A conceptual approach and an overall framework for the development of data warehouses

Juan Trujillo

Grupo de Investigación LUCENTIA

Dpto. Lenguajes y Sistemas Informáticos
(Language and Information Systems)
Universidad de Alicante

Departamento de Lenguajes y Sistemas Informáticos

Universitat d'Alacant
Universidad de Alicante

Outline

- Introduction
- Overall Method based on the UML
- MDA: Model Driven Architecture
- Data Warehouse Quality
- Other issues: Security and Data Mining
- CASE tools
- Conclusions and further works

Outline

- Introduction
- Overall Method based on the UML
- MDA: Model Driven Architecture
- Data Warehouse Quality
- Other issues: Security and Data Mining
- CASE tools
- Conclusions and further works

Introduction

- Definition by W. Inmon (1992)
  - "A subject-oriented, integrated, time-variant, and non-volatile collection of data used in support of management's decisions"

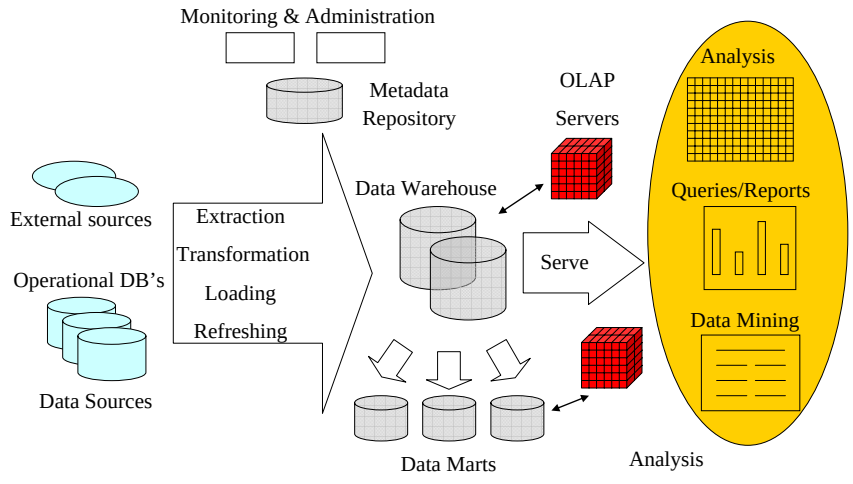
Introduction

- Subject oriented
  - The DW is organised by “data subjects” that are relevant to the organisation.
    - Subjects: Sales, Purchases, shipments, etc.
    - Context of analysis: clients, suppliers, products, etc...
  - Multidimensional Modeling (First approach)
    - Facts → Activities of a high interest
    - Dimensions → context of analysis
  - At the logical level: Star schema and its variants (snowflake, fact constellations)
    - Fact and dimension tables

Introduction

- Integrated
  - Data integrated from different data sources to provide a comprehensive view
- Time variant
  - Historical data: related to a time period and incremented periodically
- Non-volatile
  - Data are not updated or erased by end users. New data is always (*most of times*) added.

Introduction



Introduction

- Data Warehouses (DWs) are complex Information Systems
- They support:
  - OLAP
  - Data mining
  - Decision Support Systems
  - ...
- Building a DW:
  - A complex task and time consuming
  - DWs → prone to failures
  - High costs

Introduction

- Partial approaches:
  - ETL processes
  - Logical and conceptual approaches
  - Deriving DW schemas from the available data sources
  - ...
- Data warehouses, MD databases, OLAP applications
  - Multidimensional (MD) modeling

Introduction

- Different approaches for conceptual modeling (graphical approaches):
  - Golfarelli *et al.*
  - Husemann *et al.*
  - Sapia *et al.*
  - Tryfona *et al.*
  - Abello *et al.*
  - *Akoka et al.*
  - *...*

→ Own graphical notations

↓

Learn a new notation

- Different methods for desiging DWs, but not a global method covering main relevant parts of a DW (e.g. sources, DW repository, ETL, ...)

Outline

- Introduction
- Overall Method based on the UML
- MDA: Model Driven Architecture
- Data Warehouse Quality
- Other issues: Security and Data Mining
- CASE tools
- Conclusions and further works

Outline

- Introduction
- Overall Method based on the UML
- MDA: Model Driven Architecture
- Data Warehouse Quality
- Other issues: Security and Data Mining
- CASE tools
- Conclusions and further works

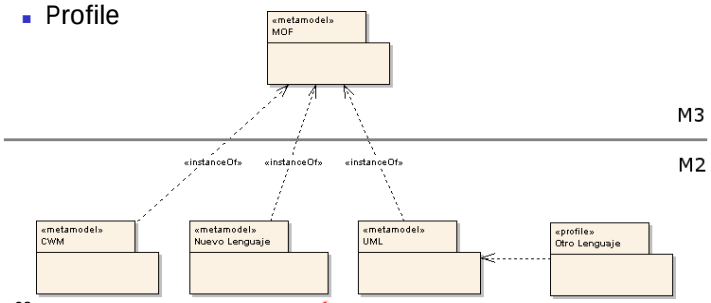
Overall Method based on the UML

- Aim: A complete method for the design of DWs
- Basics of our approach:
  - Standard notation → UML
  - Complete → Including main design phases
  - Flexible method → Allows using 15 different diagrams
    - No necessary to use all of them
  - Different detail levels or different users (IT professionals, final users
    - Packages
  - Method Scientific production
    - JISBD'03, DWDM'03, ADVIS'04 etc...
    - The whole proposal has more than 90 papers
  - Aplicable → Supported by CASE tools

Overall Method based on the UML

Unified Modeling Language (UML)

- UML Extension mechanisms
  - UML is a general purpose visual modeling language
  - Extension mechanisms allow us to adapt it to specific domains
    - Metamodel from MOF
    - Profile



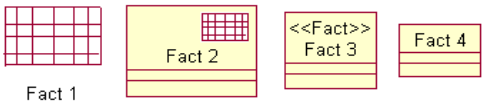
Overall Method based on the UML

Unified Modeling Language (UML)

- …
  - Metamodel from MOF
  - Profile

Profile
- Stereotypes → New modeling constructiors
- Tagged values → New properties
- Constraints → New semantics

Different views of a stereotype

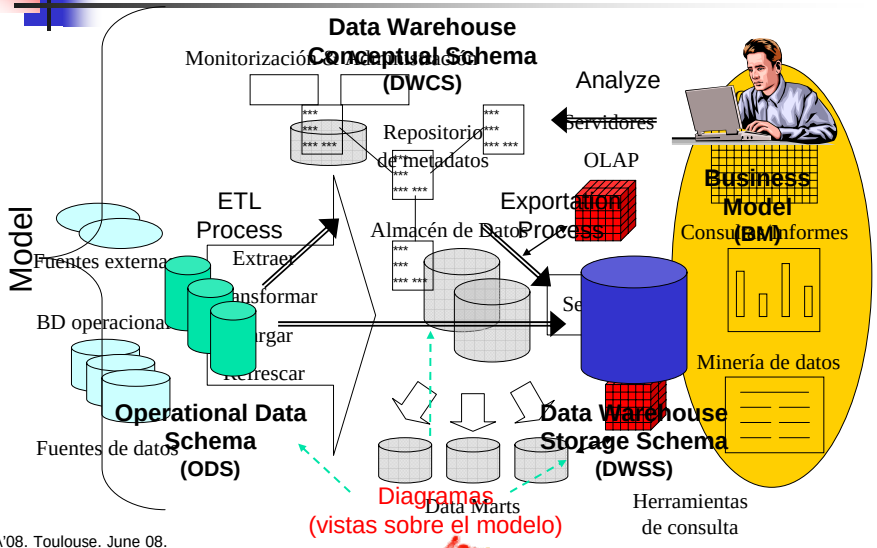


Overall Method based on the UML

- The development of the DW is structured in an integrated *framework*:
  - Five stages
  - Three levels

  } Fifteen diagramas
- The different diagrams of the same DW are not independent but overlapping (UML importing mechanism)

Overall Method based on the UML

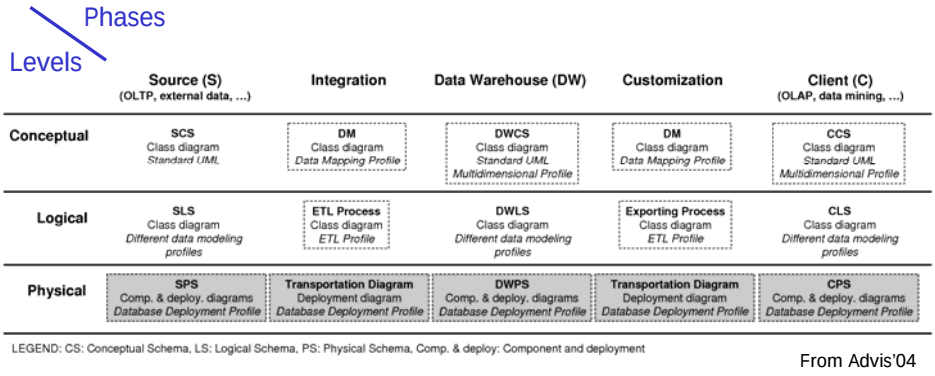


Overall Method based on the UML

- Stages:
  - *Source*: data sources (OLTP, external data sources, etc.)
  - *Integration*: mapping between source and data warehouse
  - *Data Warehouse*: structure of the DW
  - *Customization*: mapping between data warehouse and clients' structures
  - *Client*: structures used by the clients to access the DW (data marts, OLAP applications, etc.)
- For each stage, different levels:
  - Conceptual
  - Logical
  - Physical

Overall Method based on the UML

15 UML diagrams can be used



From Advis'04

Overall Method based on the UML

Based on the Unified Process (UP)

- DW development method
  - Based on
    - *Unified Software Development Process* o *Unified Process* (UP)
- UP is generic → must be instantiated
- Life-cycle project divided into:
  - 4 phases/stages
  - 5+2 core workflows

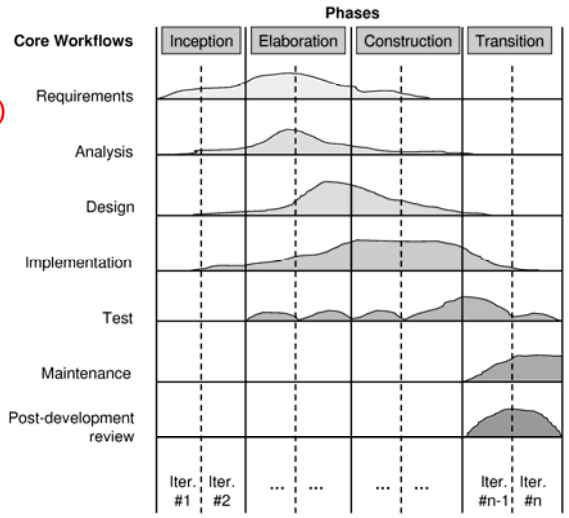
Overall Method based on the UML

Based on the Unified Process (UP)

Our Unified Process (UP) instantiation

Hybrid approach

Top-Down + bottom-up



Overall Method based on the UML

Based on the Unified Process (UP)

- ***Requirements***
  - What the final user expects to do with the DW (measures, dimensions, aggregations, etc.)
- Method:
  - Use cases:
    - UML use cases diagrams
    - Template: name, actor, etc.
  - Adapt i* (TROPOS) to capture *goals*
    - Other work: Rizzi et al. (DOLAP,05)

Overall Method based on the UML

Based on the Unified Process (UP)

- Obtaining requirements
  - Business Use case diagrams
  - Represent main objectives to be achieved by implementing the DW
- Requirement specification
  - Use case diagrams
  - Defining the requried information requirements to satisfy the objectives
- Validating requirements
  - To build a MD schema from use case diagrams
  - To check that this schema will allow fullfilling user requirements.
- In Rebnita'05, RIGIM'07, IDEAS'08 → i* adapted for DWs

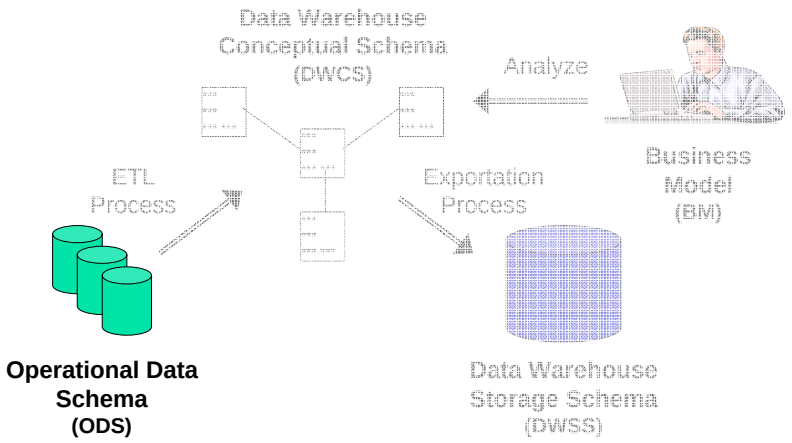
Overall Method based on the UML

Based on the Unified Process (UP)

- ***Analysis***:
  - Refine and structure the requirement
  - Document data sources
- Method:
  - Conceptual schema of data sources
    - Class Diagram
    - Component Diagram
    - Deployment Diagram

Overall Method based on the UML

Based on the Unified Process (UP)



Overall Method based on the UML

Based on the Unified Process (UP)

	Source (S) (OLTP, external data, ...)	Integration	Data Warehouse (DW)	Customization	Client (C) (OLAP, data mining, ...)
Conceptual	**SCS** Class diagram *Standard UML*	**DM** Class diagram *Data Mapping Profile*	**DWCS** Class diagram *Standard UML Multidimensional Profile*	**DM** Class diagram *Data Mapping Profile*	**CCS** Class diagram *Standard UML Multidimensional Profile*
Logical	**SLS** Class diagram *Different data modeling profiles*	**ETL Process** Class diagram *ETL Profile*	**DWLS** Class diagram *Different data modeling profiles*	**Exporting Process** Class diagram *ETL Profile*	**CLS** Class diagram *Different data modeling profiles*
Physical	**SPS** Comp. & deploy. diagrams *Database Deployment Profile*	**Transportation Diagram** Deployment diagram *Database Deployment Profile*	**DWPS** Comp. & deploy. diagrams *Database Deployment Profile*	**Transportation Diagram** Deployment diagram *Database Deployment Profile*	**CPS** Comp. & deploy. diagrams *Database Deployment Profile*

LEGEND: CS: Conceptual Schema, LS: Logical Schema, PS: Physical Schema, Comp. & deploy: Component and deployment

Overall Method based on the UML

Based on the Unified Process (UP)

- *Operational Data Schema*
- Represents:
  - OLTP
  - External data sources
- There is not a UML extension to model different types of data sources

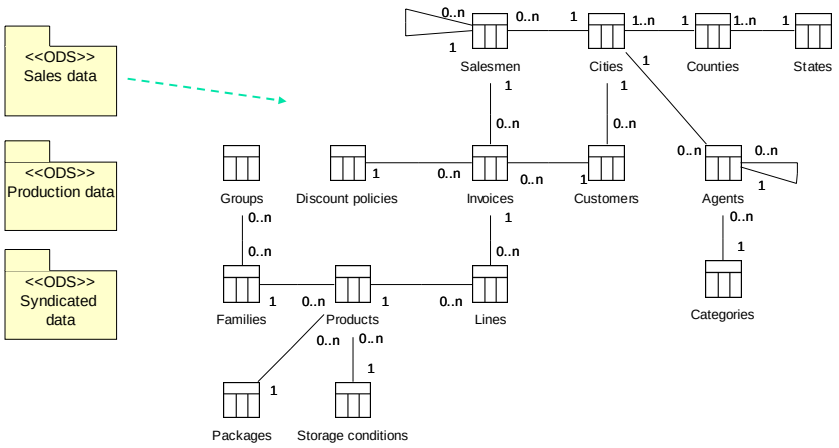
Overall Method based on the UML

Based on the Unified Process (UP)

- RDBMS → *Rational's UML Profile for Database Design*: <<Database>>, <<Schema>>, <<Table>>, ...
- ORDBMS → *Marcos et al. UML Profile for Object-Relational Database Design*: <<array>>, <<row>>, <<ref>>, ...
- XML → *Rational's XML-DTD UML Profile*: <<DTDElement>>, <<DTDElementEmpty>>, <<DTDEntity>>,
- ...

Overall Method based on the UML

Based on the Unified Process (UP)



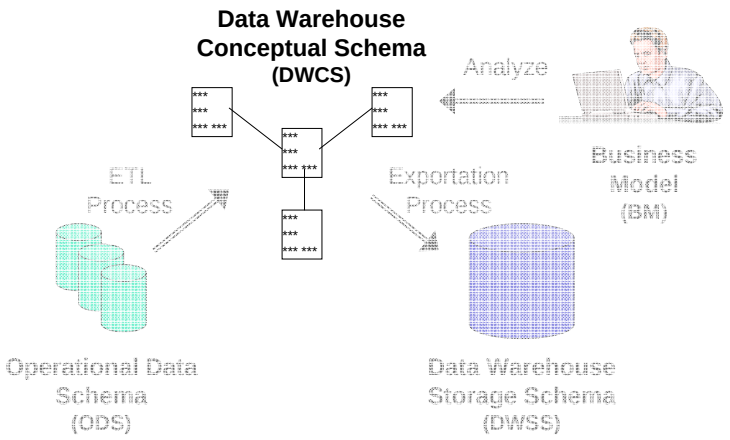
Overall Method based on the UML

Based on the Unified Process (UP)

- ***Design***:
  - DW conceptual model
  - *Source to target data mapping diagram*
- Method:
  - *Data Warehouse Conceptual Schema* → *Class diagram*
  - *Client Conceptual Schema* → Class diagram
  - *Data Mapping* → Diagrama de clases Class diagram

Overall Method based on the UML

Based on the Unified Process (UP)



Overall Method based on the UML

Based on the Unified Process (UP)

	Source (S) (OLTP, external data, ...)	Integration	Data Warehouse (DW)	Customization	Client (C) (OLAP, data mining, ...)
Conceptual	**SCS** Class diagram *Standard UML*	**DM** Class diagram *Data Mapping Profile*	**DWCS** Class diagram *Standard UML* *Multidimensional Profile*	**DM** Class diagram *Data Mapping Profile*	**CCS** Class diagram *Standard UML* *Multidimensional Profile*
Logical	**SLS** Class diagram *Different data modeling profiles*	**ETL Process** Class diagram *ETL Profile*	**DWLS** Class diagram *Different data modeling profiles*	**Exporting Process** Class diagram *ETL Profile*	**CLS** Class diagram *Different data modeling profiles*
Physical	**SPS** Comp. & deploy. diagrams *Database Deployment Profile*	**Transportation Diagram** Deployment diagram *Database Deployment Profile*	**DWPS** Comp. & deploy. diagrams *Database Deployment Profile*	**Transportation Diagram** Deployment diagram *Database Deployment Profile*	**CPS** Comp. & deploy. diagrams *Database Deployment Profile*

LEGEND: CS: Conceptual Schema, LS: Logical Schema, PS: Physical Schema, Comp. & deploy: Component and deployment

Outline

- Introduction
- Overall Method based on the UML
- MDA: Model Driven Architecture
- Data Warehouse Quality
- Other issues: Security and Data Mining
- CASE tools
- Conclusions and further works

Outline

- Introduction
- Overall Method based on the UML
  - A UML profile for conceptual MD modeling
- MDA: Model Driven Architecture
- Data Warehouse Quality
- Other issues: Security and Data Mining
- CASE tools
- Conclusions and further works

Overall Method based on the UML

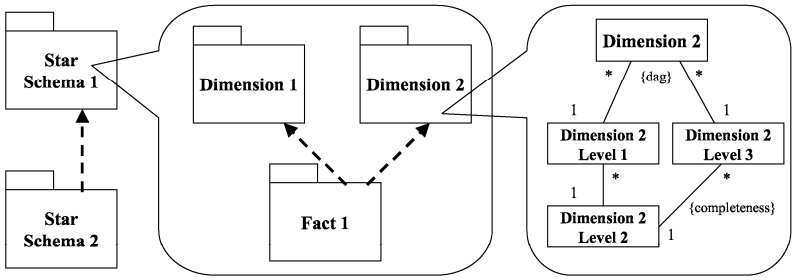
UP (Method). UML profile for MD modeling

- *Data Warehouse Conceptual Schema*
- Proposal: *UML Profile for Multidimensional Modeling*
  - Use of the Object Constraints Language (OCL)
- Basic components:
  - Facts and dimensions
- Represented properties:
  - Shared dimensions
  - Heterogenous dimensions
  - Degenerated facts and dimensions
  - Multiple and alternative classification hierarchies
  - …

- Scientific production:
  - ER'02, UML'02, JISBD'02, ER'03, etc.
  - IEEE Computer 2001, JDM'04, JDM'06, DKE'06

Overall Method based on the UML

UP (Method). UML profile for MD modeling

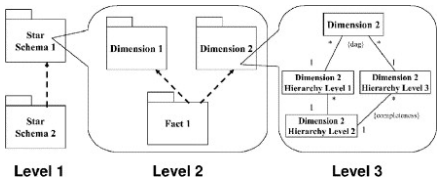


Level 1	Level 2	Level 3
Model Definition	Star schema Definition	Fact/Dimension Definition

Multidimensional modeling guidelines for using packages

No.	Level	Guideline
0a		At the end of the design process, the MD model will be divided into three levels: model definition, star schema definition, and dimension/fact definition
0b		Before starting the modeling, define facts and dimensions and remark the shared dimensions and dimensions that share some hierarchy levels
1	1	Draw a package for each star schema, i.e., for every fact considered
2a	1	Decide which star schemas will host the definition of the shared dimensions; according to this decision, draw the corresponding dependencies
2b	1	Group together the definition of the shared dimensions in order to minimize the number of dependencies
2c	1	Avoid cycles in the dependency structure
3	2	Draw a package for the fact (only one in a star package) and a package for each dimension of the star schema
4a	2	Draw a dependency from the fact package to each one of the dimension packages
4b	2	Never draw a dependency from a dimension package to a fact package
5	2	Do not define a dimension twice; if a dimension has been previously defined, import it
6	2	Draw a dependency between dimension packages in order to indicate that the dimensions share hierarchy levels
7	3	In a dimension package, draw a class for the dimension class (only one in a dimension package) and a class for every classification hierarchy level (the base classes)
8	3	In a fact package, draw a class for the fact class (only one in a fact package) and import the dimension classes with their corresponding hierarchy levels
9	3	In a dimension package, if a dependency from the current package has been defined at level 2, import the corresponding shared hierarchy levels
10	3	In a dimension package, when importing hierarchy levels from another package, it is not necessary to import all the levels

Design guidelines

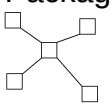


Overall Method based on the UML

UP (Method). UML profile for MD modeling

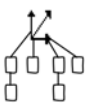
- Main stereotypes of the first extension (based on UML 1.5)

Package Stereotypes



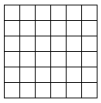
StarPackage
(Level 1)

FactPackage
(Level 2)

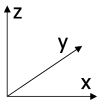


DimensionPackage
(Level 2)

Class Stereotypes



Fact
(Level 3)



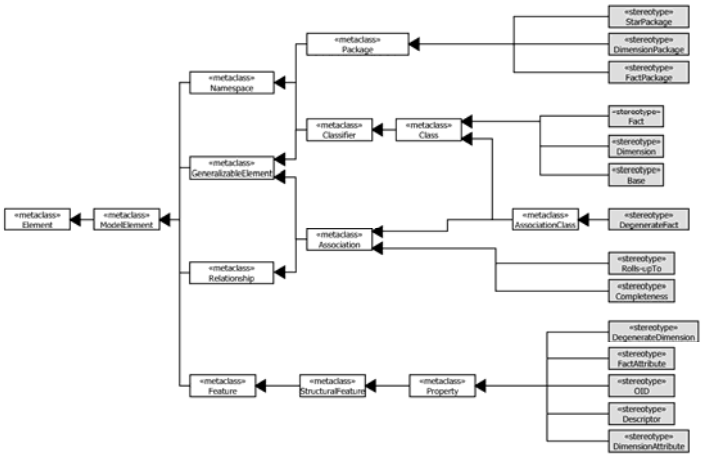
Dimension
(Level 3)

Base
(Level 3)

Overall Method based on the UML

UP (Method). UML profile for MD modeling

- Proposal based on the UML 2.0 (DKE'06)

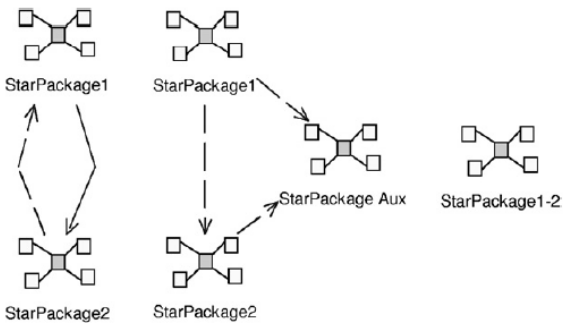


Overall Method based on the UML

UP (Method). UML profile for MD modeling

Model definition (Level 1)

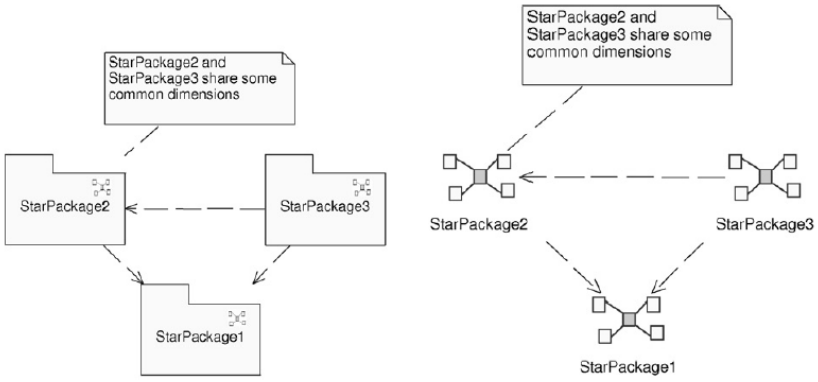
<<StarPackage>>



Overall Method based on the UML

UP (Method). UML profile for MD modeling

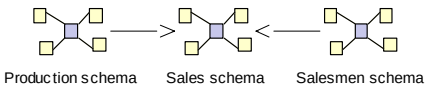
Model definition (Level 1): Different representations
<<StarPackage>>



Overall Method based on the UML

UP (Method). UML profile for MD modeling

Model definition (Level 1)
<<StarPackage>>

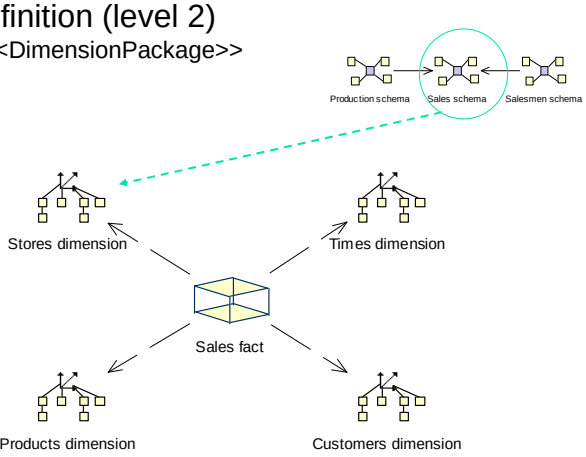


Overall Method based on the UML

UP (Method). UML profile for MD modeling

Star Schema definition (level 2)

<<FactPackage>>, <<DimensionPackage>>

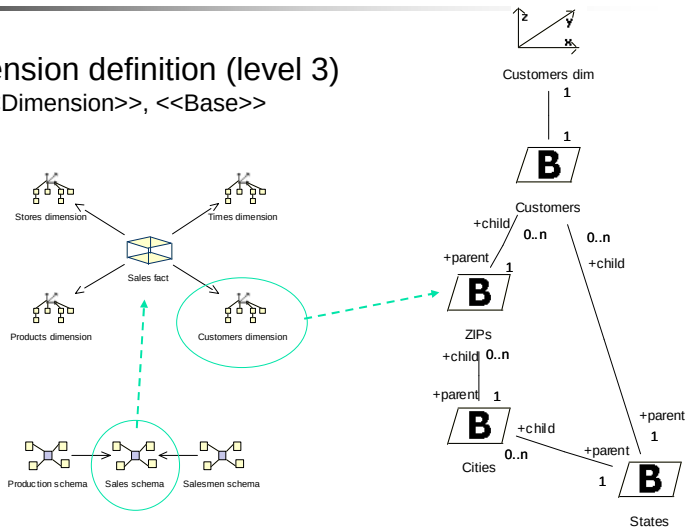


Overall Method based on the UML

UP (Method). UML profile for MD modeling

Fact/Dimension definition (level 3)

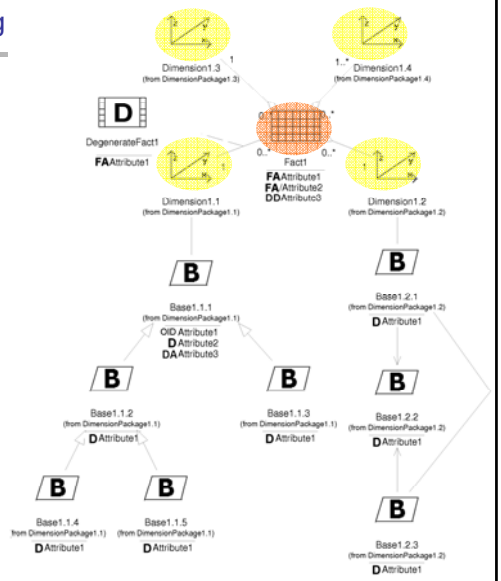
<<Fact>>, <<Dimension>>, <<Base>>



Overall Method based on the UML

UML profile for MD modeling

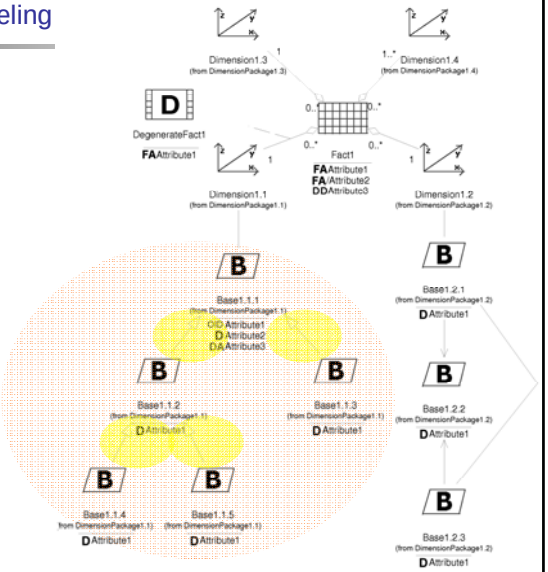
- Level 3: Content of facts and dimensions



Overall Method based on the UML

UML profile for MD modeling

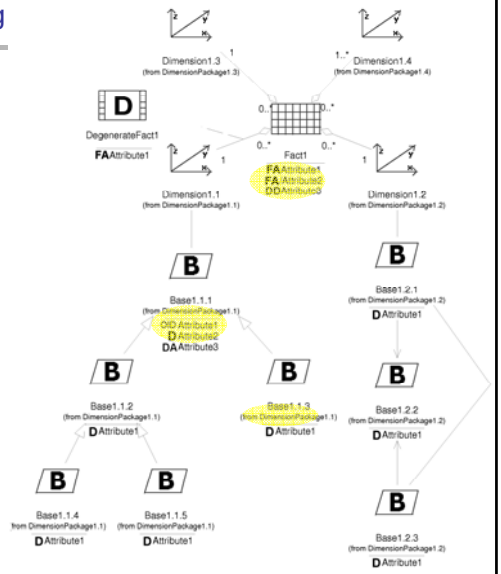
- Level 3: Content of fact and dimensions
  - Categorization hierarchies



Overall Method based on the UML

UML profile for MD modeling

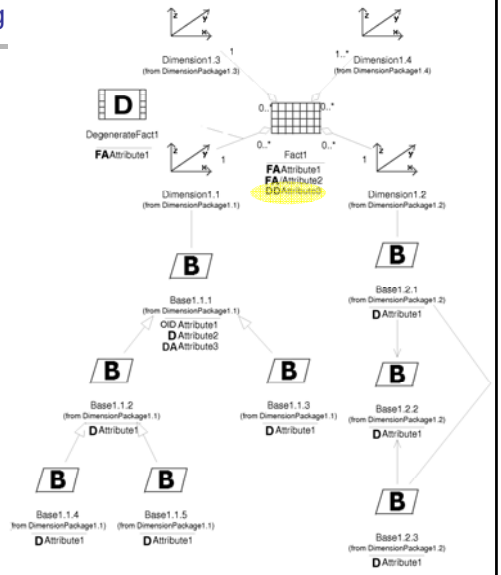
- Level 3: Content of fact and dimensions
  - Attributes
    - FA
    - DD
    - OID
    - D
    - DA



Overall Method based on the UML

UML profile for MD modeling

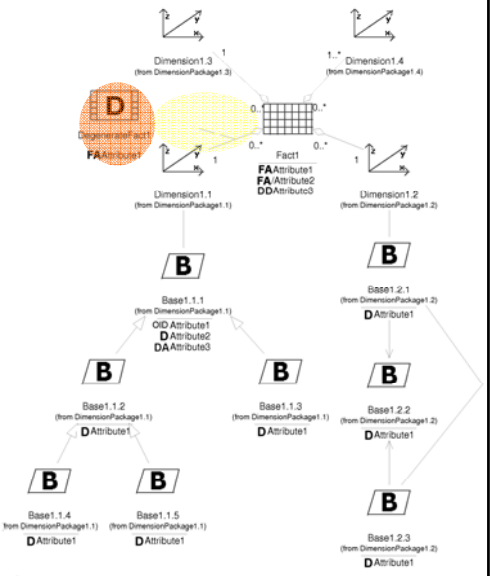
- Level 3: Content of fact and dimensions
  - Degenerated dimensions



Overall Method based on the UML

UML profile for MD modeling

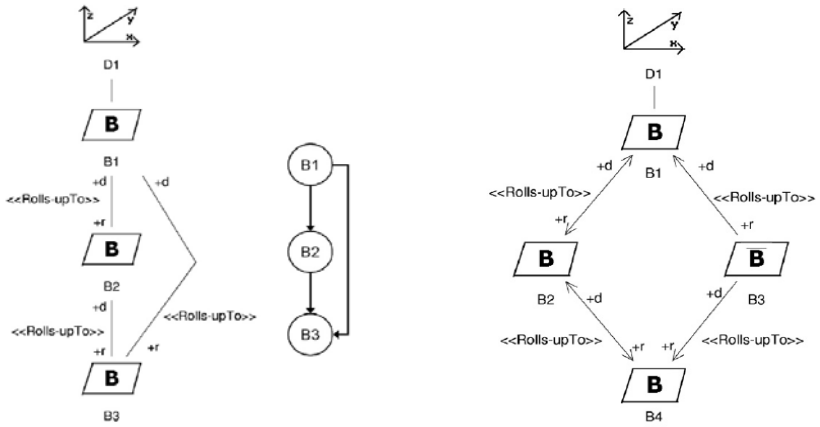
- Level 3: Content of fact and dimensions
  - Degenerated facts



Overall Method based on the UML

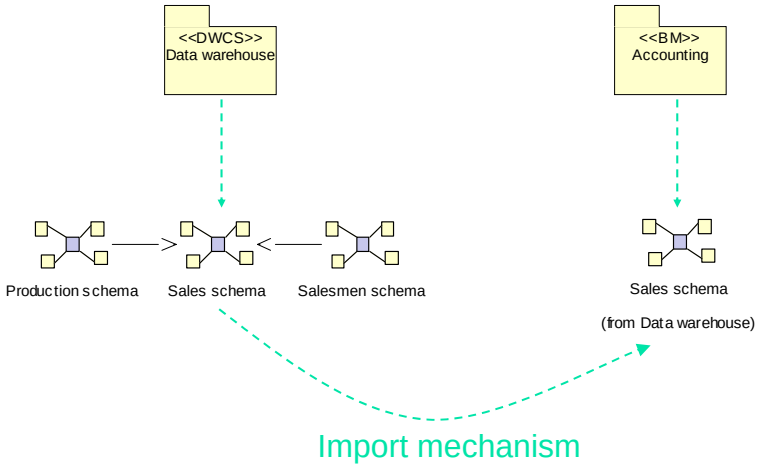
UML profile for MD modeling

- Level 3: Classification hierarchies



Overall Method based on the UML

UP (Method). UML profile for MD modeling



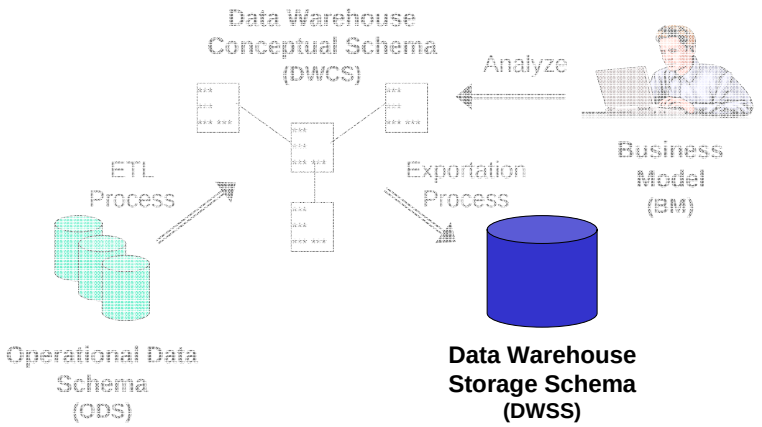
Overall Method based on the UML

Based on the Unified Process (UP)

- ***Implementation***
  - Build of physical structures of the DW
- Method:
  - Logical and physical schema
    - Class diagram
    - Component diagram
    - Deployment diagram
  - Client Logical and physical schema
    - Class diagram
    - Component diagram
    - Deployment diagram
  - **ETL** processes → Class diagram
  - **Exportation** → Class diagram
  - **Transport** → Deploymenyt diagram

Overall Method based on the UML

Based on the Unified Process (UP)



Overall Method based on the UML

Based on the Unified Process (UP)

	Source (S) (OLTP, external data, ...)	Integration	Data Warehouse (DW)	Customization	Client (C) (OLAP, data mining, ...)
Conceptual	**SCS** Class diagram *Standard UML*	**DM** Class diagram *Data Mapping Profile*	**DWCS** Class diagram *Standard UML* *Multidimensional Profile*	**DM** Class diagram *Data Mapping Profile*	**CCS** Class diagram *Standard UML* *Multidimensional Profile*
Logical	**SLS** Class diagram *Different data modeling profiles*	**ETL Process** Class diagram *ETL Profile*	**DWLS** Class diagram *Different data modeling profiles*	**Exporting Process** Class diagram *ETL Profile*	**CLS** Class diagram *Different data modeling profiles*
Physical	**SPS** Comp. & deploy. diagrams *Database Deployment Profile*	**Transportation Diagram** Deployment diagram *Database Deployment Profile*	**DWPS** Comp. & deploy. diagrams *Database Deployment Profile*	**Transportation Diagram** Deployment diagram *Database Deployment Profile*	**CPS** Comp. & deploy. diagrams *Database Deployment Profile*

LEGEND: CS: Conceptual Schema, LS: Logical Schema, PS: Physical Schema, Comp. & deploy: Component and deployment

Overall Method based on the UML

Based on the Unified Process (UP)

- *Data Warehouse Storage Schema (DWSS)*
- Specification depending on the implementation (RDMS, ORDBMS, MD, ...) → Similar to ODS
- Two possibilities: manual o automatic
- Advantages:
  - Treacability from the conceptual to physical model
  - Reduce development costs as implementation details are covered in the early stages of a DW project
  - Different levels of abstraction
  - Scientific production
    - DOLAP'04, JDM'06 (Luján-Mora, Trujillo)

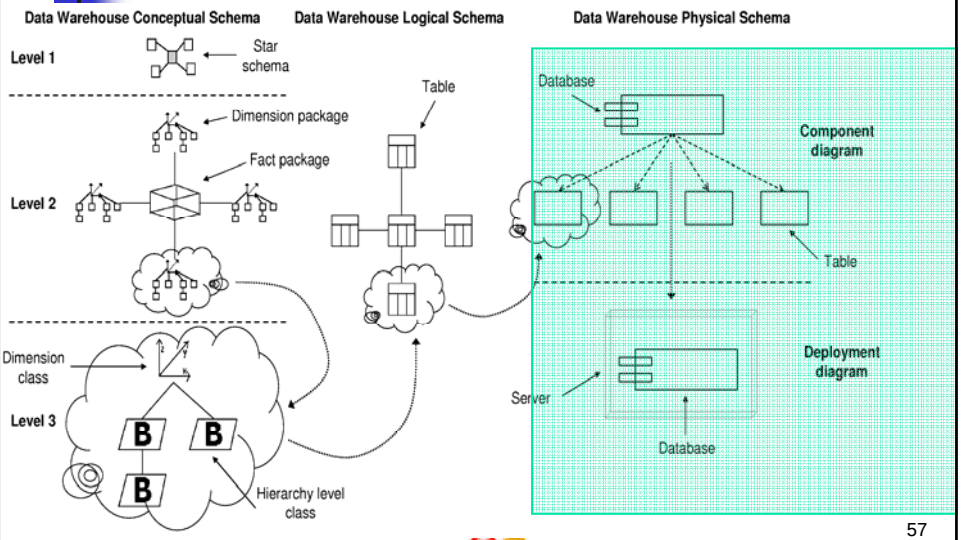
Overall Method based on the UML

Based on the Unified Process (UP). Physical design

- Implementation decisions:
  - Storing of different disks
  - Replication
  - Vertical and horizontal partitioning
  - Influence the final mantenability and performance
  - ...
- Solution:
  - To cover physical design since the early stages of the development
    - Allow designers to anticipate decisions of the physical design
    - Reduce development time and costs

Overall Method based on the UML

Based on the Unified Process (UP). Physical design



Overall Method based on the UML

Based on the Unified Process (UP). Physical design

- Extended Component and deployment diagrams
  - Database Deployment Profile
    - <<Database>>, <<Tablespace>>, <<Table>>, etc.

Diagrams

Component and Deployment Diagrams
- Source Physical Schema
- Data Warehouse Physical Schema
- Client Physical Schema

Deployment Diagrams
- Integration Transportation Diagram
- Customization Transportation Diagram

Overall Method based on the UML

Based on the Unified Process (UP). Physical design. Example

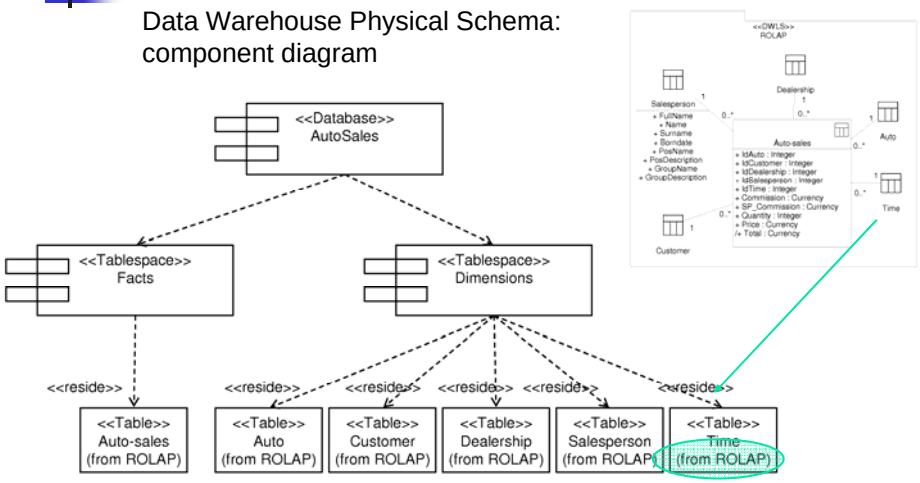
- Example:
  - DW with the daily sales of vehicles (cars and trucks)
  - Dimensions: vehicle, client, Store, salesman, time
  - Two data sources
    - Sales server: transactions and sales
    - CRM server: clients
  - Different user requirements :
    - MacOS y Windows
    - web and desktop interfaces

Overall Method based on the UML

Based on the Unified Process (UP). Physical design. Example

DWLS

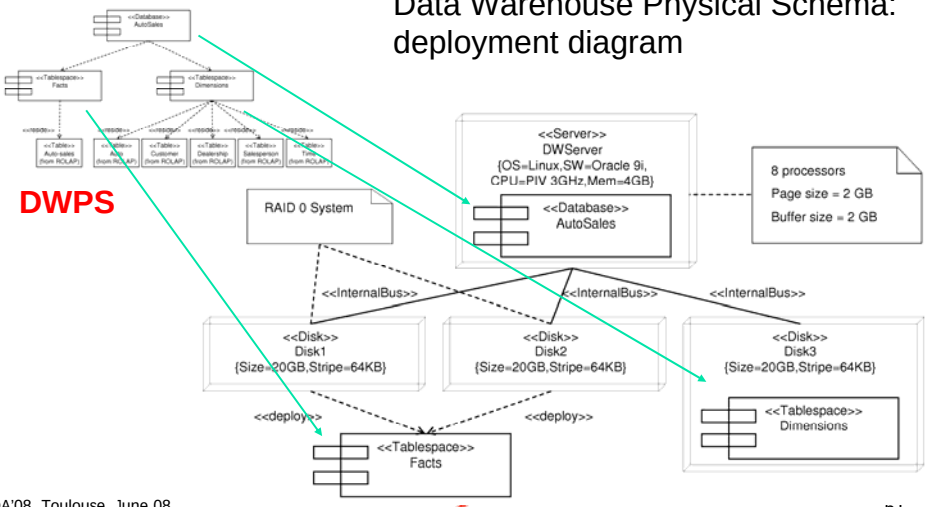
Data Warehouse Physical Schema: component diagram



Overall Method based on the UML

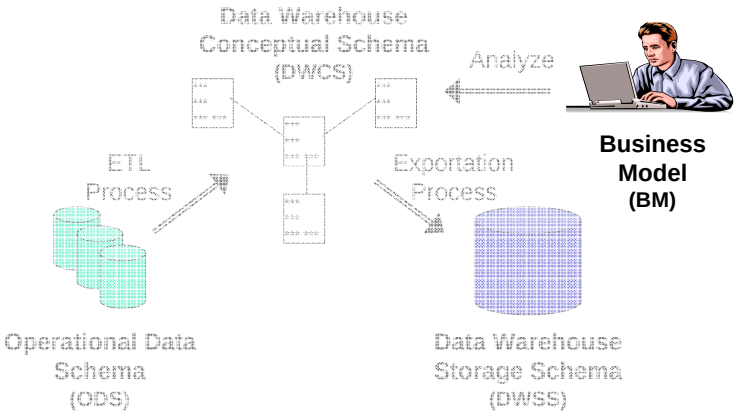
Based on the Unified Process (UP). Physical design. Example

Data Warehouse Physical Schema: deployment diagram



Overall Method based on the UML

Based on the Unified Process (UP).



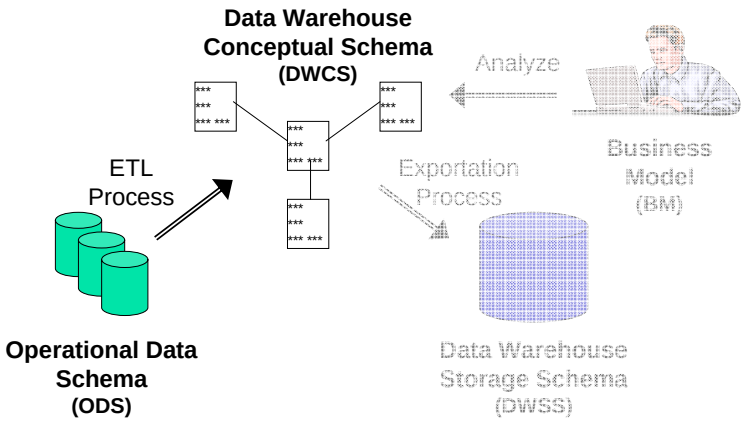
Overall Method based on the UML

Based on the Unified Process (UP).

- *Business Model (BM)*
- Adapt the DW to the different users:
  - Easier to understand
  - Including security aspects
  - ...
- UML importing mechanim→ Different sub-models from the DWCS

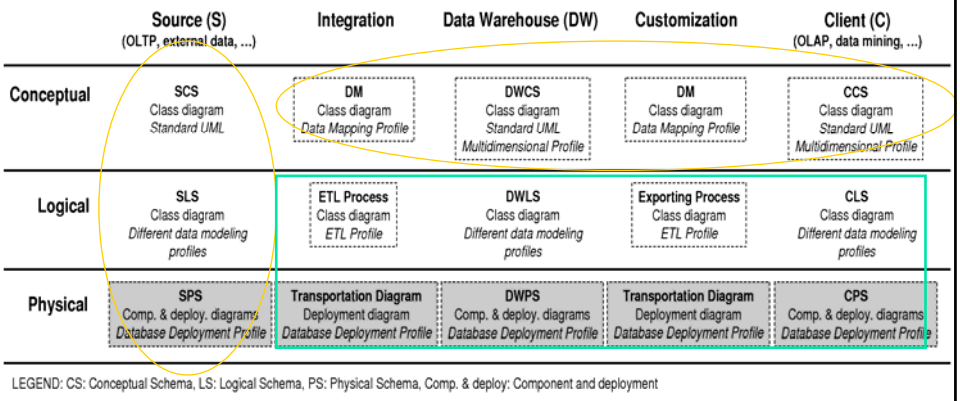
Overall Method based on the UML

Based on the Unified Process (UP). ETL process design



Overall Method based on the UML

Based on the Unified Process (UP).



Overall Method based on the UML

Based on the Unified Process (UP). ETL process design

- *Extraction-Transformation-Loading*
- Mapping between ODS y DWCS
- Proposal:
  - ***Conceptual**: data mapping diagrams*
  - ***Logical**: UML Profile for Modeling ETL Processes*
- Common transformation pallette:
  - Integrating different data sources
  - Trasnformations
  - Generation of *surrogate keys*
  - ...

Overall Method based on the UML

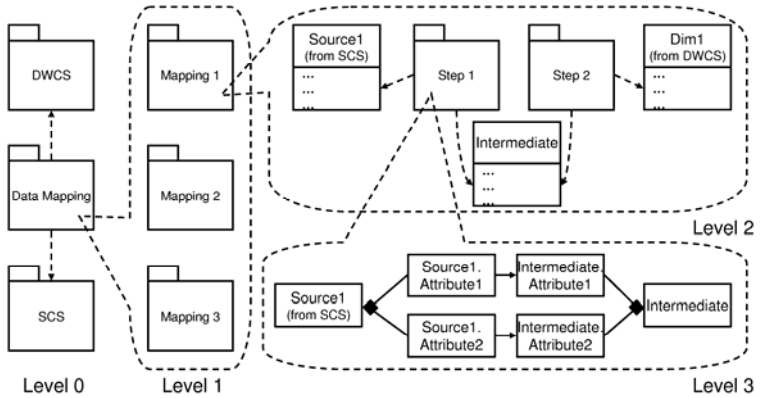
Based on the Unified Process (UP). ETL process design

- Conceptual level
  - Data Mapping Diagram (DMD) to represent attribute transformations at the conceptual level
- Logical level
  - ETL proceses modeling
    - Basic and complete operations

Overall Method based on the UML

Based on the Unified Process (UP). ETL process design

- Different levels of the Data Mapping Diagram (DMD)
  - At the conceptual level



Overall Method based on the UML

Based on the Unified Process (UP). ETL process design

- Summary: More detail in ER'04 (Luján-Mora, Trujillo, Vassiliadis)

Overall Method based on the UML

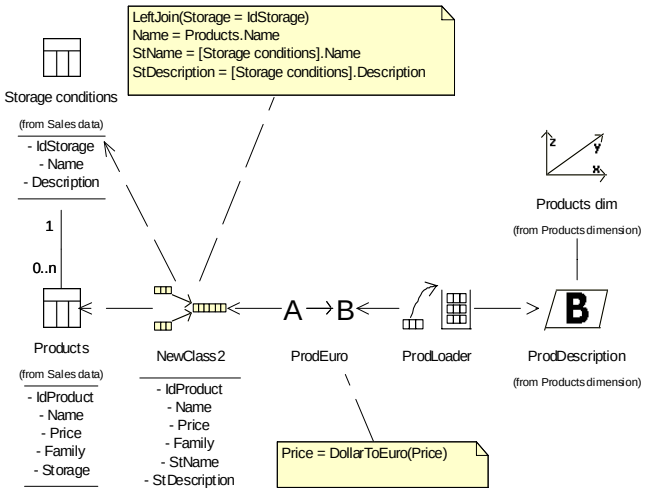
Based on the Unified Process (UP). ETL process design

- Logical level: stereotypes for operations

	Aggregation		Loader
	Conversion		Log
	Filter		Merge
	Incorrect		Surrogate
	Join		Wrapper

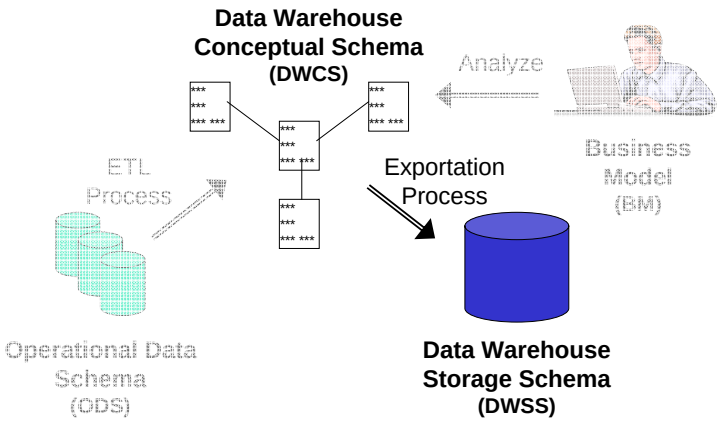
Overall Method based on the UML

Based on the Unified Process (UP). ETL process design. Example



Overall Method based on the UML

Based on the Unified Process (UP).



Overall Method based on the UML

Based on the Unified Process (UP).

- *Exportation Process*
  - Mapping between DWCS and DWSS

Overall Method based on the UML

Based on the Unified Process (UP).

- **Test:**
  - No new diagrams
- ***Maintenance:***
  - Come back to a new iteration → Requirments
    - It affects all diagrams
- ***Post-development review*:**
  - It is not part of the development efort
  - It helps improving future projects

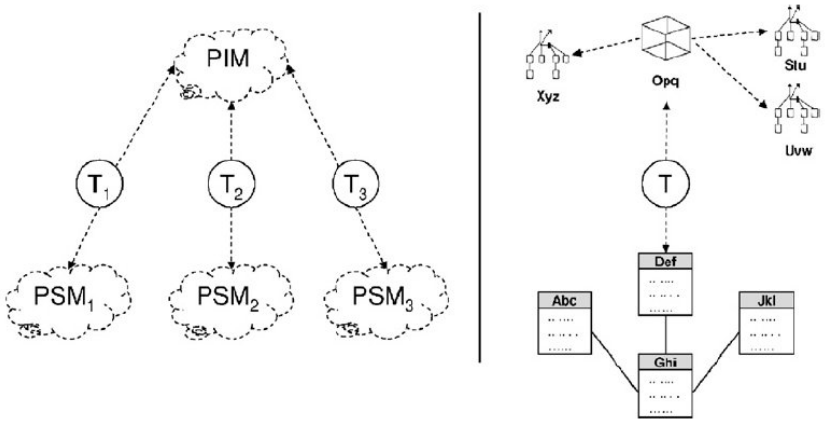
Outline

- Introduction
- Overall Method based on the UML
- MDA: Model Driven Architecture
- Data Warehouse Quality
- Other issues: Security and Data Mining
- CASE tools
- Conclusions and further works

Outline

- Introduction
- Overall Method based on the UML
- MDA: Model Driven Architecture
- Data Warehouse Quality
- Other issues: Security and Data Mining
- CASE tools
- Conclusions and further works

MDA: Model Driven Architecture



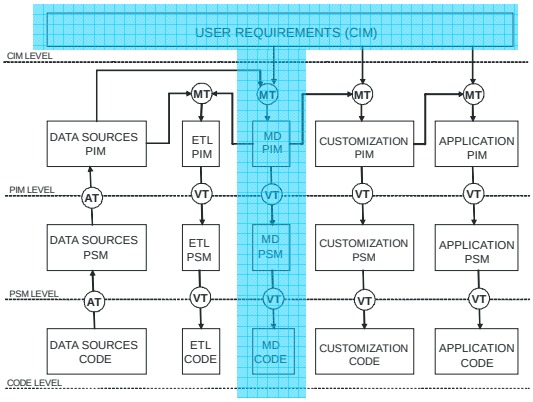
MDA: Model Driven Architecture

Transforming facts into tables

```
mapping FactToTable {
  domain {
    (Fact) [name = fn, attibutes = atts, associations = ass]
  }
  body {
    ta = new Table()
    ta.name = fn
    ta.columns = atts->iterate(a cols = {} | cols + FactAttributeToColumn(a))
    ta.keys = atts->forAll(a keys = {} | keys + DDToKey(a))
    ta.foreignKeys = ass->forAll(a fkeys = {} | fkeys + AggregationToForeignKey(a))
  }
}
mapping AggregationToForeignKey {
  domain {
    (Association)[name = fn, source = aggS, destination = aggD]
  }
  body {
    fk = new ForeignKey()
    fk.name = fn
    // ForeignKey is autoincrement
    fk.type = 'auto'
  }
}
```

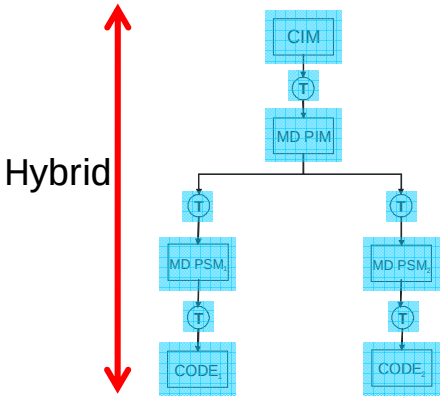
MDA: Model Driven Architecture

- MDA framework



MDA: Model Driven Architecture

- MDA framework



CIM: modelos basados en i*
MD PIM: profile UML para modelado multidimensiona
MD PSM$_1$: metamodelo relacional de CWM.
CODE$_1$: código SQL.
MD PSM$_2$: metamodelo multidimensional de CWM.
CODE$_2$: código para Oracle Express.
T: transformaciones entre modelos.

Outline

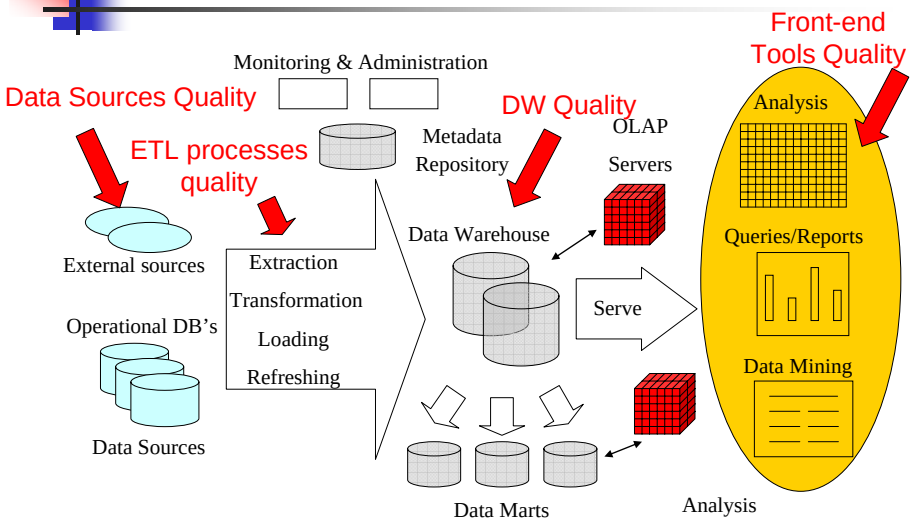
- Introduction
- Overall Method based on the UML
- MDA: Model Driven Architecture
- Data Warehouse Quality
- Other issues: Security and Data Mining
- CASE tools
- Conclusions and further works

Outline

- Introduction
- Overall Method based on the UML
- MDA: Model Driven Architecture
- Data Warehouse Quality
- Other issues: Security and Data Mining
- CASE tools
- Conclusions and further works

Data Warehouse Quality

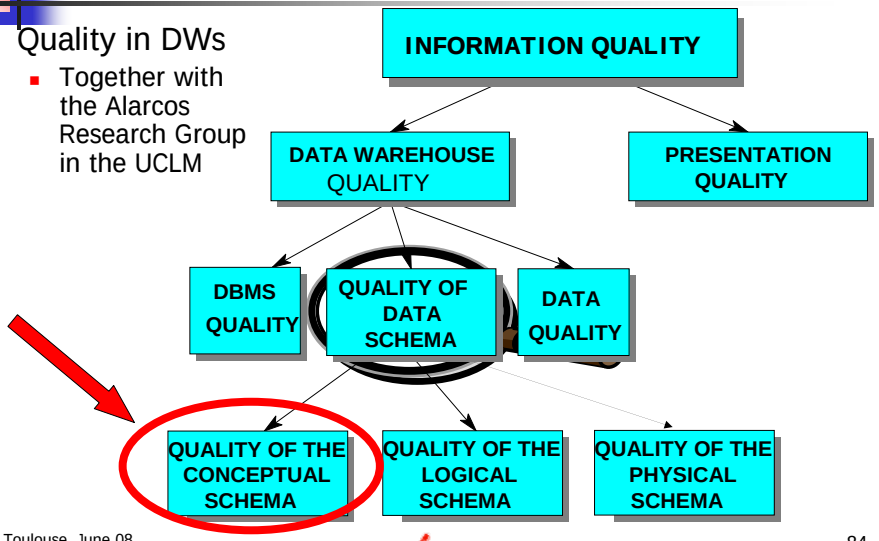
Quality Risks



Data Warehouse Quality

Model Metrics

- Quality in DWs
  - Together with the Alarcos Research Group in the UCLM



Data Warehouse Quality
Model Metrics

- In the last years,
  - Different conceptual modeling approaches (see Int.)
  - Different authors have proposed interesting recommendations for achieving a “good” DW data model
- However, quality criteria are not enough on their own to ensure quality in practice
- There is a lack of more objective indicators (measures) to guide the designer

Data Warehouse Quality
Model Metrics

- Aim: To replace the intuitive notions of quality by quantitive and formal measures to reduce the subjectivity and slant in the evaluation
- In doing a data modeling task progressing from an artisan activity to an engineering discipline, desirable quality in data schema must be explicit (Lindland et al., 1994).

Data Warehouse Quality

Model Metrics

- A measure is a way to measure a quality factor in a consistent, explicit and objective way.
- Measures for data warehouse schemas
  - The Data Warehouse Quality (DWQ, Jarke y Vassiliou, 1997)
    - Measures for data quality (not for models)
    - Freshness, Consistency, Accuracy y Completeness
  - Si-Saïd y Prat (2003)
    - Conceptual schemas
    - Measure Simplicity and Analizability
    - Neither theoretical nor empirical validation

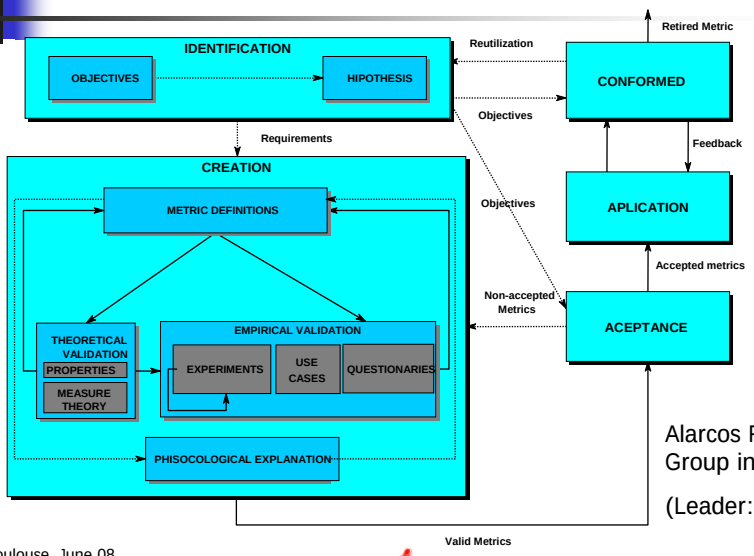
Data Warehouse Quality

Model Metrics

- Other metrics for OO conceptual models:
  - Chidamber and Kemerer (1991; 1994)
  - Li and Henry (1993)
  - Brito e Abreu and Carapuça (1994)
  - Lorenz and Kidd (1994)
  - Briand et al.´s (1997)
  - Marchesi (1998)
  - Harrison et al. (1998)
  - Banisya et al. (1999)

Data Warehouse Quality

Model metrics. Method



Alarcos Research Group in the UCLM

(Leader: M. Piattini)

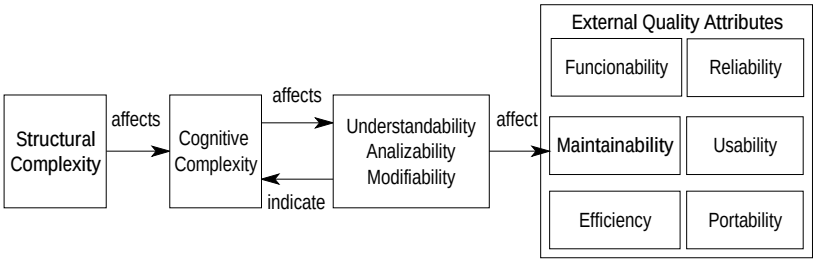
Data Warehouse Quality

Model Metrics: Method

- The final aim of our work is to define a set of product metrics to assure the data warehouse quality.
- We focus on the conceptual model complexity that affects understandability

Data Warehouse Quality
Model Metrics.

- Hypothesis



Relationship between structural properties, cognitive complexity, understandability and external quality attributes (Briand et al., 1999)

Data Warehouse Quality
Model Metrics.

- UML profile for MD modeling developed in U. Alicante
  - Last version. DKE 2006

Nombre	Descripción	Icono
Fact	Las Clases de este estereotipo representan hechos en un modelo multidimensional	
Dimension	Las Clases de este estereotipo representan dimensiones en un modelo multidimensional	
Base	Las Clases de este estereotipo representan niveles de una jerarquía dimensional en un modelo multidimensional	B

Nombre	Descripción	Icono
OID	Los atributos con este estereotipo representan los OID de clases factuales, dimensionales o base	**OID**
FactAttribute	Los atributos con este estereotipo representan atributos de clases factuales	**FA**
Descriptor	Los atributos con este estereotipo representan atributos descriptores de clases dimensionales o base	**D**
DimensionAttribute	Los atributos con este estereotipo representan atributos de clases dimensionales o base	**DA**

Data Warehouse Quality

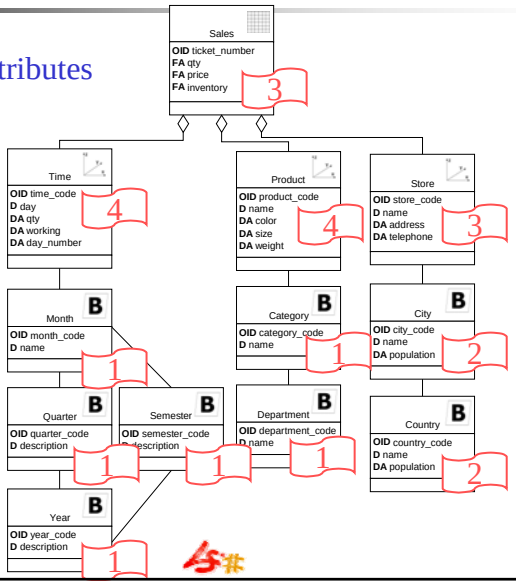
Model Metrics. Measures for MD conceptual Models.

Class	Star	Schema
NA(C)	NA(S)	NA
NR(C)	-	-
-	NH(S)	NH
-	-	NFC
-	NDC(S)	NDC
-	NBC(S)	NBC
-	NC(S)	NC
-	-	NSDC
-	NADC(S)	NADC
-	NAFC(S)	NAFC
-	NABC(S)	NABC
-	-	NASDC
-	DHP(S)	DHP
-	RBC(S)	RBC
-	-	RDC
-	RSA(S)	RSA

Data Warehouse Quality

Model Metrics. Metrics for MD conceptual Models. Class metrics

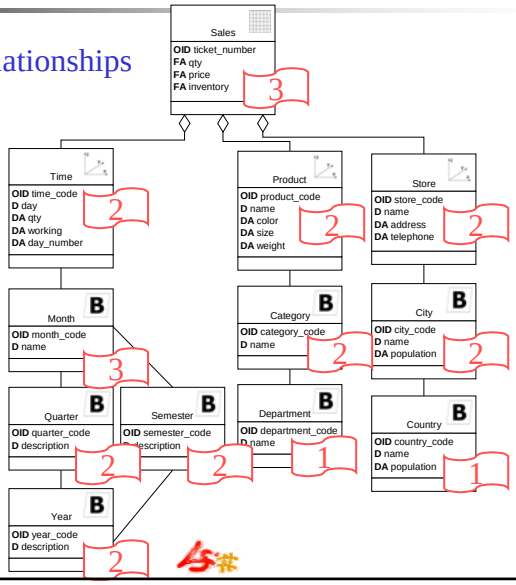
NA: Number of attributes



Data Warehouse Quality

Model Metrics. Metrics for MD conceptual Models. Class metrics

NR: Number of Relationships



Data Warehouse Quality

Model Metrics. Theoretical validation

Theoretical Validation

- Measures have been theoretically validated by using:
  - The DISTANCE formal framework (Poels y Dedene, 2000)
  - Property-based software engineering measurement (Briand et al., IEEE TOSE, 96)
- All the measures are in the ratio scale
  - The Zuse framework is not suitable
- For further read on our theoretical validation
  - Serrano, M., Calero, C., Trujillo, J., Lujan, S. and Piattini, M. *(QSSE 2004), ISOFT'2007*

Data Warehouse Quality

Model Metrics. Empirical Validation.

Empirical Validation

Goal definition

To analyze	the metrics for datawarehouse conceptual models
for the purpose of	evaluating if they can be used as useful mechanisms
with respect of	the datawarehouse maintainability
from the point of view of	designer
In the context of	profesionals

- For further read on Empirical validation
  - Serrano, M., Calero, C., Trujillo, J., Lujan, S. and Piattini, M. *(CAiSE, 2004; ISOFT 2007)*

Outline

- Introduction
- Overall Method based on the UML
- Data Warehouse Quality
- Other issues: Security and Data Mining
- CASE tools
- Conclusions and further works

Other issues: Security

- Data Warehouses
  - Very powerful Mechanism to discover crucial information
- The survival of the organizations depends on the correct management, security and confidentiality of the information
- Majority of methodologies that incorporate security are based on OLTP and not OLAP
- Laws of protection of data
- It is important to protect the Data stores

Other issues: Security

- The protection of Data Warehouses is a serious requirement that must be considered in all the stages of life-cycle.
- Currently, there exist
  - Methodologies to design DWs, but…
    - They do not consider security aspects in their stages.
  - Proposals to integrate security in the development of DWs, but ...
    - Neither in all design stages, nor in the MD modeling

Other issues: Security

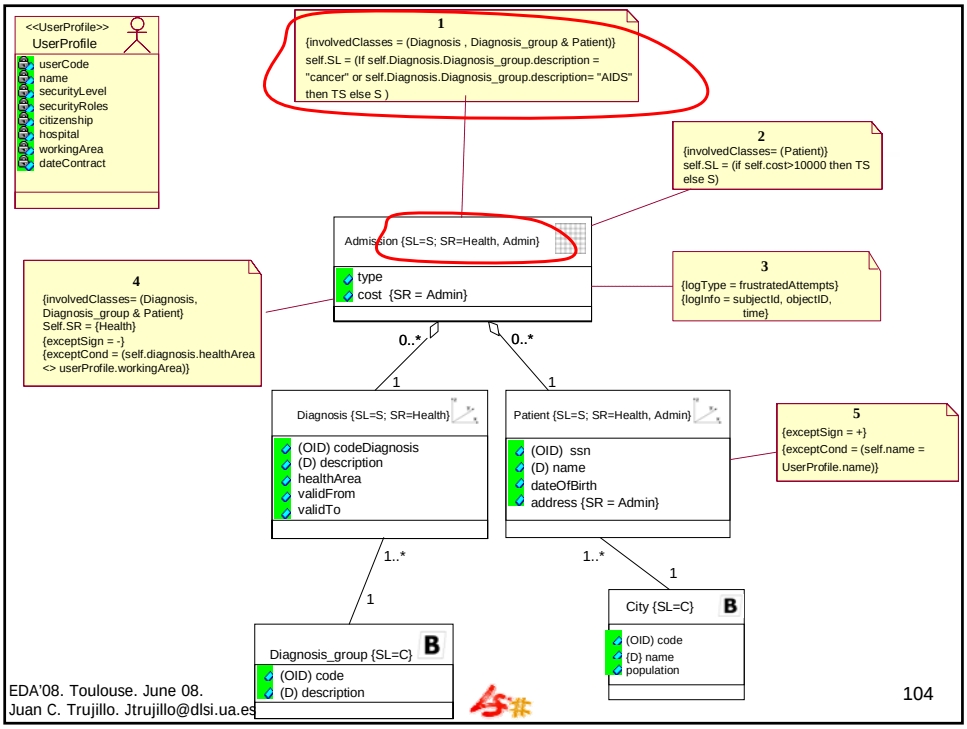
- To create a profile for MD modeling at the conceptual level:
  - Classifying (using multi-level security) information and users following our ACA (Acess Control and Audit) model
  - Specify constraints, authorization and adit rules
  - To implement safe secure MD models with commerical DBMS such as universal OLS (Oracle Label Security) and DB2 Database (UDB).

Other issues: Security

- The aim of this profile in to represent in the same conceptual modeling approach for DWs:
  - MD modeling
  - Acess Control and Audit (ACA) Model
    - Confidentiality Information
    - Authorization Rules
    - Audit Rules
  - Further read:
    - ER'04, JISBD'04, DSS'06, WOSIS'05, JIRP, IS'07, EJIS'07,...
    - (Ferrandez-Medina, Trujillo, Villarroel, Piattini)
      - Together with Alarcos Research Group of UCLM

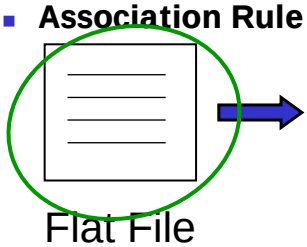
Other issues: Security

- **SIAR1**
  - For each instance of the Admission fact class, if the description of the diagnosis group is specially sensible (cancer or AIDS), then its security level will be High Secret, otherwise will be Secret. This is, when we include Diagnosis in a query, Diagnosis Group and Patient.
  - **OBJECTS CL** Admission **INVCLASSES CL** Diagnosis **AND CL** Diagnosis_Group **AND CL** Patient **COND IF** self.Diagnosis.Group_Diagnos.description="cáncer" or self.Diagnosis.Group_Diagnos.description ="SIDA" **THEN SL** topSecret **ELSE SL** Secret **ENDIF**



Other issues: Data Mining

- **Association Rule** mining



Flat File

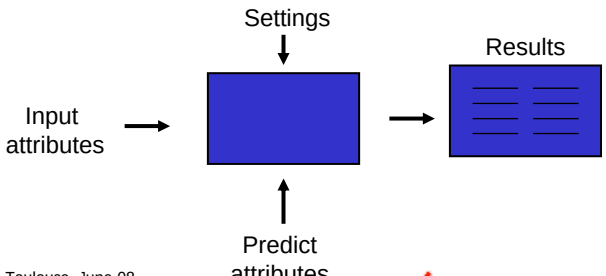
Weaknesses

- Usually several sources
- No integrated data
- No data structure
- The overall vision is lost

- **Data warehouses** as a framework
- Multidimensional analysis
  - Easily represent business data
  - Widely known (star-schema)
  - Focus on main DW concepts

Other issues: Data Mining

- AR look for relationships in an entity
  - The Case
- Input attributes → Predict attributes



Other issues: Data Mining

- In the profile, we propose
  - 18 new tagged values
    - Tagged values for the Model (1)
    - Tagged values for Classes (3)
    - Tagged values for Instances (3)
    - Tagged values for Attributes (3)
    - Tagged values for the Constraints (8)
  - 3 well-formedness rules
    - About Case tagged value
    - About Input and Predict attributes
    - About categorization of continuous values

DAWAK'05-07
DKE'06
ISOFT(submitted)

Other techniques
Time Series
Clustering
Classification
...

Other issues: Data Mining

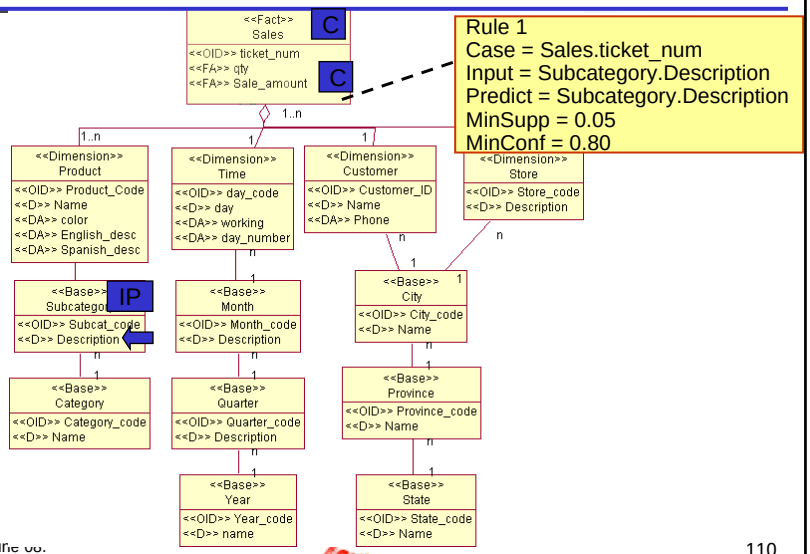
- It allows us to design all type of AR
  - Single level or multiple level AR
  - Single and multidimensional rules
  - Single and multiple predicates rules
  - Hybrid and inter dimension AR

Other issues: Data Mining

Advantages

- The tagged values allow us to design AR rules in a MD model
- Use attribute from MD model
- Put constraints over a MD model as Notes
- Easy to model AR
- Easy to see the structure of AR

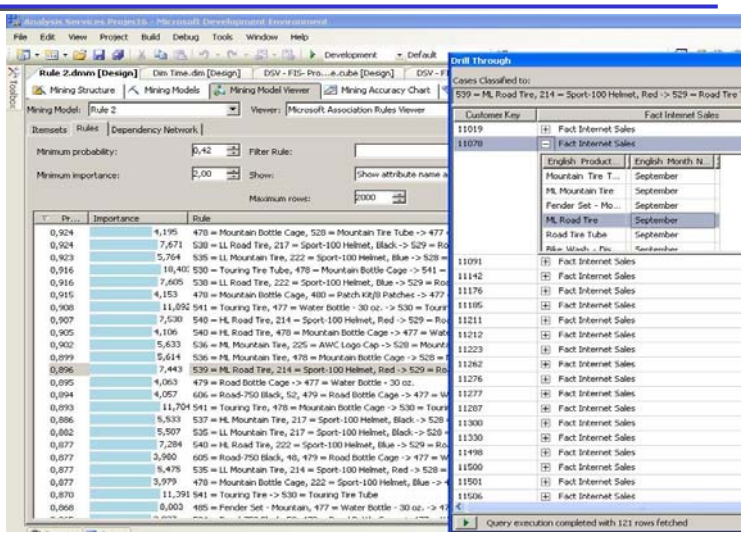
Other issues: Data Mining



Other issues: Data Mining

Implementation:
SQL Server2005

*"**If** helmet and mountain bike **then** road bikes (0.52, 0.3)"*



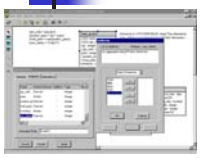
Outline

- Introduction
- Overall Method based on the UML
- Data Warehouse Quality
- Other issues: Security and Data Mining
- CASE tools
- Conclusions and further works

Outline

- Introduction
- Overall Method based on the UML
- Data Warehouse Quality
- Other issues: Security and Data Mining
- CASE tools
- Conclusions and further works

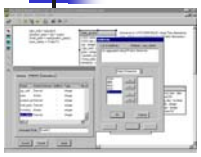
CASE Tools

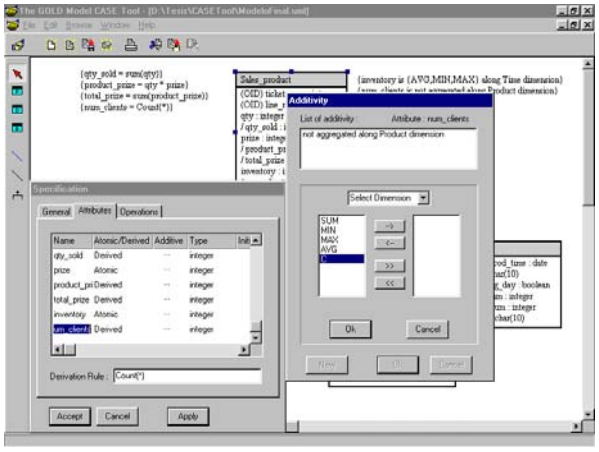


CASE Tools

CASE Tools

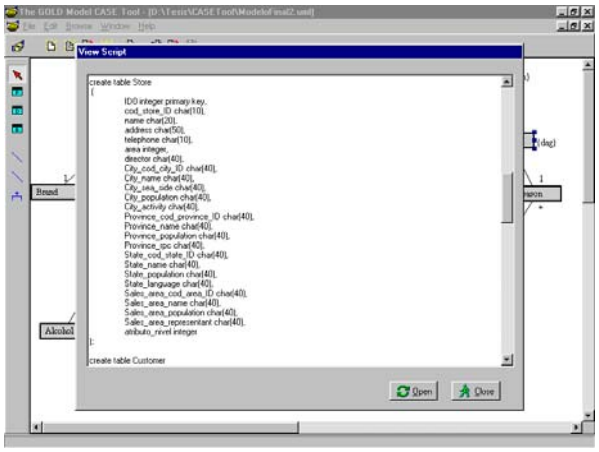
Propetary CASE Tool





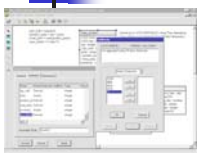
CASE Tools

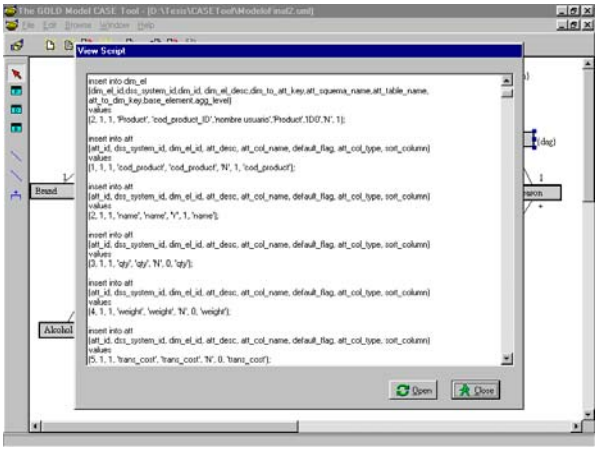
Propetary CASE Tool



CASE Tools

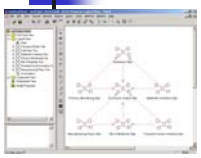
Propetary CASE Tool





CASE Tools

Rational Rose Extension



Add-in For Rational Rose

CASE Tools

Rational Rose Extension

- Rational Rose is one of the most well-known visual modeling tools
- RR is extensible by means of add-ins through the Rose Extensibility Interface:
  - Main menu items
  - Stereotypes
  - Properties (*tagged values*)
  - Data types
  - Event handling
  - Scripts
  - ...

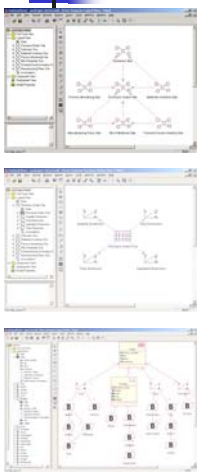
CASE Tools

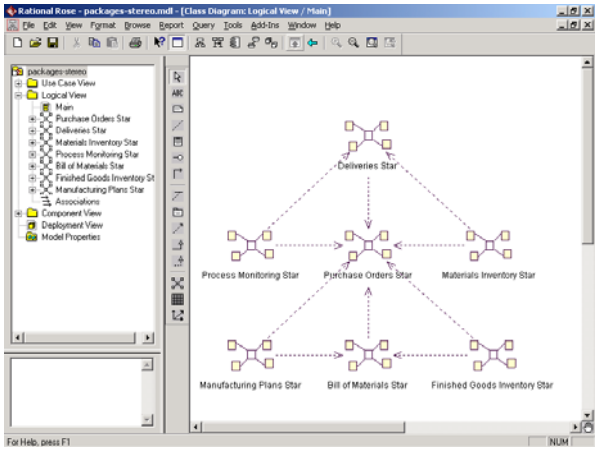
Rational Rose Extension

- Our add-in customizes:
  - Stereotypes → Stereotype configuration file
  - Properties → Property configuration file
  - Constraints
    - Rose Extensibility Interface (REI) does not allow us to specify OCL constraints in a straight way
    - A new menu option: *MD Validate* in the file Menu configuration
      - It runs a Rose script to validate the MD model cheking all the defined constraints

CASE Tools

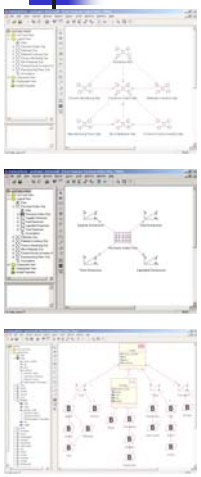
Rational Rose Extension

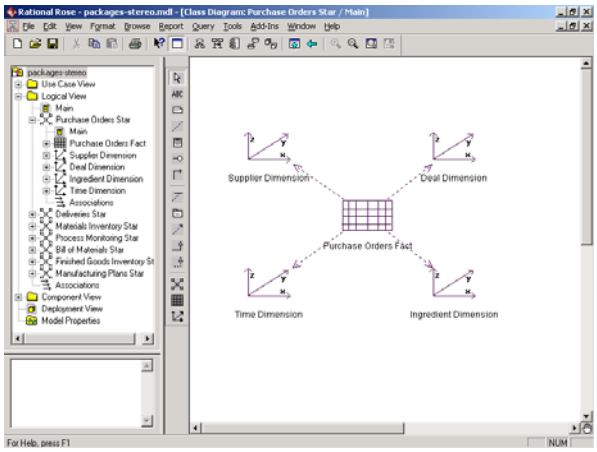




CASE Tools

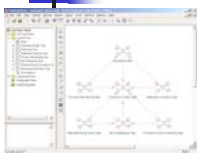
Rational Rose Extension

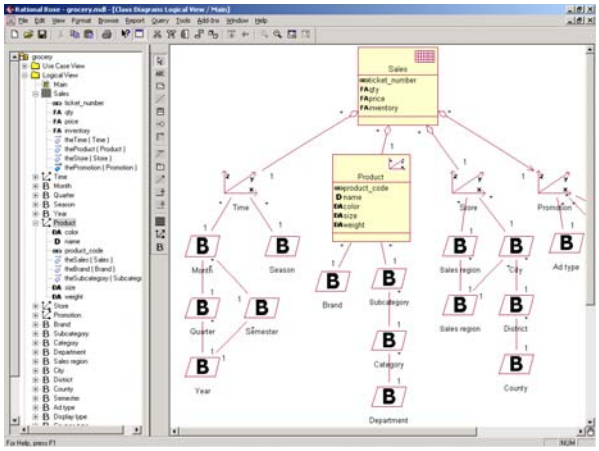




CASE Tools

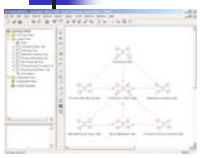
Rational Rose Extension

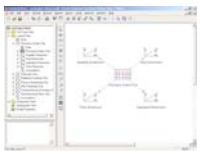


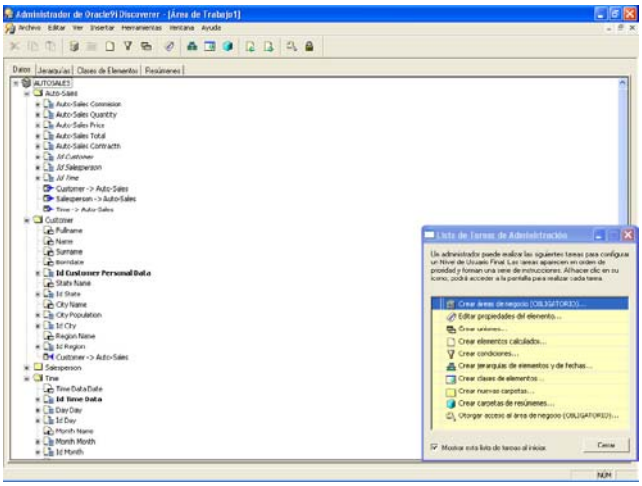


CASE Tools

Rational Rose Extension. Final implementation in commercial tool.

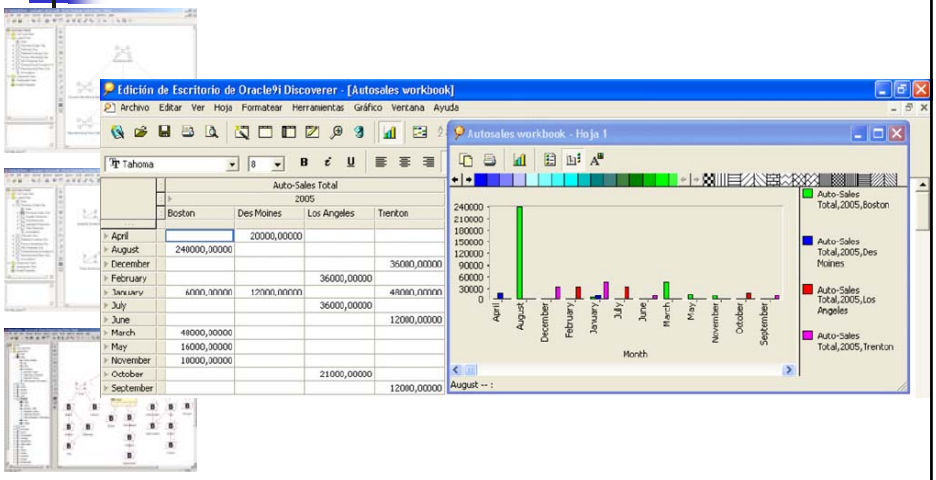






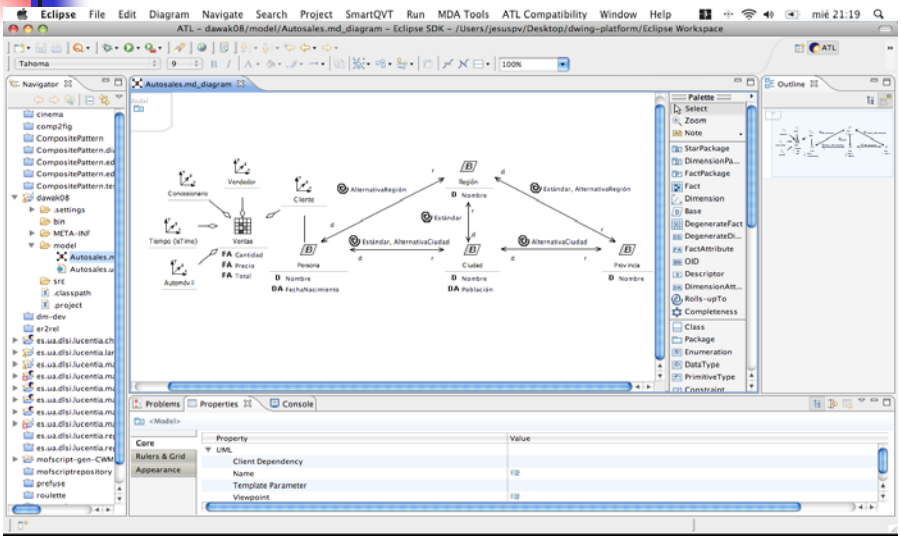
CASE Tools

Rational Rose Extension. Final implementation in commercial tool.



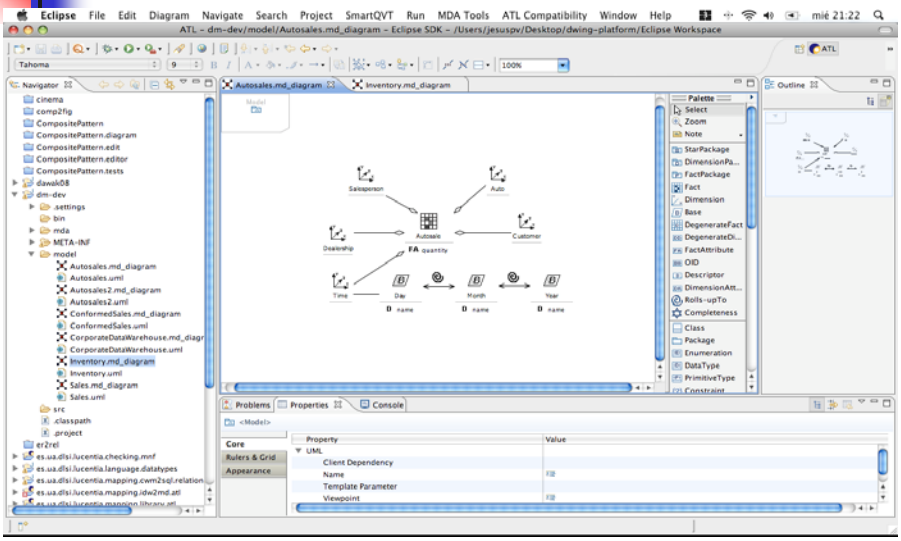
CASE Tools

Plugg-in in the Eclipse Platform.



CASE Tools

Plugg-in in the Eclipse Platform.



Outline

- Introduction
- Overall Method based on the UML
- MDA: Model Driven Architecture
- Data Warehouse Quality
- Other issues: Security and Data Mining
- CASE tools
- Conclusions and further works

Outline

- Introduction
- Overall Method based on the UML
- MDA: Model Driven Architecture
- Data Warehouse Quality
- Other issues: Security and Data Mining
- CASE tools
- Conclusions and further works

Conclusions

- An overall method based on the UML and UP for modeling various aspects of DWs
  - Requirement analysis
  - Profiles
    - MD modeling of the DW repository
      - Stereotypes, constraints and tagged values
    - All relevant properties at the conceptual level
    - Object Constraint Language (OCL)
      - Avoid arbotrary use of the new elements
  - Conceptual design of physical aspects

Conclusions

- Some Scientific production:
  - DOLAP'05, Rebnita'05, RIGIM'07 → Requirements
  - ER'02, UML'02, JISBD'02, ER'04 – ER'07
    - IEEE Computer 2001, JDM'04, JDM'06, DKE'06
    - JDM'04, JDM'06, JCIS'07, etc → MD profile
  - ER'03 (Trujillo, Luján-Mora), → ETL
  - ER'04 (Luján-Mora, Trujillo, Vassiliadis) → ETL
  - DOLAP'04, JDM'06 (Luján-Mora, Trujillo) → UP, MD
  - In total → more than 90 (see DB&LP). Downlable.

Conclusions

- Quality of MD conceptual models
  - Set of metrics applied to the MD UML profile
  - To help us choose the best conceptual schema
    - Based on the understandability
  - Scientific production:
    - CAiSE'03, SAM'03, QSE'03, DAWAK'05, QUOSE'05…
      - (Serrano, Calero, Trujillo, Luján-Mora, Piattini)
      - Together with Alarcos Research Group of UCLM
      - ISOFT'07

Conclusions

- Access Control and Audit Model (ACA)
  - Integrate UML profile for MD modeling and the ACA Model
- Basic aspects of the ACA Model:
  - Classification of the information and users to control the non-authorised access by users
  - Authorisation rules
  - Audit rules
- Scientific production:
  - ER'04, JISBD'04, DSS'06, WOSIS'05, JIRP, IS'07, EJIS'07, C&SI
    - (Ferrandez-Medina, Trujillo, Villarroel, Piattini)
    - Together with Alarcos Research Group of UCLM

Conclusions

Future works

- (1) Method
  - Unified Process → Model Driven Architecture (MDA)
    - DOLAP'05, DSS'07, DKE'07
    - Set of QVT Transformations for formal transformation between models
- (2) Requirement engineering
  - Modeling goals and business context for DWs and to obtain the MD modeling
- (3) Metrics for Models for DWs (with *ALARCOS*)
  - Complete set of metrics
  - Quality Indicators to coherently group metric definitions
  - Metric treacability from conceptual to logical level
  - Metrics thresholds

Conclusions

Future works

- (4) Metrics for ETL processes
  - Applying process metrics to ETL processes
- (5) Data Mining and Data warehouses
  - Currently, integrating more DM techniques, not only association rules
    - UML profile including MD modeling and several DM techniques
- (6) Security
  - Treacability of the ACA model and the adaptation to MDA
    - Definition of QVT transformations for secure rules

Conclusions

Future works

- (7) Security in OLAP tools
  - Considering inference
- (8) Visualization and personalization for specific and advaced Data Warehouses
- (9) Geografic Data warehouses

Invited Talk. EDA'08. Toulouse.

A conceptual approach and an overall framework for the development of data warehouses

Juan Trujillo

Grupo de Investigación LUCENTIA

Dpto. Lenguajes y Sistemas Informáticos
(Language and Information Systems)
Universidad de Alicante

Departamento de Lenguajes y Sistemas Informáticos

Universitat d'Alacant
Universidad de Alicante