Data Quality Management and Dimensional Modeling : Issues and Solutions

Rome University Roma3 – 26 May 2005

Angelo Sironi
angelo_sironi@it.ibm.com

Contents

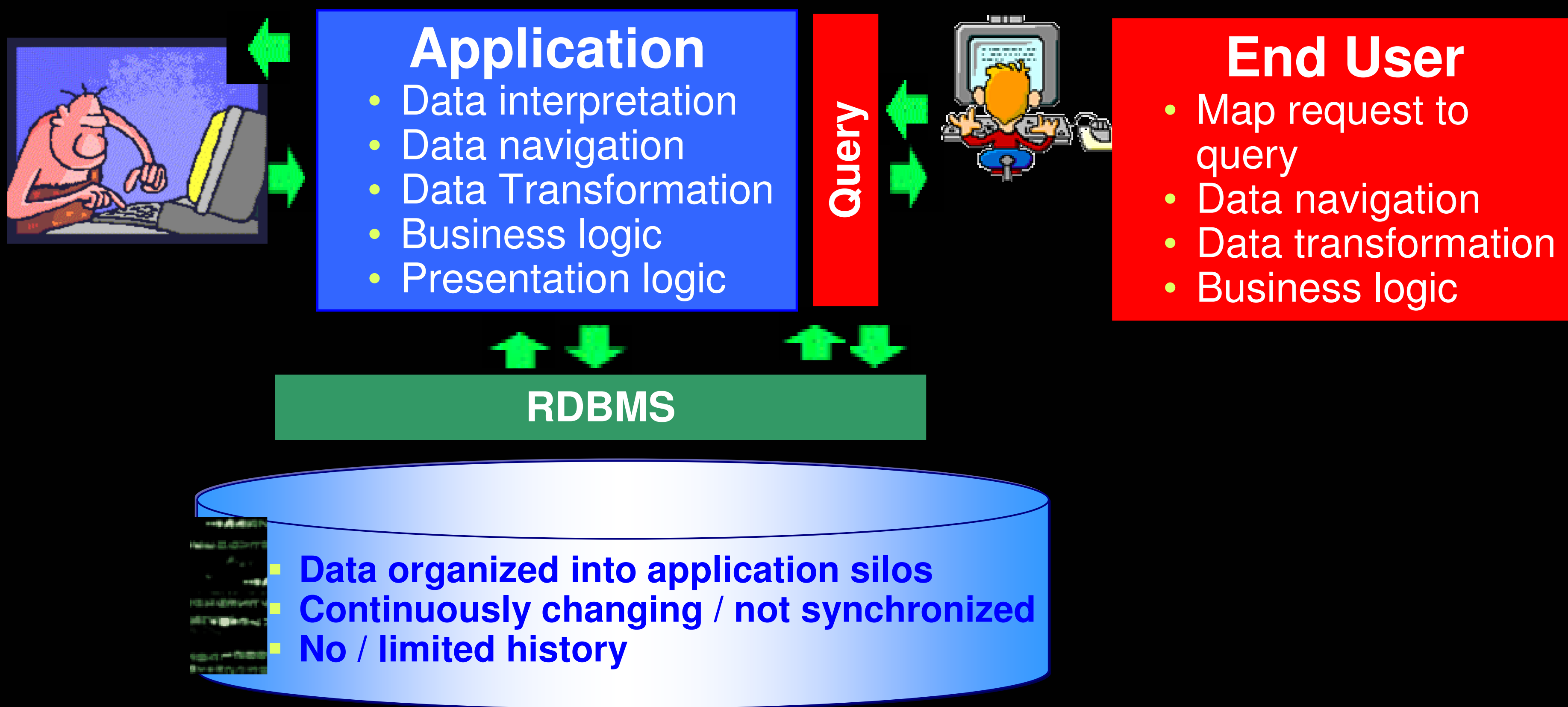
- Data Warehouse goals and architecture
- Dimensional Model
  - Characteristics
  - Surrogate keys
- Bi-temporal Model
  - What and why
  - Issues
- Data Quality
  - What, How
- Wrapping all together

Data Warehouse Solutions: goals

- Support Decision Makers
- Help finding answers to
  - Questions they know how to express
  - Questions they are not sure how to express
  - Finding the unknown
- by means of a set of specialized tools
  - But, usually, not “ad hoc applications”
- Queries usually span back in the past (5+ years)
- Users may be very or little skilled in IT
  - From high to minimal autonomy

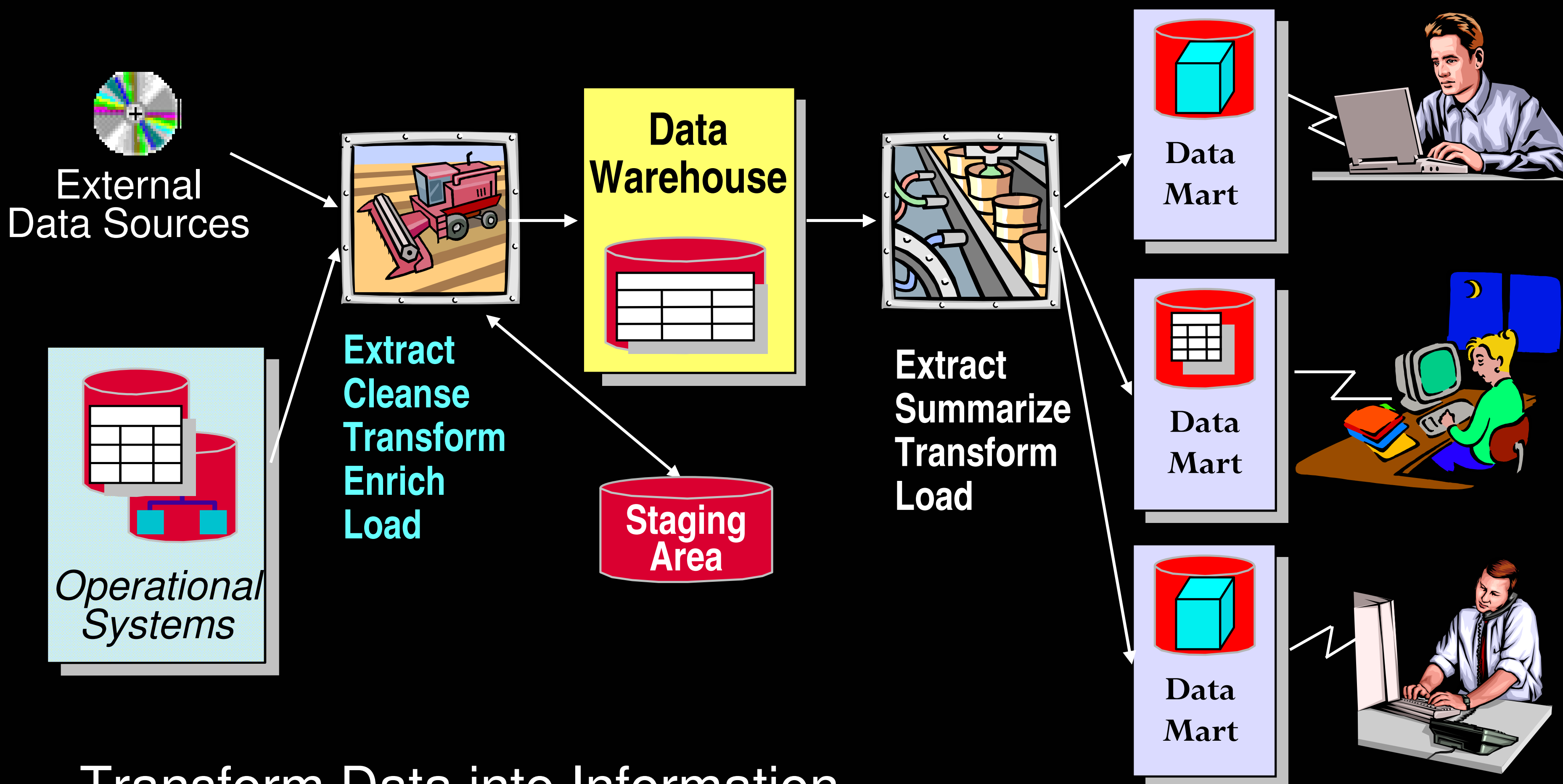
a. sironi

Data Access and Manipulation: OLTP vs. User Query



a. sironi

A Data Warehouse Architecture



Transform Data into Information

Transform Information into Knowledge

a. sironi

Which Data Model for Data Warehousing?

- "ER (Entity Relationship) Modeling is a showstopper
  - End users cannot understand or remember an ER model
  - Software cannot usefully query a general ER model
    - Cost-based optimizers that attempt to do this are notorious for making the wrong choices, with disastrous consequences for performance
- Use of ER modeling techniques defeats the basic allure of data warehousing, namely
  - Intuitive and high performance retrieval of data
- DM (Dimensional Modeling) is a logical design technique
  - Seeks to present data in a standard, intuitive framework
  - Allows for high-performance access"

Ralph Kimball, "A Dimensional Modeling Manifesto", http://www.dbmsmag.com/9708d15.html
Fabian Pascal, "On Data Warehouses", http://www.dbdebunk.com/page/page/622521.htm

The Dimensional Model

- Term coined by Ralph Kimball (www.rkimball.com)
  - Red Brick Systems founder
- Describes a “star-like” structure composed of
  - A Fact table at the center, surrounded by …
  - A number of smaller tables, called Dimension tables
  - The Fact table represents a many-to-many-to…-many relationship
- Dimension table are highly denormalized
  - Their hierarchies are flattened
- A very special dimension
  - The Time dimension

Data Models and Data Modeling: An Aside

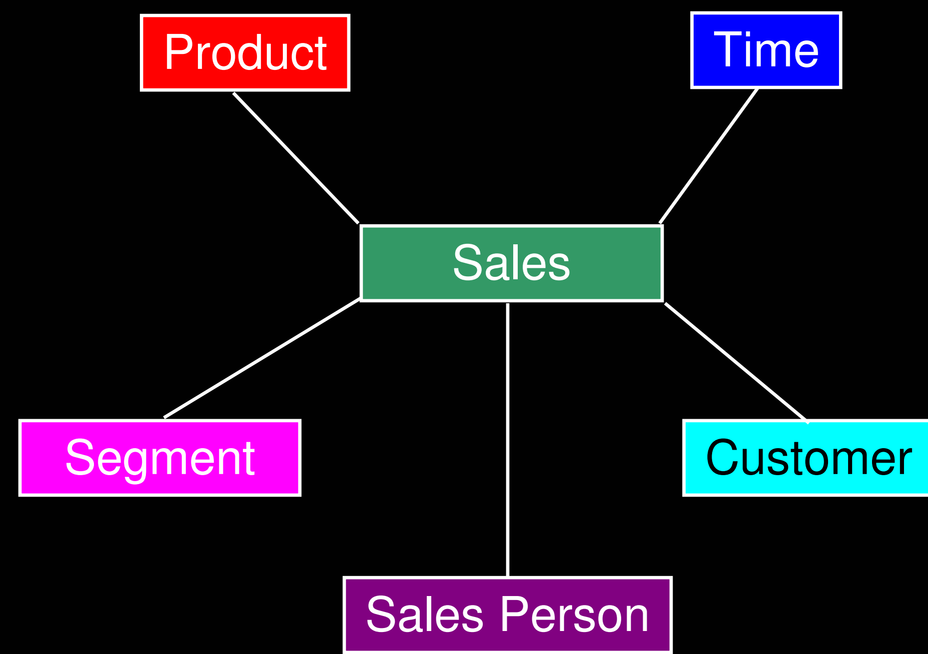
"I would never claim to be "someone who really understands [data] modeling," as you put it. Personally, I find the discipline of data modeling to be still far too lacking in solid scientific principles. As a consequence of this fact, I don't think *anyone* can claim to "really" understand it. The principles of normalization are scientific, of course, and there are a few other similarly solid principles, but all in all (as we all know) the principles in question address only a tiny part of the total database design problem. The rest is subjective! (I sometimes refer to normalization and the other related principles as "the one small piece of science in this otherwise artistic endeavor.")

Chris Date, "**BUSINESS RULES AND INTEGRITY CONSTRAINTS: A REPLY TO RALPH KIMBALL**", http://www.dbdebunk.citymax.com/page/page/622790.htm

a. sironi

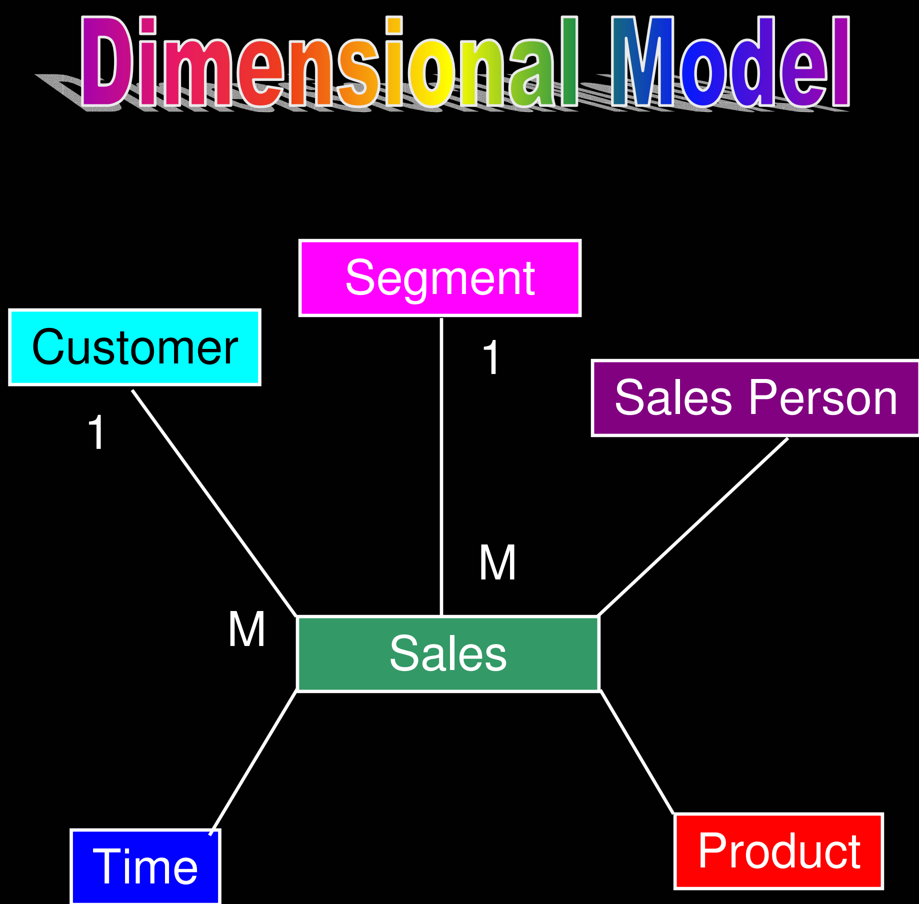
End User Logical View

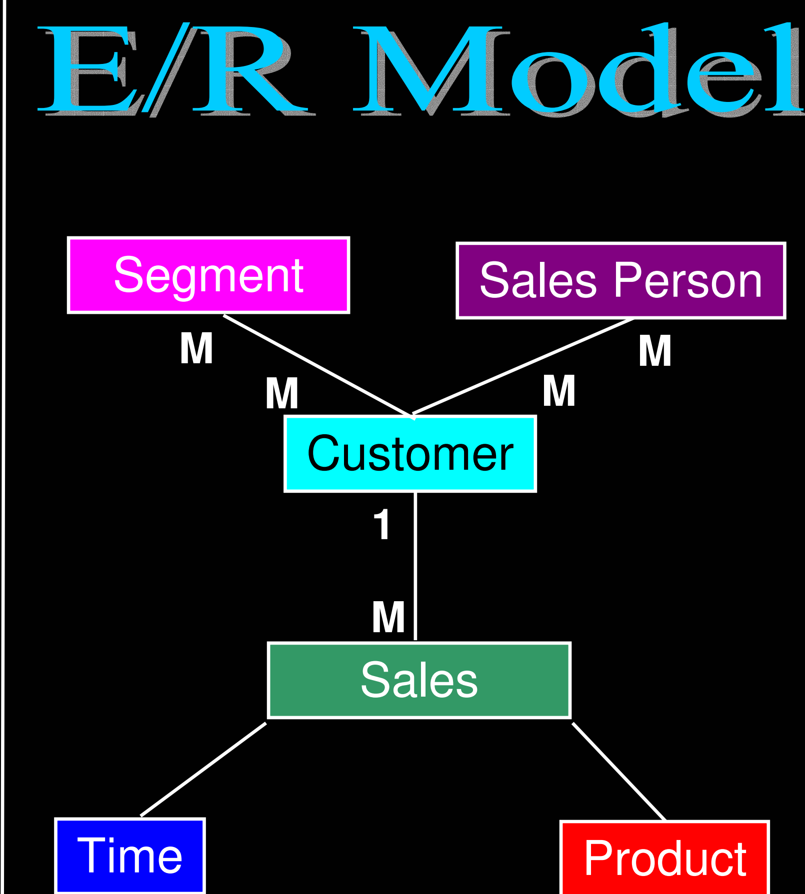
- Show best-selling products by sales person and customer segments
- Show revenue for top customers by product type and sales person over the last four weeks
- Etc.



The End-Users view is the starting point when developing a DW Dimensional Model

Dimensional Model vs. “E/R Model”

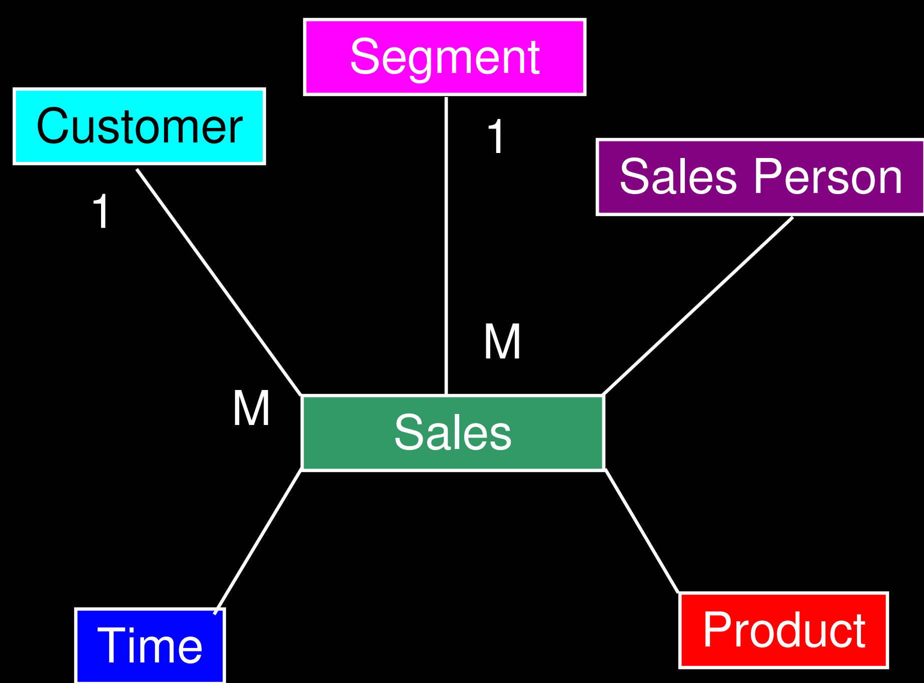




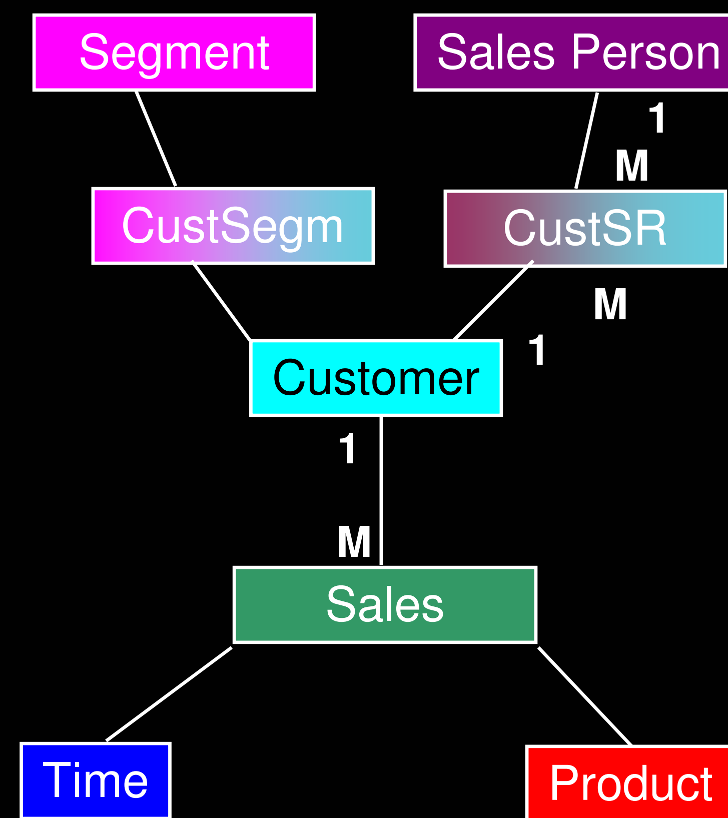
a. sironi

Dimensional Model vs. “E/R Model”: In Reality …

Dimensional Model



E/R Model



a. sironi

The Dimensional Model: Surrogate Keys

- A surrogate key is an artificial or synthetic key that is used as a substitute for a natural key[1]. Even better …
- … a surrogate key is a necessary generalization of the natural production key and is one of the basic elements of data warehouse design[1]
  - Anonymous integers
- Why Surrogate Keys
  - Production keys reuse
  - Production key re-definition
  - Merge & Acquisition
  - Support for Slowly Changing Dimension

[1] R. Kimball, Surrogate Keys, May 1998, www.dbmsmag.com/9805d05.html

Slowly Changing Dimensions

- Recording changes to dimension records
  1. Overwrite
  2. Creating another dimension record
  3. Creating a current value field

PSuKey	SKU	SkuDesc	Price	Cat	CatDesc	Valid_From	Valid_To
1	122	**Pirate Rum**	100	K1	Spirits	01.01.2001	12.06.2004
2	643	**Baking Powder**	12	K3	Baker's Ware	01.01.2002	31.12.9999
3	352	…	…	…	…		
4	276	…	…	…			
…	…						
19	122	**Pirate Rum**	103	K1	Spirits	12.06.2004	31.12.9999

a. sironi

Creating another dimension record

This technique for tracking slowly changing dimensions is very powerful because new dimension records automatically partition history in the fact table.

- The old version of the dimension record points to all history in the fact table prior to the change.
- The new version of the dimension record points to all history after the change.

There is no need for a timestamp in the product table to record the change. In fact, a timestamp in the dimension record may be meaningless because the event of interest is t he actual use of the new product type in a shipment.

Ralph Kimball, “Slowly Changing Dimensions”, DBMS, April 1996, http://www.dbmsmag.com/9604d05.html

a. sironi

Facts & Dimensions: Linking Them Together

PSuKey	SKU	SkuDesc	Price	Cat	CatDesc	Valid_From	Valid_To
1	122	**Pirate Rum**	100	K1	Spirits	01.01.2001	12.06.2004
4	276	…	…	…			
…	…						
19	122	Pirate Rum	103	K1	Spirits	12.06.2004	31.12.9999

PSuKey	SKU	CustSuKey	Sales_Date	SpSuKey	Qty
1	122	1263	11.06.2004	12	12
1	122	194	11.06.2004	21	6
4	276	1263	11.06.2004	12	1
19	122	1263	12.06.2004	15	6
…		…	…	…	…

Sales Record:

SKU	CustNo	SalesPers	Date	Qty
122	C9856	SR24	12.06.2004	6

a. sironi

Querying with & without using Surrogate Keys

- Using Surrogate Keys

```
SELECT *
  FROM PRODUCTS P
     , SALES    S
WHERE P.PSUKEY = S.PSUKEY
  AND S.SALES_DATE > '31.05.2004'
```

- Not Using Surrogate Keys

```
SELECT *
  FROM PRODUCTS P
     , SALES    S
WHERE P.SKU = S.SKU
  AND S.SALES_DATE >= P.VALID_FROM
  AND S.SALES_DATE  < P.VALID_TO
  AND S.SALES_DATE > '31.05.2004'
```

An Aside: On the semantics of "NOW"

An aspect of time that has been intriguing philosophers for centuries and that is difficult to describe fully is the concept of the current time, which we term now.

This concept is unique to time; indeed, there really does not exist any other notion quite like it. Among its properties, the current time is ever-increasing, all activity is trapped at the current time, and the current time separates the past from the future. The spatial equivalent, here, simply fails to enjoy the properties of now.

As Merrick Furst puts it, **"The biggest difference between time and space is that you can't reuse time."** The uniqueness of now is one of the reasons why techniques from other research areas are not readily, or not at all, applicable to temporal data; now offers new data management challenges, which are particular to temporal databases.

C.S.Jensen, *Introduction to Temporal Database Research*, www.cs.auc.dk/~csi/Thesis/

a. sironi

What if the information on the new price for "Pirate Rum" gets stored into the DB later than '12.06.2004'?

PSuKey	SKU	SkuDesc	Price	Cat	CatDesc	Valid_From	Valid_To
1	122	Pirate Rum	100	K1	Spirits	01.01.2001	12.06.2004
4	276	…	…	…			
…	…						
19	122	Pirate Rum	103	K1	Spirits	12.06.2004	31.12.9999

PSuKey	SKU	CustSuKey	Sales_Date	SpSuKey	Qty
1	122	1263	11.06.2004	12	12
1	122	194	11.06.2004	21	6
4	276	1263	11.06.2004	12	1
1	122	1263	12.06.2004	15	6
…		…	…	…	…

Input Record:

SKU	CustNo	SalesPers	Date	Qty
122	C9856	SR24	12.06.2004	6

a. sironi

Back to Consistent Results: Bi-Temporal Model

- Some issues related to using Surrogate Keys… Beside that, and even more important…
- Storing Valid Time is not enough for getting consistent results and understanding them
  - Valid Time: when the Fact becomes true in the real world
- We need also to track
  - Transaction Time(*): when the Fact is being recorded into the DB
- Updating and querying a Bi-temporal Data Model is quite complicated
- An example follows

(*) Transaction Time: Data or Meta-Data?

a. sironi

Updating/querying a Bi-Temporal DM: An Example

Key	SKU	SkuDesc	P	Valid_From	Valid_To	Trx_From	Trx_To
1	122	Pirate Rum	100	01.01.2001	31.12.9999	30.12.2000	14.06.2004
4	276	…	…				
…	…						
19	122	Pirate Rum	100	01.01.2001	12.06.2004	14.06.2004	31.12.9999
20	122	Pirate Rum	103	12.06.2004	31.12.9999	14.06.2004	31.12.9999

- Additional Complexities
  - Avoiding / Identifying duplicates
  - Querying …

```
SELECT *
  FROM PRODUCTS
 WHERE P.SKU = 122
   AND VALID_FROM <= '17.06.2004'
   AND VALID_TO    > '17.06.2004'
   AND TRX_FROM   <= '18.06.2004'
   AND TRX_TO      > '18.06.2004'
```

a. sironi

Updating Bi-Temporal DM: considerations

- Tracking a change of an existing record requires (for the simplest case)
  - One SQL update operation
  - Two SQL Insert operations
- More complex situation exist (e.g., delayed and out-of-sequence)
- Recommendation on Primary Key
  - Follow Kimball rules: i.e. Primary Key = Surrogate Key
    - No need for the user to specify additional date constraints
  - Query examples on following pages

Querying Bi-Temporal DM

- Suppose &current date = ’13.6.2004’ (&current_date is a variable)
- Query: what was the price of SKU = 122 on date = ’13.06.2004’?

```
SELECT P as PRICE
  FROM PRODUCTS P
WHERE P.SKU = 122
  AND '13.06.2004' >= P.VALID_FROM
  AND '13.06.2004'  < P.VALID_TO
  AND &current_date >= P.TRX_FROM
  AND &current_date < P.TRX_TO
```

  - Query Result - Price = 100
- Same query, &current_date = ’16.6.2004’
  - Query Result – Price = 103

a. sironi

Querying Bi-Temporal DM: Joining Facts and Dimensions

- Using Surrogate Keys (you trust ETL correctness)

```
SELECT *
  FROM PRODUCTS P
     , SALES    S
WHERE P.PSUKEY = S.PSUKEY
  AND S.SALES_DATE > '31.05.2004'
```

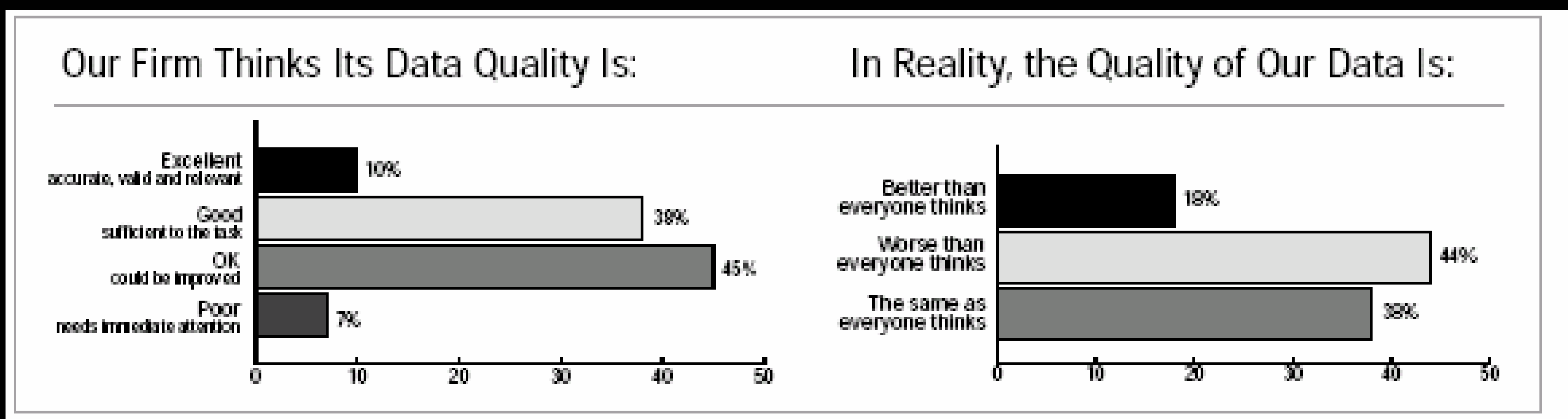
- Not Using Surrogate Keys (you don't trust ETL correctness)

```
SELECT *
  FROM PRODUCTS P
     , SALES    S
WHERE P.SKU = S.SKU
  AND S.SALES_DATE >= P.VALID_FROM
  AND S.SALES_DATE  < P.VALID_TO
  AND &current_date >= P.TRX_FROM
  AND &current_date  < P.TRX_TO
  AND S.SALES_DATE > '31.05.2004
```

a. sironi

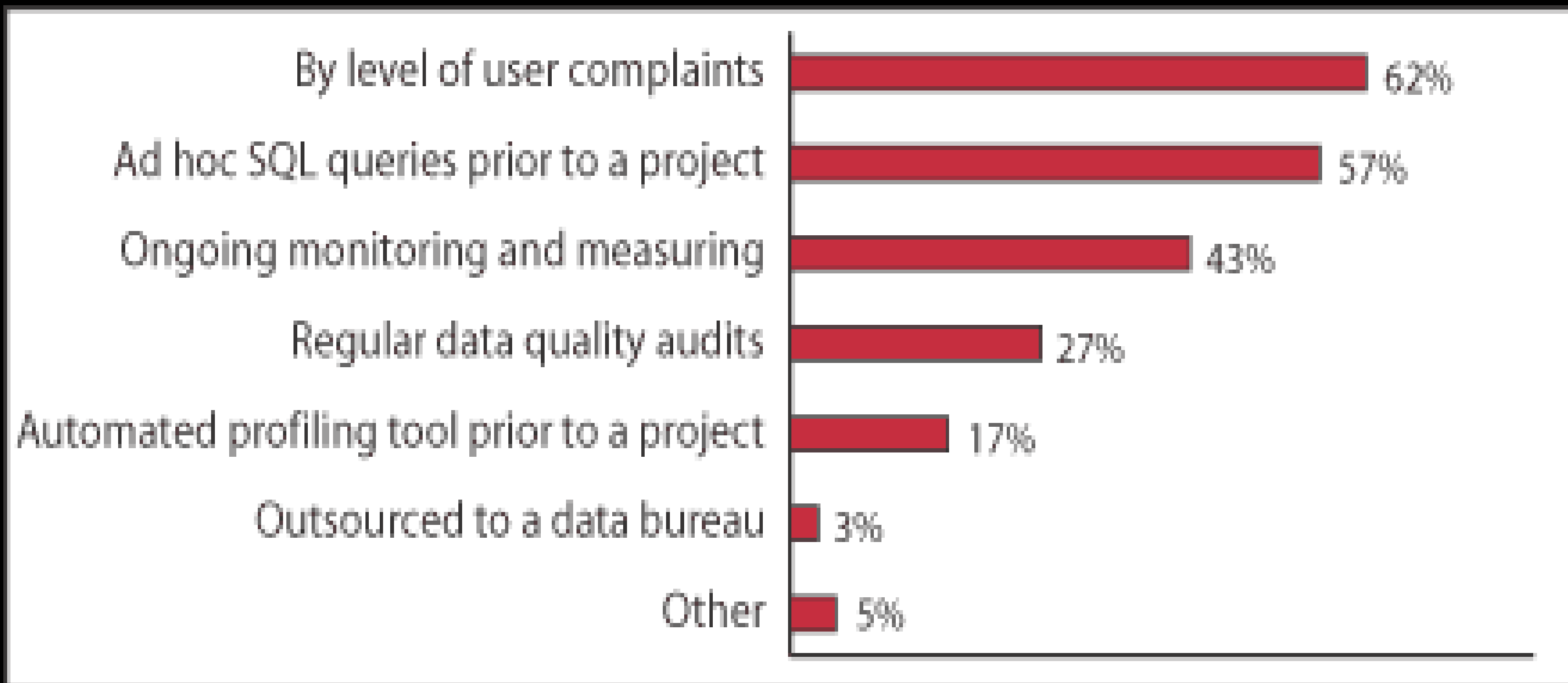
Data Quality

- The Data Warehousing Institute (TDWI) Report - 2002
  - "Poor quality customer data costs U.S. businesses a staggering $611 billion a year in postage, printing, and staff overhead"
  - "almost half of all companies have no plan for managing data quality"



a. sironi

How is the quality of data determined?



Data Quality and the Bottom Line," TDWI Report Series, p. 21, www.tdwi.org/research.

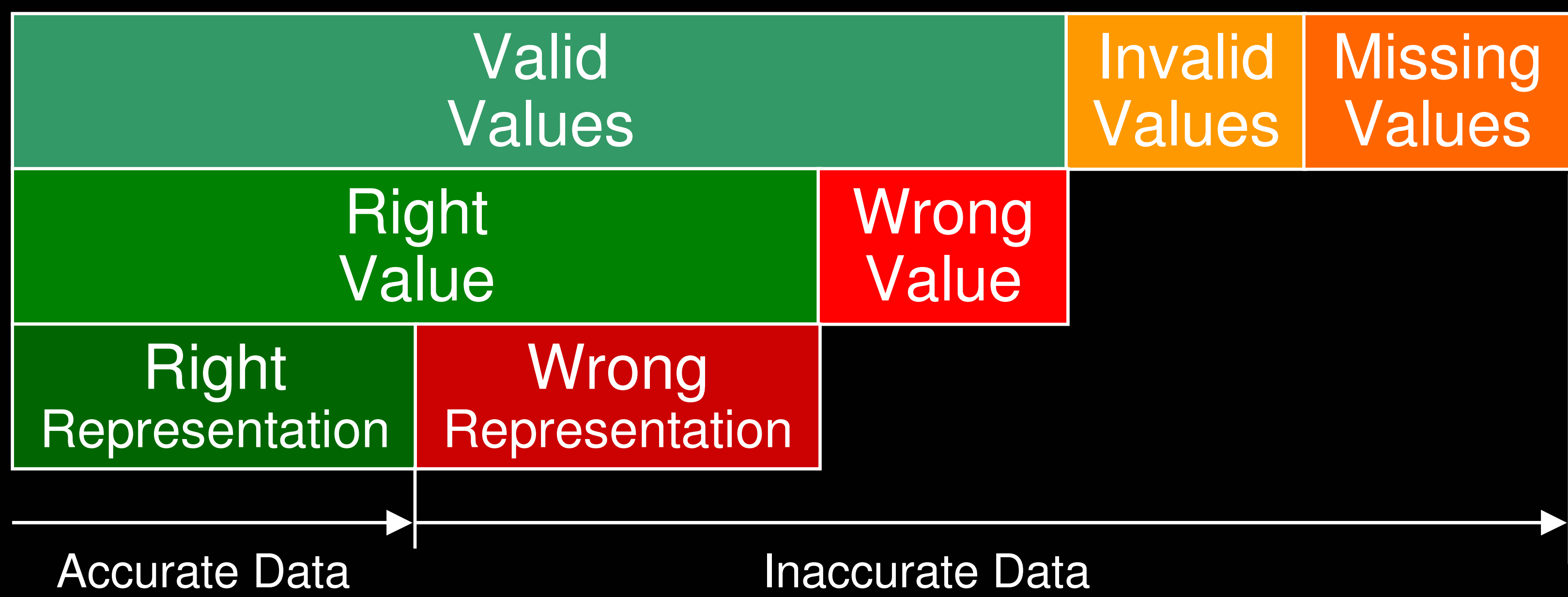
a. sironi

Data Quality Characteristics and Integrity / Business Rules

Validity	Data values pertain to the applicable domain	Domain Integrity
Completeness	Degree to which values are present in the attributes that require them	Domain Integrity
Uniqueness	Data values that are constrained to a set of distinct entries	Entity Integrity
Relatability	Agreement or logical coherence that permits rational correlation in comparison with other similar or like data	Referential Integrity
Consistency	Agreement or logical coherence among data that free them from variation or contradiction	Business Rules
Timeliness	Data item or multiple items that are provided at the time required or specified	Business Rules
Accuracy	Degree of agreement between a set of data values and a corresponding set of correct values	All of the above

a. sironi

Accuracy Defined



Jack E. Olsen, *Data Quality: The Accuracy Dimension*, Morgan Kaufmann Publishers, 2003

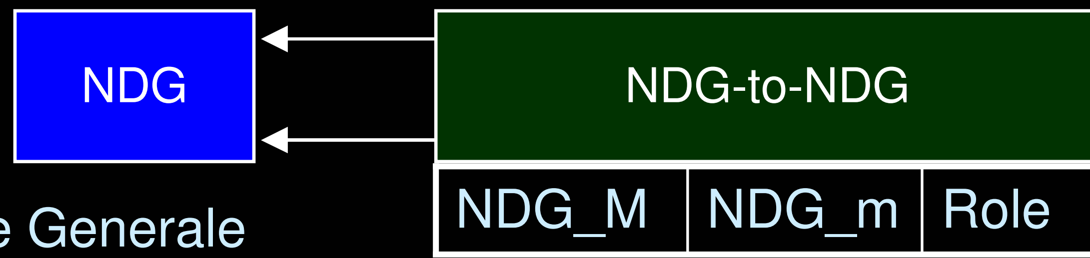
a. sironi

Assessing Data Quality

- Data quality assessment is based on Metadata
  - Integrity Rules
    - Domain Integrity
    - Entity Integrity
    - Referential Integrity
    - Constraint Integrity (more general Business Rules)
- Metadata
  - Are usually hidden in application program
  - Are very often incomplete
- The most important technology for assessing data accuracy is Data Profiling

a. sironi

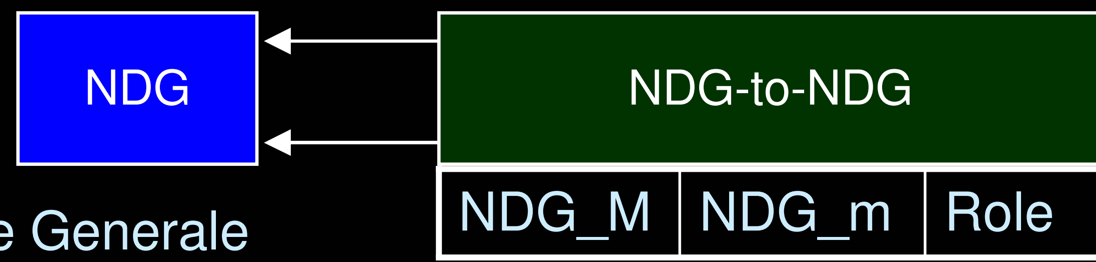
Rules: A Bank Example



- NDG – Numero di Direzione Generale
  - Identifies a Client (a Person or a Company) or a Group of Clients (“*Cointestazione*”)
- “Business” Rules for NDG-to-NDG
  - Role = ‘U’ <u>IFF</u> NDG does <u>not</u> identify a Group of Clients
  - If NDG1 identifies a Group of Clients, then
    - there must exist <u>one & only one</u> “Primary” Client in the Group
    - there must exist <u>at least one</u> “Secondary” in the Group
  - If NDG1 identifies a single Client, the NDG1 cannot identify also a Group of Clients
  - If NDG1 identifies a Group of Clients, then NDG1 cannot identify also a single Client
  - ...
- Three columns, five rules … or even more …!!!

a. sironi

Rules: A Bank Example



- NDG – Numero di Direzione Generale
  - Identifies a Client (a Person or a Company) or a Group of Clients (Cointestazione)
- "Business" Rules for NDG-to-NDG
  - Role = 'U' <u>IFF</u> NDG_M = NDG_m
  - For Each (NDG1, NDG2) for which Exists (NDG1, NDG2, x) where x in ('P', 'S')
    - there must exist <u>one & only one</u> row with x = 'P'
    - there must exist <u>at least one row</u> with x = 'S'
  - If Exists NDG1 and a row (NDG1, NDG1, U) then there must not exists a row (NDG1, NDG2, x) where NDG2 <> NDG1 <u>and</u> x <> 'U'
  - If Exists NDG1 and a row (NDG1, NDG2, x) where NDG1 <> NDG2 and x in ('P', 'S'), then there must not exist a row (NDG1, NDG1, U)
  - For Each (NDG1, NDG2) for which Exists (NDG1, NDG2, x) where x in ('P', 'S'), then NDG1 <> NDG2
- Three columns, five rules … or even more …!!!

a. sironi

Rules mapping to SQL - 1

Inconsistent definitions

```
SELECT NDG_M
     , ndg_m
     , Role
     , `Inconsistent Definition' as ERROR
  FROM COINTEST
 WHERE ((Role = 'U`             and
        NDG_M <> ndg_m)
    or ( Role in ('P', 'S`)   and
         NDG_M =  ndg_m ))
   and current date between valid_date_begin and
                             valid_date_end
order by 1,2,3
```

a. sironi

Rules mapping to SQL - 2

Duplicated Roles

```
SELECT NDG_M
     , max(case when Role = 'U'
                then 1 else 0 end) as Unique
     , max(case when Role = 'P' or
                     Role = 'S'
                then 1 else 0 end) as Group
FROM COINTEST
where current date between valid_date_begin and
                           valid_date_end
group by NDG_M
having max(case when Role = 'U'
                then 1 else 0 end) > 0
   and max(case when Role = 'P' or
                     Role = 'S'
                then 1 else 0 end) > 0
with ur;
```

Rule Assessment

- Some rules only require inspecting the value of a column within each record, individually
  - E.g. Domain Integrity Rules
- Some rules require comparing the value of a column in a record with the value of one or more other columns on the same record
  - E.g. If Role = ‘U’ then NDG = NDG-Comp
- Some rules are set-oriented
  - Some are simple: e.g. Entity Integrity rules
  - Some are more complex to handle, especially with “delta update”
    - Example: Change of role (‘P’ -> ‘S’) in a group

Back to Data Warehousing: Managing Data Quality

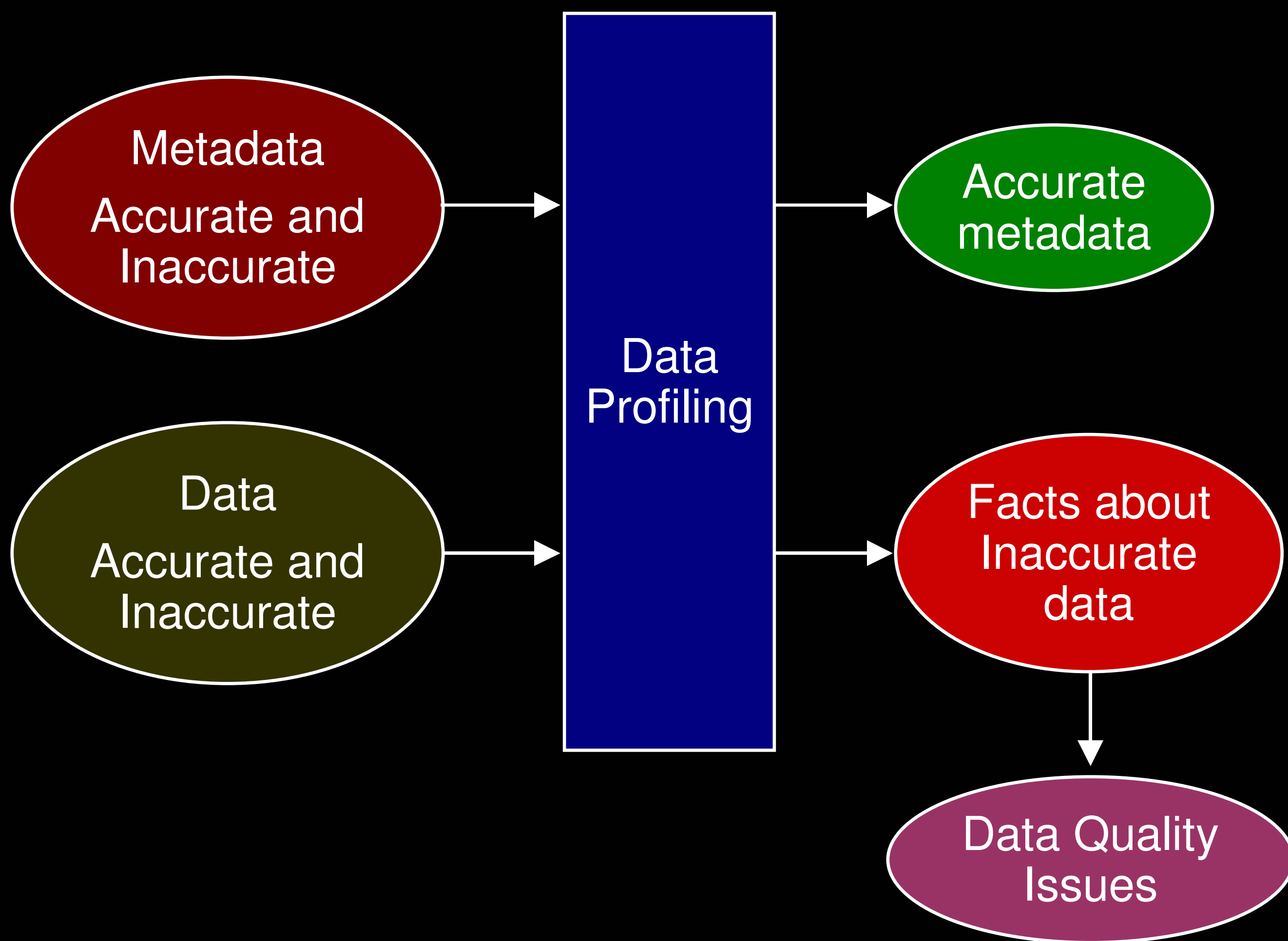
- Data Quality Management is usually considered part of ETL process
  - Data Cleansing
- Data Cleansing based on Rules
  - Discovering the Rules
  - Applying the Rules
    - When
    - How

Discovering the Rules: Data Profiling

- Data profiling is defined as the use of analytical techniques to discover the true structure, content, and quality of a collection of data.
- The output of data profiling is accurate metadata plus additional information on content and quality of the data.
- Data profiling is a process that involves learning from the data. It employs discovery and analytical techniques to find characteristics of the data that can then be looked at by a business analyst to determine if the data matches the business intent.
- But … NO Miracles… Usually, quality checks only about
  - Domain Integrity
  - Entity Integrity
  - Referential Integrity
- The need exists for custom declarative rules

a. sironi

Data Profiling Model



a. sironi

Applying the Rules: When?

- Extract
  - Often out of control of the DW people
  - May not be possible, because of impacts on legacy systems
- Transform
  - Probably the best place
  - Allows deciding what to do before loading
  - But requires anticipating all checks, including the ones usually done during load
- Load
  - Limited by RDBMS capabilities, if using Load Utility (Performance)
  - Why waiting until then?
- After Load
  - Very useful, especially when Cleansing not fully considered during development phase

a. sironi

Applying the Rules: What?

- Assess
  - Identify and trace rule violations at record / field level
  - Define a metric for quality (accuracy) at record level
    - Based on number of rules violated
    - Weighted on rule importance
  - Track all rule violations
- Filter
  - As for Assess … but filter-out record for re-processing
- Correct
  - Automatically correct the offending value according to the rule

Applying Rules: Issues with Assessment Metrics

NDG	NDG-Comp	Role	Rules	Quality
1	1	U	6	6?
1	2	P	6	5?
1	3	P	6	4?
2	2	P	6	4?
...	...	...	...	...

- Rules could be mutually exclusive
  - Can never violate more than a sub-set of them, each sub-set having a different number of rules
  - Rules should have different weight
- Accuracy may be related to a set of record, not to individual ones
  - E.g. a role missing in a set
- Accuracy level should not be dependent on processing order
- Not all rules are equally important
  - Introduce weights

a. sironi

Applying the Rules: Issues with Filtering

- Overall issue: what's better …
  - Not knowing about some facts …
  - Knowing that it is not 100% accurate?
- Filtering Facts
  - No additional major issues
- Filtering Dimensions
  - Will introduce delays in loading records
  - What about incoming Facts?
    - Facts might be linked to "wrong" Dimension record
    - Should filter out also linked Facts?

What if storing information on the new price for "Pirate Rum" gets delayed until after '12.06.2004'?

PSuKey	SKU	SkuDesc	Price	Cat	CatDesc	Valid_From	Valid_To
1	122	Pirate Rum	100	K1	Spirits	01.01.2001	12.06.2004
4	276	…	…	…			
…	…						
19	122	Pirate Rum	103	K1	Spirits	12.06.2004	31.12.9999

PSuKey	SKU	CustSuKey	Sales_Date	SpSuKey	Qty
1	122	1263	11.06.2004	12	12
1	122	194	11.06.2004	21	6
4	276	1263	11.06.2004	12	1
1	122	1263	12.06.2004	15	6
…		…	…	…	…

Input Record:

SKU	CustNo	SalesPers	Date	Qty
122	C9856	SR24	12.06.2004	6

a. sironi

Applying Rules: Recommendations

- Dimensions <u>must always</u> be processed first
- Decide which errors are showstoppers
  - For the record
  - For the entire file
- If a Dimension delta-file is fully filtered
  - Stop processing all Fact files
- If individual Dimension records filtered
  - Delay loading linked Facts… OR
  - Implement consistency checks for linked Facts when Dimension record eventually loaded
    - User will see a change in query result
    - In most cases, user won't be able to understand why!
  - Implement bi-temporal model (be ready & handle complexities!)

a. sironi

Concluding Remarks

- Data quality is a well known issue
  - Difficult to tackle
  - Too often ignored…
- Assessing and managing DQ is almost always more complicated than expected
  - We covered some technical aspects
  - Fixing the roots cause of the issues may be even more challenging
- Make sure you fully understand all the ramifications
  - DB design
  - End-user impact

a. sironi

Summary

- Data Warehouse architecture
- Data modeling
  - E/R and Normalization vs. Dimensional Model
- Managing time
  - Temporal and bi-temporal models
- Data Quality
  - Issues & solutions … & issues …
- Linking all together ….

Questions?

a. sironi

Bibliography - 1

1. Christian S. Jensen, *Introduction to Temporal Database Research*, www.cs.auc.dk/~csj/Thesis/ (some 40 files, one per chapter, over 1,300 pages!))
2. Duane Hufford, *Data Warehouse Quality: Special Feature from January 1996*, DM Review
3. Karol Duncan, David L. Wells, *Rule Based Data Cleansing for Data Warehousing*
4. *Data Quality Management at the United States Patent and Trademark Office*, Rivised March 1998
5. Markus Helfert, Clemens Herrman, *Proactive Data Quality Management for Data Warehouse Systems – A Metadata Based Data Quality System*
6. Larry P. English, *Improving Data Warehouse and Business Information Quality: Methods for Reducing Costs and Increasing Profits*, John Wiley & Sons © 1999
7. Leo L. Pipino, Yang W. Lee, and Richard Y. Wang, *Data Quality Assessment*, COMMUNICATIONS OF THE ACM April 2002/Vol. 45, No. 4ve

a. sironi

Bibliography - 2

8. Richard T. Snodgrass, *Managing Temporal Data: A Five-Part Series*, (Appeared in Database Programming and Design, September, 1998). Can be found at http://www.cs.auc.dk/TimeCenter/pub.htm, look for TR-28.
9. Rick Snodgrass, *Current Modifications of Time-Varying Tables*, Intelligent Enterprise's Database Programming and Design, www.dbpd.com/vault/9901/Snod0199.shtml
10. R. Bliujute et al., *Systematic Change Management in Dimensional Data Warehousing*, www.cs.auc.dk/~simas/tss.html
11. R.Kimball, The Data Warehouse Toolkit, John Wiley & Sons (1996)
12. Tim Guinlan et al., *Modeling History for the Data Warehouse*, Intelligent Enterprise's Database Programming and Design, www.dbpd.com/vault/9811/quin.shtml
13. Christian S. Jensen and Richard T. Snodgrass, *Semantics of Time-Varying Information*, TR 96-02, March 7, 1996, ftp://ftp.cs.arizona.edu/reports/1996/TR96-02.pdf

Bibliography - 3

14. Seung-Kyum Kim and Sharma Chakravarthy, *A Retrospective Analysis of Time Concepts in Temporal Databases,* http://wuarchive.wustl.edu/doc/techreports/ufl.edu/cs/tr92/tr92-044.ps.Z
15. Ralph Kimball, *It's Time for Time*, DBMS – July 1997, Data Warehouse Architect, www.dbmsmag.com/9707d05.html
16. Rick Snodgrass, *Referential Integrity Over Time-Varying Tables*, Intelligent Enterprise's Database Programming and Design, October 1998, www.dbpd.com/vault/9810/temporal.html
17. Rick Snodgrass, *Temporal Support in Standard SQL*, Intelligent Enterprise's Database Programming and Design, October 1998, www.dbpd.com/vault/9810snod.html
18. Christian S. Jensen et al., *The Consensus Glossary of Temporal database Concepts – February 1998 Version*, downloadable from www.cs.auc.dk/~csj/Glossary/main/pages/download.html
19. Ralph, Kimball, *Stitching late-arriving records into an existing data warehouse,* www.intelligententerprise.com/000929/webhouse.shtml

a. sironi

Bibliography - 4

20. Ralph, Kimball, *Surrogate Keys*, DBMS, May 1998, www.dbmsmag.com/9805d05.html
21. TimeCenter (TimeCenter is an international center for the support of temporal database applications on traditional and emerging DBMS technologies), http://www.cs.auc.dk/TimeCenter/
22. J. Bair, *It's About Time! Supporting Temporal Data in a Warehouse*, in InfoDB Vol. 10 Number 1, http://www.dbaint.com/pdf/v10n11.pdf
23. Ralph Kimball & Associates, www.rkimball.com (papers, articles and books related to Data Warehousing)
24. Jack E. Olsen, *Data Quality: The Accuracy Dimension*, Morgan Kaufmann Publishers, 2003