User-Specific Semantic Integration of Heterogeneous Data: The SIRUP Approach

Patrick Ziegler and Klaus R. Dittrich

Database Technology Research Group, Department of Informatics,
University of Zurich, Winterthurerstrasse 190, CH-8057 Zürich, Switzerland
{pziegler|dittrich}@ifi.unizh.ch

Abstract. We give an overview of the SIRUP (Semantic Integration Reflecting User-specific semantic Perspectives) approach to semantic data integration that takes into account heterogeneity of data receivers. Our goal is to provide means that allow data from heterogeneous sources to be integrated in a way that it perfectly fits to a particular user's information needs, emphasizing his individual way to perceive a domain of interest. To achieve this, we propose to use a semantic multidatasource language to declaratively manipulate so-called *IConcepts*; these are basic conceptual building blocks to which attribute data that refers to the same real-world concept is linked by data providers. We provide explicit, queryable semantics by connecting IConcepts to concepts of ontologies. Additionally, pre-integrating data on a conceptual level through IConcepts shields SIRUP end-users from low-level heterogeneity and technical details of underlying data sources.

1 Introduction

In today's ever increasing abundance of online data sources, integration is becoming more and more indispensable in order not to drown in data while starving for information. In general, the goal of data integration is to combine data from different sources by applying a global data model and by detecting and resolving schema and data conflicts so that a homogeneous, unified view can be provided. The reason for data integration is twofold: First, given a set of existing data sources, an integrated view is to be created to facilitate data access and reuse through a single data access point. Second, given a certain information need, data from different complementing sources is to be combined to gain a more comprehensive basis to satisfy the information need.

There is a remarkable history of research projects in the area of data integration. The spectrum ranges from early multidatabase systems (e.g., Multibase [18]) over mediator systems (e.g., Garlic [4]) to ontology-based integration approaches (e.g., OBSERVER [19]). These approaches have in common that autonomy of the data sources to be integrated is considered to be of paramount importance.

Besides this autonomy of data sources, there is the often neglected autonomy and sovereignty of data receivers, i.e., human users and applications [11].

Data receivers are autonomous in the sense that they typically have different information needs and vary in the ways they perceive their particular domain of interest. Sovereignty of data receivers refers to the fact that using integrated data must be *non-intrusive* [25]; i.e., users should not be forced to adapt to any standard concerning structure and semantics of data they desire. Therefore, to take a "one integrated schema fits all" approach is definitely not a satisfactory solution. We address the problem of how user-specific ways to perceive a particular application domain can be taken into account in the process of semantically integrating data from heterogeneous data sources.

In this paper, we give an overview of the SIRUP (Semantic Integration Reflecting User-specific semantic Perspectives) approach to data integration that supports semantic integration by modeling user-specific ways to perceive an application domain. The focus of this paper is on the general foundations of our approach; advanced integration concepts and query processing are not covered. Note that we generally concentrate on querying, not on manipulation of integrated data. For integration, we consider alphanumeric data from a broad range of data sources (i.e., database systems, web services, application interfaces, file systems, and the web).

Our general goal is to provide means that allow data from heterogeneous sources to be integrated in a way that it perfectly fits to a particular user's information needs, emphasizing his individual way to perceive a domain of interest. Additionally, we aim at abstracting the user from low-level heterogeneity and technical details of underlying data sources. In contrast to traditional ex-post definition of views on top of already existing integrated global schemas, we advocate a method called *ex-ante view definition* which allows that only data items are integrated which are semantically related according to the user's individual perception of the particular application domain. In the end, we aim at providing well-structured user-specific schemas with extensive metadata and explicit, queryable semantics for integrated data from selected sources:

- By extensive metadata, explicit information on structural aspects of integrated data is given (attributes of classes/relations, attribute data types, measurement unit, precision, constraints, etc.). Additionally, we aim at providing data lineage information for all integrated data items.
- By explicit, queryable semantics, information on the real-world semantics (see Sect. 3) of the integrated data items is given. By making such semantic metadata retrievable through queries, users do not have to interpret schema and data elements themselves, which is generally erroneous. Misinterpretations can therefore be avoided.

This paper is structured as follows: The following Sect. 2 discusses integration mistakes that can occur when integration approaches are applied that provide only a predefined global schema. Sect. 3 deals with semantics and ontologies. The foundations of our solution are presented in Sect. 4 and Sect. 5 discusses some aspects in more detail. Sect. 6 describes the software architecture of the SIRUP prototype. In Sect. 7, an overview of related work and a comparison between SIRUP and related approaches is given. Sect. 8 concludes the paper.

2 Global Schema Approaches to Data Integration

Data from heterogeneous sources is often integrated by defining one single global schema that represents a unified view over this data. Global schema approaches can be classified as follows:

Traditional Global Schema Approaches These approaches use a data model that originates from the era before object-orientation, such as the functional or relational data model, to provide one single global schema for all users. As in Multibase [18] and Mermaid [30], export schemas from the data sources are directly mapped to the global schema.

Object-Oriented Global Schema Approaches Data sources provide interfaces which can be used to define a global schema using an object-oriented data model. These approaches generally employ integration by creating superclasses to subsume related data from several data sources. Examples for this type of approach are Pegasus [1], TSIMMIS [5], and Garlic [4].

Single Domain Model/Ontology Approaches These approaches use a single domain model or ontology against which all data is integrated, e.g., as in SIMS [2], Carnot [6], and PICSEL [10]. A "semantic" approach to integration is chosen by integrating against one general domain model.

Different users often have diverse views of reality — i.e., they perceive and conceptualize the same real-world part differently, according to their relative points of view, their information needs, and expectations [23, 29, 15]. It is due to this fact that imposing a single global schema for all users can have severe limitations that seriously interfere with the users' individual work. We illustrate these limitations with the MOMIS [3] approach to data integration. In general, MOMIS can be considered as an object-oriented global schema approach in our classification. In MOMIS, a common thesaurus for terminological relationships is built from source schemas and clusters of similar classes from the source schemas are identified. Then, for each cluster a single unifying class is defined and an integrated global schema consisting of these unified classes is built.

For example, assume that there are three classes C_1, C_2, and C_3 concerning educational meetings, each from a different data source, as shown in Fig. 1. Semantically, these three classes are very similar in name. Additionally, the attributes they provide are very similar in name semantics and data type. Due to this high similarity, it is very likely that a single class is created to represent the three classes in a global schema. An example of such an integrated class $C_{integrated}$ is shown in Fig. 1.

A first drawback of such a global schema approach is that users may be given a global schema providing a unified view over data that may be – from the users' perspective – inappropriately collected and selected. In general, it is up to the designer of the global schema to choose what information from a particular local schema is relevant to be available in the global schema. Differences between these choices and the information a particular user expects can lead to situations where the global schema is inappropriate for the information needs of certain users. We refer to this problem as a *data selection mistake*.

Class C_1 from data source 1:
lecture(id_number:int, theme:varchar, auditorium:varchar, time:time, lecturer:varchar)

Class C_2 from data source 2:
colloquium(identifier:int, topic:varchar, location:varchar, date:date, speaker:varchar)

Class C_3 from data source 3:
seminar(id:int, subject:varchar, seminar_room:varchar, time:time, professor:varchar)

Integrated class $C_{integrated}$ in the global schema:
course(id:int, subject:varchar, room:varchar, date:date, lecturer:varchar)

Fig. 1. Example Schema Parts from Different Data Sources[1]

[1] We assume data type `date` to consist of information on year, month, day, and time.

The same problem of inappropriate data selection can occur not only within single source schemas, but also with entire data sources. It is up to the designer of the global schema to select from which local data sources data is integrated. However, users may differ in their preference for data from different origins (due to quality, reliability, etc.) from the preference of the global schema designer. Regardless of this, all users are given the same single global schema. We refer to this problem as a *source selection mistake*.

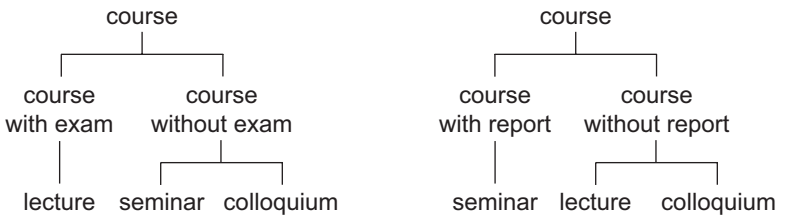
Second, even if the global schema generally provides all the information to satisfy a certain information need, the granularity in which this information is presented may be inappropriate. On the one hand, the view provided by the global schema may be too coarse-grained; i.e., the available information can be too general. In our example, the resulting global schema class $C_{integrated}$ may be useful for users who are satisfied with the global `course` class and do not need to distinguish between different types of educational meetings. However, the global `course` class is of very limited benefit for other users who need a more fine-grained distinction between different types of courses. On the other hand, entity information provided by the global schema can also be too fine-grained.[2] We refer to this problem as an *entity granularity mistake*.

The same problem of inappropriate granularity can occur not only with entities, but also with attributes. For example, the integrated class $C_{integrated}$ only offers information on rooms where the courses take place. The information whether this is an auditorium or a seminar room, which was originally available

[2] In this case, the user might try to define a unifying external view for subsumption, if supported.

in the source schemas C_1 and C_3, respectively, is lost. We refer to this problem as an *attribute granularity mistake*.

Third, classes in the global schema may provide an integrated view on data that is semantically not related according to the individual perception of specific users. We refer to this problem as a *data semantics mistake*. Assume that in our example, class `lecture` and `seminar` are implicitly defined in their respective data sources to consist of a series of class meetings, whereas a `colloquium` is implicitly defined to consist only of a single meeting. In this case, the global class `course` – designed to encompass all available information on educational meetings regardless how many times they take place – is inappropriate for users who care about this distinction. Even worse, important underlying assumptions concerning source schemas and the global schema may be fully implicit due to the lack of explicit metadata and documentation.



(a) Course classification considering exams

(b) Course classification considering reports

Fig. 2. Different Ways to Classify Courses

Fourth, there may be users who need completely different taxonomies of entities than provided by the global schema. For instance, assume a secretary who works for a university needs information on all the courses where students have to pass a final exam. In this case, a classification as shown in Fig. 2(a) can be suitable to provide a schema for queries. On the other hand, another secretary might need information on all the courses where a written report has to be handed in. In this case, a different classification of courses as shown in Fig. 2(b) can be preferable. However, in both cases, there is only the global `course` class $C_{integrated}$ available that cannot satisfy the two different information needs. We refer to this problem as a *taxonomy mistake*.

We have seen six types of integration mistakes that lead to situations where a global schema is inappropriate for particular users. We refer to problems of this type with the general notion of *perspectual integration mistakes*. Note that perspectual integration mistakes can be independently combined to form *combined perspectual integration mistakes*. All six integration mistakes presented are generally caused by differences between the ways a global schema designer and a particular user perceive a certain application domain for which data is integrated; i.e., they are caused by data receiver heterogeneity. *Data receiver*

heterogeneity refers to the fact that users are generally situated in different real-world contexts and widely differ in both, their conceptual interpretation and data preference [11]. Regardless how sophisticated a predefined global schema is, there is no global schema that fits all the needs of all potential users. Therefore, users must be able to specify their individual information needs. Based on this, user-specific integration should take place to give the user access to information in a way that perfectly fits his perception of an application domain of interest.

3 Semantics and Ontologies

Semantics refer to meaning, in contrast to syntax that refers to structure. In the database area, semantics can be regarded as people's interpretation of data and schema items according to their understanding of the world in a certain context. In data integration, the type of semantics considered is generally real-world semantics. According to [21], real-world semantics are concerned with the "mapping of objects in the model or computational world onto the real world [...] [and] the issues that involve human interpretation, or meaning and use of data and information." Differences in interpretations of the same schema or data item between data providers and data users lead to semantic heterogeneity.

One idea to overcome semantic heterogeneity is to exhaustively specify the intended real-world semantics of all data and schema elements. Unfortunately, it is impossible to completely define what a data or schema element denotes or means in the database world [27]. Therefore, database schemas do typically not provide enough explicit semantics to interpret data always consistently and unambiguously [28]. Moreover, there are no absolute semantics that are valid for all potential users; semantics are relative [9]. Nevertheless, a means in form of semantic metadata is necessary to explicitly and semantically characterize in adequate form the information content provided for integration so that it can be reasonably interpreted by humans and computers.

Ontologies are one way to represent explicit, formal semantics. An ontology is "an explicit specification of a conceptualization" [12]. In other words, an ontology is an explicit, formal description of concepts and their relationships that exist in a certain universe of discourse and provides a shared vocabulary to refer to these concepts. Compared with other classification schemes, such as taxonomies, thesauri, or keywords, ontologies allow more complete and more precise domain models [14].

In the area of data integration, ontologies can be applied to ensure semantic interoperability between data sources. By using ontologies, the semantics of data provided by data sources for integration can be made explicit with respect to an ontology a particular user group commits to. Based on this shared understanding, the danger of semantic heterogeneity can be reduced. Note that to avoid problems similar to single global schemas, no single global ontology should be predetermined for all possible user groups. Such an approach would force users to adapt to one single conceptualization of the world. Therefore, a proper ap-

proach to data integration should support different ontologies so that different community-specific semantics can be used in parallel.

4 Foundations of the SIRUP Approach

We propose a novel approach to data integration (see Fig. 3) that mainly aims at avoiding perspectual integration mistakes. It is based on the following principles:

Semantic Perspectives A Semantic Perspective is a user-defined conceptual model of an application domain with explicit queryable semantics for all entities and relationships appearing in it. In particular, a Semantic Perspective expresses user-specific taxonomies and categorizations of real-world entities that belong to a specific application domain according to a particular user's notion. Semantic Perspectives are built on top of data from data sources that are selected by the user and reflect an individual way to perceive a particular real-world part. With such an explicit specification of the desired view, data from user-selected data sources can be integrated reflecting the desired entities and structures defined in the Semantic Perspective.

Bipartite Integration Process In a data integration system, there are generally two main roles: data providers and data users. Data providers know the semantics, local structure, and technical access information[3] of the data they provide and data users know their own information needs for which they want integrated data. Except for small sample cases, none of these two groups has the full knowledge of the other group.

To reflect this dichotomy between the two main roles in data integration systems, the integration process in our approach is separated into two distinct phases: (1) A *data provision phase* where administrators of local data sources explicitly declare the data and its semantics that is offered for integration and (2) a *Semantic Perspective modeling phase* where users who know their application domain for which data is to be integrated define the desired Semantic Perspective.

IConcepts An IConcept (short for *Intermediate Concept*[4]) is a basic conceptual building block that acts as a linking element between data providers and data users interested in data for their information needs. Each IConcept has a queryable link to at least one concept of an ontology to explicitly define the semantics of the real-world concept it represents (e.g., "professor" or "lecture", etc.). Data sources can provide attributes for an ontological concept represented by a particular IConcept. That way, data sources can declare what attribute data they are capable and willing to provide concerning a given IConcept. For each of these attributes, additional structural metadata (data type, measurement unit, precision, constraints, etc.) is provided.

[3] Such as network addresses, protocols, interfaces/APIs, login information, etc.

[4] "Intermediate", because an IConcept is (1) not a fully-fledged user-specific concept (see the next paragraph) but just a building block to construct user-specific concepts and (2) it has an intermediate position between data sources/data providers and data users/query issuers.

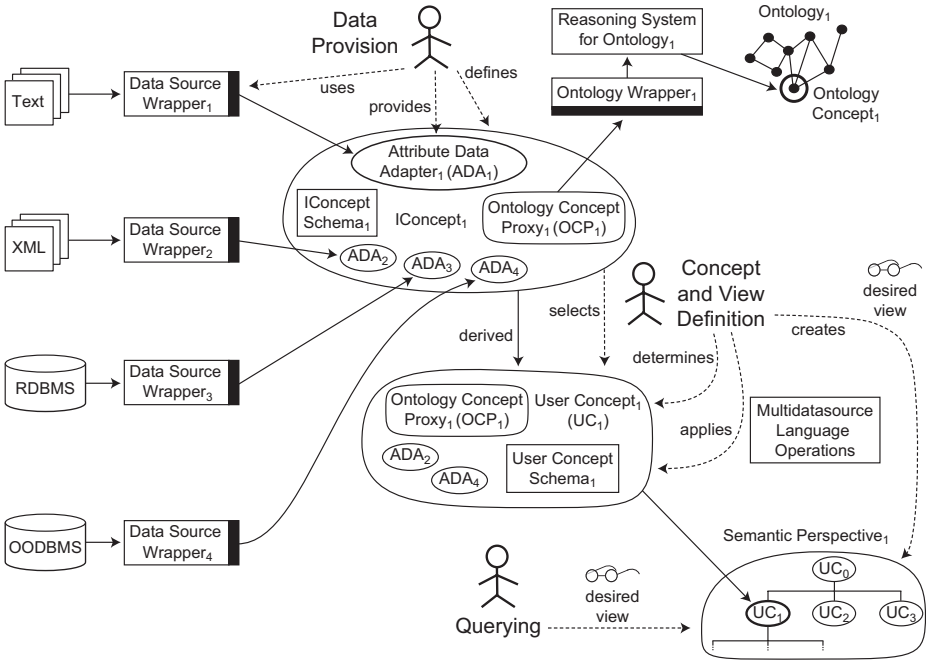


Fig. 3. Overview of the SIRUP Approach to Data Integration

For data providers, IConcepts provide a means to explicitly specify semantics and structure of the data they offer for user-specific integration. Using an ontology index (see Sect. 5.5), data providers identify IConcepts which they are willing to provide data for. For data users, IConcepts are an access point to retrieve data from different data sources referring to the same real-world concept. Additionally, IConcepts hide technical and structural heterogeneity from data users and help to resolve semantic conflicts according to the user's perception of the application domain. Starting from an ontology index, users can browse for IConcepts relevant for their information needs and select, combine and restructure IConcepts to build user-specific concepts and concept hierarchies that define the users' individual Semantic Perspectives.

User Concepts A User Concept is a user-specific concept that is built by selecting and combining user-specific copies of IConcepts. Whenever a user selects an IConcept for his own perspective modeling, a copy of that selected IConcept (together with all its metadata and subcomponents, see Sect. 5.6) is assigned to a namespace private to the user. Using selection, projection, join, cartesian product, and set operators (union, set difference, intersection), users can declaratively define User Concepts based on the selected IConcept copies and already existing other User Concepts. That way, IConcept copies constitute the root nodes on top of which users can incrementally build in a

bottom-up manner user-specific concepts and concept hierarchies that form together the desired Semantic Perspective. In query processing, relevant attribute data can be retrieved from the data sources by the IConcept copies which user-specific concept hierarchies are based on. During concept definition, ontology links inherited from underlying IConcept copies as well as attribute metadata are automatically maintained for User Concepts so that explicitly defined queryable semantics and up-to-date structural metadata are still available. By this metadata, each User Concept possesses a highly structured, explicit schema that describes the attributes that are available.
In case the User Concept's ontology link that is inherited from its underlying IConcept does not exactly fit the user's own intended semantics for that User Concept, the intended semantics of the derived User Concept can be changed by modifying the concept's ontology link and documentation.[5] However, each User Concept must always be assigned to at least one existing ontological concept to ensure that explicit, queryable semantics is anytime available.
In the SIRUP approach, users are abstracted from technical, structural, and semantic integration issues by IConcepts that provide a conceptually homogeneous view on data. However, users who want to define their own Semantic Perspective have to do schema integration on a conceptual level by selecting and combining IConcept copies. In contrast to classical approaches to schema integration, users in our approach can benefit from explicit semantics and from conceptual-level pre-integration of data according to ontological concepts. That way, users are enabled to effectively do the necessary integration activities to build their individual Semantic Perspectives.

Semantic Multidatasource Language In our approach, a declarative language is provided for data provision as well as for specifying User Concepts and Semantic Perspectives. This language supports querying of explicit semantics and metadata assigned to User Concepts and IConcepts. Additionally, data queries against integrated data from Semantic Perspectives are supported. For data providers, our semantic multidatasource language offers the means to perform integration of semantically equivalent IConcept attributes that originate from different data sources. Our semantic multidatasource language supports IConcept definition and linking of attributes from structured, semi-structured, and unstructured data sources. For metadata and data access from these data sources, we employ data source wrappers.

Ex-ante View Definition In traditional view definition in both centralized and distributed database environments, users can specify views only on top of already existing schemas, e.g., using the `create view` command of SQL. We refer to this approach as *ex-post view definition* since the view is created after a (global) schema is defined. When data from different data sources is integrated, perspectual integration mistakes have to be avoided. We believe

[5] Note that this is not intended for arbitrarily changing source data semantics, e.g., from "course" to "cat", but for cases where User Concepts are defined by generalizing or specializing other User Concepts or IConcepts, e.g., specializing from "course" to "database course". Here, it is desirable that the ontology link can be adjusted to refer more precisely to the intended *database* course semantics.

that the best way to prevent these mistakes is to allow the user to specify his own individual way to perceive the application domain of interest, i.e., his own Semantic Perspective. Therefore, integrated schemas or predefined views should not be offered to the user for later refinement but the integrated schemas have to built *before* by the user himself according to his information needs. In our approach, the definition of Semantic Perspectives and declarative integration of IConcept-attributed data are mutually intertwined processes.

Pragmatic Data Integration Approaches that integrate data against one or more global ontologies assume an ideal world in which data for all ontology concepts is available. If data sources do not provide data for all the ontology concepts, issuing queries against ontologies acting as a query schema is not of much use. Our approach is pragmatic in that sense that only concepts for which data that is actually provided by one or more data sources is available for building Semantic Perspectives. That way, data for all concepts appearing in Semantic Perspectives can really be provided in general.[6]

Based on these foundations, perspectual integration mistakes can be avoided by enabling user-specific semantic data integration and restructuring. This is illustrated in the example in Fig. 4: Data on seminars, lectures, colloquia, and conferences from local data sources at a university is linked to IConcepts. That way, this data can be used in Semantic Perspective modeling to provide integrated information in a way that it perfectly fits the secretary's desired view from Fig. 2(a)[7]. At the same time, a completely different Semantic Perspective concerning database technology research meetings can be supported based on selected (i.e., specialized) local colloquium and external conference data.

5 A Closer Look at the SIRUP Approach

In this section, more details on several aspects and components of our integration approach are given.

5.1 Roles in the SIRUP approach

In the SIRUP approach to data integration, there are three roles in which human users appear (see Fig. 3):

Data Provision Each data provider is assumed to have detailed technical knowledge about his particular data source (e.g., about its location in a network, how to access it and its data, etc.) and the structure and semantics

[6] In practice, some data sources may nevertheless be temporarily unavailable due to server failure or maintenance, etc.

[7] For local course data, we assume that the (local) Semantic Perspective modeler can easily determine — based on his knowledge of the local domain — which courses require a final exam.

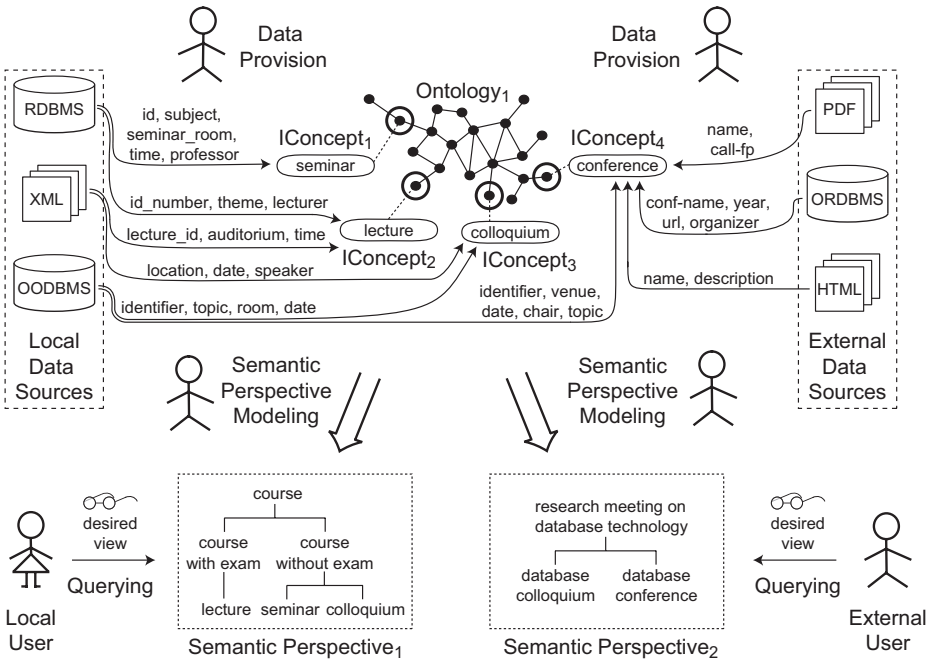


Fig. 4. Example on Applying the SIRUP Approach to Provide Tailored Semantic Perspectives on Course and Conference Data

of all the data it provides for integration. On that basis, data providers create IConcepts[8] and supply metadata for the attributes their data sources are capable and willing to provide for a given IConcept.

Concept and View Definition Each person defining User Concepts for a Semantic Perspective can select from the attributes that are provided by IConcepts and other publicly available User Concepts. For this conceptual modeling process, we assume that the particular person has a clear idea of the relevant concepts to be modeled for the desired Semantic Perspective. Concept and view definition can be done by (sufficiently knowledgeable) end-users as well as by information architects or information engineers who define tailored views for particular user communities.

Querying After a Semantic Perspective is defined, queries against the global schema it represents can be asked. Note that this global schema is application- or user-specific; i.e., there are *several* co-existing user-specific schemas available in SIRUP, in contrast to traditional integration approaches providing a *single* global schema for all users. For querying, we assume the user has a clear idea what data is desired as a query result.

[8] Note that for each ontological concept, only one single IConcept can be created.

5.2 Data sources and source data

For the type of data to be integrated in our approach, we focus on alphanumeric data. Other types of data, such as images, audio and video data, or binary data files (Word, Excel, PDF, etc.) are only considered as atomic files with no additional internal structure. Therefore, this type of data can only be available to users as entire files that appear as "file" attributes of IConcepts (i.e., a data source can provide images of professors as a "professor_image" attribute for an IConcept representing the ontological concept of "professor").

We aim at supporting a broad range of data sources in our approach. First, data sources can be traditional database systems (relational, object-oriented, etc.). Besides this, data from web services and applications that provide data export facilities (e.g., an API to retrieve XML data) are to be supported. Last, but not least, data from text files (i.e., XML, SGML, HTML, plain text, etc.) provided by file systems as well as data from the WWW shall be available for integration.

5.3 External Ontologies

For making semantics of data explicit, ontologies are used in the SIRUP approach (see Fig. 3). In order not to constrain the set of applicable ontologies, we intend to support different ontology languages [7]. That way, ontologies that are specified in various ontology languages can be used in SIRUP for data content explication. We employ ontology wrappers to cope with heterogeneity caused by differences in these ontology languages. Concerning their content, the set of available ontologies is generally open and extensible so that general foundational ontologies as well as specialized domain-specific ontologies can be used for accurate data content explication.

To provide a sophisticated query service on data semantics, we plan to use the reasoning systems provided for the various supported ontology languages. Hence, our semantic multidatasource language will act as a unified query interface for access to different ontological reasoning systems. Thus, users can be provided with explicit, queryable semantics regardless what ontology language is used to represent the intended real-world semantics of data.

5.4 Wrappers

A wrapper is a coupling software component that is specific to an external system and that bridges the gap between this external system and a target system by translating queries, commands and data between internal (local) and external (global) formats. In the SIRUP approach, all available wrappers are centrally registered and can therefore be easily accessed whenever needed. We use two types of wrappers to provide a uniform interface to data sources and ontological reasoning systems (see Fig. 3):

Data Source Wrappers This type of wrapper is responsible for exporting metadata on the attributes a particular data source provides for an IConcept. In query processing, requested attribute data can be retrieved by the wrappers and is converted — if necessary — into XML format, which is globally used for all data in our approach.

Ontology Wrappers By ontology wrappers, queries concerning data semantics are translated between our semantic multidatasource language and the languages used by ontological reasoning systems. Ontology wrappers also convert returned query results from external reasoning systems into a homogeneous format. That way, users can issue queries on data semantics without having to use different ontology languages.

5.5 Ontology Index

In order to provide efficient access to ontological concepts, a centralized ontology index is maintained in our approach (see Fig. 5). All ontologies whose concepts are used to explicitly define data semantics must be registered to that index.[9] That way, users looking for data are provided with a single point of access. Data providers can only use ontological concepts in the data provision phase whose ontologies are registered. During registration, at least one ontology wrapper has to be specified in order to uniformly access the reasoning system of the particular ontology.

When Semantic Perspectives are defined, users implicitly express which IConcepts (and, consequently, which underlying ontological concepts) they regard as semantically equivalent or related. For instance, union operations between IConcepts specify semantic equivalence from the particular user's point of view. Besides this, semantic similarities and relationships between IConcepts are expressed by joins. Whenever a user applies union or join operations on IConcepts, the expressed semantic relation between these IConcepts is automatically reported to the ontology index. That way, user-specific intra- and inter-ontological mappings can be recorded. By offering these mappings to other users and data providers, the process of finding appropriate ontological concepts and IConcepts for data provision and Semantic Perspective building can be facilitated.

5.6 IConcept Components

In order to abstract users from heterogeneity of underlying data sources, IConcepts use services provided by two subcomponents (see Fig. 3):

Attribute Data Adapter (ADA) ADAs are software components that link a certain data source wrapper to one single IConcept. Each ADA encapsulates information about which wrapper is to be accessed and which queries or data access scripts are to be used in order to retrieve attribute data for all attributes a data source provides for one particular IConcept.

[9] Additionally, all available IConcepts must be centrally registered in a global metadata repository. See Fig. 5.

Ontology Concept Proxy (OCP) Each IConcept contains exactly one OCP that encapsulates information about which ontology wrapper is to be accessed in order to retrieve data semantics information.[10]

5.7 Maintenance of Ontology Links

During Semantic Perspective and User Concept building, it has to be ensured that the links to ontology concepts used for expressing explicit data semantics remain valid. In the SIRUP approach, ontology links are automatically maintained while the user declaratively specifies the desired User Concepts. In particular, the following link adjustments are made for operations supported by our multi-datasource language:

- For projection and selection on IConcepts or User Concepts, a new User Concept is created. The ontology link remains unchanged, i.e., the new User Concept is assigned a copy of the ontology link of the concept[11] the operation is performed on.
- For inner join, full outer join, and cartesian product, a new User Concept is created to which a copy of the ontology links from both involved concepts is assigned.
- For left outer join, a new User Concept is created. To the new User Concept, a copy of the ontology link from the left concept is assigned. ("Left" and "right" in this context means the following: For reference to position, we consider the position of the concepts in the following notation for join operations, as also supported by SQL: `employee join department on emp_depnr = dep_nr`. In this example, `employee` is considered as the left, `department` as the right concept.)
- For right outer join, a new User Concept is created to which a copy of the ontology link from the right concept is assigned.
- For union, outer union and intersection, a new User Concept is created to which a copy of the ontology links from both involved concepts is assigned.
- For set difference, a new User Concept is created to which a copy of the ontology link from the left concept is assigned.
- For changes in attribute metadata entries (attribute renaming, type conversion, etc.),[12] only the affected attribute metadata of the User Concept is changed. The ontology link of the User Concept remains unchanged.

5.8 Integrated Data and Querying

After Semantic Perspectives tailored for certain information needs are created, they are available for querying. In general, integrated data for querying and

[10] In case of union and join operations on User Concepts (which are derived from IConcepts), more than one OCP may be available in the User Concept since ontological concepts from more than one ontologies may be referenced. See Sect. 5.7.

[11] In this section, "concept" refers to both, IConcept and User Concept.

[12] Note that these changes can only be applied to User Concepts.

further use is to be provided in a data format that is widely accepted and usable. Therefore, we plan to use XML to offer all integrated data in a structurally self-describing way.

In SIRUP, two types of queries can be asked:

Data Queries This type of query represents requests for data that is integrated and structured in a user-specific way. Data queries can be formulated by using User Concept and IConcept names as well as names of their attributes in declarative SQL-/OQL-like query statements.

Semantics Queries By semantics queries, requests for explicit data semantics for IConcepts, User Concepts, and their attributes can be expressed. Semantics queries either refer to requests for the real-world semantics of a particular IConcept or User Concept. Alternatively, semantics queries can be requests for metadata on structural aspects of IConcepts, User Concepts, and their attributes (e.g., number and names of attributes, attribute data types, data lineage information, etc.).

6 Outlook on the SIRUP Prototype Architecture

As a final proof of concept, we plan to implement a fully functional SIRUP prototype. Each SIRUP system generally consists of one central Global SIRUP Metadata Server and one or more Local SIRUP Clients (see Fig. 5):

- The Global SIRUP Metadata Server provides information on all registered IConcepts for which data is provided as well as information on all registered ontological concepts that are available to explicitly define data semantics. Additionally, it stores information about all available data source and ontology wrappers and manages central repositories for Attribute Data Adapters and Ontology Concept Proxies.
- The Local SIRUP Client is in charge of accepting and processing declarative User Concept specifications and modifications; alternatively, user input can be a query for data or semantics. For all locally defined User Concepts as well as for all IConcept copies that are in local use, the Local SIRUP Client stores all necessary metadata.

There can be more than one Global SIRUP Metadata Server instance at the same time, e.g., each one run by a different enterprise or (commercial) data provider. Each Local SIRUP Client can use IConcepts from one or more Global SIRUP Metadata Servers at the same time.

Both, the Global SIRUP Metadata Server and each Local SIRUP Client have five common subcomponents (see Fig. 5):

SIRUP Multidatasource Language Parser The SIRUP Multidatasource Language Parser accepts declarative user input that is formulated with our semantic multidatasource language, grammatically analyzes it and produces a parse tree for subsequent processing.

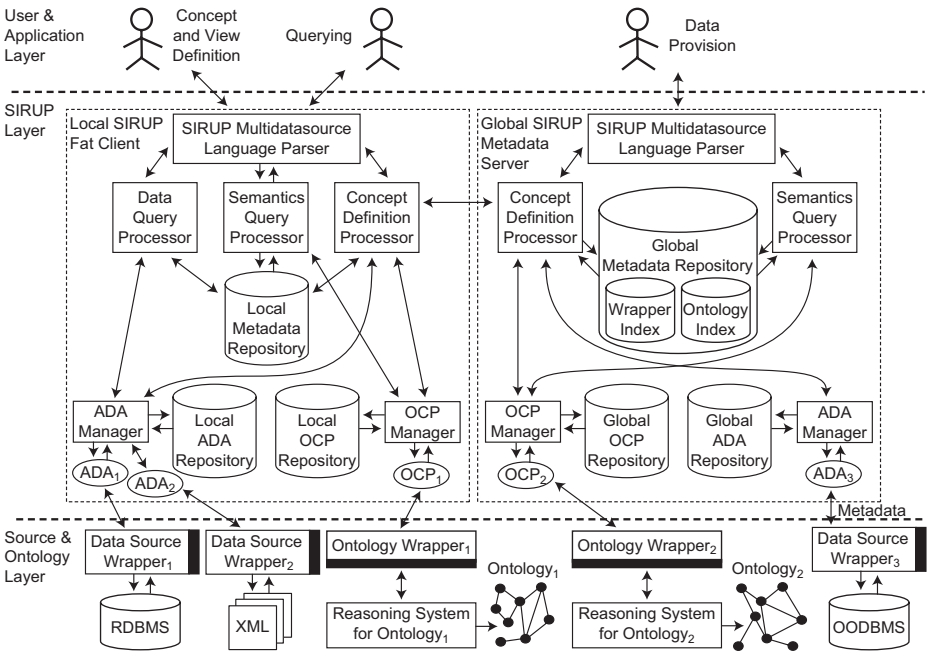


Fig. 5. Fat Client Software Architecture of the SIRUP Prototype

Concept Definition Processor In general, this component is capable of processing all IConcept and User Concept definition requests. The Concept Definition Processors of Local SIRUP Clients and Global SIRUP Metadata Servers communicate in order to exchange metadata.

Semantics Query Processor The Semantics Query Processor processes all requests for explicit data semantics concerning IConcepts, User Concepts, and their attributes. For queries on the real-world semantics of IConcepts and User Concepts, the Semantics Query Processor is in charge of dispatching them via an ontology wrapper to the appropriate external ontology system for evaluation.

ADA Manager The ADA Manager component is responsible for storing all necessary ADAs and retrieving individual ADAs whenever they are needed — e.g., when IConcepts are copied from a Global SIRUP Metadata Server to a Local SIRUP Client.

OCP Manager Each OCP Manager has to store all necessary OCPs in its OCP Repository and to retrieve individual OCPs whenever needed.

Besides the five subcomponents that are also part of a Global SIRUP Metadata Server, each Local SIRUP Client has an additional subcomponent:

Data Query Processor The Data Query Processor accepts parse trees concerning data queries from the SIRUP Multidatasource Language Parser as

input and processes them. Note that Global SIRUP Metadata Servers do not have to be able to process data queries since they do not provide integrated data but conceptual building blocks (i.e., IConcepts) for user-specific data integration that is performed on Local SIRUP Clients.

7 Related Work

With respect to our goals (see Sect. 1), related work can mainly be found in the area of multidatabase languages and declarative integration languages, such as MSQL+ [20], SQL/M [16], SchemaSQL [17], and FRAQL [24]. Besides this, approaches that provide conceptual-level abstraction from data sources (e.g., InfoQuilt [26] and KIND [13]), object-oriented virtual integration approaches (e.g., TSIMMIS [5] and Garlic [4]), ontology-based integration approaches (e.g., SIMS [2] and OBSERVER [19]), Semantic Web approaches (e.g., On2broker [8]), and taxonomic database systems (e.g., Prometheus [22]) can be regarded as areas of related work. As we concluded in [31], no related approach is able to attain all of our main goals. Therefore, an integration approach capable of (1) preserving user sovereignty by enabling the user to express his desired view for data integration, (2) shielding the user from technical-level heterogeneities, and (3) providing explicit, queryable data semantics is desirable.

Our work in the SIRUP approach differs from related work in the following aspects:

- First, compared with multidatabase languages like SchemaSQL [17] or FRAQL [24], users are in our approach abstracted from underlying data sources. Since multidatabase languages do not provide transparency, users have to directly locate and access the component database systems from which data is requested. Therefore, applying multidatabase languages is quite demanding since all tasks of data integration and reconciliation are solely put on the user. Additionally, multidatabase languages provide no explicit queryable data semantics to ensure semantically correct interpretations of schemas and data. Another difference to our approach is that multidatabase languages focus on data integration in homogeneous environments — usually, data from relational databases is considered.
- Second, there are integration approaches that abstract from the local data sources by providing a conceptual layer on top of which data integration can be performed, like KIND [13]. These approaches, in contrast to our approach, principally take a single global schema approach. First, data of each source is separately and conceptually described. Then, these descriptions are used as a basis for global schema creation and query processing.
  Other approaches in this category, like InfoQuilt [26], require the user to model his domain of interest. Then, it is up to the user to identify relevant data sources and to create wrappers. Our approach, on the opposite, does not require users to deal with data sources directly but shields them from underlying technical details and heterogeneity. Additionally, data in our approach is pre-integrated by linking attribute data to IConcepts.

– Third, compared with object-oriented virtual integration approaches like TSIMMIS [5] or Garlic [4], our approach provides explicit, queryable semantics on all available data. Moreover, users in our approach are more abstracted from underlying data sources and do not have to cope with low-level heterogeneities.
– Fourth, our approach can be compared with ontology-based integration approaches. In contrast to single-ontology approaches like SIMS [2], no ontological commitment is needed in our approach in that respect that a user has to accept once and for all one global ontology, i.e., a fixed way to perceive a particular domain. In contrast to multi-ontology approaches like OBSERVER [19], no mapping and similarity detection between involved ontologies is needed in our approach. Ontologies in our approach are just used as a means to explicitly define data semantics for data provision and selection; i.e., we use ontologies for data content explication. Whenever a user selects an IConcept to be a useful source of data for a desired Semantic Perspective, the user is free to change the intended Semantics of his derived User Concept by modifying the concept's ontology link and documentation.[13] Thus, Semantic Perspectives are user-specific conceptual models with ontology-based semantics as intended by a user. Apart from that, our pragmatic approach to data integration is generally capable of providing data for *all* concepts used in Semantic Perspectives, not only for certain ones as might be the case in ontology-based systems.
– Fifth, compared with approaches from the Semantic Web like On2Broker [8], our approach is not limited to data from the WWW. In our approach, data is not just annotated to represent explicit semantics, but a rich semantic multidatasource language is provided to express Semantic Perspectives that form user-specific schemas for querying well-structured data.
– Last, but not least, taxonomic database systems like Prometheus [22] do not consider integration but provision of multiple, overlapping taxonomies for single centralized database systems.

8 Conclusions and Future Work

In this paper, we presented SIRUP, a novel approach to semantic data integration that supports integration by modeling of user-specific ways to perceive an application domain. In essence, our approach provides concepts and a multidatasource language to semantically integrate data that is related according to an individual user's notion. This data is provided with explicit, queryable semantics and is structured according to the user's own conceptual model of his domain of interest. That way, schemas tailored for specific information needs and perceptions are available for querying. In our approach, structured, semi-structured, and unstructured data sources are incorporated.

[13] However, each User Concept must always be assigned to at least one existing ontological concept to ensure that explicit, queryable semantics is anytime available. See also Footnote 5 on this.

Our work is not intended to be a complete replacement for existing integration approaches but as a complementing approach suitable for situations with considerable data receiver heterogeneity — e.g., for business and financial analysts with changing information needs, flexible cooperation between enterprises, virtual organizations, rapid enterprise portal development, and information support for business processes and workflows. The main intention of our approach is to provide semantically integrated data for users with heterogeneous conceptual models in mind who differ in both, their information needs and their preference for integrated data. That way, we aim at combining user-specific semantic data integration with truly individual views so that integrated data tailored to individual perceptions of a particular domain can be supplied.

The distinctive features of the SIRUP approach to semantic data integration are:

- conceptual-level supply of pre-integrated data with explicit, queryable semantics and extensive metadata;
- use of an ontology-enhanced multidatasource language to support declarative modeling of data to be integrated and virtual views, both tailored to user-specific information needs.

That way, data from heterogeneous data sources can be semantically integrated and structured according to specific information needs without having to cope with low-level heterogeneity and technical details of underlying data sources. To the best of our knowledge, this combination of ontology-based pre-integration on a conceptual level with an ontology-enhanced multidatasource language is unique.

Future work includes refinement of the concepts presented in this paper as well as incorporation of additional considerations. Additionally, it is planned to implement a fully functional SIRUP prototype as a proof of concept.

References

1. R. Ahmed, P. D. Smedt, W. Du, W. Kent, M. A. Ketabchi, W. Litwin, A. Rafii, and M.-C. Shan. The Pegasus Heterogeneous Multidatabase System. *IEEE Computer*, 24(12):19–27, 1991.
2. Y. Arens, C. Y. Chee, C.-N. Hsu, and C. A. Knoblock. Retrieving and Integrating Data from Multiple Information Sources. *International Journal of Cooperative Information Systems (IJCIS)*, 2(2):127–158, 1993.
3. S. Bergamaschi, S. Castano, S. De Capitani di Vimercati, S. Montanari, and M. Vincini. An Intelligent Approach to Information Integration. In N. Guarino, editor, *1st International Conference on Formal Ontologies in Information Systems (FOIS 1998)*, pages 253–267, Trento, Italy, June 6-8, 1998. IOS Press.
4. M. Carey, L. Haas, P. Schwarz, M. Arya, W. Cody, R. Fagin, M. Flickner, A. Luniewski, W. Niblack, D. Petkovic, J. Thomas, J. Williams, and E. Wimmers. Towards Heterogeneous Multimedia Information Systems: The Garlic Approach. In *5th International Workshop on Research Issues in Data Engineering-Distributed Object Management (RIDE-DOM 1995)*, pages 124–131, Taipei, Taiwan, March 6-7, 1995.

5. S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom. The TSIMMIS Project: Integration of Heterogeneous Information Sources. In *16th Meeting of the Information Processing Society of Japan (IPSJ)*, pages 7–18, Tokyo, Japan, October 1994, 1994.
6. C. Collet, M. N. Huhns, and W.-M. Shen. Resource Integration Using a Large Knowledge Base in Carnot. *IEEE Computer*, 24(12):55–62, 1991.
7. O. Corcho and A. Gómez-Pérez. A Roadmap to Ontology Specification Languages. In R. Dieng and O. Corby, editors, *12th International Conference on Knowledge Acquisition, Modeling and Management (EKAW 2000)*, volume 1937 of *Lecture Notes in Computer Science*, pages 80–96, Juan-les-Pins, France, October 2-6, 2000. Springer.
8. D. Fensel, J. Angele, S. Decker, M. Erdmann, H.-P. Schnurr, S. Staab, R. Studer, and A. Witt. On2broker: Semantic-based Access to Information Sources at the WWW. In P. D. Bra and J. J. Leggett, editors, *World Conference on the WWW and Internet (WebNet 99)*, pages 366–371, Honolulu, Hawaii, USA, October 25-30, 1999. Association for the Advancement of Computing in Eduction (AACE).
9. M. García-Solaco, F. Saltor, and M. Castellanos. Semantic Heterogeneity in Multidatabase Systems. In O. A. Bukhres and A. K. Elmagarmid, editors, *Object-Oriented Multidatabase Systems. A Solution for Advanced Applications*, pages 129–202. Prentice-Hall, 1996.
10. F. Goasdoué and C. Reynaud. Modeling Information Sources for Information Integration. In D. Fensel and R. Studer, editors, *Knowledge Acquisition, Modeling and Management, 11th European Workshop (EKAW 1999)*, volume 1621 of *Lecture Notes in Computer Science*, pages 121–138, Dagstuhl Castle, Germany, May 26-29, 1999. Springer.
11. C. H. Goh, S. E. Madnick, and M. Siegel. Context Interchange: Overcoming the Challenges of Large-Scale Interoperable Database Systems in a Dynamic Environment. In *Third International Conference on Information and Knowledge Management (CIKM 1994)*, pages 337–346, Gaithersburg, USA, November 29 - December 2, 1994. ACM.
12. T. R. Gruber. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. In N. Guarino and R. Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Kluwer, 1993.
13. A. Gupta, B. Ludäscher, and M. E. Martone. Knowledge-Based Integration of Neuroscience Data Sources. In O. Günther and H.-J. Lenz, editors, *12th International Conference on Scientific and Statistical Database Management (SSDBM 2000)*, pages 39–52, Berlin, Germany, July 26-28, 2000. IEEE Computer Society.
14. M. N. Huhns and M. P. Singh. Agents on the Web: Ontologies for Agents. *IEEE Internet Computing*, 1(6):81–83, 1997.
15. W. Kent. *Data and Reality. Basic Assumptions in Data Processing Reconsidered*. North-Holland, 1978.
16. W. Kim, I. Choi, S. K. Gala, and M. Scheevel. On Resolving Schematic Heterogeneity in Multidatabase Systems. *Distributed and Parallel Databases*, 1(3):251–279, 1993.
17. L. V. S. Lakshmanan, F. Sadri, and I. N. Subramanian. SchemaSQL - A Language for Interoperability in Relational Multi-Database Systems. In T. M. Vijayaraman, A. P. Buchmann, C. Mohan, and N. L. Sarda, editors, *22nd International Conference on Very Large Data Bases (VLDB 1996)*, pages 239–250, Bombay, India, September 3-6, 1996. Morgan Kaufmann.

18. T. Landers and R. L. Rosenberg. An Overview of MULTIBASE. In H.-J. Schneider, editor, *Second International Symposium on Distributed Data Bases (DDB 1982)*, pages 153–184, Berlin, Germany, September 1-3, 1982. North-Holland.
19. E. Mena, V. Kashyap, A. P. Sheth, and A. Illarramendi. OBSERVER: An Approach for Query Processing in Global Information Systems Based on Interoperation Across Pre-existing Ontologies. In *First IFCIS International Conference on Cooperative Information Systems (CoopIS 1996)*, pages 14–25, Brussels, Belgium, June 19-21, 1996. IEEE Computer Society.
20. P. Missier and M. Rusinkiewicz. Extending a Multidatabase Manipulation Language to Resolve Schema and Data Conflicts. In R. Meersman and L. Mark, editors, *Sixth IFIP TC-2 Working Conference on Data Semantics (DS-6)*, volume 74 of *IFIP Conference Proceedings*, pages 93–115, Stone Mountain, Atlanta, Georgia, USA, May 30 - June 2, 1995. Chapman & Hall.
21. A. M. Ouksel and A. P. Sheth. Semantic Interoperability in Global Information Systems: A Brief Introduction to the Research Area and the Special Section. *SIGMOD Record*, 28(1):5–12, 1999.
22. C. Raguenaud, J. B. Kennedy, and P. J. Barclay. The Prometheus Taxonomic Database. In N. G. Bourbakis, editor, *1st IEEE International Symposium on Bioinformatics and Biomedical Engineering (BIBE 2000)*, pages 63–70, Arlilngton, Virginia, USA, November 8-10, 2000. IEEE Computer Society.
23. F. Saltor and M. García-Solaco. Diversity with Cooperation in Database Schemata: Semantic Relativism. In J. I. DeGross, R. P. Bostrom, and D. Robey, editors, *Fourteenth International Conference on Information Systems (ICIS 1993)*, pages 247–254, Orlando, Florida, USA, December 5-8, 1993. ACM.
24. K.-U. Sattler, S. Conrad, and G. Saake. Adding Conflict Resolution Features to a Query Language for Database Federations. In M. Roantree, W. Hasselbring, and S. Conrad, editors, *Third Workshop on Engineering Federated Information Systems (EFIS 2000)*, pages 41–52, Dublin, Ireland, June 19-20, 2000. IOS Press / infix.
25. P. Scheuermann, A. K. Elmagarmid, H. Garcia-Molina, F. Manola, D. McLeod, A. Rosenthal, and M. Templeton. Report on the Workshop on Heterogenous Database Systems held at Northwestern University, Evanston, Illinois, December 11-13, 1989. *SIGMOD Record*, 19(4):23–31, 1990.
26. A. Sheth, S. Thacker, and S. Patel. Complex Relationships and Knowledge Discovery Support in the InfoQuilt System. *The VLDB Journal*, 12(1):2–27, 2002.
27. A. P. Sheth, S. K. Gala, and S. B. Navathe. On Automatic Reasoning for Schema Integration. *International Journal of Intelligent and Cooperative Information Systems*, 2(1):23–50, 1993.
28. A. P. Sheth and J. A. Larson. Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. *ACM Computing Surveys*, 22(3):183–236, 1990.
29. J. F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, 1984.
30. M. Templeton, D. Brill, A. Hwang, I. Kameny, and E. Lund. An Overview of the Mermaid System - A Frontend to Heterogeneous Databases. In J. M. Walker, editor, *Sixteenth Annual Electronics and Aerospace Conference and Exposition - Technology Shaping the Future (IEEE EASCON 1983)*, pages 387–402, Washington, D.C., USA, September 19-21, 1983. IEEE Aerospace and Electronic Systems Society.
31. P. Ziegler. User-Specific Semantic Integration of Heterogeneous Data: What Remains to be Done? Technical Report ifi-2004.01, Department of Informatics, University of Zurich, 2004. http://www.ifi.unizh.ch/techreports/TR_2004.html.