Flexible Integration and Derivation of Heterogeneous Schemata in Federated Database Systems [1]

Ingo Schmitt

Universität Magdeburg
Fakultät für Informatik
Postfach 4120, D–39016 Magdeburg
Germany

Email: schmitt@iti.cs.uni-magdeburg.de

Tel.: ++49-391-67-12994
Fax: ++49-391-67-12020

November 1995

[1] This research was partially supported by the German Country Sachsen-Anhalt under FKZ: 1987A/0025 ("Föderierung heterogener Datenbanksysteme und lokaler Datenhaltungskomponenten zur systemübergreifenden Integritätssicherung").

Abstract

In this report concepts for the process of schema integration and transformation in Federated Database Systems (FDBS) are suggested, which are flexible in the integration of local and in the derivation of external schemata. An FDBS has to support different data models of local database systems. Furthermore, it has to offer a means to derive external schemata, which can differ in their underlying data models and in the way of representing the federated data according to specific views of global applications. In order to move local applications from their local schemata to external schemata above the federation layer, these schemata must no be different. Concepts to derive such external schemata are introduced. In addition to this, applying the suggested concepts, the process of integration and transformation can be done in a methodological manner. In this report, the use of a semantically poor data model as the canonical data model and a two-phase-restructuring process for the schema integration and transformation are motivated. These concepts are important steps to fit the requirements above.

Keywords: federated database systems,
canonical data model,
integration transparency,
schema integration,
schema transformation,
two-phase-restructuring,
Generic Integration Model.

Acknowledgements

Thanks to Gunter Saake, Michael Höding and Can Türker for many creative discussions. I also have benefited from linguistic advice provided by Victoria Forrest-Hampson, Mark Krogel and Can Türker.

Contents

Chapter 1

Introduction

In most companies, a huge amount of data is stored and managed by different database systems. The databases of these systems were designed independently from each other. Local applications use such systems and the data managed by them. New applications often need global and transparent access to the distributed databases. Due to the heterogeneity of the legacy database systems, the data models, the data representations and the semantic heterogeneity, an access as mentioned above is difficult to obtain. A physical integration approach means that legacy systems have to be redesigned, which would be too costly. Therefore, a logical (virtual) integration approach is more suitable to fulfill the integration task. Global applications can then interact with a homogeneous, federated schema defined explicitly and the heterogeneous database systems and local applications continue to exist. In this context one talks of a *Federated Database System (FDBS)* that couples legacy database systems tightly.

There are many approaches and prototypes of FDBS's, for instance Multibase [LR82], MERMAID [Tem83], Pegasus [Raf91], MYRIAD [CGH$^+$93], OpenDM [BEK$^+$94], IRO-DB [GGF$^+$95, BFN94] and ZOO$_{IFI}$ [Här94]. A comparative analysis of methodologies for database schema integration is given in [BLN86]. Aspects of *flexible* schema integration and transformation regarding *different data models* and *views* are usually not dealt with in sufficient depth. Therefore, this report focuses on the following questions:

- How can the complexity of the homogenizing and integrating process of the component schemata be reduced?
- How should the schema integration and transformation process be carried out to enable flexibility with regard to varying data models and views, respectively, on external and component level?

Solving both questions at the same time seems to be contradictory, but the next chapters show how to reconcile them.

The selection of the right data model for the federated schema, often called *canonical data model*, is essential with regard to the questions above [BLN86]. The use of a semantically rich data model complicates the integration process and restricts the flexibility to derive external, federated schemata, which are equivalent to local schemata. Therefore, the use of a semantically poor data model as canonical data model for integration purposes is proposed here.

The next chapter gives some fundamental principles of the federation of database systems. In Chapter 3 requirements on the integration and transformation of schemata in FDBS's are formulated. Chapter 4 and Chapter 5, respectively, show how the use of a semantically rich data model and a semantically poor data model, respectively, meets the requirements. In Chapter 6 some characteristics of a suitable semantically poor data model are presented, which has been named *Generic Integration Model (GIM)*. The process of integration and restructuring of schemata following the two-phase-restructuring method is outlined in Chapter 7. Finally, Chapter 8 concludes by summing up and giving a short outlook to the possible future development of the area.

Chapter 2

Foundations

In [SL90] an overview of the topic of FDBS's is given and a generally accepted five-level schema architecture is introduced:

1. The *local schemata* are the schemata given by the existing local databases and are based on specific data models.

2. The *component schemata* are the local schemata transformed into the data model of the federated schema. On this level the data model heterogeneity is overcome.

3. The distinction between data allowed to be federated and those which are not allowed to be federated is expressed in the *export schemata*. Only exported schema elements and their data are accessible by the federation layer.

4. The *federated schema* plays two roles:

   - Integrated schema of the component schemata
   - Interface for global applications

5. The *external schemata* serve as specific interfaces for global applications. A specific interface means a schema which represents a specific view or the federated schema which is based on a specific data model.

Now, some terms have to be defined which are used in the next chapters.

- The term *universe of discourse (UoD)* is used to mean an arbitrary portion of the real world to be represented in a conceptual schema [BL84].

- The term *semantical relativism* is used to express a specific view of an UoD. Such a view is subjective, i.e. it depends on the way an observer looks at the UoD. Database systems including FDBS's have to support semantic relativism. It is represented by external schemata of the ANSI SPARC three-level-architecture [TK78]. The dimension of semantical relativism also depends on the underlying data model.

For instance, using a semantically rich data model like an object-oriented data model allows more semantical relativism to be expressed than using the relational data model [SPD92, BLN86]. In an object-oriented data model there are many concepts supporting semantical relativism, e.g.:

- *type constructors* allowing complex data types instead of flat relations;
- *inheritance hierarchies*;
- *embedding objects* allowing modeling of static aggregation relations;
- *directions of references*;
- *operational methods* allowing specific behavior to be described;
- *derived attributes*.

The semantical relativism concepts increase the potential for complex integration conflicts. For example, in Figure 2.1 two local object-oriented schemata are introduced. An object-oriented data model is used, which is similar to C++ and ODMG DDL.

```
class LCS1.Book {
        public:
          string title;
          number isbn;
          ref Set<Author> authors;
          number no-authors = card(authors);
  }

class LCS1.Author {
        public:
          string name;
  }

class LCS2.Publication {
        private:
          string title;
          date   year;
          string type;
          Set<string> authors;
        public:
          string get-title();
          ...
  }

class LCS2.Book:Publication {
        public:
          number isbn;
  }
```

Figure 2.1: Local conceptual object-oriented schema LCS1 and LCS2

In the following list the specific semantical relativism information of the classes defined in Figure 2.1 is given:

1. *type constructor*: `LCS2.Publication.authors` with the type: `set of string`;
2. *inheritance hierarchy*: `LCS2.Book` is a subclass of `LCS2.Publication`;
3. *directions of references*: from `LCS1.Book.authors` to `Author`, but not vice versa;
4. *operational methods*: `LCS2.Publication.get-title()`;
5. *derived attributes*: `LCS1.Book.no-authors`;
6. *others*: public, private classification of attributes and methods in each class;

- In general, a *data schema transformation* is a pair of mappings. The first mapping, called the *data schema mapping*, describes how to produce the target schema from a source schema. The second mapping, called the *database instance mapping*, describes how a valid database instance for the target data schema is obtained from a valid database instance of the source data schema [Tro93].

- A data schema transformation is *reversible* and *total* and *surjective*, respectively, iff the mappings are reversible and total and surjective, respectively. Surjectivity in this context means totality of the inverse mappings.

- A data schema transformation is *lossless* (free of loss) iff it is reversible, total and surjective.

- The term *lossless schema mapping* is used iff the feature of losslessness only concerns the data schema mapping. It corresponds to the term *preservation of information capacity* in [Hul86].

- In general, in FDBS's exist stratified schema transformations. For example, the transformation from a local schema to a external schema is composed of two transformations, from the local schema to the federated schema and from the federated schema to the external schema. If there exists such a composed, lossless data schema transformation and a composed, lossless schema mapping, respectively, between two schemata and both of them offer the same interface to applications, which includes the same underlying data model and same names, then they are *data schema equivalent* and *schema equivalent*, respectively.

Chapter 3

Requirements on Federation

In order to compare the federation with a semantically rich data model to the federation with a semantically poor data model, requirements on federation are presented in this chapter. The requirements listed below are not complete. So the export schemata are omitted, because they are not relevant to the approach presented here. Only requirements concerning the two questions formulated in Chapter 1 are considered here. The requirements are classified regarding the data schema transformations from a local to the federated schema, from the federated to an external schema and the process of schema integration and transformation in general:

1. The FDBS has to support the federation of local database systems no matter which local data models are used. Total and reversible data schema transformations have to exist from the local schemata to the integrated federated schema independent from underlying local data models.

2. Process of schema integration and transformation:

   (a) During the integration process the reconciliation of the component schemata has to be adequate and correct. Any element of the component schemata needs an equivalent in the federated schema (total and reversible data schema transformation). The federated schema has to look like a schema, which could be designed by an expert regarding the global view of the UoD before local schemata are designed. Furthermore, the reconciliation contains also the reconciliation of integrity conditions within the component schemata. [BLN86] demand completeness and correctness, minimality and understandability on the merging process of component schemata.

   (b) In general, an FDBS has to be a dynamical system. New local schemata can be integrated into an existing federated schema or local schemata can be changed. The integration process has to bear such dynamics in mind.

   (c) The integration process cannot be done completely automatically [BLN86]. Human decisions are unavoidable. A methodical and comprehensible way for integration would make the complex process easier and, hence, more correct.

(d) There are performance aspects to be considered on the transformation of queries and data caused by running global applications. Many schema levels between a global application and a local database system decrease the performance of response time on application requests. If a global application interacts with an external schema representing a view on another external schema, which represents the federated schema but in a different data model, then the data has to be transported through five processors. This aspect seems to show that the demand for flexibility contradicts the demand for performance.

3. Schema transformations to external schemata:

   (a) A flexible data schema transformation from the federated schema to an external schema, which could be based on any data model, has to be functional and independent from the data models of the local schemata.

   (b) Global applications need specific external schemata corresponding to their specific perceptions (semantical relativism) of the UoD. In this context there exists the demand for logical data independence. Two different situations are possible.

      i. In order to migrate a local application from local to global level, the FDBS has to enable the definition of an external schema which is data schema equivalent or schema equivalent to the local schema. Most publications on federation of databases, e.g. [BLN86], do not consider this aspect. [KLK91] use the term *integration transparency* in this context. If an external schema is schema equivalent to a local schema then additional database instances of other local database systems federated by the FDBS are accessible.

      ii. New global applications need specific, integrated and consistent views to the federated data. The consistency refers also to the different integrity conditions of the local schemata or the local data models to be reconciled. Total and reversible data schema transformations from the local schemata to the external schema must exist.

As mentioned in the previous chapter there are many approaches to FDBS's. The selection of the right data model for the federated schema, often called *canonical data model*, is essential with regard to the requirements above [BLN86]. Most approaches choose the relational or an object-oriented data model as the canonical data model. In many publications, for instance [SCG91], it is argued that the usage of the relational data model makes the integration of local schemata with an underlying object-oriented data model difficult because of the missing semantic expressiveness. Owing to this, the most recent approaches selected an object-oriented data model as the canonical model. In [Här94] the choice of an object-oriented canonical data model is further motivated.

The next chapter shows how the usage of an object-oriented data model as the canonical data model fits the requirements listed above.

Chapter 4

Federation with an Object-Oriented Data Model

The most frequently used object-oriented data model as canonical data model is the ODMG-93 object model [Cat94]. In order to be able to use it for integration most prototypes (e.g. [GGF$^+$95] and [BEK$^+$94]) extend the data model and the languages of ODMG-93.

Global applications interact with the OML and OQL. The ODMG-93 proposal intends to become a quasi-standard. This chapter shows how far the usage of an object-oriented data model meets the requirements.

1. The total and reversible data schema transformation from a local schema, which is based on any data model, to an object-oriented schema is feasible. Due to the semantical richness of the object-oriented data model in comparison with most other data models most semantics can be transformed. If the local data model is semantically rich (e.g. EER-model [Hoh93]) then the transformation can be difficult because of the many concept mappings. On the other hand, if the local data model is semantically poor then the transformation has to be accompanied by semantical enrichness [HK95].

2. Process of schema integration and transformation:

   (a) Many data model concepts increase the potential for conflicts, which have to be reconciled. In [Kim95] many kinds of conflicts are described, which can occur in using an object-oriented data model. More conflicts make the integration process more complex. So, it is difficult to guarantee an federated schema to be adequate and correct.

   For instance, the heterogeneity between the classes of the example introduced in Chapter 1 is expressed in the following conflicts:

   - conflicts, which are independent of semantical relativism:

  - attribute versus class:
    `LCS2.Publication.authors` versus `LCS1.Author`
  - meta-conflict: the objects of class `LCS1.Book` correspond to objects of `LCS2.Publication` with the value "Book" of attribute `LCS2.Publication.type`
  - description-conflict: there exists no corresponding attribute in `LCS1.Book` to attribute `LCS2.Publication.year`
- conflicts, which are dependent of semantical relativism:
  - visibility-conflict: e.g. `LCS1.Book.title` is public versus `LCS2.Publication.title` is private
  - inheritance-hierarchy-conflict: the extensions of `LCS2.Publication`, `LCS2.Book` and `LCS1.Book` are overlapping (depicted in Figure 4.1)

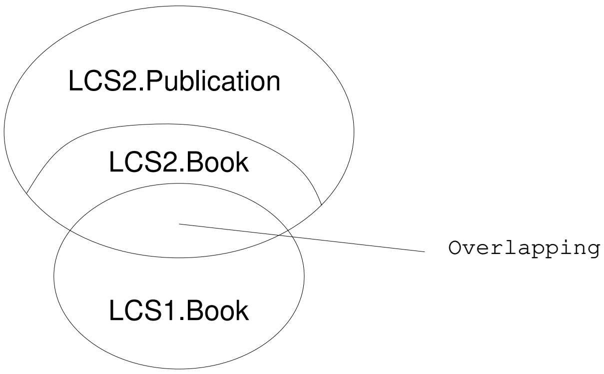


Figure 4.1: Extensional overlapping

One possible federated schema, which reconciles most conflicts, is described in Figure 4.2. The extensions of any class is described by set-operations on class-extensions of the local schemata. The meta-conflict regarding the attribute `type` mentioned above is solved by introducing a constant-valued attribute type="Book" for objects, which are instances of the class `LCS1.Book`. The visibility-conflict cannot be solved, because the extensions of `LCS1.Book` and `LCS2.Publication` are united.

One can see that already in this little schemata the test of correctness of the federated schema is very difficult.

(b) The complexity of the reconciliation restricts the dynamics of the integration process in accordance with new local database systems to be integrated and changes of local schemata. An important but still unsolved question is how much would the integration of a new local schema influence existing transformations? This interesting question is not detailed here.

The complex nature of evolution in object-oriented systems is also described in [Tre95].

```
class IS.Publication {
    public:
      string title;
      string type;
      ref Set<Author> authors;
      number no-authors = card(authors);
  }

class IS.Book:Publication {
    public:
      number isbn;
  }

class IS.Publication-year:Publication {
    private:
      date year;
    public:
      string get-title();
  }

class IS.Author {
    public:
      string name;
  }

class IS.Book-Publication:Publication-year,Book {}

with extensions: IS.Publication      = LCS1.Book ∪ LCS2.Publication
                 IS.Book             = LCS1.Book ∪ LCS2.Book
                 IS.Publication-year = LCS2.Publication
                 IS.Book-Publication = LCS2.Book
constraints:     x ∈ LCS1.Book → x.type = "Book"
```

Figure 4.2: Federated object-oriented schema IS

(c) Up to now, to the author no comprehensible, complete and comprehensive integration methodology using an object-oriented data model is known. Therefore, the integration using an object-oriented data model remains an intuitive and non-systematic work for experts.

(d) Global applications can interact directly with the federated schema. In this case the maximum of three processors exists (see also [SL90]: from local to component schema; from component to export schema; from export to federated schema).

3. Schema transformations to external schemata:

(a) A data schema transformation from a schema, which is based on an object-oriented data model, to a schema, which is based on a semantically poor data model like the relational model, is feasible. This transformation causes semantical loss during the normalization process. In general, the normalization process is accompanied by splitting classes (or relations).

The transformation to a schema, which is based on another semantically rich data model like the EERM [Hoh93] is possible, but in general a very complex task. There are many concepts in the source and target data model to be transformed. Many transformation variants can exist.

(b) The topic of the derivation of views with specific semantical relativism from an object-oriented schema is a very complex and also a current problem. Many publications, e.g. [Heu94], show the complicated nature of this task. The problems are caused by using an schema, which is based on an object-oriented data model as the base schema for views.

> Selecting a semantically rich data model like the object-oriented model as canonical data model allows too much semantical relativism to be expressed in the federated schema.

This leads to a base schema, which represents already a specific way of looking at the UoD. If a view (external schema) has to be derived from the object-oriented base schema, the semantical relativism has to be changed instead of being added. Changing semantical relativism means a complex restructuring of the schema. For instance, attribute values become independent objects or inheritance relationships change.

Both, the derivation of an external schema that is data schema equivalent or schema equivalent to a local schema as well as the derivation of an integrating external schema with specific semantical relativism are therefore very complex tasks. Only a small part of these tasks can be done automatically. Sometimes, there exists no way to derive an external schema from the federated schema, which is data schema equivalent to a given local schema.

For example, it is impossible to derive the extension of the class `Book` of an external schema data which should be identical to the local schema `LCS1` from the federated schema depicted in Figure 4.2. This problem is caused by the loss of location information about federated objects, which is required by [Dat90] (location transparency).

The author believes that an object-oriented data model is only good for representing a specific view for an application because it offers many concepts on the expression of semantical relativism. However, the semantical relativism in object-oriented data models complicates the integration process immensely. In addition to this the formulation of other views, which are based on an object-oriented schemata, is also a very complex task.

Chapter 5

Federation with a Generic Integration Model

The data model of the federated schema has to be a relatively simple data model with as few as possible concepts allowing semantical relativism to be expressed. The concepts have to be orthogonal, because a lack of model orthogonality embodies dependencies between the concepts. Such dependencies cause a complex integration task [BL84]. In this report such a data model is called a *Generic Integration Model GIM*. [SCG91] suggest the usage of a canonical model, which should have just one basic structure rather than two or more. In [McL93] the author describes the kernel object data model (KODM) as the suitable interoperation data model.

If the data model of the federated schema is the GIM, which is tailored to meet the integration requirements, then the federated schema should not be the interface to global applications. They should only interact with external schemata. Therefore, the GIM has to be defined in such a way that it can be the basis for transformations to other data models with more specific semantical relativism concepts. From now on the term *integrated schema* will be used instead of the term federated schema.

Building a federated database with the GIM as canonical data model encompasses the following steps (described in more detail in Chapter 7):

1. The data schema transformation of a local schema to a component schema is characterized by a semantic reduction (normalization). The loss of semantical relativism information has to be rescued into a specific *semantical relativism schema* (*SR-schema*).

2. Now the integration of the component schemata can be carried out better than it would be possible using an object-oriented data model.

3. After the integration process the integrated schema can be enriched with semantical relativism information (e.g. with information from an SR-schema obtained in the first step) in order to define external schemata for global applications.

Introducing a new data model in the field of database research is not a new idea. Most of them do not survive. Therefore, the author tries to find an existing data model which meets the needs (as described in more detail in Chapter 6) rather than to design a new one. Before requirements on the GIM will be described it is necessary to show how the GIM approach fulfills the requirements described in Chapter 3:

1. The data schema transformation from a local data schema, which is based on an arbitrary data model, to the component data schema, which is based on GIM, is accompanied by splitting the local schema. Schema elements expressing semantical relativism information have to be separated (into an SR-schema) from the local schema. The remainder has to be transformed to the component schema. It is very important to transform all local integrity constraints (explicit and data model implicit constraints) to the component schema. They are no semantical relativism information.

   There has to exist a reverse and total data schema transformation (or better a merge operation) from the elements of a component schema and its corresponding SR-schema elements to all schema elements of the corresponding local schema.

2. Process of schema integration and transformation:

   (a) Due to the normalized (freed from semantical relativism information) component schemata relatively little conflicts have to be solved to integrate them. This makes the integration process easier and therefore more adequate and correct. [BLN86] pointed out that in using a simple data model fewer type conflicts are caused. Note that also during the integration of the component schemata integrity constraints have to be considered.

   (b) Due to the little conflict potential in GIM it is improbable that the integration of new component schemata to an existing integrated schema or the reintegration of changed component schemata cause more problems than using a semantically rich data model.

   (c) Less conflicts makes the integration process easier. The interaction during the integration process with experts can be developed in a methodical and comprehensible way. This way will be outlined in Chapter 7.

   (d) As described above, global applications should not interact with the integrated schema directly. They have to use external schemata. This constraint increases the number of schema transformations to the maximum of five processors and decreases the performance.

3. Schema transformations to external schemata:

   (a) A data schema transformation from a schema which is based on GIM to an external schema which is based on a semantically poor data model is relatively simple because of few concept mappings.

The transformation to a schema which is based on a semantically rich data model is accompanied with an enrichment of semantical information. The information can be either within an SR-schema or information obtained by an interaction with an expert.

(b) The problems to define a view on an external schema, which is based on a data model different to the GIM, is determined by the underlying data model.

On the other hand, it is possible to derive a view with specific semantical relativism and in a specific data model directly from the integrated schema, which is based on GIM. For instance, in [BKNW91] the formulation of object-oriented views on relational schemata is shown. In order to build an external schema, which is data schema equivalent or schema equivalent to a local schema, the corresponding, isolated SR-schema can be used.

In general, the definition of views do not force to change the semantical relativism of the integrated schema. Semantical relativism information must "only" be added.

The generation of an external schema as an integrating view causes some problems. The problems arise because the SR-schemata, which are derived from the local schemata, have to be brought into line. This effort is not more serious than the effort using an object-oriented model as a canonical data model. In this case, next to the generation of external schemata additionally the integration of component schemata has to be carried out in one step.

In this chapter the utilization of a specific, semantically poor integration data model (GIM) was proposed in order to simplify the integration process and the data schema transformation to external schemata. Especially the flexibility of the schema integration and schema transformations can be improved using GIM. For a better understanding of federation with GIM, the introduced example will be used in Chapter 7.

Chapter 6

Characteristics of GIM

The fundamental requirement on describing the GIM is the existence of a formally defined syntax and semantics of it and its languages. Therefore, the author looks for an appropriate, existing data model, which fulfills this requirement and offers the characteristics listed below.

The characteristics of GIM are derived from the purposes of their operation inside the architecture approach proposed in the previous chapter. The list below is only outlined here and hence not complete.

In general, the GIM has to offer concepts to model schemata freed from semantical relativism information. Therefore, there have to exist as little as possible concepts allowing semantical relativism information to be expressed. GIM does not support modeling of behavior. The following list shows the basic concepts of GIM:

- *Simple datatypes*: The author believes that complex data structures are a means to express semantical relativism. For instance, often complex attributes modeled by one person are objects modeled by another person depending on their perceptions. Therefore, only simple datatypes are supported, i.e. a tuple of sets of tuples (similar to named relations in first normal form). The definition of derived (calculated) attributes is not allowed. A reversible transformation of any complex data structure to a tuple of sets of tuples is feasible [Sch95]. In this way structural conflicts can be minimized.

- *Non-overlapping class extensions* means that the extensions of any two classes are either disjoint or identical. Subset relations or partial overlapping between class extensions are not allowed. This frees a local schema from extensional subset hierarchies expressing semantical relativism.

- *Object identifier* (OID) known from object-oriented systems [KC86]: Using OID's simplifies the way of referencing objects. For instance, key-conflicts in relational database can be avoided. One way to generate and manage OID's in FDBS's is described in [SS95].

- *Bidirectional references* (binary relationships) using the OID concept: A reference expresses the relationship between exactly two objects. Relationships between more than two objects have to be transformed to new objects. In [Win90] a way to express n-ary relationships by using binary relations only is shown. Relationships between entity types are characterized by their cardinality. GIM supports different cardinalities (1:1, 1:n, m:n) of reference types between classes (relations). Directions of references are considered as semantical relativism information. Therefore, always bidirectional references are used in GIM. An inverse reference has to exist to any reference in GIM.

- *Flexible integrity constraints*: Integrity constraints are independent of semantical relativism. The handling of integrity constraint conflicts during the schema integration cannot be avoided. An essential type of a global integrity constraint is the existential dependence between objects, because of the fact that attribute values can become virtual objects but they cannot be created and deleted independently from their virtually referencing objects.

- The mentioned concepts have to be designed as *orthogonal* as possible to each other: Orthogonal concepts decrease complex dependencies between different types of conflicts and conflict solving algorithms.

Some concepts of GIM are adopted from other data models. The GIM stands nearer to the relational model than to an object-oriented model. Especially the concepts of simple datatypes and integrity constraints are adopted from the relational model. On the other hand, the concepts of the identification (OID) and referencing (binary and bidirectional) comes from the ODMG-93 object model [Cat94]. The concept of existential dependencies are similar to weak entities in the Entity Relationship model [EN94]. A special feature is the requirement for non-overlapping class extensions. It stands in opposition to the concept of specialization in object-oriented models and is essential during the two-phase-restructuring process exemplified at the end of the next chapter. In contrast to the Binary Relationship model described in [Win90] the distinction between attribute and object is retained.

Chapter 7

Schema Integration in GIM

This chapter shows how the integration process can be done using GIM in more detail than it was described in Chapter 4. The approach proposed differs in various aspects from the integration approaches described in [BLN86, SPD92, TS93, Tre95]. The schema architecture of the integration approach is presented graphically in Figure 7.1. It is a modification of the schema architecture proposed in [SL90].

A system supporting the approach proposed has to be an open system regarding the support of local database systems with any underlying data model [BG92]. For the transformation from a local schema to a component schema (in GIM) general relationships between concepts of the local data model and concepts of GIM are necessary. The author proposes the use of a *meta model* (MM) as the data model, whose instances are the data models involved within the federation. There exist various inter-model relationships between the concepts of its instances (data models) expressing general transformation relationships. The previous chapter introduced the term SR-schema. SR-schemata are instances of specific data models, which are dependent on the used local data models, that is to say the data model of an SR-schema obtained by the transformation from a local, relational schema is different to one of a SR-schema from a local object-oriented schema. Therefore, they are distinguished by different names, e.g. *SR-RM* (*SR r*elational data *m*odel) and *SR-OOM* (*SR o*bject-*o*riented data *m*odel). Both data models also have to be instances of MM. This report does not focus on the meta model.

In FDBS's it is necessary to distinguish between conceptual schemata and external schemata in accordance with the three-level architecture [TK78] and between local and external level in accordance with the five-level architecture [SL90]. In Figure 7.1 these terms are combined:

- Local conceptual schema (`LCS`) and local external schema (`LES`)
- Integrated conceptual schema (`ICS`) and integrated external schema (`IES`)

The relational schema `LCS1` and the object-oriented schema `LCS2` have to be integrated. The external schema `LES2` is defined on `LCS2`. The transformation from the

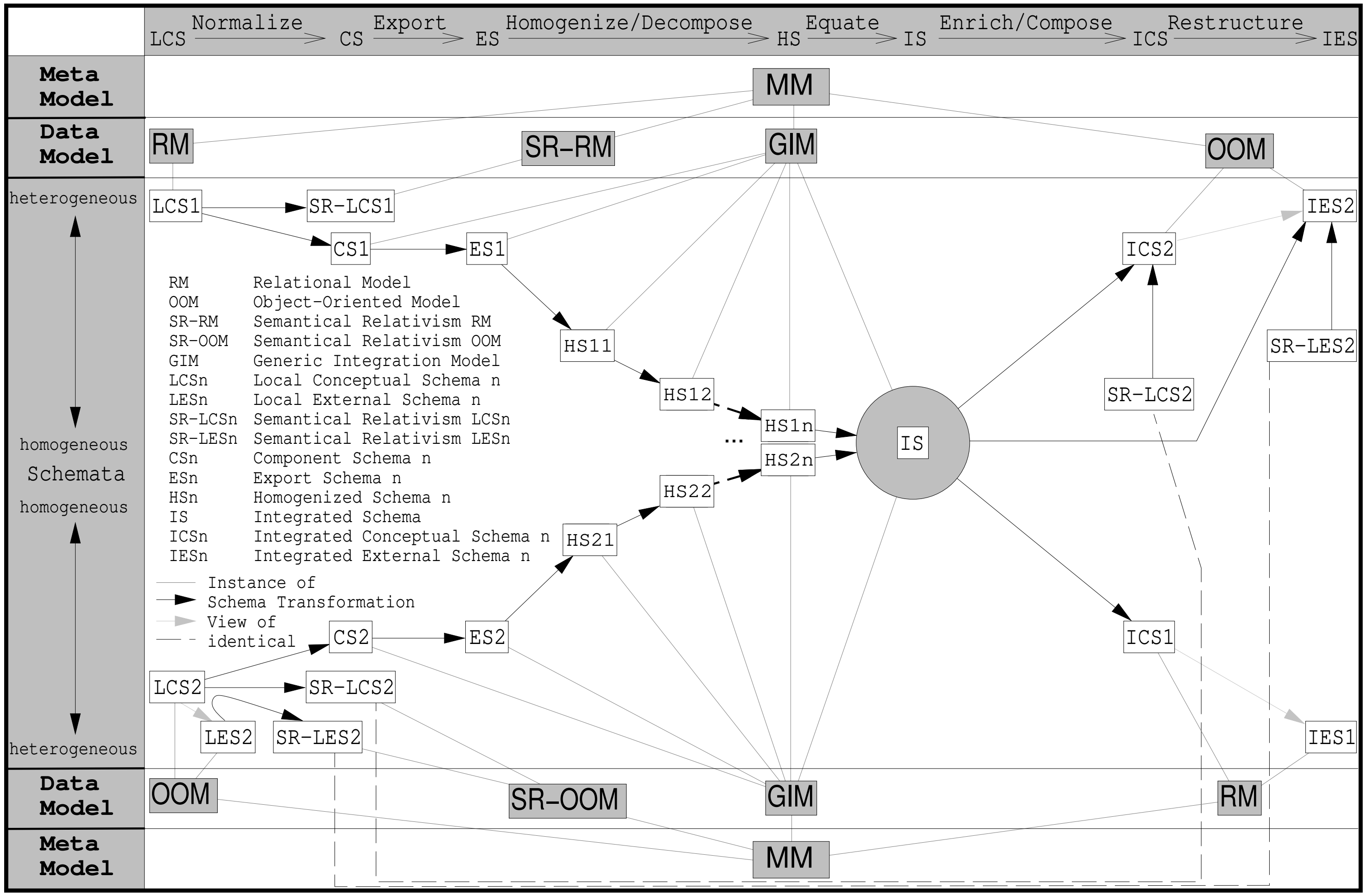


Figure 7.1: Integration process

local schemata produce corresponding component and SR-schemata. The semantical relativism information from the external schema `LES2` is grasped into `SR-LES2`. The export schemata only contain schema elements, which are allowed to be federated. The following homogenizing/decomposition steps remove the mutual heterogeneity of the component schemata in accordance with the two-phase-restructuring approach outlined below. Identical schema elements can then be equated and the integrated schema (`IS`) can be formed. In order to derive the integrated conceptual schema `ICS2`, which should be schema equivalent to `LCS2`, the information of `SR-LCS2` in combination with those of the `IS` is used. The `IES2`, which should be schema equivalent to `LES2`, is directly derived from `IS` using information of `SR-LES2`. An integrated, conceptual, relational view is established in `ICS1`, which reconciles the SR-schemata `SR-LCS1` and `SR-LCS2`.

Now, the depicted process will be exemplified using the example from Chapter 1. Figure 7.2 shows the local schemata transformed to component schemata, which base on GIM. Bidirectional references are introduced (e.g. `CS1.Author.books` and `CS1.Book.authors`) using OID's. In order to guarantee non-overlapping class extensions the extension of class `LCS2.Publication` is decomposed into the class `CS2.Publication-Book`, whose instances are instances of `LCS2.Publication` but not of `LCS2.Book`, and into the class `CS2.Book`, which contains the instances of `LCS2.Book` with their inherited attributes. Because of the demand for simple datatypes, the set-of-string-valued attribute `LCS2.Publication.authors` is transformed to the class `CS2.Author` and to the references `CS2.Publication-Book.authors`, `CS2.Book.authors`, `CS2.Author.publications` and `CS2.Author.books`. The component schemata are also the export schemata in the example.

The semantical relativism information, which reside in the local schemata but not in the component schemata, are rescued in the SR-schemata (Figure 7.3). The SR-schemata are structured in such a way that the derivation of the local schemata from the component schemata and SR-schemata can be carried out.

The flexibility of the approach proposed is achieved by using a two-phase-restructuring process. The two-phase-restructuring approach describes the way of decomposing and composing schema elements during the derivation of the integrated schema and external schemata. Composition of schema elements are only allowed after the last decomposition step has been taken. In other words, the normalization and integration of the component schemata to the integrated schema is only accompanied by intensional and extensional decompositions of schema elements. The derivation of external schemata "only" means to compose the schema elements of the integrated schema in accordance with a specific view and a specific data model.

Now, the example is continued to demonstrate the two-phase-restructuring approach. During the transformation (in Figure 7.1 the normalization) of the local schemata to the component schemata, classes were decomposed (e.g. `LCS2.Publication` and `LCS2.Publication.authors` to the class `CS2.Author`). After this, the component schemata have to be homogenized, i.e. the heterogeneity (conflicts) between the component schemata has to be overcome. Before conflicts can be solved they must be detected.

```
class CS1.Book {                              class CS1.Author {
        string title;                                 string name;
        number isbn;                                  ref Set<Book> books;
        ref Set<Author> authors;                  }
  }

with extensions: CS1.Book   = LCS1.Book
                 CS1.Author = LCS1.Author

class CS2.Publication-Book {          class CS2.Book {
        string title;                         string      title;
        date   year;                          date        year;
        string type;                          string      type;
        ref Set<Author> authors;              ref Set<Author> authors;
  }                                           number      isbn;
                                        }
class CS2.Author {
        string name;
        ref Set<Publication-Book> publications;
        ref Set<Book> books;
  }

with extensions:
CS2.Publication-Book = LCS2.Publication \ LCS2.Book
CS2.Book             = LCS2.Book
CS2.Author           = attribute2class(LCS2.Publication.authors)
```

Figure 7.2: Component schemata CS1 and CS2

In this report only the intensional and extensional conflicts are shown. In the example the extensions and intensions of class `CS1.Book` and the class `CS2.Book` overlap. There are common instances but also instances, which are instances of only one class. Common instances have overlapping intensions. For example, the attribute `CS1.Book.title` is equivalent to the attribute `CS2.Book.title`. On the other hand, to the attribute `CS2.Book.year` there is no equivalent in class `CS1.Book`. Due to the fact that any object of the class `CS1.Book` would have as object of the class `CS2.Book` the value *Book* for the attribute *type*, the intension of class `CS1.Book` is extended by the constant attribute *type="Book"*.

These conflicts are expressed in Figure 7.4 graphically in form of a cartesian diagram. One dimension represents the extensional and the other dimension the intensional aspect of a class. A class itself can be depicted by a rectangle. For simplicity, the referencing attributes of class `Author` are omitted. Mutual conflicts (extensional and intensional

```
SR-CS1.Book {
     public:             title, isbn, authors, no-authors;
     derived attributes: number no-authors = card(authors);
  }

SR-CS1.Author {
     public: name;
  }

SR-CS2.Publication {
     private:   title, year, type, authors;
     public:    get-title();
     methods:   get-title();
     extension: CS2.Publication-Book ∪ CS2.Book;
     attribute: Set<string> authors =
       class2attribute(CS2.Publication-Book.authors,CS2.Book.authors);
  }

SR-CS2.Author {}

SR-CS2.Book {
     public: isbn;
  }
```

Figure 7.3: Semantical relativism conceptual - schemata SR-CS1 and SR-CS2

overlapping) between the component schemata are resolved by decomposition, depicted in Figure 7.5.

The classes 6, 7 and 11 play a particular role. Its instances and attributes are located in both local database systems. The classes are generated by the decomposition process combined with an equation process, which treats instances of both classes as same proxy objects [Pu91, LSPR93, SS95]. Now, the resulting integrated schema contains the classes 1, 2, ..., 11, which have non-overlapping class extensions.

On the base of the IS there are many ways to derive an external schema. In order to derive an external schema, which is equivalent to a local schema, only classes of the integrated schema have to be composed accompanied by a class-to-attribute-operation.

In order to generate a global, integrated, conceptual schema, which contains all exported data from the local database systems, the designer only has to compose classes of the integrated schema, depicted as rectangles to greater ones. In the example one can see that the external schema cannot contain only one class for publications. It is impossible to define a rectangle which contains the classes 1, 2, ..., 10 and not more.

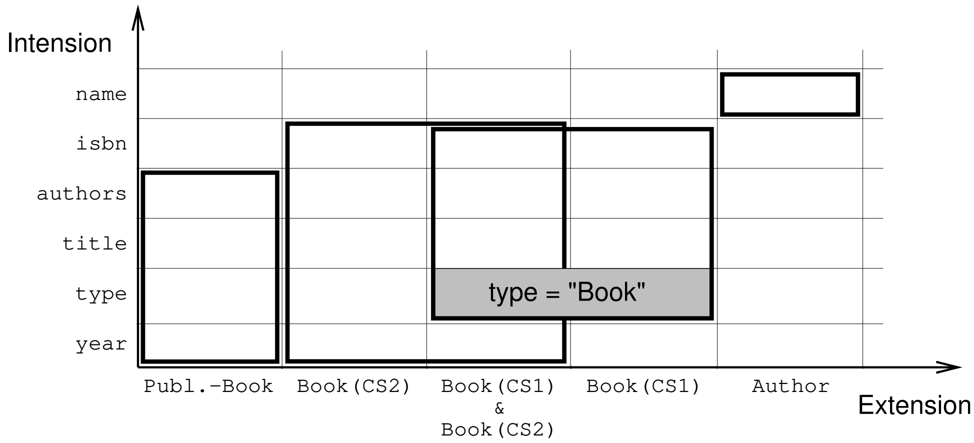
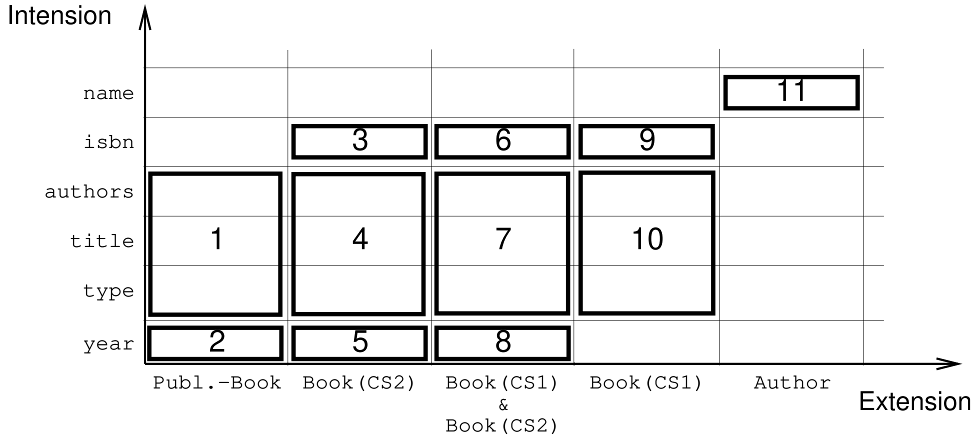


Figure 7.4: Extensional and intensional overlapping

One correct way to build an integrated external schema would be the composition of classes 1,4,7 and 10 to class ICS1.Publication, classes 1, 2, 4, 5, 7 and 8 to class `ICS1.Publication-year`, classes 3, 4, 6, 7, 9 and 10 to class `ICS1.Book`, class 11 to class `ICS1.Author` and class 3,4, ..., 8 to class `ICS1.Book-Publication` (Figure 7.6). The resulting schema is the integrated schema described in Figure 4.2. The classes `ICS1.Publication-year` and `ICS1.Book` are subclasses of class `ICS1.Publication` and the class `ICS1.Book-Publication` is a subclass of `ICS1.Publication-year` and `ICS1.Book`. The subclass-hierarchy is depicted in Figure 7.7.

A subclass relationship between composed classes produces overlapping rectangles in the form of a cross in the diagram. The superclass rectangle always contains the subclass rectangle horizontally and the subclass rectangle always contains the superclass rectangle vertically. Following these constraints it is possible to decide automatically whether a given, integrating specialization hierarchy is valid or not and to generate valid specialization hierarchies from a given integrated schema.




```
Decomposition:
CS1.Book ⇒ 6, 7, 9, 10
CS1.Author ⇒ 11
CS2.Publication-Book ⇒ 1, 2
CS2.Book ⇒ 3, 4, 5, 6, 7, 8
CS2.Author ⇒ 11
```

Figure 7.5: Integrated schema IS

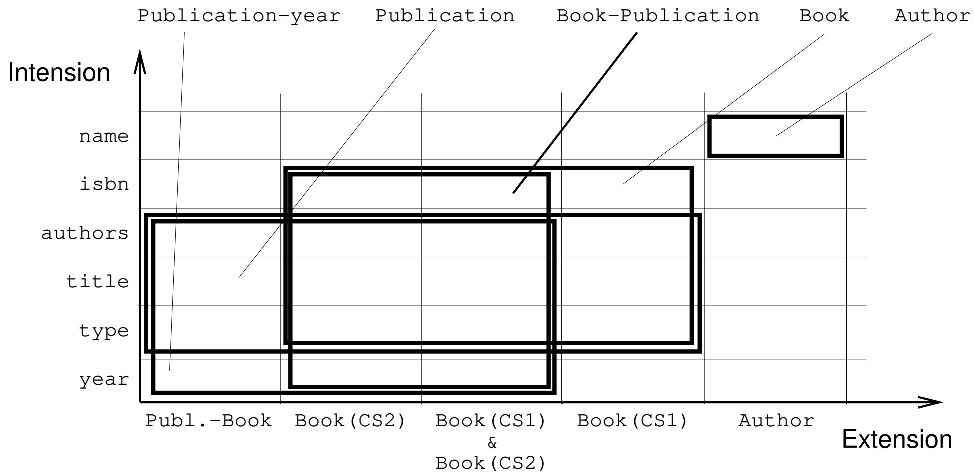


Figure 7.6: Integrated conceptual schema corresponding to Figure 4.2

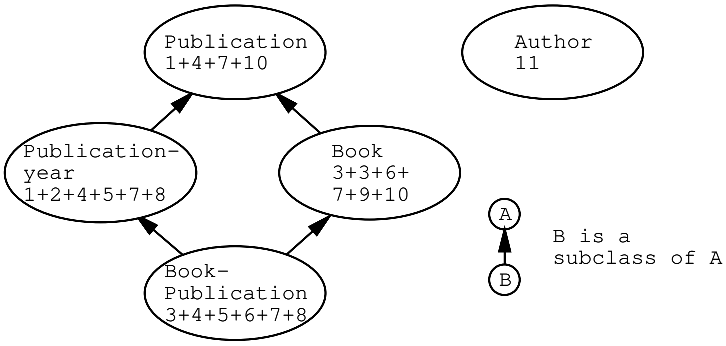


Figure 7.7: Inheritance hierarchy of the integrated conceptual schema of Figure 7.6

Chapter 8

Conclusion and Future Work

In this report the deficiencies (with regard to flexibility and complexity) of conventional federation approaches using an object-oriented data model as canonical data model are outlined. In my opinion the deficiencies are caused by using an object-oriented data model as as canonical data model. Selecting a semantically rich data model like an object-oriented model allows too much semantical relativism to be expressed in the federated schema. This complicates the integration of schemata and leads to an federated schema, which represents already a specific way of looking at the UoD. If a view (external schema) has to be derived from the object-oriented federated schema, the semantical relativism has to be changed instead of being added. Changing semantical relativism means a complex restructuring of the schema.

To overcome this deficiencies a semantically poor data model as canonical data model is proposed. In order to avoid a semantical loss by integrating local, semantically richer schemata, the local schemata are split into SR-schemata and into component schemata. The component schemata are integrated to the integrated schema. The SR-schemata can be reused to derive external schemata. The complexity of the federation is reduced by using a canonical data model with few concepts (GIM) and is further reduced by applying the two-phase-restructuring approach. During the restructuring process the handling of object identifier is an important topic. One way to manage this problem is shown in [SS95]. Due to the fact, that only suggestions are proposed here, further investigation is necessary to solve following questions:

- How can GIM be defined exactly (formal syntax and semantics)?
- How should local, heterogeneous schemata be normalized and integrated to the integrated schema following a methodology?
- How much of the integration and derivation of external schemata have to be done manually?
- How can correctness of the integration be defined and proved for given transformations?

- How should integrating of integrity conditions be dealt with?
- How can multilevel mappings be optimized to increase the performance of database access caused by running global applications?
- How can the meta model (MM) be defined, in order to be a model for an open set of data models?

Bibliography

[BEK+94] R. Böttger, Y. Engel, G. Kachel, S. Kolmschlag, D. Nolte, and E. Radeke. Enhancing the Data Openness of Framworks by Database Federation Services. Technical Report, Cadlab Paderborn, Germany, 1994.

[BFN94] R. Busse, P. Fankhauser, and E. J. Neuhold. Federated Schemata in ODMG. In J. Eder and L. A. Kalinichenko, editors, *Extending Information Systems Technology – Proc. of the 2nd Int. East/West Database Workshop, Klagenfurt, Austria*, pages 356–379. Workshops in Computing, Springer-Verlag, September 1994.

[BG92] T. Barsalou and D. Gangopadhyay. M(DM): An Open Framework for Interoperation of Multimodel Multidatabase Systems. In *Proc. of the 8th IEEE Int. Conf. on Data Engineering (ICDE'92), Tempe, AR, USA*, pages 218–227. IEEE Computer Science Press, February 1992.

[BKNW91] T. Barsalou, A. Keller, Siambela N., and G. Wiederhold. Updating Relational Databases through Object-Based Views. In J. Clifford and R. King, editors, *Proc. of the 1991 ACM SIGMOD Int. Conf. on Management of Data, Denver, Colorado, SIGMOD RECORD 20(2)*, pages 248–257. ACM Press, June 1991.

[BL84] C. Batini and M. Lenzerini. A Methodology for Data Schema Integration in the Entity-Relationship Model. *IEEE Transaction on Software Engineering*, SE-10(6):650–664, Nov 1984.

[BLN86] C. Batini, M. Lenzerini, and S. B. Navathe. A Comparative Analysis of Methodologies for Database Schema Integration. *ACM Computing Surveys*, 18(4):323–364, December 1986.

[Cat94] R. G. Cattell. *The Object Database Standard: ODMG-93.* Morgan Kaufmann Publishers, San Mateo, CA, 1994.

[CGH+93] D. Clements, M. Ganesh, S.-Y. Hwang, E.-P. Lim, K. Mediratta, J. Srivastav, J. Stenoien, and H.-R. Yang. Myriad: Design and Implementation of a Federated Database Prototype. Technical Report 93-76, Departement Computer Science, University of Minnesota, 1993.

[Dat90] C. J. Date. *An Introduction to Database Systems. Volume 1.* Addison-Wesley, Reading, MA, 5 edition, 1990.

[EN94] R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems.* Benjamin Cummings, Redwood City, CA, 2 edition, 1994.

[GGF+95] Gardarin. G., S. Gannouni, B. Finance, P. Fankhauser, W. Klas, D. Pastre, R. Legoff, and A. Ramfos. IRO-DB — A Distributed System Federating Object and Relational Databases. In O. Bukhres and A. K. Elmagarmid, editors, *Object Oriented Multidatabase Systems – A Solution for Advanced Applications.* Prentice Hall, 1995.

[Här94] M. Härtig. *Objektorientierte Integration von autonomen Datenhaltungssystemen.* Ph.D. thesis, Verlag Dr. Kovač, Hamburg, 1994, in german.

[Heu94] A. Heuer. *Zur Rolle generischer Operationen in objektorientierten Datenbanken.* Infix, 1994, in german.

[HK95] U. Hohenstein and C. Körner. Semantische Anreicherung relationaler Datenbanken. In G. Lausen, editor, *Proc. GI-Fachtagung "Datenbanksysteme in Büro, Technik und Wissenschaft" (BTW'95), Dresden*, pages 130–149. Informatik aktuell, Springer-Verlag, May 1995, in german.

[Hoh93] U. Hohenstein. *Formale Semantik eines erweiterten Entity-Relationship-Modells.* Teubner-Texte zur Informatik, Band 4, Teubner-Verlag, Stuttgart, 1993, in german.

[Hul86] R. Hull. Relative Information Capacity of Simple Relational Database Schemata. *SIAM Journal on Computing*, 15(3):856–886, 1986.

[KC86] S. N. Khoshafian and G. P. Copeland. Object Identity. In N. Meyrowitz, editor, *Proc. of the 1st Int. Conf. on Object Oriented Programming Systems, Languages and Applications (OOPSLA'86), Portland, Oregon, SIGPLAN Notices 21(11)*, pages 406–416. ACM Press, November 1986.

[Kim95] W. Kim, editor. *Modern Database Systems.* ACM Press, New York, 1995.

[KLK91] R. Krishmamurthy, W. Litwin, and W. Kent. Language Features for Interoperability of Databases with Schematic Discrepancies. In Y. Kambayashi, M. Rusinkiewics, and A. Sheth, editors, *Proc. of the 1st Int. Workshop on Interoperability in Multidatabase Systems (IMS'91), Kyoto, Japan*, pages 144–151. IEEE Computer Society Press, April 1991.

[LR82] T. Landers and R. Rosenberg. An Overview of Multibase. In *Distributed Data Bases*, pages 153–184. North-Holland, 1982.

[LSPR93] E.-P. Lim, J. Srivastava, S. Prabhakar, and J. Richardson. Entity Identification in Database Integration. In A. Elmagarmid and E. Neuhold, editors, *Proc. of the 9th IEEE Int. Conf. on Data Engineering (ICDE'93), Vienna, Austria*, pages 294–301. IEEE Computer Society Press, April 1993.

[McL93] D. McLeod. Beyond Object Databases. In W. Stucky and A. Oberweis, editors, *Proc. of GI-Conf. Datenbanksysteme in Büro, Technik und Wissenschaft (BTW'93)*, pages 1–17. Informatik aktuell, Springer-Verlag, May 1993.

[Pu91] C. Pu. Key Equivalence in Heterogenous Databases. In Y. Kambayashi, M. Rusinkiewics, and A. Sheth, editors, *Proc. of the 1st Int. Workshop on Interoperability in Multidatabase Systems (IMS'91), Kyoto, Japan*, pages 314–316. IEEE Computer Society Press, April 1991.

[Raf91] Rafi, A. et al. The Pegasus Heterogeneous Multidatabase System. *IEEE Computer*, 24(12):19–27, December 1991.

[SCG91] F. Saltor, Castellanos, and M. Garcia-Solaco. Suitability of Data Models as Canonical Models for Federated Databases. In J. Clifford and R. King, editors, *Proc. of the 1991 ACM SIGMOD Int. Conf. on Management of Data, Denver, Colorado, SIGMOD RECORD 20(2)*, pages 44–48. ACM Press, June 1991.

[Sch95] U. Scholz. Untersuchung der Möglichkeiten einer effizienten Datenabbildung einer TROLL-Spezifikation auf das DBMS ORACLE 7. Diplomarbeit, Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik, September 1995, in german.

[SL90] A. P. Sheth and J. A. Larson. Federated Database Systems for Managing Distributed, Heterogenous, and Autonomous Databases. *ACM Computing Surveys*, 22(3):183–236, September 1990.

[SPD92] S. Spaccapietra, C. Parent, and Y. Dupont. Model Independent Assertions for Integration of Heterogeneous Schemas. *The VLDB Journal*, 1(1):81–126, 1992.

[SS95] I. Schmitt and G. Saake. Managing Object Identity in Federated Database Systems. In *Proc. of the 14th Int. Conf. on Object-Oriented & Entity-Relationship Modelling (OOER'95), Gold Coast, Queensland, Australia.* Springer-Verlag, December 1995. *To appear.*

[Tem83] Templeton et al. An Overview of the MERMAID System - a Frontend to heterogeneous Databases. In *Proceedings IEEE EASCON*, Washington, D.C., USA, 1983.

[TK78] D.C. Tsichritzis and A. Klug. The ANSI/X3/SPARC DBMS Framework Report of the Study Group on Database Management Systems. *Information Systems*, 3(3):173–191, 1978.

[Tre95] M. Tresch. *Evolution in Objekt-Datenbanken.* Dissertation, Teubner Verlag, Stuttgart, Leipzig, 1995, in german.

[Tro93] O. M. F. Troyer. *On Data Schema Transformation.* Ph.D. thesis, Wibro Dissertatiedrukkerij, Helmond, The Netherlands, 1993.

[TS93] M. Tresch and M. Scholl. Schema Transformation Processors for Federated Objectbases. In S. C. Moon and H. Ikeda, editors, *Proc. of the 3rd Int. Conf. on Database Systems for Advanced Applications (DASFAA'93), Daejon, Korea.* World Scientific Press, April 1993.

[Win90] J.J. Wintraecken. *The NIAM Information Analysis Method - Theory and Practice.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1990.