Integrating Heterogeneous Databases

— Tutorial —

Stefan Conrad & Gunter Saake
University of Magdeburg, Germany

EDBT'98, Valencia

March 23, 1998

Acknowledgements:

We are particularly grateful to the following colleagues for their contributions to the material of this tutorial:

- Andreas Christiansen (U Magdeburg)
- Wilhelm Hasselbring (U Dortmund)
- Eyk Hildebrand (U Magdeburg)
- Michael Höding (U Magdeburg)
- Ingo Schmitt (U Magdeburg)
- Kerstin Schwarz (U Magdeburg)
- Can Türker (U Magdeburg)

Contents

Contents (II)

1. Introduction

aim:

integration of pre-existing data /schemata

needed:

concepts for integrating pre-existing databases
and database systems

→ federated databases / multidatabases

focus of this tutorial:

- basic notions and architectures of federated databases
- schema integration
  - integration strategy
  - integration process
  - criteria for integration methodologies
  - conflicts and their classification
  - concrete integration approaches:
    focusing on basic principles
- other aspects of federated database systems

Motivation

aim:

uniform access to data managed by heterogeneous and autonomous (database) systems

- (logically) bringing together distributed data
  independently managed by different systems

  $\leadsto$ **integration**

Motivation (II)

2 basically different ways of system integration:

a) integration of systems within a company or organization:

→ direct and uniform access to all relevant data

e.g., by means of a *data warehouse*

→ simplifies the introduction of *workflow management systems*

for controlling and optimizing business processes

examples for integration within an organization:

– hospital information systems
– (company) environmental information systems

b) global connection of databases and information systems of different companies and organizations:

→ *global information-sharing systems*

Federation

origin of the concept of federation:

political science (and social sciences)

"federation" describes a certain kind of political structure of states and confederations of states

example:

- *Federal Republic of Germany:*

  consists of 16 federal states having their own legislation and jurisdiction

  alliance/confederation in which certain tasks and responsibilities are delegated to the Federal Government as superior authority

  based on common rules and laws (basic law / constitution)

  → federal states only partly *autonomous*

  → *conflicts* due to competing jurisdiction

Federation (II)

further examples:

- European Community
- United States of America

⇒ different forms of federations !!!

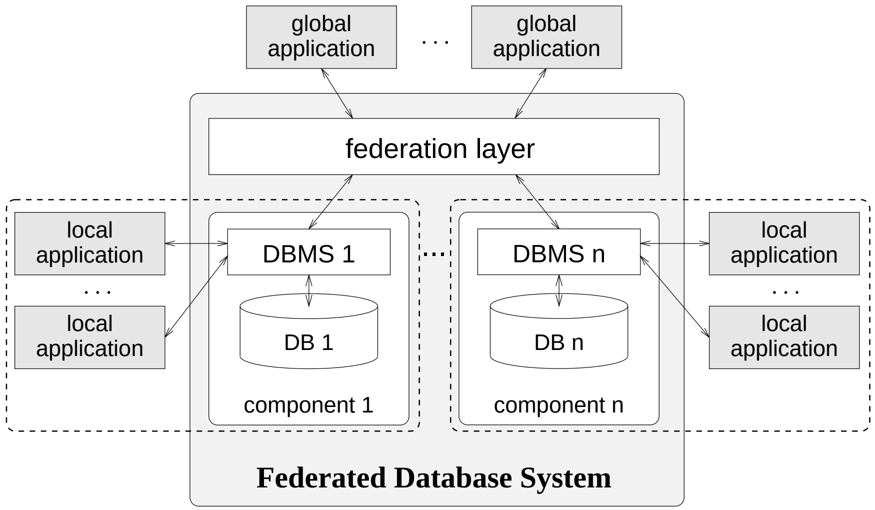
depending on

- which tasks and responsibilities are delegated to the Federal Government
  (→ degree of remaining *autonomy*)
- how much the internal structures of the federal states differ from each other
  (→ degree of *heterogeneity*)

⇒ transferring these ideas to database systems ??

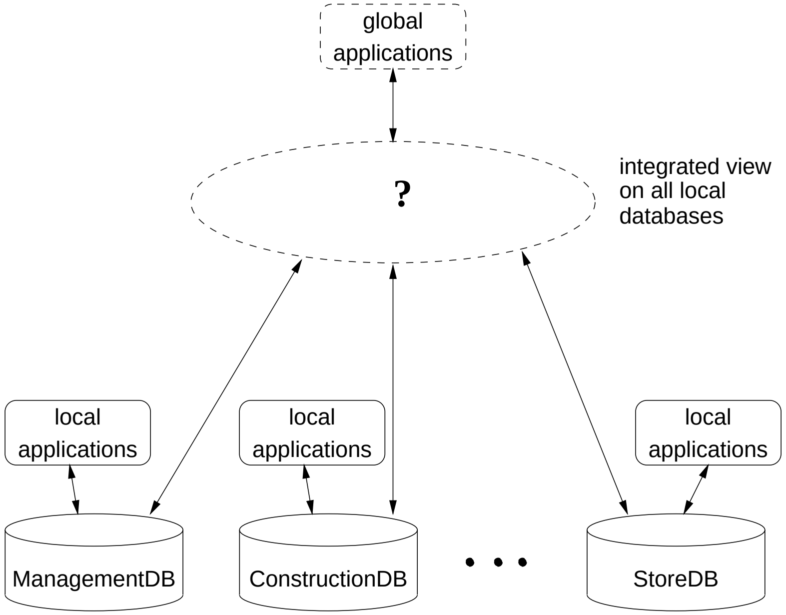
Federation (III)

general architecture for federated database systems:



- tasks of the federation layer
- communication between federation layer and component systems
- conflicts between local applications and global applications
- ...

A General Integration Scenario



requirements:

- autonomy of local systems
  → no changes of local application programs necessary
- (some way of) global integrity control
  → consistency of redundantly stored data
  → checking inter-database dependencies
- global transaction control
  → synchronizing global (and local?) applications
- distribution transparency for global interface

2. Federated Database Systems

- basic notions

  other database architectures

  distribution, heterogeneity, and autonomy

- architectures for federated database systems

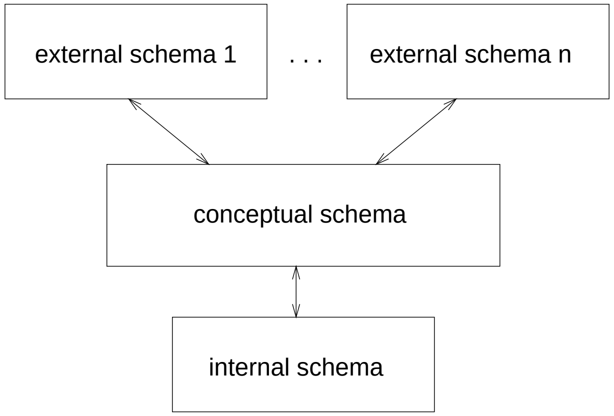
  loose coupling vs. tight coupling

  schema architectures

Basic Notions

3-level-schema architecture (ANSI/X3/SPARC)

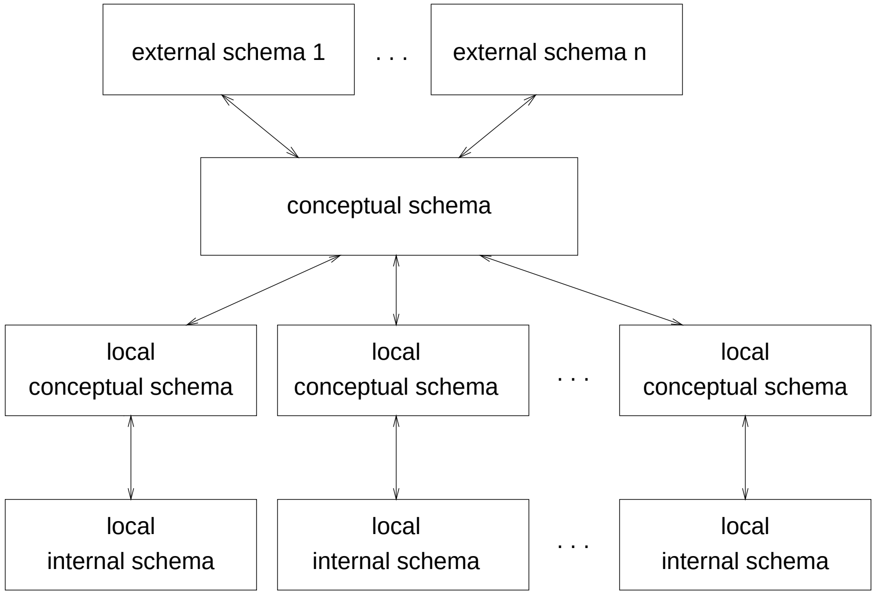
for traditional (centralized) database systems:



Basic Notions (II)

4-level-schema architecture

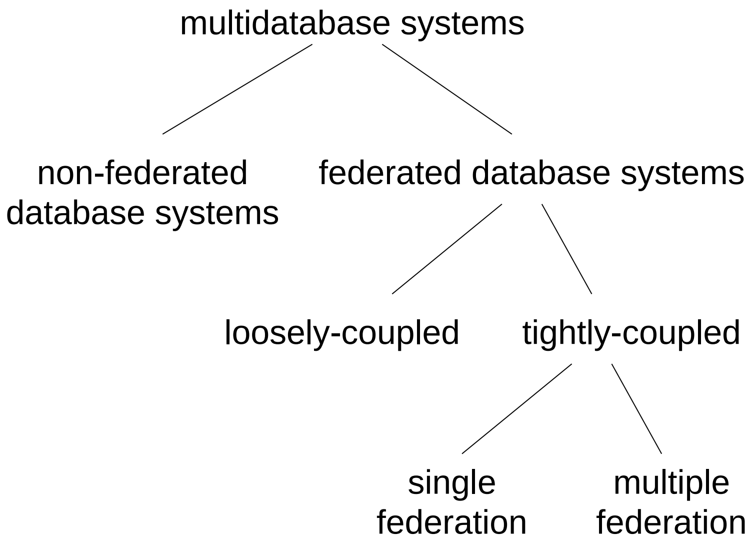
for distributed database systems:



Basic Notions (III)

classification of multidatabase systems:

[Sheth and Larson, 1990]

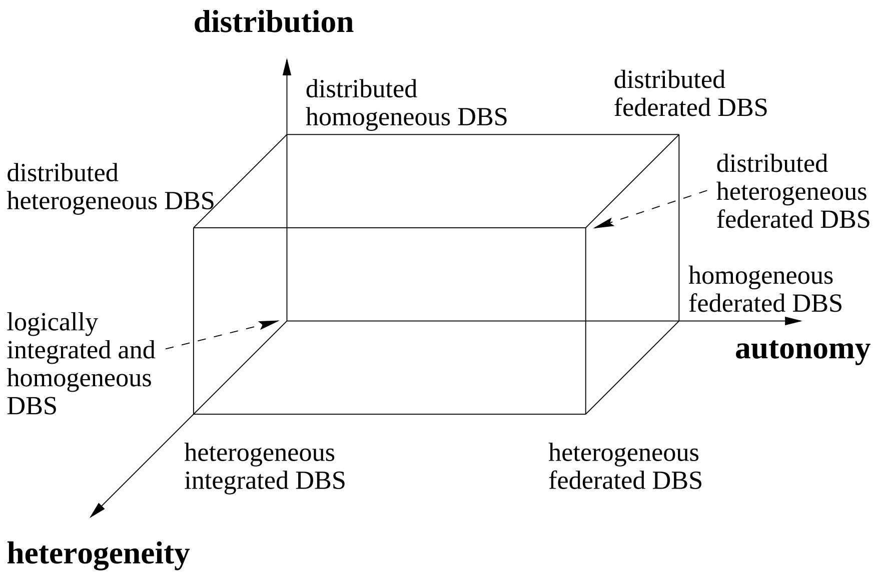


Basic Notions (IV)

variants of architectures for database systems depending on

- distribution,
- heterogeneity, and
- autonomy:

[Özsu and Valduriez, 1991a]



Basic Notions (V)

distribution of data:

(partitioning, redundancy, replication)

heterogeneity:

- heterogeneous database systems
  $\rightarrow$ heterogeneity on different system levels
  (data model, data manipulation language, transaction management, integrity control, . . .)
- semantic heterogeneity
  differences w.r.t. the meaning of data and its use by application programs

autonomy:

- design autonomy
- communication autonomy
- execution autonomy

Schema Architectures for Federated Databases

here:

3 architectures (frequently referenced in the literature):

- import-/export-schema architecture

  [Heimbigner and McLeod, 1985]

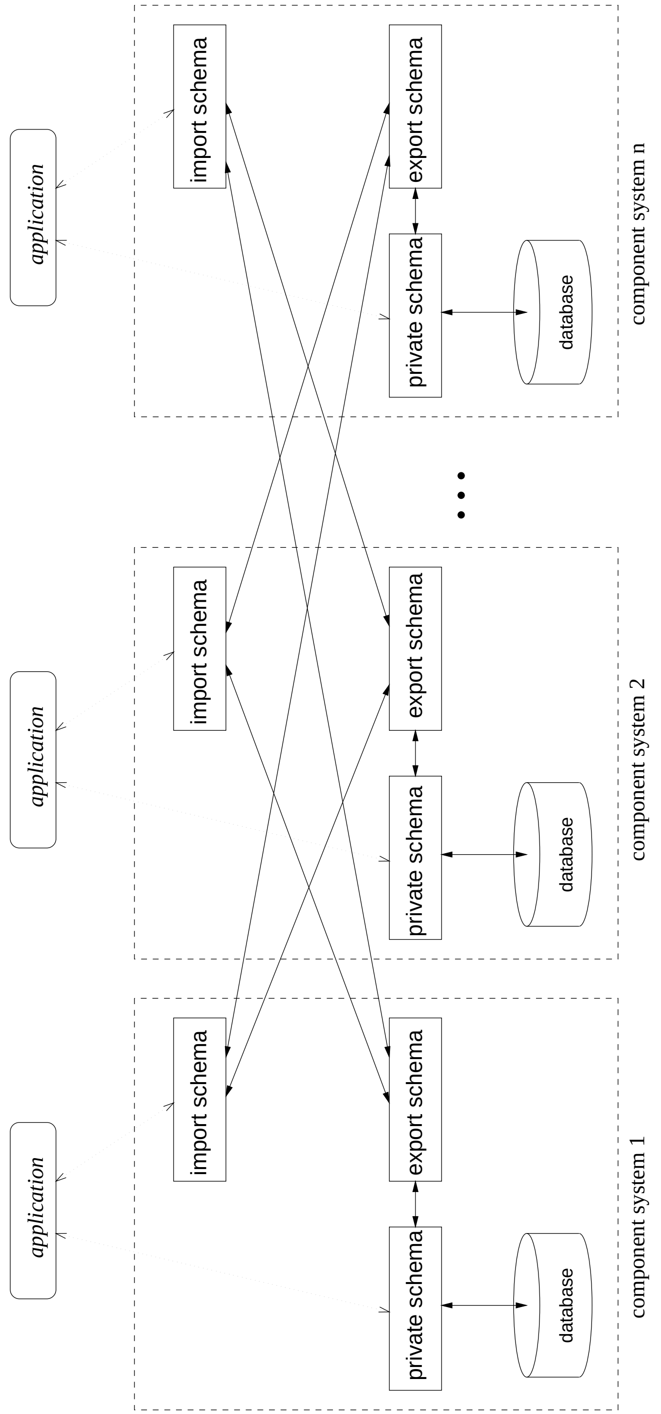
- multidatabase architecture

  [Litwin et al., 1990]

- 5-level-schema architecture

  [Sheth and Larson, 1990]

Import-/Export-Schema Architecture



Import-/Export-Schema Architecture (II)

private schema:

schema for all data stored at this node

i.e., the local conceptual schema

export schema:

describes the data which may be accessed by other systems,

i.e., not all data accessible by all systems

including access rights for the other systems

import schema:

describes data imported from other systems

access to the export schemata of other systems must be negotiated

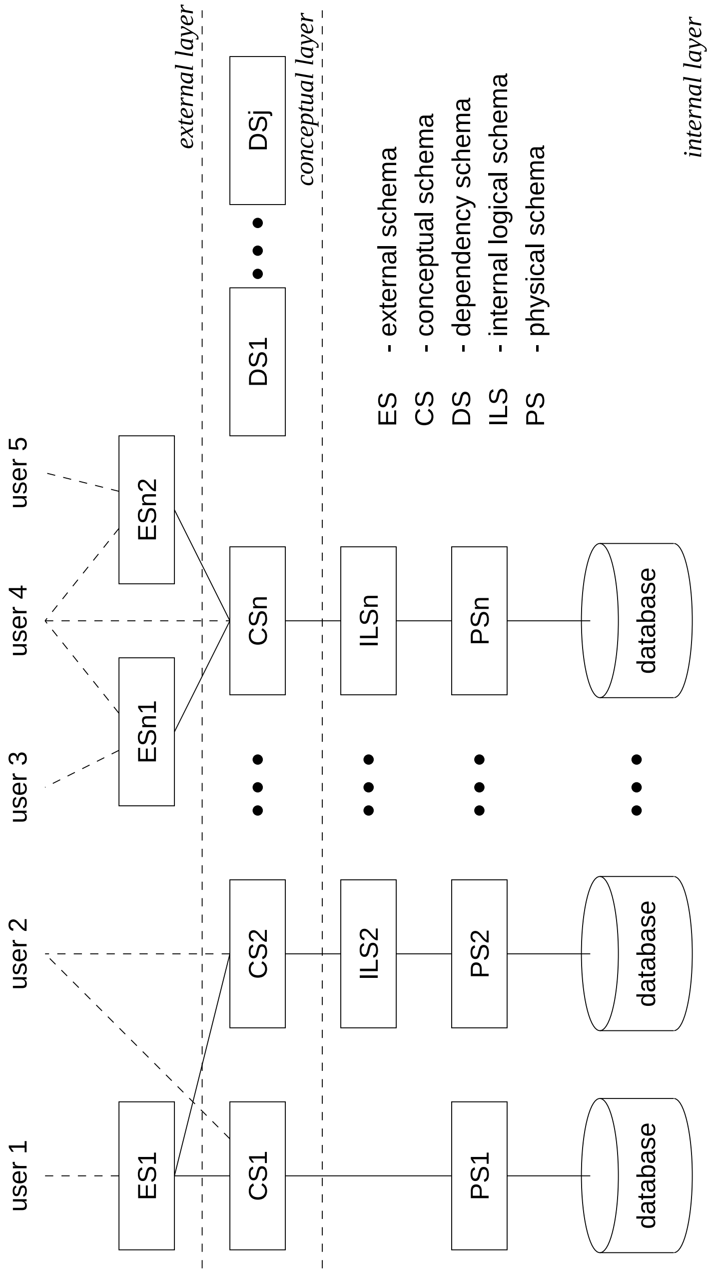
user / application programmer responsible for correct integration of data accessed via different schemata, (no *a priori* integration)

access (rights) must be negotiated bilaterally between component systems

basic architecture for loosely coupled federated database systems

→ full autonomy

Multidatabase Architecture



Multidatabase Architecture (II)

physical schema:

internal/physical schema of a component system

internal logical schema:

local conceptual schema of a component system

conceptual schema:

local external schema of a component system

defines the part of the internal logical schema accessible from the multidatabase layer

one global data model for all conceptual schemata in a multidatabase system

Multidatabase Architecture (III)

external schema:

usually defined by multidatabase user

view definition based on possibly several conceptual schemata

allows filtering, restructuring, and integration of conceptual schemata according to the personal needs or preferences of the user

requires a special multidatabase language

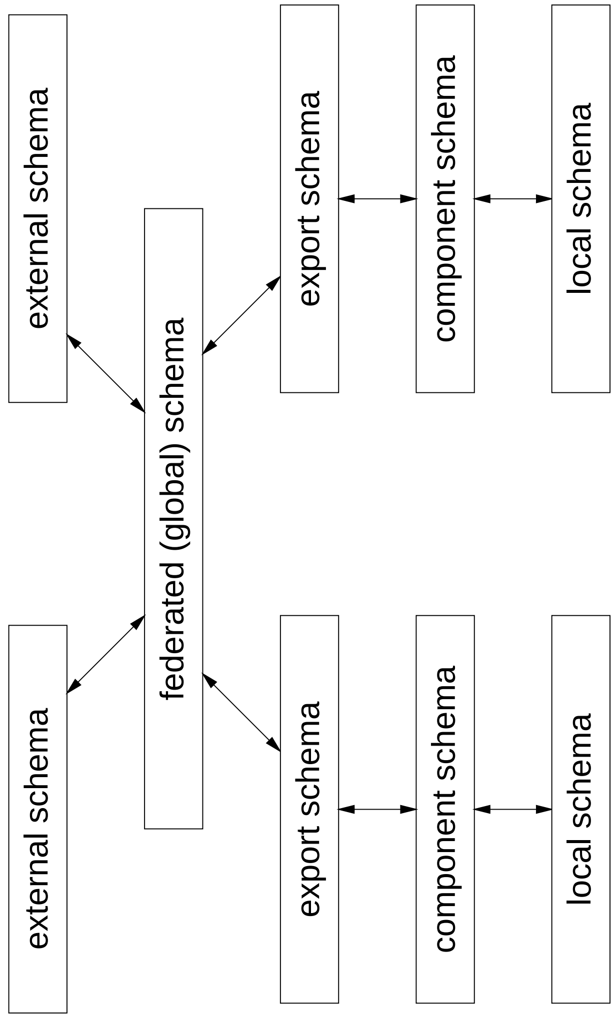
dependency schema:

for describing inter-database dependencies and
for defining additional global integrity constraints

→ basis for global integrity checking and enforcement

user / application programmer responsible for correct integration of data accessed from different schemata

5-Level-Schema Architecture



5-Level-Schema Architecture (II)

local schema:

local conceptual schema of a component system

expressed in the local data model of that system

component schema:

transformation of local schema into global data model

no separate component schema necessary,
if local data model is global data model

export schema:

contains the parts of the component schema exported to the federation layer

i.e., parts which may be accessed by global application

federated (global/integrated) schema:

integrated view on all given export schemata

removes redundancy in export schemata

resolves structural differences and other conflicts between export schemata

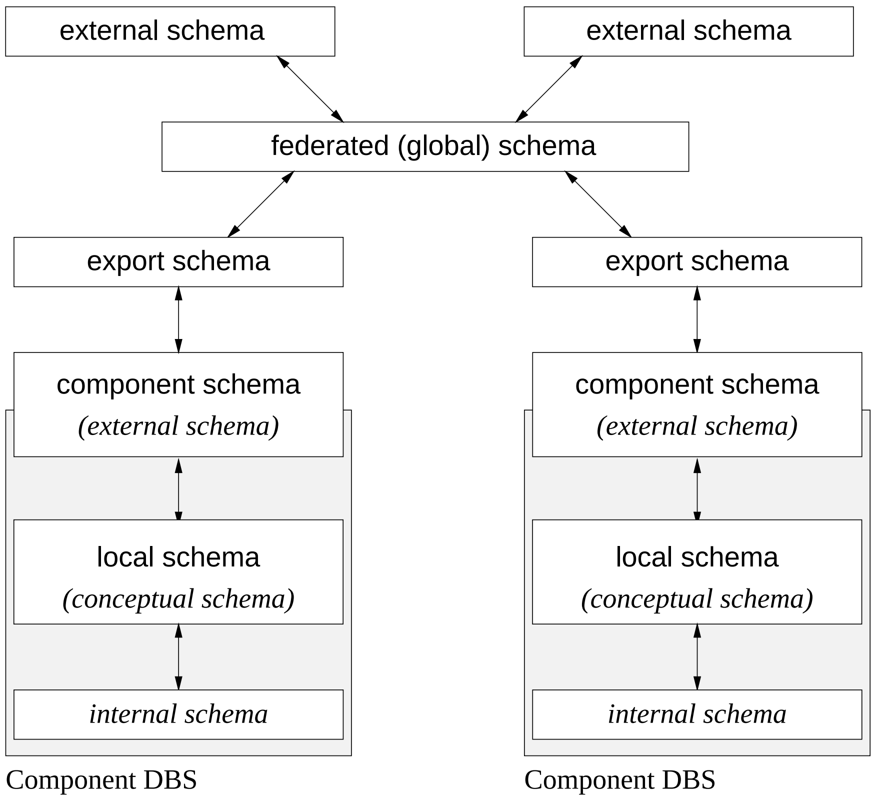
external schema:

view on the federated schema for certain applications or users

like external schema in the traditional 3-level-schema architecture

5-Level-Schema Architecture (III)

mapping 5-level-schema architecture onto
classical 3-level CDBS (1st variant):

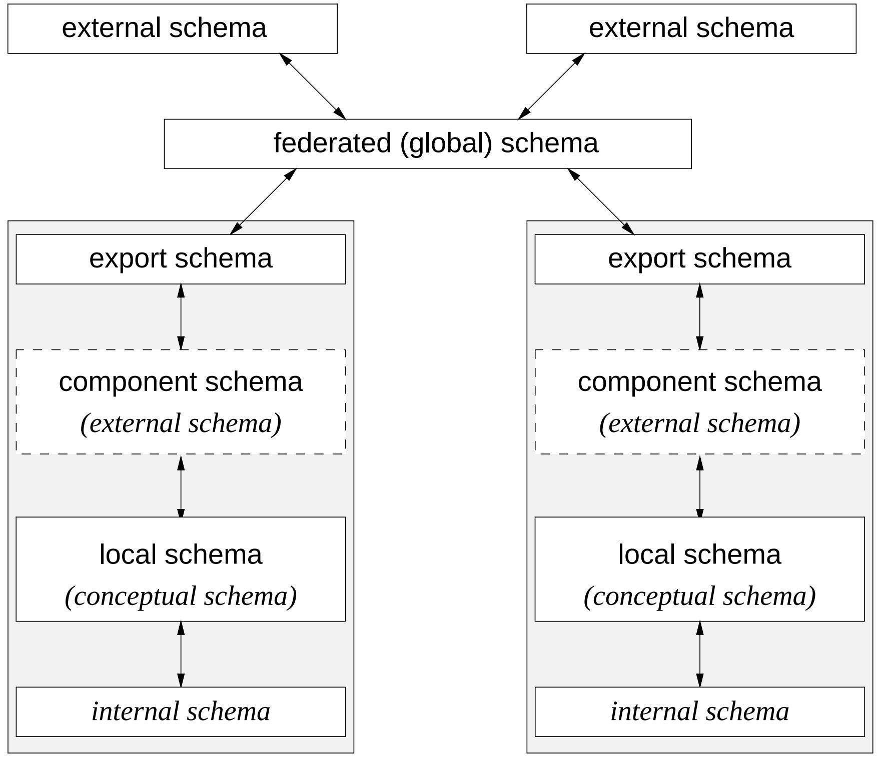


→ export schema is defined outside CDBS

→ component schema may have different data model than local (conceptual) schema

5-Level-Schema Architecture (IV)

mapping 5-level-schema architecture onto classical 3-level CDBS (2nd variant):

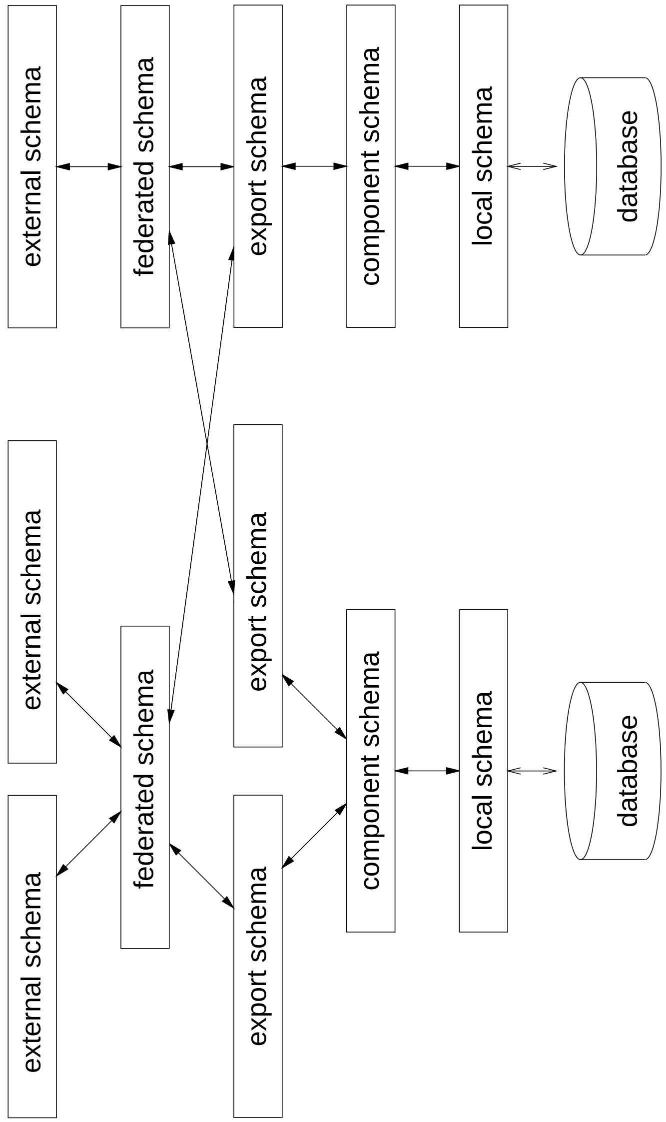


Component DBS

Component DBS

→ export schema and component schema on external layer of CDBS (both possibly using global data model)

5-Level-Schema Architecture with Multiple Federation



Comparison of the Architectures

	Architecture		
	import/export	multidatabase	5-level-schema
supported kind of coupling of component systems	loose coupling	loose coupling	loose and tight coupling
responsible for integration	user and local administrator	user	global administrator
access to the federation	from local system	via global interface	via global system
support of global DBMS-functionality	none	partially (e.g., by MDB-language)	aims at full DBMS-functionality

Selected References

textbooks on distributed databases:
[Ceri and Pelagatti, 1984], [Özsu and Valduriez, 1991b], [Bell and Grimson, 1992], [Özsu et al., 1994]

survey articles introducing into multidatabases and federated databases: [Heimbigner and McLeod, 1985], [Sheth and Larson, 1990], [Litwin et al., 1990],

. . . focussing on object-orientation:
[Pitoura et al., 1995]

textbook on federated database systems:
[Conrad, 1997, in German]

further articles as general introduction: [Litwin, 1988], [Hsiao and Kamel, 1989], [Breitbart, 1990], [Özsu and Valduriez, 1991a], [Hsiao, 1992a], [Hsiao, 1992b], [Özsu and Valduriez, 1994]

collection of introductory articles: [Hurson et al., 1994]

classifications of multi-database systems / federated database systems: [Sheth and Larson, 1990], [Özsu and Valduriez, 1991a], [Bright et al., 1992], [Özsu and Valduriez, 1994]

3. Federated Database Design

- integration strategies
- integration process
- criteria for integration methods
- integration conflicts
- classification of conflicts
- basic approaches to schema integration

Federated Database Design: General Description of the Problem

integration of pre-existing schemata

→ heterogeneity due to design autonomy

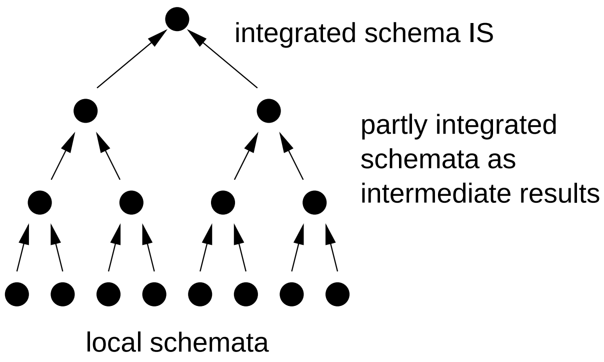
- overlapping Universes of Discourse
- different data models
- different modeling constructs
- different modeling perspective

  i.e., different views of database designers
- different ways and possibilities of modeling
- semantic heterogeneity
- . . .

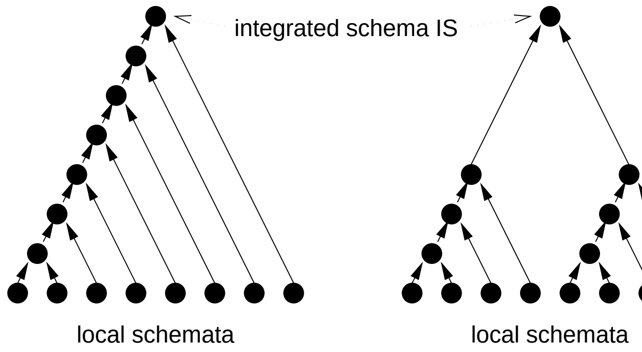
Integration Strategies

binary integration strategies:

- balanced:



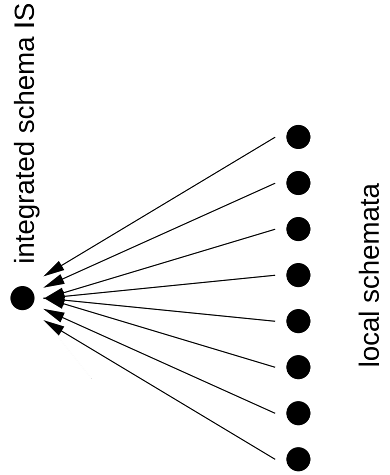
- non-balanced / weighted: for instance



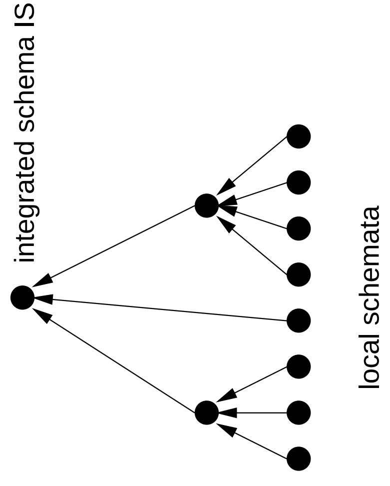
(a) **(b)**

Integration Strategies (II)

n-ary integration strategies: (a) one shot; (b) iterative



(a)



(b)

Integration Process

integration process with 2 phases:

[Spaccapietra et al., 1992]

pre-integration phase:

finding all correspondences and differences between the local schemata

$\leadsto$ inter-schema correspondences

integration phase:

generating the integrated schema

- using the given inter-schema correspondences
- applying general integration rules

result of integration not always uniquely determined

$\rightarrow$ interaction with database administrator necessary

Integration Process (II)

more detailed description based on 4 phases:

[Batini et al., 1986]

1. pre-integration:

   choose an integration strategy

   assign weights/preferences to the local schemata (if necessary)

2. comparison of schemata:

   detect correspondences between elements of the schemata

   detect possible conflicts

3. adaptation of schemata:

   transform/modify the local schemata such that all conflicts are resolved

4. integration (and restructuring, if necessary)

Criteria for Integration Methods

[Batini et al., 1986]

completeness

integrated schema must contain all schema elements given in at least one of the local schemata

i.e., no loss of information contained in local schemata

correctness

for each element in the integrated schema there must exist a corresponding (semantically equivalent) element in one of the local schemata

exception:
inter-schema dependencies found during schema comparison

no conflicts between inter-schema dependencies and information of local schemata
(i.e., only consistent "completions" are allowed)

minimality

concepts of the real world modeled in several local schemata may only be represented once in the integrated schema

understandability

the integrated schema should be understandable
(easy-to-use for global users)

Classification of Integration Conflicts

one possible classification:

[Spaccapietra et al., 1992]

- semantic conflicts
- description conflicts
- heterogeneity conflicts
- structural conflicts

usually, combined occurrences of these conflict types!!

Semantic Conflicts

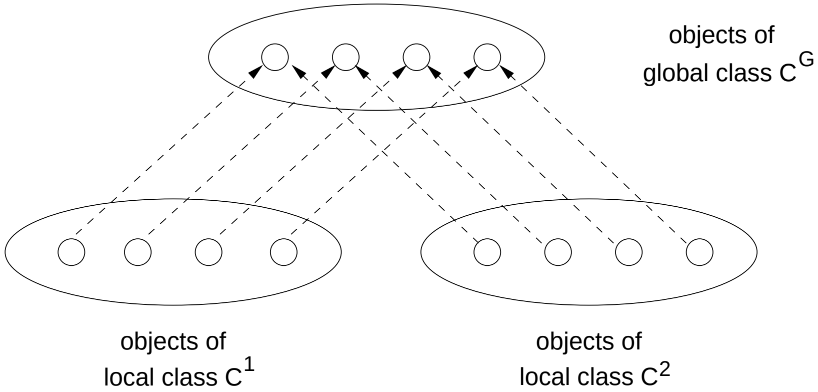
semantic conflicts:

semantically overlapping Universes-of-Discourse with corresponding object classes

problem: often, not exactly the same set of real-world objects represented

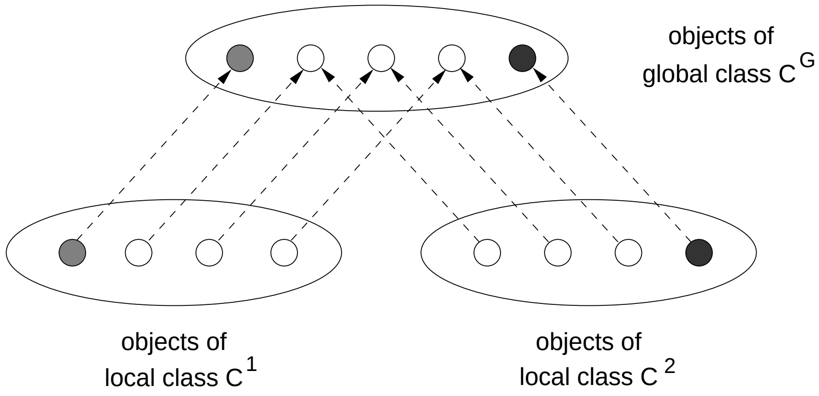
⇝ semantically *equivalent*, *overlapping* or *disjoint* class extensions

a) semantically equivalent class extensions:

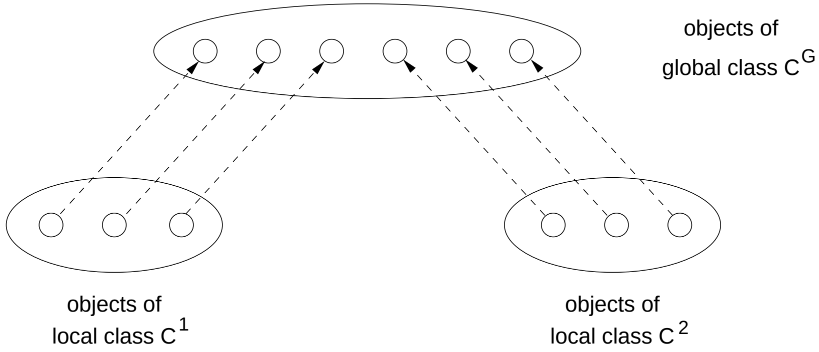


Semantic Conflicts (II)

b) semantically overlapping class extensions:



c) semantically disjoint class extensions (belonging together):



Description Conflicts

description conflicts:

different sets of properties / attributes in local schemata for the same real-world objects

homonyms and synonyms

range conflicts:
different ranges for corresponding attributes

scaling conflicts:
different units of measurement, different scaling

conflicts due to integrity constraints or special manipulation operations

Example:

DB1:	*attributes*	Price	(USD)
		Quantity	(integer)
DB2:	*attributes*	Value	(PTA)
		Number	(float)

Heterogeneity Conflicts

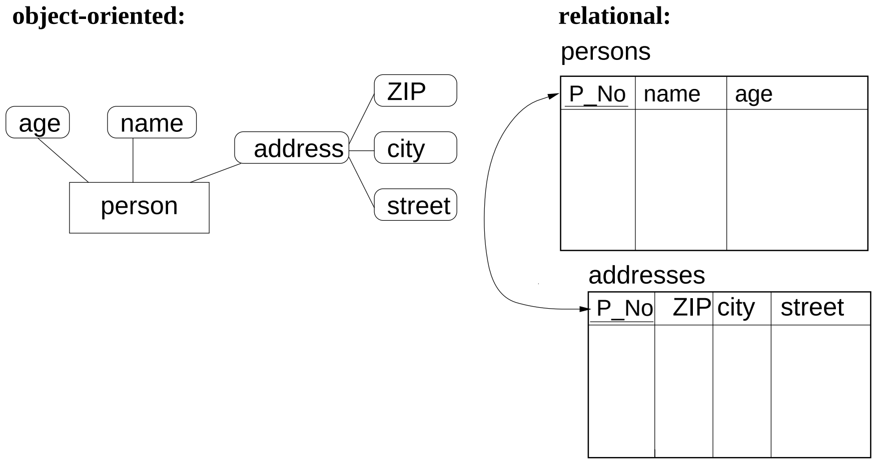
heterogeneity conflicts:

different data models used for local schemata

$\rightarrow$ different sets of modeling concepts, differences w.r.t. expressive power

$\rightarrow$ often, also reason for structural conflicts

Example:



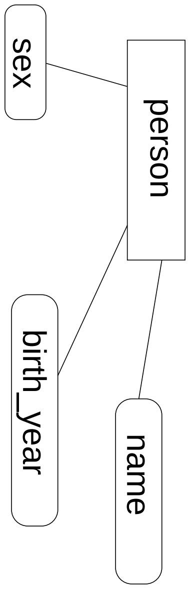
$\Rightarrow$ transformation into one global data model

Structural Conflicts

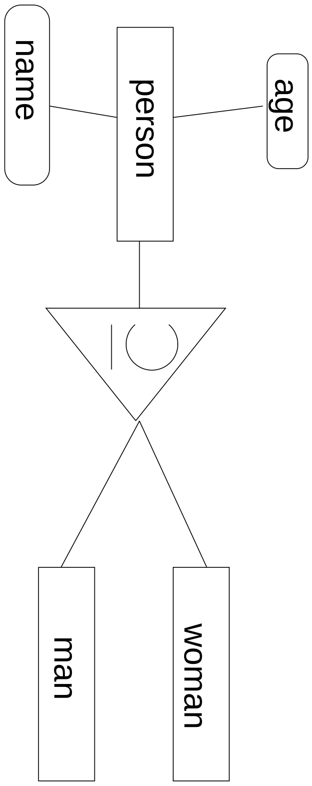
structural conflicts:
different possibilities to model the same real-world fact
(even when using the same data model)
particularly for semantically rich data models
(offering a lot of modeling concepts)

Example:

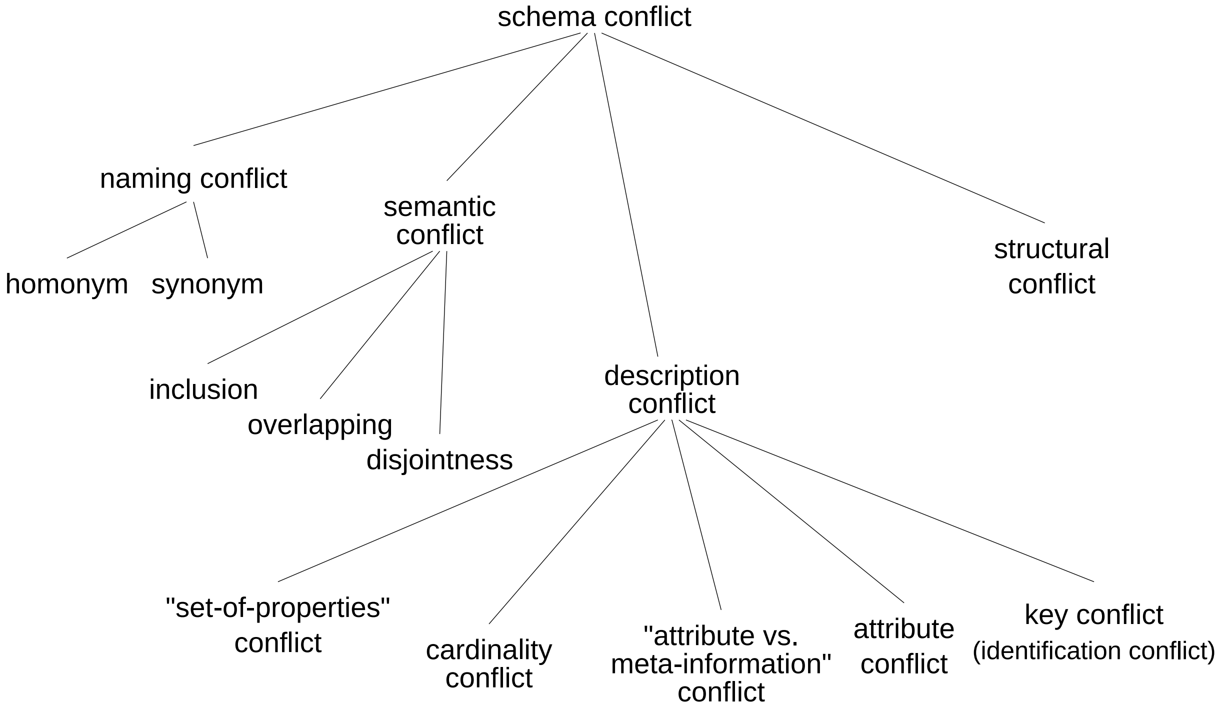
A:



B:



Classification of Integration Conflicts: Overview



A More Detailed Classification

classification of conflicts between relational schemata:
[Kim et al., 1995]

I. **schema conflicts**
  A. table-table conflicts
    1. one table vs. one table
      a. table name conflicts
        1) different names for the same table
        2) same names for different tables
      b. table structure conflicts
        1) missing attributes
        2) missing, but implicit attributes
      c. integrity constraint conflicts
    2. multiple tables vs. multiple tables
  B. attribute-attribute conflicts
    1. one attribute vs. one attribute
      a. attribute name conflicts
        1) different names for the same attributes
        2) same names for different attributes
      b. default-value conflicts
      c. integrity constraint conflicts
        1) data type conflicts
        2) condition conflicts
    2. multiple attributes vs. multiple attributes
  C. table-attribute conflicts

A More Detailed Classification (II)

II. **data conflicts**
  A. wrong data
    1. incorrect data
    2. obsolete data
  B. different representations
    1. different expressions
    2. different units of measurement
    3. different precision

Example:

Persons

name	birth_year	profession
K. Zuse	1902	scientist
C. Wolff	1928	Poet
...	...	...

Persons

name	birth_year	profession
Konrad Zuse	2	scientits
Christa Wolff	28	Writer
...	...	...

Concrete Integration Approaches

large number of integration approaches proposed in the literature

often only minor differences

presenting one integration approach in full detail ?
$\leadsto$ *a separate tutorial !!!*

here:

presentation of basic principles used by a large number of concrete integration approaches:

- assertion-based integration
- *upward inheritance* principle
  (used by most object-oriented integration approaches)

and one "completely different" approach

- Generic Integration Model (GIM)

Assertion-based Integration

correspondence assertions:

- element-correspondence assertions

  with
  related attribute-correspondence assertions

- path-correspondence assertions

notation adopted from [Spaccapietra et al., 1992]

Element-Correspondence Assertions

X_1, X_2 schema elements of schema S_1 and S_2, resp.

(object types / object classes, complex attributes, etc.)

- $X_1 \equiv X_2$ (equivalence):
  X_1 and X_2 always represent the same set of real-world objects
  (in every possible state)
  $\rightarrow X_1$, X_2 semantically equivalent
- $X_1 \supseteq X_2$ (inclusion):
  X_2 always represent a subset of the real-world objects represented by X_1
- $X_1 \cap X_2$ (overlapping):
  the sets of real-world objects represented by X_1 and X_2 can have a non-empty intersection
  (for some states)
- $X_1 \neq X_2$ (disjointness):
  the sets of real-world objects represented by X_1 and X_2 are always disjoint
  [this kind of assertion is useful in case that in the integrated schema X_1 and X_2 are to be united into one schema element]

Corresponding Attribute Assertions

describing the correspondences of attributes of two schema elements for which an element-correspondence assertion is given

$=$ the attributes have always the same value
(for each state)

$\supseteq$ values of one attribute always include values of the other attribute

for set-valued attributes: usual subset relationship

for single-valued attributes:
values equal or
null value on the right-hand side attribute

$\cap$ overlapping sets of values

($\rightarrow$ requires set-valued attributes)

$\neq$ disjoint (sets of) values

notation for element-correspondence assertions with additional attribute assertions, e.g.:

$$X_1 \supseteq X_2 \text{ with corresponding attributes:}$$
$$A_{11} = A_{21},\ A_{12} \cap A_{22}$$

Path-Correspondence Assertions

path $X_1 - X_2 - \ldots - X_n$ in a schema
based on relationships (links) $X_i - X_{i+1}$

a path-correspondence assertion compares sets of pairs of real-world objects;
such a pair of objects describes two objects related to each other by a relationship which is represented by a path in a schema

- path equivalence:
  $X_1 - X_2 - \ldots - X_n \equiv Y_1 - Y_2 - \ldots - Y_p$
- path inclusion:
  $X_1 - X_2 - \ldots - X_n \supseteq Y_1 - Y_2 - \ldots - Y_p$
- path overlapping:
  $X_1 - X_2 - \ldots - X_n \cap Y_1 - Y_2 - \ldots - Y_p$
- path disjointness / path exclusion:
  $X_1 - X_2 - \ldots - X_n \neq Y_1 - Y_2 - \ldots - Y_p$

there must be element correspondences for X_1 and Y_1 as well as for X_n and Y_p

Integration Rules

here:
simplified presentation of some example rules
(where mapping information is omitted)

remark:
special adaptation of these rules for concrete data models needed !

Rule 1: (independent elements)

for a schema element X_1 of schema S_1
with no assertion relating X_1 to an element of schema S_2

copy X_1 (together with its attributes) into the integrated schema S

X is semantically equivalent to X_1 ($X \equiv X_1$)

analogously for links $X_1 - Y_1$ with no correspondence

Integration Rules (II)

Rule 2: (equivalent elements)

given an element-correspondence

$X_1 \equiv X_2$ with corresponding attributes:
$attcor_1(A_{11}, A_{21}), \ldots, attcor_j(A_{1j}, A_{2j})$.

integrate X_1 and X_2 to X ($X \equiv X_1$, $X \equiv X_2$)

for each attribute A of X_1 or X_2 with no correspondence $attcor_i$:
copy A into the integrated schema
(as attribute of X)

for $attcor_i(A_{1i}, A_{2i})$:

$A_{1i} = A_{2i}$: copy A_{1i} (or A_{2i}) as attribute of X
$A_{1i} \supseteq A_{2i}$: copy A_{1i} as attribute of X
$A_{1i} \cap A_{2i}$: insert A as attribute of X
where domain of A is the union of the domains of A_{1i}, A_{2i}
$A_{1i} \neq A_{2i}$: insert A as attribute of X
where domain of A is the union of the domains of A_{1i}, A_{2i}

basic principle: always bring the less restricted structure into the integrated schema

Integration Rules (III)

Rule 3: (path integration)

given:
$A_1, B_1, \ldots, X_1$ elements in schema $S1$,
$A_2, B_2, \ldots, X_2$ elements in schema $S2$,
and $A_1 \equiv A_2$, $X_1 \equiv X_2$.

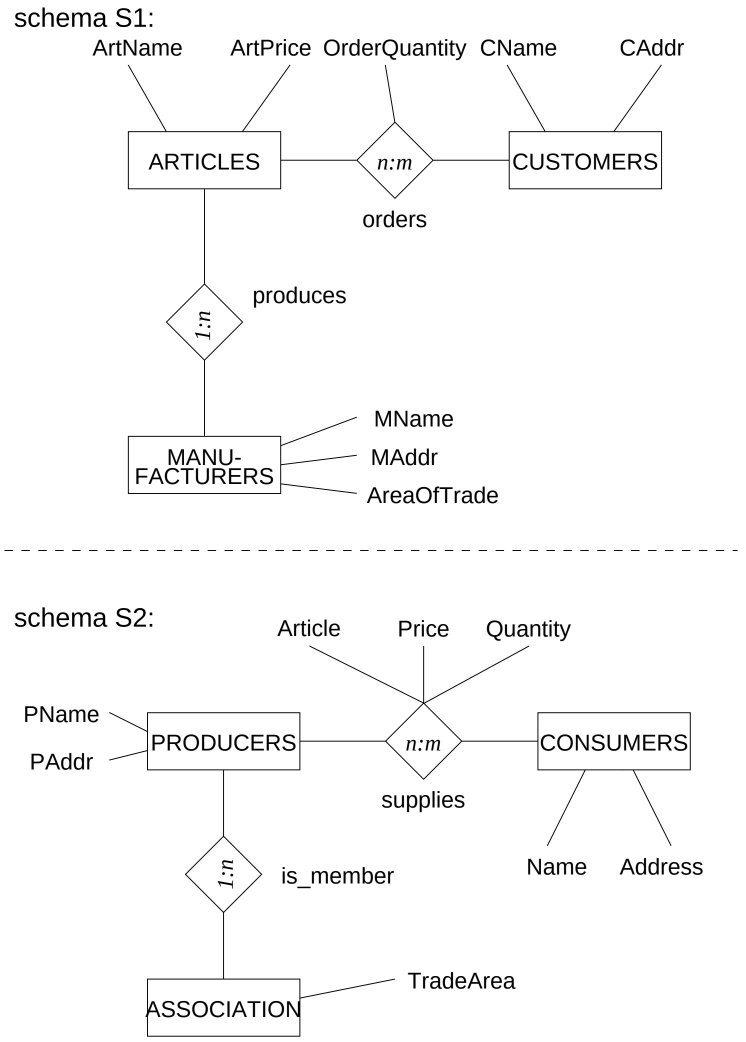
assume A and X to be the integrated elements corresponding to A_1, A_2 and X_1, X_2, respectively:

- $A_1 - X_1 \equiv A_2 - B_2 - \ldots - X_2$:
  requires $A - B_2' - \ldots - X$ as path in the integrated schema
  (where B_2' element in S corresponding to B_2)
- $A_1 - B_1 - \ldots - X_1 \equiv A_2 - B_2 - \ldots - X_2$
  results in two paths in the integrated schema:
  $A - B_1' - \ldots - X$ and $A - B_2' - \ldots - X$
  (with B_1', B_2' integrated elements corresponding to B_1 and B_2, resp.)
  additional integrity constraint needed:
  $\rightarrow$ the two paths always relate the same pairs of objects

special case: two paths of length 1

Example

Entity-Relationship schemata to be integrated:



Example (II)

correspondence assertions:

1. CUSTOMERS $\equiv$ CONSUMERS
   with corresponding attributes:
   CName = Name
   CAddr = Address

2. MANUFACTURERS $\equiv$ PRODUCERS
   with corresponding attributes:
   MName = PName
   MAddr = PAddr

3. ARTICLES $\equiv$ supplies
   with corresponding attributes:
   ArtName = Article
   ArtPrice = Price

4. MANUFACTURERS–produces–ARTICLES
   = PRODUCERS–supplies

5. ARTICLES–orders–CUSTOMERS
   = supplies–CONSUMERS

Example (III)

application of integration rules:

1. *rule 2 applied to correspondence 1:*

   S1.CUSTOMERS, S2.CONSUMERS

   integrated to entity type CUSTOMERS
   with attributes CName, CAddr

2. *rule 2 applied to correspondence 2:*

   S1.MANUFACTURERS, S2.PRODUCERS

   integrated to entity type MANUFACTURERS
   with attributes MName, MAddr, AreaOfTrade

3. *rule 2 applied to correspondence 3:*

   S1.ARTICLES, S2.supplies

   integrated to entity type ARTICLES
   with attributes ArtName, ArtPrice, Quantity

Example (IV)

4. *rule 1 applied to:*

   schema elements without correspondences:

   - relationship type S1.orders with attribute OrderQuantity
   - relationship type S1.produces
   - entity type S2.ASSOCIATION with attribute TradeArea
   - link between S1.orders and S1.CUSTOMERS
   - link between S1.orders and S1.ARTICLES
   - link between S1.produces and S1.ARTICLES
   - link between S1.produces and S1.MANUFACTURERS
   - link between S2.is_member and S2.PRODUCERS
   - link between S2.is_member and S2.ASSOCIATION

5. *rule 3 applied to correspondence 4:*

   S1.MANUFACTURERS–S1.produces–S1.ARTICLES,
   S2.PRODUCERS–S2.supplies

   path MANUFACTURERS–produces–ARTICLES
   already exists in the integrated schema

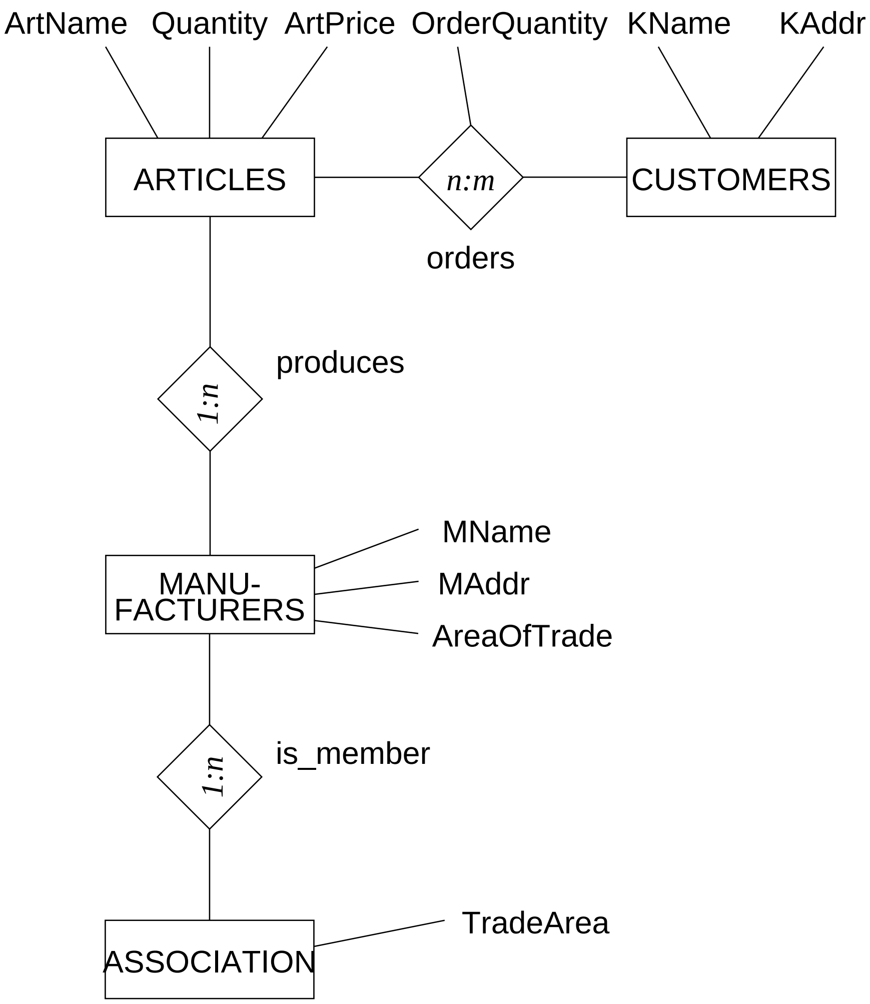
6. *rule 3 applied to correspondence 5:*

   S1.ARTICLES–S1.orders–S1.CUSTOMERS,
   S2.supplies–S2.CONSUMERS

   integrated path already exists, too

Example (V)

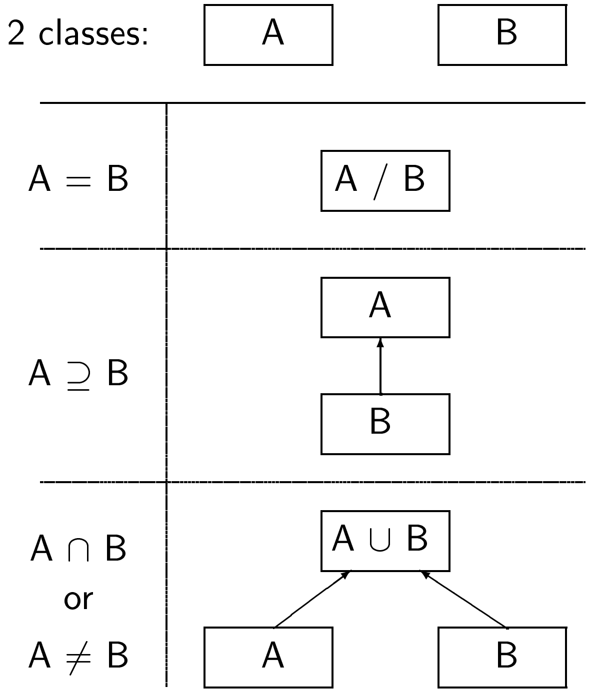
integrated Entity-Relationship schema



Upward Inheritance

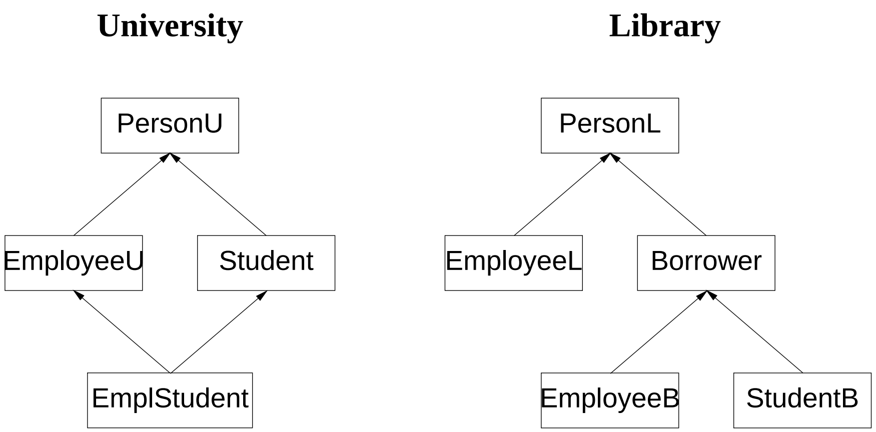
basic principle for integrating class hierarchies

given two classes A and B
for which a correspondence has been found:



Upward Inheritance: Example

2 class hierarchies which are to be integrated:



correspondences found by comparing the schemata and their "real-world semantics":

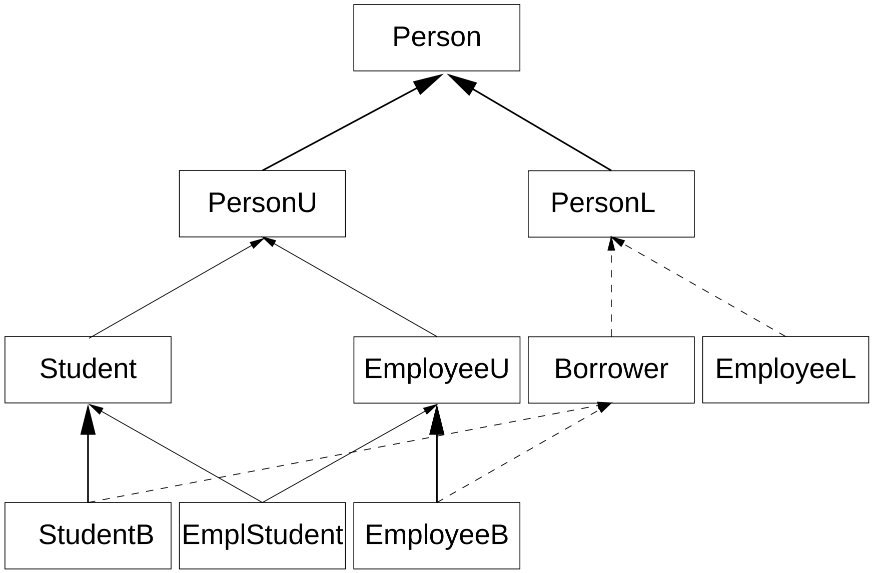
- PersonU $\cap$ PersonL
- EmployeeU $\supseteq$ EmployeeB
- Student $\supseteq$ StudentB

Upward Inheritance: Example (II)

result of integration using *upward inheritance*:

- new superclass Person for PersonU, PersonL
- EmployeeB becomes subclass of EmployeeU
- StudentB becomes subclass of Student

integrated class hierarchy:



Upward Inheritance: Example (III)

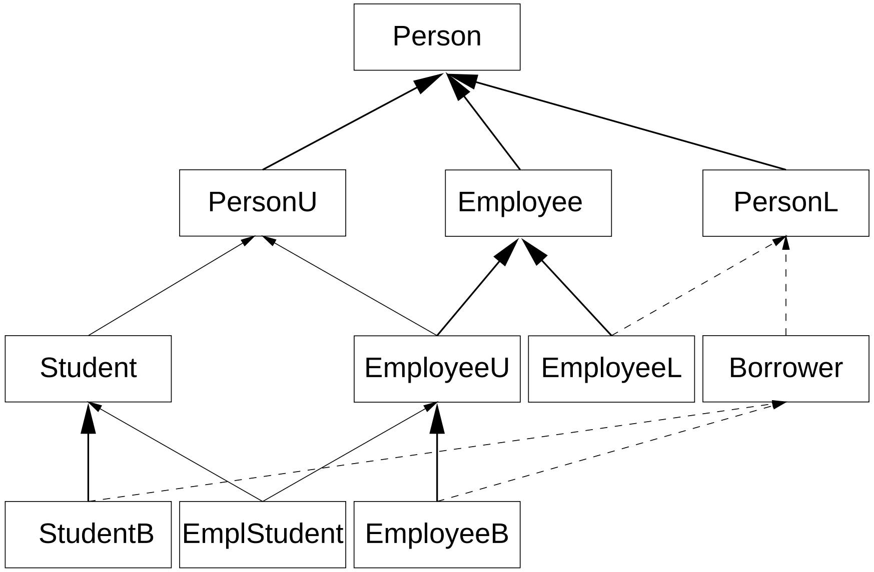
modification of the example:

additional correspondence

$$\text{EmployeeU} \cap \text{EmployeeL}$$

$\leadsto$ modified integrated schema:

- new superclass for EmployeeU, EmployeeL



GIM: Generic Integration Model

goal: methodology for deriving integrated schemata

problems:

- extensional overlappings
- intensional overlappings
- integration of inheritance/class hierarchies
- different integrated schemata possible

main ideas:

- decomposition and composition
- using a semantically poor "data model" (GIM) (only as an intermediate representation)

[Schmitt and Saake, 1996b], [Schmitt and Saake, 1996c], [Schmitt and Conrad, 1997], [Schmitt, 1998]

remark:
resolution of other kinds of integration conflicts not considered here

GIM: Generic Integration Model (II)

usage of GIM:

- transformation of local schemata into GIM
- resolving schema conflicts in GIM schemata
- deriving object-oriented integrated schemata from a GIM schema

main properties of GIM:

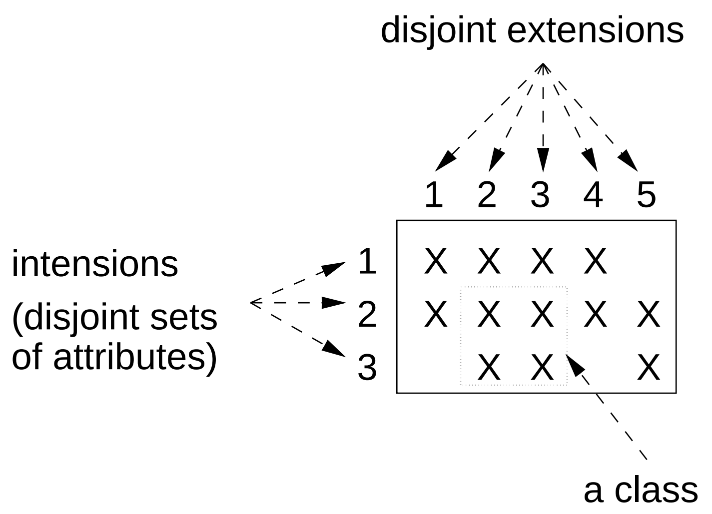
- classes with extension (set of *potential* objects) and intension (type)
- class extensions are either *disjoint* or *identical*

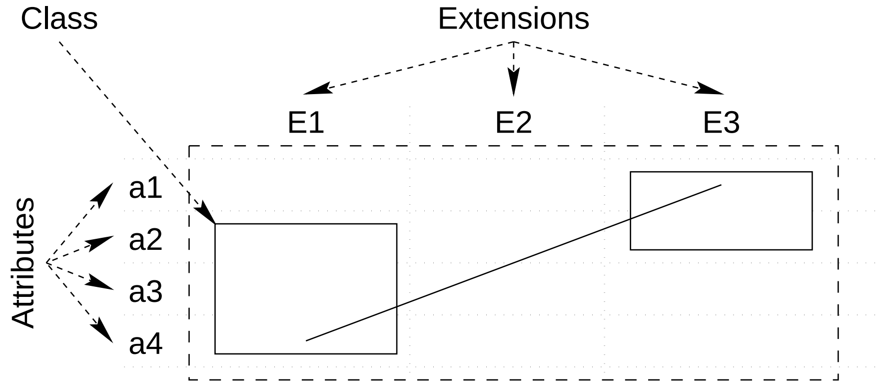
consequence:

- GIM cannot function as interface to applications
- main usage during federated database *design*, i.e. during the integration process

GIM: Generic Integration Model (III)

GIM notation:



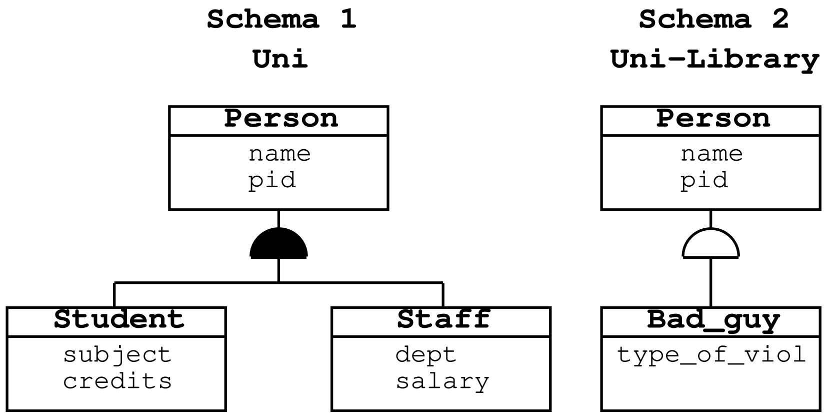


Homogenization in GIM

GIM as "canonical data model"
(for integration purposes)

→ necessary to transform local schemata into GIM representations

Example:

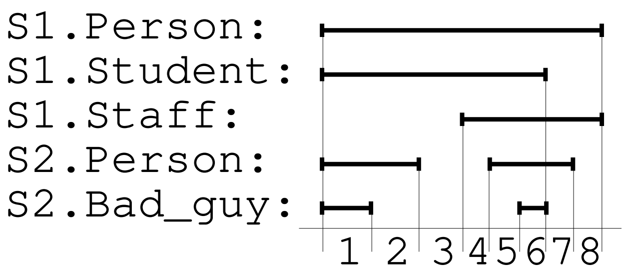


extensional conflicts:

→ extensional overlappings

→ requires decomposition

Homogenization in GIM (II)

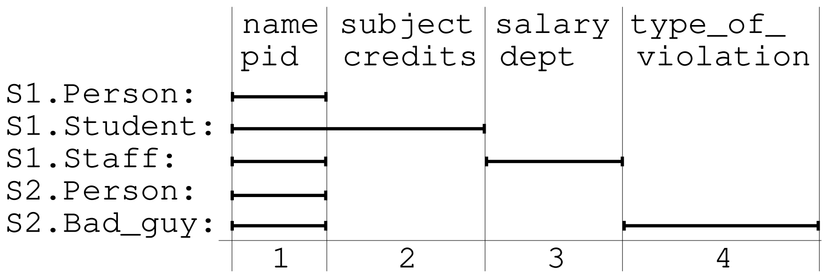


intensional conflicts:

→ resolving synonyms and homonyms

→ finding intensional overlappings

→ intensional decomposition (attribute clustering)



Homogenization in GIM (III)

resulting attribute partition:

1	name, pid
2	subject, credits
3	dept, salary
4	type_of_viol

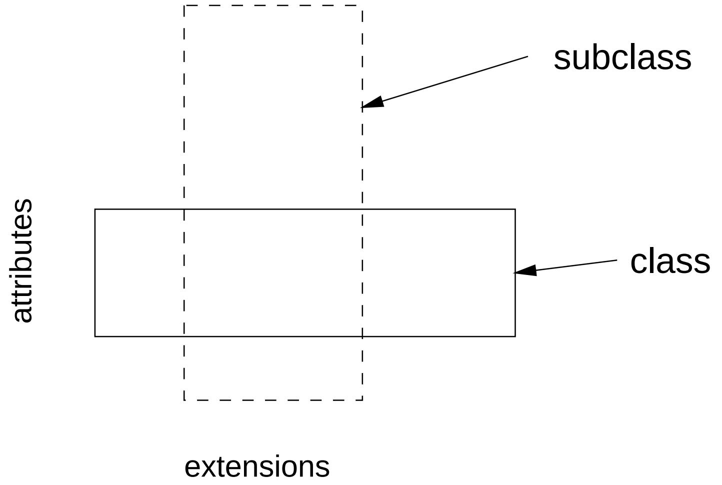
normalized GIM schema:

	1	2	3	4	5	6	7	8
1	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X		
3				X	X	X	X	X
4	X					X		

(filled) rectangles represent classes !

Classes and Subclasses in GIM

finding candidates for classes and subclasses in the integrated schema:



Schema	Class name	Extension	Intension
S1	Person	1,2,3,4,5,6,7,8	1
S1	Student	1,2,3,4,5,6	1,2
S1	Staff	4,5,6,7,8	1,3
S2	Person	1,2,5,6,7	1
S2	Bad_guy	1,6	1,4

Deriving an Integrated Schema

- starting with the normalized GIM schema (result of the homogenization phase)
- composing "GIM classes" to larger classes in a meaningful way
- finding (and using) subclass relationships

naive solution:

1. Generate all possible classes

i.e. compute all combinations of normalized classes and construct the corresponding schemata

2. Select the best schema

test all these "integrated schemata"
and take the best one

→ exponential complexity !!!

→ not practical for large schemata

Deriving an Integrated Schema (II)

a better algorithm:

presentation by means of the example introduced before

1. Intensional and extensional sorting

sort the normalized GIM schema by interpreting rows and column as binary numbers

first, sort the matrix w.r.t. rows (intensions)

then, sort the matrix w.r.t. columns (extensions)

repeat this step until no further changes occur

Deriving an Integrated Schema (III)

Example

	1	2	3	4	5	6	7	8
1	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X		
3				X	X	X	X	X
4	X					X		

↓ sorting rows

	1	2	3	4	5	6	7	8
1	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X		
4	X					X		
3				X	X	X	X	X

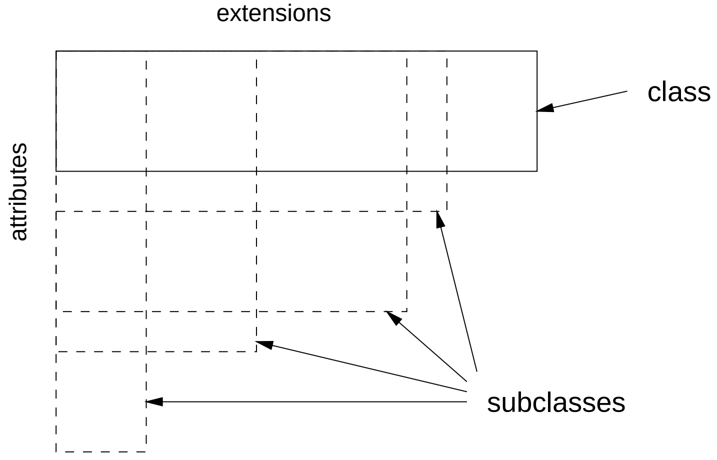
↓ sorting columns

	6	1	4	5	2	3	7	8
1	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X		
4	X	X						
3	X		X	X			X	X

Deriving an Integrated Schema (IV)

2. Deriving external classes

resolve the main triangle (upper left part of the matrix) into classes with subclass relationship (→ first part of the main class hierarchy)

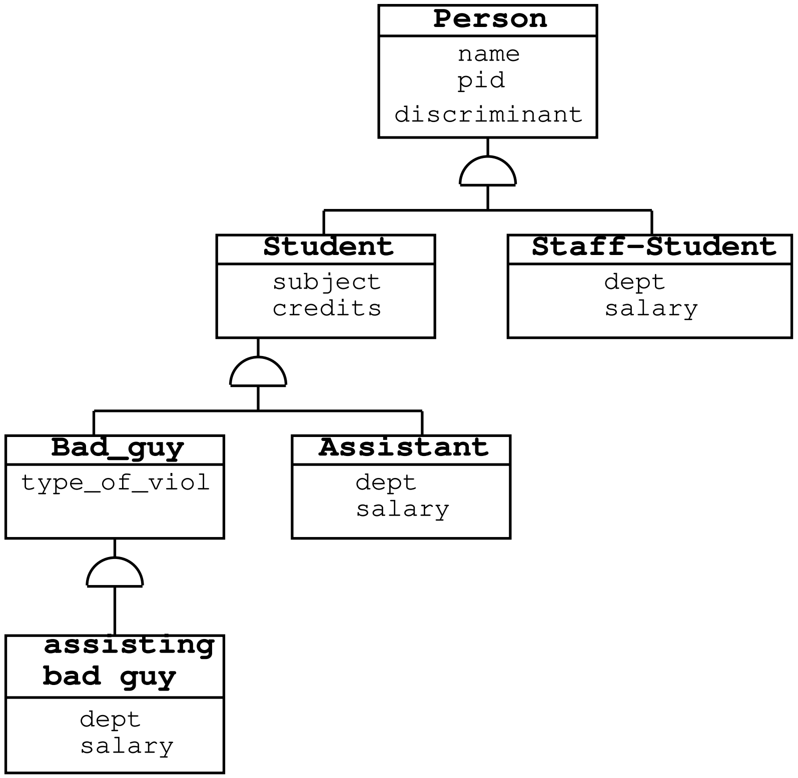


then, recursively apply the algorithm to all remaining non-empty sub-matrices and add their resulting class hierarchies as sub-hierarchies to the main hierarchy

Deriving an Integrated Schema (V)

Example:

	6	1	4	5	2	3	7	8
1	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X		
4	X	X						
3	X		X	X			X	X



Another Derivation for the Example

another starting order of the normalized GIM schema:

	4	5	6	7	8	1	2	3
1	X	X	X	X	X	X	X	X
2	X	X	X			X	X	X
3	X	X	X	X	X			
4			X			X		

sorting rows:

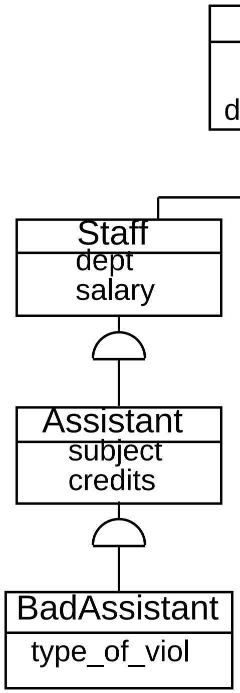
	4	5	6	7	8	1	2	3
1	X	X	X	X	X	X	X	X
3	X	X	X	X	X			
2	X	X	X			X	X	X
4			X			X		

sorting columns:

	6	5	4	7	8	1	2	3
1	X	X	X	X	X	X	X	X
3	X	X	X	X	X			
2	X	X	X			X	X	X
4	X					X		

Another Derivation for the Example (II)

resulting integrated schema:



$\rightarrow$ efficient algorithm !

$\rightarrow$ integrated schema without multiple inheritance

but: result depends on starting order !

Concept Lattices

formal concept analysis:

- area of applied mathematics (applied lattice theory)
- formalization of *concept* und *concept lattice*
- used to model human understanding of concepts
- various publications
  e.g., [Wille, 1992], [Ganter and Wille, 1996]

formal context: $K := (G, M, I)$

- G is set of entities of the context
- M is set of properties of the context
- $I \subseteq G \times M$ with gIm: entity g **has** property m

definition of two functions:

- for $A \subseteq G$: $A' := \{m \in M \mid \forall g \in A : (g, m) \in I\}$,
  i.e., set of common properties of entities of A
- for $B \subseteq M$: $B' := \{g \in G \mid \forall m \in B : (g, m) \in I\}$,
  i.e. set of entities having all properties of B
- Galois-connection between $\mathcal{P}(G)$ und $\mathcal{P}(M)$

Concept Lattices (II)

formal concept (A, B) **of context** $K := (G, M, I)$**:**

- $A \subseteq G,\ A' = B$
- $B \subseteq M,\ B' = A$
- corresponds to a maximal rectangle

concept lattice $\mathcal{B}(G, M, I)$**:**

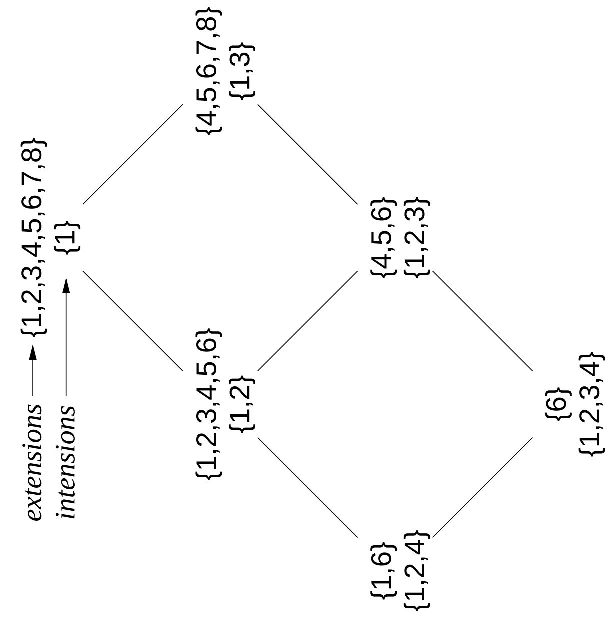
- contains all concepts of a context
- $(A_1, B_1) \leq (A_2, B_2) \Longleftrightarrow A_1 \leq A_2$ (partial order)
- infimum: (M', M)
- supremum: (G, G')
- computation cost: $O(2^n)$ due to the power set

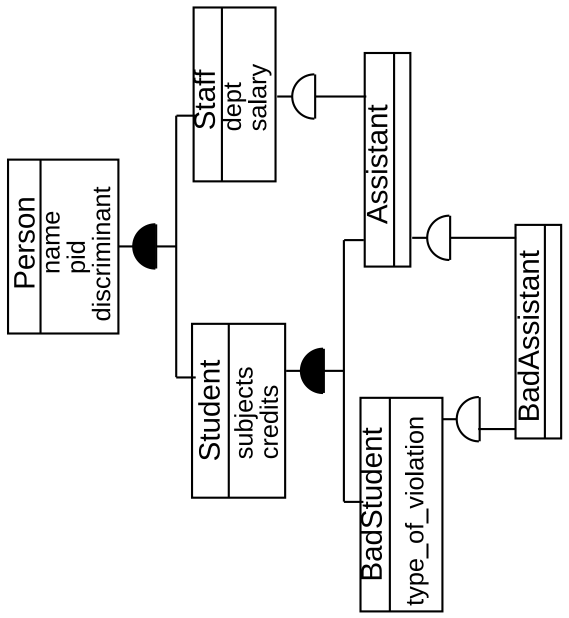
problems with computation of (general) concept lattices:

- cost is too high
- superfluous classes (concepts) are generated (neither extension nor intension of their own)

Concept Lattices (III)

applied to our previous example:

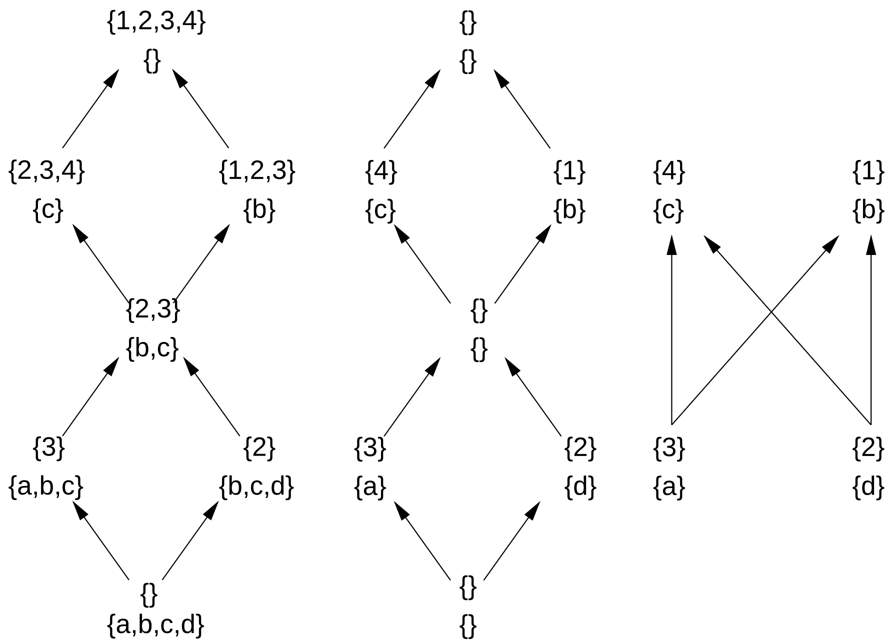




Superfluous Classes in Concept Lattices: Example

M/G	1	2	3	4
a			$\surd$	
b	$\surd$	$\surd$	$\surd$	
c		$\surd$	$\surd$	$\surd$
d		$\surd$		

results in:
(from left to right: reduction of useless classes)



Integration based on Concept Lattices

result:

- specialized algorithm for deriving hierarchies without superfluous classes

  complexity: O(n^3)
  (n: maximal number of rows and columns in matrix)

  [in contrast to O(2^n) for general concept lattices !]

- algorithm can be adapted to special requirements
  - null values instead of subclasses
  - disriminants for distinguishing merged classes
  - additional merging and splitting of classes (avoiding undesired multiple inheritance)

Selected References

"classical" database design: [Batini et al., 1992]

design of distributed databases: in [Ceri et al., 1987]

survey article on view integration and schema integration: [Batini et al., 1986]

brief survey on object-oriented integration approaches: in [Pitoura et al., 1995].

criteria for global data model: [Saltor et al., 1991]

classification of integration conflicts: [Kim and Seo, 1991], [Spaccapietra et al., 1992], [Kim et al., 1995]

assertion-based integration: [Spaccapietra et al., 1992], [Spaccapietra and Parent, 1994]

integration based on upward-inheritance: [Dayal and Hwang, 1984], [Schrefl and Neuhold, 1988]

GIM: [Schmitt, 1995], [Schmitt and Saake, 1996a], [Schmitt and Saake, 1996b], [Schmitt and Saake, 1996c], [Schmitt and Conrad, 1997], [Schmitt, 1998]

Selected References (II)

further articles on concrete approaches for view integration and schema integration: [Batini and Lenzerini, 1984], [Mannino and Effelsberg, 1984], [Larson et al., 1985], [Motro, 1987], [Neuhold and Schrefl, 1988], [Larson et al., 1989], [Fang et al., 1991], [Gellert et al., 1992], [Saltor et al., 1993], [Andersson et al., 1993], [Bright et al., 1994], [Dupont, 1994], [Reddy et al., 1994], [Reddy et al., 1995], [Garca-Solaco et al., 1995], [Benn et al., 1996].

resolution of structural conflicts: [Schrefl and Neuhold, 1990], [Krishnamurthy et al., 1991], [Hammer and McLeod, 1993], [Lee and Ling, 1995], [Missier and Rusinkiewicz, 1997].

schema transformation: [Papazoglou et al., 1990], [Tresch and Scholl, 1993], [Atzeni and Torlone, 1997]

semantic enrichment: [Castellanos and Saltor, 1991], [Fahrner and Vossen, 1995], [Hohenstein and Körner, 1995], [Hohenstein and Plesser, 1996]

object identity / identification: [Hammer et al., 1994], [Schmitt and Saake, 1995], [Chen, 1996]

4. Semantic Integrity

outline:

- integration of integrity constraints
- mechanisms for constraint enforcement in federated database systems

semantic integrity:

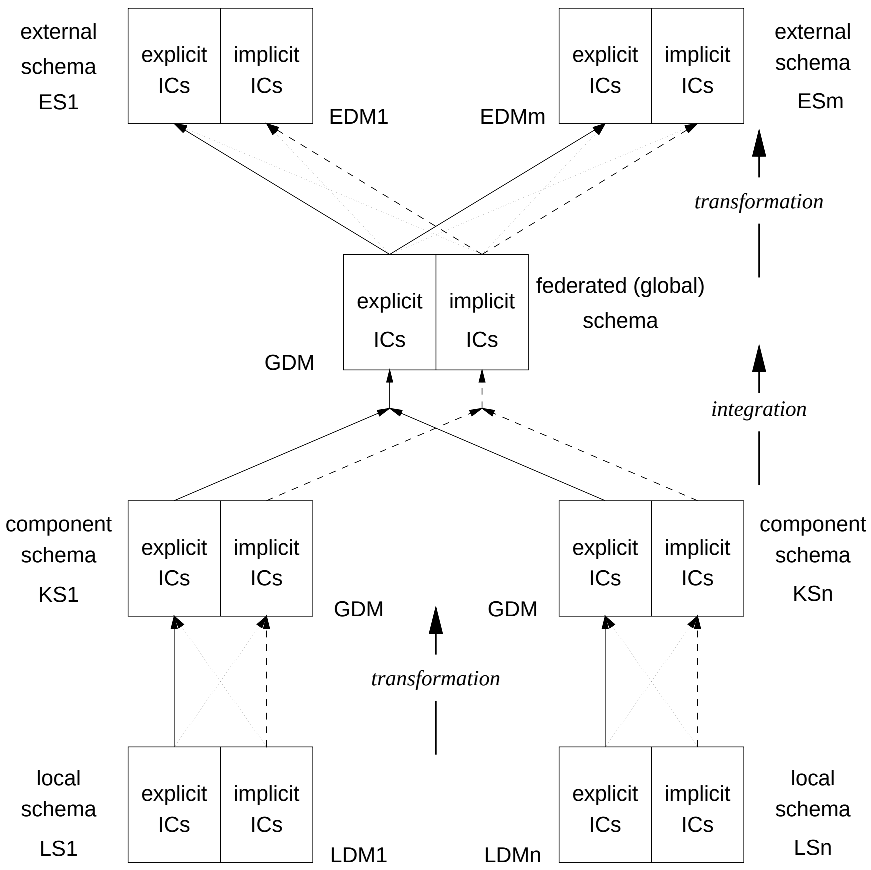
- description of consistent data by means of integrity constraints
- reaction on violation of integrity constraints
- local integrity vs. global integrity

global integrity:

- global integrity constraints:
  involving data of several component systems
- schema integration often causes global ICs
  e.g., semantic equivalence of classes
- global ICs as integrated local ICs

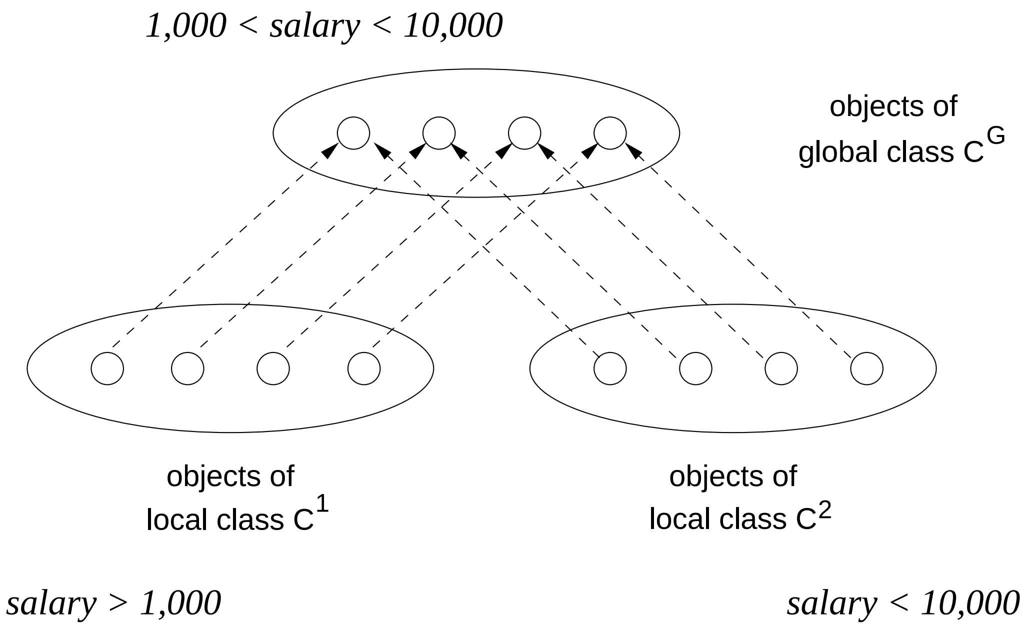
Schema Integration and Integrity Constraints

transformation and integration of ICs:



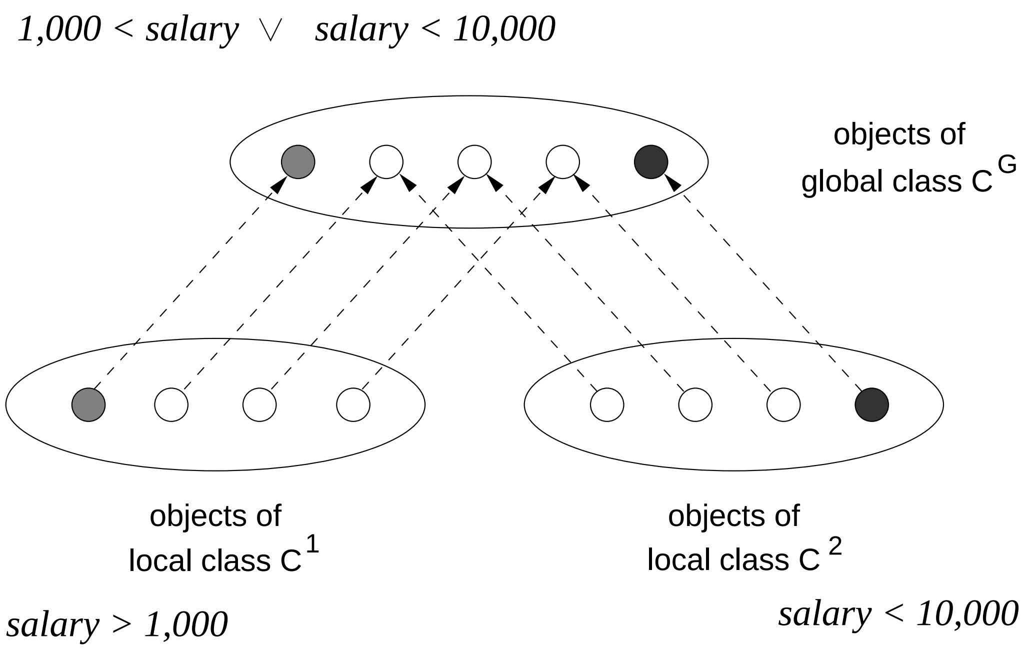
Integration of ICs: Example (I)

classes with semantically equivalent extensions:



Integration of ICs: Example (II)

semantically overlapping extensions:



Integration of ICs

- treatment of implicit ICs by schema integration approaches
- treatment of explicit ICs ???

classification of constraints:

- object constraints
- class constraints
  - uniqueness/key constraints (**unique**)
  - referential integrity constraints (**ref**)
  - constraints using aggregation functions (e.g., **min**, **max**, **count**)
  - ...
- ...

Integration Rules for ICs

some rules for dealing with constraints during schema transformation and schema integration:
(let φ, ψ be object constraints)

equivalence rule

$$\Theta_{\varphi}(E_1), \Theta_{\psi}(E_2), E_1 \equiv E_2 \Longrightarrow \Theta_{\varphi \wedge \psi}(E_1)$$

specialization rule

$$\Theta_{\varphi}(E_1 \cup E_2) \Longrightarrow \Theta_{\varphi}(E_1)$$

generalization rule

$$\Theta_{\varphi}(E_1), \Theta_{\psi}(E_2) \Longrightarrow \Theta_{\varphi \vee \psi}(E_1 \cup E_2)$$

intersection rule

$$\Theta_{\varphi}(E_1), \Theta_{\psi}(E_2) \Longrightarrow \Theta_{\varphi \wedge \psi}(E_1 \cap E_2)$$

... + general logical derivation rules

[Conrad et al., 1997c]

Applying Rules on Class Constraints

- uniqueness constraints (**unique**)

  → also valid for all subsets of the original class extension (weakening)

- referential integrity constraints within one class (**ref**)

  → valid only for supersets of the original class extension

- constraints using aggregation functions (e.g., **min**, **max**)

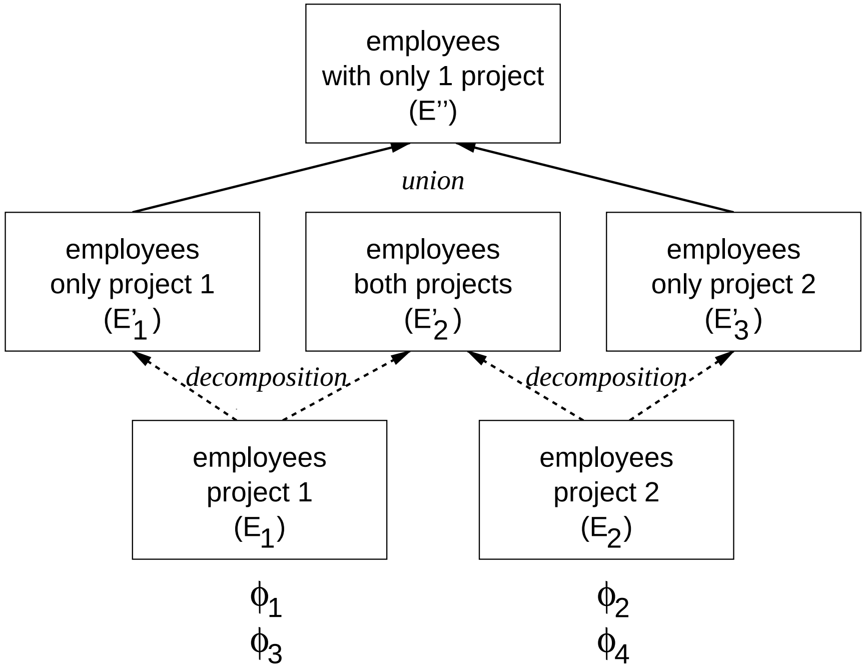
  notation: **min**(x) θ z
  with $\theta \in \{<, >, \leq, \geq, =, \neq\}$
  and constant z

  → validity depends on θ and z

Applying Rules to Class Constraints (II)

rule	**unique**	**ref**	**max$_\theta$**	**min$_\theta$**
equivalence	$\surd$	$\surd$	$\surd$	$\surd$
specialization	$\surd$	—	$\theta \in \{\leq, <\}$	$\theta \in \{\geq, >\}$
generalization	—	$\surd$	$\theta \in \{\geq, >, <, \leq\}$	$\theta \in \{\leq, <, \geq, >\}$
intersection	$\surd$	—	$\theta \in \{\leq, <\}$	$\theta \in \{\geq, >\}$

Example



$$E_1' = E_1 - E_2, \quad E_2' = E_1 \cap E_2, \quad E_3' = E_2 - E_1$$
$$E'' = E_1' \cup E_3'$$

$\phi_1 \equiv \forall e : e.\text{qualification} > 4$
$\phi_3 \equiv \forall e_1, e_2 : e_1.\text{pid} = e_2.\text{pid} \Longrightarrow e_1 = e_2$

$\phi_2 \equiv \forall e : e.\text{qualification} > 3$
$\phi_4 \equiv \exists e_1 : \forall e_2 : \quad e_1.\text{salary\_level} \geq e_2.\text{salary\_level}$
$\qquad \wedge\ e_1.\text{salary\_level} > 3$

Example (II)

transferring object constraints:

$$\Theta_{\phi_1}(E_1) \Longrightarrow \Theta_{\phi_1}(E_1' \cup E_2') \overset{spec.}{\Longrightarrow} \Theta_{\phi_1}(E_1')$$
$$\Theta_{\phi_1}(E_1) \Longrightarrow \Theta_{\phi_1}(E_2' \cup E_1') \overset{spec.}{\Longrightarrow} \Theta_{\phi_1}(E_2')$$
$$\Theta_{\phi_2}(E_2) \Longrightarrow \Theta_{\phi_2}(E_2' \cup E_3') \overset{spec.}{\Longrightarrow} \Theta_{\phi_2}(E_2')$$
$$\Theta_{\phi_2}(E_2) \Longrightarrow \Theta_{\phi_2}(E_3' \cup E_2') \overset{spec.}{\Longrightarrow} \Theta_{\phi_2}(E_3')$$

transferring the key constraint ϕ_3:

$$\Theta_{\phi_3}(E_1) \quad \Longrightarrow \quad \Theta_{\phi_3}(E_1' \cup E_2')$$
$$\Theta_{\phi_3}(E_1' \cup E_2') \quad \overset{spec.}{\Longrightarrow} \quad \Theta_{\phi_3}(E_1')$$
$$\Theta_{\phi_3}(E_2' \cup E_1') \quad \overset{spec.}{\Longrightarrow} \quad \Theta_{\phi_3}(E_2')$$

transferring the max constraint ϕ_4:

$$\Theta_{\phi_4}(E_2) \Longrightarrow \Theta_{\phi_4}(E_2' \cup E_3')$$

Example (III)

combining integrity constraints:

$$\Theta_{\phi_1}(E'_1), \Theta_{\phi_3}(E'_1) \overset{equiv.}{\Longrightarrow} \Theta_{\phi_1 \wedge \phi_3}(E'_1)$$

$$\begin{aligned} \Theta_{\phi_1}(E'_2), \Theta_{\phi_2}(E'_2), \Theta_{\phi_3}(E'_2) &\overset{equiv.}{\Longrightarrow} \Theta_{\phi_1 \wedge \phi_2 \wedge \phi_3}(E'_2) \\ &\Longrightarrow \Theta_{\phi_1 \wedge \phi_3}(E'_2) \end{aligned}$$

union, and logical simplification:

$$\begin{aligned} \Theta_{\phi_1 \wedge \phi_3}(E'_1), \Theta_{\phi_2}(E'_3) &\overset{gen.}{\Longrightarrow} \Theta_{(\phi_1 \wedge \phi_3) \vee \phi_2}(E'_1 \cup E'_3) \\ &\Longrightarrow \Theta_{\phi_2}(E'') \end{aligned}$$

→ *problem:* undesired weakening !!!

Introducing Discriminants

introduction of a discriminant for characterizing the origin of each object:

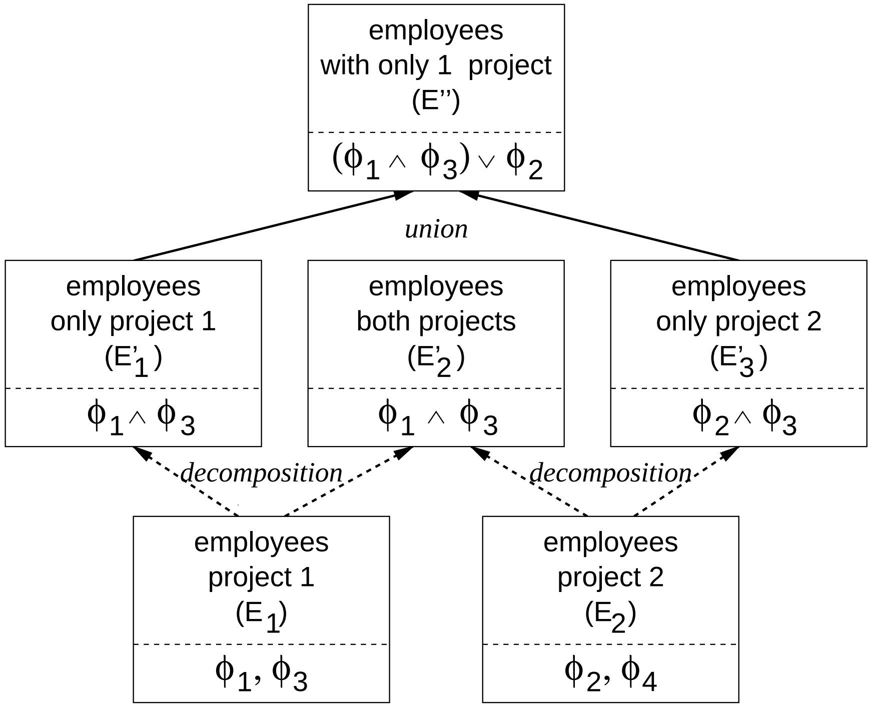
$$\begin{aligned} \Theta_{\phi'_1 \wedge \phi'_3}(E'_1) \quad with \quad & \phi'_1 \equiv \forall e : e.\text{project} = 1 \rightarrow e.\text{qualification} > 4 \\ & \phi'_3 \equiv \forall e_1, e_2 : \; (e_1.\text{project} = 1 \wedge e_2.\text{project} = 1) \\ & \qquad\qquad \rightarrow (e_1.\text{pid} = e_2.\text{pid} \rightarrow e_1 = e_2) \end{aligned}$$

$$\Theta_{\phi'_2}(E'_3) \quad with \quad \phi'_2 \equiv \forall e : e.\text{project} = 2 \rightarrow e.\text{qualification} > 3$$

$$\Theta_{\phi'_1 \wedge \phi'_3}(E'_1), \Theta_{\phi'_2}(E'_3) \Longrightarrow \Theta_{\phi'_1 \wedge \phi'_2 \wedge \phi'_3}(E'_1 \cup E'_3) \Longrightarrow \Theta_{\phi'_1 \wedge \phi'_2 \wedge \phi'_3}(E'')$$

→ avoiding undesired logical weakening of constraints

Final Result



and additionally: $\Theta_{\phi_4}(E'_2 \vee E'_3)$

Possible Reactions to Violation of ICs

reasons for violations of (global) ICs:

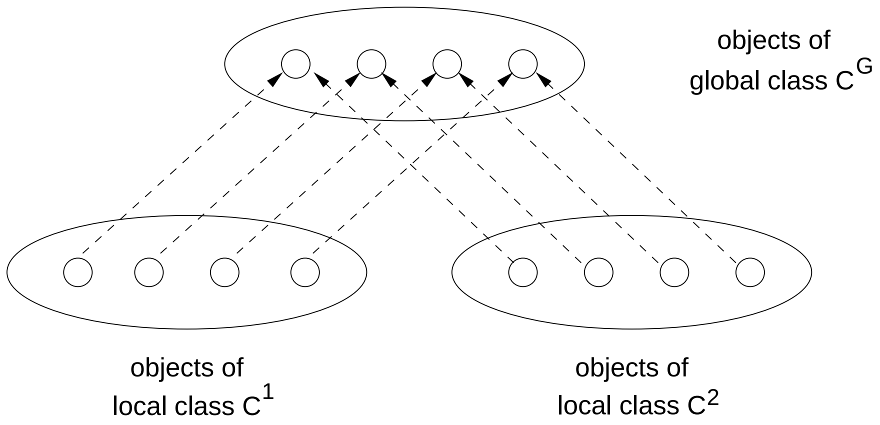
- insertion or deletion of objects,
- update of properties (attribute values)

possible reactions in case of violations:

- reject the operation
- repair by propagating operation (if possible)

Insertion or Deletion of Objects

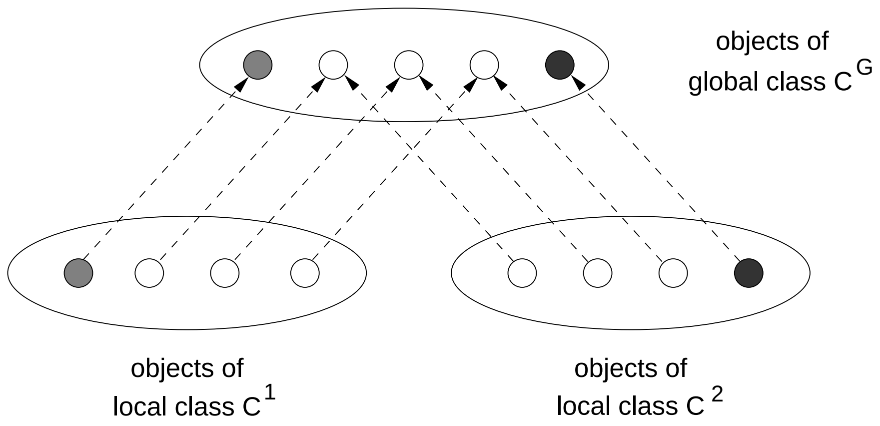
(A) semantically equivalent extensions:



- global insertion or deletion:
  propagate to all component systems
- local insertion or deletion:
  propagate operation to federation layer
  $\rightarrow$ global insertion or deletion

Insertion or Deletion of Objects (II)

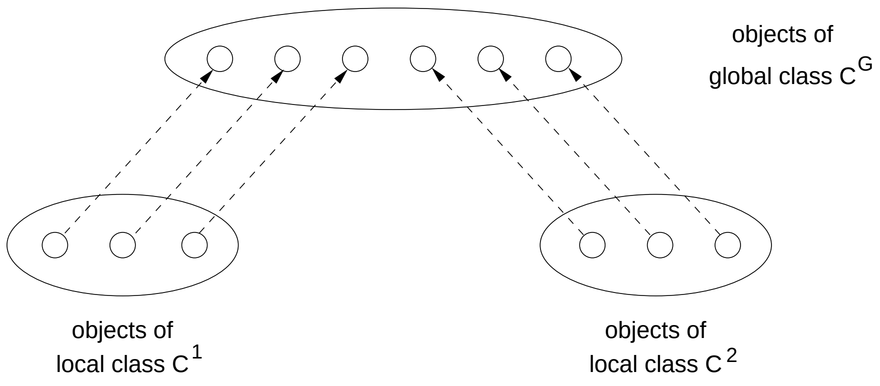
(B) semantically overlapping extensions:



- global insertion:
  propagate to one or more component systems
- local insertion:
  propagate to federation layer;
  if global object already exists, bo further propagation
- global deletion:
  propagate to all component systems where this object exists
- local deletion:
  - propagate to federation layer ($\rightarrow$ global deletion)
  - migrate object to another component system
    (if it is not already there)

Insertion or Deletion of Objects (III)

(C) disjoint local extensions:



- global insertion:
  propagate insertion to only one component system
- local insertion:
  - if no corresponding object exists: insert locally and globally
  - otherwise: migrate object from another component system into this one (if possible)
- global deletion:
  propagate to that component system where the object is stored
- local deletion:
  - delete locally and globally
  - migrate object to another component system

Insertion or Deletion of Objects (IV)

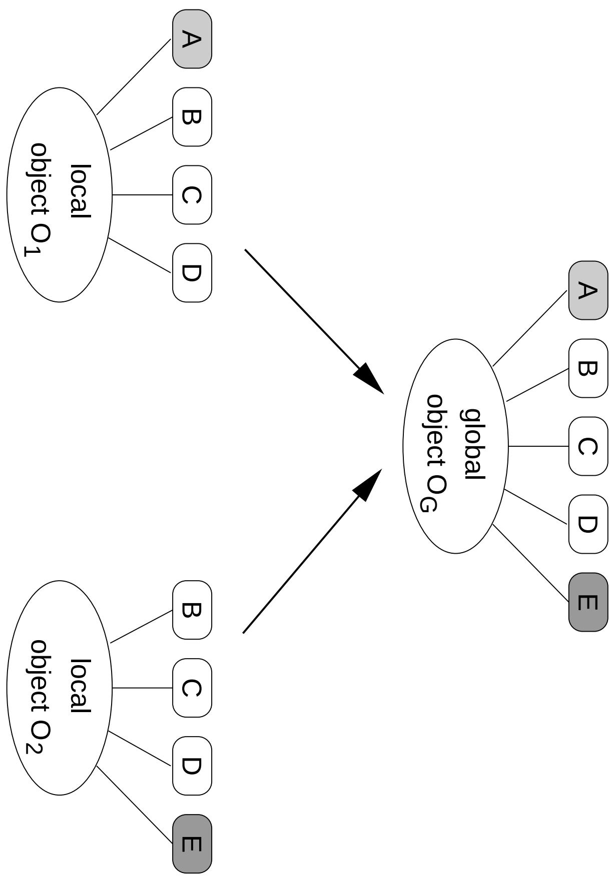
	insertion	
	global	local
semantically equivalent extensions	insert into all local extensions	propagate insertion to all other local extensions
semantically overlapping extensions	insert into one or more local extensions	propagate insertion to other local extensions (possibly)
semantically disjoint extensions	insert into only one local extension	check whether the object is already member of another local extension (then migrate, if possible)

Insertion or Deletion of Objects (V)

	deletion	
	global	local
semantically equivalent extensions	delete from all local extensions	propagate deletion to all other local extensions
semantically overlapping extensions	delete from all local extensions (containing this object)	propagate to all other local extensions (→ global deletion); *or* delete locally without propagation; *or* delete locally and insert into another local extension (→ migration)
semantically disjoint extensions	delete from that local extension which contains the object	delete locally; *or* delete locally and insert into another local extension (→ migration)

Updates

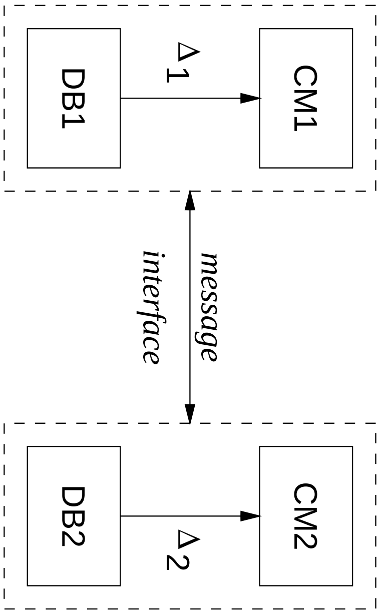
semantically equivalent local and global objects:



- global update
- local update

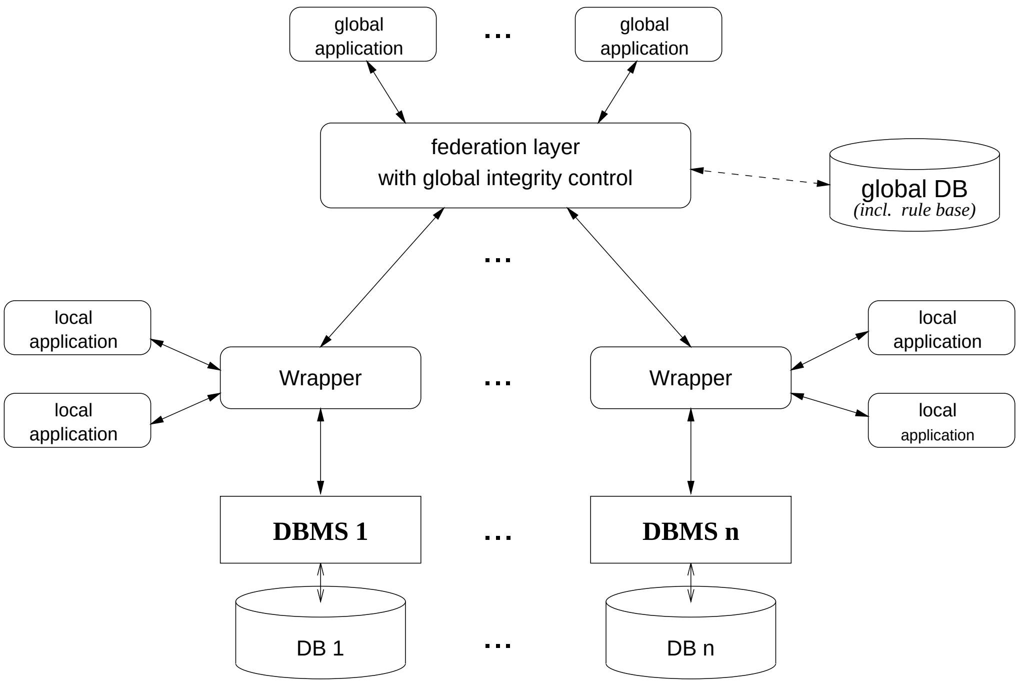
Mechanisms for Ensuring Integrity

architecture with Constraint Manager for protocol-based integrity enforcement:
[Grefen and Widom, 1995, Grefen and Widom, 1996]

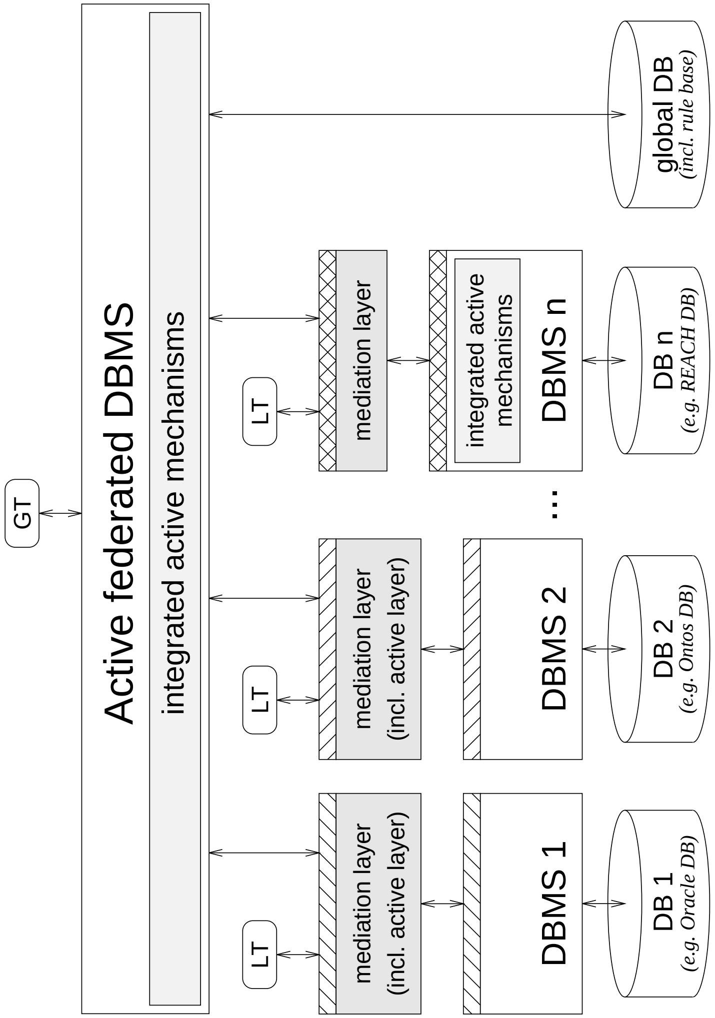


→ different protocols proposed

Extended Architecture for Global Integrity Control



Architecture of Active Federated Database Systems



Selected References

dealing with integrity constraints during schema integration:
[Conrad et al., 1997b], [Conrad et al., 1997c]

for certain kinds of integrity constraints see also: [Reddy et al., 1995]

protocols for integrity control:
[Barbará and Garcia-Molina, 1992],
[Barbará-Millá and Garcia-Molina, 1994],
[Grefen and Widom, 1996]

active mechanisms for integrity control in federated systems:
[Do and Drew, 1995], [Bültzingsloewen et al., 1996],
[Türker and Conrad, 1997],
[Spiliopoulou and Conrad, 1997]

consistency of replicated data:
[Mukhopadhyay and Thomas, 1993],
[Seligman and Kerschberg, 1993], [Sidell et al., 1996]

integrity violation by asynchronous processing of updates:
[Barghouti and Kaiser, 1991]

5. Transactions in Federated Database Systems

topics:

- problems caused by autonomy of component systems
- special transaction models

classical transactions:

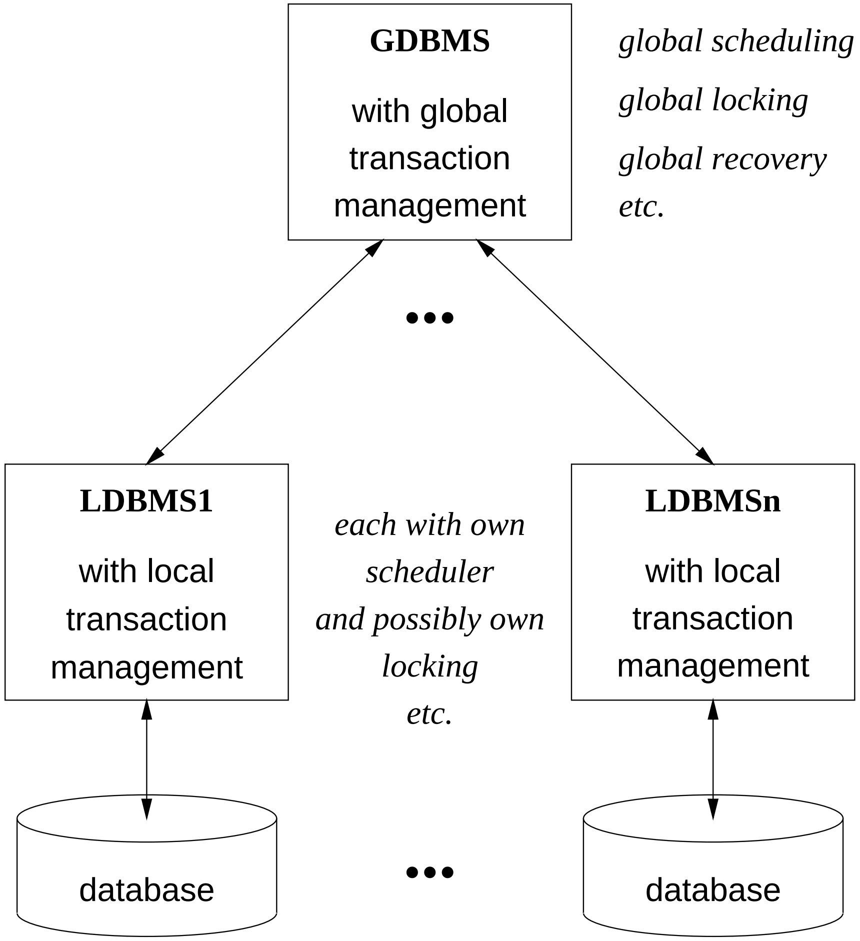
ACID properties:

- atomicity
- consistency
- isolation
- durability

correctness criteria:

- serializability
- avoiding cascading aborts
- etc

Transactions in Heterogeneous Environments

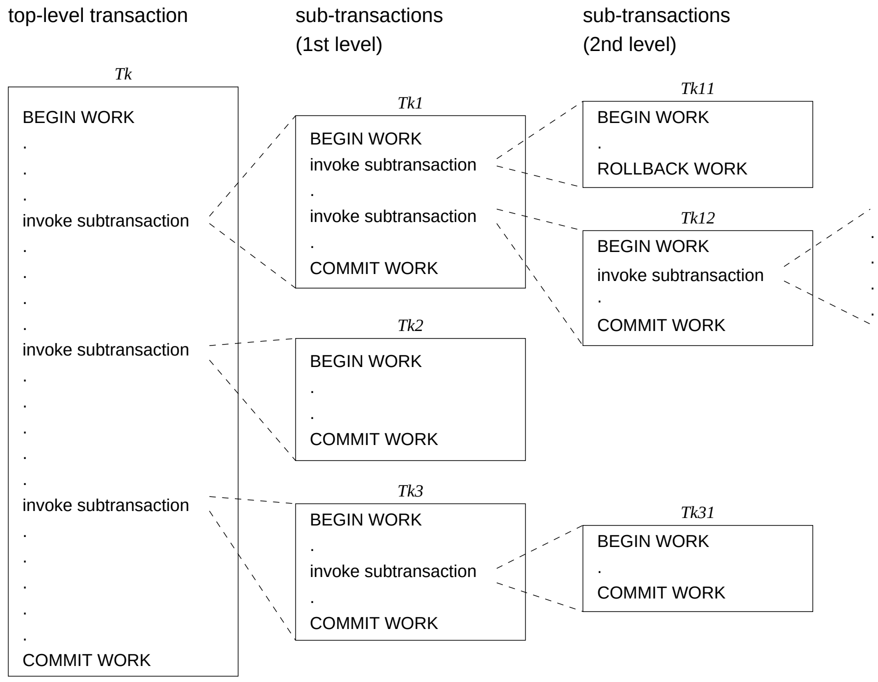


Advanced Transaction Models

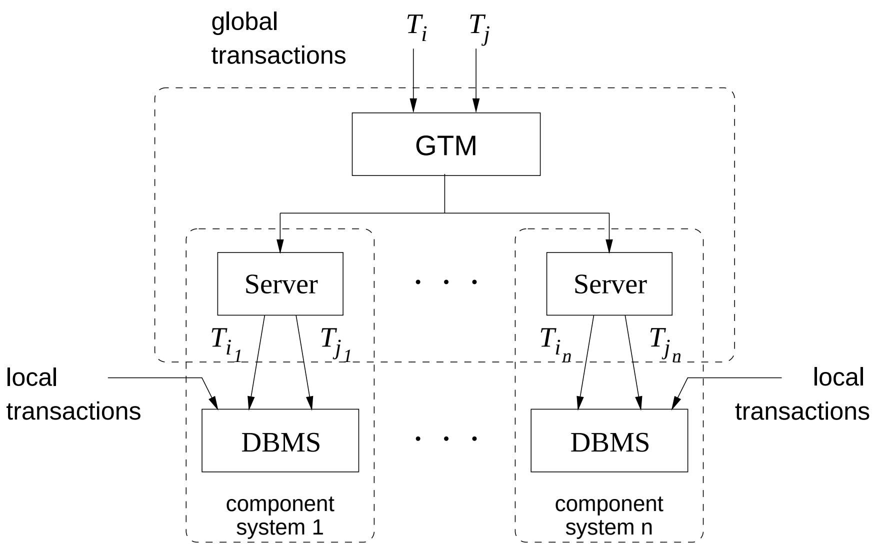
- two-phase commit
- nested transactions
  - parent / child relation
  - open vs. closed subtransactions
  - contingency transactions
  - vital vs. non-vital subtransactions
- transaction compensation
- transactional workflow

cf. overview in [Elmagarmid, 1992]

Nested Transactions



Transactions in Federated Systems



Problems with Autonomous Components

autonomy of component systems

$\rightarrow$ classical transaction processing is impossible

$\rightsquigarrow$ ACID is not guaranteed

specific problems:

- *global serializability*
- *global atomicity*
- *global deadlocks*

Global Serializability: The Problem

- different *concurrency control* algorithms in component systems
- encapsulated control information
- different serialization orders in different systems

example:
two component systems CS_1 (objects a and b) and CS_2 (objects c and d)

global transactions:

$$\mathrm{GT}_1: \quad r_1(a)\ r_1(c)$$
$$\mathrm{GT}_2: \quad r_2(b)\ r_2(d)$$

two local transactions:

$$\mathrm{LT}_3: \quad w_3(a)\ w_3(b)$$
$$\mathrm{LT}_4: \quad w_4(c)\ w_4(d)$$

Global Serializability: Global Example Schedule

GT1		r1(a)											r1(c)	
GT2							r2(b)		r2(d)					
LT3				w3(a)	w3(b)									
LT4	w4(c)										w4(d)			
S1		r1(a)	c1	w3(a)	w3(b)	c3	r2(b)	c2						
S2	w4(c)								r2(d)	c2	w4(d)	c4	r1(c)	c1

Global Serializability: Local Concurrency Control

local serialization orders:

$$\text{CS}_1: \quad \text{GT}_1 \rightarrow \text{LT}_3 \rightarrow \text{GT}_2$$
$$\text{CS}_2: \quad \text{GT}_2 \rightarrow \text{LT}_4 \rightarrow \text{GT}_1$$

possible solution:

new correctness criteria [Zhang and Elmagarmid, 1993]

- *chain conflict serializability* (ticket algorithms)
- *shared serializability*
- *hybrid serializability*

Global Atomicity: Example

example:
two component systems CS_1 (object a) and CS_2 (object c)

global transaction GT_1:

$$\text{GT}_1: \quad r_1(a)\ w_1(a)\ w_1(c)$$

1. global request for commit
2. successful commit in CS_2, but abort in CS_1 (autonomy!)
3. local transaction LT_2 in CS_1:

$$\text{LT}_2: \quad r_2(a)\ w_2(a)$$

4. resubmission of $w_1(a)$ to CS_1 (due to abort)
   $\rightarrow$ new local transaction in CS_1: $w_3(a)$, c_3

$\rightarrow$ incorrect global schedule!

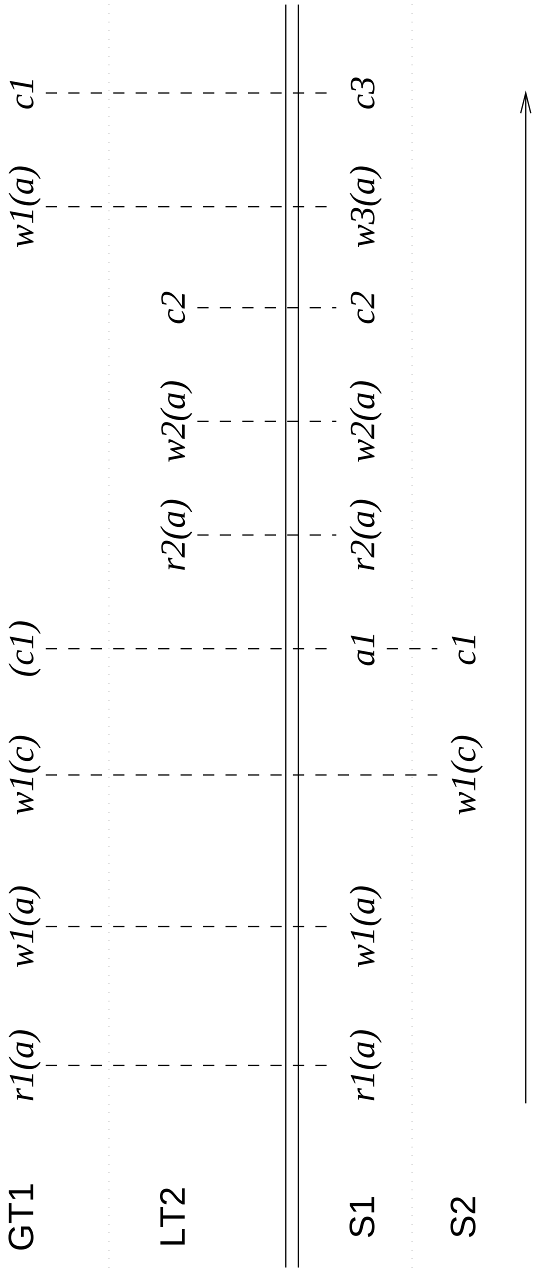
local view:

$$\text{S}_1: r_2(a)\ w_2(a)\ c_2\ w_3(a)\ c_3$$

global view:

$$r_1(a)\ r_2(a)\ w_2(a)\ c_2\ w_3(a)\ c_1$$

Global Atomicity: Example Schedule



Global Deadlocks: Example

global deadlocks caused by local transactions

example:
two component systems CS_1 (objects a and b) and CS_2 (objects c and d)

global transactions:

$$\mathrm{GT}_1: \quad r_1(a)\ r_1(d)$$
$$\mathrm{GT}_2: \quad r_2(c)\ r_2(b)$$

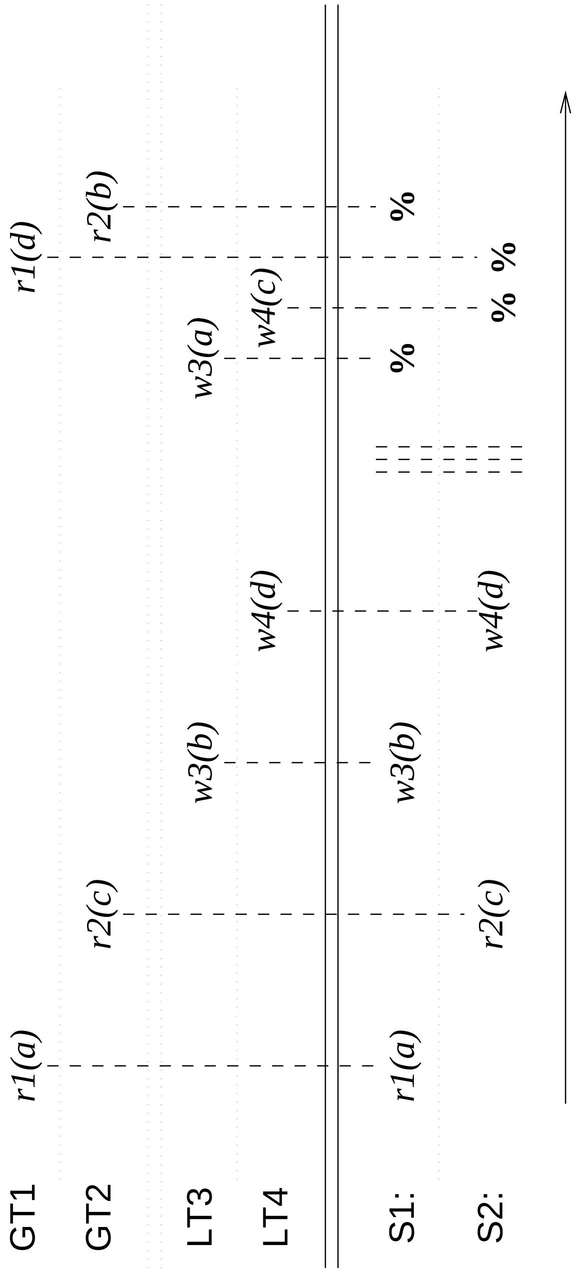
two local transactions:

$$\mathrm{LT}_3: \quad w_3(b)\ w_3(a)$$
$$\mathrm{LT}_4: \quad w_4(d)\ w_4(c)$$

possible situation:

- LT_3 waits for GT_1 (lock on a)
- LT_4 waits for GT_2 (lock on c)
- GT_1 waits for LT_4 (lock on d)
- GT_2 waits for LT_3 (lock on b)

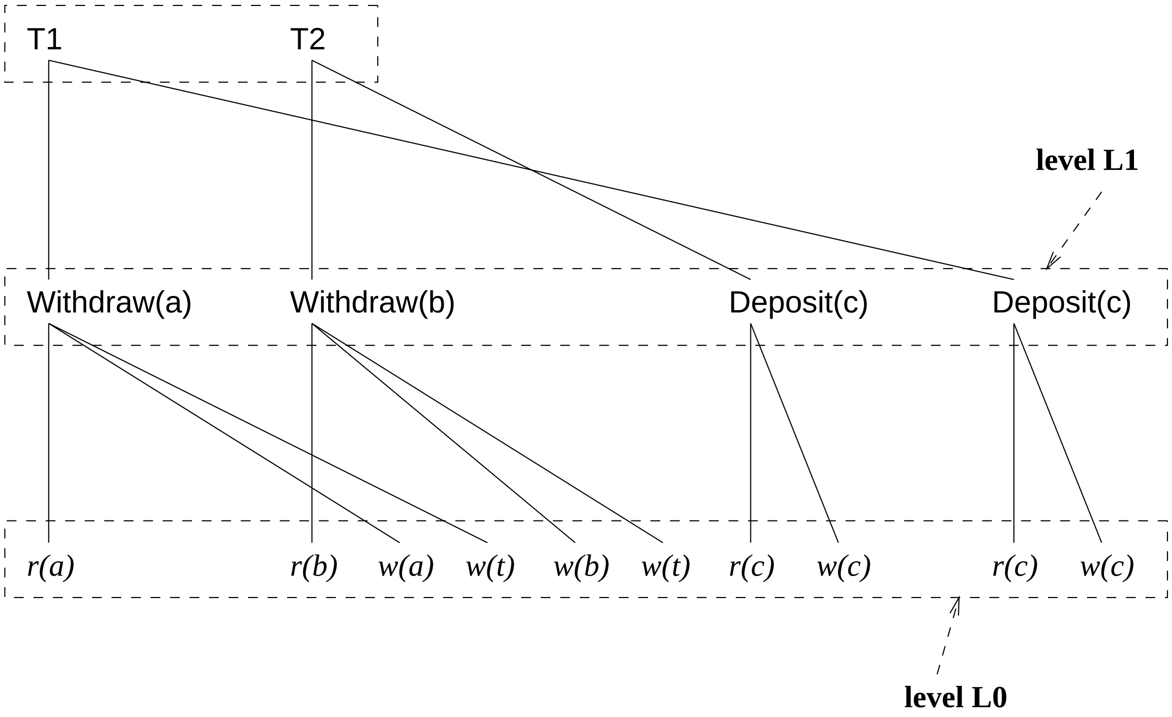
Global Deadlocks: Example Schedule



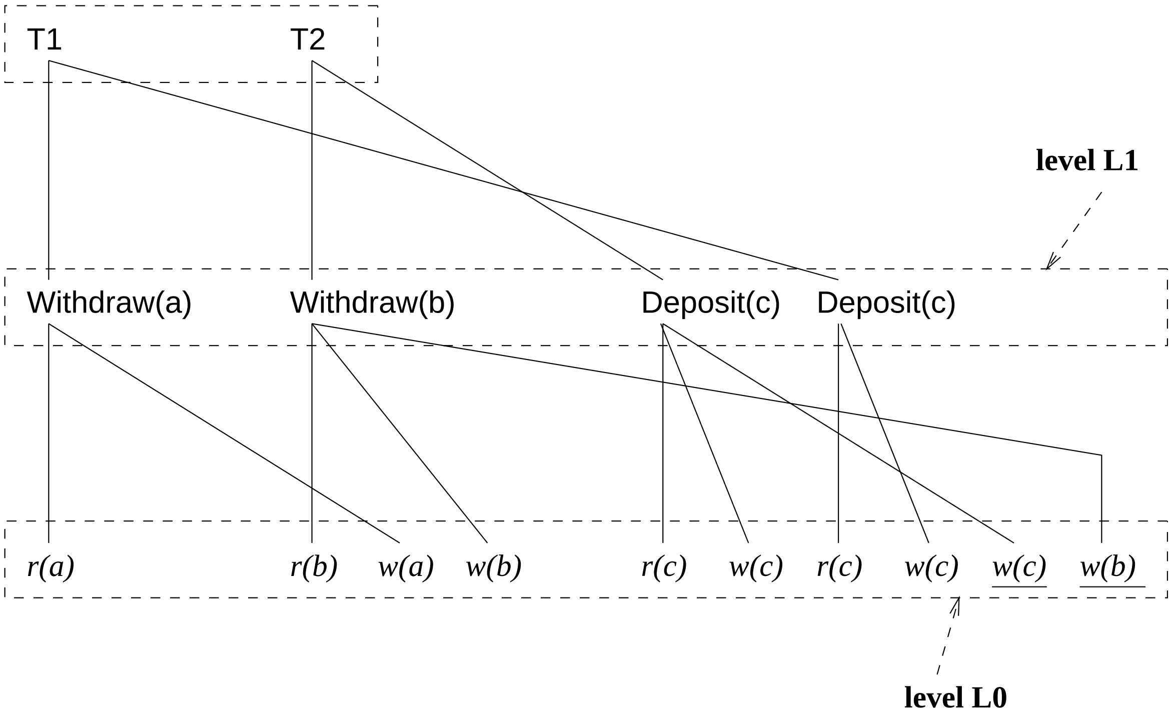
Specific Transaction Models

- multi-level transactions
  - different levels of nesting
  - semantic operations instead of read/write (on higher levels)
- ConTract model
  - processes are atomic transactions $\rightarrow$ workflow support
  - control structures
  - compensating transactions

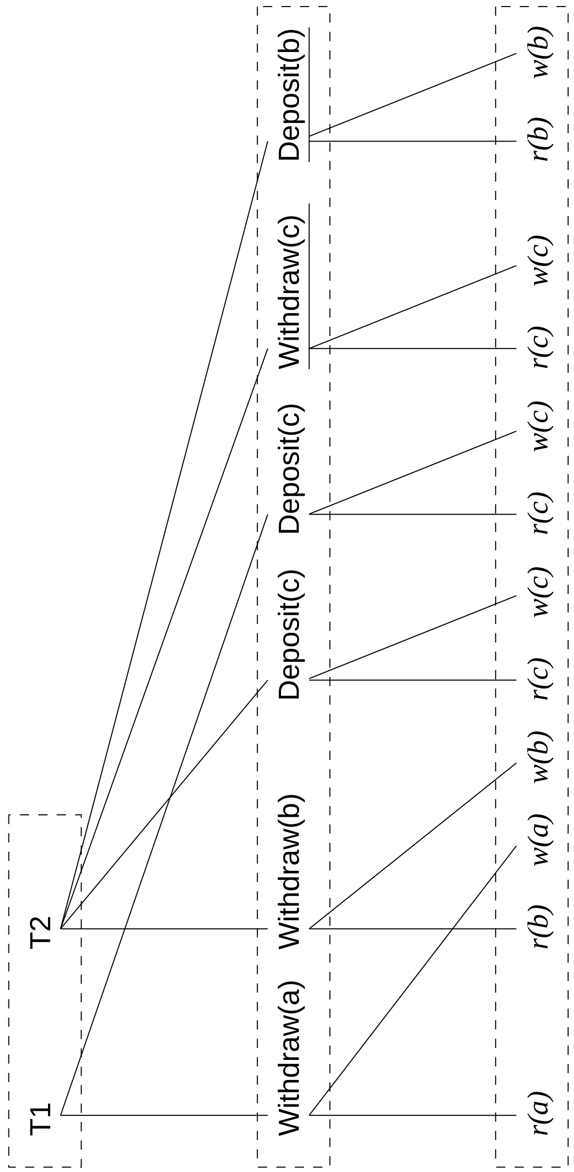
Multi-level Transactions



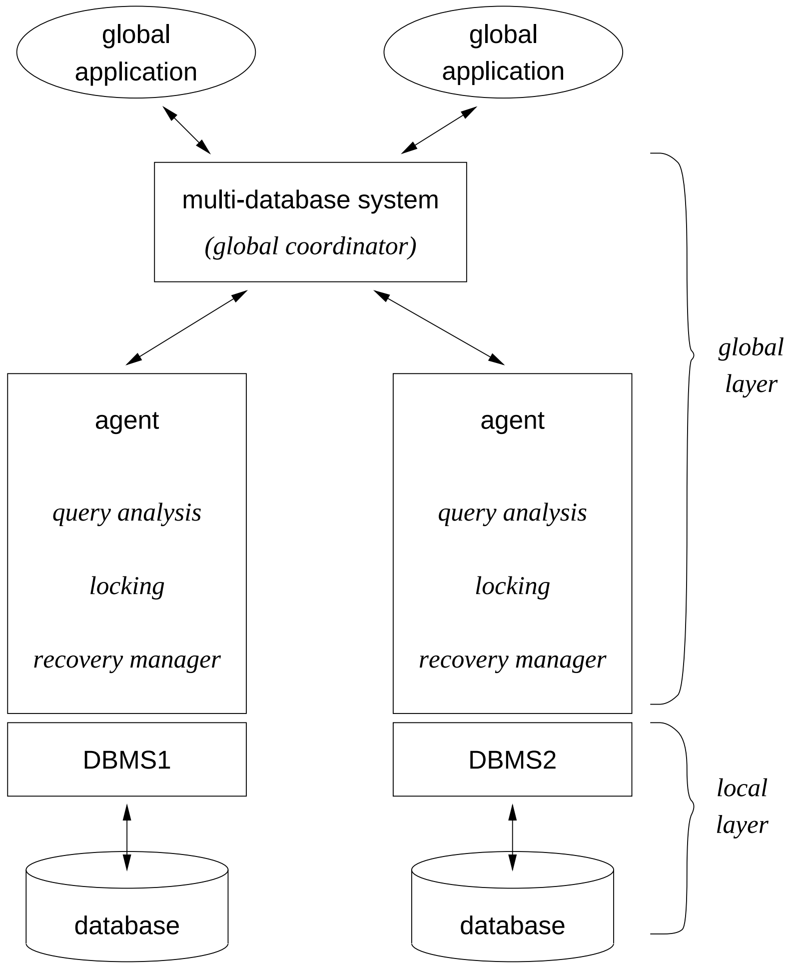
Multi-level Transactions: Non-serializable Schedule



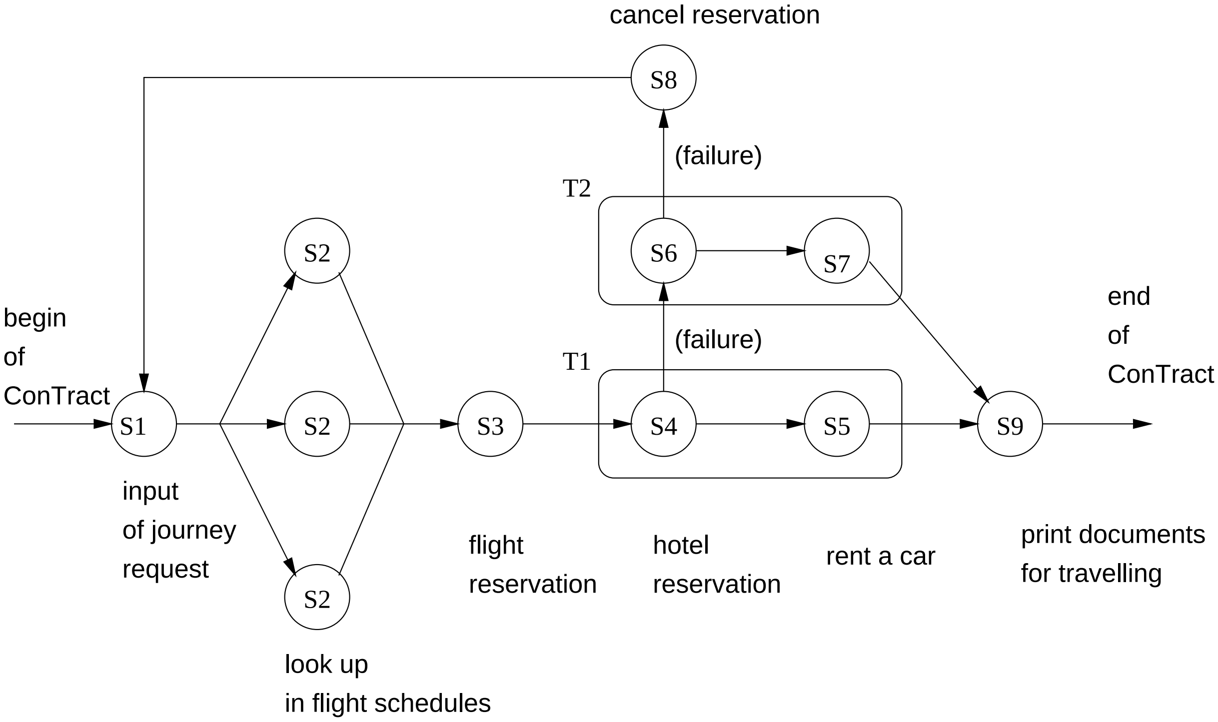
Multi-level Transactions: Serializable Schedule



Multi-Level Transaction Architecture

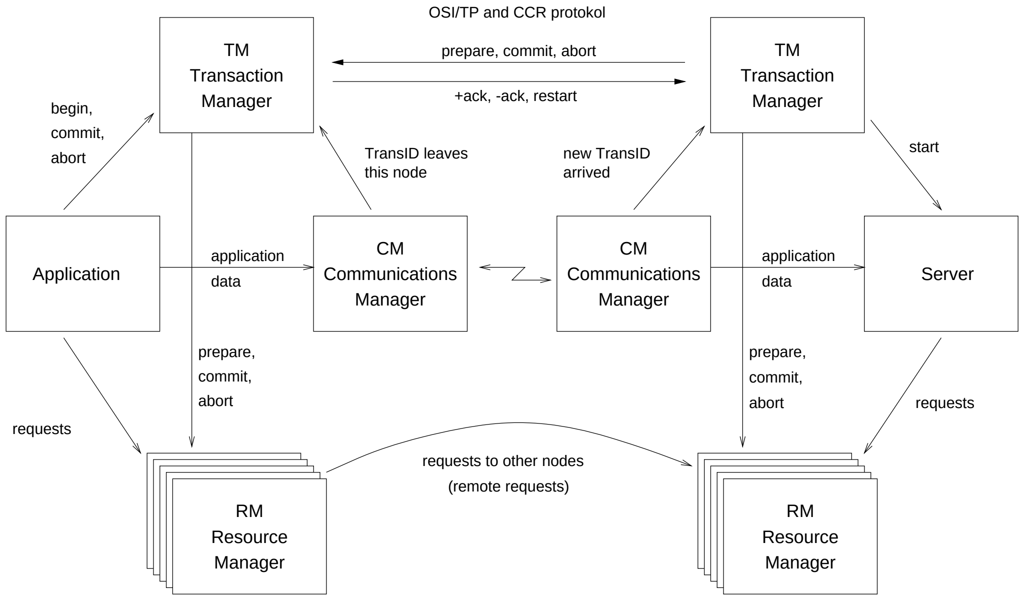


ConTracts



X/OpenDTP

standard for distributed transaction processing:



Selected References

textbooks on transaction management: [Bernstein et al., 1987], [Gray and Reuter, 1993]

problems of transaction management in federated database systems: [Breitbart et al., 1992a], [Breitbart et al., 1995] sketched in [Pitoura et al., 1995]

surveys on transaction management approaches for federated database systems: [Ramamritham and Chrysanthis, 1996]

nested transactions: [Moss, 1981], [Moss, 1985]

some recent approaches based on open nested transactions: [Buchmann et al., 1992], [Muth et al., 1992], [Weikum and Schek, 1992], [Wächter and Reuter, 1992]

Selected References (II)

multi-level transactions (foundations): [Weikum, 1991], [Schek et al., 1991], [Weikum and Schek, 1992], [Schaad and Schek, 1993]

multi-level transactions in federated database systems: [Deacon et al., 1994], [Schaad et al., 1995], [Schaad, 1996]

ConTracts [Wächter and Reuter, 1990], [Wächter and Reuter, 1992]

special approaches to global atomicity: [Muth and Rakow, 1991], [Breitbart et al., 1992b], [Mullen et al., 1992] [Mehrotra et al., 1992a];
. . . to global serializability: tickets [Georgakopoulos et al., 1991], [Georgakopoulos et al., 1994]; [Du and Elmagarmid, 1989], [Elmagarmid and Du, 1990], [Wolski and Veijalainen, 1990], [Barghouti and Kaiser, 1991], [Mehrotra et al., 1992b], [Mills, 1992], [Agrawal et al., 1993], [Hwang et al., 1993], [Zhang and Elmagarmid, 1993], [Zhang and Jing, 1993]

further special aspects of global transaction management: [Breitbart and Silberschatz, 1988], [Breitbart et al., 1991], [Alonso et al., 1994]

7. Additional Topics

- multi-database languages
- query processing and optimization
- integration on instance level
- security in federated systems
- gateways and database coupling

Multi-Database Languages

- local databases schemata are visible
- user may define their own external schema
- → explicit access to several databases in queries

multi-database systems have no global schema in contrast to federated databases!

MSQL

multi-database language MSQL

- MRDS
  (Multics Relational Data Store Multi-database)

  [Litwin and Abdellatif, 1986, Litwin, 1988, Litwin et al., 1990, Grant et al., 1993]

- SQL extension for relational multi-databases

MSQL versus SQL

extensions to SQL:

- composition of multi-database from component databases

  multi-database defines scope for queries / updates

- DDL for multi-databases:
  - exclusion of component relations from multi-database
  - renaming of attributes and relations
  - definition of new relations

- measurement units and precision for attributes

  → automatic conversion of data values

- *multiple queries:*

  queries applied to a set of (heterogeneous) relations

Example for MSQL

bank (bnp) with three branches:

```
bnp:    account(acc#, cl#, balance, brname)
etoile: account(balance, acc#, cl#, open_date)
nation: acc(type, acc#, balance, open_date)
opera:  accounts(acc#, open_date, balance,
        cl#, category)
```

definition of a multi-database:

```
CREATE MULTIDATABASE branches (etoile,
            nation, opera).
```

example query:

```
USE bnp branches
LET x BE branches.*
STORE bnp.account(* *)
SELECT * NAME(x)
FROM x.acc%
```


Explanation of MSQL Query

example query:

```
USE bnp branches
LET x BE branches.*
STORE bnp.account(* *)
SELECT * NAME(x)
FROM x.acc%
```

- `LET`: bind `x` to database names
- `STORE`: replacement of tuples (key *acc#*)
- `(* *)`: mapping of local attributes (order-insensitive)
- `Name`: mapping relation name → value
- `acc%`: prefix of relation names

SchemaSQL

SchemaSQL [Lakshmanan et al., 1996]:
integrated access to schema information
characteristics of SchemaSQL:

- access to data and meta data
  → conflict resolution
- vertical and horizontal aggregation
  → aggregation crossing relation borders
  → aggregation of data and meta data
- arbitrary schema restructuring
- SQL as subset of SchemaSQL

SchemaSQL: Example

4 component databases with heterogeneous schemata but storing similar information:

Database **univ-A**:

`salInfo`

category	dept	salFloor
Prof	CS	65.000
AssProf	CS	50.000
Prof	Math	60.000
AssProf	Math	55.000

Database **univ-B**:

`salInfo`

category	CS	Math
Prof	55.000	65.000
AssProf	50.000	55.000

Database **univ-C**:

`CS`

category	salFloor
Prof	60.000
AssProf	55.000

`Math`

category	salFloor
Prof	70.000
AssProf	60.000

Database **univ-D**:

`salInfo`

dept	Prof	AssProf
CS	75.000	60.000
Math	60.000	45.000

SchemaSQL: Example Queries (I)

Query 1:

```
select    A.dept
from      univ-A::salInfo A,
          univ-B::salInfo B,
          univ-B::salInfo-> AttB
where     AttB <> "category"   and
          A.dept = AttB        and
          A.category = "Prof"  and
          B.category = "Prof"  and
          A.salFloor > B.AttB
```

- range of `AttB`:

  attribute names of relation `salinfo` in database `univ-B`

SchemaSQL: Example Queries (II)

Query 2:

variable binding to meta data and aggregation of both data and meta data in one query:

```
select   T.category, avg(T.D)
from     univ-B::salInfo-> D,
         univ-B::salInfo T
where    D <> "category"
group by T.category
```

- range of `D`:

  attribute names of relation `salInfo` in database `univ-B`

- horizontal aggregation in tuples

  → extension to SQL

SchemaSQL: Example Queries (III)

Query 3:

query 2 for schema univ-C

```
select    T.category, avg(T.salFloor)
from      univ-C-> D,
          univ-C::D T
group by  T.category
```

- range of D:

  relation names in database univ-C

- $\rightarrow$ multi-relation (vertical) aggregation

SchemaSQL: Example Queries (IV)

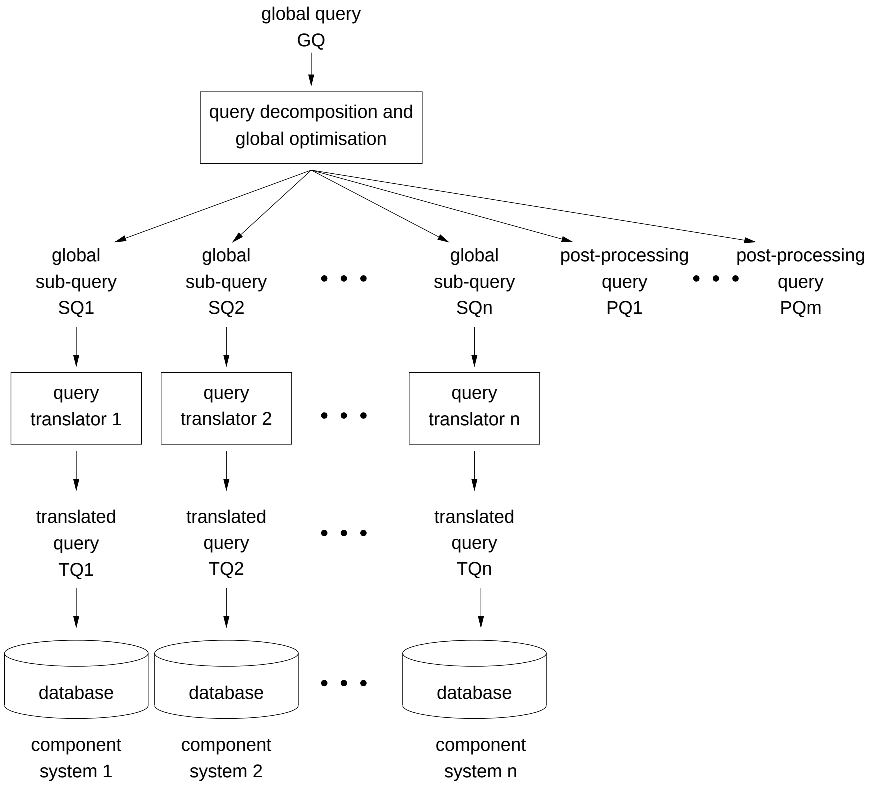
Query 4:

view definition integrating all four databases:

```
create view
 globalSchema::salInfo(univ,category,dept,salFloor)
as
    select "univ-A", T.category, T.dept, T.salFloor
    from   univ-A::salInfo T
  union
    select "univ-B", T.category, D, T.D
    from   univ-B::salInfo T,
           univ-B::salInfo-> D
    where  D <> "category"
  union
    select "univ-C", T.category, D, T.salFloor
    from   univ-C-> D,
           univ-C::D T
  union
    select "univ-D", D, T.dept, T.D
    from   univ-D::salInfo T,
           univ-D::salInfo-> D
    where  D <> "dept"
```

Query Processing

query processing in FDBS:



Query Processing Steps

- query decomposition of global query GQ into local queries (SQ1, SQ2, . . . , SQn) and post-processing queries (PQ1, PQ2, . . . , PQm)

- translation of SQ1, . . . , SQn into local query language (result: TQ1, . . . , TQn)

  translation into equivalent local query maybe impossible: further decomposition, etc.

- local processing of TQ1, . . . , TQn in CDBS

- translation of query results into global data model

- composition of global result
  (using post-processing queries)

Query Optimization

problems:

- autonomy of local optimizers
- different optimization strategies
- local encapsulation of statistical data
- heterogeneity (data models, languages, . . .)
- handling of replicated extensions
- inconsistencies between component databases

Instance Integration

- schema integration

  $\rightarrow$ mapping of queries

- presence of *extensional overlapping*

  $\rightarrow$ instance integration

  – identification of proxi-objects
  – key conflicts
  – manipulation of `same`-relation
  – identification of reference attributes

Instance Integration (II)

aim:
bijective function between classes with identical extension
(cf. Chapter 3: semantic conflicts)

- based on value-based identification: database keys

  → bijective mapping between key domains

- problems:
  - existence of the same key in all database?
  - incomplete mappings
  - local updates / insertions
    → propagation to other component databases
    * generating values for attributes
    * not possible for keys
  - explicitly stored mappings (`same`-relation)
  - databases with system-generated OID identification
  - use of keys for referencing attributes

Instance Integration (III)

example:

semantically equivalent relations with incompatible keys

Employee

<u>PersNr</u>	Name	FirstName	Salary
...	...	...	...
...	...	...	...

EMP

<u>Name</u>	FirstName	<u>DateOFBirth</u>	Salary
...	...	...	...
...	...	...	...

→ no bijective mapping between keys!

Security in FDBS

requirements and problems of security in FDBS

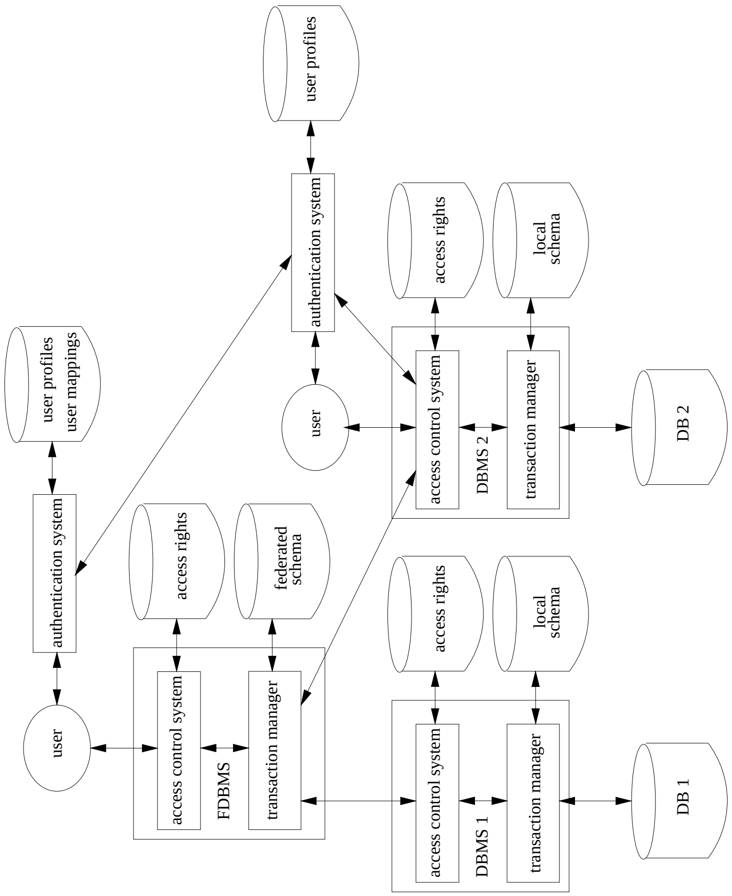
local view:

- autonomy: no restriction of local security measures through FDBMS
- cooperation: only if FDBMS more trustworthy than any local system
- heterogeneity: same results of authentication and authorisation before and after integration

global view:

- creation of a uniform security interface for all federated users
- including: authentication, authorisation and security policy
- solution of inference and aggregation problems

A Security Architecture for FDBS



Authentication in FDBS

requirements

- autonomy: decision about a user's wish to access to a local system on the basis of his local identity and identifier
- heterogeneity: one authentication per global user and session
- population: correct authentication of every user besides different identities in different local systems

possible solutions

- global: user identifies himself via the FDBMS
- direct: user identifies himself direct via all relevant systems
- indirect: user identifies himself via the FDBMS and FDBMS delivers the relevant identities to the local systems

Security Modeling in FDBS

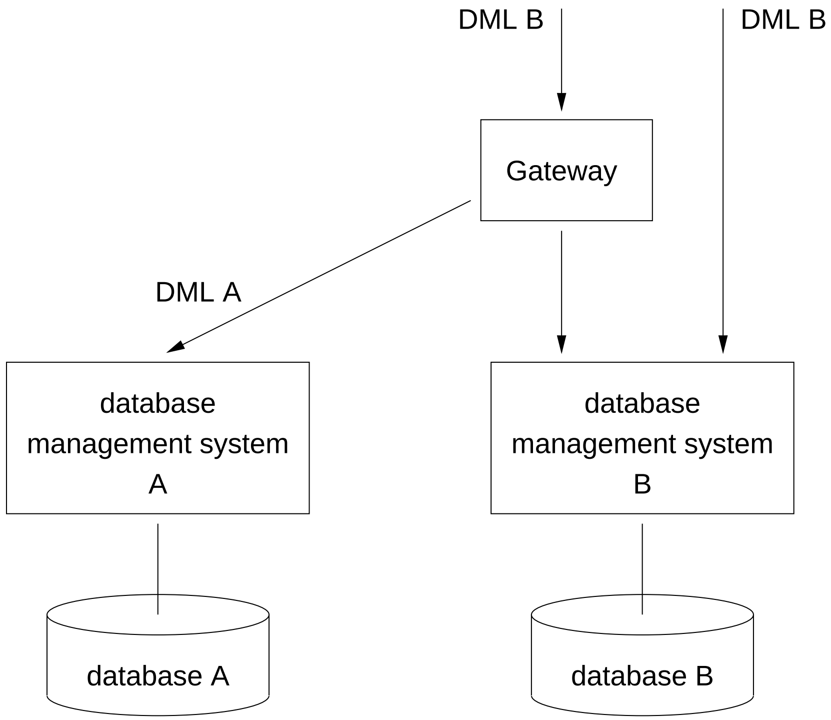
development steps

- selection of a global security model
- transformation of local security concepts to global ones
- integration of the local pecularities (users, roles, rights, etc) into a global security schema

example of an integration problem

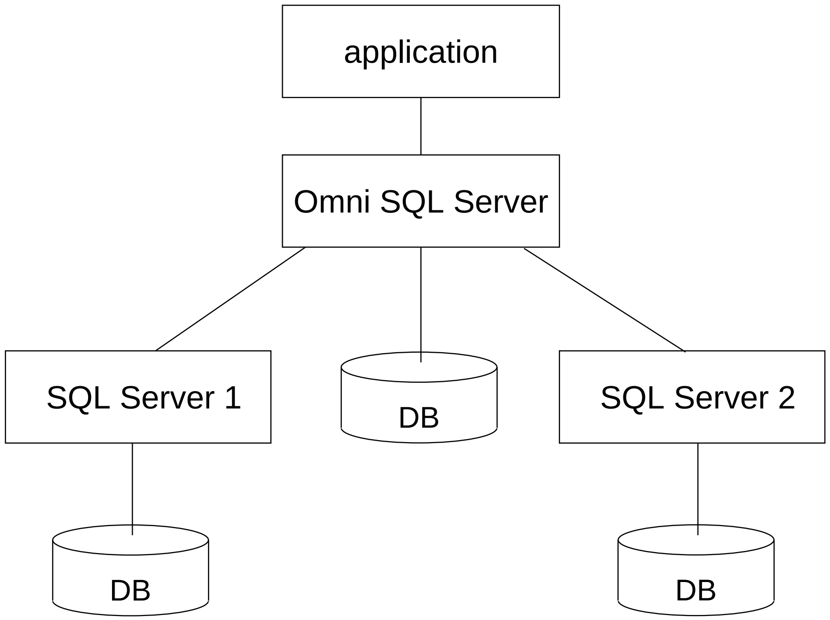
- DBMS 1 (discretionary: explicit access rights): user *John* has right to read and write table *sales*
- DBMS 2 (mandatory: security classes, no read–up, no write down): user *John* has clearance top secret, table *sales* is secret
- FDBMS (discretionary): user *John* has right to read table *sales*

Gateways and Database Coupling



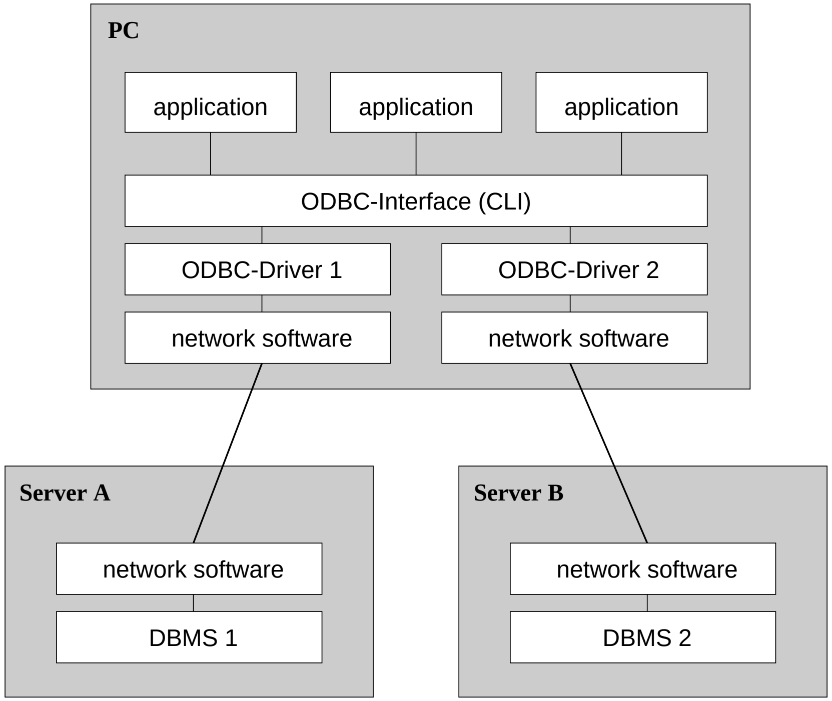
Omni-SQL-Server

Omni-SQL-Server: coupling of relational DBMS



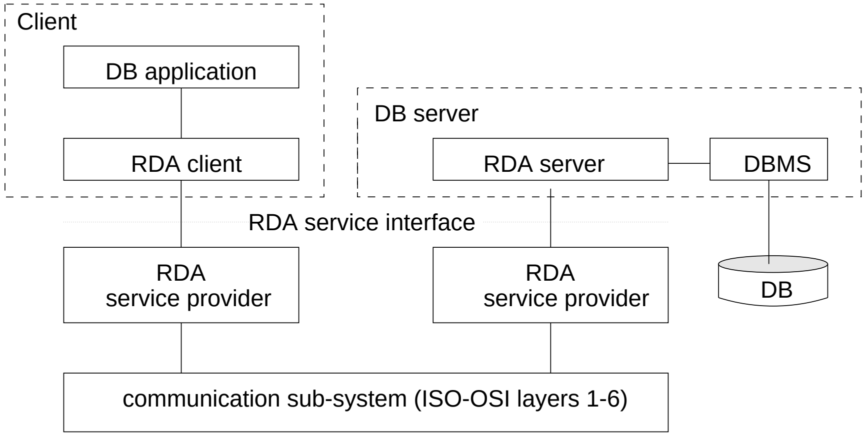
ODBC — Open Database Connectivity

ODBC architecture



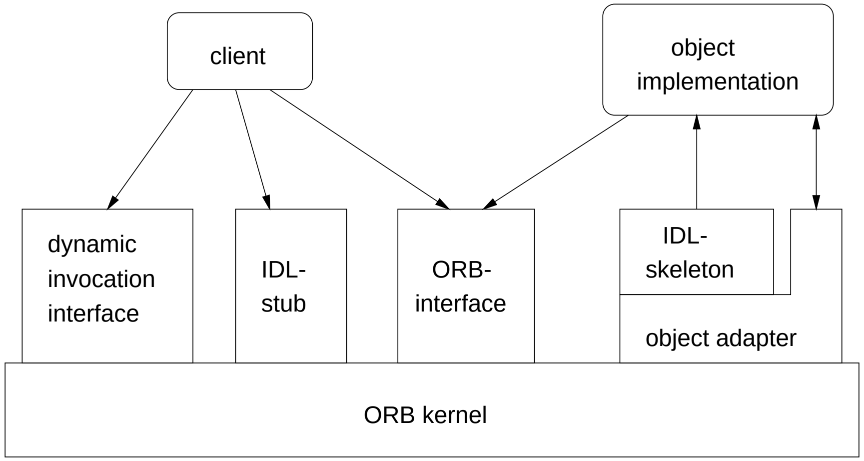
RDA — Remote Database Access

ISO/OSI communication protocol for distributed database operations in heterogeneous environments



CORBA — Common Object Request Broker Architecture

Common Object Request Broker Architecture (CORBA):



Selected References

multi-database language MSQL: [Litwin and Abdellatif, 1986], [Litwin, 1988], [Litwin et al., 1990], [Grant et al., 1993]

SchemaSQL: [Lakshmanan et al., 1996]

query processing and optimization for federated databases: [Czejdo et al., 1987], [Meng and Yu, 1995]; also [Rusinkiewicz and Czejdo, 1985], [Dong and Ramamohanarao, 1993], [Zhu and Larson, 1996]

determining local optimization strategies: [Spiliopoulou, 1996b], [Spiliopoulou, 1996a]

security in federated database systems: [Morgenstern et al., 1991], [Pernul, 1993], [Jonscher and Dittrich, 1993], [Idris et al., 1994], [Jonscher and Dittrich, 1994], [Eßmayr et al., 1995], [Tresch and Scholl, 1993], [Templeton et al., 1995], [Eßmayr et al., 1996], [Tari and Fernandez, 1997]

database gateways: [Brodie and Stonebraker, 1995]

7. Current Developments of FDBS

selected prototype systems:

- IRO-DB
- OpenDM/Efendi
- PEGASUS
- MARIPOSA

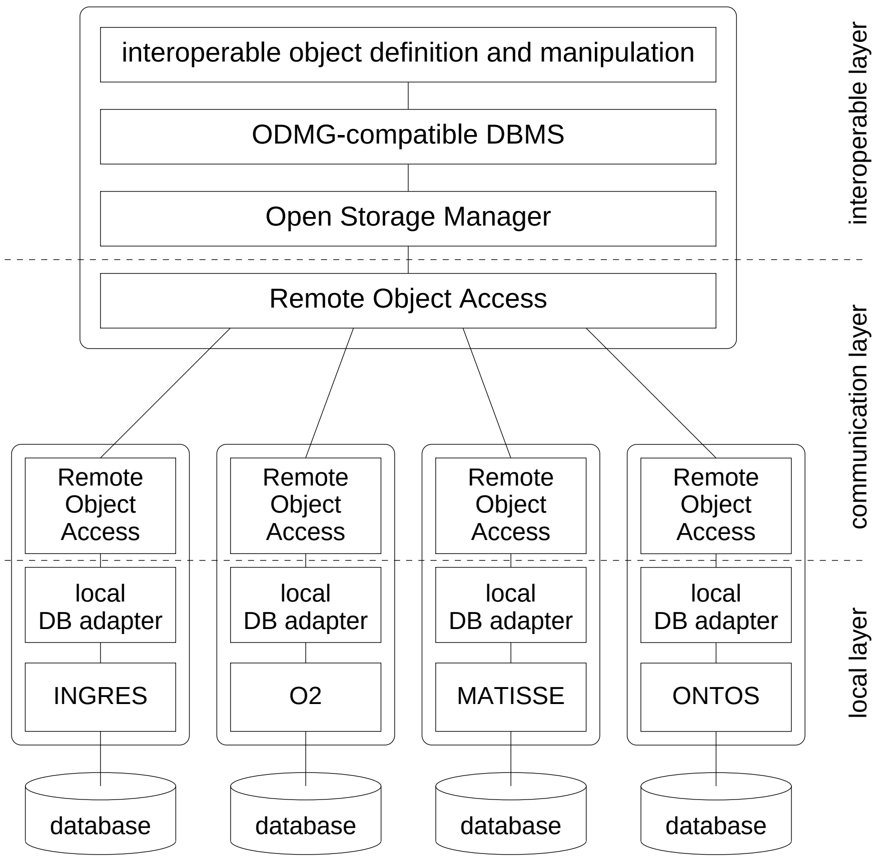
IRO-DB

IRO-DB
=
Interoperable Relational and Object-oriented Data Bases

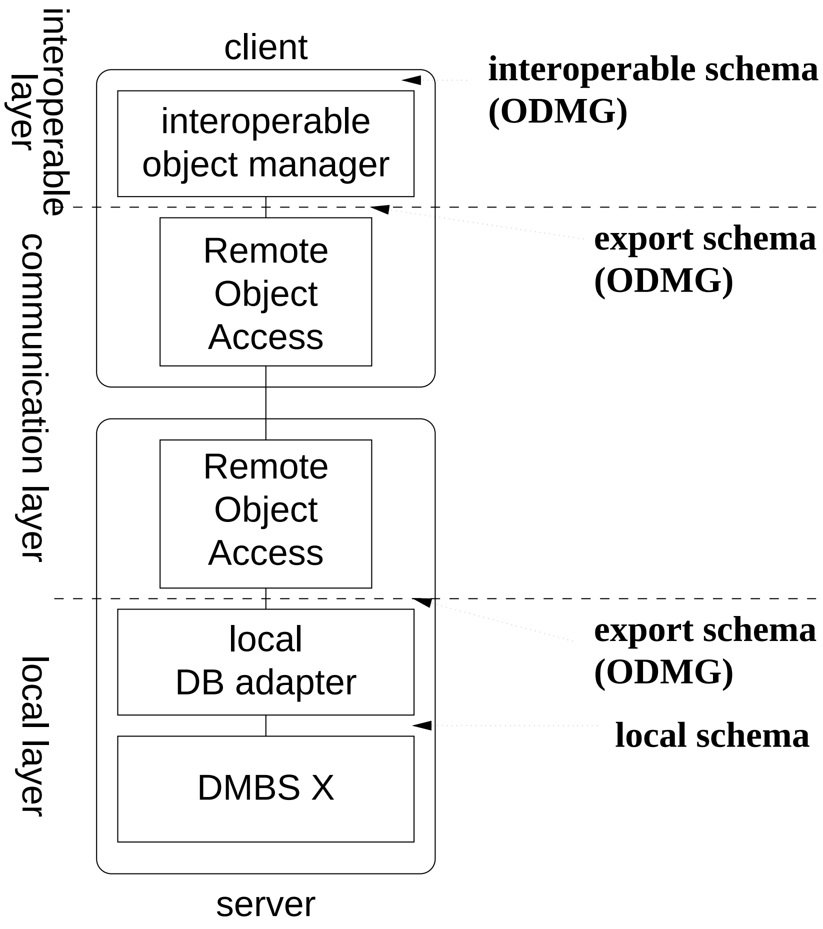
EU ESPRIT project:
[Busse et al., 1994a], [Busse et al., 1994b], [Gardarin et al., 1996a]

- 5-level architecture
- global data model: ODMG
- mapping of local sources: adapters
- communication: RDA
- open nested transactions (multi-level transactions)

IRO-DB System Architecture



Schemata in IRO-DB



OpenDM/Efendi

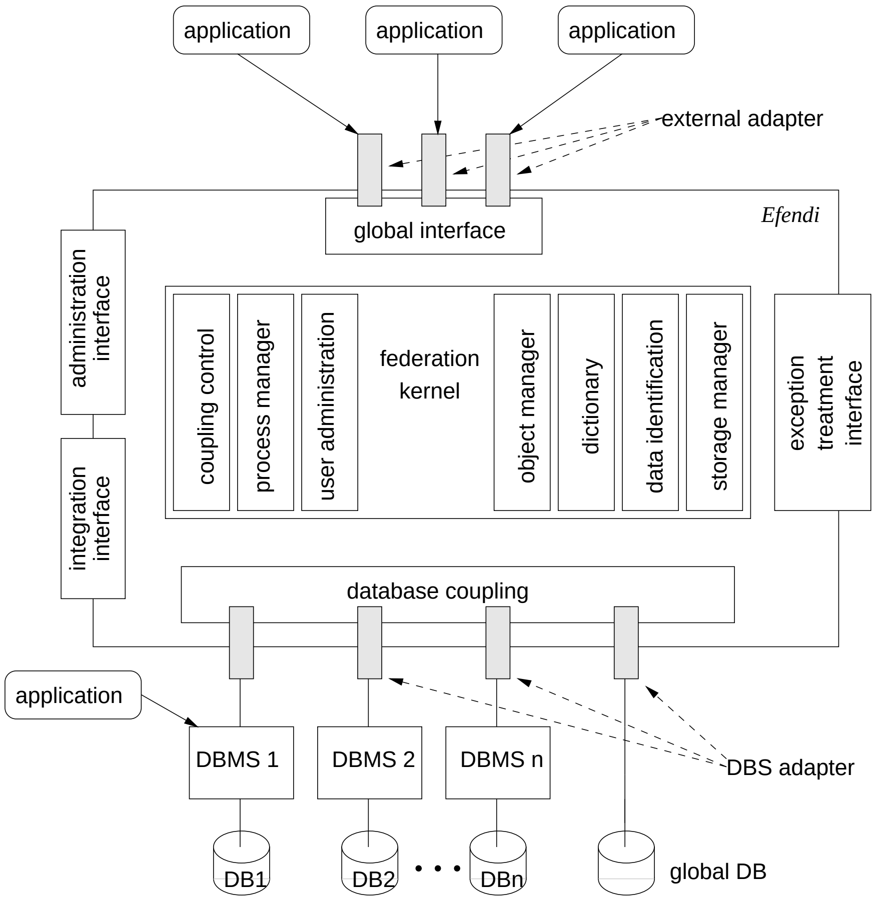
OpenDM = *Open Database Middleware*

product development at C-Lab (Paderborn)

- arbitrary database systems as CDBMS
- global layer: ODMG or SDAI for STEP/C++
- modular system architecture
  - OpenDM : system kernel
  - OpenDM/ODMG-93 as ODMG DBMS [Middleware Project Team, 1996b]
  - OpenDM/Efendi as FDBS [Böttger et al., 1995] (Efendi = *E*xtensible *F*ederation *EN*gine for *D*atabase *I*ntegration)

[Middleware Project Team, 1996a],
[Middleware Project Team, 1996b], [Böttger et al., 1995]

Architecture of OpenDM/Efendi



Pegasus

development at Hewlett-Packard Laboratories
[Ahmed et al., 1991], [Shan et al., 1995]

PEGASUS system architecture in three levels:

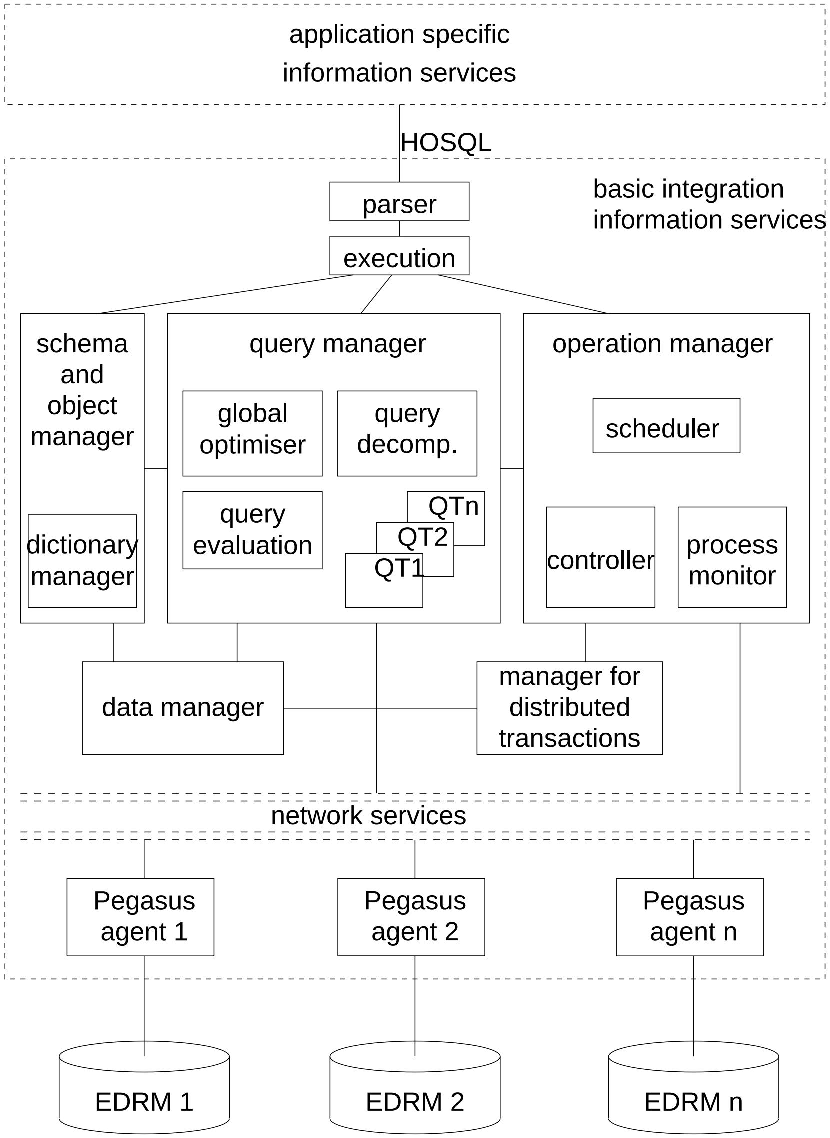
- global level: application-specific information services on global interface

  HOSQL: *Heterogeneous Object SQL*

- middle level: basic integration services

- external data source managers

Pegasus Architecture



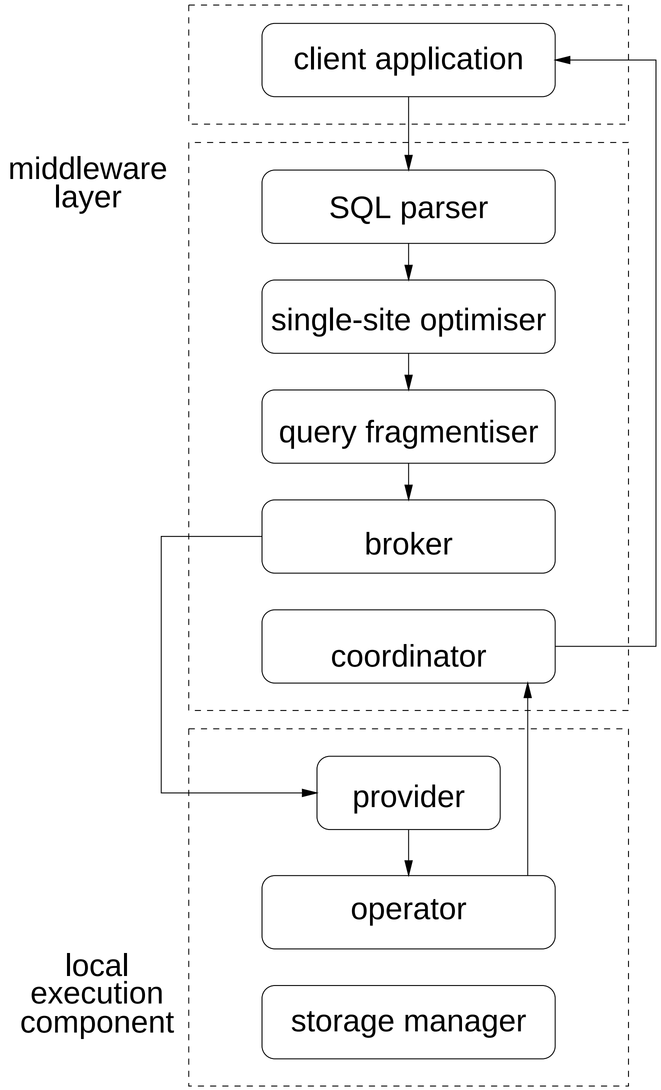
Mariposa

"distributed database system for wide area networks"
[Stonebraker et al., 1996]

aims:

- scalability
- data mobility (object migration)
- no global synchronization
- full local autonomy
- configurable strategies for system behavior

Mariposa Architecture



Other Projects

this list is *not* complete!

Carnot

conceptual integration of distributed and heterogeneous information

information sources: databases, database applications, expert systems, knowledge bases

support of workflow management

AI technology for integration

[MCC Inc.]

CIS COMANDOS Integration System

integration of heterogeneous application environments

(relational DBs, graphics DB, public DBs)

[EU ESPRIT project COMANDOS]

Other Projects (II)

DOMS Distributed Object Management System

object-oriented environment for integration of autonomous and heterogeneous local systems

not restricted to DBMS !

[GTE Laboratories]

EIS/XAIT Project Engineering Information Systems

Object Management System (OMS) as interoperable frame for EIS

advanced transaction model for cooperating transactions

$\rightarrow$ support of Cooperative Engineering

[XAIT=Xerox Advanced Information Technology]

FIB Project Federated Information Bases

main focus: semantic interoperation

global data model: terminological knowledge model (CANDIDE)

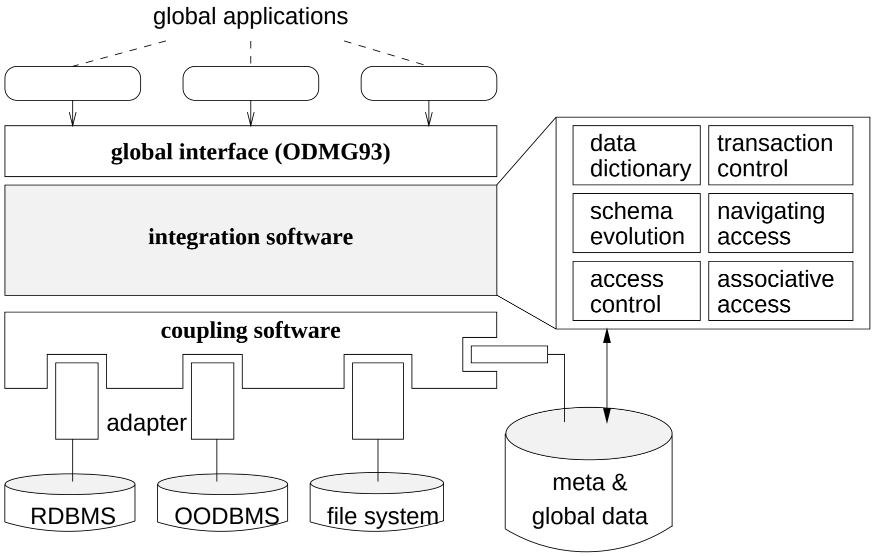
Other Projects (III)

FIHD Project *F*lexible *I*ntegration of *H*eterogeneous *D*atabase systems

ODMG as global data model

focus: semantic enrichment of relational databases for transformation into ODMG

system architecture of FIHD:



[Siemens AG, ZFE]

Other Projects (IV)

InterBase Project

two parts:

- FBASE: prototype for data modeling, schema transformation and integration
- InterBase*: complete transaction system

[Purdue University]

InterViso

database frontend

uniform view on several databases

previous prototype: MERMAID

[Data Integration Inc.]

MIND *M*ETU *IN*teroperable *D*atabase system

architecture based on OMG CORBA

global data model: ODMG

extension of SQL as global DML

integration of OODBMS and RDBMS

[Middle East Technical University (METU), Turkey]

Other Projects (V)

OIS Operational Integration System

- integration tool for application environments (uniform data access interface)
- integration of DBMS with file-based systems
- [EU ESPRIT Project TOOTSI]

Thor

- object-oriented distributed database management system
- heterogeneous distributed systems
- database-centered integration (→ no federation!)
- [MIT]

UniSQL/M

- object-relational DBMS
- integration of other SQL-based DBMS
- [UniSQL Inc.]

Other Projects (VI)

ViewSystem

- prototype of KODIM project (Knowledge Oriented Distributed Information Management)
- VODAK: object-oriented data model
- dynamic integration of heterogeneous and autonomous information bases
- object-oriented query language; virtual classes as views
- [GMD-IPSI Darmstadt]

Selected References

articles/books containing surveys on (current) developments:
[Litwin et al., 1990], [Thomas et al., 1990], [Hurson et al., 1994, chapter 5], [Pitoura et al., 1995], [Bukhres and Elmagarmid, 1996]

IRO-DB system: [Busse et al., 1994a], [Busse et al., 1994b], [Gardarin et al., 1996a];
global query optimization: [Gardarin et al., 1996b];
application in CIM scenario: [Ramfos et al., 1995]

OpenDM — versions: OpenDM [Middleware Project Team, 1996a], OpenDM/ODMG-93 [Middleware Project Team, 1996b], OpenDM/Efendi [Böttger et al., 1995] ;
object migration: [Radeke and Scholl, 1994a], [Radeke and Scholl, 1994b], [Radeke and Scholl, 1995], [Radeke, 1995];
internal object model COM (*Context Object Model*): [Kachel, 1992]

PEGASUS system: [Ahmed et al., 1991], [Shan et al., 1995]

MARIPOSA: [Stonebraker et al., 1996];
query processing: [Stonebraker et al., 1994];
data replication: [Sidell et al., 1996]

Selected References (II)

other projects and systems:
Carnot [Collet et al., 1991], [Huhns et al., 1992];
Comandos Integration System (CIS) [Bertino et al., 1989];
Distributed Object Management System (DOMS) [Buchmann et al., 1992], [Manola et al., 1992], [Georgakoupoulos et al., 1994];
EIS/XAIT project (Engineering Information Systems) [Heiler and Zdonik, 1990], [Heiler et al., 1992];
Federated Information Bases (FIB) [Savasere et al., 1991];
FIHD project [Hohenstein and Körner, 1995], [Hohenstein and Plesser, 1996];
InterBase [Du and Elmagarmid, 1989], [Elmagarmid et al., 1990], [Bukhres et al., 1993];
InterViso [Thomas et al., 1990], [Templeton et al., 1995];
MIND (*M*ETU *IN*teroperable *D*atabase system) [Dogac et al., 1995];
OIS (Operational Integration System) [Gagliardi et al., 1990];
THOR [Liskov et al., 1994];
UniSQL/M [Kim et al., 1995], [Kelley et al., 1995];
ViewSystem [Kaul et al., 1991]

MRDSM system (Multics Relational Data Store Multidatabase) [Litwin and Abdellatif, 1986], [Litwin, 1988], [Litwin et al., 1990]

Mediator-based approaches: [Papakonstantinou et al., 1996], [Zhou et al., 1996]

8. Examples for Application Areas

brief view on selected application scenarios for database federations:

- hospital information systems
- factory planning:

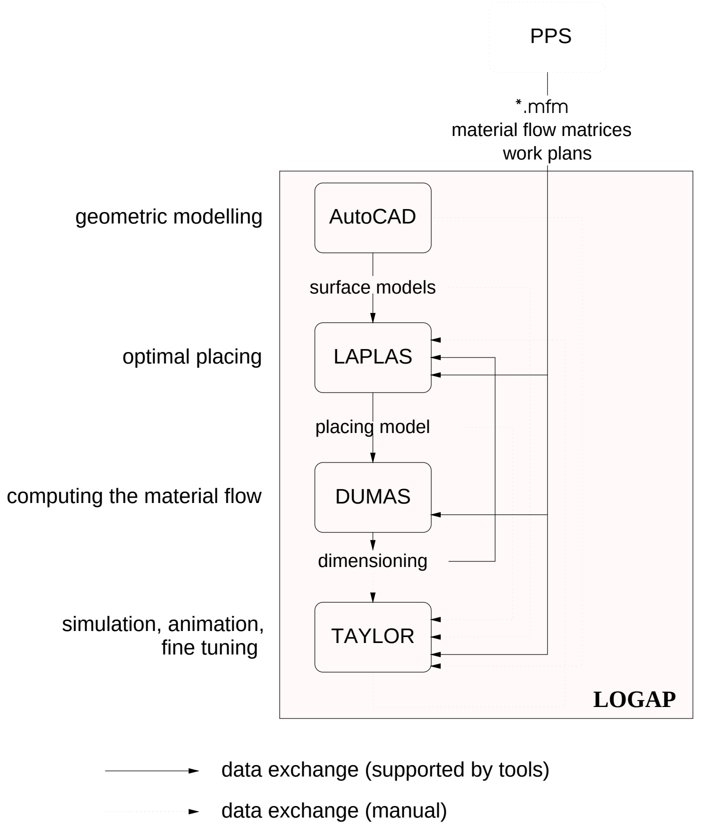
  tool integration and cooperative engineering

Factory Planning

main requirements

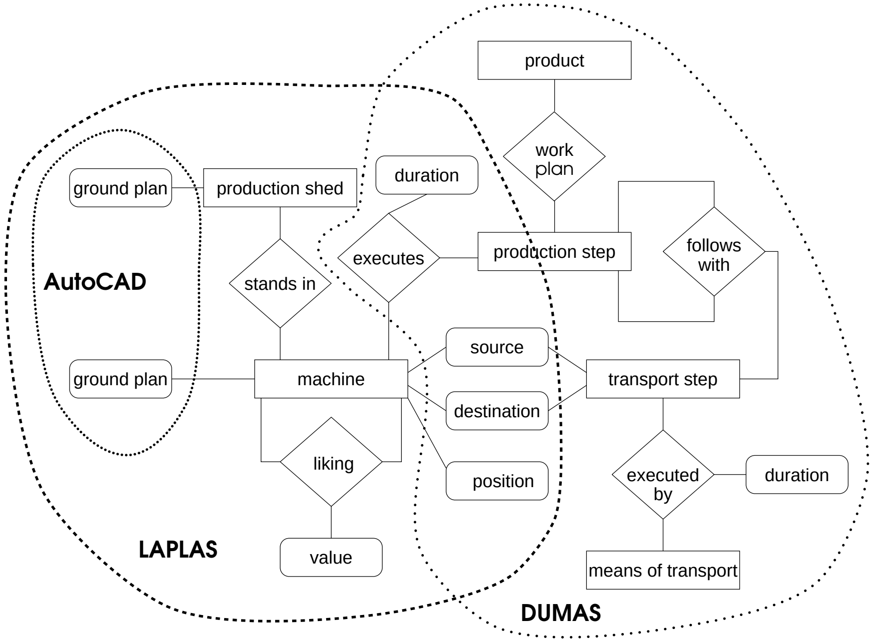
- tool integration
- cooperative engineering
  (cooperative design/development)

existing software tools:



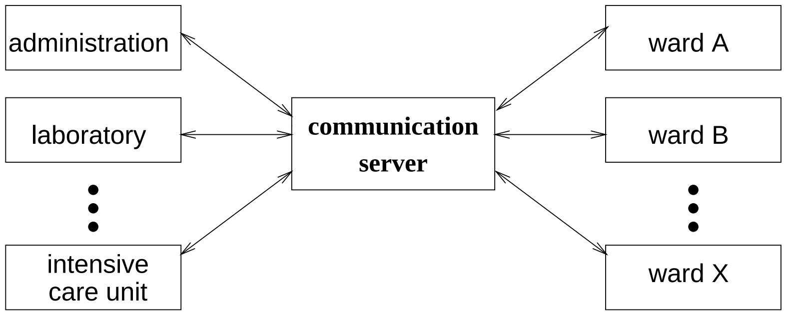
Factory Planning (II)

conceptual schema (simplified):



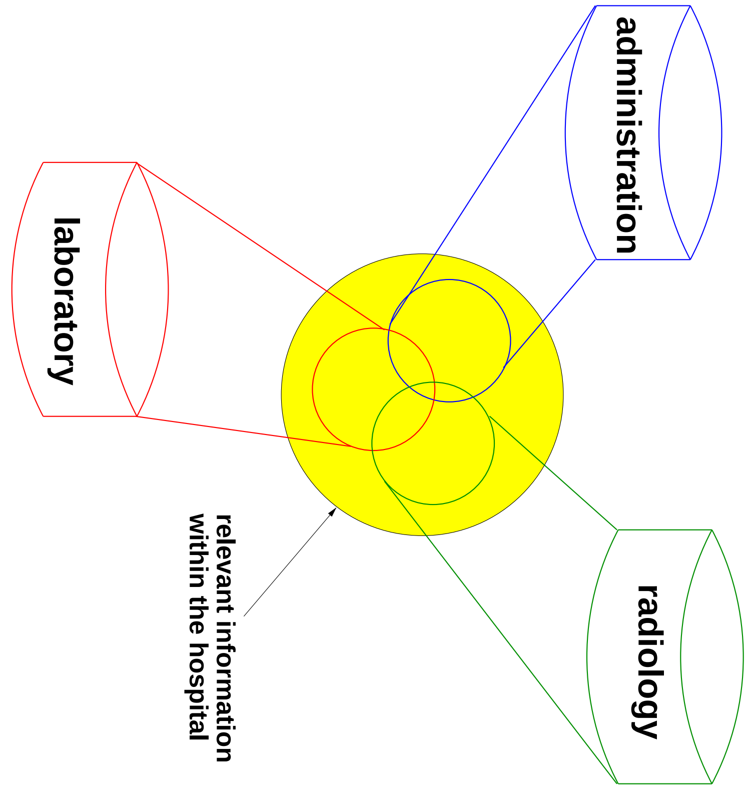
Hospital Information Systems

"traditional approach":
communication server architecture



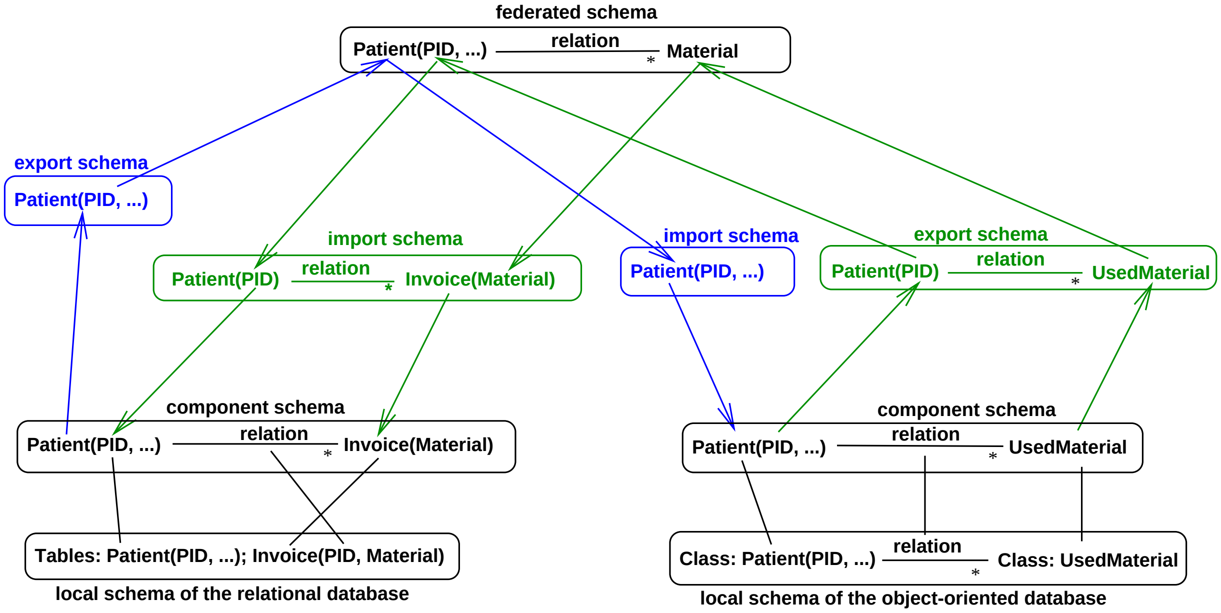
Hospital Information Systems (II)

overlapping of data within a hospital information system:

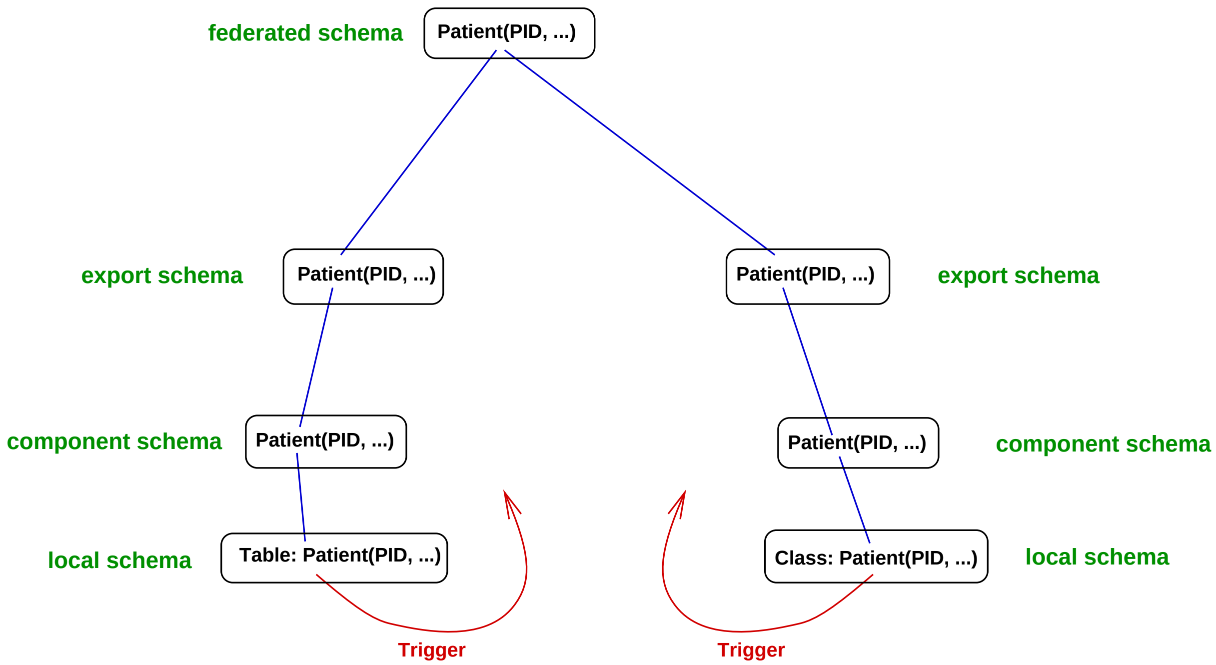


[Hasselbring, 1997b]

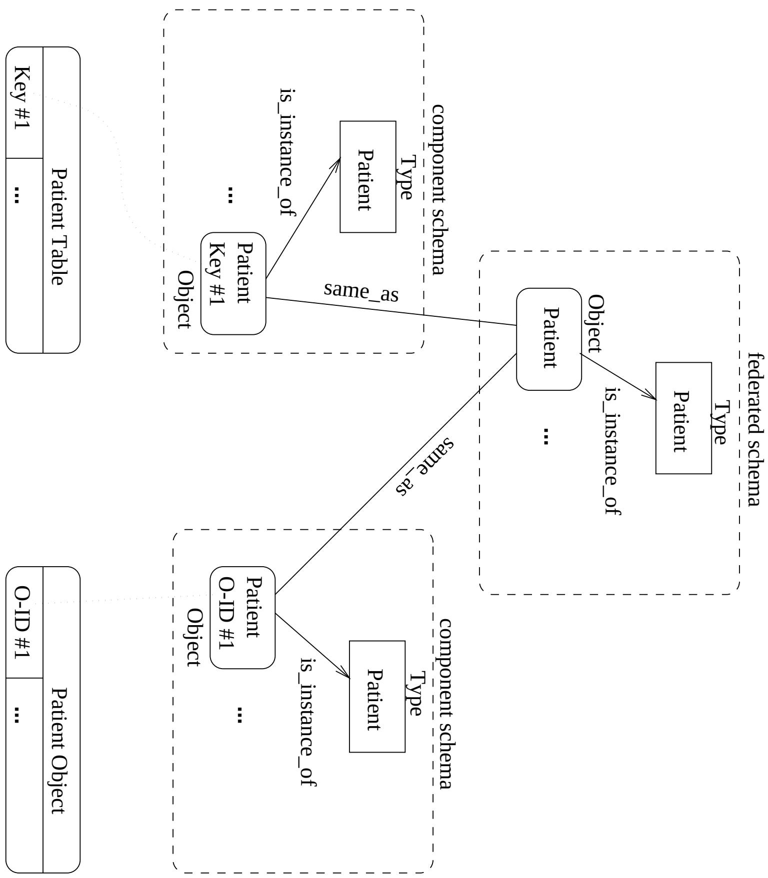
Hospital Information Systems



Hospital Information Systems



Hospital Information Systems



Selected References

factory planning:
[Höding et al., 1995, in German], [Köller, 1997, in German]

hospital information systems: [Hasselbring, 1997a], [Hasselbring, 1997b]

other application areas: [Ramfos et al., 1995]
[Encarnação and Lockemann, 1990]
[Günthör et al., 1996] [Batini et al., 1996]
[Buhalis, 1996] [Parent et al., 1996]

References

[Agrawal et al., 1993] Agrawal, D., El Abbadi, A., and Singh, A. (1993). Consistency and Orderability: Semantics-Based Correctness Criteria for Databases. *ACM Transactions on Database Systems*, 18(3):460–486.

[Ahmed et al., 1991] Ahmed, R., DeSchedt, P., Du, W., Kent, W., Ketabchi, M., Litwin, W., Rafii, A., and Shan, M.-C. (1991). The Pegasus Heterogeneous Multidatabase System. *IEEE Computer*, 24(12):19–27.

[Alonso et al., 1994] Alonso, G., Vingralek, R., Agrawal, D., Breitbart, Y., El Abbadi, A., Schek, H., and Weikum, G. (1994). A Unified Approach to Concurrency Control and Transaction Recovery (Extended Abstract). In Jarke, M., Bubenko, J., and Jeffery, K., editors, *Advances in Database Technology — EDBT'94*, number 779 in Lecture Notes in Computer Science, pages 123–130. Springer-Verlag.

[Andersson et al., 1993] Andersson, M., Dupont, Y., Spaccapietra, S., Yétongnon, K., Tresch, M., and Ye, H. (1993). FEMUS: A Federated Multilingual Database System. In Adam, N. and Bhargava, B., editors, *Advanced Database Systems*, number 759 in Lecture Notes in Computer Science, pages 359–380. Springer-Verlag.

[Atzeni and Torlone, 1997] Atzeni, P. and Torlone, R. (1997). Schema Translation between Heterogeneous Data Models in a Lattice Framework. In Meersman, R. and Mark, L., editors, *Database Applications Semantics — Proc. of the IFIP WG 2.6 Working Conf. DS-6 (1995, Atlanta, Georgia)*, pages 345–364. Chapman&Hall, London.

[Barbará and Garcia-Molina, 1992] Barbará, D. and Garcia-Molina, H. (1992). The Demarcation Protocol: A Technique for Maintaining Linear Arithmetic Constraints in Distributed Database Systems. In Pirotte, A., Delobel, C., and Gottlob, G., editors, *Advances in Database Technology — EDBT'92*, number 580 in Lecture Notes in Computer Science, pages 373–388. Springer-Verlag.

[Barbará-Millá and Garcia-Molina, 1994] Barbará-Millá, D. and Garcia-Molina, H. (1994). The Demarcation Protocol: A Technique for Maintaining Constraints in Distributed Database Systems. *The VLDB Journal*, 3(3):325–353.

[Barghouti and Kaiser, 1991] Barghouti, N. and Kaiser, G. (1991). Concurrency Control in Advanced Database Application. *ACM Computing Surveys*, 23(3):269–317.

[Batini et al., 1996] Batini, C., Castano, S., De Antonellis, V., Fugini, M., and Pernici, B. (1996). Analysis of an Inventory of Information Systems in the Public Administration. *Requirements Engineering Journal*, 1(1):47–62.

[Batini et al., 1992] Batini, C., Ceri, S., and Navathe, S. (1992). *Conceptual Database Design — An Entity-Relationship Approach*. Benjamin/Cummings.

[Batini and Lenzerini, 1984] Batini, C. and Lenzerini, M. (1984). A Methodology for Data Schema Integration in the Entity-Relationship Model. *IEEE Transactions on Software Engineering*, 10(6):650–664.

[Batini et al., 1986] Batini, C., Lenzerini, M., and Navathe, S. (1986). A Comparative Analysis of Methodologies for Database Schema Integration. *ACM Computing Surveys*, 18(4):323–364.

[Bell and Grimson, 1992] Bell, D. and Grimson, J. (1992). *Distributed Database Systems*. Addison-Wesley.

[Benn et al., 1996] Benn, W., Chen, Y., and Gringer, I. (1996). A New Strategy for Integrating Heterogeneous OO Schemas. In [Conrad et al., 1996a], pages 19–27.

[Bernstein et al., 1987] Bernstein, P., Hadzilacos, V., and Goodman, N. (1987). *Concurrency Control and Recovery in Database Systems*. Addison-Wesley, Reading, MA.

[Bertino et al., 1989] Bertino, E., Negri, M., Pelagatti, G., and Sbattella, L. (1989). Integration of Heterogeneous Database Applications through an Object-Oriented Interface. *Information Systems*, 14(5):407–420.

[Böttger et al., 1995] Böttger, R., Burkert, B., Engel, Y., Heijenga, W., Kachel, G., Kolmschlag, S., Meckenstock, A., Nolte, D., Radeke, E., and Zimmer, D. (1995). Efendi 1.0 Database Federation Services. System overview, Cadlab, Paderborn.

[Breitbart, 1990] Breitbart, Y. (1990). Multidatabase Interoperability. *ACM SIGMOD Record*, 19(3):53–60.

[Breitbart et al., 1992a] Breitbart, Y., Garcia-Molina, H., and Silberschatz, A. (1992a). Overview of Multidatabase Transaction Management. *The VLDB Journal*, 1(2):181–240.

[Breitbart et al., 1995] Breitbart, Y., Garcia-Molina, H., and Silberschatz, A. (1995). Transaction Management in Multidatabase Systems. In [Kim, 1995], pages 573–591.

[Breitbart et al., 1991] Breitbart, Y., Georgakopoulos, D., Rusinkiewicz, M., and Silberschatz, A. (1991). On Rigorous Transaction Scheduling. *IEEE Transactions on Software Engineering*, 17(9):954–960.

[Breitbart and Silberschatz, 1988] Breitbart, Y. and Silberschatz, A. (1988). Multidatabase Update Issues. In Boral, H. and Larson, P., editors, *Proc. of the 1988 ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'88)*, pages 135–142. ACM Press.

[Breitbart et al., 1992b] Breitbart, Y., Silberschatz, A., and Thompson, G. R. (1992b). Transaction Management Issues in a Failure-Prone Multidatabase System Environment. *The VLDB Journal*, 1(1):1–39.

[Bright et al., 1992] Bright, M., Hurson, A., and Pakzad, S. (1992). A Taxonomy and Current Issues in Multidatabase Systems. *IEEE Computer*, 25(3):50–60.

[Bright et al., 1994] Bright, M., Hurson, A., Pakzad, S., and Sarma, H. (1994). The Summary Schemas Model—An Approach for Handling Multidatabases: Concept and Performance Analysis. In [Hurson et al., 1994], pages 199–216.

[Brodie and Stonebraker, 1995] Brodie, M. and Stonebraker, M. (1995). *Migrating Legacy Systems*. Morgan Kaufmann Publishers, San Francisco, CA.

[Buchmann et al., 1992] Buchmann, A., Özsu, M., Hornick, M., Georgakopoulous, D., and Manola, F. (1992). A Transaction Model for Active Distrubuted Object Systems. In [Elmagarmid, 1992], pages 123–158.

[Buhalis, 1996] Buhalis, D. (1996). Enhancing the Competitiveness of Small and Medium-sized Tourism Enterprises. *Electronic Markets — International Journal of Electronic Commerce*, 6(1):1–6.

[Bukhres et al., 1993] Bukhres, O., Chen, J., Du, W., Elmagarmid, A., and Pezzoli, R. (1993). InterBase: An Execution Environment For Heterogeneous Software Systems. *IEEE Computer*, 26(8):57–69.

[Bukhres and Elmagarmid, 1996] Bukhres, O. and Elmagarmid, A., editors (1996). *Object-Oriented Multidatabase Systems: A Solution for Advanced Applications*. Prentice-Hall.

[Bültzingsloewen et al., 1996] Bültzingsloewen, G., Koschel, A., and Kramer, R. (1996). Active Information Delivery in a CORBA-based Distributed Information System. In *Proc. IFCIS International Conference on Cooperative Information Systems (CoopIS'96)*, pages 218–227. IEEE Computer Society Press.

[Busse et al., 1994a] Busse, R., Fankhauser, P., Huck, G., and Klas, W. (1994a). IRO-DB: An Object-Oriented Approach Towards Federated and Interoperable DBMS. In *Proc. of the Int. Workshop on Advances in Databases and Information Systems (ADBIS'94), Moscow, Russia*.

[Busse et al., 1994b] Busse, R., Fankhauser, P., and Neuhold, E. (1994b). Federated Schemata in ODMG. In Eder, J. and Kalinichenko, L., editors, *East/West Database Workshop: Klagenfurt 1994*, Workshops in Computing, pages 356–379. Springer-Verlag.

[Castellanos and Saltor, 1991] Castellanos, M. and Saltor, F. (1991). Semantic Enrichment of Database Schemas: An Object Oriented Approach. In Kambayashi, Y., Rusinkiewicz, M., and Sheth, A., editors, *Proc. First International Workshop on Interoperability in Multidatabase Systems (IMS'91)*, pages 71–78. IEEE Computer Society Press.

[Ceri and Pelagatti, 1984] Ceri, S. and Pelagatti, G. (1984). *Distributed Databases: Principles and Systems*. McGraw Hill, New York.

[Ceri et al., 1987] Ceri, S., Pernici, B., and Wiederhold, G. (1987). Distributed Database Design Methodologies. In *IEEE Proceedings — Issue on Distributed Databases*, pages 533–546. IEEE Computer Society Press.

[Chen, 1996] Chen, A. (1996). Identifying Object Isomerism in Multidatabase Systems. *Distributed and Parallel Databases*, 4(2):143–168.

[Collet et al., 1991] Collet, C., Huhns, M., and Shen, W.-M. (1991). Resource Integration Using a Large Knowledge Base in Carnot. *IEEE Computer*, 24(12):55–62.

[Conrad, 1997] Conrad, S. (1997). *Föderierte Datenbanksysteme: Konzepte der Datenintegration*. Springer-Verlag, Berlin/Heidelberg.

[Conrad et al., 1997a] Conrad, S., Hasselbring, W., Heuer, A., and Saake, G., editors (1997a). *Engineering Federated Database Systems EFDBS'97 (Proc. of the International CAiSE'97 Workshop, Barcelona, June 16–17, 1997)*, number 6 in Fakultät für Informatik, Preprint. Universität Magdeburg.

[Conrad et al., 1996a] Conrad, S., Höding, M., Janssen, S., and Saake, G., editors (1996a). *Workshop "Föderierte Datenbanken": Kurzfassungen der Beiträge*, number ITI-96-01 in Institutsbericht. Universität Magdeburg.

[Conrad et al., 1997b] Conrad, S., Höding, M., Saake, G., Schmitt, I., and Türker, C. (1997b). Schema Integration with Integrity Constraints. In Small, C., Douglas, P., Johnson, R., King, P., and Martin, N., editors, *Advances in Databases, 15th British National Conf. on Databases, BNCOD 15, London, UK, July 1997*, volume 1271 of *Lecture Notes in Computer Science*, pages 200–214, Berlin. Springer-Verlag.

[Conrad et al., 1996b] Conrad, S., Saake, G., Schmitt, I., and Türker, C., editors (1996b). *Kurzfassungen — 8. Workshop "'Grundlagen von Datenbanken"'*, Preprint 3. Fakultät für Informatik, Universität Magdeburg.

[Conrad et al., 1997c] Conrad, S., Schmitt, I., and Türker, C. (1997c). Dealing With Integrity Constraints During Schema Integration. In Conrad, S., Hasselbring, W., Heuer, A., and Saake, G., editors, *Engineering Federated Database Systems EFDBS'97 — Proc. of the Int. CAiSE'97 Workshop, Barcelona, June 16–17, 1997*, pages 13–22. Fakultät für Informatik, Universität Magdeburg.

[Czejdo et al., 1987] Czejdo, B., Rusinkiewicz, M., and Embley, D. (1987). An Approach to Schema Integration and Query Formulation in Federated Database Systems. In Wah, B., editor, *Proc. of the 3rd IEEE Int. Conf. on Data Engineering, (ICDE'87)*, pages 477–484. IEEE Computer Society Press.

[Dayal and Hwang, 1984] Dayal, U. and Hwang, H.-Y. (1984). View Definition and Generalization for Database Integration in a Multidatabase System. *IEEE Transactions on Software Engineering*, 10(6):628–644.

[Deacon et al., 1994] Deacon, A., Schek, H.-J., and Weikum, G. (1994). Semantics-based Multilevel Transaction Management in Federated Systems. In *Proc. of the 10th IEEE Int. Conf. on Data Engineering (ICDE'94)*, pages 452–461. IEEE Computer Society Press.

[Do and Drew, 1995] Do, L. and Drew, P. (1995). Active Database Management of Global Data Integrity Constraints in Heterogeneous Database Environments. In Yu, P. and Chen, A., editors, *Proc. of the 11th IEEE Int. Conf. on Data Engineering (ICDE'95)*, pages 99–108. IEEE Computer Society Press.

[Dogac et al., 1995] Dogac, A., Dengi, C., Kilic, E., Ozhan, G., Nural, S., Evrendilek, C., Halici, U., Arpinar, B., Koksal, P., Kesim, N., and Mancuhan, S. (1995). METU Interoperable Database System. *ACM SIGMOD Record*, 24(3):56–61.

[Dong and Ramamohanarao, 1993] Dong, G. and Ramamohanarao, K. (1993). Representation and Translation of Queries in Heterogeneous Databases with Schematic Discrepancies. In [Hsiao et al., 1993], pages 177–189.

[Du and Elmagarmid, 1989] Du, W. and Elmagarmid, A. (1989). QSR: A Correctness Criterion For Global Concurrency Control in InterBase. In Apers, P. and Wiederhold, G., editors, *Proc. of the 15th Int. Conf. on Very Large Data Bases (VLDB'89)*, pages 347–355. Morgan Kaufmann Publishers, Palo Alto, CA.

[Dupont, 1994] Dupont, Y. (1994). Resolving Fragmentation Conflicts in Schema Integration. In Loucopoulos, P., editor, *Entity-Relationship Approach — ER'94, Proc. of the 13th Int. Conf., Manchester, UK*, number 881 in Lecture Notes in Computer Science, pages 513–532. Springer-Verlag.

[Elmagarmid, 1992] Elmagarmid, A., editor (1992). *Database Transaction Models For Advanced Applications*. Morgan Kaufmann Publishers.

[Elmagarmid and Du, 1990] Elmagarmid, A. and Du, W. (1990). A Paradigm for Concurrency Control in Heterogeneous Distributed Database Systems. In Liu, M., editor, *Proc. International Conference on Data Engineering (ICDE'90)*, pages 37–46. IEEE Computer Society Press.

[Elmagarmid et al., 1990] Elmagarmid, A., Leu, Y., Litwin, W., and Rusinkiewicz, M. (1990). A Multidatabase Transaction Model for InterBase. In *Proc. of the 16th Int. Conf. on Very Large Data Bases (VLDB'90)*, pages 507–518. Morgan Kaufmann Publishers, Palo Alto, CA.

[Encarnação and Lockemann, 1990] Encarnação, J. and Lockemann, P. (1990). *Engineering Databases: Connecting Islands of Automation Through Databases*. Springer-Verlag, Berlin.

[Eßmayr et al., 1996] Eßmayr, W., Kastner, F., Pernul, G., Preishuber, S., and Tjoa, A. (1996). Authorization and Access Control in IRO-DB. In Su, S., editor, *Proceedings of the 12th International Conference on Data Engineering (ICDE'96)*, pages 40–47. IEEE Computer Society Press.

[Eßmayr et al., 1995] Eßmayr, W., Kastner, F., Preishuber, S., Pernul, G., and Tjoa, A. (1995). Access Controls for Federated Database Environments — Taxonomy of Design Choices. In *Proc. Joint IFIP TC 6 and TC 11 Working Conference on Communication and Multimedia Security*.

[Fahrner and Vossen, 1995] Fahrner, C. and Vossen, G. (1995). Transformation relationaler Datenbank-Schemas in objektorientierte Schemas gemäß ODMG-93. In Lausen, G., editor, *Datenbanksysteme in Büro, Technik und Wissenschaft (BTW'95)*, Informatik aktuell, pages 111–129. Springer-Verlag.

[Fang et al., 1991] Fang, D., Hammer, J., and McLeod, D. (1991). The Identification and Resolution of Semantic Heterogeneity in Multidatabase Systems. In Kambayashi, Y., Rusinkiewicz, M., and Sheth, A., editors, *Proc. First International Workshop on Interoperability in Multidatabase Systems (IMS'91)*, pages 136–143. IEEE Computer Society Press.

[Gagliardi et al., 1990] Gagliardi, R., Caneve, M., and Oldano, G. (1990). An Operational Approach to the Integration of Distributed Heterogeneous Environments. In Rishe, N., Navathe, S., and Tal, D., editors, *PARBASE-90: Proc. International Conference on Databases, Parallel Architectures, and Their Applications*, pages 368–377. IEEE Computer Society Press.

[Ganter and Wille, 1996] Ganter, B. and Wille, R. (1996). *Formale Begriffsanalyse*. Springer-Verlag, Berlin, Heidelberg.

[Garca-Solaco et al., 1995] Garca-Solaco, M., Saltor, F., and Castellanos, M. (1995). A Structure Based Schema Integration Methodology. In Yu, P. and Chen, A., editors, *Proc. of the 11th IEEE Int. Conf. on Data Engineering (ICDE'95)*, pages 505–512. IEEE Computer Society Press.

[Gardarin et al., 1996a] Gardarin, G., Gannouni, S., Finance, B., Fankhauser, P., Klas, W., Pastre, D., Legoff, R., and Ramfos, A. (1996a). IRO-DB: A Distributed System Federating Object and Relational Databases. In Bukhres, O. and Elmagarmid, A., editors, *Object Oriented Multidatabase Systems: A Solution for Advanced Applications*, chapter 20, pages 684–712. Prentice Hall, Englewood Cliffs, New Jersey.

[Gardarin et al., 1996b] Gardarin, G., Sha, F., and Tang, Z.-H. (1996b). Calibrating the Query Optimizer Cost Model of IRO-DB, an Object-Oriented Federated Database System. In Vijayaraman, T., Buchmann, A., Mohan, C., and Sarda, N., editors, *Proc. of the 22nd Int. Conf. on Very Large Data Bases (VLDB'96)*, pages 378–389. Morgan Kaufmann Publishers.

[Gellert et al., 1992] Gellert, J., Mehta, A., Perl, T., Neuhold, E., and Sheth, A. (1992). Algorithms for Structural Schema Integration. In Ng, P., Ramamoorthy, C. V., Seifert, L., and Yeh, R., editors, *Proc. of The Second International Conference on System Integration (ICSI'92)*, pages 604–614. IEEE Computer Society Press.

[Georgakopoulos et al., 1991] Georgakopoulos, D., Rusinkiewicz, M., and Sheth, A. (1991). On Serializability of Multidatabase Transactions through Forced Local Conflicts. In Cercone, N. and Tsuchiya, M., editors, *Proc. of the 7th IEEE Int. Conf. on Data Engineering (ICDE'91)*, pages 314–323. IEEE Computer Society Press.

[Georgakopoulos et al., 1994] Georgakopoulos, D., Rusinkiewicz, M., and Sheth, A. (1994). Using Tickets to Enforce the Serializability of Multidatabase Transactions. *IEEE Transactions on Knowledge and Data Engineering*, 6(1):166–180.

[Georgakoupoulos et al., 1994] Georgakoupoulos, D., Hornick, M., Krychniak, P., and Manola, F. (1994). Specification and Management of Extended Transactions in a Programmable Transaction Environment. In *Proc. of the 10th IEEE Int. Conf. on Data Engineering (ICDE'94)*, pages 462–473. IEEE Computer Society Press.

[Grant et al., 1993] Grant, J., Litwin, W., Roussopoulos, N., and Sellis, T. (1993). Query Languages for Relational Multidatabases. *VLDB Journal*, 2(2):153–171.

[Gray and Reuter, 1993] Gray, J. and Reuter, A. (1993). *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann Publishers, San Mateo, CA.

[Grefen and Widom, 1995] Grefen, P. and Widom, J. (1995). Integrity Constraint Checking in Federated Databases. Technical Report, Department of Computer Science, University of Twente.

[Grefen and Widom, 1996] Grefen, P. and Widom, J. (1996). Protocols for Integrity Constraint Checking in Federated Databases. In Aberer, K. and Helal, A., editors, *Proceedings of the First IFCIS International Conference on Cooperative Information Systems (CoopIS'96)*, pages 38–47. IFCIS, The International Foundation on Cooperative Information Systems, IEEE Computer Society Press.

[Günthör et al., 1996] Günthör, R., Krause, G., Zimmermann, M., Bever, M., Bapat, A., and Haake, J. (1996). The VORTEL Approach to Application Integration in Distributed Systems. In Yetongnon, K. and Hariri, S., editors, *Parallel and Distributed Computing Systems: Proceedings of the ISCA International Conference (PDCS'96)*, pages 744–750. International Society for Computers and Their Applications.

[Hammer and McLeod, 1993] Hammer, J. and McLeod, D. (1993). An Approach to Resolving Semantic Heterogeneity in a Federation of Autonomous, Heterogeneous Database Systems. *International Journal of Intelligent & Cooperative Information Systems*, 2(1):51–83.

[Hammer et al., 1994] Hammer, J., McLeod, D., and Si, A. (1994). An Intelligent System for Identifying and Integrating Non-Local Objects in Federated Database Systems. In *Proceedings of the 27th Hawaii International Conference on System Sciences*, pages 389–407. IEEE Computer Society Press.

[Hasselbring, 1997a] Hasselbring, W. (1997a). Extending the Schema Architecture of Federated Database Systems for Replicating Information in Hospital Information Systems. In [Conrad et al., 1997a], pages 33–44.

[Hasselbring, 1997b] Hasselbring, W. (1997b). Federated Integration of Replicated Information within Hospitals. *International Journal on Digital Libraries*, 1(3):192–208.

[Heiler et al., 1992] Heiler, S., Haradhvala, S., Zdonik, S., Blaustein, B., and Rosenthal, A. (1992). A Flexible Framework for Transaction Management in Engineering Environment. In [Elmagarmid, 1992], pages 88–121.

[Heiler and Zdonik, 1990] Heiler, S. and Zdonik, S. (1990). Object Views: Extending the Vision. In Liu, M., editor, *Proc. International Conference on Data Engineering (ICDE'90)*, pages 86–93. IEEE Computer Society Press.

[Heimbigner and McLeod, 1985] Heimbigner, D. and McLeod, D. (1985). A Federated Architecture for Information Management. *ACM Transactions on Office Information Systems*, 3(3):253–278.

[Höding et al., 1995] Höding, M., Türker, C., Janssen, S., Sattler, K.-U., Conrad, S., Saake, G., and Schmitt, I. (1995). Föderierte Datenbanksysteme — Grundlagen und Ziele des Projektes **SIGMA**$_{FDB}$. Preprint 12, Fakultät für Informatik, Universität Magdeburg.

[Hohenstein and Körner, 1995] Hohenstein, U. and Körner, C. (1995). Semantische Anreicherung relationaler Datenbanken. In Lausen, G., editor, *Datenbanksysteme in Büro, Technik und Wissenschaft (BTW'95)*, Informatik aktuell, pages 130–149. Springer-Verlag.

[Hohenstein and Plesser, 1996] Hohenstein, U. and Plesser, V. (1996). Semantic Enrichment: A First Step to Provide Database Interoperability. In [Conrad et al., 1996a], pages 3–17.

[Hsiao and Kamel, 1989] Hsiao, D. and Kamel, M. (1989). Heterogeneous Databases: Proliferations, Issues, and Solutions. *IEEE Transactions on Knowledge and Data Engineering*, 1(1):45–62.

[Hsiao et al., 1993] Hsiao, D., Neuhold, E., and Sacks-Davis, R., editors (1993). *Interoperable Database Systems (DS-5), Proc. of the IFIP WG 2.6 Database Semantics Conference (1992)*, IFIP Transactions (A-25). Elsevier Science Publishers (North-Holland).

[Hsiao, 1992a] Hsiao, D. K. (1992a). Federated Databases and Systems: Part I – A Tutorial on Their Data Sharing. *VLDB Journal*, 1(1):127–179.

[Hsiao, 1992b] Hsiao, D. K. (1992b). Federated Databases and Systems: Part II – A Tutorial on Their Recource Consolidation. *VLDB Journal*, 1(2):285–310.

[Huhns et al., 1992] Huhns, M., Jacobs, N., Ksiezyk, T., Shen, W.-M., Singh, M., and Cannata, P. (1992). Enterprise Information Modeling and Model Integration in Carnot. In *Enterprise Integration Modeling, Proc. of the First International Conference*, pages 290–299. MIT Press, Cambridge, Mass.

[Hurson et al., 1994] Hurson, A., Bright, M., and Pakzad, S., editors (1994). *Multidatabase Systems: An Advanced Solution for Global Information Sharing*. IEEE Computer Society Press.

[Hwang et al., 1993] Hwang, S.-Y., Huang, J., and Srivastava, J. (1993). Concurrency Control in Federated Databases: A Dynamic Approach. In Bhargara, B., Finn, T., and Yesha, Y., editors, *CIKM'93 Proc. of the Second International Conference on Information and Knowledge Management*, pages 694–703. ACM Press.

[Idris et al., 1994] Idris, N., Gray, W., and Churchhouse, R. (1994). Providing Dynamic Security Control in a Federated Database. In Bocca, J., Jarke, M., and Zaniolo, C., editors, *Proc. of the 20th Int. Conf. on Very Large Data Bases (VLDB'94)*, pages 13–23. Morgan Kaufmann Publishers.

[Jonscher and Dittrich, 1994] Jonscher, D. and Dittrich, K. (1994). An Approach for Building Secure Database Federations. In Bocca, J., Jarke, M., and Zaniolo, C., editors, *Proc. of the 20th Int. Conf. on Very Large Data Bases (VLDB'94)*, pages 24–35. Morgan Kaufmann Publishers.

[Jonscher and Dittrich, 1993] Jonscher, D. and Dittrich, K. R. (1993). Access Control for Database Federations; a discussion of state-of-the-art. In *Proceedings DBTA Workshop on Interoperability of Database Systems and Database Applications (Fribourg, Switzerland)*, pages 156–178.

[Kachel, 1992] Kachel, G. (1992). *Kontextobjektmodell für Entwurfsanwendungen*. Dissertation. Universität/GH Paderborn.

[Kaul et al., 1991] Kaul, M., Drosten, K., and Neuhold, E. (1991). ViewSystem: Integrating Heterogeneous Information Bases By Object-oriented Views. In Cercone, N. and Tsuchiya, M., editors, *Proc. of the 7th IEEE Int. Conf. on Data Engineering (ICDE'91)*, pages 2–10. IEEE Computer Society Press.

[Kelley et al., 1995] Kelley, W., Gala, S., Kim, W., Reyes, T., and Graham, B. (1995). Schema Architecture of the UniSQL/M Multidatabase System. In [Kim, 1995], chapter 30, pages 621–648.

[Kim, 1995] Kim, W., editor (1995). *Modern Database Systems: The Object Model, Interoperability, and Beyond*. ACM Press/Addison-Wesley Publishing.

[Kim et al., 1995] Kim, W., Choi, I., Gala, S., and Scheevel, M. (1995). On Resolving Schematic Heterogeneity in Multidatabase Systems. In [Kim, 1995], pages 521–550.

[Kim and Seo, 1991] Kim, W. and Seo, J. (1991). Classifying Schematic and Data Heterogeneity in Multidatabase Systems. *IEEE Computer*, 24(12):12–18.

[Köller, 1997] Köller, A. (1997). Dateien in föderierten Datenbanksystemen — Untersuchungen an Systemen zur Fabrikplanung. Preprint 4, Fakultät für Informatik, Universität Magdeburg.

[Krishnamurthy et al., 1991] Krishnamurthy, R., Litwin, W., and Kent, W. (1991). Interoperability of Heterogeneous Databases with Schematic Discrepancies. In Kambayashi, Y., Rusinkiewicz, M., and Sheth, A., editors, *Proc. First International Workshop on Interoperability in Multidatabase Systems (IMS'91)*, pages 144–151. IEEE Computer Society Press.

[Lakshmanan et al., 1996] Lakshmanan, L., Sadri, F., and Subramanian, I. (1996). SchemaSQL: A Language for Interoperability in Relational Multi-database Systems. In Vijayaraman, T., Buchmann, A., Mohan, C., and Sarda, N., editors, *Proc. of the 22nd Int. Conf. on Very Large Data Bases (VLDB'96)*, pages 239–250. Morgan Kaufmann Publishers.

[Larson et al., 1985] Larson, J., Navathe, S., and Elmasri, R. (1985). Schema Integration for Federated Databases. *Bulletin of the IEEE Technical Committee on Data Engineering*, 8(2).

[Larson et al., 1989] Larson, J., Navathe, S., and Elmasri, R. (1989). A Theory of Attribute Equivalence in Databases with Application to Schema Integration. *IEEE Transactions on Software Engineering*, 15(4):449–463.

[Lee and Ling, 1995] Lee, M. and Ling, T. (1995). Resolving Structural Conflicts in the Integration of Entity-Relationship Schemas. In Papazoglou, M., editor, *Proc. of the 14th Int. Conf. on Object-Oriented & Entity-Relationship Modelling (OOER'95)*, number 1021 in Lecture Notes in Computer Science, pages 424–433. Springer-Verlag.

[Liskov et al., 1994] Liskov, B., Day, M., and Shira, L. (1994). Distributed Object Management in Thor. In Özsu, M., Dayal, U., and Valduriez, P., editors, *Distributed Object Management: Papers from the International Workshop on Distributed Object Management (IWDOM, Edmonton, Alberta, Canada, August 19-21, 1992)*, pages 97–91. Morgan Kaufmann Publishers.

[Litwin, 1988] Litwin, W. (1988). From Database Systems to Multidatabase Systems: Why and How. In Gray, W., editor, *Advances in Data Bases: 6th British National Conference on Databases (BNCOD-6)*, pages 161–188. Cambridge University Press, New York.

[Litwin and Abdellatif, 1986] Litwin, W. and Abdellatif, A. (1986). Multidatabase Interoperability. *IEEE Computer*, 19(12):10–18.

[Litwin et al., 1990] Litwin, W., Mark, L., and Roussopoulos, N. (1990). Interoperability of Multiple Autonomous Databases. *ACM Computing Surveys*, 22(3):267–293.

[Mannino and Effelsberg, 1984] Mannino, M. and Effelsberg, W. (1984). Matching Techniques in Global Schema Design. In *Proc. IEEE Int. Conf. on Data Engineering (ICDE'84)*, pages 418–425. IEEE Computer Society Press.

[Manola et al., 1992] Manola, F., Heiler, S. Georgakopoulos, D., Hornick, M., and Brodie, M. (1992). Distrubted Object Management. *International Journal on Intelligent & Cooperative Information Systems*, 1(1).

[Mehrotra et al., 1992a] Mehrotra, S., Rastogi, R., Breitbart, Y., Korth, H., and Silberschatz, A. (1992a). Ensuring Transaction Atomicity In Multidatabase Systems. In *Proc. of the 12th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'92)*, pages 164–175. ACM Press.

[Mehrotra et al., 1992b] Mehrotra, S., Rastogi, R., Breitbart, Y., Korth, H. F., and Silberschatz, A. (1992b). The Concurrency Problem in Multidatabases: Characteristics and Solutions. In Stonebraker, M., editor, *Proc. of the 1992 ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'92)*, pages 288–297. ACM Press.

[Meng and Yu, 1995] Meng, W. and Yu, C. (1995). Query Processing in Multidatabase Systems. In [Kim, 1995], pages 551–572.

[Middleware Project Team, 1996a] Middleware Project Team, O. D. (1996a). OpenDM System Overview. 1. Edition, C-LAB Cooperative Computing & Communication Laboratory.

[Middleware Project Team, 1996b] Middleware Project Team, O. D. (1996b). OpenDM/ODMG-93. Manual, 1. Edition, C-LAB Cooperative Computing & Communication Laboratory.

[Mills, 1992] Mills, J. A. (1992). Large Scale Interoperability and Distributed Transaction Processing. In Ng, P., Ramamoorthy, C., Seifert, L., and Yeh, R., editors, *Proc. of The Second International Conference on System Integration (ICSI'92)*, pages 392–400. IEEE Computer Society Press.

[Missier and Rusinkiewicz, 1997] Missier, P. and Rusinkiewicz, M. (1997). Extending a Multidatabase Manipulation Language to Resolve Schema and Data Conflicts. In Meersman, R. and Mark, L., editors, *Database Application Semantics — Proc. of the IFIP WG 2.6 Working Conf. DS-6 (1995, Atlanta, Georgia)*, pages 93–115. Chapman&Hall, London.

[Morgenstern et al., 1991] Morgenstern, M., Lunt, T., Thuraisingham, B., and Spooner, D. (1991). Security Issues in Federated Database Systems: Panel Contributions. In Landwehr, C. and Jajodia, S., editors, *Database Security, V: Status and Prospects. Results of the IFIP WG 11.3 Workshop on Database Security, West Virginia, USA, 4–7 November, 1991*, pages 131–148. Elsevier Science, North–Holland.

[Moss, 1981] Moss, J. (1981). Nested Transactions: An Approach to Reliable Distributed Computing. Ph.D. Thesis/Technical Report LCS-TR-260, MIT.

[Moss, 1985] Moss, J. (1985). *Nested Transactions: An Approach to Reliable Distributed Computing*. MIT Press, Cambridge, MA.

[Motro, 1987] Motro, A. (1987). Superviews: Virtual Integration of Multiple Databases. *IEEE Transactions on Software Engineering*, 13(7):785–798.

[Mukhopadhyay and Thomas, 1993] Mukhopadhyay, D. and Thomas, G. (1993). Practical Approaches to Maintaining Referential Integrity in Multidatabase Systems. In Schek, H.-J., Sheth, A., and Czejdo, B., editors, *RIDE-IMS'93 Research Issues in Data Engineering: Interoperability in Multidatabase Systems (Wien, 1993)*, pages 42–49. IEEE Computer Society Press.

[Mullen et al., 1992] Mullen, J., Elmagarmid, A., Kim, W., and Sharif-Askary, J. (1992). On the Impossibility of Atomic Commitment in Multidatabase Systems. In Ng, P., Ramamoorthy, C., Seifert, L., and Yeh, R., editors, *Proc. of The Second International Conference on System Integration (ICSI'92)*, pages 625–634. IEEE Computer Society Press.

[Muth and Rakow, 1991] Muth, P. and Rakow, T. (1991). Atomic Commitment For Integrated Database Systems. In Cercone, N. and Tsuchiya, M., editors, *Proc. of the 7th IEEE Int. Conf. on Data Engineering (ICDE'91)*, pages 296–304. IEEE Computer Society Press.

[Muth et al., 1992] Muth, P., Rakow, T., Klas, W., and Neuhold, E. (1992). A Transaction Model for an Open Publication Environment. In [Elmagarmid, 1992], pages 159–218.

[Neuhold and Schrefl, 1988] Neuhold, E. and Schrefl, M. (1988). Dynamic Derivation of Personalized Views. In Bancilhon, F. and DeWitt, D., editors, *Proc. of the 14th International Conference on Very Large Databases (VLDB'88)*, pages 183–194. Morgan Kaufmann Publishers.

[Özsu et al., 1994] Özsu, T., Dayal, U., and Valduriez, P., editors (1994). *Distributed Object Management*. Morgan Kaufmann Publishers.

[Özsu and Valduriez, 1991a] Özsu, T. and Valduriez, P. (1991a). Distributed Database Systems: Where Are We Now? *IEEE Computer*, 24(8):68–78.

[Özsu and Valduriez, 1991b] Özsu, T. and Valduriez, P. (1991b). *Principles of Distributed Database Systems*. Prentice-Hall, Eaglewood Cliffs, New Jersey.

[Özsu and Valduriez, 1994] Özsu, T. and Valduriez, P. (1994). Distributed Data Management: Unsolved Problems and New Issues. In Casavant, T. and Singhal, M., editors, *Readings in Distributed Computing Systems*, chapter 10, pages 512–544. IEEE Computer Society Press.

[Papakonstantinou et al., 1996] Papakonstantinou, Y., Garcia-Molina, H., and Ullman, J. (1996). MedMaker: A Mediation System Based on Declarative Specifications. In Su, S., editor, *Proceedings of the 12th International Conference on Data Engineering (ICDE'96)*, pages 132–141. IEEE Computer Society Press.

[Papazoglou et al., 1990] Papazoglou, M., Marinos, L., and Bourbakis, N. (1990). Distributed Heterogeneous Information Systems & Schema Transformation. In Rishe, N., Navathe, S., and Tal, D., editors, *PARBASE-90: Proc. International Conference on Databases, Parallel Architectures, and Their Applications*, pages 388–397. IEEE Computer Society Press.

[Parent et al., 1996] Parent, C., Spaccapietra, S., and Devogele, T. (1996). Conflicts in Spatial Database Integration. In Yetongnon, K. and Hariri, S., editors, *Parallel and Distributed Computing Systems: Proceedings of the ISCA International Conference (PDCS'96)*, pages 772–778. International Society for Computers and Their Applications.

[Pernul, 1993] Pernul, G. (1993). Canonical Security Modeling for Federated Databases. In [Hsiao et al., 1993], pages 207–222.

[Pitoura et al., 1995] Pitoura, E., Bukhres, O., and Elmagarmid, A. (1995). Object-Orientation in Multidatabase Systems. *ACM Computing Surveys*, 27(2):141–195.

[Radeke, 1995] Radeke, E. (1995). *Federation and Migration among Database Systems*. Dissertation. Universität/GH Paderborn.

[Radeke and Scholl, 1994a] Radeke, E. and Scholl, M. (1994a). Framework for Object Migration in Federated Database Systems. In *Proc. of the 3rd Int. Conf. on Parallel and Distributed Information Systems (PDIS'94)*, pages 187–194. IEEE Computer Society Press.

[Radeke and Scholl, 1994b] Radeke, E. and Scholl, M. (1994b). Object Migration in Federated Database Systems. Cadlab-Report 3/94, Cadlab Paderborn.

[Radeke and Scholl, 1995] Radeke, E. and Scholl, M. (1995). Functionality for Object Migration among Distributed, Heterogenous, Autonomous Databases Systems. In Bukhres, O., Özsu, T., and Shan, M.-C., editors, *Proc. of the 5th Int. Workshop on Research Issues in Data Engineering: Distributed Object Management (RIDE-DOM'95)*, pages 58–66. IEEE Computer Society Press.

[Ramamritham and Chrysanthis, 1996] Ramamritham, K. and Chrysanthis, P. K. (1996). A Taxonomy of Correctness Criteria in Database Applications. *VLDB Journal*, 5(1):85–97.

[Ramfos et al., 1995] Ramfos, A., Fessy, J., Finance, B., and Smahi, V. (1995). IRO-DB: A Solution for Computer Integrated Manufacturing Applications. In *Proc. Third International Conference on Cooperative Information Systems (CoopIS'95)*.

[Reddy et al., 1995] Reddy, M. P., Prasad, B. E., and Gupta, A. (1995). Formulating Global Integrity Constraints During Derivation of Global Schema. *Data & Knowledge Engineering*, 16(3):241–268.

[Reddy et al., 1994] Reddy, M. P., Prasad, B. E., Reddy, P. G., and Gupta, A. (1994). A Methodology for Integration of Heterogeneous Databases. *IEEE Transactions on Knowledge and Data Engineering*, 6(6):920–933.

[Rusinkiewicz and Czejdo, 1985] Rusinkiewicz, M. and Czejdo, B. (1985). Query Transformation in Heterogeneous Distributed Database Systems. In *Proc. Fifth International Conference on Distributed Computing Systems*, pages 300–307. IEEE Computer Society Press.

[Saltor et al., 1991] Saltor, F., Castellanos, M., and García-Solaco, M. (1991). Suitability of Data Models as Canonical Models for Federated Databases. *ACM SIGMOD Record*, 20(4):44–48.

[Saltor et al., 1993] Saltor, F., Castellanos, M., and García-Solaco, M. (1993). Overcoming Schematic Discrepancies in Interoperable Databases. In [Hsiao et al., 1993], pages 191–205.

[Savasere et al., 1991] Savasere, A., Sheth, A., Gala, G., Navathe, S., and Markus, H. (1991). On Applying Classification to Schema Integration. In Kambayashi, Y., Rusinkiewicz, M., and Sheth, A., editors, *Proc. First International Workshop on Interoperability in Multidatabase Systems (IMS'91)*, pages 258–261. IEEE Computer Society Press.

[Schaad, 1996] Schaad, W. (1996). *Transaktionsverwaltung in heterogenen, f"oderierten Datenbanksystemen*, volume 12 of *Dissertationen zu Datenbanken und Informationssystemen DISDBIS*. infix-Verlag, Sankt Augustin.

[Schaad and Schek, 1993] Schaad, W. and Schek, H.-J. (1993). Federated Transaction Management Using Open Nested Transactions. In *Proc. of the DBTA Workshop on Interoperability of Database Systems and Database Applications (Fribourg, Schwitzerland)*.

[Schaad et al., 1995] Schaad, W., Schek, H.-J., and Weikum, G. (1995). Implementation and Performance of Multi-level Transaction Management in a Multidatabase Environment. In Bukhres, O., Özsu, T., and Shan, M.-C., editors, *Proc. of the 5th Int. Workshop on Research Issues on Data Engineering: Distributed Object Management (RIDE-DOM'95)*, pages 108–115. IEEE Computer Society Press.

[Schek et al., 1991] Schek, H.-J., Weikum, G., and Schaad, W. (1991). A Multi-level Transaction Approach to Federated DBMS Transaction Management. In Kambayashi, Y., Rusinkiewicz, M., and Sheth, A., editors, *Proc. of the 1st Int. Workshop on Interoperability in Multidatabase Systems (IMS'91)*, pages 280–287. IEEE Computer Society Press.

[Schmitt, 1995] Schmitt, I. (1995). Flexible Integration and Derivation of Heterogeneous Schemata in Federated Database Systems. Preprint 10, Fakultät für Informatik, Universität Magdeburg.

[Schmitt, 1998] Schmitt, I. (1998). *Schemaintegration für den Entwurf föderierter Datenbanken*. Dissertation, Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik.

[Schmitt and Conrad, 1997] Schmitt, I. and Conrad, S. (1997). Restructuring Class Hierarchies for Schema Integration. In Topor, R. and Tanaka, K., editors, *Database Systems For Advanced Applications'97 (DASFAA'97)*, pages 411–420. World Scientific Publishing, Singapore.

[Schmitt and Saake, 1995] Schmitt, I. and Saake, G. (1995). Managing Object Identity in Federated Database Systems. In Papazoglou, M., editor, *Proc. of the 14th Int. Conf. on Object-Oriented and Entity-Relationship Modeling (OOER'95)*, number 1021 in Lecture Notes in Computer Science, pages 400–411. Springer-Verlag.

[Schmitt and Saake, 1996a] Schmitt, I. and Saake, G. (1996a). Flexible Generation of Global Integrated Schemata Using GIM. In [Conrad et al., 1996a], pages 29–43.

[Schmitt and Saake, 1996b] Schmitt, I. and Saake, G. (1996b). Integration of Inheritance Trees as Part of View Generation for Database Federations. In Thalheim, B., editor, *Proc. of the 15th Int. Conf. on Conceptual Modelling (ER'96)*, number 1157 in Lecture Notes in Computer Science, pages 195–210. Springer-Verlag.

[Schmitt and Saake, 1996c] Schmitt, I. and Saake, G. (1996c). Schema Integration and View Generation by Resolving Intensional and Extensional Overlappings. In Yetongnon, K. and Hariri, S., editors, *Parallel and Distributed Computing Systems: Proceedings of the ISCA International Conference (PDCS'96)*, pages 751–758. International Society for Computers and Their Applications.

[Schrefl and Neuhold, 1988] Schrefl, M. and Neuhold, E. (1988). Object Class Definition by Generalization Using Upward Inheritance. In *Proceedings 4th International Conference on Data Engineering (ICDE'88)*, pages 4–13. IEEE Computer Society Press.

[Schrefl and Neuhold, 1990] Schrefl, M. and Neuhold, E. (1990). A Knowledge-based Approach to Overcome Structural Differences in Object Oriented Database Integration. In Meersman, R., Shi, Z., and Kung, C.-H., editors, *Artificial Intelligence in Databases and Information Systems (DS-3), Proc. of the IFIP WG 2.6/8.1 Working Conference, Guangzhou, China, 1988*, IFIP Transactions, pages 265–304. Elsevier Science Publishers (North-Holland).

[Seligman and Kerschberg, 1993] Seligman, L. and Kerschberg, L. (1993). Knowledge-base/Database Consistency in a Federated Multidatabase Environment. In Schek, H.-J., Sheth, A., and Czejdo, B., editors, *RIDE-IMS'93 Research Issues in Data Engineering:*

Interoperability in Multidatabase Systems (Wien, 1993), pages 18–25. IEEE Computer Society Press.

[Shan et al., 1995] Shan, M.-C., Ahmed, R., Davis, J., Du, W., and Kent, W. (1995). Pegasus: A Heterogeneous Information Management System. In [Kim, 1995], pages 664–682.

[Sheth and Larson, 1990] Sheth, A. and Larson, J. (1990). Federated Database Systems for Managing Distributed, Heterogenous, and Autonomous Databases. *ACM Computing Surveys*, 22(3):183–236.

[Sidell et al., 1996] Sidell, J., Aoki, P., Barr, S., Sah, A., Staelin, C., Stonebraker, M., and Yu, A. (1996). Data Replication in Mariposa. In Su, S., editor, *Proceedings of the 12th International Conference on Data Engineering (ICDE'96)*, pages 485–494. IEEE Computer Society Press.

[Spaccapietra and Parent, 1994] Spaccapietra, S. and Parent, C. (1994). View Integration: A Step Forward in Solving Structural Conflicts. *IEEE Transactions on Knowledge and Data Engineering*, 6(2):258–274.

[Spaccapietra et al., 1992] Spaccapietra, S., Parent, C., and Dupont, Y. (1992). Model Independent Assertions for Integration of Heterogeneous Schemas. *VLDB Journal*, 1(1):81–126.

[Spiliopoulou, 1996a] Spiliopoulou, M. (1996a). A Calibration Mechanism Identifying the Optimization Technique of a Multidatabase Participant. In Yetongnon, K. and Hariri, S., editors, *Parallel and Distributed Computing Systems: Proceedings of the ISCA International Conference (PDCS'96)*, pages 633–639. International Society for Computers and Their Applications.

[Spiliopoulou, 1996b] Spiliopoulou, M. (1996b). Query Optimization in Multidatabases: A Calibration Mechanism Identifying the Optimization Principles of a Participant Database. In [Conrad et al., 1996b], pages 91–95.

[Spiliopoulou and Conrad, 1997] Spiliopoulou, M. and Conrad, S. (1997). FACE: A Flexible Approach for Constraint Enforcement and Compensation of Constraaint Violation in Federated Databases. Technical Report ITI-97-01, Institut für Technische Informationssysteme, Universität Magdeburg.

[Stonebraker et al., 1996] Stonebraker, M., Aoki, P. M., Litwin, W., Pfeffer, A., Sah, A., Sidell, J., Staelin, C., and Yu, A. (1996). Mariposa: A Wide-area Distributed Database System. *VLDB Journal*, 5(1):48–63.

[Stonebraker et al., 1994] Stonebraker, M., Devine, R., Kornacker, M., Litwin, W., Pfeffer, A., Sah, A., and Staelin, C. (1994). An Economic Paradigm for Query Processing and Data Migration in Mariposa. In *Proceedings of 3rd International Conference on Parallel and Distributed Information Systems (PDIS'94)*. IEEE Computer Society Press.

[Tari and Fernandez, 1997] Tari, Z. and Fernandez, G. (1997). Security Enforcement in the DOK Federated Database System. In Samarati, P. and Sandhu, R., editors, *Database Security, X: Status and Prospects, Proceedings of the IFIP WG 11.3 Working Conference on Database Security, Como, Italy, 22–24 July, 1996*, pages 23–42. Chapman&Hall, London.

[Templeton et al., 1995] Templeton, M., Henley, H., Maros, E., and Van Buer, D. J. (1995). InterViso: Dealing with the Complexity of Federated Database Access. *VLDB Journal*, 4(2):287–317.

[Thomas et al., 1990] Thomas, G., Thompson, G. R., Chung, C.-W., Barkmeyer, E., Carter, F., Templeton, M., Fox, S., and Hartman, B. (1990). Heterogeneous Distributed Database Systems for Production Use. *ACM Computing Surveys*, 22(3):237–266.

[Tresch and Scholl, 1993] Tresch, M. and Scholl, M. (1993). Schema Transformation Processors for Federated Objectbases. In Moon, S. and Ikeda, H., editors, *Proc. of the 3rd Int. Conf. on Database Systems for Advanced Applications (DASFAA'93)*, pages 37–46. World Scientific Press.

[Türker and Conrad, 1997] Türker, C. and Conrad, S. (1997). Towards Maintaining Integrity of Federated Databases. In *Proc. Basque International Workshop on Information Technology BIWIT'97 (Biarritz, France, 2.-4. Juli 1997)*. IEEE Computer Society Press.

[Wächter and Reuter, 1990] Wächter, H. and Reuter, A. (1990). Grundkonzepte und Realisierungsstrategien des ConTract-Modells. *Informatik: Forschung und Entwicklung*, 5(4):202–212.

[Wächter and Reuter, 1992] Wächter, H. and Reuter, A. (1992). The ConTract Model. In [Elmagarmid, 1992], pages 219–263.

[Weikum, 1991] Weikum, G. (1991). Principles and Realization Strategies of Multilevel Transactions Management. *ACM Transactions on Database Systems*, 16(1):132–180.

[Weikum and Schek, 1992] Weikum, G. and Schek, H.-J. (1992). Concepts and Applications of Multi-level Transactions and Open Nested Transactions. In [Elmagarmid, 1992], pages 515–553.

[Wille, 1992] Wille, R. (1992). Concept Lattices and Conceptual Knowledge Systems. *Computer & Mathematics with Applications*, 23(6-9):493–515.

[Wolski and Veijalainen, 1990] Wolski, A. and Veijalainen, J. (1990). 2PC Agent Method: Achieving Serializability in Presence of Failures in a Heterogeneous Multidatabase. In Rishe, N., Navathe, S., and Tal, D., editors, *PARBASE-90: Proc. International Conference on Databases, Parallel Architectures, and Their Applications*, pages 321–330.

[Zhang and Elmagarmid, 1993] Zhang, A. and Elmagarmid, A. K. (1993). A Theory of Global Concurrency Control in Multidatabase Systems. *The VLDB Journal*, 2(3):331–360.

[Zhang and Jing, 1993] Zhang, A. and Jing, J. (1993). On Structural Features of Global Transactions in Multidatabase Systems. In Schek, H.-J., Sheth, A., and Czejdo, B., editors, *RIDE-IMS'93 Research Issues in Data Engineering: Interoperability in Multidatabase Systems (Wien, 1993)*, pages 199–206. IEEE Computer Society Press.

[Zhou et al., 1996] Zhou, G., Hull, R., and King, R. (1996). Generating Data Integration Mediators that Use Materialization. *Journal of Intelligent Information Systems*, 6(2/3):199–222.

[Zhu and Larson, 1996] Zhu, Q. and Larson, P.-Å. (1996). Global Query Processing and Optimization in the CORDS Multidatabase System. In Yetongnon, K. and Hariri, S., editors, *Parallel and Distributed Computing Systems: Proceedings of the ISCA International Conference (PDCS'96)*, pages 640–646. International Society for Computers and Their Applications.