Information Brokering:
Design, Search and Transformation*

Manfred A. Jeusfeld (*), **Mike Papazoglou** (+)
(*) Informatik V (Information Systems), RWTH Aachen
52056 Aachen, Germany
(+) Tilburg University, INFOLAB, PO Box 90153
5000 LE Tilburg, The Netherlands

Abstract. In large human-computer networks, information brokers provide links among syntactically and semantically heterogeneous information sources with information users who are equally diverse in their interests and capabilities. After giving a brief overview of the field, we compare three approaches to information brokering which share the idea of using domain-oriented meta models to focus the brokering process. One is intended for search in large open database networks without any predefined organizational infrastructure, the second for information exchange in distributed and frequently changing organizations. The third approach demonstrates that the "traditional" multi-database task of notational transformation required to cover the syntactic differences among information sources can be addressed in a rather similar manner.

*This work has been supported in part by the Commission of the European Communities under EC-Australia Collaborative Activity on Information Systems Interoperability ECAUS003 (ISI), and Esprit LTR project Foundations of Data Warehouse Quality No. 22469 (DWQ).

1 Introduction

The globalization of today's businesses, the decentralization of corporate operations and services, and the accelerated range in market requirements have forced corporations to take advantage of new technologies and features by "right-sizing" their once relatively homogeneous, static, and weakly connected information systems into multi-vendor, dynamic, highly interconnected networks. The way of building systems has changed from custom-oriented programming to system integration from pre-existing software components. A component can be combined with other components to form a complete application, whereby each software component performs some limited range of useful functions. Components should interoperate across address spaces, networks, languages, operating systems and tools.

From a systems engineering viewpoint, two brokering approaches are followed to support decentralized organizations by information technology. The *request brokering approach* regards the organization as a set of agents that offer operations. The task of the broker is to gather knowledge about offered operations and then direct an operation request to the suitable agent. In contrast, the *information brokering approach* regards the organization as a set of agents that maintain information. Consequently, such a broker gathers knowledge about this information to direct queries and updates on the information to the right agent.

In contrast to traditional multi-database systems, an information broker therefore does not just translate between different data model notations but creates a specific *domain of common interest* across the domains of existing (and possibly future) information sources. Thus, multiple information brokers defining different meta databases can exist in the same network of databases. A functional definition that stresses the information aspect is given in section 2.

Afterwards, we present three types of information brokers that stress a certain function. The first type (section 3) is an information broker for *search* in heterogeneous databases. It employs semantic knowledge about information sources to construct queries. The databases exist prior to the information broker (bottom-up integration). The second information broker type (section 4) is used in the *design and operation* phase of a distributed information system. The databases are integrated in the design phase by modeling the information flow between them (top-down integration). The knowledge about the information flow is maintained by the broker and used in the operation phase for passing queries and requests to the right agent. It turns out that for both types of information brokers an abstract representation of information, hereafter called a *meta model*, is useful to encode and facilitate the respective purposes of the information brokering. Finally, the third type of information broker (section 5) shows that such meta models are not only useful for discovering and designing information but also for supporting syntactic *transformation* of information from one data model into the other.

2 Information Brokers as Meta Database Systems

Information brokers are mediating tools between some application or ad hoc user and a vast resource of information sources. Mediation can be understood in many different ways. Therefore, a definition for information brokers is needed that abstracts from specific uses. The best way to achieve this, is to initially regard an information broker as an advanced database system that stores data about some heterogeneous information sources. Such data is generally called *meta data*. Examples for meta data are data structure definitions (schema concepts), data about the location and access rights for certain databases, tables on statistical distribution of values within some databases, and others.

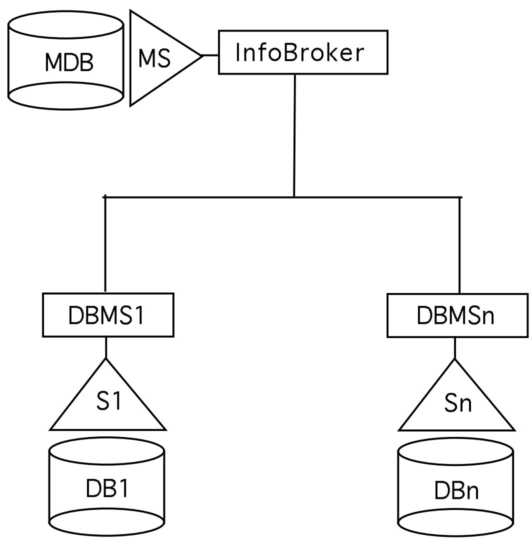

Figure 1: Standard diagram for information brokers

Figure 1 presents the building blocks of an information broker. The information sources are databases DB1 to DBn with schemata S1 to Sn. We collectively refer to them as the *brokered databases*. Both the database instance and the database schema are subject to be described within the information broker's *meta database* MDB. The core concept of the information broker is the schema MS of its database MDB. It defines which information shall be maintained, and thus ultimately circumscribes its possible uses. Following this schema-centered view, the design of an information broker should start with the design of its schema MS just like the design of a database starts with the design of its schema which regards data items to be valid independent from their use in specific applications.

As a demonstration example, we consider a 'schema' broker that allows to query a collection of brokered relational databases. For simplicity, we assume that there are no naming conflicts between the relations schemas S1 to Sn, i.e., each relation has a unique name. What information would be needed in the information broker to support cross-database query processing? The primary requirement is that schema definitions in S1 to Sn are explicitely associated to the database systems that maintain them. Thus, the schema MS would contain a concept 'relation' with the ability to assign a 'location' attribute to it. Schema element definitions represented in the terms of such a meta schema MS are meta data: they describe at this stage only syntactic properties of data. Evolution of some schema Si is translated into a corresponding update to the meta database of the schema broker. Moreover, when a relation is moved from one DBMS to another, this translates into an update of some 'location' attribute in the meta database. It should be noted that the above meta database can also be used as a data dictionary for the design of a distributed homogeneous database system.

The schema of the information broker's database determines its possible uses. We classify information brokers (IB) by their ability to manipulate their database. A *query IB* is limited to perform read operations on the meta database. An *update IB* is allowed to write its meta database as well. The schema broker demonstrated above is a query IB when used for query processing. It would be an update IB when used as a data dictionary. Although there exist IBs which possess the ability to read only, it is unreasonable to assume that they can be used for such complicated functions as evolution and cooperation between databases. Therefore, in this paper we will concentrate on update IBs.

Another characteristic of an IB is the richness of the language it employs to describe its schema MS: its so-called meta model. If the meta database contains all abstraction level found in the ISO Information Resource Dictionary System (IRDS) standard [14], then it can manipulate the schemata S1 to Sn, as well as the data models used to represent those schemata. Such information brokers are thus suited for heterogeneous information systems. We can apply the query and update criterion to this case as well: an update IB with full IRDS [14] abstraction levels has the ability to introduce new data models into its meta database. This type of system can easily support upgrades by allowing the inclusion of new and possibly heterogeneous data models on top of the already existing ones.

The increasingly diverse nature of the support required by the brokered database systems in order to achieve interoperation introduces a variety of functions that must be provided by an IB. These must be combined with characteristics of an IB as described above, and include (see also [38]):

1. communication facilities which allow the IB to invoke brokered database services;
2. search techniques to locate the best match for the client based on the context of the requested service;
3. meta-data query and update facilities
4. conversion of data and reconciliation of incompatible syntax (and possibly semantics) between heterogeneous databases;
5. access to and integration of distributed information.

We view all the above ingredients as effectively constituting possible building blocks of any modern information broker. The units of information brokering can be data objects, methods (services), existing record-based structured data and unstructured data (possibly represented in an objectified form).

There are some parallels between the functions of an information broker, as listed above, and those of the trader service and object requests broker (ORB) in distributed computing systems. Their functions are to manage the interaction between client and server objects. However, they mainly focus on low-level responsibilities ranging from location and referencing of objects to marshaling of request parameters and results. For example, the trader service in CORBA [30] relies on a yellow-paging scheme to enable clients to discover component systems and get information about their services. The trader service relies on consulting the interface repository which describes exported object types, the attributes they export, the methods they export and method signatures.

The standard diagram in figure 1 forms a building block for more complex brokering architectures. As brokers cover sets of information sources, one can aggregate several brokers to form a federation. The *hierarchical topology* in figure 2 considers information brokers as information sources subject to brokering. The result is a meta information broker: it collects and abstracts information about the contents of MDBs. Hierarchical topologies require sophisticated abstraction mechanisms for describing the schema (MMS) of the collected information (MMDB). They are typically encountered when heterogeneous information sources (coordinated by information brokers) are associated with heterogeneous design environments (coordinated by meta information brokers) – for instance, when integrating systems developed under multiple software development cultures in organizations, e.g. as the result of a merger, and so on.

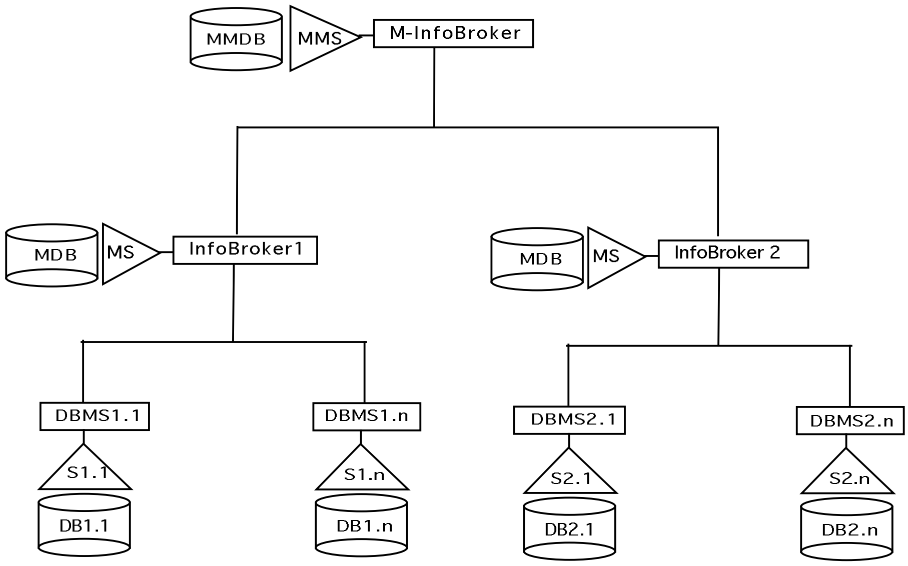

Figure 2: Hierarchical coupling of information brokers

When there is less emphasis on design/engineering environments and more on decentralization of information management the *flat topology* (figure 3) is used. This topology avoids the necessity of abstraction by treating each broker as a unit at the same abstraction level. Their meta databases all describe knowledge about brokered database systems, presumably under a common formalism or at least a common service interface. Coupling of information brokers is realized by a query gateway. Queries beyond the scope of an information broker can be answered by referring them to external meta databases.

Flat architectures are preferred in open environments where links to external data sources are locally detected and established on demand. The actual network of links between information brokers is a result of a distributed process without central control or goals. The common language between any pair of brokers must be established by access to a common service protocol, or through bilateral negotiation when they first establish contact. Depending on the nature of this common service protocol any combination of hierarchical and flat topologies can be realized.

In the following we describe three approaches based on the functionality and rep-

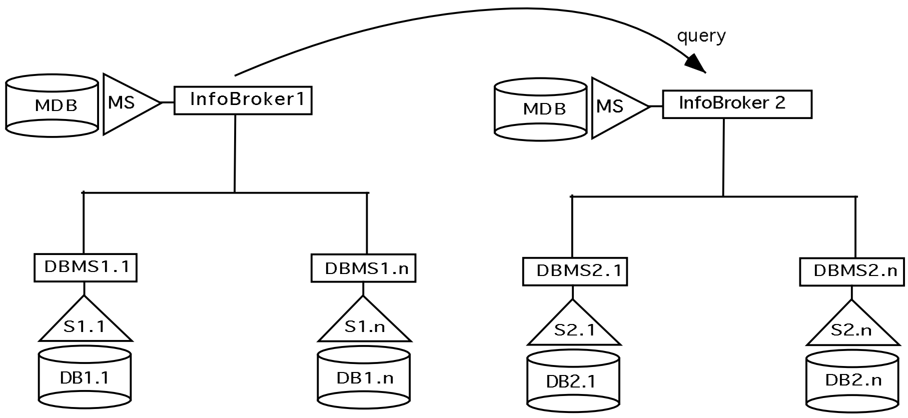

Figure 3: Flat coupling of information brokers

resentation facilities of the IBs as described above. In particular, we concentrate on three aspects of IBs:

- semantically assisted search facilities;
- access and change management;
- translation between heterogeneous data models.

The first IB uses a meta schema that allows for enriching the search process by semantic information called about the brokered databases. The second IB is focused on design and evolution aspects within a cooperative information system and thus maintains a rich meta schema covering the software life cycle of the brokered distributed system. Finally, an IB is presented that is specialized on syntactical data translation employing classification of database schemas into a hierarchically organized meta model.

3 The Semantically-assisted Search Process

The rapid growth of information available across machine boundaries in networked environments is placing severe burdens on methods of information discovery and sharing. The multitude, diversity and complexity of on–line information sources (typically exemplified by database systems) makes discovery of appropriate information a significant obstacle. Not only is there a huge number of items to be accessed, but naming conventions, data structures, meanings and modes of usage can vary significantly. Information sources have heterogeneous domains of expertise, extents of domain coverage, and use different vocabularies to deal with a particular domain. Users of such systems are often overwhelmed by the amount of cross-domain subject, terminology as well as system knowledge that is required of them to access this information in a combined and coherent manner.

We consider that devising a scalable approach to information elicitation is critical to the success of open database networks and networked applications. Critical to the success of this approach is the content and organization of meta data in the information broker which will assist the search process. For this we require that the meta database of the IB possesses the ability to represent the following types of meta data:

1. *Description of a brokered database domain:* this contains information about terms, composition of terms, remarks about the meaning of terms, hypernym, hyponym, antonyms-of, part-of, member-of (and the inverses), pertains-to relations and lists of keywords. It may also include certain details such as: geographical location, access authorization and usage roles, narrative explanations regarding corporate term usage, domains of applicability and so on.

2. *Associations between brokered database terms:* These may come, for example, in the form of a semantic network which connects term summaries (see step-1) found in the brokered database nodes. Each of these term summaries nodes (in the semantic network) defines a common structured vocabulary of terms and a specification of term relationships within that particular subject and links to other related subjects.

3. *Linkages to external brokered database domains:* this contains information about how the term summaries found in a brokered database domain domain relate to other term summaries found in different (but related) brokered database domains.

3.1 Logical Organization of the Information Space

To facilitate pro–active elicitation of *schema information*[1] from multiple inter-networked information sources, we provide a consistent, organized view of semantically related schema partitions in logical space across multiple discrete heterogeneous information sources. Database clusters are formed on the basis of subject areas of cross-database interest and co-occurrences of semantically related terms in these subject categories, so

[1]Here the emphasis is on schema information items and not on data values. Accordingly, we use the word information to imply schema information throughout this section.

that the elements within a cluster have a high degree of semantic correlation. Hence, the information space is "concept driven" rather than pre–specified or explicitly classified. To form database clusters we record meta-information (refined and enhanced meta-meta data) about cross-database subject categories in high-level objects called *generic concepts* (GCs) and then link the database nodes to them depending on their target areas and "interest". GCs essentially represent centroids of customizable information space - around which databases cluster - and are engineered to span the universe of discourse. Databases form weighted links to generic concepts in ways which reflect their particular interests. We may broadly think of a GC as being a semantically enhanced version (containing the meta-data described in section 3) of the meta-data content of an IB.

A single database may be simultaneously involved in several clusters of databases to varying degrees, as dictated by the weights of its links to the various GCs. A *strong link* from a database node to a certain GC, implies that this database node agrees to associate with other nodes in its database cluster. The resulting GC structure forms a massive dynamic semantic network. Within this network, the interaction of generic concepts, and locally known link weights, gives rise to a node's *partial global view* of the entire network of databases.

The organization of the conceptual information space can be better understood by considering the case of the Universal_Accreditation_Company database which is set up to provide networked information about accreditation of courses and subjects between institutions and various private/public educational training providers and other similar or related services. In its original form, the Universal_Accreditation database maintains information on various education service providers, their courses, accreditation committee members, accreditation processes and so on. Now, after being connected to other databases it can also provide *matching* information about enrollment programs, training schemes, research activities and publication titles. The Universal_Accreditation_Company database node, along with a partial representation of its associated relation schema, and its weighted interconnections to its surrounding GCs (and their associated databases) is shown in Figure 4. This configuration is based on the flat topology (figure 3) as described above since communications are sparse and demand-driven.

Figure 4 also illustrates that a database node (e.g., Universal_Accreditation) can be totally described in terms of three sections (used for viewing by users): a *feature descriptions*, a *context graph*, and a *GC connections* section. The feature descriptions section collects all the information associated with the description of a brokered database domain (as described in section 3). The feature descriptions entries (for the

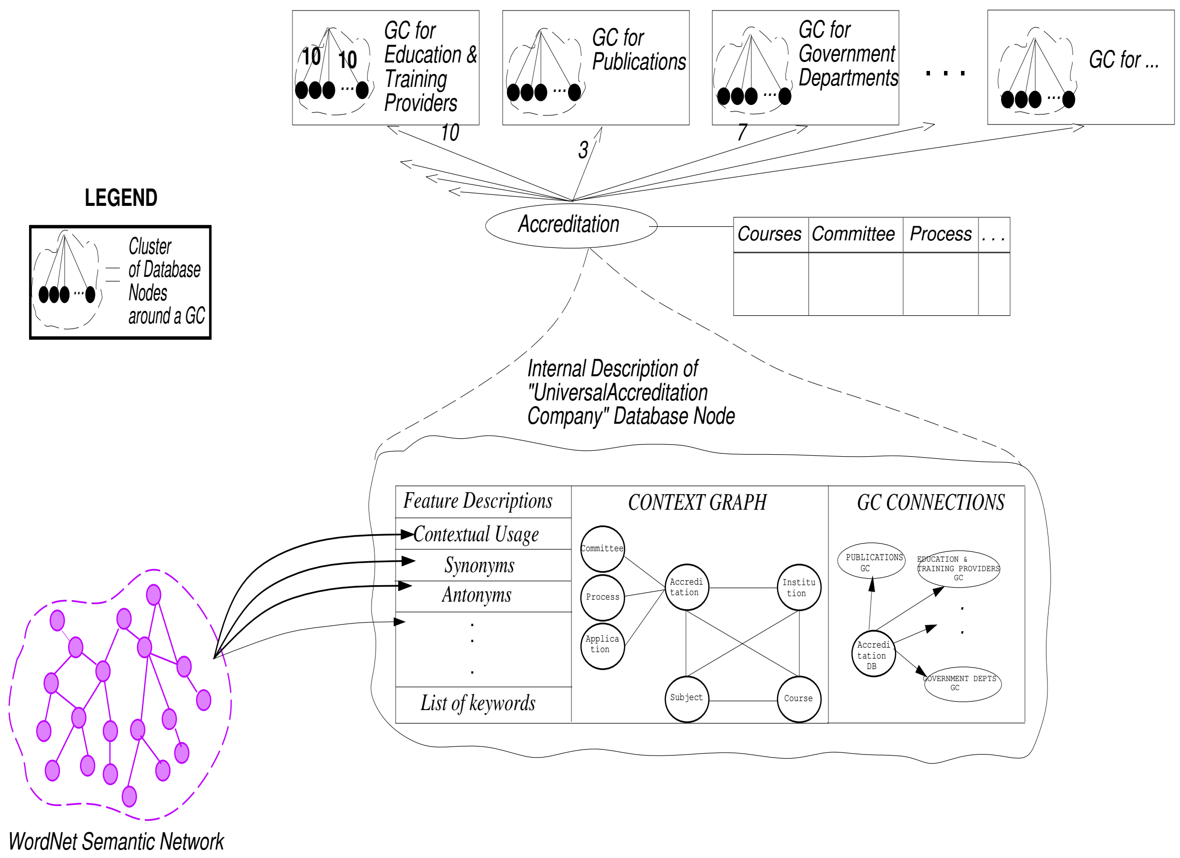

Figure 4: Connecting a Database Node

particular system described here) are generated by the linguistic tool WordNet [24] that supports semantic term matching through the use of an extensive network of word meanings of over 30,000 terms connected by a variety of textual and semantic relations, (see bottom of Figure 4). The context graph section contains a non–directed graph which connects term summaries (in the form of subject descriptor nodes) found in the Universal_Accreditation database node. These subject descriptor nodes and their link structure are used in the clustering of databases to form the generic concepts and its purpose is to establish associations between brokered database terms (see section 3). Each of the subject descriptor nodes defines (in conjunction with its respective entry in the feature descriptions window) a common structured vocabulary of terms – describing the term in question, e.g., *course*, – and a specification of term relationships for that particular subject, e.g., with em subject or *institution*. Finally, the GC connection section shows how the Universal_Accreditation database is related, i.e., link weights, to other GCs in the network.

3.2 Abstracting Meta-Information and Forming Ontologies

By analyzing the co-occurrence probabilities of terms in specific subject categories (described in terms of GCs) we create a *concept space* for each subject category. Accordingly a GC provides an *ontology* to describe a cross-database application domain and facilitates access to the information services within that domain. For example, the Education and Training Providers ontology provides a common basis upon which database nodes dealing with enrollments, courses, training, accreditation etc. (Figure 4), achieve an understanding of each others information content. This ontology consists of abstract descriptions of classes of objects in the domain, relationships between these classes, terminology descriptions, an on-line thesaurus and dictionary and establishes a common vocabulary for interacting with the GC and its underlying information sources. Such a concept space represents the important terms within concept by recording naming equivalences between inter-component exported meta-data elements, contextual usage (narrative descriptions), term senses, term genericity, a list of key-words, and other domain specific information, and applies them to the entire collection of members of a GC. The GC structure is akin to an associative thesaurus and on-line lexicon (created automatically for each subject category).

Overall the system may be viewed in terms of three logical levels. This logical structure is shown in Figure 5, and includes the database node depicted in Figure 4 to give an overview of the system (and its scale). The bottom level of Figure 5 depicts a simplified view of diverse database nodes (ovals) and their schemas in relational form. The middle level represents *exported meta–data* for the database schemas. Meta–data contains a description of the schema items' structure (e.g. relations, attributes, fields, references, modes of authorization). The top level corresponds to the *meta-information*, or generic concept, level. This level contains abstract dynamic objects which implement the clustering of related portions of the underlying component meta-data and materialize the GCs in an object-oriented form [25]. In the case of multiple database nodes, a common ontology, such as Scientific Publications, Government Departments etc. (Figure 5), is implemented by a specific GC, and serves as an agreed upon context for interactions between a group of related database nodes. This type of meta-information is the key ingredient to information elicitation in distributed, scalable systems. It allows independently developed system components to dynamically discover each other and to collaborate. It also enables the system to configure itself and to adapt to extensions and upgrades. To simplify things we have chosen to illustrate only the strong links of three database systems: Scientific Publications, Government Departments (both represented by solid arcs) and the Universal_Accreditation database (represented by dotted

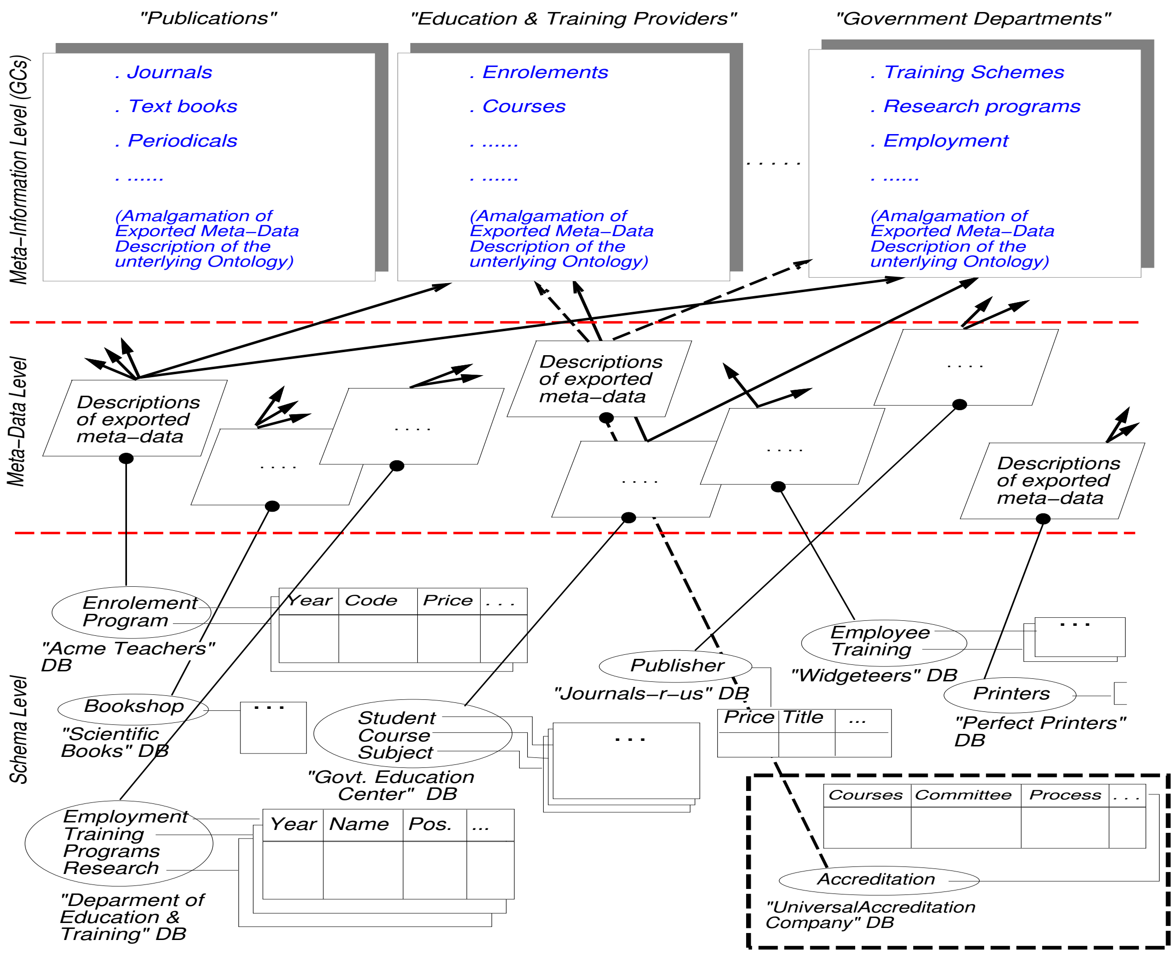

Figure 5: Levels of Abstraction in the Three–tier Inter–database Organization

lines).

The aggregation of the context graphs (Figure 4) from various database nodes, results in the clustering of inter–related information resources. For each semantically related database node group, a GC is created to represent the area of interest (or concept) that the group embodies, e.g., Education and Training Providers GC for the Employee Training, Accreditation, and Government Education Center databases as depicted in Figure 5. This process is achieved by obtaining pairwise similarities between subject descriptor nodes within the context graphs of the various database nodes and by applying a hierarchical agglomerative cluster generation process on the basis of a similarity metric that takes into account inter-node pairings (in graphs representing different nodes) and comparison of graph structures. The comparison and clustering process is described in [25].

3.3 The Information Elicitation Process

The overall information elicitation process can be viewed as a multi–level process which requires a logical organization and powerful responsive tools that provide feedback relevant to the user's interests and needs. Global concepts assist in meeting these requirements because they provide compact stratified descriptions of the structure, semantics and terminology of the contents of a database cluster (underlying this GC). Therefore, they facilitate browsing and associative access.

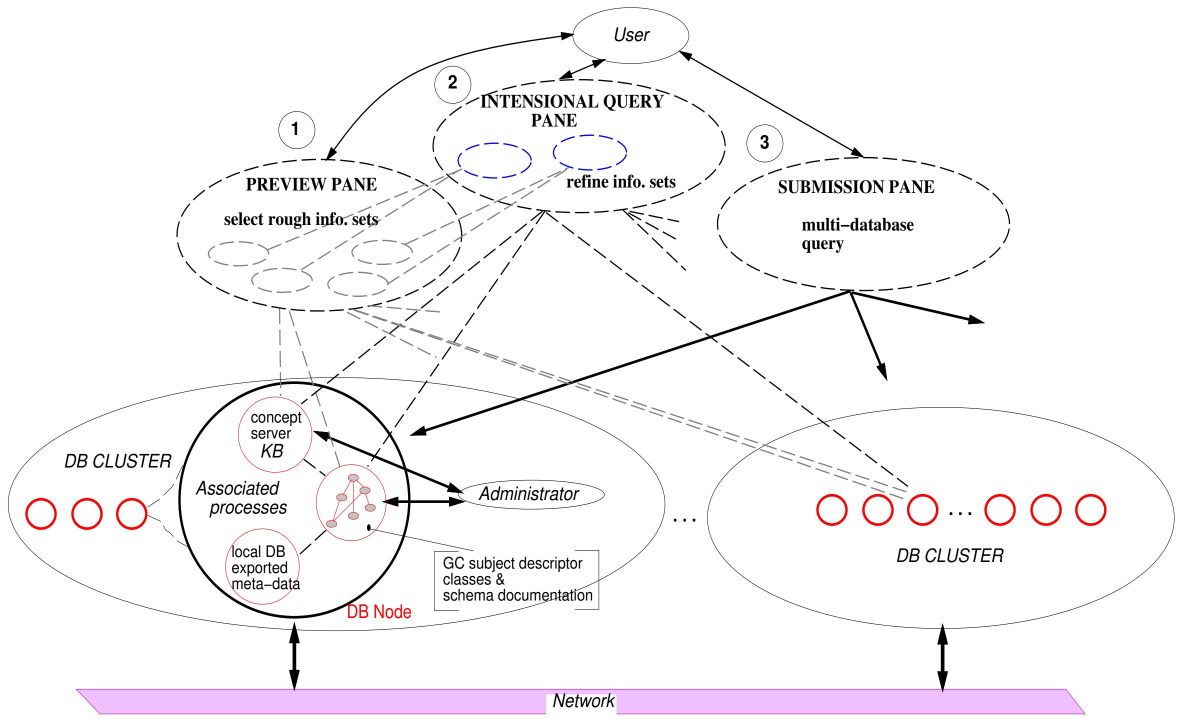

Figure 6: Top level view of the information elicitation approach for networked databases.

Figure 6 depicts the top level view of an open database network comprising a set of interacting database nodes. Each database node in the network stores in a repository the meta-data schema elements it chooses to export. In addition to this repository each database node comprises the GC meta-level class graph (in the form of a net combining abstract classes with terminology descriptions, an on-line thesaurus and schema documentation) coupled with an associated knowledge base for organizing and querying the information sources. The *concept server knowledge–base* (a meta-database in the sense of Figure 1) contains meta-information summarizing the databases clustered around a GC. This meta-information includes the representation of topics from the information sources in a node, as well as properties having to do with the logical characteristics of the local node and relevant remote nodes including: naming conventions and aliases, service descriptors for invoking remote services, symbolic addresses of nodes, weight associations and so on.

In a typical session the user can interact with a database node via three window panes: (1) an interactive browser pane, (2) an intentional query interaction pane, and (3) a pane for submitting multi-database-like queries or transactions (Figure 6), see Figure 6. Database searches are implemented as a set of decentralized interoperable *information space elicitation* (ISEs) processes. These allow the user to deal with a controlled amount of information and provide more detail as the user looks more closely. ISEs implement a GUI–based browser which combines exploration of remote site meta–data and a search engine, in order to scan and review information relevant to a specific search task. ISEs facilitate the exploration of information/meta–information at a greater level of detail (results are displayed in the preview pane Figure 6). Matching data are shown graphically on preview bars which aid users to eliminate undesired schema description information, and focus on a manageable number of relevant information sets.

When the number of information items displayed in the preview pane is low enough and includes a sketchy description of the search target, *intentional*, or schema queries [34] – which return meta data from selected schema items – can be posed to further restrict the information space and clarify the meaning of the information items (displayed in the intentional query pane Figure 6). The intentional querying pane allows the user to ask for properties and documentation related to the conceptual structure of schema items and supports incremental refinement of the information elicitation process. This iterative process prevents zero–hit queries and supports the formulation of meaningful multi-database requests (constructed in the submission pane Figure 6).

3.4 Related Work

Information brokering has its roots both in distributed environments and the informal, text-oriented world of information retrieval. Somewhat orthogonal to this traditional distinction, recent activities in distributed directory management, WWW search engines, and distributed agent systems are approaching the problem from a more generic, AI-based perspective.

Distributed environments use resource discovery tools in the form of directories of services, or trading facilities, for enabling client processes to choose appropriate services in a dynamic fashion [31, 2, 30]. Distributed computing environment directories support features such as white pages (straightforward name-to-entry lookup facilities), yellow pages (lookup facilities based on descriptive object attributes) and group services (mapping a name into a set of names) to allow users to identify resources by name. Current approaches to distributed environments assume that complete description of the services available through out a distributed environment are specified a priori and do not normally change. This is an unrealistic assumption when considering a network with large numbers of databases each of which contributes to a large number of data items.

In most *information retrieval* systems, the emphasis is usually on how to build an indexing scheme to efficiently access information given some hints about the resource [36]. Most of the distributed information retrieval systems are designed to work in a homogeneous environment. There has been some work to extend schemes to a network of heterogeneous information retrieval systems [37]. In [1], an approach is described that relies on *external indexing* for finding information in a network of information systems. Each node of the index contains a network address along with a set of condensed descriptions called *skeleton*. Resource providers are added to the index using knows-about relationships. This approach tends to centralize the search in a single index and is used for the actual resource discovery. Potentially (if users make queries about all existing information space), all nodes would have the same index. It is not clear from the above references how the actual node selection is performed.

The recent growth of the *World Wide Web* as a distributed multimedia information system stimulated the development of programs that build directories for WWW documents. The programs, also known as *search engines*, systematically scan the WWW and extract information *about* documents in a directory. A user then can submit a query to the directory to find documents matching the search criteria whereby document the mode of access moves from hypertext-based browsing to content-based search [9]. Simple search engines just extract keywords from a document and provide the above-mentioned information retrieval technique to match a query with a directory entry. The problem with this approach is the low precision: keyword-based search does not take into account the context of information. More sophistication techniques are currently under investigation. These include the exact classification of documents by using techniques based on neural networks [9] and tackling the search precision problem based on a federated architecture of search engines [5].

Software agent systems have concentrated on improving information discovery methods on the WWW by employing some form of knowledge representation to enable more sophisticated representation of information sources and inferencing abilities [28], [11], [7]. Two of the most notable activities are: *information matchmaking* [21] and *information brokering* using *context logic* [11].

Matchmaking is an automated process whereby information providers and consumers are cooperating assisted by an intelligent facilitator utilizing a knowledge sharing infrastructure [21]. Matchmaking depends on messaging and content languages and allows information providers and consumers to continuously issue and retract advertisements and requests, so that information does not become stale.

Fikes et al. [11] describe a tool-kit for information broker development based on the *Ontolingua* system [12]. Ontolingua is an integrated tool system for developing domain-specific ontologies in the Knowledge Exchange Format (KIF) and for translating the resulting ontologies into application-oriented representation languages.

4 Brokering Change in Virtual Organizations

The previous section presented a bottom-up approach for integrating heterogeneous databases by an information broker that links those databases by a semantic network of generic concepts. In this section, an information broker is used for the top-down design and incremental evolution of a distributed information system. The application domain for such design-oriented brokers is best described by the term virtual organization: a small number of departments has an interest to support cooperation for a common goal by linking their information systems. Examples include joint projects, strategic alliances, or simply a particular new perspective on an existing distributed business which requires different information paths than the established ones.

Frequent changes in the virtual organization require the ability of the distributed information system to adapt to those changes. An information broker for supporting such changes should consequently have concepts in its meta database to represent

- the virtual organization in an abstract way that allows to reason on properties of the information and workflow,
- the distributed information system that is used for implementing the operation of the virtual organization, and
- the dependencies between the abstract representation of the virtual organization and its implementation

Such an IB has to manage multiple data models: conceptual models for workflows in the virtual organization, logical database schemas, activity models that describe the proper procedures within the virtual organization, and others. Since changes to all models may occur at any time, an update IB with the ability to change concepts at all IRDS abstraction levels is the logical choice. Full access to all IRDS levels also means that the designers of the information system may define the kind and diversity of representation formalisms within the IB's meta database. Diverse representation languages are typical for information systems for virtual organizations because the participating design teams have their specific suite of design and implementation languages. A meta model at the top IRDS level is used to formally define the inter-relationships between the multiple models and to provide the facilities to integrate the heterogeneous part models of the brokered databases.

As a running example, we shall use the meta database of *quality management in manufacturing organizations*. Quality management involves all departments of an organization. The tasks of quality management use information produced during the product life cycle but not at the same aggregation level. The patterns of interactions between the involved persons, the so-called workflow, are very sophisticated and time-variant. Changes in the workflow frequently require new information sources (or sinks) to be established. In many organizations, the workflows cross the organization's own boundary. Therefore, both distributed engineering as well as distributed operation are necessary to support quality management by an information system. The experiences reported below stem from WibQuS [16], an interdisciplinary project on cooperative quality information systems conducted in Germany between 1992 and 1995. Figure 7 shows the steps of the broker development process, from the basic idea of a quality cycle embedded in the product life cycle, via the definition of a shared meta model and the distributed conceptual modeling of the domain, to the mapping onto an implementation that uses a federated SQL system (here: OMNI-SQL from Sybase Inc.) as the technical interoperability layer. All described aspects of the information broker were implemented using ConceptBase [17], a meta database manager which supports the Telos language, and has been linked to the OMNI-SQL system for the purpose of the project.

Figure 7 shows the dual role of the information broker. The *design* part contributes conceptual models for information flow, workflow, and database schemas. Moreover it provides the mapping of these models to implementation models (relational database schemas, query definitions, data structures for workflow enactment). The second role of the IB is *operational*: it initializes the brokered database schemas, it evaluates multi-database queries, and it brokers task delegations between users of the brokered databases. Though the IB is implemented here as a single component, its full IRDS representation makes it an example for the hierarchical topology shown in figure 2. The meta model can be regarded as the schema MMS of a meta IB. Its purpose is to define a collection of modeling languages for designing distributed information systems for virtual organizations (see subsection 4.1 below). For a specific virtual organization,

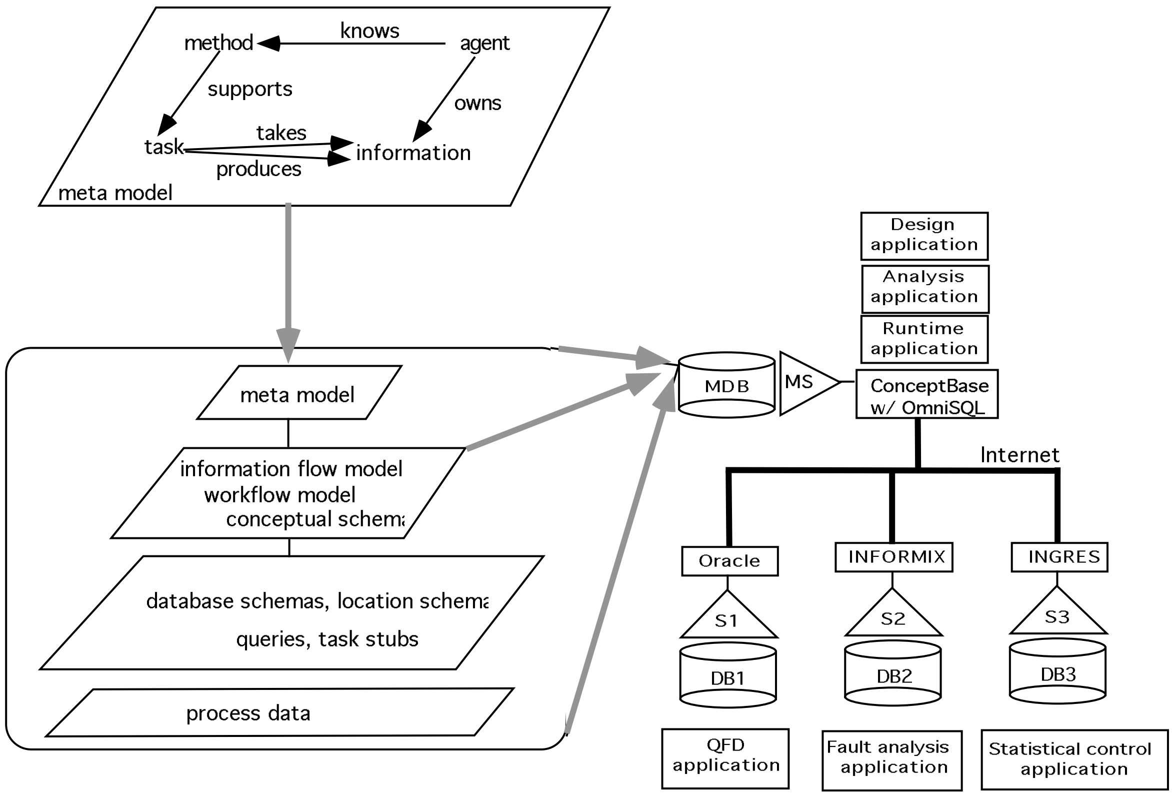

Figure 7: Information broker for design and operation

an information broker is then attached below this meta IB to support design and operation. As a consequence, different IB's for virtual organization that share the same meta model can be compared via the meta IB. This potential function is not elaborated in this paper.

In the following, the functions of the IB are described in more detail. They are realized by the application programs (fig. 7) which manipulate the meta database of the information broker.

4.1 Meta Model Design

The purpose is to define a set of key concepts that shall be used in later modeling and design phases. The meta model is the schema of the design level information broker to be included in the running system. The definition of the key concepts itself is a cooperative design process. It determines the language in which designers may represent their knowledge about the information systems to be integrated. Therefore, the concepts are carefully negotiated in advance in order to focus the distributed design process on the expected problems, thus reducing the need for subsequent re-negotiation and re-modeling at the detailed level. In the case of quality management, four such concepts were identified with corresponding relationships (compare upper part of fig. 7):

- *Information* covers alls kinds of data to be exchanged between distributed quality information systems and their users;

- *Tasks* represent steps in the virtual organization to fulfill the common goal (here: quality management). Tasks can be decomposed into sub tasks. They take some information as input and produce some new information. Tasks are the typical unit of delegation in a workflow.

- *Methods* are prescribed procedures for performing a certain task. The prescription may be highly formal (e.g., a computerized program), or informal like a handbook in natural language.

- Finally, *agents* represent individuals or organizational units involved. They appear in different roles in the meta models: in terms of their capabilities (with respect to methods), as owners of information, as executors or delegators of tasks, and thus as resources in a planning context.

The meta model is itself part of the meta database and subject to change. Is is represented in the language Telos [26] that has been extended for meta modeling expressiveness in the ConceptBase meta database management system [17]. The graphical representation of the meta model in figure 7 is excerpted from the complete logic-based representation of the meta model. The Telos language allows to represent links and nodes uniformly as objects. Special link types are reserved for instantiation of an object x to a class c (denoted by a binary relation In(x,c) in Telos) and specialization/generalization between two classes (denoted as Isa(c,d)). Attributes between objects as well as between classes are represented as relation A(x,m,l,y) between two objects x and y where 'm' is the attribute category and 'l' is the attribute label. In ConceptBase, updates to the meta database are done by updating these relations. Queries, integrity constraints, and rules are represented as restricted first-order formulae and translated into the Datalog [8] query language which allows efficient computation of query results.

Note that the meta model covers different aspects of the information system. The task concept together with the information concept forms an information flow language. The information object together with includes and refersTo links not shown in fig. 7

forms a conceptual data model equivalent to Bachman diagrams [39]. Finally, the agent concept together with task and information forms a workflow model. All three part meta models are inter-related by the overlapping of concepts, e.g., information is used in all part meta models. This is exploited in the distributed conceptual modeling phase and in the subsequent analysis phase presented below.

4.2 Distributed Conceptual Modeling

Distributed conceptual modeling is the second function of IB for virtual organizations. It belongs to the 'design application' running on top of the ConceptBase system (see fig. 7). The meta modeling facility provides different perspectives [27] on the modeling task: information flow modeling, conceptual data modeling, and workflow modeling. The overall modeling process is characterized as follows:

1. the design teams work in a distributed process on their portions of the modeling task, contributing part models
2. the modeling is heterogeneous is the sense that different modeling languages are used for different purposes

The IB supports the conceptual modeling via the perspectives extracted from the meta model. Each perspective forms a graphical modeling language. We use the conceptual data model (concept information) for elaborating this claim. It has to be shown that the expressiveness of Telos is suitable to represent a data model, i.e., the data structures and the axioms. The data structure is given by the concept information comparable to the concept 'entity type' in Entity-Relationship- Diagrams. Instances of information are represented by the binary predicate In. For example, In(Product,information), In(Weight,information) denotes two entities in a conceptual schema. Properties of such information entities are denoted by attributes, for example, A(Product,includes,part,Product), A(Product,refersTo,weight,Weight).

The axioms of the conceptual data model are first order formulae represented in Telos that define legal instances of a conceptual schema, here:

- type consistency: an attribute value must belong to the right concept, for example any object 'w occurring as weight attribute of a product must be an instance of Weight, or generally

$$\forall\ C,m,D,x,l,y\ In(x,C) \wedge A(C,refersTo,m,D) \wedge A(x,m,l,y) \Rightarrow In(y,D)$$

- non-circularity: attributes of category includes must be non-circular, for example, a product p1 that has a part p2 may not be part of p2 (in its transitive closure), i.e.,

$$\forall\, C, m, D, x, l, y \quad In(x, C) \wedge A(C, includes, m, D) \wedge A(x, m, l, y) \Rightarrow \neg \exists\, l_1\ A(y, m, l_1, x)$$

Both formulae are integrity constraints and are included in the meta model concept information within Telos. The axiomatic definition of the information flow modeling language and the workflow modeling language are similar (see [18] for a formal representation of the axioms).

As a consequence the distributed conceptual modeling with heterogeneous modeling languages can be fully achieved by making updates to the meta database of the IB. Conceptual data modeling contributes instances of information, information flow modeling contributes instances of task and its two links to information, and workflow modeling contributes instances of agent and its links. A graphical editor with the ability to display dedicated graphical types for the four concepts of the meta model can serve as a user interface for the designers to the meta database.

4.3 Cooperative Model Analysis & Integration

The models generated in the previous step are typically incomplete and inconsistent. Moreover, they come from different design teams using different vocabularies. Therefore, the IB provides an analysis function to detect such weaknesses. It is realized as an application running on the meta database holding the conceptual models. We investigated three techniques for the analysis task: querying, voting, and simulation.

The *querying technique* exploits the redundancy in the perspectives. The part models described above are linked by overlapping concepts. This provides a partly redundant conceptual modeling process because their instances are modeled in more than one perspective. For example, the information instances are modeled both in the information flow model and the conceptual data model. Thereby, any information instance occurring in one part model but not the other indicates an incompleteness. This trick is applied to all four abstract concepts of the meta model. The incompletely modeled instances can be detected by a simple query to the meta database: give all instances of meta model concepts that are defined one part model but not in an overlapping part model. A second type of queries are more domain-specific. For example, a task taking some information as input that was never modeled as output of another task is indicating an error in the information flow. In many cases, the information is delivered by another task but named differently. In WibQuS, about three dozens of queries were defined and imposed on the meta database. The answers indicating errors or incompleteness are returned back to the design teams who refine their part models to cope with them.

The query mechanism is well-suited for detecting flaws within the meta database. However, how can a distributed design team decide whether some parts of the conceptual models are relevant for the purpose of the virtual organization? The models may well be consistent but may have little to do with the application domain. A *voting technique* can be used to identify such irrelevant or semantically incorrect definitions. The voting process is similar to a review process for a scientific conference. For each defined concept, the meta database memorizes its author by a simple attribute. Then, members of the design team take the role of reviewers for the defined concepts. Possible ratings are 'accept', 'not-accept', and 'unsure'. These ratings are also assigned as attributes to the defined concepts within the meta database. After the voting, the concepts are ordered by their rating. The group then decides which concepts should be retracted from the conceptual model or refined.

The last model analysis technique is *dynamic simulation* [35] of the models. The information flow model has a dynamic interpretation in the sense that some chunks of information are passed between agents. By providing additional parameters, such as time to perform a task, available resources, and ability to store information one can identify bottlenecks in the information flow via dynamic simulation. Long-term effects can be revealed by modeling factors like increase of organizational knowledge and human expertise. Empirical studies of the approach have shown the validity of the simulation results. Technically, the dynamic simulation package is an application program that reads the information flow model (with the additional parameters) from the meta database and then performs a simulation run.

4.4 Schema & View Implementation

The method models present a conceptual description of the planned system. They use terms on the level of abstraction for the semantic search in section 3. To build a running system, the information concepts have to be implemented by data structures (schemata), the tasks by operations of the participating systems, and the method concepts by procedures or workflows. We concentrate on the schema and view design using relational databases as implementation platforms. Each data transfer is implemented by moving tuples between the database systems of the participating methods.

Views are the simplest realization of inter-method tasks: they realize the transfer of data between distributed databases. The information broker supports the mapping by an abstract model for relational schemas and views. Both are maintained in the meta database and associated to the conceptual models by implements[2] links [18]. Like in the analysis activity presented above, a collection of queries guides the implementation process. The central query defines unimplemented information concepts (C) as those objects with no implementation concept (D) associated to it. Note that the formula below is part of the (deductive) meta database. At any time, the set of unimplemented concepts can be retrieved by querying the relation In(C,NotImplemented).

$$\forall\, C \quad In(C, Information) \wedge \neg\exists\; y, m\;\; A(D, implements, m, C) \\ \Rightarrow In(C, NotImplemented)$$

This approach supports evolution. Whenever a new information concept (or relation schema) is defined the analysis queries expose the need of incremental implementation (or reverse engineering). The implementation process is represented within the meta database of the broker. It can be made effective by broadcasting it to the participating database systems. Not any update is legal, e.g. removing a relation in a schema that is accessed by some (relational) view definition is forbidden.

4.5 Runtime Operation

The runtime function of the information broker initializes the brokered databases and supports the data and task exchange during operation.

The logical database schemas represented in the meta database are passed by an initialization step to the brokered databases (schemas Si in fig. 7). By using standard database systems, the SQL interface ('create table') of the databases provides the necessary operations. Thereby, the schemas of the brokered databases become materialized views of their abstract representations in the meta database.

The information broker also controls the actual exchange of data between the systems. Information flows have been planned by using the task concept. Simple remote database accesses can be realized by view definitions in a federated database system. The views are included in the schemas of the participating database systems. Thus, whether an access is local or remote is transparent to the user of the system.

A second category of task are workflows between the users. For example, the engineer responsible for the quality function deployment may ask the service expert to

[2]This link is basically the same as the connection of database nodes to context graphs in section 3. The conceptual schema in this case is a network of information concepts linked to each other by refersTo and includes attributes. Consequently, the IB can also be used to construct ad hoc queries from the conceptual model.

interview customers about the satisfaction with some product feature. This information is not contained in any database at the time the request is issued. The information broker serves as an agent that collects such requests according to the tasks in the method model. A request is formally an instance of the concept `task` in the method model. It changes its state (issued, accepted, rejected, declared-complete, complete) following a negotiation protocol based on Winograd & Flores [23]. State changes are reported to the information broker triggering messages to the involved experts. The list of possible recipients for a request can be obtained from the information broker by asking for instances of agents in the method model who can perform the requested task. In this sense, the information broker becomes a workflow engine.

A third kind of task are searches for information where no view definition exists but where the information is stored in the distributed system. The interactive discovery of potential information sources can be guided by the broker as in section 3. The semantic layer of the search is the integrated conceptual model. The selection of the appropriate sources then leads to a skeleton of a query that realizes the new task. Thus, ad hoc searches for information are equivalent to online evolution of the conceptual model and its implementation.

4.6 Related Work

The IB presented here combines results from the area of software engineering, especially design languages, with methods for enterprise modeling.

Conceptual modeling languages like Entity-Relationship diagrams [10] and its successors have been widely used and are standard parts of *software engineering* environments (SEE) like Oracle CASE [32]. Such tools even provide partial code generation from the conceptual layer to implementation languages. Typically, the set of modeling languages is fixed as well as their axiomatic definition.

Adaptable *graphical modeling* languages are also provided by the MetaEdit [22] tool. It allows to represent the syntactic properties of the languages, and then to generate a specialized graph editor for it. The difference to the IB here is that its meta database axioms cover not only the syntactic properties (one instantiation level within IRDS) but also model-semantic properties like non-circularity of includes links (two instantiation levels within IRDS).

The recent trend to support change in virtual organizations by information technology has lead to an initiative for *enterprise modeling* [4]. The trend there is to enhance existing modeling languages in order to scale them up to the complexity of enterprises. One aspect are activities within enterprises which are represented by workflow models with a rich set of constructors (parallel execution, exception handling, non-deterministic choice). The CIMOSA methodology [20] views an enterprise model as collection of organization, resource, information and function models. The first corresponds to the agent concept, the second has no direct correspondent in the meta model shown here, the third one is obvious, and the last is comparable to the task concept. The main difference to the IB approach presented in this section is the adaptability of the meta model. In other words, the CIMOSA methodology is just another meta model.

For business applications, a relatively fixed set of modeling languages on top of generic schema components (coding domains like supply management, payroll service, etc.) are widely used [13]. In contrast, an update IB can be adapted to the specific modeling needs via its meta modeling capability: the modeling languages to be used are modeled within the IB's meta database.

5 Transformation of Information Representations

Having found an information source via the broker does not necessarily mean that it can be accessed by some application program. The search yields information sources based on semantic knowledge about information independent from its representation. Data processing requires conformance to data formats expressed in schemas which are themselves expressed in a some data modeling languages. To exchange information between heterogeneous systems, data structure transformation is a necessary step. The problem has been investigated from different viewpoints ever since multiple applications started to access shared data. The task of data transformation has the following degrees of complexity:

- *Value transformation*: Data elements of the same data structure have different units. A transformation procedure can be derived from the relationship between the involved units. A typical example is the transformation of temperature values measured in Celsius into Fahrenheit degrees. Here, the relationship is a simple algebraic formula. A second example is the transformation of currencies. Here, the relationship is an exchange rate table that may dynamically change over time.
- *structure transformation*: Data elements are represented in different data structures within a homogeneous data model. The transformation procedure is inductively build out of transformations between the parts of the two data structures. For example, an employee record with a set-valued project attribute may be normalized into a flat relational representation.

- *data model transformation*: Data elements are represented in some data structures of different data models. This hardest kind of transformation since it subsumes the previous transformation together with the task to find transformations between heterogeneous data models. A well-known example is the transformation between entity-relationship model (ERD) and the relational model (though ERD data bases hardly exist). The transformation requires to find the relationship between the type constructors of the participating data models.

Here, we concentrate on the data model transformation. Two approaches have been proposed in the literature.

The first is known under the term *canonical data model*. It subsumes all other data models in the sense that any information can be mapped into and from its canonical representation. The transformation of a data element from a source data model into its target data model is divided into two steps: First, the data element is transformed into its canonical representation. Second, the canonical representation is transformed into its target representation. This approach requires the establishment of two transformation methods per participating data model.

The second approach is *pairwise transformation*. For two participating data models transformations in both directions are established without an intermediate representation. Here, a quadratic number of transformation methods has to be implemented if all directions are required. The advantage of the direct method is that the transformation method between two similar data models can be kept simple. A specialized meta model of representations can be used to define compromise methods that create pairwise transformations, but in a standardized manner described by the meta model.

Figure 8 shows one such meta model [19] a hierarchy of concepts that cover the most widely used data modeling languages. For example, the construct Relation of the relational data modeling language classifies into the concept Link (an element that links other element) as well as TypeUnit (an element that may exist on its own right and has a type). Similarly, the entity relationship model consists of EntitySet (classified into TypeUnit) and RelationshipSet (classified into TypeUndirectedLinkUnit).

The meta model is used as class hierarchy for the data model level below. Using the Telos language as in section 4, the meta database if the IB contains facts In(Relation,Link), In(Relation,TypeUnit) etc. to encode the participating data models as instances of the meta model. The schema concepts below the data model layer are then represented as instances of the data model layer: In(PSE,Relation), In(PSE2R,Relation) etc. We then can add an axiom

$$\forall\ x, C, M\ \ In(x,C) \wedge In(C,M)) \Rightarrow metaIn(x,M)$$

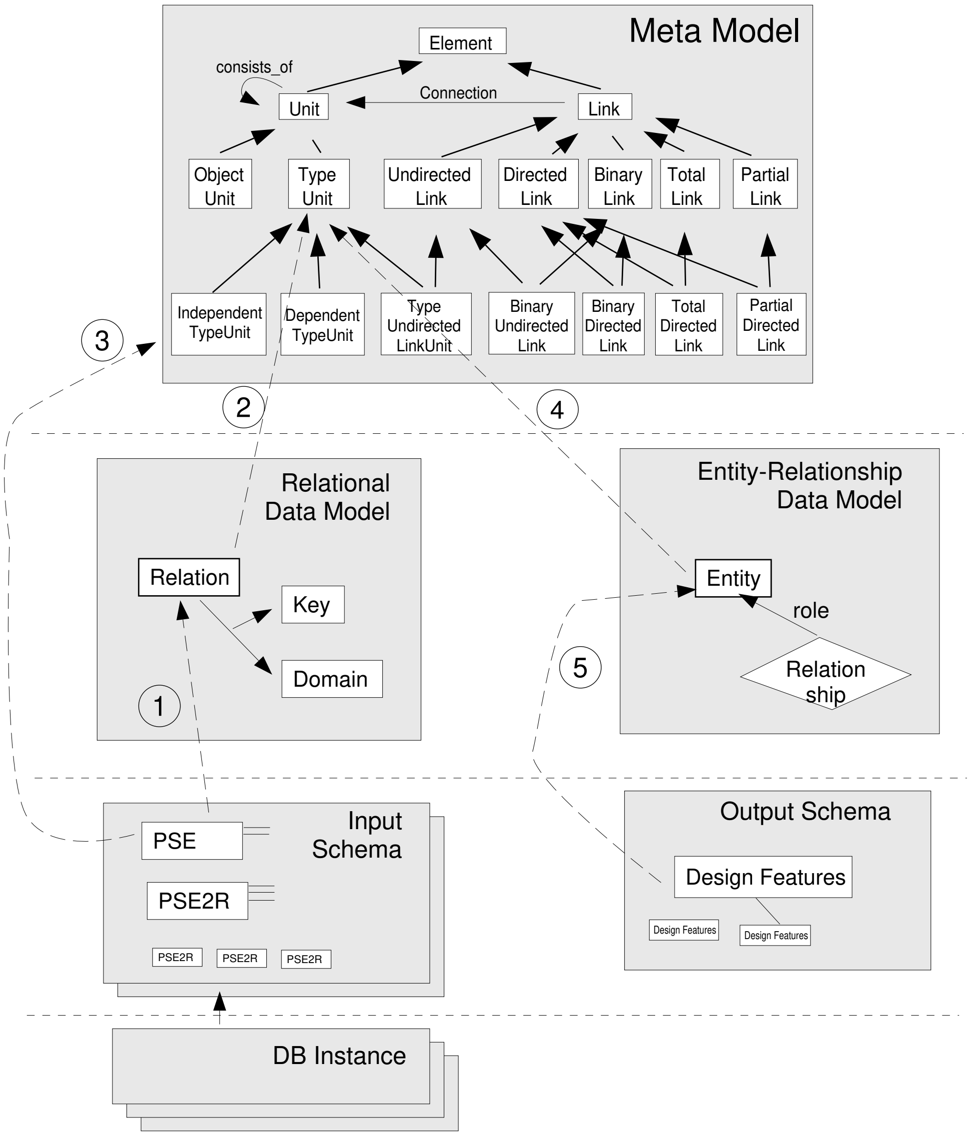

Figure 8: Notation transformation using a graph-oriented meta model

schema level concepts to the meta model querying the relation $\mathsf{metaIn(x,M)}$ of the meta database. The above formula just gives the solutions that are obtained when taking into account into which meta model concept the data model concept is classified into ($\mathsf{In(C,M)}$). More information about the kind of a schema concept can be obtained when accessing the structure of a schema. An example is the meta model concept binary link: the fact that a relation is a link can be derived from its foreign key dependencies. Then, a relation would be classified into $\mathsf{BinaryLink}$ if is has connections two exactly two relations.

The classification of the data model constructs encodes their semantic similarity. The lower part of figure 8 shows excerpts from the WibQuS repository defined under the notation meta model. The numbers indicate, as an example, how the mapping of a relational schema (here: the relation PSE for product structure elements) proceeds at system integration time [19]. First, PSE is classified as a relation (1). From the definition of the relational model, relations are known to be classified (among others) as type units in the meta model (2). Further semantic information (heuristically determined by integrity constraint analysis of the RDB schema, plus possibly interactive ambiguity resolution) allows a further specialization of the meta type to IndependentTypeUnit. This determines the concept Entity to be the most appropriate representation within the ER diagram (4). Browsing within the conceptual quality model entities determines that the entity DesignFeatures is the one represented in the relation (5).

In the example above, we assumed that the entity DesignFeatures would already exist. In a reverse engineering setting, this is not the case. Then, the entity must be created. From a logical point of view, this is not deduction of properties but abduction of possible models constrained by a set of conditions: the output concept to be generated must be instance of Entity and all links of the original input concept PSE must be mapped to corresponding links of the output concept. Due to the diversity of choices (naming, alternative structures in the target data model), the creation process is more complex than the analysis. Current solutions are hard-coded programs with some builtin precedence on the choices.

The IB with such a meta model can be used for transforming schemata of heterogeneous data models. It is even possible to support new upcoming data models, e.g. object-relational models: only the steps (1) and (2) have to be repeated for the new data model (and schemas represented in the new data model). By adding new data model description to the IB's meta database, the IB gets more and more sophisticated in its transformation function.

5.1 Related Work

The main technique for reverse engineering of database schemas are *production rules* [3]. The condition part tests the input schema and the action part of the rule generates equivalent schema concepts in the output schema. For each pair of data model a different set of production rules has to be developed. There is no generic model that represents the common parts of the mapping process as presented here.

Canonical data base schemas were proposed for *multi-databases* [29, 6] to allow uniform access to heterogeneous databases. Each schema of a brokered database is first translated to a canonical data model (schema transformation) and then integrated into a global schema of the multi-database (schema integration). The difference to the IB approach is that multi-databases present a uniform database to the user while IB's offer a set of methods (search, design, transformation).

For the transformation and integration of semi-formal and formal schema notations, the literature exhibits many almost identical *meta models* of graphs (e.g. [15, 33, 22]). They share the idea of nodes and links as basic meta types, which are then specialized according to frequently occurring kinds of constraints such as links representing partial and total functions, dependent vs. independent units, elementary vs. embedded graphs.

6 Conclusions

Information brokers are tools that facilitate various aspects in the integration of distributed databases. Up to now, information brokers were vaguely seen as middleware systems that are placed somewhere between distributed information systems and programs that operate on these systems. In this section, we defined information brokers as programs that offer their services by operating on a meta database representing information about the brokered databases. Thereby, the purpose of an information broker can be circumscribed by the schema of its meta database and their ability to make updates to the meta database.

Search and querying is facilitated by information brokers maintaining a meta database of semantic concepts. Bottom-up integration of new brokered databases is done by representing their schema plus the generic concepts covered by the new database into the IB's meta database. A common structure for search are hyponym/hypernym lattices.

Access and change management for information systems of virtual organization is another application type for information brokers. Here, a combination of both top-down design, schema integration, and run-time request brokering is supported by the IB. The meta database has a rich schema that allows to manage multiple models and their relationships.

Syntactic data transformation from heterogeneous data models forms the third application type for IB's. Here, a meta model that classifies schema concepts independently from the data model solves the task of analyzing the structure of a database schema. Such meta models are typically generalization lattices. The information broker accepts new data models by classifying them into the meta model. The analysis of input schemas is a implemented by queries to the meta database.

The architectural view of an information broker as an application program running on a meta database is the common basis for describing an information broker. Which formalism is best suited for representing a meta schema and the meta database objects remains an open question. Another research area are query and update languages for information brokers that solve tasks that are currently done by hard-coded functions of the IB.

References

[1] R. Alonso and D. Barbara and L. L. Cova "Data Sharing in Large Heterogeneous Information Network" Workshop on Heterogeneous Databases, IEEE–CS TC on Distributed Processing, Dec. 1989.

[2] *ANSA Reference Manual*, Release 1.01, APM Ltd., Cambridge United Kingdom, July 1989.

[3] C. Batini, S. Ceri, S. Navathe *Conceptual Design – an Entity-Relationship Approach*, Benjamin-Cummings, Redwood City, CA, USA, 1992.

[4] P. Bernus, L. Nemes (eds.) *Modelling and Methodologies for Enterprise Integration*, Chapman & Hall, London, 1996.

[5] C.M. Bowman, P.B. Danzig, D.R. Hardy, U. Manber, M.F. Schwartz, D.P. Wessels "Harvest: A Scalable, Custumizable Discovery and Access System", Technical Report CU-CS-732-94, University of Colorado-Boulder, March 1995.

[6] M. Bright, A. Hurson, S. Pakzad "Automated Resolution of Semantic Heterogeneity in Multidatabases" ACM ToDS, vol. 19, no.2, June 1994.

[7] R. Burke, K.J. Hammond "Combining Databases and Knowledge Bases for Assisted Browsing", AAAI '95 Symposium on Information Gathering from Distributed, Heterogeneous Environments, Palo Alto CA, March 1995.

[8] S. Ceri, G. Gottlob, L. Tanca *Logic Programming and Databases*, Springer-Verlag, Berlin, 1990.

[9] H. Chen, C. Schuffels, R. Orwig "Internet Categorization and Search: A Self-Organizing Approach", J. Visual Communication and Image Representation, 7, 1, March 1996.

[10] P.P.-S. Chen "The Entity-Relationship Model - Toward a Unified View of Data", ACM ToDS, vol. 1, no. 1, 9–36, 1976.

[11] R. Fikes, R. Engelmore, A. Farquhar, W. Pratt "Network–based Information Brokers", AAAI '95 Symposium on Information Gathering from Distributed, Heterogeneous Environments, Palo Alto CA, March 1995.

[12] T. R. Gruber "Ontolingua: A Mechanism to Support Portable Ontologies", Stanford Univ., CS dept., techn. report KSL 91–66, March 1992.

[13] IDS GmbH *ARIS-Toolset Manual*, Version 1.0, IDS Prof. Scheer GmbH, Saarbrücken, 1994.

[14] ISO/IEC "Information Technology – Information Resource Dictionary System (IRDS) – Standard ISO/IEC 10027, Technical Report, ISO/IEC International Standard, 1990.

[15] T. Janning "Integration von Sprachen und Werkzeugen zum Requirements Engineering und Programmieren im Grossen ", Dissertation, RWTH Aachen, 1992 (in German).

[16] M. Jarke, M.A. Jeusfeld, P. Szczurko "Three Aspects of Intelligent Cooperation in the Quality Cycle", Intl. J. Cooperative Information Systems, vol. 2, no. 4, 355–374, 1993.

[17] M. Jarke, R. Gallersdörfer, M. A. Jeusfeld, M. Staudt, S. Eherer "ConceptBase - A Deductive Object Base for Meta Data Management", Journal of Intelligent Information Systems, vol. 4, no. 2, 167–192, 1995.

[18] M. A. Jeusfeld, M. Jarke "Enterprise Integration by Market-Driven Schema Evolution", Intl. Journal Concurrent Engineering Research and Applications (CERA), vol. 4, no. 3, 207–218, September 1996.

[19] M. A. Jeusfeld, U. A. Johnen "An Executable Meta Model for Re-Engineering of Database Schemas", Intl. J. Cooperative Information Systems, vol. 4, no. 2&3, 237–258, 1995.

[20] K. Kosanke "Process Oriented Presentation of Modelling Methodologies", In P. Bernus, L. Nemes (eds.): *Modelling and Methodologies for Enterprise Integration*, Chapman & Hall, London, 45–55, 1996.

[21] D. Kuokka, L. Harad "Supporting Information Retrieval via Matchmaking", AAAI '95 Symposium on Information Gathering from Distributed, Heterogeneous Environments, Palo Alto CA, March 1995.

[22] S. Kelly, K. Lyytinen, M. Rossi "MetaEdit+: A Fully Configurable Multi-User and Multi-Tool CASE and CAME Environment", Proc. 8th Intl. Conf. Advanced Information Systems Engineering, Heraklion, Greece, 1–21, 1996.

[23] R. Medina-Mora, T. Winograd, R. Flores, C.F. Flores "The Action Workflow Approach to Worlflow Management Technology", 4th Intl. Conf. on Computer-Supported Cooperative Work, 281–288, Toronto, Canada, 1992.

[24] G. Miller "WordNet: A Lexical Database for English", Communications of the ACM, vol. 38, no. 11, 1995.

[25] S. Milliner, M. Papazoglou "Architectural Support for Scalable Information Elicitation in Open Database Networks", Technical Report, IS-TR-7, QUT, 1996.

[26] J. Mylopoulos, A. Borgida, M. Jarke, M. Koubarakis "Telos - A Language for Representing Knowledge about Information Systems", ACM Trans. Information Systems, vol. 8, no. 4, 325–362, 1990.

[27] H.W. Nissen, M.A. Jeusfeld, M. Jarke, G.V. Zemanek, H. Huber "Managing Multiple Requirements Perspectives with Metamodels", IEEE Software, vol. 13, no. 2, 37–48, March 1996.

[28] T. Oates, N. Nagendra, V. Lesser "Cooperative Information Gathering: A Distributed Problem Solving Approach", UMass CS tech,. report 94–66, version 2.

[29] M.T. Özsu, P. Valduriez *Principles of Distributed Database Systems*, Prentice Hall, Englewood Cliffs, NJ, USA, 1991.

[30] Object Management Group *The Common Object Request Broker: Architecture and Specification (CORBA)*, Object Management Group Publications, Framingham, MA, USA, 1994.

[31] Open Systems Foundation *Introduction to OSF–DCE*, Revision 1.0, Prentice Hall, Englewood Cliffs, New Jersey, USA, 1992.

[32] ORACLE Deutschland GmbH *Oracle im Überblick*, Technical Report, Munich, Germany, 1996.

[33] M.P. Papazoglou, N. Russell, D. Edmonds "A Semantic-Oriented Translation Protocol for Heterogeneous Federated Database Systems", Technical Report, Queensland University of Technology, Brisbane, Australia, 1995.

[34] M.P. Papazoglou "Unraveling the Semantics of Conceptual Schemas", Communications of the ACM, 38, 9, 1995.

[35] P. Peters, M. Mandelbaum, M. Jarke "Simulating the Impact of Information Flows on Networked Organizations", Proc. 17th Intl. Conf. Information Systems, Cleveland, Ohio, 1996.

[36] G. Salton and M. J. McGill. "Introduction to Modern Information Retrieval", McGraw Hill Computer Science Series, New York, 1983.

[37] P. Simpson "Query Processing in a Heterogeneous Retrieval Network", 11th International Conference on Research & Development in Information Retrieval, ACM SIGIR, Grenoble, June 1992.

[38] G. Wiederhold, M. Genesereth "The Basis for Mediation", Proc. 3rd Intl. Conf. Cooperative Information Systems, Vienna, Austria, 140–155, 1990.

[39] E. Yourdan *Modern Strcutured Analysis*, Prentice Hall, Englewood Cliffs, New Jersey, USA, 1989.

7 Glossary

database schema logical structure of a database defined in a given data model prescribing the syntactical representation of objects in the database

data model collection of data structures and axioms for defining a database schema

generic concept natural language name for an abstract or physical entity in some domain of discourse

meta model collection of common concepts (canonical model) into which the data structures of different data models can be mapped

meta database database containing information about other databases including their schema

meta schema schema of a meta database