FedDW Global Schema Architect

UML-based Design Tool for the Integration of Data Mart Schemas

Dr. Stefan Berger

Department of Business Informatics – Data & Knowledge Engineering
Johannes Kepler University Linz

ACM 15th DOLAP '12 — November 2, 2012

Outline

Problem definition; our contribution

Problem: similar autonomous data marts/DWs, but heterogeneous schemata and/or data

- Business collaboration
- Mergers and acquisitions

⇒ Preexisting DW data across autonomous organizations

Contribution: comprehensive tool suite for integration of autonomous data marts/DWs

- Visual integration of multidimensional schemas
- OLAP front-end prototype, based on SQL-MDi [Berger and Schrefl, 2006]

Problem definition; our contribution

Problem: similar autonomous data marts/DWs, but heterogeneous schemata and/or data

- Business collaboration
- Mergers and acquisitions

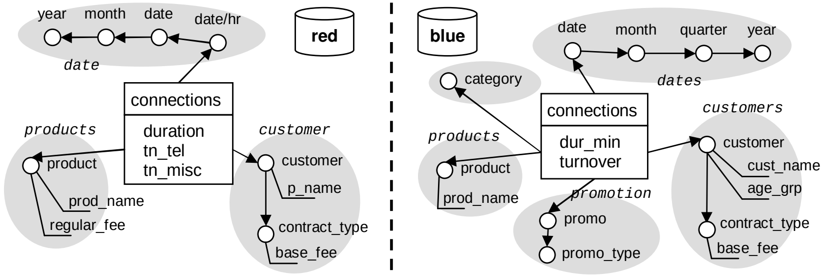
⇒ Preexisting DW data across autonomous organizations

Contribution: comprehensive tool suite for integration of autonomous data marts/DWs

- Visual integration of multidimensional schemas
- OLAP front-end prototype, based on SQL-MDi [Berger and Schrefl, 2006]

Motivating example

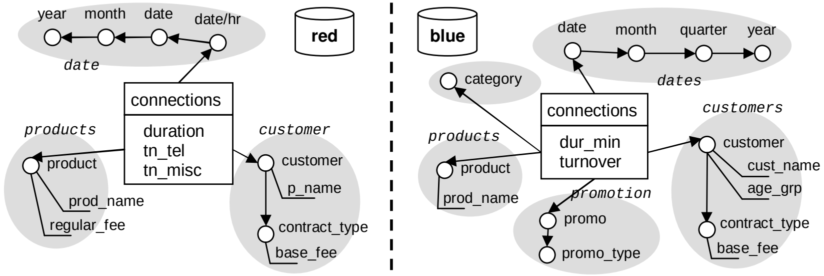
Telecommunications sector—sample, heterogeneous conceptual data mart schemas:



- Dimensionality (extra dimension `blue.promotion`)
- Hierarchy of `date` dimensions
- Decorations of `product` dimensions
- Measures of `connections` facts

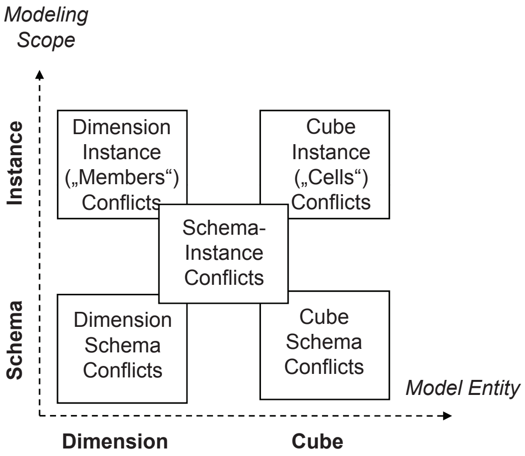
Motivating example

Telecommunications sector—sample, heterogeneous conceptual data mart schemas:



- Dimensionality (extra dimension `blue.promotion`)
- Hierarchy of `date` dimensions
- Decorations of `product` dimensions
- Measures of `connections` facts

Conflict classification I


Modeling Scope
Instance
Schema
Dimension Instance („Members“) Conflicts
Cube Instance („Cells“) Conflicts
Schema-Instance Conflicts
Dimension Schema Conflicts
Cube Schema Conflicts
Model Entity
Dimension
Cube

Conflict classification II

Facts: conflicts	**Relevant operator of FedDW**
Schema-instance	Merge measures: PIVOT MEASURES (Fact) Split measures: PIVOT SPLIT MEASURES (Fact)
Dimensionality	Choose attributes: add DIM reference (Cube)
Different measures	Choose measures: add MEASURE reference (Cube)
Domain (measures)	Convert domain: CONVERT MEASURES APPLY ... (Measure)
Naming of attributes	Rename attributes: operator "–> ..." (Measure, Dimension)
Base levels	Roll-up dimension attributes: ROLLUP TO LEVEL ... (Dimension)
Cube cells (fact extensions)	Join cubes: MERGE CUBES (*n-ary*) Derive measure values: AGGREGATE MEASURE (*n-ary*)

Conflict classification III

Dimensions: conflicts	Relevant operator of FedDW
Hierarchies	Map corresponding levels: add level reference `[...]` (Dimension)
Domain (levels / decorations)	Convert domain: CONVERT ATTRIBUTES APPLY ... (Dimension)
Naming (levels)	Rename attributes: operator "–> ..." (Level)
Naming (decorations)	Map decorations: MATCH ATTRIBUTES (under Merge Dimensions—*n-ary*)
Members (dim. extensions)	Merge sets of members: MERGE DIMENSIONS (*n-ary*)
Roll-up functions	Overwrite hierarchies: RELATE Expression (under Merge Dimensions clause—*n-ary*)
Decoration values	Correct values: add RENAME function (under Merge Dimensions clause—*n-ary*)

Integration workflow

Establish a *federation* of autonomous data marts:

1. Import data mart schemas (CWM supported)
   - (Optional: enrich roll-up hierarchies $\Rightarrow$ *minimum* ***match*** integration strategy)
2. Design global multidimensional schema (canonical model)
3. Define semantic mappings – *both-as-view* paradigm [see McBrien and Poulovassilis, 2003]
   (a) Resolve schema–instance conflicts
   (b) Intensional integration – map conceptual schemata
      - Fact tables
      - Dimension tables + hierarchies

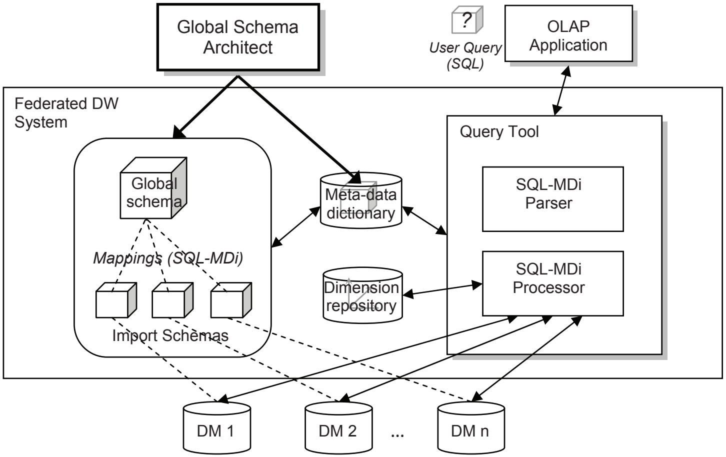
   (c) Extensional integration – consolidate data

Overview of FedDW tool support I

Java- and Eclipse-based interactive tool suite (EMF, GMF, UML2)

- Visual data mart integration:
  FedDW Global Schema Architect (GSA)
- OLAP front-end prototype:
  FedDW Query Tool [Berger and Schrefl, 2009]
- Auxiliary components:
  Metadata Dictionary, Dimension Repository

Overview of FedDW tool support II



Overview of FedDW GSA

Visual design environment for multidimensional schemas

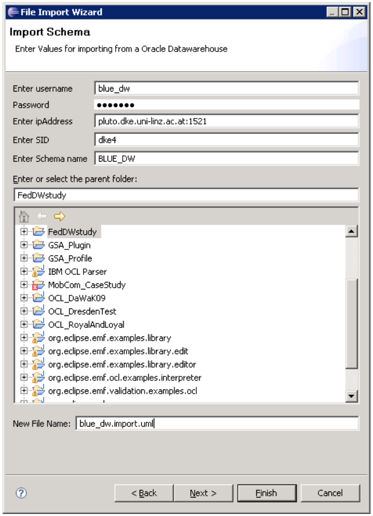
- **Schema Editor** — nested UML diagrams
  - *Import schemas*
  - *Global schema*
- **Mapping Editor** — graphical, high-level code editor (*Master–Detail* layout)
  - *Import mappings*: unary operators (Fact, Dimension entities) — intensional
  - *Global mappings*: n-ary operators — extensional

Sample GSA Workflow

1. Import local, autonomous `connections` schemas
2. Design global `connections` schema
3. Create import mappings
4. Create one global mapping file
5. Export the mappings to metadata repository
6. Export fact and dimension metadata

FedDW Global Schema Architect

GSA: Step 1, Import Wizard I



File Import Wizard

Import Schema

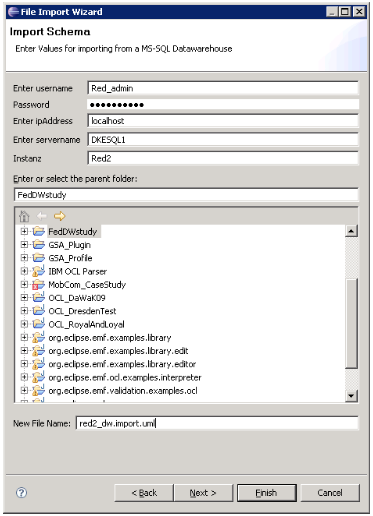
Enter Values for importing from a Oracle Datawarehouse

Enter username	blue_dw
Password	●●●●●●●
Enter ipAddress	pluto.dke.uni-linz.ac.at:1521
Enter SID	dke4
Enter Schema name	BLUE_DW

Enter or select the parent folder: FedDWstudy

- FedDWstudy
- GSA_Plugin
- GSA_Profile
- IBM OCL Parser
- MobCom_CaseStudy
- OCL_DaWaK09
- OCL_DresdenTest
- OCL_RoyalAndLoyal
- org.eclipse.emf.examples.library
- org.eclipse.emf.examples.library.edit
- org.eclipse.emf.examples.library.editor
- org.eclipse.emf.ocl.examples.interpreter
- org.eclipse.emf.validation.examples.ocl

New File Name: blue_dw.import.uml



File Import Wizard

Import Schema

Enter Values for importing from a MS-SQL Datawarehouse

Enter username	Red_admin
Password	●●●●●●●●●●
Enter ipAddress	localhost
Enter servername	DKESQL1
Instanz	Red2

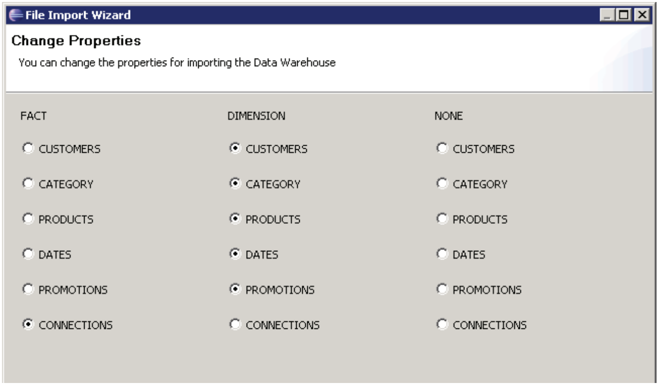
Enter or select the parent folder: FedDWstudy

- FedDWstudy
- GSA_Plugin
- GSA_Profile
- IBM OCL Parser
- MobCom_CaseStudy
- OCL_DaWaK09
- OCL_DresdenTest
- OCL_RoyalAndLoyal
- org.eclipse.emf.examples.library
- org.eclipse.emf.examples.library.edit
- org.eclipse.emf.examples.library.editor
- org.eclipse.emf.ocl.examples.interpreter
- org.eclipse.emf.validation.examples.ocl

New File Name: red2_dw.import.uml

GSA: Step 1, Import Wizard II

Wizard suggests appropriate UML stereotypes (based on PK/FK constraints):


File Import Wizard

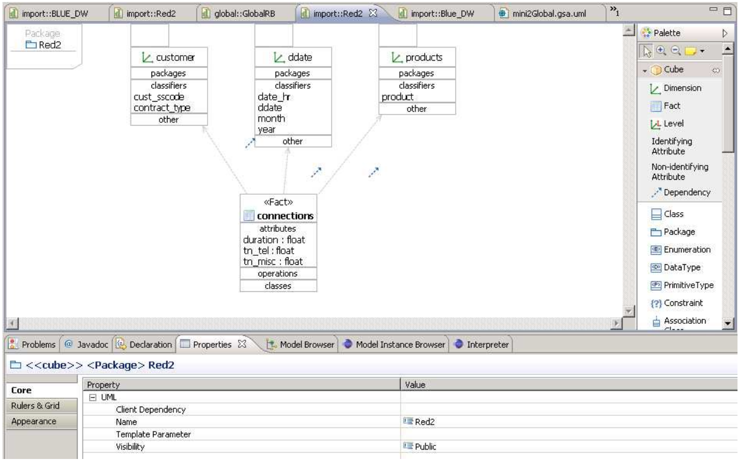
Change Properties

You can change the properties for importing the Data Warehouse

FACT	DIMENSION	NONE
CUSTOMERS	[✓ CUSTOMERS]	CUSTOMERS
CATEGORY	[✓ CATEGORY]	CATEGORY
PRODUCTS	[✓ PRODUCTS]	PRODUCTS
DATES	[✓ DATES]	DATES
PROMOTIONS	[✓ PROMOTIONS]	PROMOTIONS
[✓ CONNECTIONS]	CONNECTIONS	CONNECTIONS

GSA: Step 1, Import Wizard III

Initialized class diagram of `red.connections`:

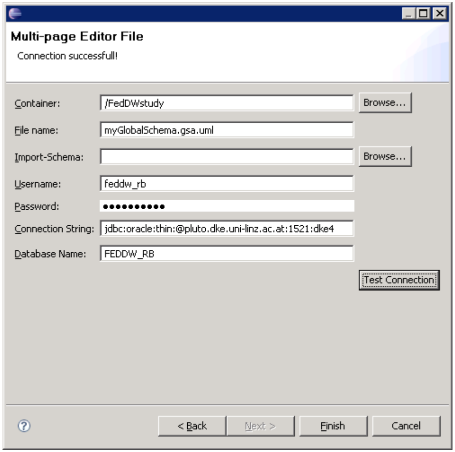


Property	Value
UML	
Client Dependency	
Name	Red2
Template Parameter	
Visibility	Public

GSA: Step 2, Global Schema Editor

Global Schema wizard:

- Comfortably create global schema as copy of one import schema
- Edit the schema later on

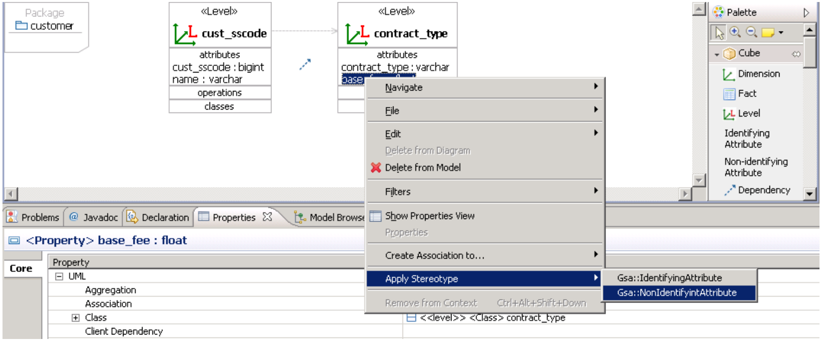

Multi-page Editor File

Connection successfull!

Container:	/FedDWstudy
File name:	myGlobalSchema.gsa.uml
Import-Schema:	
Username:	feddw_rb
Password:	●●●●●●●●●●●
Connection String:	jdbc:oracle:thin:@pluto.dke.uni-linz.ac.at:1521:dke4
Database Name:	FEDDW_RB

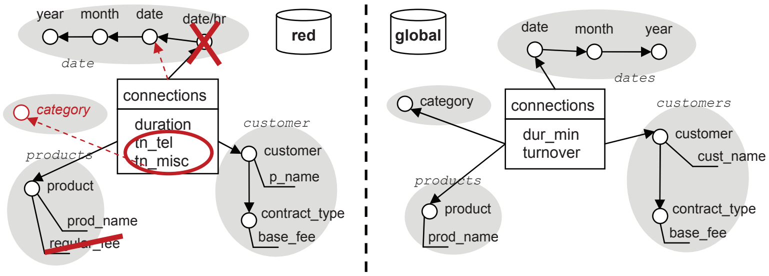
GSA Schema Editors: edit dimension

User-friendly editing of UML diagram possible (context menus, UML palette):



GSA: Step 3, Import Mappings I

Recall the heterogeneities among `red` and `global`:

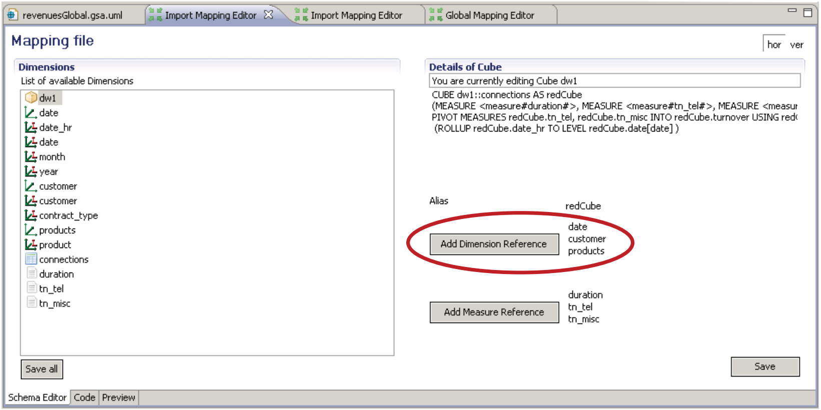


GSA: Step 3, Import Mappings II

Repair `red.connections` import schema:

- Dimensionality: *add references* to all three dimensions
- Date hierarchy: *roll-up to LEVEL [date]*
- Product decorations:
  delete regular_fee from Red's import schema
- Measures (schema–instance conflict):
  PIVOT MEASURES tn_tel, tn_misc

Import Mappings: dimensionality

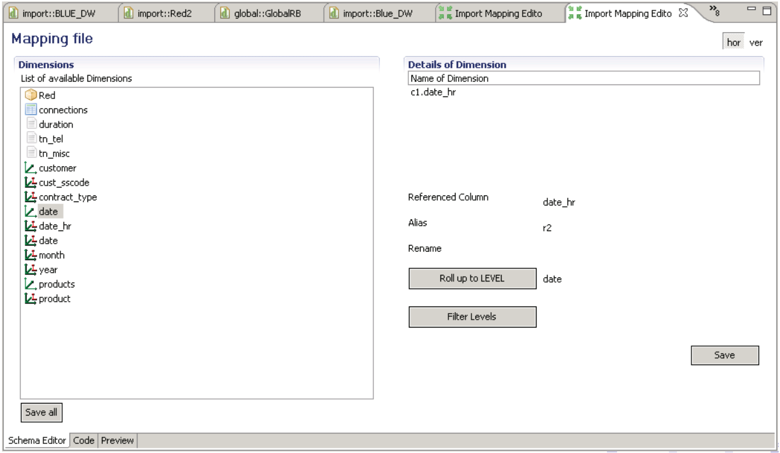


You are currently editing Cube dw1

```
CUBE dw1::connections AS redCube
(MEASURE <measure#duration#>, MEASURE <measure#tn_tel#>, MEASURE <measur
PIVOT MEASURES redCube.tn_tel, redCube.tn_misc INTO redCube.turnover USING red
 (ROLLUP redCube.date_hr TO LEVEL redCube.date[date] )
```

Import Mappings: hierarchy

Prerequisite: delete level [`date/hr`] from `red.date` (see Schema Editor, slide 21)



Mapping file

Dimensions
List of available Dimensions

- Red
- connections
- duration
- tn_tel
- tn_misc
- customer
- cust_sscode
- contract_type
- date
- date_hr
- date
- month
- year
- products
- product

Details of Dimension

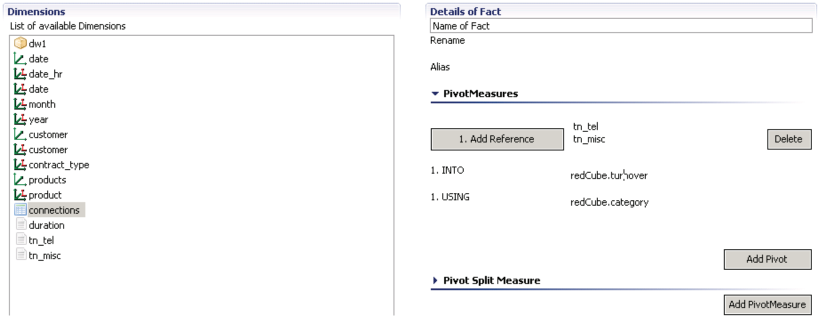
Name of Dimension
c1.date_hr

Referenced Column	date_hr
Alias	r2
Rename	
Roll up to LEVEL	date
Filter Levels	

Save

Import Mappings: pivot measures

"Merge" measures `tn_tel`, `tn_misc` into turnover, extracting values "tn_tel", "tn_misc" as members of the new `red.category` dimension:


Dimensions
List of available Dimensions

- dw1
- date
- date_hr
- date
- month
- year
- customer
- customer
- contract_type
- products
- product
- connections
- duration
- tn_tel
- tn_misc

Details of Fact
Name of Fact
Rename
Alias

PivotMeasures

1. Add Reference	tn_tel tn_misc	Delete
1. INTO	redCube.turnover	
1. USING	redCube.category	

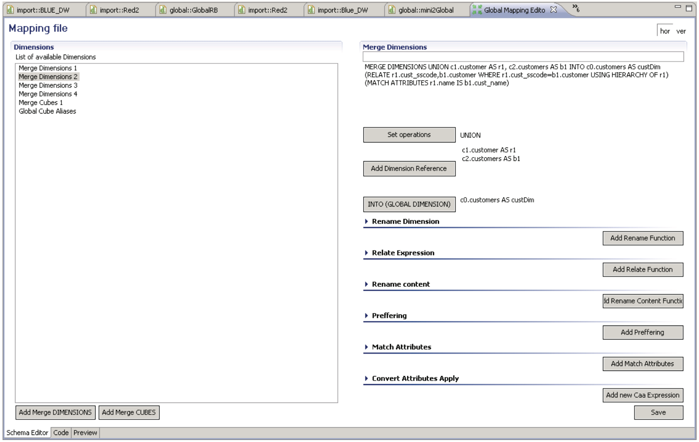
Add Pivot

Pivot Split Measure

Add PivotMeasure

GSA Step 4, Global Mapping I

Merge Dimensions:



MERGE DIMENSIONS UNION c1.customer AS r1, c2.customers AS b1 INTO c0.customers AS custDim
(RELATE r1.cust_sscode,b1.customer WHERE r1.cust_sscode=b1.customer USING HIERARCHY OF r1)
(MATCH ATTRIBUTES r1.name IS b1.cust_name)

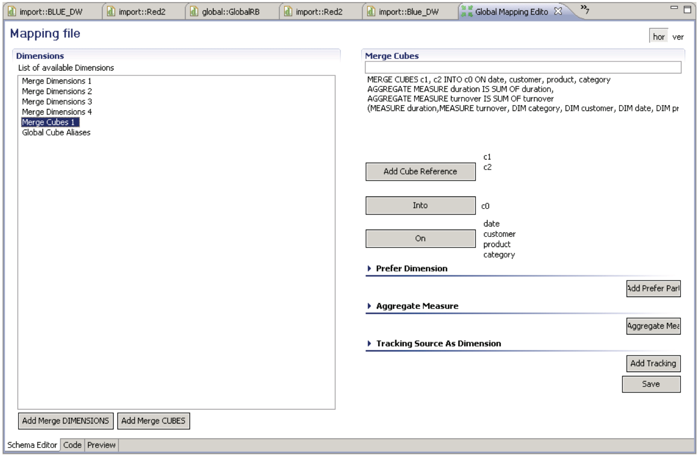
Set operations: UNION

c1.customer AS r1
c2.customers AS b1

INTO (GLOBAL DIMENSION): c0.customers AS custDim

GSA Step 4, Global Mapping II

Merge Cubes:



```
MERGE CUBES c1, c2 INTO c0 ON date, customer, product, category
AGGREGATE MEASURE duration IS SUM OF duration,
AGGREGATE MEASURE turnover IS SUM OF turnover
(MEASURE duration,MEASURE turnover, DIM category, DIM customer, DIM date, DIM pr
```

GSA Step 5–6, export project metadata

Step 5: Export mapping file — export wizard

- Starts generation of SQL-MDi code
- Static syntax check
- Interface to FedDW Query Tool: file system

Step 6: populate Metadata Dictionary

- Facts + dimensions conceptual and physical metadata
- Later accessed by FedDW Query Tool

GSA Step 5–6, export project metadata

Step 5: Export mapping file — export wizard

- Starts generation of SQL-MDi code
- Static syntax check
- Interface to FedDW Query Tool: file system

Step 6: populate Metadata Dictionary

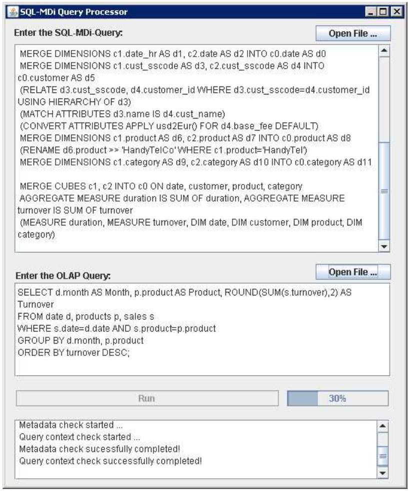
- Facts + dimensions conceptual and physical metadata
- Later accessed by FedDW Query Tool

GSA: Summary

Intelligent features:

- Import heuristics: analyzes PK/FK constraints in import schemas to suggest adequate UML stereotypes
- Create global schema as copy of one import schema
- User-friendly and intuitive UML notation
- Visual conversion modeling avoids “cheap” SQL-MDi syntax errors
- Automatically populates *Dimension Repository* from the exported metadata
- Supports the CWM standard [Poole, 2003]

Query Tool in a Nutshell


SQL-MDi Query Processor

Enter the SQL-MDi-Query:

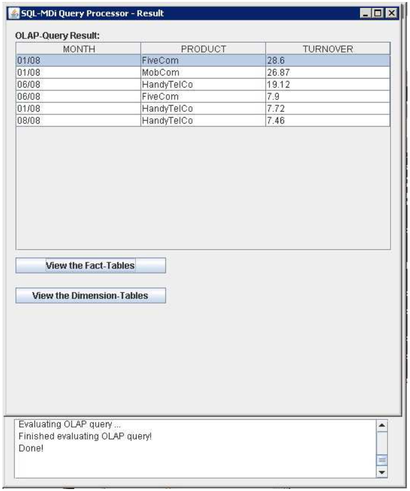
```
MERGE DIMENSIONS c1.date_hr AS d1, c2.date AS d2 INTO c0.date AS d0
MERGE DIMENSIONS c1.cust_sscode AS d3, c2.cust_sscode AS d4 INTO
c0.customer AS d5
(RELATE d3.cust_sscode, d4.customer_id WHERE d3.cust_sscode=d4.customer_id
USING HIERARCHY OF d3)
(MATCH ATTRIBUTES d3.name IS d4.cust_name)
(CONVERT ATTRIBUTES APPLY usd2Eur() FOR d4.base_fee DEFAULT)
MERGE DIMENSIONS c1.product AS d6, c2.product AS d7 INTO c0.product AS d8
(RENAME d6.product >> 'HandyTelCo' WHERE c1.product='HandyTel')
MERGE DIMENSIONS c1.category AS d9, c2.category AS d10 INTO c0.category AS d11

MERGE CUBES c1, c2 INTO c0 ON date, customer, product, category
AGGREGATE MEASURE duration IS SUM OF duration, AGGREGATE MEASURE
turnover IS SUM OF turnover
(MEASURE duration, MEASURE turnover, DIM date, DIM customer, DIM product, DIM
category)
```

Enter the OLAP Query:

```
SELECT d.month AS Month, p.product AS Product, ROUND(SUM(s.turnover),2) AS
Turnover
FROM date d, products p, sales s
WHERE s.date=d.date AND s.product=p.product
GROUP BY d.month, p.product
ORDER BY turnover DESC;
```

Run 30%

```
Metadata check started ...
Query context check started ...
Metadata check sucessfully completed!
Query context check successfully completed!
```


SQL-MDi Query Processor - Result

OLAP-Query Result:

MONTH	PRODUCT	TURNOVER
01/08	FiveCom	28.6
01/08	MobCom	26.87
06/08	HandyTelCo	19.12
06/08	FiveCom	7.9
01/08	HandyTelCo	7.72
08/08	HandyTelCo	7.46

View the Fact-Tables

View the Dimension-Tables

```
Evaluating OLAP query ...
Finished evaluating OLAP query!
Done!
```

FedDW Global Schema Architect

Thanks for your attention!

References

Berger, S. and Schrefl, M. (2006). Analysing multi-dimensional data across autonomous data warehouses. In Tjoa, A. M. and Tho, N., editors, *DaWaK*, pages 120–133.

Berger, S. and Schrefl, M. (2009). FedDW: A tool for querying federations of data warehouses. In *ICEIS (1)*.

McBrien, P. and Poulovassilis, A. (2003). Data integration by bi-directional schema transformation rules. In Dayal, U., Ramamritham, K., and Vijayaraman, T. M., editors, *ICDE*, pages 227–238. IEEE Computer Society.

Poole, J. M. (2003). *Common Warehouse Metamodel Developer's Guide*. John Wiley & Sons, Inc., New York, NY, USA.