University of Vienna, Sommersemester 2015

INTEROPERABILITY
- Information Integration-

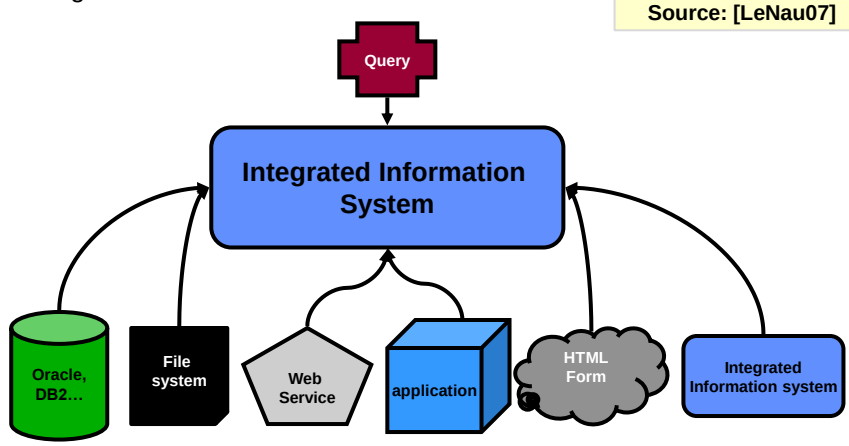
Univ. Prof. Dr. Stefanie Rinderle-Ma
Workflow Systems and Technology Group
Faculty of Computer Science
University of Vienna
stefanie.rinderle-ma@univie.ac.at

Outline

3.1 Motivation

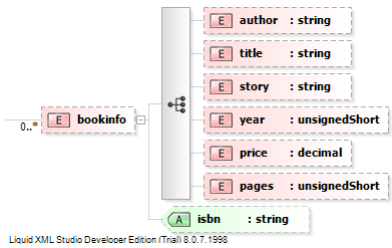
Challenge

Source: [LeNau07]

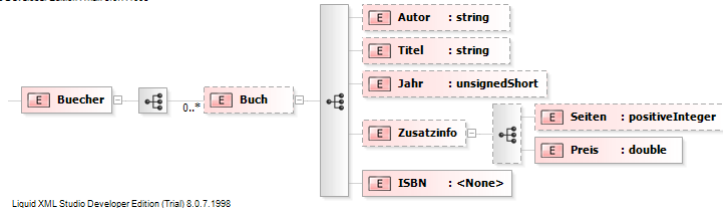


3.1 Motivation

- Small example: Integration of two XML schemas

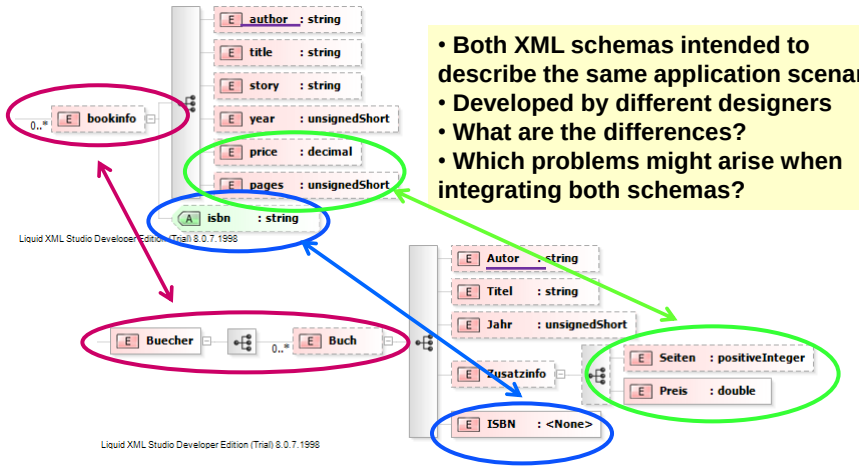


- **Both XML schemas intended to describe the same application scenario**
- **Developed by different designers**
- **What are the differences?**
- **Which problems might arise when integrating both schemas?**



3.1 Motivation

- Small example: Integration of two XML schemas



• Both XML schemas intended to describe the same application scenario
• Developed by different designers
• What are the differences?
• Which problems might arise when integrating both schemas?

3.1 Motivation

- In addition: Data Integration!!

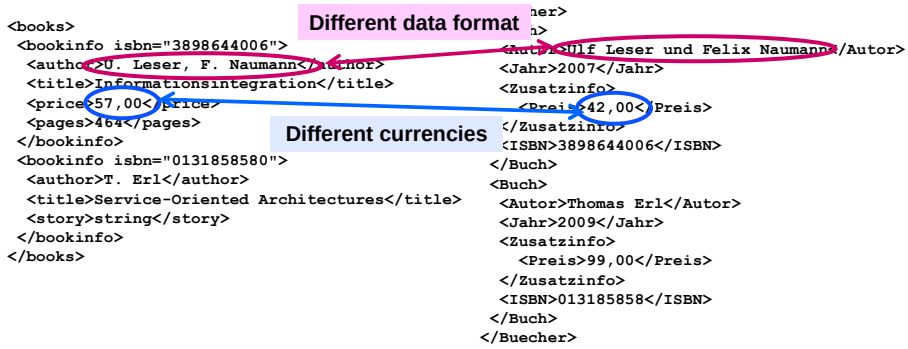
```
<books>
 <bookinfo isbn="3898644006">
  <author>U. Leser, F. Naumann</author>
  <title>Informationsintegration</title>
  <price>57,00</price>
  <pages>464</pages>
 </bookinfo>
 <bookinfo isbn="0131858580">
  <author>T. Erl</author>
  <title>Service-Oriented Architectures</title>
  <story>string</story>
 </bookinfo>
</books>
```

```
<Buecher>
 <Buch>
  <Autor>Ulf Leser und Felix Naumann</Autor>
  <Jahr>2007</Jahr>
  <Zusatzinfo>
    <Preis>42,00</Preis>
  </Zusatzinfo>
  <ISBN>3898644006</ISBN>
 </Buch>
 <Buch>
  <Autor>Thomas Erl</Autor>
  <Jahr>2009</Jahr>
  <Zusatzinfo>
    <Preis>99,00</Preis>
  </Zusatzinfo>
  <ISBN>013185858</ISBN>
 </Buch>
</Buecher>
```

Which problems arise at data level?

3.1 Motivation

- In addition: Data Integration!!



- These structural problems can be SOMEHOW solved.
- Even harder: semantical problems → Example?

3.1 Motivation

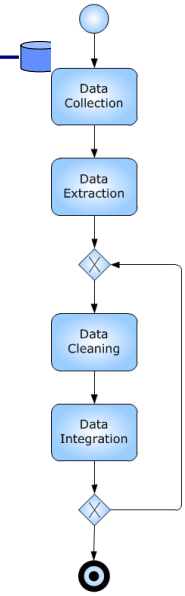
- Before schema and data integration
  - Where is the data extracted?
  - How can the data be extracted?
- Typically: heterogeneous data sources
- Examples:
  - Databases
  - File systems
  - XML documents
  - Legacy applications
- **ETL process** (Extraction → Transformation → Loading)
- Typically applied in Data Warehouse Systems (DWH)
- Goal of DWH: Integration and Analysis

Teaching objectives:

- Understanding the challenges of information integration
- Learning different techniques for schema and data integration
- Particularly being able to distinguish different terminology, for example
  - Data cleasing
  - Data scrubbing
  - Data fusion
- Schema mapping
- Schema matching
- Understanding „Pre-integration“ process (→ ETL)
- Getting insights into specific IS application by means of Data Warehouses

3.2 Data Extraction

- „It‘s all about the data. […] But data doesn’t come to you…” [Mark]
- In practice different situations
- Data sources are already existing (and accessible) → assumed in literature, practically not always the case
- Nonetheless, the relevant sources have to be selected
- Necessary data is collected „on-demand“ (or in the right format)
- Conclusio 1: **Data collection** is an active task



3.2 Data Extraction

Conclusio 1: **Data collection** is an active task

- Identification of relevant data sources
- Clarification of issues such as data access (particularly, if external data sources are to be accessed)

Use Case 1: Patient treatment processes

- EBMC2 project: http://ebmc2.univie.ac.at/
  - Co-funded by University of Vienna and Medical University of Vienna
  - Formalizing medical guidelines for skin cancer treatment
  - Mining and analysis of real-world treatment processes
  - In particular regarding their compliance with the guidelines

3.2 Data Extraction

Balance between:

- What data sources do we need (to fulfill a certain analysis goal) and
- Which data sources are actually available and accessible (privacy, data ownership, data access costs, etc.)

Available data sources [Dunkl12]:

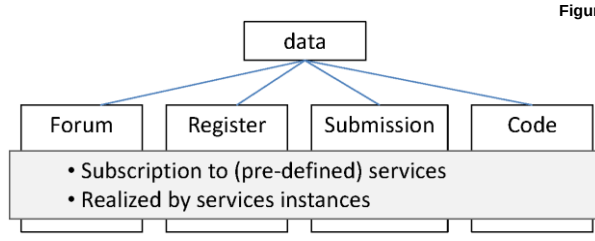
- detailed data collection of clinical Cutaneous Melanoma (CM) stage IV protocols (Stage IV Melanoma Database, **S4MDB**, for short)
- administrative data of the Main Association of Austrian Social Security Institutions comprising a billing-oriented view of medical patient treatments (**GAP-DRG**)

3.2 Data Extraction

Use Case 2: Higher-Education Data (HEP)

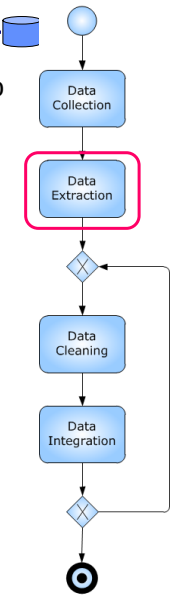
- www.wst.univie.ac.at/communities/hep/
- Collected from service-oriented learning platform CEWebs [Dern04]

Figure taken from [Ly12]



3.2 Data Extraction

- After selecting and / or collecting data sources, data has to be extracted
- Data extraction is a rather technical question:
- Classically: ETL (Extraction – Transformation – Load)
- Access to heterogeneous data sources
  - Depends on the type of data source
  - Important: do we need the the entire data (or fragments) OR do we need a data update (delta file)?
  - Example (relational) databases: offer access by query language (SQL), but also by logging
  - Example legacy systems: do not offer any support → many approaches for determining snapshot deltas, e.g., by Window algorithm [Labi96]



3.2 Data Extraction

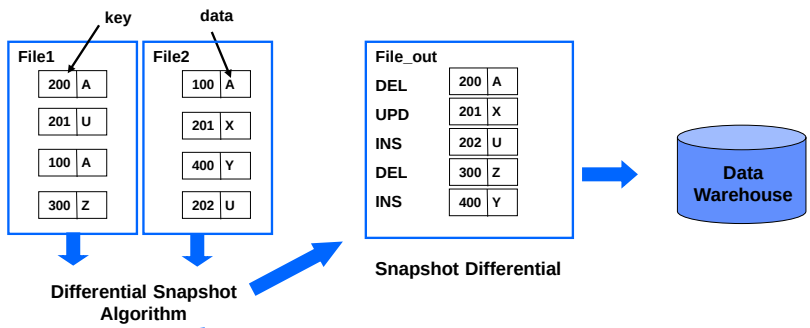
Differential snapshot → basic problem [LaGa96]

- Data extraction from legacy systems → *snapshots*, e.g., item list
- However: Possibly expensive (why?)
- Challenge:
  - How can we detect the actual changes?
  - → Calculating delta between old and new snapshot

Differential snapshot problem – basic scenario

- Snapshot F extracted from source:
  - File F: set of records (K, A_1, …, A_n) where K denotes the key and A_1, …, A_n denote the data fields of the record
- Let F_1 and F_2 be two snapshot files where F_1 has been generated before F_2
- **Goal**: determine the smallest set of UPDATE /INSERT / DELETE operation (within a so called **snapshot differential file**) that transforms F_1 into F_2, formally:
  - For two files F_1 and F_2 calculate smallest set O = {INS, DEL, UPD}* with $O(F_1) = F_2$
- O is not unique, e.g., O_1 = {INS(X), ∅, DEL(X)} ≡ O_2 = {∅, ∅, ∅}
- → **differential snapshot problem**

[LaGa96]

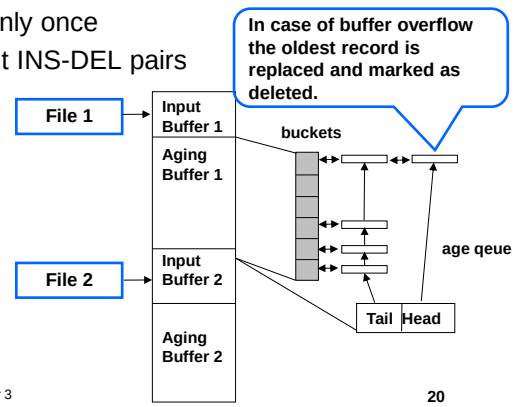


- **How to determine snapshot file?**
- **Which algorithms can you think of?**
- **Which parameters are important for performance?**

1st approach: **Nested Loop (DS_{nl})**

2nd approach: Window Algorithm (DS_{window}) [LaGa96]

- assumption: files have a fuzzy order, i.e., matching records of F_1 and F_2 are „physical neighbors“
- Merge both files over *sliding window*
- Snapshots have to be read only once
- Potentially leads to redundant INS-DEL pairs
- Number IO: f1 + f2



3.2 Data Extraction

WINDOW ALGORITHM [LaGa96]
INPUT: F1, F2, n
OUTPUT: F_{out} /* the snapshot differential */

```
(1) Input Buffer1 ← Read n blocks from F1
(2) Input Buffer2 ← Read n blocks form F2
(3) while ((Input Buffer1 ≠ ∅) and (Input Buffer2 ≠ ∅))
(4)   Match Input Buffer1 against Input Buffer2
(5)   Match Input Buffer1 against Aging Buffer2
(6)   Match Input Buffer2 against Aging Buffer1
(7)   Put contents of Input Buffer1 to Aging Buffer1
(8)   Put contents of Input Buffer2 to Aging Buffer2
(9)   Input Buffer1 ← Read n blocks from F1
(10) Input Buffer2 ← Read n blocks from F2
(11)Report records in Aging Buffer1 as deletes
(12)Report records in Aging Buffer2 as inserts
```


3.2 Data Extraction

Operation pair	Effect	Example
INSERT(R)/ DELETE(R)	Neutral	INSERT(2, „HI“), DELETE(2, NULL)
UPDATE(R)/ DELETE(R)	delete	UPDATE(2, „HI“), DELETE(2, NULL) → DELETE(2, NULL)
DELETE(R)/ INSERT(R)	a) Same attribute value → neutral b) Different attribute value → UPDATE	In DB: (2, „HI“): DELETE(2, NULL), INSERT(2,“HI“) DELETE(2, NULL),INSERT(2, „BYE“)
UPDATE(R)/ UPDATE(R)		
INSERT(R)/ UPDATE(R)		

Redundant operations in DS ([RJR07])

Remark: the first entry in each tuple corresponds to the primary key of the underlying relation.

Outline

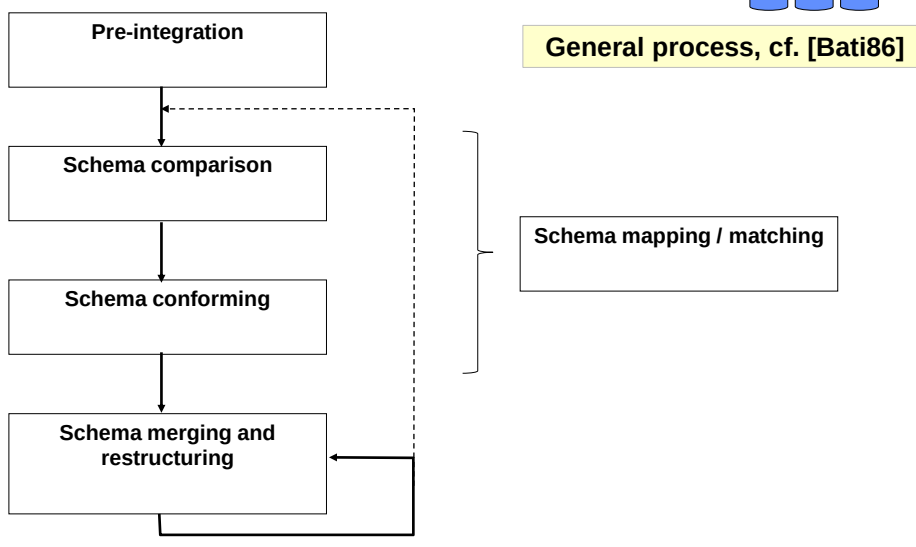
3.3 Schema and Data Integration

- General Problems:
  - Heterogeneous data sources
  - Heterogeneous schemas
  - Heterogeneous data
- General Challenges:
  - **Integration at Schema Level**: Goal is to unite different and contradicting schemas in such a way that all local entities are represented in a unique way

  - **Integration at Data Level**

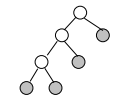
Characteristics of integrated schema – Quality Criteria for measuring the success of the integration process, cf. [Bati86]

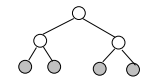
- Completeness: all objects within the participating local schemas have to be present in the global schema as well
- Correctness:
  - Semantic correctness (content)
  - No contradictions
- Minimality: no redundancies, every object just once
- Understandability: Documentation of processes, transformation steps



Integration Strategies → [Bati86]

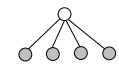
- Analysis and documentation of source schemas with respect to structure
- Determining integration strategy
  - Binary approaches: in each step two schemas are merged
    - left deep tree:
      - Local schemas are successively merged into developed schema
      - Global schema is completed or extended with new entities
    - Balanced method:
      - Level-wise iterative merging of two schemas
      - Build starting point for next integration cycle

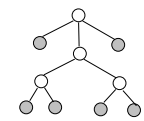




Integration Strategies ctd. → [Bati86]

- n-ary approaches
  - One-Shot strategy:
    - All existing schemas are integrated in one step
  - Iterative variant:
    - Arbitrary combination of schemas
    - Choice of integration strategy corresponds to an implicit preference (schema that are integrated in the beginning have higher influence than those that are brought in later)





- Find out about commonalities and differences between the affected schemas

3.3 Schema and Data Integration

- Frequently used taxonomy for schema integration conflicts (4 conflict classes) [SPD92]:
  - **Semantic conflicts**

    "*The two designers do not perceive exactly the same set of real world objects, but instead they visualize overlapping sets (included or intersecting sets). […] "Student" object class in one schema, while a more restrictive "CS-Student" object class in another schema."*
  - **Descriptive conflicts**

"When describing related sets of real-world objects, two designers do not perceive exactly the same set of properties (e.g., because of their different interest in the many available pieces of information on car models in the real World). "

- `Schema S1: Expensive_car (modelname, manufacturer, maximumspeed, price)`
- `Schema S2: Car_model (name, horsepower, fuelconsumption, price)`

3.3 Schema and Data Integration

- Frequently used taxonomy for schema integration conflicts (4 conflict classes) [SPD92] – continued:
  - **Heterogeneity conflicts**

    "*The designers use different data models, for example a relational one and an object-oriented one."*
  - **Structural conflicts**

    "*Lastly, even if they use the same data model, they can choose different constructs to represent common real-world objects."*
- Often: combination of different conflicts
- When looking at the different kinds of conflicts, obviously transformation of semantics and content is necessary!
  - Semantics → identify and apply precise semantics
  - Structure → Schema integration
  - Data → Data integration and Data cleansing

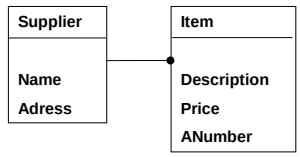
A few more words on descriptive conflicts:

- Same real-world objects within different systems are described in a different way
- Detecting and eliminating such semantic conflicts has often be done by the (human) experts (maybe partly supported by ontologies)
- Important and typical semantic conflicts:
  - Synonyms
    - Example: vehicle←→ car
  - Homonyms
    - Example: bank means either an institution for receiving, lending, exchanging, and safeguarding money or a long pile or heap; mass: a bank of earth; a bank of clouds (source: dictionary.com)
- Resolving synonyms and homonyms requires context knowledge (e.g., using „bank“ in the context of a loan application scenario indicates the actual meaning)
- Furthermore: the real semantic can be described within by using meta data

A few more words on Heterogeneity conflicts

- For a discussion of different heterogeneity conflicts and their reasons see [Lese2007]
- Different data models of the schemas to integrate
  - Examples: Network data model, hierarchical data model, OO data model, relational data model
  - Using different modeling constructs
  - Implying structural conflicts
- Transformation in a common global data model

Object-oriented:

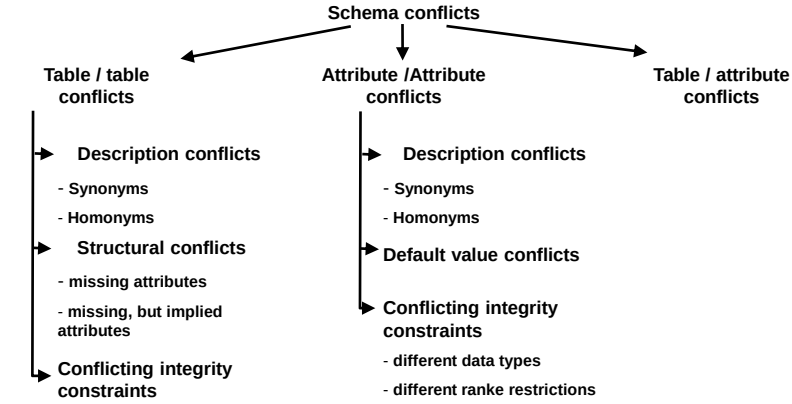


Relational:

SID	Name	Adress

SID	Description	Price	ANumber

A more detailed view on schema conflicts in relational schemas, cf. [Rahm2001]



- Schema Integration Approaches cf. [Lese2007]
  - Correspondence Assertions
  - Generic Integration Model (GIM)
- Both approaches assume that all schemas are object-oriented
- [Lese2007], p. 122:
  - Schema integration is an art
  - Highly complex
  - Mostly manual

Schema mapping, cf. [Rahm 2011]

- Part of schema comparison
- Goal: based on two schemas as input a mapping between elements of these schemas that are semantically corresponding should be found
- Formally [Bella2011, Chapter 4, pp. 82 ff.] :

> Let S* and T* be two relational schemas. Then a mapping between S* and T* is defined as (S, T, m) where S is a relation in S* and T is a relation in T* and m is a set of attribute correspondences between S and T.

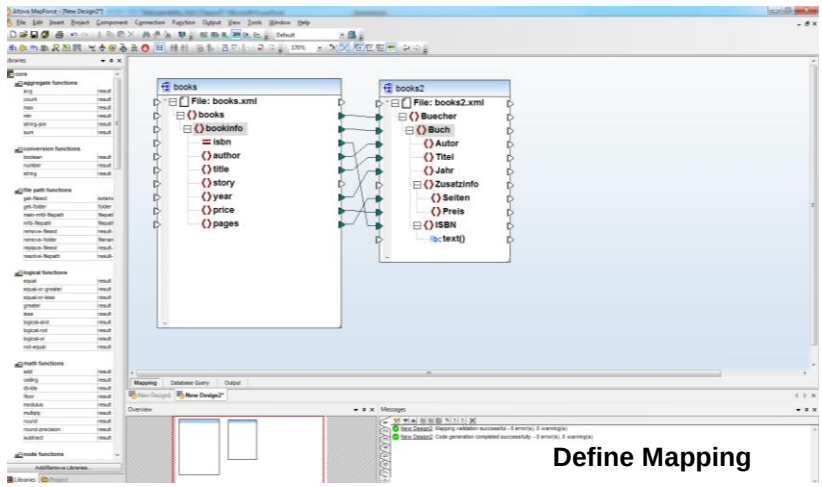
> At instance level: Let D_S and D_T be instances of S and T. Then D_S and D_T **satisfy** mapping m if for $\forall$ t_s in D_S $\exists$ t_t in D_T such that $\forall$ attribute correspondences (s, t) $\in$ m, the value of attribute s in t_s is the same value of attribute t in t_t.

Schema mapping, cf. [Rahm 2011]

- Manual task → errorprone and tedious
- General algorithm:
  - Given two schemas S and T (two relations) with attribute sets A and B
  - Core idea:
    - Build cross product A × B between all attributes from A and B
    - For each pair calculate similarity
      - E.g., regarding attribute name
      - E.g., regarding stored data
    - Choose a mapping
    - Most similar pairs until threshold
    - In addition: consider constraints

3.3 Schema and Data Integration

Altova Mapforce, http://www.altova.com/mapforce.html



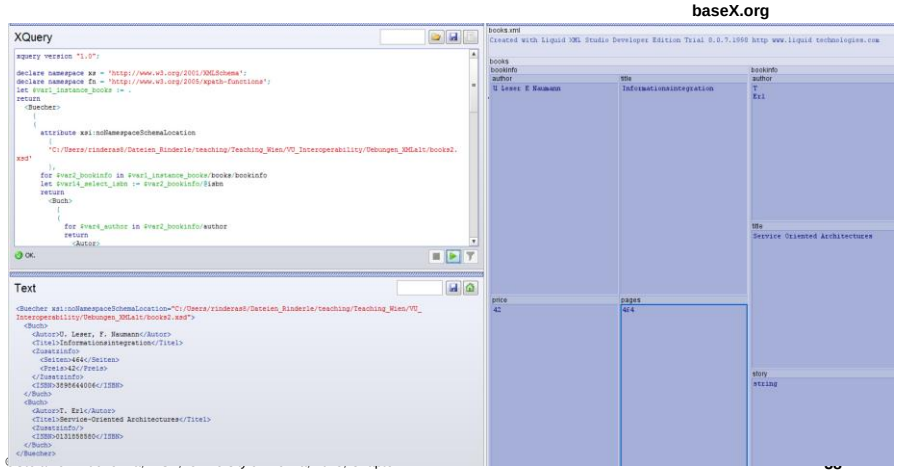
Define Mapping

3.3 Schema and Data Integration

Generate output transformation

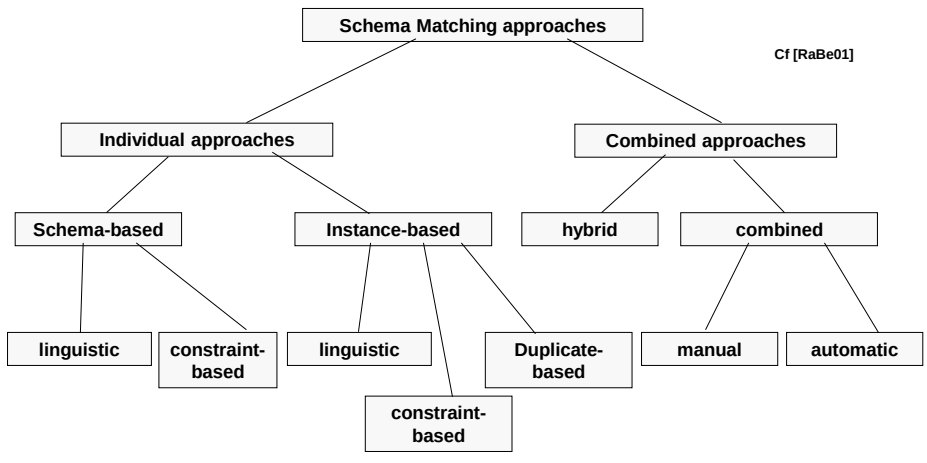
- XSLT, JAVA, XQuery

baseX.org



Schema matching, cf. [Lese2007]

- Trying to find mapping, i.e., set of correspondences between to schemas, **automatically**
- Automatic finding (supported by tools) is mostly confined to find similar labels
  - String edit distance
  - Homonyms / synonyms (→ontologies)
- Tools (commercial), cf. [Bella2011, p. 20]
  - Microsoft Biztalk
  - SAP Netweaver Process Integration
  - Altova MapForce
- Tools (open source, research)
  - COMA 3.0 (http://dbs.uni-leipzig.de/de/Research/coma.html)
  - Protègè (http://protege.stanford.edu/) and Prompt (http://protege.stanford.edu/plugins/prompt/prompt.html)
- Protègè, for example, provides support for **ontology matching**

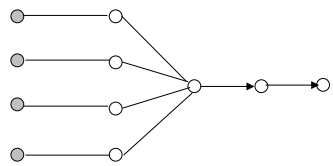


Schema Matching approaches ctd. (cf. [Lese2007]

- Label-based:
  - Considers distances between the attribute labels
  - Not always very helpful, e.g., d(year, Jahr) = 0
- Instance-based:
  - Is based on (manually defined) properties on the database instances
  - Example: length of the attribute name + attribute type
  - Limited success
- Structure-based, take into consideration:
  - Belonging of attributes to relations
  - Neighborhood of attributes
  - Newer approaches: model managament → represent relations as graphs and transfer similarity calculations to graph

Schema merging and restructuring

- The adapted schemas are overlapped and merged such that a global schema is created
- Main steps:
  - Overlapping of adapted schemas
  - Quality tests along quality criteria (completeness, correctness, minimality, understandability)
  - Further transformations of resulting schemas

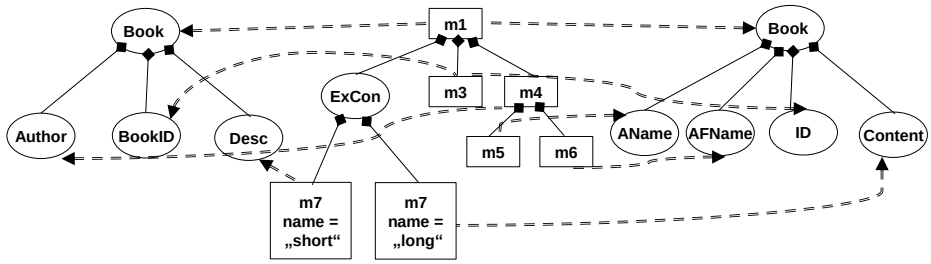


Integration **Transformation(s)**

Cf. [BaLe86]

3.3 Schema and Data Integration

- Fundamentals on schema merging have been proposed in [Bune1992]
- As for schema matching, newer approaches use model management, i.e., graph representations, as basis for the merge
- [Pott2003]: example adapted from [Bella2011, p. 240]



3.3 Schema and Data Integration

- Approaches based on 3-Ways Merge
- Typically applied in version control systems in software development, e.g., [Mens2002], e.g., SmartGit
- Prerequisite: merging two schemas where common ancestor of both schemas is known

Remember:

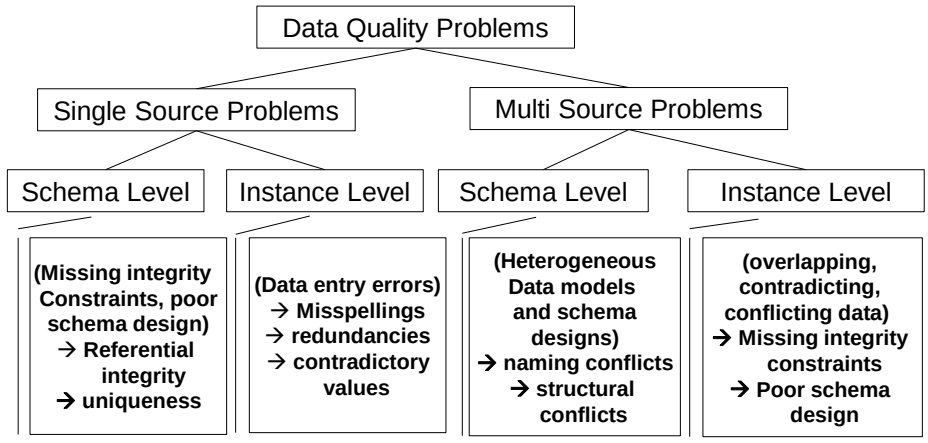
- General Problems:
  - Heterogeneous data sources
  - Heterogeneous schemas
  - Heterogeneous data
- General Challenges:
  - **Integration at Schema Level**: Goal is to unite different and contradicting schemas in such a way that all local entities are represented in a unique way
  - **Integration at Data Level**

Problems with data (cf. [Lese2007])

- Data errors
  - Different formats
  - Errors (e.g., typos)
  - Inconsistencies (e.g., zip code does not match city)
- Duplicates
- Data quality
  - Credibility
  - Relevance
- Completeness
  - Are all real world objects considered?
  - Do all attributes have values?

Data cleaning (cf. [Lese2007])

- Simple data errors: can be detected by analyzing one tuple
- Difficult data errors: can only be detected by analyzing several tuples



Dealing with data errors, cf. [Lese2007]:

- Profiling:
  - Statistical analysis of the data, typically on numeric values
  - Pattern analysis
- Assessment:
  - Stating certain conditions on the data values, e.g., weight < 100 kg
- Stating measures:
  - Fixing data errors
  - Removing error sources
- Monitoring:
  - Controlling data quality

Data normalization (cf. [Lese2007])

- (De-)capitalization
- Abbreviations / spelling: Str., street, Straße ←→ strasse, ...
- Stemming
- Names
- Formats:
  - Date: 18 February 2005, 18.02.2005, 2/18/05
  - Coding: 1: female, 2: male
  - Preciseness, field length, digits
  - Scales: grades, temperature, currency, etc.
- Good support by commercial systems (SQL)
- In addition: outlier detection, detection of duplicates

3.4 Summary & Outlook

- Dealing with data is THE prerequisite for many applications
- Often a complex and expensive task
- Calculate enough time and manpower
- Include the domain experts
- Document every step of the integration process (→ tool support)
- Always keep an eye on your analysis goals
- Also keep in mind maintenance issues (your data sources might be changing!)

References (1)

[Mark] http://mashable.com/2009/12/23/marketing-data/

[Bern08] P. A. Bernstein und L. M. Haas, „Information integration in the enterprise“, *Commun. ACM*, Bd. 51, Nr. 9, S. 72–79, Sep. 2008.

[Dunkl12] R. Dunkl, M. Binder, W. Dorda, K. A. Fröschl, W. Gall, W. Grossmann, K. Harmankaya, M. Hronsky, S. Rinderle-Ma, C. Rinner, S. Weber: On Analyzing Process Compliance in Skin Cancer Treatment: An Experience Report from the Evidence-Based Medical Compliance Cluster (EBMC2). Int'l Conf. on Advanced Information Systems Engineering (CaISE 2012), pp. 398-413 (2012)

[RJR07] Rinderle, Stefanie and Jurisch, Martin and Reichert, Manfred (2007) *On Deriving Net Change Information From Change Logs - The DELTALAYER-Algorithm.* In: Proc. 12th Conf. Datenbanksysteme in Business, Technologie und Web (BTW'07), Aachen, Germany, Lecture Notes in Informatics (LNI) P-103, Koellen-Verlag, pp. 364-381.

References (2)

[Dern04] Derntl, M., Mangler, J.: Web services for blended learning patterns. In: Proc. IEEE International Conference on Advanced Learning Technologies. 614-618 (2004)

[Ly12] Linh Thao Ly and Conrad Indiono and Jürgen Mangler and Stefanie Rinderle-Ma: Data Transformation and Semantic Log Purging for Process Mining, Int'l Conf. on Advanced Information Systems Engineering (CaISE 2012), pp. 238-253 (2012)

[LaGa96] W.J. Labio, H. Garcia-Molina: Efficient Snapshot Differential Algorithms for Data Warehousing. In Proc. Very Large Databases, pp. 63 - 74 (1996)

[Rahm2000] Erhard Rahm, und Hong Hai Do. *Data Cleaning: Problems and Current Approaches*. IEEE Data Engineering Bulletin, 23(4):3-13 (2000)

References (3)

[Bati86] Batini, Carlo, Maurizio Lenzerini, and Shamkant B. Navathe. "A comparative analysis of methodologies for database schema integration." *ACM computing surveys (CSUR)* 18, no. 4 (1986): 323-364.

[Rahm2001] Rahm, Erhard, and Philip A. Bernstein. "A survey of approaches to automatic schema matching." *the VLDB Journal* 10, no. 4 (2001): 334-350.

[Lese2007] U. Leser, F. Naumann: Information Integration. Dpunkt (2007)

[Bella2011] Z. Bellahsene, A. Bonifati, E. Rahm: Schema Matching and Mapping. Springer (2011)

[Mens2002] T. Mens, "A state-of-the-art survey on software merging", IEEE Transactions on Software Engineering, vol. 28, no. 5, pp. 449-462 (2002)

[Bune1992] A. Buneman, P. Davidson, S. Kosky: Theoretical aspects of schema merging. Advances in Database Technology (EDBT '92), pp. 152-167 (1992)