Open Problems in Data Warehousing: *8 Years Later...*

Stefano Rizzi

DEIS - University of Bologna

srizzi@deis.unibo.it

Summary

- Archeology
  - ✓ The early 90's
  - ✓ Back to 1995
  - ✓ Into 2k
- At present
  - ✓ Achievements
  - ✓ Hot issues
- Some insights into...
  - ✓ Project documentation
  - ✓ Evolution
- What's next?

Summary

- Archeology
  - ✓ The early 90's
  - ✓ Back to 1995
  - ✓ Into 2k
- At present
  - ✓ Achievements
  - ✓ Hot issues
- Some insights into...
  - ✓ Project documentation
  - ✓ Evolution
- What's next?

The early 90's

- Inmon coins the term "data warehousing"
- Widening interest from enterprises
- Widening interest from vendors
- Almost ignored from the academic world
- Some topics from traditional dbs:
  - ✓ integration of heterogeneous sources
  - ✓ materialized views
  - ✓ aggregation queries
  - ✓ .....

Back to 1995

- Start of the *Stanford DW Project*
  - ✓ *develop algorithms and tools for the efficient collection and integration of information from heterogeneous sources*
- Widening interest from the academic world
  - ✓ flourishing of dedicated workshops and conferences
  - ✓ gaining attention by major conferences
- Dedicated commercial tools for DWing

Back to 1995

- The CIKM'95 paper by J. Widom
  - ✓ Research problems:
    - change detection (incremental refresh)
    - view maintenance
    - data scrubbing (ETL)
    - optimization
    - design
    - evolution

Into 2k

- End of the *DWQ European Project*
  - ✓ *study of quality-of-service factors and their relationship with design, operation, and evolution of DWing systems*
- On-going interest from research
- A large selection of commercial tools and platforms available
- Many mature implementations of DWs in enterprises

Into 2k

- The DMDW'00 paper by Vassiliadis
  - ✓ a significant gap between researchers and practitioners
    - researchers overlook practical problems
    - little acceptance of research results by the industrial world
  - ✓ increasing market for DWing systems
  - ✓ about 20 papers per year in VLDB, PODS, SIGMOD
    - mainly on query processing, view technology, integration
  - ✓ problems and failures:
    - no "textbook" design methodology
    - no standards for metadata
    - no solutions for ETL
    - no approach for view size estimation

Summary

- Archeology
  - ✓ The early 90's
  - ✓ Back to 1995
  - ✓ Into 2k
- At present
  - ✓ Achievements
  - ✓ Hot issues
- Some insights into...
  - ✓ Project documentation
  - ✓ Evolution
- What's next?

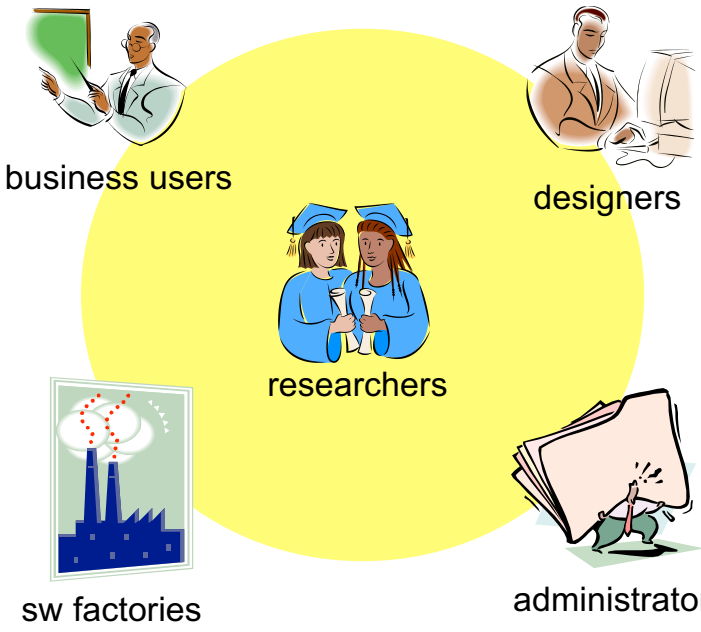
Achievements in research

	Tool Implementation	User satisfaction
architectures	😐	☺
conceptual modeling	☹	☹
OLAP	☺	☺
query lang. and processing	😐	😐
optimization and tuning	☹	😐
physical aspects, indexing	☺	☺

DOLAP 2003 submissions

- Hot topics:
  - ✓ Queries: language, optimization, processing
  - ✓ Consistency and quality
  - ✓ XML
  - ✓ ETL
  - ✓ Optimization and tuning
- Good impact:
  - ✓ Evolution
  - ✓ OLAP
  - ✓ Physical aspects
- Some interest:
  - ✓ Architectures
  - ✓ Tools and applications
  - ✓ Maintenance
  - ✓ Metadata
  - ✓ Multidimensional modeling
  - ✓ Source integration
  - ✓ View materialization

The actors



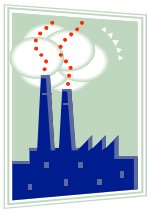
Open issues: the designer point of view

- Source integration
  - ✓ need for a standard, usable methodology
- Design
  - ✓ need for a standard, usable methodology
- Project documentation
  - ✓ conceptual schemas of facts are not sufficient, a multi-level and integrated approach is needed

Open issues: the user point of view

- Project documentation
  - ✓ high-level descriptions are needed for better understanding the informative assets
- Metadata
  - ✓ need for a standard to be used for interoperability in federated architectures
- Data quality
  - ✓ need for an approach to clearly assess data quality
- Evolution
  - ✓ in order to keep the DW in sync with the evolving business requirements

Open issues: the "tool" point of view



- Source integration
  - ✓ though a huge research has been carried out in the field, only prototypes were delivered so far
- Design
  - ✓ need for comprehensive and reliable CASE tools
- Project documentation
  - ✓ 360-degrees documentation should be supported
- Optimization
  - ✓ effective and efficient approaches to logical+physical optimization should be implemented
- Metadata
  - ✓ capability of exporting/importing from open repositories
- Evolution
  - ✓ tool-support of an approach to versioning and evolution is highly required

Open issues: the administrator point of view

- Source integration
  - ✓ in order to handle changes in the information sources
- Project documentation
  - ✓ in order to easily perform adaptive maintenance
- Evolution
  - ✓ in order to avoid premature obsolescence of the DW
- Data quality
  - ✓ in order to keep the reliability of information closely monitored

Summary

- Archeology
  - ✓ The early 90's
  - ✓ Back to 1995
  - ✓ Into 2k
- At present
  - ✓ Achievements
  - ✓ Hot issues
- Some insights into...
  - ✓ Project documentation
  - ✓ Evolution
- What's next?

Modeling and documentation

- Using a comprehensive documentation including a wide and coherent array of artifacts is highly necessary, especially for huge and complex projects
- Several conceptual models were proposed in the literature:
  - ✓ to statically model facts as multidimensional objects
  - ✓ to functionally/dinamically model the ETL process
  - ✓ to functionally model use cases for the DWing process

But that's not all!

Requirements

- DW documentation should:
  - exhaustively cover all phases of design, and yield a full picture of the process leading from initial requirements to the DW
  - be readable at multiple abstraction levels: on one extreme return a summary view of the main design activities and of the DW architecture; on the other include all useful details and the crucial choices made to determine the DW
  - act as an effective support for maintaining and extending the DW
  - allow, even to a new team of designers, to understand the design solutions previously undertaken
  - include both technical artifacts oriented to designers/ implementers and conceptual artifacts oriented to business users, the latter to be used for discussing, verifying and refining specifications
  - include glossaries aimed at getting non-experts acquainted with the terminology of the application domain

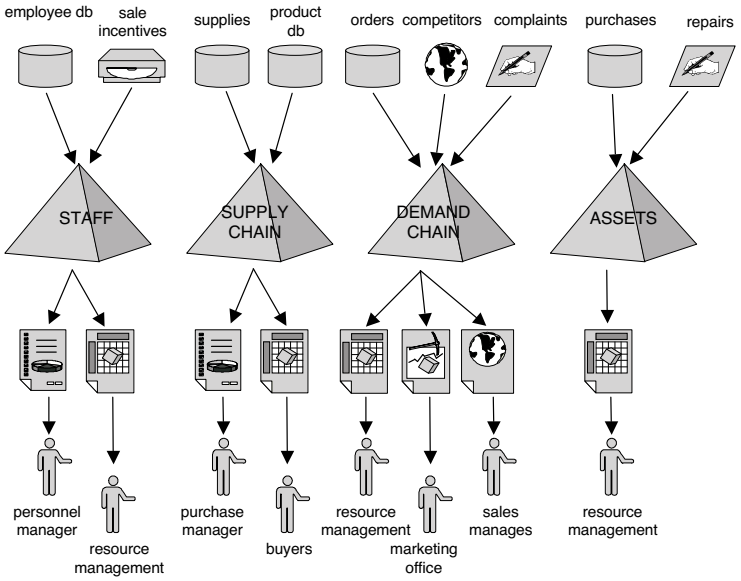
Multi-level organization

- We envision three different levels of abstraction for documentation, each including several diagrams or schemas and integrated by glossaries

1. Data warehouse level. It describes the overall architecture of the DW, emphasizing the user profiles and the data sources
2. Data mart level. It summarizes the structure of each data mart by documenting their logical and physical schemes, their workloads, their feeding processes
3. Fact level. It details each cube at the conceptual level, also in terms of data volumes

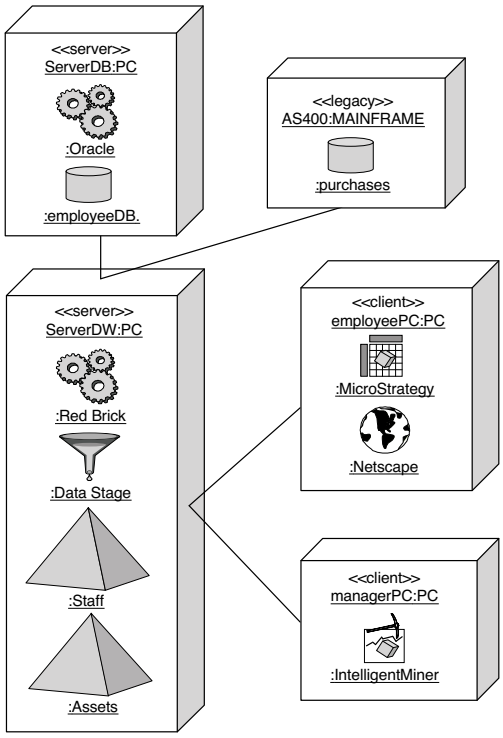
The data warehouse level

1) Data warehouse schema



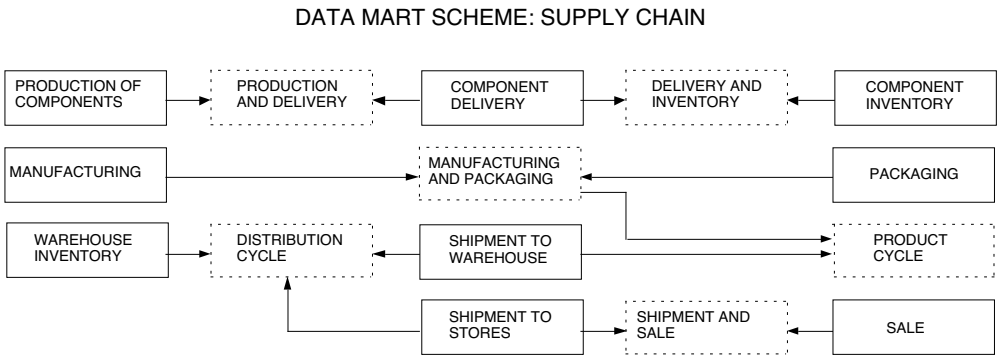
The data warehouse level

2) Deployment schema



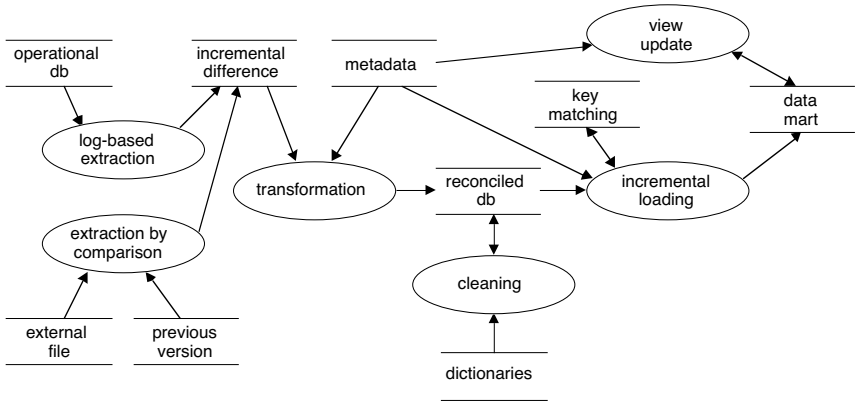
The data mart level

1) Data mart schemas



The data mart level

2) Feeding schemas



The data mart level

3) Operational schemas
- ✓ The conceptual/logical/physical documentation available for the source and the reconciled schemas

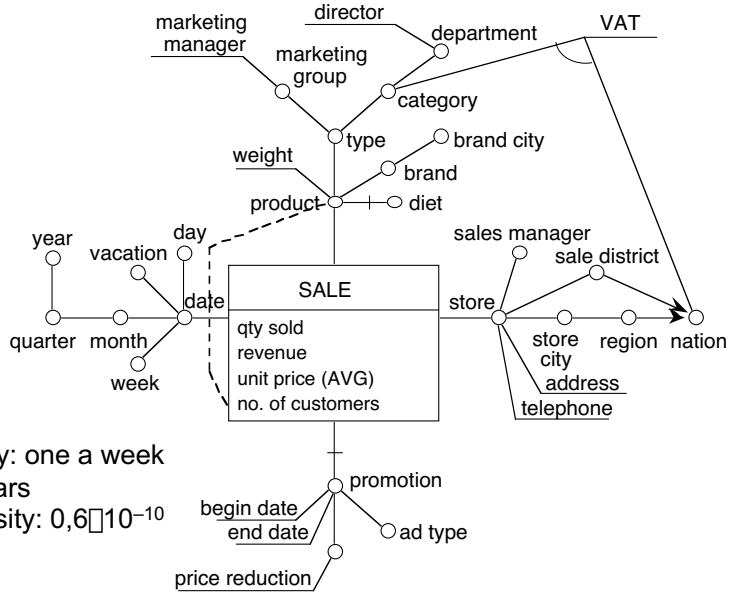
4) Glossary of domains
- ✓ A description of all the domains involved in the data mart (e.g. products, cities). Useful for drill-across

5) Workload
- ✓ A graphical or symbolic specification for the core queries (e.g. periodical reports)

6) Logical and physical schemas
- ✓ The logical and physical schemas of the data mart (star schemas, indexes, allocation)

The fact level

1) Fact schemas



refresh frequency: one a week
time range: 5 years
percentage sparsity: $0,6 \cdot 10^{-10}$

The fact level

2) Attribute and measure glossaries

ATTRIBUTE GLOSSARY: SHIPMENT TO STORES

name	description	domain	card.	query
product		products	5000	`select prodName,brandName, cityName,… from PRODUCTS P,BRANDS B, CITIES C,… where P.brandId = B.brandId and B.cityId = C.cityId and . . . . . .`
brand		brands	800	
brand city	Where brands are manufactured	cities	50	
type	(pasta, soft drink, …)	pr. types	200	
category	(food, clothing, music,…)	pr. categories	10	
department	Deps. managing categories	deps.	5	
marketing group	Responsible for product types	groups	20	
stores		stores	100	`select storeName,cityName, stateName from STORES S,CITIES C where S.cityId = C.cityId`
store city		cities	80	
store state		states	5	
.....	.....	.....	.....	. . . . .

MEASURE GLOSSARY: SHIPMENT TO STORES

name	description	type	query
qty shipped	Quantity of each product being shipped	INTEGER	`select SUM(PS.qty) from PRODUCTS P,SHIP S,PRODSHIP PS,… where P.prodId = PS.prodId and PS.shipId = S.shipId and . . . . . group by P.prodId,S.date, . . .`
shipping cost	Cost of the shipment	MONEY	. . . . .

Evolution

- As several mature implementations of DWing systems are fully operational, the continuous evolution of the application domains brings to the forefront the dynamic aspects related to describing how the information stored changes over time:
  - ✓ At the extensional level
  - ✓ At the intensional level
- Temporal issues are pressing in DWs since queries frequently span long periods of time; thus, it is very common that they are required to cross the boundaries of different versions of data and/or schema
- The problem is highly critical for DWs that have been established for a long time, since unhandled evolutions will determine a stronger gap between the reality and its representation, that will soon become obsolete and useless

Evolution

- A crucial role in preserving the up-to-dateness of DWs is played by the ability to manage the changes that the DW schema undergoes over time in response to the evolving business requirements
- Schema versioning in DWs has only partially been explored and no commercial tools or restructuring methodologies are available to the designer

Schema versioning vs. evolution

- *Schema versioning*: past schema definitions are retained so that all data may be accessed both retrospectively and prospectively through user-definable version interfaces
- *Schema evolution*: allows modifications of the schema without loss of data but does not require the maintenance of a schema history

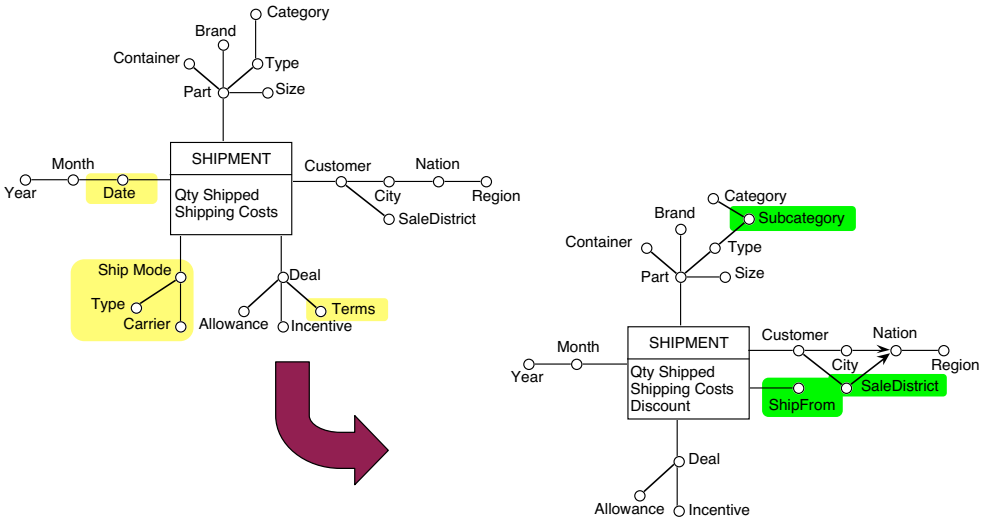
- In most approaches in the literature, versioning is not supported and the problem of querying multiple schema versions is not mentioned
- In the COMET approach to schema evolution [Eder et al. 2002] the problem of queries spanning multiple schema versions is mentioned, but the different temporal scenarios are not considered

An approach to schema versioning in DWs

A joint work with Matteo Golfarelli (Univ. of Bologna)
Jens Lecthenbörger (Univ. of Münster)
Gottfried Vossen (Univ. of Münster)

- Augmented schemata are introduced as a support for increasing flexibility in formulating *cross-version queries*
- An algebra of operators to handle changes in a DW schema is defined, and the sequencing of operators to form *schema histories* is considered
- The relationship between the *temporal horizon* spanned by a query and the schema on which it can consistently be formulated is analyzed

Example

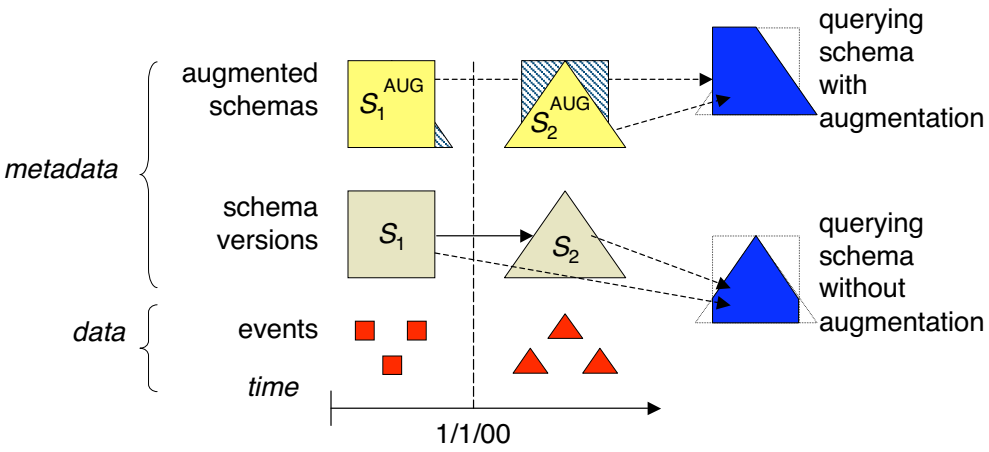


- What is the distribution of the shipping costs for 2002 according to subcategories, introduced in 2003?

The idea

- Cross-version queries can only be answered by undertaking some *actions* on past data. These actions may entail checking data for additional constraints or inserting new data
  - ✓ By deciding which actions to undertake, the designer trades querying flexibility for complexity of keeping the database up-to-date
- The set of actions undertaken by the designer leads to associating each schema version S with an augmented schema S^{AUG}, that is *the most general schema satisfied by data associated to S*
  - ✓ S^{AUG} will be used, instead of S, to determine if a given query q spanning the validity interval of S is correct

Augmented schemas



Examples

- Operation
  - ✓ Add a new attribute
  - ✓ Add a new measure
  - ✓ Add a new dimension
  - ✓ Add a new functional dependency
- Action on past data
  - ✓ Input of attribute values
  - ✓ Function-based computation
  - ✓ Disaggregation of measures by fractions
  - ✓ Verify functional dependency

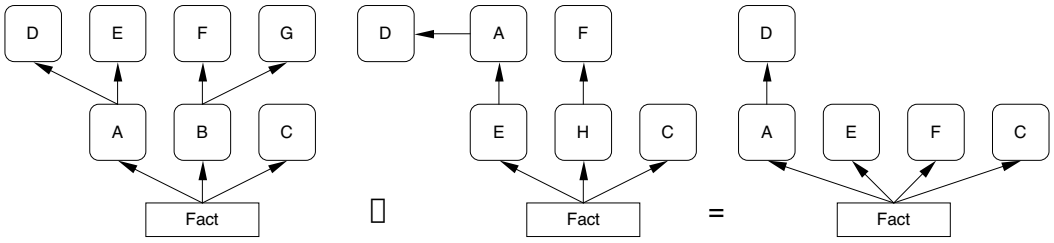
Cross-version querying

- OLAP operators navigate the FDs expressed by the hierarchies in the multidimensional schema
- Specifying the schema version for query formulation means
  - ✓ declaring which attributes are available for formulating next query q'
  - ✓ representing the FDs that relate them in order to determine how q' can be obtained from the previous query q

Cross-version querying

- The *formulation context* for an OLAP query is well represented by a directed graph of attributes and FDs
  - ✓ If the OLAP session spans a single version V, the graph is the one describing V
  - ✓ When multiple versions are involved, a schema under which *all* data involved can be uniformly queried must be determined.
    - In our approach, such schema is univocally determined by the temporal interval T covered by the data to be analyzed, as the largest schema that retains its validity throughout T
    - Since T may span different schema versions, we define an *intersection* operator for determining the common schema between two different schema versions

Example of intersection

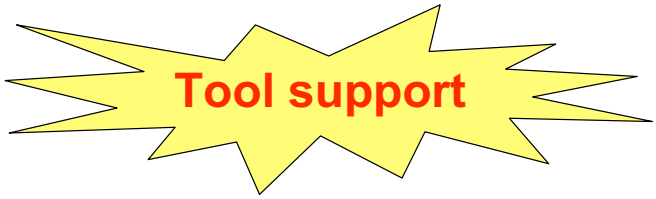


Summary

- Archeology
  - ✓ The early 90's
  - ✓ Back to 1995
  - ✓ Into 2k
- At present
  - ✓ Achievements
  - ✓ Hot issues
- Some insights into...
  - ✓ Project documentation
  - ✓ Evolution
- What's next?

Some promising trends

- Architecture:
  - ✓ Federations of XML warehouses under P2P environments
- Optimization and processing:
  - ✓ for complex nested aggregation queries
- Maintenance
  - ✓ in highly dynamical environments



Bibliography

- Abiteboul. Managing an XML warehouse in a P2Pcontext. CAiSE, 2003.
- Chaudhuri, Dayal. An overview of data warehousing and OLAP technology. ACM SIGMOD Record, 26(1), 1997.
- Eder, Koncilia, Morzy. The COMET Metamodel for temporal data warehouses. CAiSE, 2002.
- Golfarelli, Rizzi. Data Warehouse: teoria e pratica della progettazione. McGraw-Hill Italia, 2002.
- Vassiliadis. Gulliver in the land of data warehousing: practical experiences and observations of a researcher. DMDW, 2000.
- Widom. Research problems in Data Warehousing. CIKM, 1995
- Wu, Buchmann. Research issues in Data Warehousing. BTW, 1997.