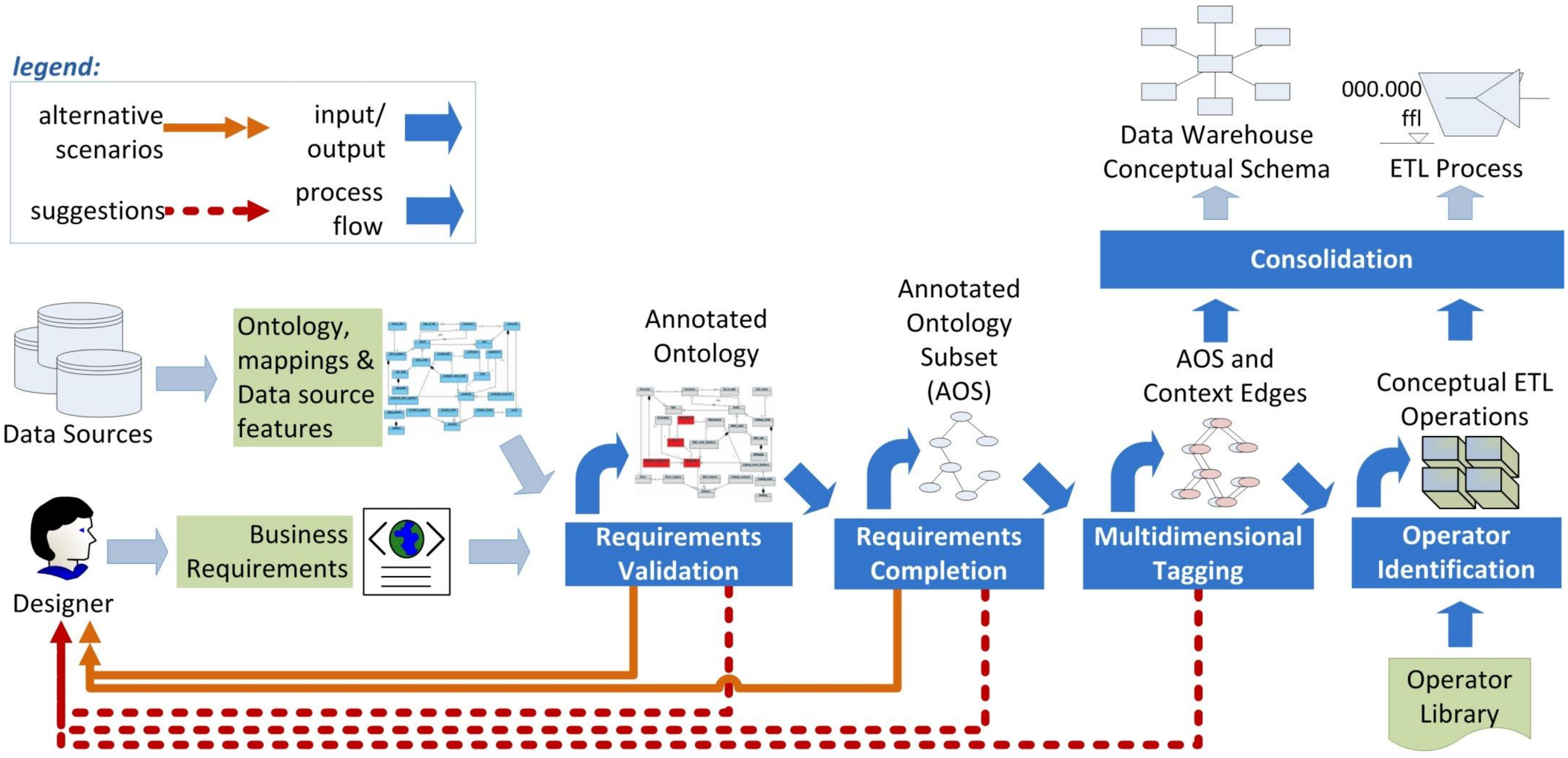
Problem: Building a DW system

- Complex and evolving business environments
  - Constantly posed information requirements
  - Semantics and heterogeneity of the underlying data sources
  - Monolithic approach not realistic
- Necessary optimization and reuse
- Expensive maintenance

Our approach: ORE

- Constructing the MD schema of a DW in an iterative fashion
- Starting from single business requirements
  - Obtaining MD information for single requirement
- Incrementally build the unified MD schema
  - Satisfying the entire set of requirements

GEM: ORE as a part of a bigger picture



legend:

alternative scenarios

input/ output

suggestions

process flow

Data Sources

Ontology, mappings & Data source features

Designer

Business Requirements

Annotated Ontology

Requirements Validation

Annotated Ontology Subset (AOS)

Requirements Completion

AOS and Context Edges

Multidimensional Tagging

Conceptual ETL Operations

Operator Identification

Operator Library

Consolidation

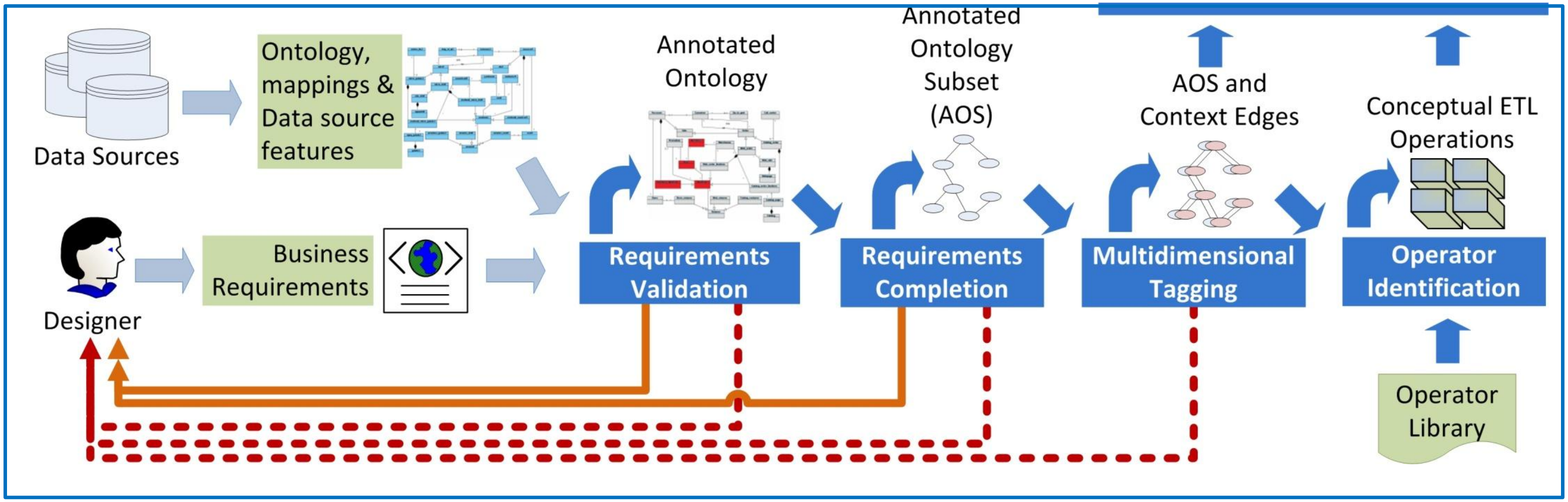
Data Warehouse Conceptual Schema

ETL Process

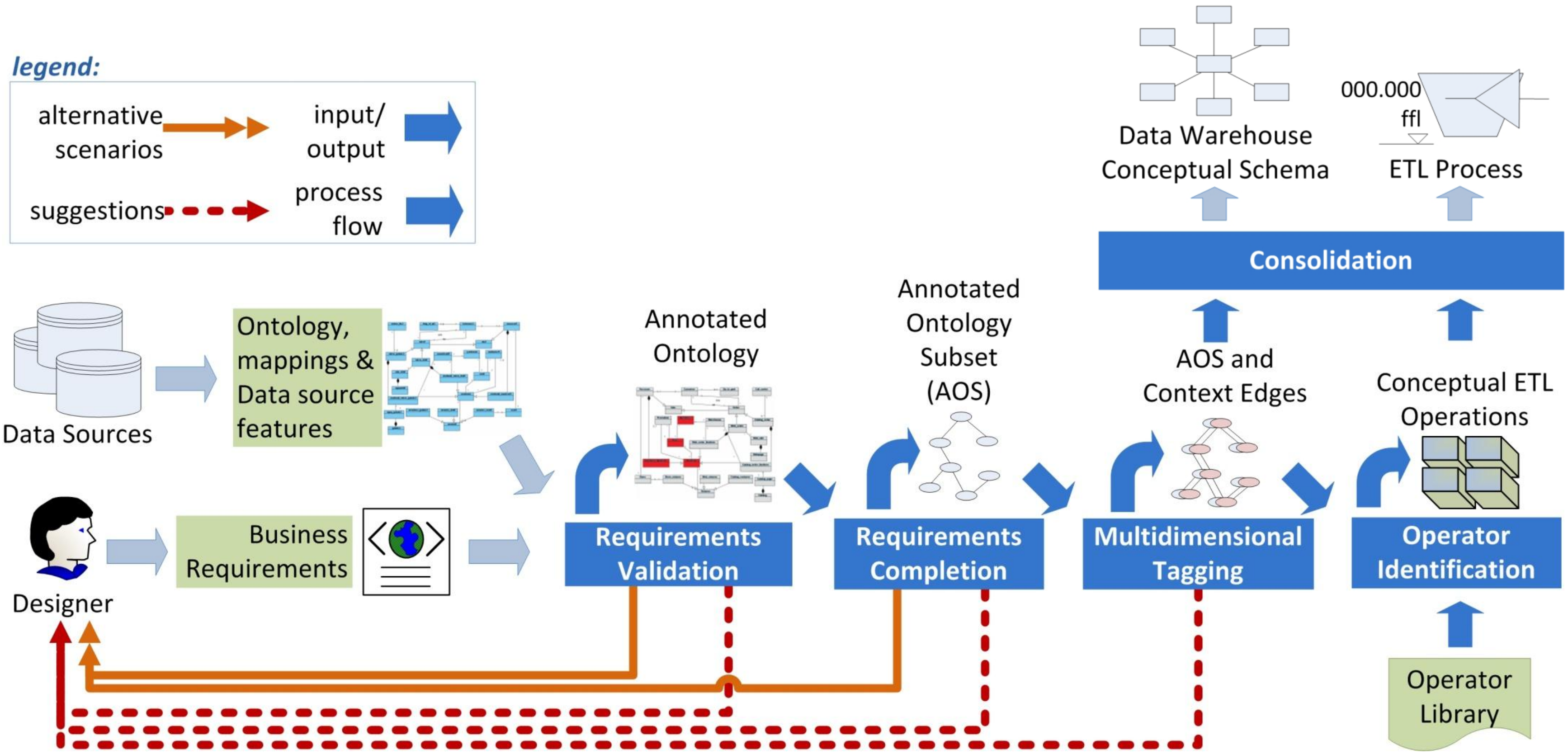
GEM: ORE as a part of a bigger picture

Semi-automatically producing multidimensional (MD) and Extract-transform-load (ETL) conceptual designs from a given set of business requirements (like SLAs) and data source descriptions

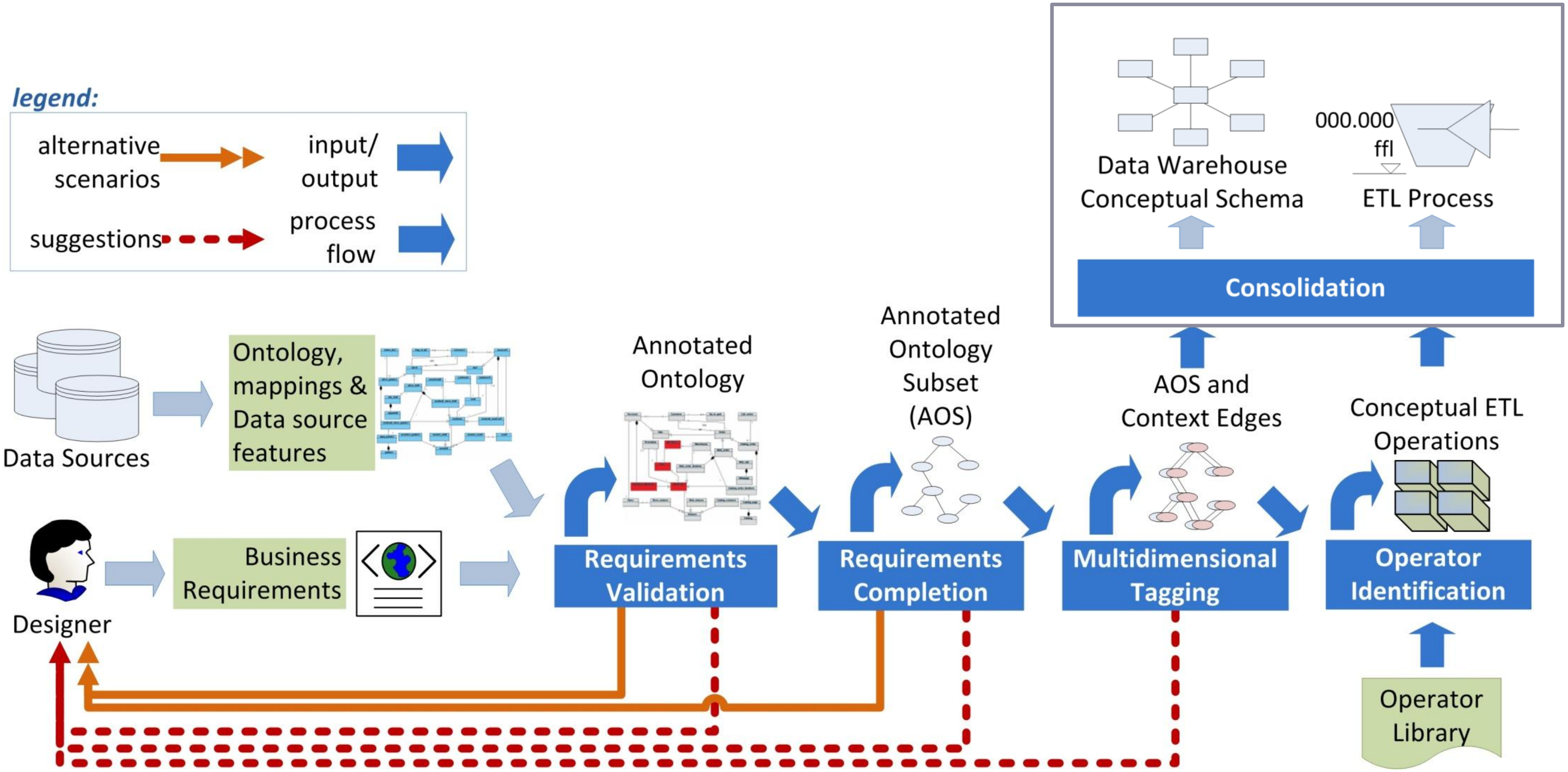
Oscar Romero, Alkis Simitsis, Alberto Abelló:
GEM: Requirement-Driven Generation of ETL and Multidimensional Conceptual Designs.
DaWaK 2011: 80-95



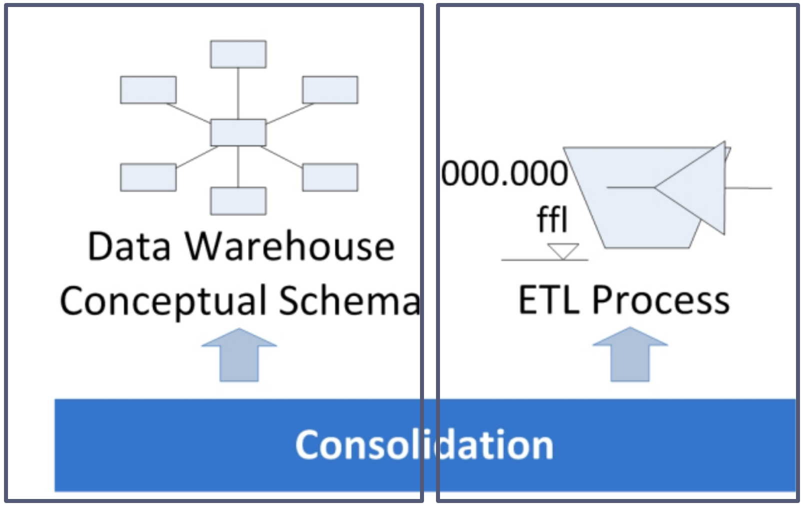
GEM: ORE as a part of a big picture



GEM: ORE as a part of a big picture



GEM: ORE as a part of a big picture

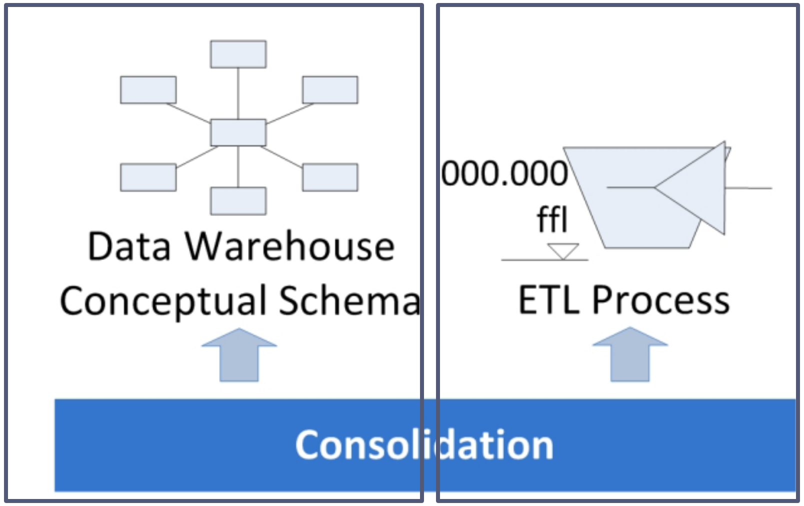


Data Warehouse Conceptual Schema

ETL Process

Consolidation

GEM: ORE as a part of a big picture



Data Warehouse Conceptual Schema

000.000 ffl

ETL Process

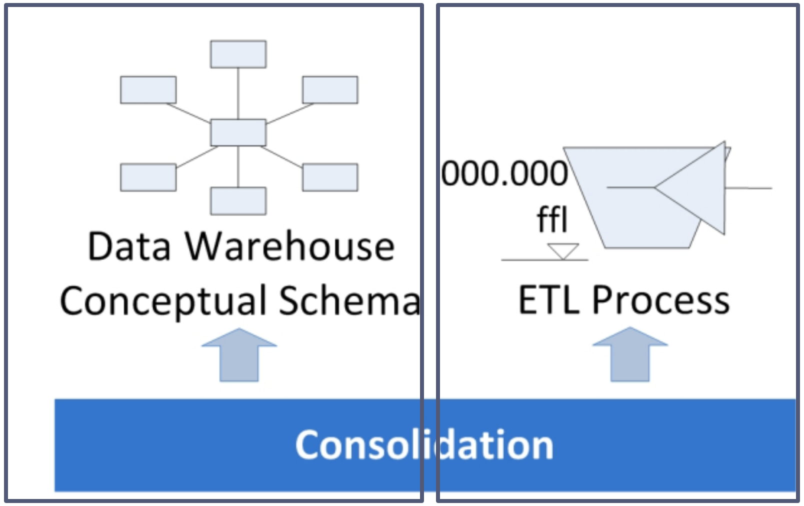
Consolidation

ORE

CoAl

Petar Jovanovic, Oscar Romero, Alkis Simitsis, Alberto Abelló
Integrating ETL Processes from Information Requirements.
DaWaK 2012

GEM: ORE as a part of a big picture



Data Warehouse Conceptual Schema

ETL Process

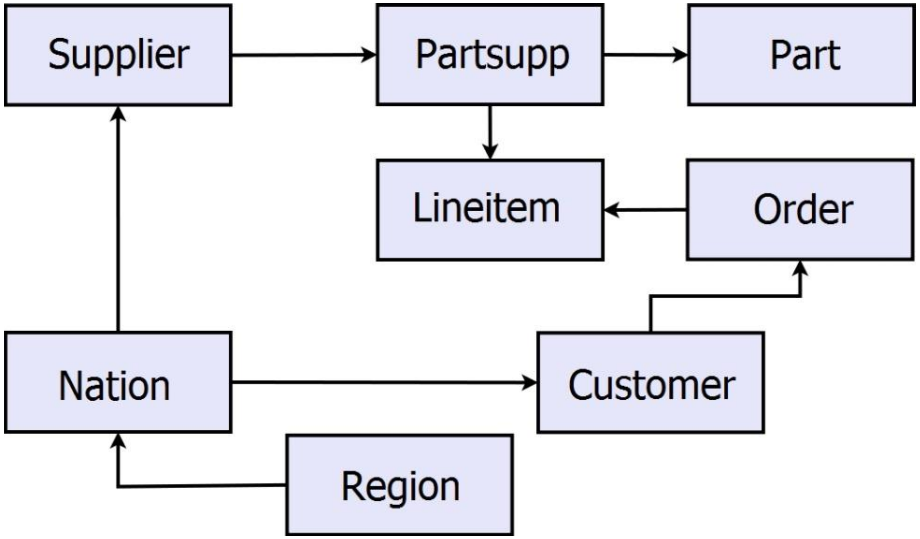
Consolidation

ORE

CoAl

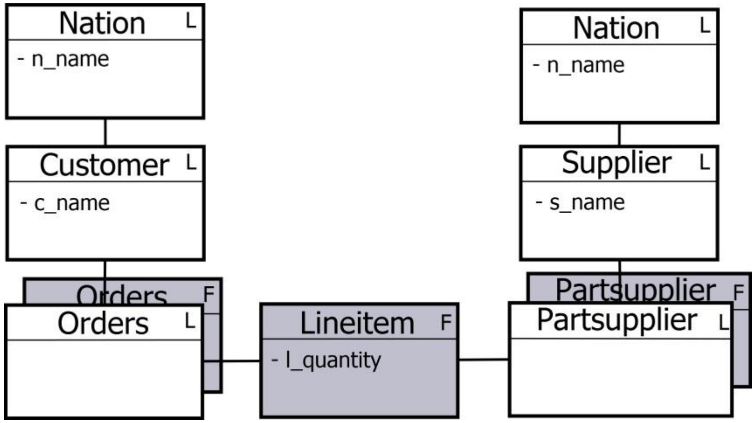
Petar Jovanovic, Oscar Romero, Alkis Simitsis, Alberto Abelló
Integrating ETL Processes from Information Requirements.
DaWaK 2012

Running example – TPC-H

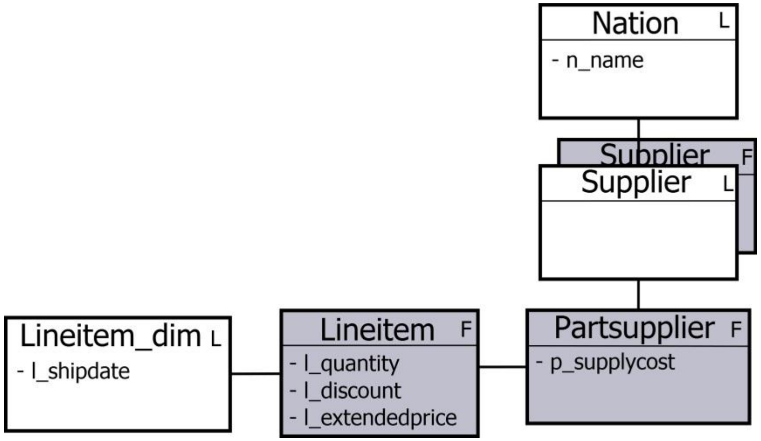


- **Example information requirements:**
  - **IR1**: The total quantity of the parts shipped from Spanish suppliers to French customers
  - **IR2**: For each nation, the profit for all supplied parts, shipped after 01/01/2011
  - **IR3**: The total revenue of the parts supplied from East Europe
  - **IR4**: For German suppliers, the total available stock value of supplied parts
  - **IR5**: Shipping priority and total potential revenue of the parts ordered before certain date and shipped after certain date to a customer of a given segment

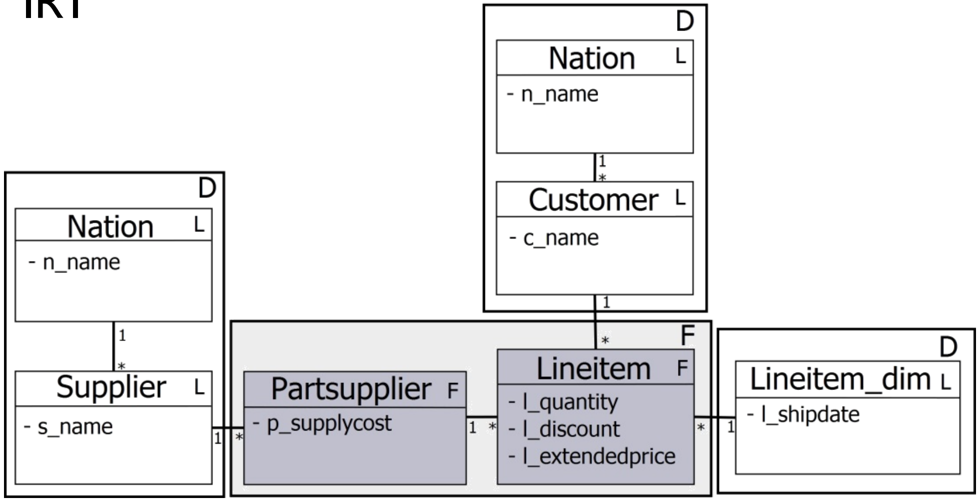
Running example – TPC-H



IR1

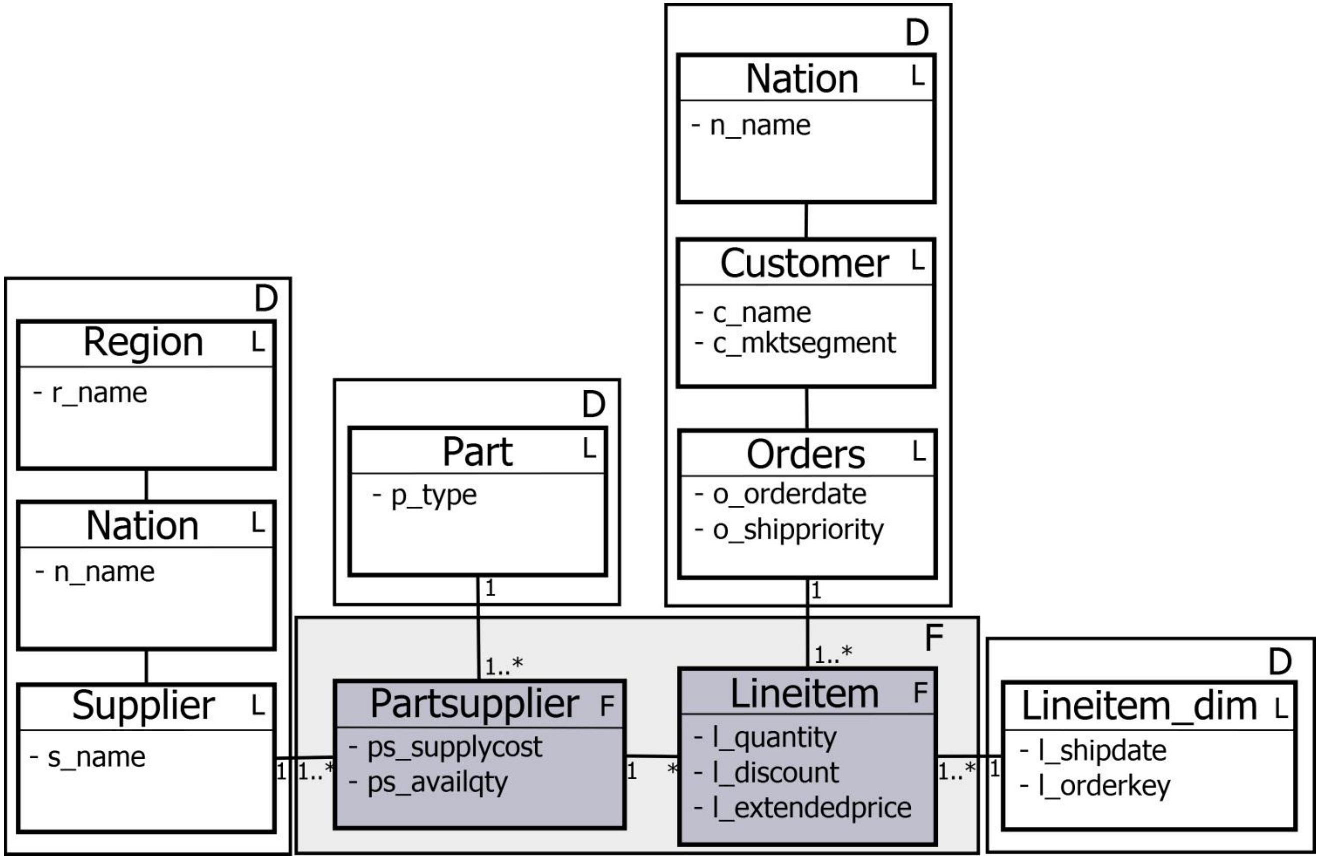


IR2



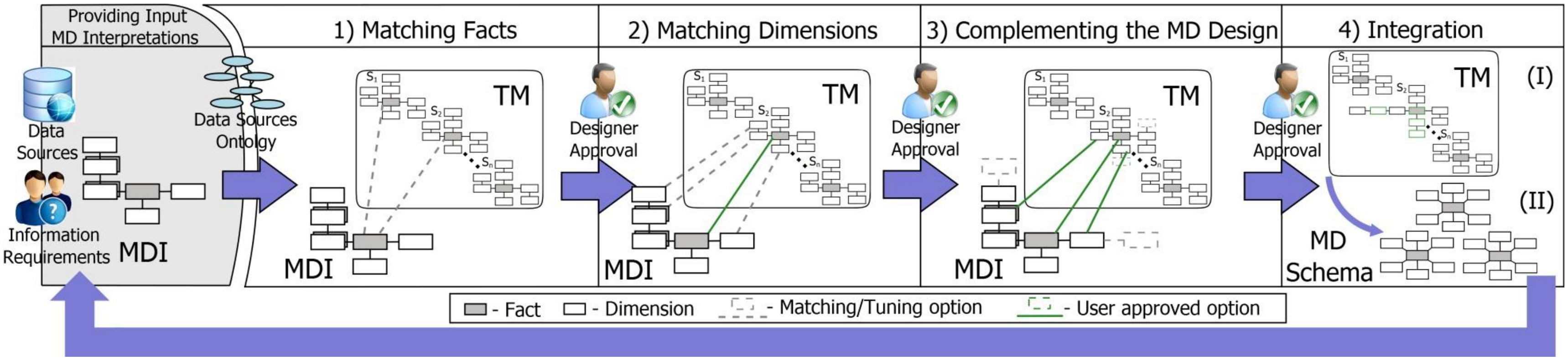
MD Schema satisfying IR1 + IR2

Running example – TPC-H



MD Schema satisfying IR1- IR5

ORE: system overview



- Inputs
  - MD interpretations of requirements (e.g., GEM)
  - Domain ontology capturing data sources' semantics and relations
- Stages
  1. Matching Facts
  2. Matching Dimensions
  3. Complementing the MD Design
  4. Integration

ORE: system overview

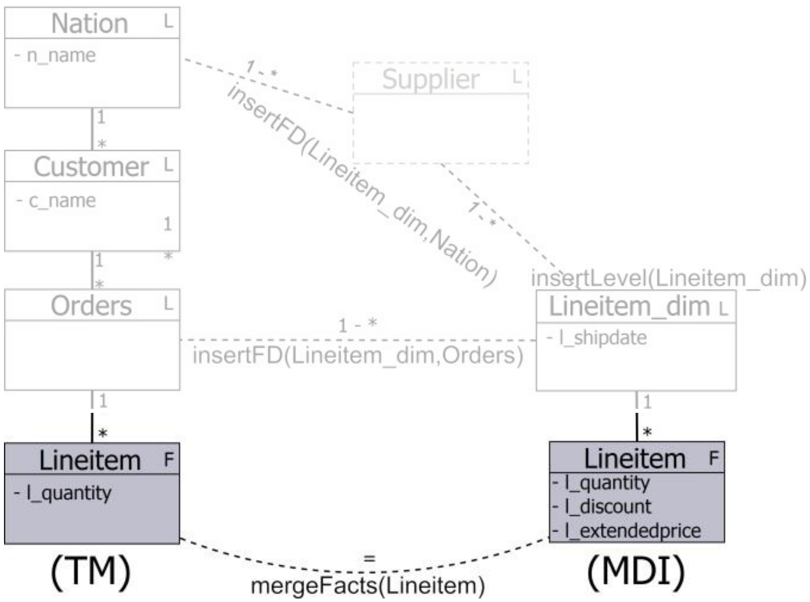
- Traceability metadata (TM)
  - Used for handling evolving requirements
  - Systematically trace everything about the MD design integrated so far (e.g., candidate improvements, alternatives)
  - Avoid overloading the produced MD schema with unnecessary details

$$IR = \{ MDI_i \mid i=1,\dots,n_1 \} \cup \gamma$$
$$S = \{ IR_i \mid i=1,\dots,n_2 \}$$
$$TM = \{ S_i \mid i=1,\dots,n_2 \}$$

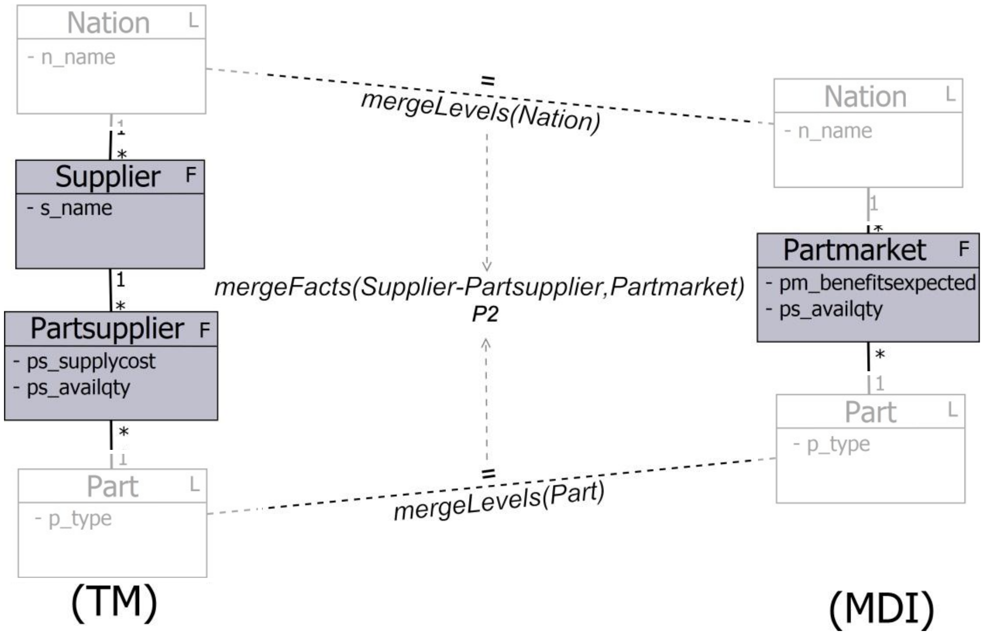
- When a requirement changes:
  - We update TM ($TM_{new} = TM_{old} - IR_{old} + IR_{new}$) and
  - Generate a new MD schema, taking into account previously registered user feedback (γ)

Matching Facts

- Two facts match if they produce an equivalent set of points in the MD space
- Alternative solutions with different costs – user choice



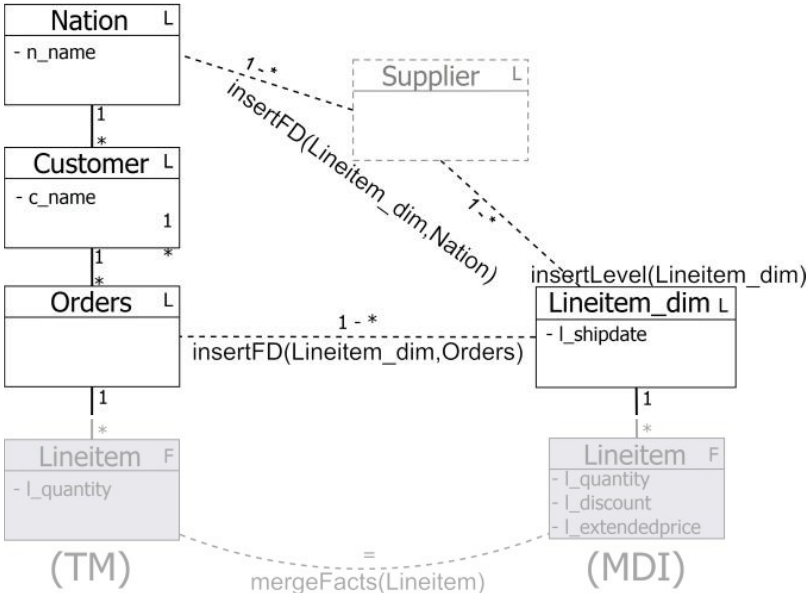
equivalent/synonym facts



facts with equivalent MD space

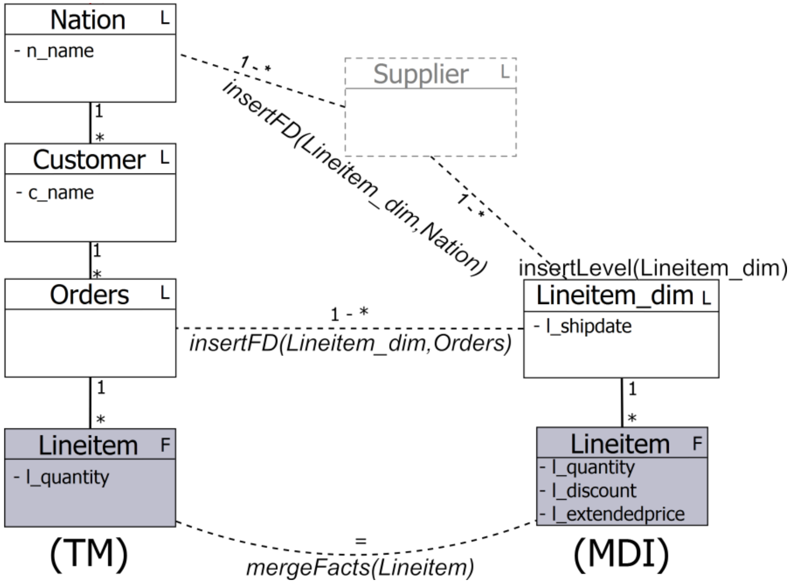
Matching dimensions

- Dimension - partially ordered set of individual levels (DAG)
- We search for possible matchings among the individual levels
  - graph matching problem
- Match levels with minimum path of a valid MD relation (=, 1-1, 1-* or *-1) between them
- Alternative solutions with different costs – user choice



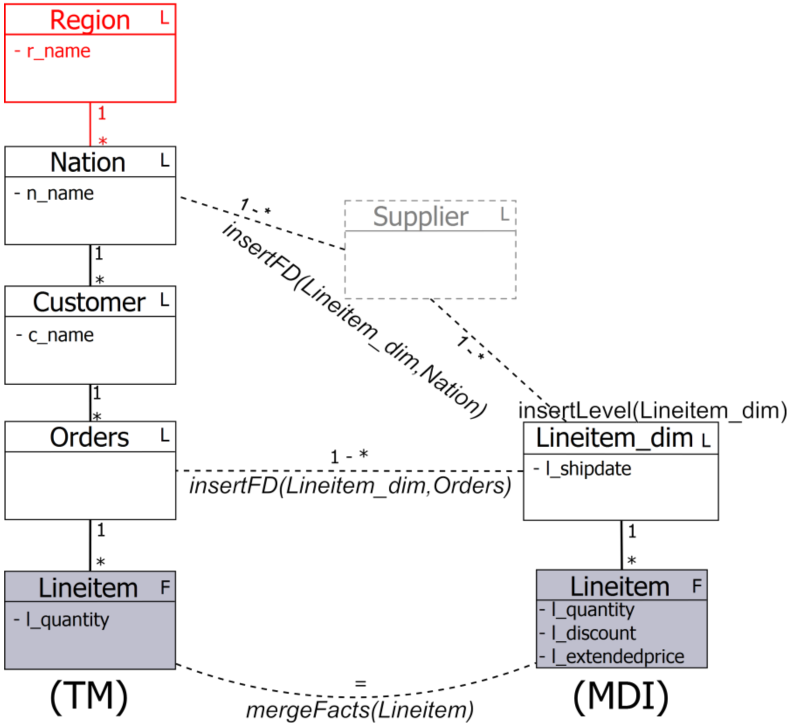
Complementing the MD design

- Starting from the integration of the new requirement in TM, identified in the previous stages
- Explores the ontology to complement the MD design with new analytically interesting concepts



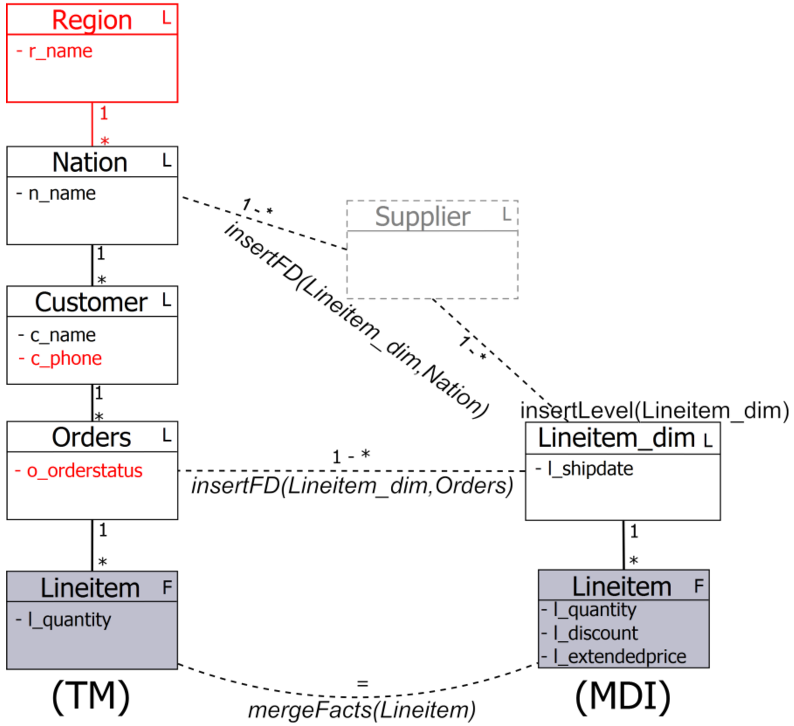
Complementing the MD design

- Starting from the integration of the new requirement in TM, identified in the previous stages
- Explores the ontology to complement the MD design with new analytically interesting concepts
  - new *levels* (functional dependencies "to-one" relationships)



Complementing the MD design

- Starting from the integration of the new requirement in TM, identified in the previous stages
- Explores the ontology to complement the MD design with new analytically interesting concepts
  - new *levels* (functional dependencies "to-one" relationships)
  - *measures, descriptive attributes* (datatype properties)



Integration

- Producing the final MD schema
  - Relaxing the final schema
    from currently irrelevant information
- Two phases

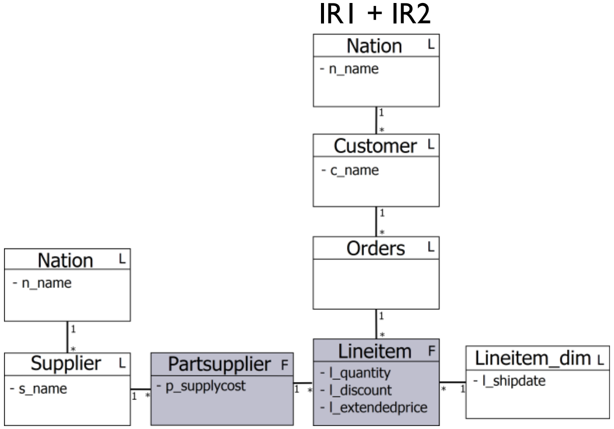
Integration

IR1 + IR2

- Producing the final MD schema
  - Relaxing the final schema from currently irrelevant information
- Two phases

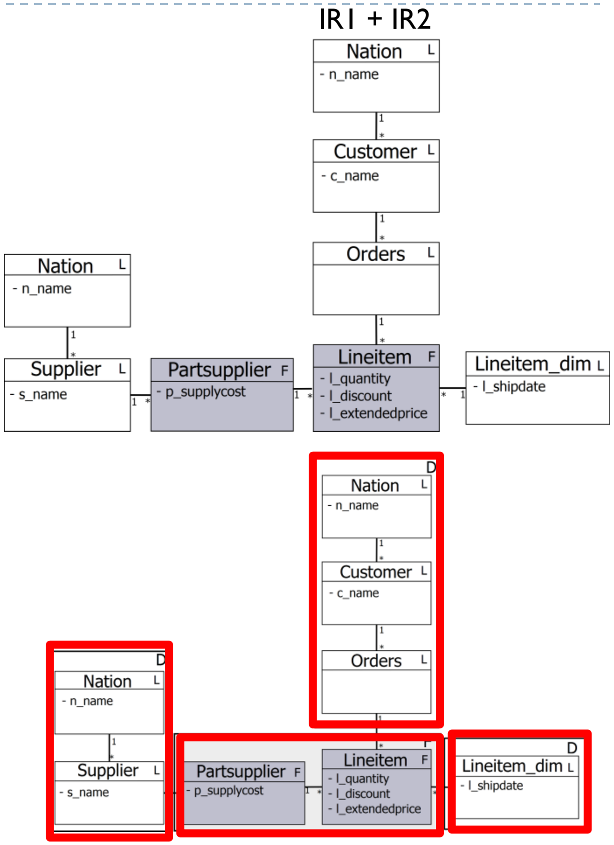
Integration

- Producing the final MD schema
  - Relaxing the final schema from currently irrelevant information
- Two phases



Integration

- Producing the final MD schema
  - Relaxing the final schema from currently irrelevant information
- Two phases
  1. Partitioning grouping different concepts that:
     - Produce a connected subgraph
     - Have the same MD interpretation

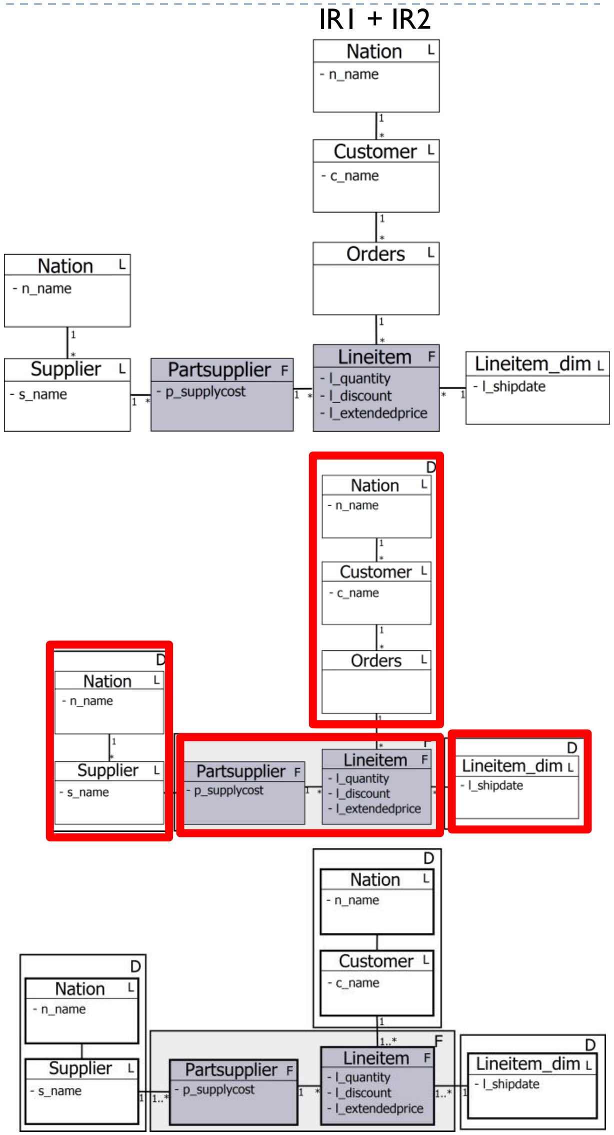


Integration

- Producing the final MD schema
  - Relaxing the final schema
    from currently irrelevant information
- Two phases

  i. Partitioning
     grouping different concepts that:
     - Produce a connected subgraph
     - Have the same MD interpretation

  ii. Folding
     - Consider only the concepts currently required by the user
     - All the knowledge still preserved in TM for future integration steps



Conclusions

- An end-to-end, requirement-driven solution for designing MD schemata and ETL flows for the DW ecosystem
- ORE
  - Incremental approach for integration and evolution of MD schemas
  - Looking for maximal and optimal matching areas (facts, dimensions)
  - Alternative options with different costs (user choice)
  - Storing all the information as traceability metadata
  - Generating the MD schema that satisfies current set of business requirements

Thank You!