# CHAPTER 5.1

# Federating Neuroscience Databases

**Wen-Hsiang Kevin Liao and Dennis McLeod**
*University of Southern California Brain Project*
*University of Southern California, Los Angeles, California*

## Abstract

There are three key aspects of sharing and interconnection in a federated database environment: information discovery, semantic heterogeneity resolution, and system-level interconnection. Although the focus here is on the system-level interconnection process of the three key aspects described above, we also provide brief discussions on the other two aspects and summarize our approaches to those two issues. Our approach to support information sharing among databases is based on the import/export paradigm. In this paradigm, a component database of a federation decides the portion of its database to be exported and offers the methods to others on how the information can be shared. Users of other component databases who are interested in using the information can import it using one of the sharing methods provided by that component database. Information sharing is thus reached based upon the agreements (contracts) between each pair of component databases in which sharing is needed. A set of sharing primitives/tools is designed to support these sharing activities in a uniform manner. The sharing primitives/tools let users focus on how their information could be shared efficiently rather than on the underlying implementation. This enables users of a component database to utilize not only the information in their own component database but also the information in other component databases of the same federation.

## 5.1.1 Introduction

With the rapid growth of computer communication networks over the last decade, a vast amount of diverse information has become available on these networks. Users often find it desirable to share and exchange information that resides either within the same organization or beyond organizational boundaries. A federated database system is a collection of autonomous, heterogeneous, and cooperating component database systems. Component database systems unite into a loosely coupled federation in order to achieve a common goal: to share and exchange information by cooperating with other components in the federation without compromising the autonomy of each component database system. Such environments are increasingly common in various application domains, including office information systems, computer-integrated manufacturing, scientific databases, etc. Fig. 1 shows a generic view of a typical federated database system. Each component database participates in the federation through the sharing network. Component databases could reside within the intranet of an organization or belong to several organizations which are interconnected through the internet. The sharing network provides services to individual components for exchanging information across database boundaries.

There are three key aspects of sharing and interconnection in a federated database environment. These may

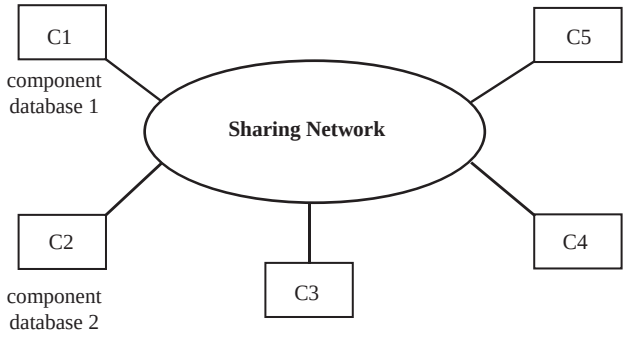


**Figure 1** Generic view of a federated database system.

be viewed in the context of a given component database (C), whose user intends to import information from and/or export information to other component databases.

1. *Information discovery:* The information discovery process pertains to finding out what information can be shared in the first place. From the viewpoint of a user of component database C, the main concern is to discover and identify the location and content of relevant, non-local information units in the federation.
2. *Semantic heterogeneity resolution:* Various kinds of diversity may exist in the autonomous component databases in a federation. The similarities and differences between the information of component C and relevant non-local information need to be resolved so they can be integrated and shared.
3. *System-level interconnection:* Based upon existing networking technologies, the mechanisms and their implementation must support actual sharing and transmission of information to and from C and other components in the federation.

The focus of this chapter is on the system-level interconnection process of the three key aspects described above. In addition, we also provide brief discussions on the other two aspects and summarize our approaches to those two issues. The remainder of the chapter is organized as follows. Section 5.1.2 briefly describes the dynamic classificational ontology approach to information discovery problem. Section 5.1.3 outlines the spectrum of semantic heterogeneity and introduces the approach of meta-data implantation and stepwise evolution. Section 5.1.4 discusses the issues related to system-level interconnection. Section 5.1.5 describes the characteristics of sharing patterns. Section 5.1.6 lays out the system architecture of our approach to providing sharing primitives/tools. We conclude the chapter with a sharing scenario using neuroscience databases in Section 5.1.7.

## 5.1.2 Information Discovery

The information discovery process involves finding relevant information units from the vast amount of data located in the component databases of a federation. Information discovery in a federated database environment faces a number of challenging problems due to its distinct characteristics: the volume of shared information is large, the unit of information is substantially structured, and the number of participants is large. A scalable and practical mechanism for information discovery should accommodate these characteristics to facilitate information sharing in the federated database environment. A common approach to mediating information discovery in such an environment is to adopt a common ontology (a collection of concepts and their relationships that are used for describing information units) as the basis for mutual understanding among participants. However, the approach does not scale well and it is difficult to build and maintain a common ontology as the number of participants becomes large. The concept and mechanism of dynamic classificational ontology (DCO; see Kahng and McLeod, 1996, 1998) was proposed to address the problems of common ontology and to illustrate how DCO can facilitate information-sharing activities.

The DCO approach is based on the observations that it is practical to minimize the size and complexity of the ontology in order to keep the system scalable, it is extremely difficult to reach a total agreement on an ontology when the number of participants is more than a handful, and it is beneficial to allow the ontology to change dynamically as the system evolves. In order to reduce the size of the common ontology, a DCO contains a small amount of high-level meta-knowledge on information exported from information providers. Specifically, it contains a collection of concepts and their relationships to be used for classification of exported information. The approach relies on classification because it is an effective scheme for organizing a large amount of information.

A DCO consists of a base ontology and a derived ontology. The base ontology contains an application-specific collection of concepts and their relationships that are agreed upon among participants and used by them for high-level description and classification of shared information units. The derived ontology contains information on additional relationships among some concepts in the base ontology, which are derived from the base ontology and the collection of exported information. Such relationships typically involve inter-related concepts that are useful for describing and classifying other information (for example, relationships among the collection of research subjects). The advantage of this approach is that participants are not required to agree on those relationships in advance; rather, those relationships are allowed to change as the usage of involved concepts changes. At the cost of information providers' cooperative efforts, this approach supports effective information sharing in the federated database environment.

### 5.1.3 Semantic Heterogeneity Resolution

The semantic heterogeneity resolution process resolves the similarities and difference of information in a component database and the information being imported from other components in the federation. Semantic heterogeneity in federated database systems is primarily caused by design autonomy of component database systems. Component database systems may employ different design principles for modeling the same or related data, thus resulting in semantic heterogeneity. The heterogeneity in a federation may be at various levels of abstraction. Components may use different database model in modeling data. Even if they use the same database model, they may come up with different conceptual schemas for the same or related data. Even object representation and low-level data formats differ from component to component employing the same database model. Finally, they may use different tools to manage and provide an interface for the same or related data.

The approach of meta-data implantation and stepwise evolution (Aslan and McLeod, 1999) to resolve semantic heterogeneity within the federated database environment involves a partial database-integration scheme in which remote and local (meta-)data are integrated in a stepwise manner over time. The meta-data implantation and stepwise evolution techniques can be used to inter-relate database elements in different databases and to resolve conflicts on the structure and semantics of database elements (classes, attributes, and individual instances). The approach employs a semantically rich canonical data model and an incremental integration and semantic heterogeneity resolution scheme. Relationships between local and remote information units are determined whenever enough knowledge about their semantics is acquired. The folding problem—folding remote meta-data (conceptual schema) into the local meta-data to create a common platform through which information sharing and exchange becomes possible—is solved by implanting remote database elements into the local database, a process that imports remote database elements into the local database environment, hypothesizes the relevance of local and remote classes, and customizes the organization of remote meta-data.

The advantages of this approach include the fact that it requires minimum global knowledge and effort from federation users both before and during interoperation. Global structures which should be maintained in the federation are minimum. Inter-relationships between schema elements in different components are highly dynamic. It recognizes the fact that knowledge required to relate inter-database elements may not be available, derivable from within the federation, or obtainable from users prior to interoperation.

### 5.1.4 System-Level Interconnection

The system-level interconnection process supports the actual sharing and transmission of information units among component databases in a federation. Considering the federation environment described above, each component database may be controlled by different database management systems and reside in different parts of the network. The sharing and transmission of information among component databases are not directly supported by most existing database management systems, and new mechanisms and tools to support the exchange of information are needed. In our approach, we adopt the import/export paradigm to facilitate information sharing among component databases. In this paradigm, the user of a component database decides the portion of the database to be exported and offers the methods to others as to how the information can be shared. Any user of a component database who is interested in using the information exported by a particular component database can then pick a portion of exported information and choose one of the sharing methods provided by that component database. Information sharing is thus achieved based upon the agreements (contracts) reached between each pair of component databases requiring sharing. A set of sharing primitives/tools is designed to support these sharing activities in a uniform manner. The sharing primitives/tools allow users to focus on how their information could be shared efficiently rather than on the underlying implementation. This enables users of component databases to utilize not only the information in their own component databases but also the information in other component databases of the same federation.

#### Import/Export Based Sharing Paradigm

Our approach to supporting information sharing between databases is based on the import/export paradigm (Fig. 2). The users of a database can acquire information units from other databases and use them to answer queries as if the imported information units are part of their own database. The goal is to allow the incorporation of objects in remote databases into the user's local database. In so doing, remote objects should appear transparent to users of the local database. This means that the user utilizes the remote data the same way she/he manipulates the local data.

The import/export approach is distinguished from most distributed query processing approaches, which send subqueries to be executed on remote environments and present the combined results back to its users, in that queries are executed under the user's local environment on the information units carried over the network from various sources. In our approach, almost all acquired information units are located locally in a consistent

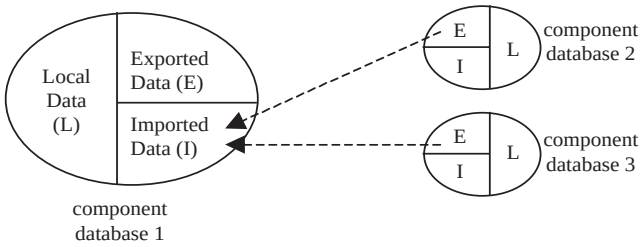


**Figure 2** Import/export-based sharing paradigm.

organization and thus allow users to treat both its own data and imported data in a uniform way.

Using the import/export paradigm, we can demonstrate how personnel information from several component databases can be used to answer a query. As shown in Fig. 3, BMW, Atlas, and Berger's lab are three autonomous databases of USCBP and each of them maintains personnel information of its group. The staff table in the local database contains information on those staff who work for USCBP but do not belong to any of the three groups. Under the import/export paradigm, the personnel information of each group can be imported using various sharing mechanisms. These imported personnel information and the local staff table can then be combined as a single USCBP personnel table which contains information on all USCBP personnel. It is then possible to efficiently answer a query such as "show me the names and phone numbers of all USCBP personnel."

### 5.1.5 Characteristics of Sharing Patterns

Several key characteristics of shared data may affect the performance of the actual sharing between two databases within our federation framework (Alonso and Barbara 1989). The number of imported information units and the size of each imported information unit determine the additional local space needed to hold all these imported information units. The frequency of updating data at the source may affect how often the changes have to be propagated to its importers. To determine the best way of information sharing is nontrivial. We believe that the information exporters know best the characteristics of data and performance and they can recommend sharing methods that are most suitable for information importers and can be afforded by the exporters (owners of the data) (Fig. 4). The importers would select the data of interest and then select one of the recommended sharing methods that would work best for the applications with the lowest overhead (Fig. 5).

Several sharing mechanisms have been designed to facilitate the retrieval of data and increase the availability of data. These mechanisms include direct link, copy, caching, time-bounded copy, and remote query execution. A set of primitives/tools has been designed to support all these sharing methods within our database federation framework. These sharing methods can be summarized as follows:

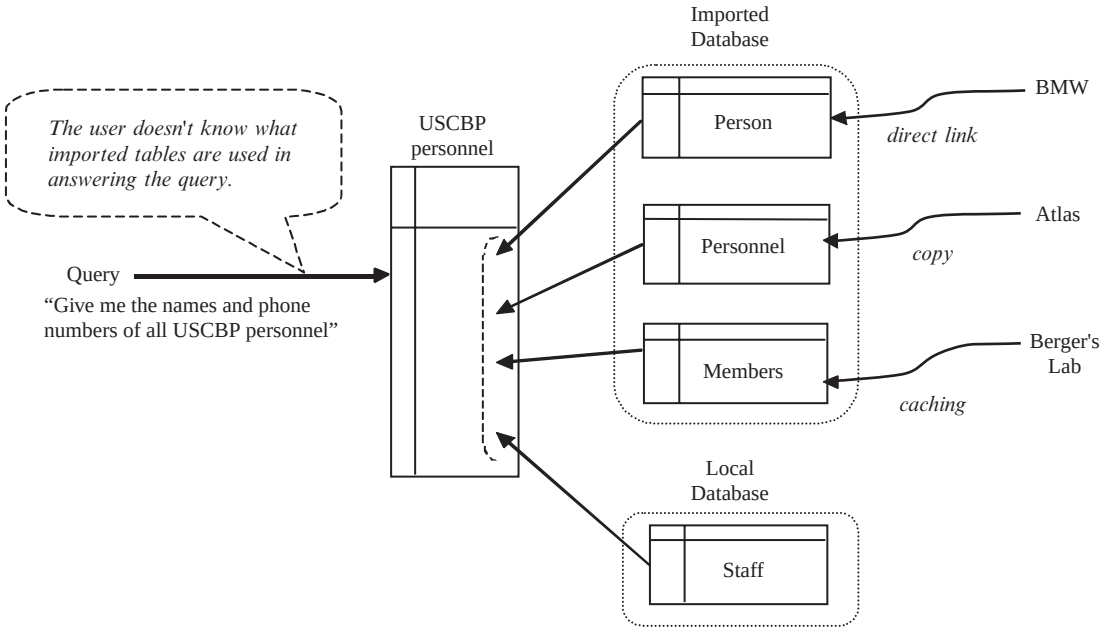


**Figure 3** Information sharing example using import/export paradigm.

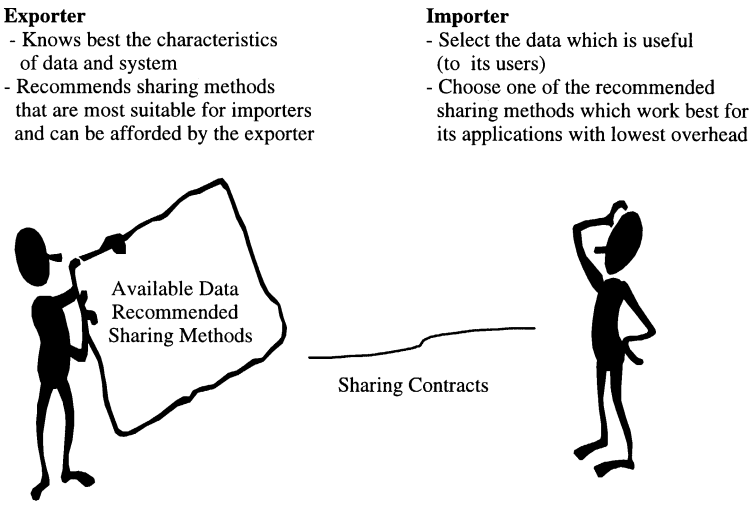


**Figure 4** Sharing scenario.

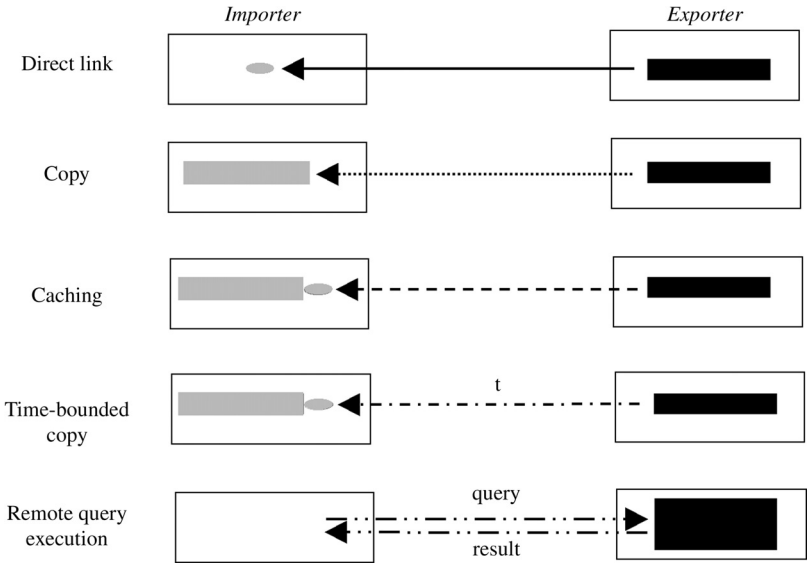


**Figure 5** Sharing methods.

1. *Direct link:* The direct-link method imports only the references to the sources of imported information, and each subsequent access will result in fetching data from their source databases. It is most suitable for frequently changed source data as it guarantees that the most current data are retrieved. It is expensive in that data must be retrieved each time it is accessed and data might not be available due to network problems or poor system performance on the exporter side.

2. *Copy:* The copy method duplicates all the data of interest on the importer database. While it has the best performance because copies of data are stored locally at the importer side and there is no need to retrieve data through a network, the imported data might become stale due to missed updates of data on the exporter database. Larger space is also required locally to store copied data. The copy mechanism is most suitable for importing those data which are rarely changed once they are published.

3. *Caching:* The caching sharing mechanism is a compromise between the direct-link and copy mechanisms. The importers keep both references to data sources as well as local copies, a procedure that provides better access performance than direct-link as the importers can use local copies directly if they are still up to date. It is also possible to use local copies without checking whether they are still current, if the network connection is not available. However, it also introduces additional overhead to the exporters as they are responsible for propagating updates to the importers' databases. The caching sharing method is most useful in sharing data that are updated irregularly.

4. *Time-bounded copy:* The time-bounded copy sharing mechanism is a special case of the caching method. It targets those data that are updated periodically. Local copies will be refreshed after a specified period of time based on the characteristics of their data sources. While it has the same benefits as those of the caching method, it also relieves some of the burden of the data exporters as the importers are responsible for retrieving data for every specified time period, but the time-bounded sharing mechanism is only suitable for those data that are updated regularly.

5. *Remote-query execution:* The remote-query execution method is designed to address those occasions when the set of data to be shared is so enormous that it is not practical to even import the references of data from an exporter's database. In such circumstances, it is more practical to execute a query request on the exporter's side rather than bring all the data to the importer's side.

### 5.1.6 System Architecture for Sharing Primitives/Tools

The implementation of sharing primitives/tools is based on the architecture as shown in Fig. 6. Each database component contains a set of data to be exported and a set of data imported from others. The import agent and export agent are responsible for negotiating the sharing agreement and transmission of data among database components. Sharing contracts, agreements between an exporter and importer on a set of data, are stored in each database as auxiliary data. Other types of data include:

1. *Local data:* Local data are all the information owned by and under the management of a component database.

2. *Exported data:* Exported data are that portion of local data intended to be shared with other component databases. Each exported table is associated with a list of available sharing methods recommended by the component database.

3. *Imported data:* Imported data contain information imported from other component databases.

4. *Imported contracts:* Imported contracts keep track of information on imported objects, including the provider, sharing method, and other parameters.

5. *Exported contracts:* Exported contracts keep track of information on exported objects, including the consumer, sharing method, and other parameters.

6. *Data export agent:* The Java-based export agent handles all the communication between the local database and remote databases. It is responsible for serving the requests for information sharing from other databases. These requests include building sharing contracts, transporting exported data, and propagating update on exported data using the caching method.

7. *Data import agent:* The Java-based import agent communicates with export agents of those databases from which the data are imported. It makes initial requests for sharing remote information and retrieves the data set once a sharing agreement is reached. It is also responsible for receiving updated data from remote databases and propagating the data to local copies.

There are also two sets of functions that allow users to specify which part of a database should be exported and from which database the data should be imported:

1. *Database exporting functions:* These functions provide users the capability to export part of their database and specify its supporting sharing mechanisms. They also allow users to browse all exported information

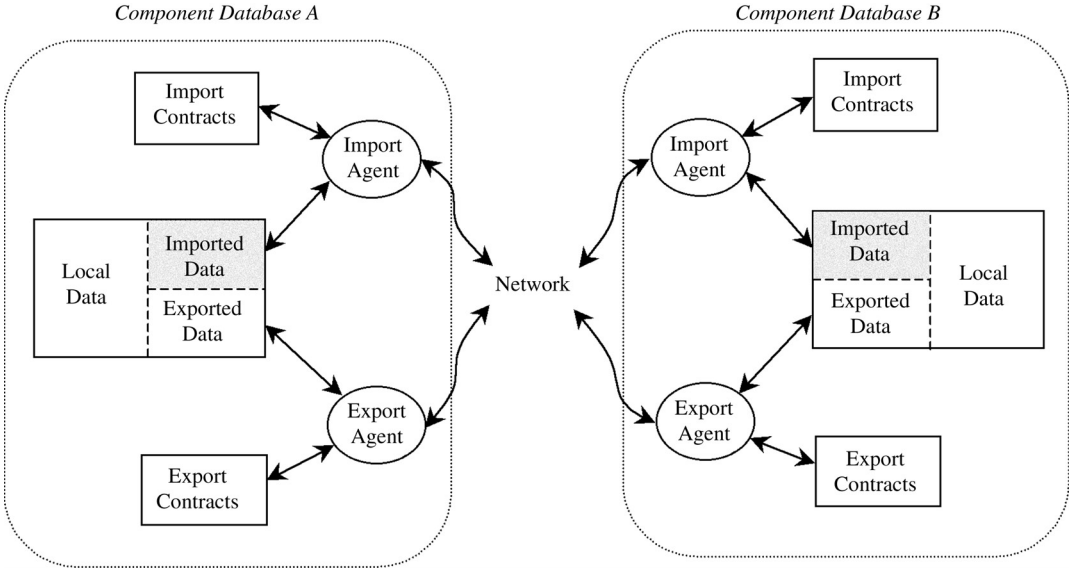


**Figure 6** System architecture.

and modify sharing mechanisms as the sharing environment evolves with time.

2. *Database importing functions:* End users use these functions to acquire information on data being exported from other databases. After the set of relevant data is selected, these functions transform the request and direct the import agent to reach agreement with remote databases. An inventory of all imported data and their sharing mechanisms is also maintained in the user's local database environment.

## 5.1.7 Federating Neuroscience Databases

The sharing primitives/tools and ontological conversions (conversions from one concept/inter-relationship/terminology framework to another) can support inter-database referencing at various levels. Developing and deploying such techniques will allow referencing database subsets across database boundaries. It is essential to construct ontologies to describe the key concepts, inter-relationships, and terminology for databases based upon NeuroCore (experimental data), NeuARt (atlas-based data), and NeuroScholar (connectivity data), as well as models stored in BMW (Brain Models on the Web) and data summarized in NSW (Neural Summaries on the Web). The sharing primitives/tools can then be used to support labeled interconnection between "information units" in the various databases. It will thus be possible to link, for example, an experiment with the protocol it employs with atlas structures that it involves. Fig. 7 shows the "USCBP federation information flow", the architecture of our approach to neuroscience information sharing. Here, we see direct interconnection among structured databases—those managed by NeuroCore or which are NeuroCore compatible. For example, experimental data units in the Thompson lab may refer to experimental data units in the Baudry lab. Given the rich meta-data in structured databases, interconnections can be established between database items and item collections—employing derived data specification techniques (e.g., SQL3 views). Sharing of atlas-based data, modeling, and other information is done via the use of specific interconnection by hand or via an ontology. Here, we employ a formal framework for specific kinds of inter-relationships (e.g., between experimental protocols and models). The role of ontologies in interconnection involves the mapping of concepts ("nouns," such as experimental protocols) and interrelationship labels ("verbs," such as supports) from one database to another. Our approach is to devise, implement, and deploy a sharing mechanism based upon these techniques. Our work draws extensively upon the results we have obtained to date at USCBP (Aslan and McLeod, 1999; Hammer and McLeod, 1999; Kahng and McLeod, 1998).

The sharing primitives/tools can be used further to devise, implement, and deploy a personal neuroscience information manager to allow a researcher to maintain a personal "view" of the USCBP federation, in particular tracking relevant subsets of information and their interconnections. Such personal neuroscience databases may contain references to other databases—specifically, subsets of those databases, items in them, etc. It will also allow "local" annotation of data, so that an investigator may bring together, inter-relate, and document working data. This system will also allow researchers to add their own "local" data or interconnections, information that is "private" to the researcher. Fig. 8 demonstrates the functional use of this personal neuroscience information

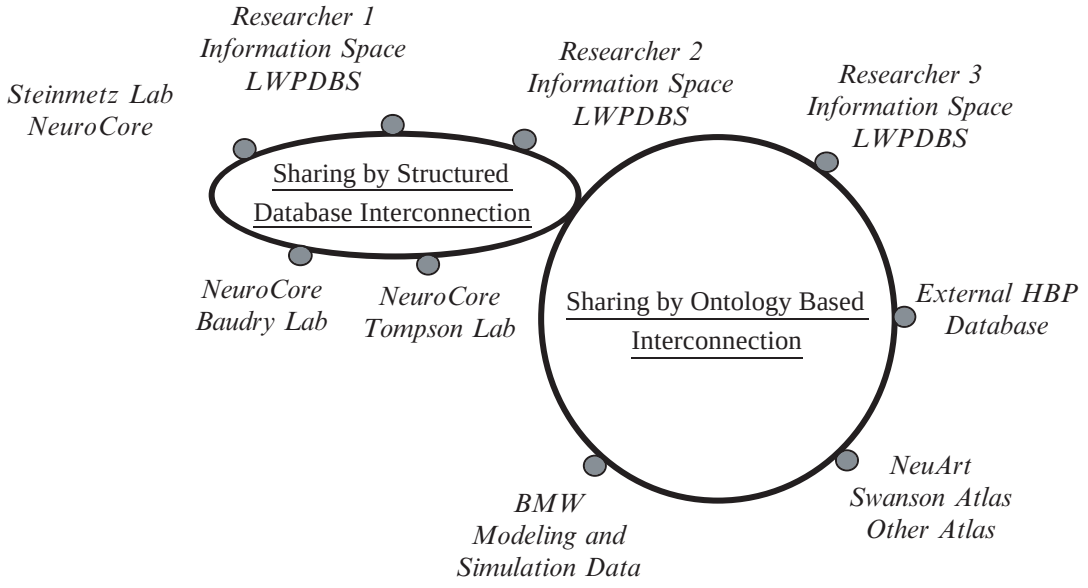


**Figure 7** USCBP federation information flow.

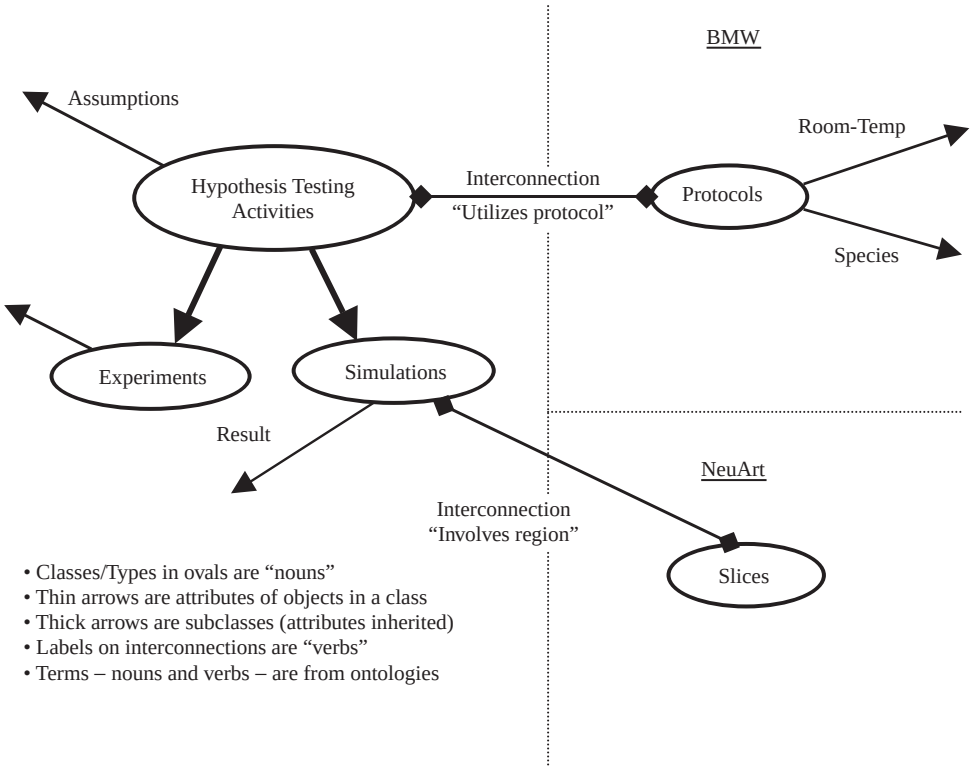


**Figure 8** Illustration of LWPDBS.

manager,—which may also be referred to as a lightweight (running on a personal machine, non-expert useable) personal database system (LWPDBS). The figure shows a simple graphical view of a portion of a researcher's information space. Here, on the left side, we see some local information classes/types, such as Experiments and Simulations, along with their superclass—Hypothesis Testing Activities. Each class represents a kind of information unit (object) and specifies the attributes that apply to members of those classes (e.g., the attribute Assumptions of Hypothesis Testing Activities). On the right side, we see two remote classes of information (from two other databases), which are interconnected to members of "local" classes. Note that the connections are labeled with a term (verb) from an ontology that covers this area of neuroscience. This LWPDBS will allow a researcher to maintain a "guide" to other databases and to locally maintain inter-connections and annotations of the researcher. In effect, the researcher is provided with a customized view of a subset of the neuroscience database federation, along with researcher-specific augmentations. Key challenges in realizing the LWPDBS and the interconnection mechanism underlying it include:

1. Developing an effective and efficient representation of interconnections
2. Managing the ownership and dynamics of interconnections
3. Allowing interconnections to be created, modified, traversed, and searched
4. Achieving an effective functional researcher interface for local data creation, manipulation, and export

Our approach will be to focus upon these key challenges, while adopting commercial product solutions to the underlying general database management issues. This is most consistent with our use to date of the object-relational database management system, Informix. Our plan is to use "small" versions of Informix or Oracle (both are object relational) to support the LWPDBS.

## References

Alonso, R. and Barbara, D. (1989). Negotiating data access in federated database systems. *ICDE*, 56–65.

Aslan, G. (1998). Semantic Heterogeneity Resolution in Federated Databases by Meta-Data Implantation and Stepwise Evolution, Ph.D. dissertation. University of Southern California, Los Angeles.

Aslan, G., and McLeod, D. (1999). Semantic heterogeneity resolution in federated databases by meta-data implantation and stepwise evolution. *Int. J. Very Large Databases*, in press.

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

Clifford, N. (1992). The Prospero file system, a global file system based on the virtual system model. *J. Comp. Syst*. **5(4)**, 407–432.

Elmasri, R., and Navathe, S. (1994). *Fundamentals of Database Systems*. Second ed. Benjamin/Cummings, Menlo Park, CA.

Fang, D., Ghandeharizadel, S., and McLeod, D. (1996). An experimental object-based sharing system for networked databases. *VLDB J*. **5**, 151–165.

Garcia-Molina, H. *et al*. (1995). Integrating and accessing heterogeneous information sources in TSIMMIS. In *Proceedings of AAAI. Symposium on Information Gathering*. Standford, CA, pp. 61–64.

Hammer, J., and McLeod, D. (1999). On the resolution of representational diversity in multidatabase systems. In *Management of Heterogeneous and Autonomous Database Systems* (Elmargamid, A., Rusinkiewicz, M., and Sheth, A., Eds.). Morgan Kaufman, San Mateo, CA.

Heimbigner, D., and McLeod, D. (1985). A federated architecture for information management. *ACM Trans. Office Inf. Syst*. **3(3)**, 253–278.

Hull, R., and Zhou, G. (1996). A framework for supporting data integration using the materialized and virtual approaches. In *Proceedings of the SIGMOD Conference*, Montreal, pp. 481–492.

Informix, Inc. (1997a). *Informix Guide to SQL*. Tutorial Version 9.1, Informix Press.

Informix, Inc. (1997b). *An Introduction to Informix Universal Server Extended Features*. Informix Press.

Kahng, J., and McLeod, D. (1998). Dynamic classificational ontologies, mediation of information sharing in cooperative federated database systems. In *Cooperative Information Systems*, Trends and Directions (Papazaglou, M., Ed.). Academic Press, New York.

Kahng, J., and McLeod, D. (1996). Dynamic classificational ontologies for discovery in cooperative federated databases. In *Proceedings of the First International Conference on Cooperative Information Systems*. Brussels, Belgium, pp. 26–36.

Kili, E. *et al*. (1995). *Experiences in Using CORBA for a Multidatabase Implementation*, DEXA Workshop 1995, London, pp. 223–230.

Litwin, W., Mark, L., and Roussopoulos, N. (1990). Interoperability of multiple autonomous databases. *ACM Computing Surv*. **22(3)**, 267–293.

Lyngbaek, P., and McLeod, D. (1990). A personal data manager. In *Proceedings of the International Conference on Very Large Databases*, pp. 14–25.

Sheth, A., and Larson, J. (1990). Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surv*. **22(3).**

Stonebraker, M. *et al*. (1996). Mariposa, a wide-area distributed database system. *VLDB J*. **5**, 48–63

Kim, W., Ed. (1995). *Modern Database systems*, The Object Model, Interoperability and Beyond. ACM Press, New York.

Kim, W. *et al*. (1993). On resolving schematic heterogeneity in multidatabase systems. *Distributed Parallel Databases* **1(3)**, 251–279.

Wiederhold, G. (1994). Interoperation, mediation, and ontologies. In *Proceedings of the International Symposium on Fifth-Generation Computer Systems*, December 1994. Tokyo, Japan, pp. 33–84.

Wiederhold, G. (1992). Mediators in the architecture of future information systems. *IEEE Computer* **25(3)**, 38–49.