The Computer Journal **40**(5) 1997

Impact of Semantic Heterogeneity on Federating Databases

Robert M. Colomb

Cooperative Research Centre for Distributed Systems Technology
School of Information Technology
The University of Queensland
Queensland 4072 Australia
colomb@it.uq.edu.au

ABSTRACT[1]

The difficult problems in design of systems which facilitate interoperation and mediation among information sources and their consumers arise from the presence of semantic heterogeneity among the schemas and ontologies supporting the different services. The purpose of this paper is to develop a taxonomy of semantic heterogeneity, and to describe, taking the perspective of text databases, the conditions under which autonomy-respecting interoperation of different kinds are likely to be feasible. The main conclusion is that interoperation can be based on structured database technology only if the participating organisations communicate among themselves, otherwise the considerations underlying text databases dominate the technology used.

Keywords: multidatabase systems, semantic heterogeneity

CR Category: H.2.5

26 September, 1996. Revised 27 August, 1997.

Introduction

The difficult problems in design of systems which facilitate interoperation and mediation among information sources and their consumers arise from the presence of semantic heterogeneity among the schemas and ontologies supporting the different services. The purpose of this paper is to develop an understanding of semantic heterogeneity, and to describe the conditions under which autonomy-respecting interoperation of different kinds is likely to be feasible.

To focus attention on the central problems, a maximal approach, *tight coupling* (Sheth and Larsen, 1990) is the reference architecture for this work. A tightly-coupled federated database is a collection of databases organised into a system which appears to the user as a single database, seamlessly integrating the component databases.

> A tightly coupled FDBS provides location, replication and distribution transparency. …a federation user can query using a classical query language against the federated schema with an illusion that he or she is accessing a single system.

Sheth and Larsen (1990) p. 205

The fundamental difficulty in achieving any sort of federation is semantic heterogeneity.

> Semantic heterogeneity occurs when there is a disagreement about the meaning, interpretation or intended use of the same or related data. … this problem is poorly understood, and there is not even an agreement regarding a clear definition of the problem.

[1] The work reported in this paper has been funded in part by the Cooperative Research Centres Program through the Department of the Prime Minister and Cabinet of the Commonwealth Government of Australia.

Sheth and Larsen (1990) p. 187

By interoperation is meant the ability to construct, using a query language equivalent to SQL, either a join or a union involving relations from two or more domains, with the possibility of using negation.

One of the desiderata described by Sheth and Larsen is the autonomy of the component databases. Their most comprehensive form of autonomy is *design autonomy*

> [which] refers to the ability of a component database system to choose its own design with respect to any matter, including the data being managed (i.e. the universe of discourse), the representation (data model, query language) and the naming of the data elements, and the conceptualization or semantic interpretation of the data.

Sheth and Larsen (1990) p. 187

Preservation of design autonomy in the strongest sense means that the component databases are federated without any change to their individual design. Compromising design autonomy means that the component databases must be changed. The feasibility of achieving federation depends partly on the amount of change needed, which translates to cost and time needed to establish the federation. In principle, a very substantial change in the component databases is possible - it consists of scrapping the old systems and building a new, more comprehensive system. Continuing the maxmial approach represented by tight coupling, the reference architecture for this work includes design autonomy on the principle that the less compromise to design autonomy, the more feasible the tight coupling is likely to be.

The main aim of this work is to elucidate the conditions under which a tightly coupled federated database system is likely to be feasible. The key issue is the concept of semantic heterogeneity, two basic types of which are discussed. The more difficult kind of semantic heterogeneity is designated *fundamental semantic heterogeneity*. This fundamental semantic heterogeneity is explored from the point of view of text databases. From this perspective fundamental semantic heterogeneity is seen to be the norm, so that it is the *absence* of semantic heterogeneity in single structured databases which is exceptional, and calls for explanation. Domain-specific database systems achieve homogeneity under certain conditions, which leads to some sufficient conditions for tight coupling. One of these conditions leads to a network of interoperations but not to a global schema.

Taxonomy of semantic heterogeneity

Much of the research into information systems interoperability has concentrated on resolution of what might be considered *structural* semantic heterogeneity among the databases supporting the component information systems (e.g. Kim and Seo, 1991; Fang and McLeod, 1992; Elmagarmid *et al*., 1993; Frankhauser and Neuhold, 1993; Hammer and McLeod, 1993; Merz and King, 1994; Reddy *et al*., 1994). Structural semantic heterogeneity occurs when the same information occurs in two different databases in structurally different but formally equivalent ways. It takes many forms: two database schemas may represent the same entity with different attribute names; the same instance may be identified differently; the same conceptual model can be represented by different database table structures; the same universe of discourse can be represented by different conceptual models; the same instances can be aggregated incommensurably in two systems; data in one system might be out of date or wrong; information represented in data in one system can be represented in the schema of another (Saltor *et al*., 1993).

Some workers, e.g. Lee and Malone (1990) have dealt with problems where two systems contain terms for which there are no equivalences, but the terms are instances of common supertypes. For example one system might have two types of organisational customer: large organisation and small organisation, while another might have two types of business customer: sales-tax paying and sales-tax exempt. No equivalences can be found between the most specific types in the two systems, but organisational customer can be identified with business customer. This partial structural integration permits some queries but not others.

A quite thorough study of interoperability with respect to data values is given by Sciore *et al.* (1994). It considers a wide variety of more-or-less resolvable heterogeneities, including those studied by Lee and Malone (1990), and it aim is to be able to extract as much semantic resolution as possible using conversion functions which may be incomplete or lossy. Its key construct is the recording of *context*, which is a property list containing descriptors supporting conversion functions (units, periodicity, granularity, etc.), and the association of a context with each attribute. Most of the content of the study is in the manipulation of contexts and their representation in relational systems.

The foregoing literature suggests that structural semantic heterogeneity can defined as heterogeneity which can be resolved essentially by a series of (possibly complex and subtle) view definitions which in principle respect the autonomy of the component information systems. Since Sciore *et al.* (1994) is so comprehensive, it will be taken as a surrogate for structural semantic heterogeneity, and will be the subject of further comment below.

Complex and subtle as structural semantic heterogeneity is, one often encounters semantic heterogeneity which cannot be resolved by view definitions, even in principle. This more general semantic heterogeneity will be referred to as *fundamental semantic heterogeneity*. Where it occurs between information systems, tight coupling cannot be achieved without changing at least one of the systems, thereby compromising design autonomy. Garcia-Solaco *et al.* (1996) present an extensive taxonomy of fundamental semantic heterogeneity, including methods for recognising it, tools for managing specifications during the resolution process, and also strategies for achieving resolution.

Two examples will serve to motivate the present work. The first occurs when two domains store data on the same objects, but share insufficient attributes for an object to be reliably identified as the same in both domains. This phenomenon has been studied by a number of authors, and has been given several different names, e.g.: *instance identification problem* by Wang and Madnick (1989), *entity identification problem* by Lim *et al.* (1993), and one type of *unresolvable semantic heterogeneity* by Colomb and Orlowska (1995). For example, a company database might identify its employees by surname and given name, while the taxation office might identify taxpayers by surname and date of birth. In addition, the company database might identify its employees by employee number, and the taxation office might identify taxpayers by a taxpayer identification number, but neither domain includes the identifier from the other. Joins or unions are therefore impossible. In practice, if interoperation must be achieved at least one of the databases must relinquish some of its autonomy and store additional information. In the example situation, the usual resolution of identification semantic heterogeneity is for the company to obtain taxpayer identification data from its employees and store it in its personnel database.

More generally, fundamental semantic heterogeneity occurs when terms in two different ontologies have similar meanings, but not quite the same. Further, neither database contains sufficient information to resolve the differences. The user for example might want to know the total spent on computer equipment in a given year from two different countries. The information needed is stored in databases in domains *A* and *B*. The semantics of domain *B* are what the user wants, but the semantics of domain *A* is expenditure on computer together with communications equipment. The attribute in *A* is more inclusive than in *B*. On the other hand, its domain might be less inclusive: it might be recorded only for organisations of at least a certain size. Fundamental semantic heterogeneity is manifested in very many ways. An excellent discussion of the issue in the field of simulation is given by Walter and Bellman (1990), and some extremely subtle but significant examples by Stamper (1985). Note that approaches like that of Lee and Malone (1990) essentially ignore the fundamental semantic heterogeneity which occurs below the common supertypes which they are able to resolve.

Once design autonomy is given up, a large range of engineering issues arises, many of which are not technical. For instance, the method of resolution suggested in the company/ taxation office example may be illegal under privacy legislation. Resolution, although technically fairly simple, would be in that case impossible in practice. In the import example described above, domain B might achieve its coverage as a by-product of its customs and import duty policies, while domain A must rely on surveys. To change either so that its data could be exactly related to the other would be prohibitively expensive, and almost none of the costs would be attributed to technical changes in the respective information systems.

The presence of fundamental semantic heterogeneity makes it impossible to achieve tight coupling without compromising design autonomy. The work of Garcia-Solaco *et al.* (1996), besides encompassing most of the approaches to structural heterogeneity described above, includes and in fact concentrates on methods which help the system designers to change the component databases. This concentration on overcoming fundamental semantic heterogeneity by change takes their work in a different direction from the present work, which seeks conditions under which fundamental semantic heterogeneity is likely to be minimal.

The remainder of this paper concentrates on the origins of the barriers raised to interoperability by fundamental semantic heterogeneity.

Fundamental semantic heterogeneity from a text database viewpoint

Tight coupling is an issue with databases of the kind generally used to support business information systems. Such databases are studied by the database community in texts such as Elmasri and Navathe (1994). At present most of these databases are implemented as relational systems using SQL data definition and manipulation language or a close relative, although many databases exist based on either the CODASYL or a Hierarchical data model.

To understand fundamental semantic heterogeneity, this section broadens the concept of database to include text databases as studied by the information storage and retrieval community in texts such as Salton (1989) and van Rijsbergen (1979). To distinguish the two classes of database, the former will be referred to as *structured databases*. Text databases store documents indexed by terms, while structured databases essentially store terms in structured tables. The relationship between these two classes is closer than might appear at first glance, as will be shown below.

The two classes of database are similar at least in that they both store terms and in that queries to each class are expressions involving terms. They differ most fundamentally in what the user expects as a result of a query.

A text database is a collection of texts in written natural language stored in a computer system. The function of the computer system is to enable a user to find a generally small subset of the body of texts which is relevant to some purpose. The mechanism of search requires the user to supply an additional text, called a query. The computer system is able to match the words in the query text with the words in the stored texts, and returns to the user a subset of the stored texts which matches most closely to the query, based on the presence or absence in the stored texts of the words contained in the query text.

Experience shows that the collection of texts returned generally contains some texts which are relevant to the user's purpose, and some which are not. Further, if the user had been able to examine personally each text in the stored collection, they would have discovered some texts relevant to their purpose which were not retrieved by the system. Two ratios, precision and recall, are used to measure the quality of retrieval. The former is the percentage of retrieved documents which are relevant to the user's need, while the latter is the percentage of documents relevant to the user's need which are retrieved by the query. Precision and recall values of 30-40% are considered good (Salton, 1989; van Rijsbergen , 1979).

This inexactness is considered to be a problem. The underlying assumption is that the criterion for retrieval is a match between the meaning of the stored text and the meaning behind the user's query. Further, the meaning of the texts are assumed to be determined by their words. A large number of sophisticated text processing algorithms have been proposed to reduce the inexactness of retrieval, but to very little avail (Salton, 1989; van Rijsbergen, 1979). That the content of a document and its relevance to a particular need are only loosely correlated with the collection of terms used in it is called by Krovets and Croft (1992) *lexical ambiguity*, which is essentially what the present paper refers to as fundamental semantic heterogeneity. Krovets and Croft argue that it is extremely difficult to eliminate.

A field of study called semiotics originated by de Saussure (1983) and Peirce (1932) is concerned with the study of signs, how they get their meanings, how they combine into texts, and how texts have meaning. (De Saussure used the name semiology, Peirce semiotics. The English-language literature seems to have settled on the name semiotics.) Some terminology from this field will be useful, so a brief sketch of the field is in order.

Benveniste (1981) argues that the meaning of a sign is determined by its position and relationships in a system of signs (called a semiotic system)- so is as much determined by other signs in the text and other signs which could potentially have been in the text but were not as by any specific denotation. This is what de Saussure (1983) calls a *system of difference*. In particular, the meaning of a given sign is not constant over time- it changes when new signs are added to, or old signs fall into disuse from, the semiotic systems to which the given sign belongs.

Since students of text databases use the word *term* to describe the unit from which texts are created, in the sequel "term" will be used where a semiotician would use the word "sign".

For example, consider the term "terminal". Before the mid-1980s, people used "terminals" to access computers. About 1985 the PC began to be used to access central computing resources. The PC used in this way and also more specialised new products did the same job as the old, but also had the capability to be used in new kinds of ways. The new products were designated "intelligent terminals", and the term "dumb terminal" introduced to designate the earlier products. (The connotations of "intelligent" and "dumb" were of course deliberate.)

After the establishment of the "intelligent" terminal, it tended to be given the more neutral designation of "workstation". However, there had been a huge investment in "dumb" terminals which persists to the present, and the term has also persisted. At the present time, a new type of terminal is being introduced, called a "network computer". The companies introducing the network computer are taking great pains to distinguish that product from a dumb terminal.

This example shows the evolution of a semiotic system: from A: {terminal} to B: {intelligent terminal, dumb terminal} to C: {workstation, dumb terminal} to D: {workstation, dumb terminal, network computer}. A query seeking documents about general methods of accessing computers which relied on the term "terminal" would work in a collection of documents using semiotic system A, and also in semiotic system B, since the new terms are created by qualifying the old. It would break down in collections where systems C or D is in use, since only a minority of documents might continue to use the now less specific term "terminal".

The structure of texts and how they get their meaning is part of semiotics. In particular, according to genre theorists such as Bakhtin (1986) and Freadman (1987), the meaning of a text is largely determined by its generic and specific context, and is only loosely based on the specific terms it is constructed from. A text occurs in a *genre*, which is an exchange of texts in a specific literature. Each new text refers to earlier ones, often responding to issues raised in the earlier texts or attempting to distinguish their content from earlier texts. A text generally anticipates future texts which may attempt to refute it from particular points of view. Genre theory is beginning to be applied to problems in computing. See for example Erickson (1997), who uses it to help understand computer-mediated social interaction among large groups of people.

Consider the introduction of the new term "network computer". In a genre of marketing literature and press releases the generic context includes the term "dumb terminal". Therefore the product descriptions and press releases often specifically mention the earlier term, and make the point that the new product is different. In a closely related genre of press commentary, a writer may disagree - arguing that the network computer is a dumb terminal under another name. At the same time, the research literature in say human-computer interaction may completely ignore the network computer, and the term "terminal" may designate a much wider range of devices which may exist only as research prototypes or even as unimplemented ideas. A paper in a conference where say EEG detectors and holographic workstations are described may easily have a different use of the term "terminal" than a press release from Microsoft. Genre contributes much to meaning.

A query to a text database system is also a text. The user must have some idea of the genres of texts held in the text database, and the semiotic systems characteristic of those genres. A query genre uses a collection of terms and the elements of the database's query language to get the text database's retrieval engine to return texts which are desired by the user. In practice, a user's interaction with the text database is a dialog, with subsequent query texts created in response to the document texts returned by previous queries. The referent of a term in a query is not in the outside world, it is how the presence of the term in the query affects the response of the database system. Queries are therefore a different genre from those to which the retrieved texts belong. The response to a query is judged by its relevance to the user's information needs, but a query is not solely an expression of those needs.

It is easy to construct examples of plausible queries using terms used in very different ways in different genres (e.g. communication, power, terminal). The text database returns texts containing the nominated terms. One user may find one returned text relevant and another returned text irrelevant. A second user might make exactly the reverse judgment. The problem of matching the meaning of documents to the meaning of a query based on the co-occurrence of terms in the query and stored texts is clearly underdetermined, so that high precision and recall should not be expected. The problem is worse the larger and more heterogeneous the collection of documents searched.

This is true even if a controlled indexing vocabulary is used. Information Storage and Retrieval Systems often use what is called *request-oriented indexing*, where the vocabulary takes account of the expected population of users. For example, the same document might be indexed differently for the pharmaceutical industry than for the medical community. Similarly, it might be indexed differently for a research community than for a population of high school students (Soergel, 1985). The indexers are also working in genres, and are making use of semiotic systems.

These issues are summarised by Eco (1990), who argues that the text of a document provides some constraints on its possible interpretations, in dialogue with researchers such as Rorty (1992), who argue that a given text can under particular circumstances be taken to mean anything.

In summary, the source of fundamental semantic heterogeneity is that the meaning of texts is only partly based on the words contained in them. Much of the meaning arises from the relationships among the terms used, the terms which could have been used but were not, the genres to which the texts belong, and the specific context of production of a particular text.

In the structured database world, the response computed by a database manager to a query is assumed to be all and only the information required by the user. Joins and negation are especially dependent on this assumption. A spurious or missing join attribute can have a very large effect on the number of tuples retrieved, and on the semantics of the answer computed. Negation is almost meaningless if the negated query can miss tuples or include spurious tuples.

The remarkable fact is that the structured database engineer does expect interoperation to be exact, - 100% precision and 100% recall - and that this expectation is founded on the broad pragmatic success of exact retrieval in single structured databases. Structured databases must be investigated more deeply to see what makes this exactness possible. The next section is devoted to this issue.

How structured databases achieve 100% precision and 100% recall.

Broadly speaking, structured databases are used for two purposes: first, to record the detailed activities of an organization (say the student records system in a University); and second, to record occurrences of instances by classificatory categories (say a census).

The detailed activities of an organization are essentially speech acts in the sense of Austin (1962). A brief sketch of speech act theory is needed at this point.

The situated production of a language object is an utterance of a *locution*. Some locutions make consequential changes in social reality. The quintessential such locution is "I pronounce you husband and wife" made by a marriage celebrant to a couple who fulfil a number of requirements, but there are a large number of others: naming something, making agreements, promises or threats, giving permission or prohibiting, etc. A locution of this sort is said to have *illocutionary force* from the point of view of the speaker, and *perlocutionary effect* from the point of view of everyone else. An utterance with illocutionary force and perlocutionary effect is called a speech act.

Using the language of genres used above in the discussion of text databases, speech acts are organised into a system of genres with sometimes very strict rules as to what can be done, by whom, and under what circumstances. These rules are called framing rules.

For example, a student is not enrolled in the University until their details are entered into the system of record. Further, the framing rules under which a person can be enrolled as a student are specified in detail and followed by the person making the record entry. This person must be in a specific position in the organization, and is often in a specific physical place (an enrolment clerk in the enrolments office during office hours, for example). Accounting systems are records of transactions under contracts, and are subject to audit.

Recording of occurrences of instances by categories is also a speech act. A census form has detailed instructions on how to fill it out, and the data entry person (a specific organisational role) has a detailed manual describing how to classify such things as employment category given a specific position title (framing rules). A person is not represented by a census record until the forms have been filled out and coded.

Further, the records of these speech acts are kept in a highly restricted language. A structured database has a metalanguage called a schema which determines what statements are possible. The schema includes constraints on the possible range of values of words in particular grammatical positions, so specifies a complete semiotic system. Only a strictly limited aspect of the organisational reality is recorded. This system works because speech acts recorded by organisations are generally highly stereotyped. Speech acts are a special sort of text recorded in a subset of natural language, so that a structured database is a special kind of text database.

Seeing a structured database as a kind of text database may seem strange to some, since structured databases usually contain non-text information, e.g. numbers, and their query models are very different. However, a structured database record might allow one to state "the part identified by 123456 is stored in warehouse bin identified by AA in quantity 231, and is purchased from the supplier identified by ZX443, whose name is Acme Manufacturing". Both this statement and the database record together with the database's conceptual model are representations of the same fact. (The structured database conceptual modelling technique called Object-Role Modelling, e.g. Halpin 1994, is based on verbalisation of this sort.) The verbalisation of a database record is a perfectly respectable document, and could plausibly be stored in an information storage and retrieval system.

Were the contents of a structured database represented as documents like this stored in a text database, the boolean query language expression "part and '123456' and stored" would retrieve the record verbalised in the previous paragraph. Further, if the record in its structured database representation were in a table named "stored" whose attribute "part" was its primary key, then only the document representing that record would be retrieved by the given query. The basic boolean query language used in text databases is a specialisation of the SQL SELECT statement. Seeing a structured database as a special kind of text database is formally correct, and its use here is expository.

Strict framing rules for speech acts and a restricted language for expressing them go some way to explaining the exactness of retrieval in structured databases, but are not wholly convincing. Why would one expect 100% exact retrieval?

Speech act theory suggests a third factor. Austin (1978, p120, footnote 2) observes that the declaration that two semiotic elements are equal is itself a speech act. A convention in the structured database world is that only the information recorded in the database exists. Suppose that Bob and Val have both enrolled in a subject RM201. Val is a full-time student who has taken the subject as an overload subject with a low priority, while Bob is a part time student taking only that subject out of a particular interest. The only information kept in the database is the record of the speech act making the enrolment. The users of the database ignore the information not recorded. This is the basis for the speech act declaring that Bob's enrolment in RM201 *is the same as* Val's except for the difference between the identifiers 'Bob' and 'Val'. This convention requires the previous two: strict framing rules and a restricted language. First, if the framing of a speech act is loose or loosely enforced, then the people affected by the consequences of the speech acts will not be willing to agree that the acts are the same and consequently the records of the acts are equal.

Furthermore, the strictly limited nature of the semiotic systems used for recording speech acts is important. The convention is that each genre of speech act has many instances, so that there will be many instances of the limited grammar statements recording them. This requires a limited range of lexical types, each with a limited range of possible lexical values. The issue of equality arises only because the same lexical types and lexical values occur many times, so that two different speech act records often share types and values.

Text databases do not have that underlying convention. In a database of newspaper articles, for example, the use of a particular word in one article is not assumed to be *the same as* the use of the same word in another. The use of the word "terminal" discussed in the previous section is an example.

In other words, a structured database can have 100% precision and recall because the language used within its user community is standardised. So is the set of things one can say with it, and so are the referents of all statements. The conceptual model of a database is a grammar of a specialised subset of a natural language. This severe limitation on the textual content and interpretation of the structured database makes it possible to use the much more powerful first order logic-based query languages, like SQL, instead of the simpler propositional or distance measure query languages characteristic of text databases.

This standardisation is brittle. When the organization changes, it can be difficult to communicate from past to present. The technical view of this situation is the well-known and very difficult problem of schema evolution.

Standardisation of the language is a necessary but not a sufficient condition for 100% precision and recall. Additionally, the person making the query must share the standardised language. This requirement places very strict limits on the population of people who can rely on these queries. They must have a deep familiarity with the organization, and perhaps also with a limited functional subgroup within the organization.

This is illustrated by the well-documented resistance of functional managers to requests by senior management for access to "raw" data via executive information systems. The fear that the senior manager will misinterpret the data is very real, since the senior manager does not share the detailed language of the functional subgroup. For example, one division of a large corporation may sell on a sale-or-return basis, while the other divisions sell on a final basis. Routine reports of revenue forecasts to headquarters are generally derived from total amount invoiced, but the sale-or-return division makes a correction for estimated returns. An executive information system report based on total amounts invoiced may therefore be misleading. In text database terms, the structured database query issued by a person not sharing the standard language of the owning organization has less than 100% precision and less than 100% recall, thereby invalidating the assumptions underlying SQL-style query languages.

From this perspective, the earlier classification of databases into text and structured is misleading. That distinction is based on storage and query technology, while the critical factor for 100% precision and recall is the existence of a standardised language community AND THE MEMBERSHIP OF THE PERSON MAKING THE QUERY IN THAT COMMUNITY. For the further discussion of interoperability, a more relevant distinction is:

- Within Standard Language Community (WSLC) query;
- Outside Standard Language Community (OSLC) query.

WSLC and OSLC are properties combining aspects both of the database and the person making the query. Recall that by language here is meant a specialised subset of natural language whose grammar is given by the explicit or implicit conceptual model of the database.

Almost by definition, database interoperability involves OSLC queries, regardless of whether the databases queried are structured or text. For this reason, one would not generally expect 100% precision and 100% recall from even an SQL query. This is not a structural problem. It is a problem with the degree of commonality between the person making the query and the people who designed and update the database of their understanding of the terms used, the semiotic systems associated with them, and the framing rules for their use. It cannot therefore be solved solely by structural transformations. In fact, if the history of text databases is any guide, the problem of fundamental semantic heterogeneity is unlikely to be solved by computing technology for a very long time, if at all.

Two projects which have been described in the literature will illustrate the issues involved. Both projects have been successful. The presentation here will emphasise how they have dealt with the issues of tight coupling and semantic heterogeneity.

The FAST project at the Information Sciences Institute of the University of Southern California (Neches, 1991; 1993) is an experimental electronic brokerage agency which assists computing hardware developers in purchase of components. Customers request components from FAST via email or Electronic Data Interchange (EDI). FAST has a database of information about potential suppliers (excluding parts catalogues). When a request is received, FAST semi-automatically contacts appropriate suppliers, either by EDI or e-mail. If a part is available, FAST selects a supplier and issues an order as its customers' agent. The supplier sees FAST as its customer, so that billing arrangements are between the supplier and FAST. FAST maintains its own arrangements with its customers.

The FAST system is not a standard language community, since there is no standard nomenclature for electronics parts, so that it is not tightly coupled. Interestingly, there in fact exists a standard, called Federal Supply Clauses, which is available only to U.S. Government purchases but not available to FAST.

The second project considered is a recently completed European Union-funded federated database called MIPS, which is focussed on sharing multi-media data (Austin *et al.* , 1994). The demonstrator application is travel, and the system is seen as a federation of travel industry information systems.

The system is conceptually highly centralised around a global conceptual model of the application domain constructed by the global system operator. Each participating information system must produce an export schema which relates their local database to the global conceptual model. (In order to join the federation, a system must be able to create views which can be related to the pre-existing global schema.)

A query is made on the global schema, and decomposed into queries on the participating databases by the global system. The global system retains the origin of any data seen by the user. In particular, the query manager records in a hypertext document the construction of each instance in the query, so that the user can trace back the construction of the result seen.

The global schema can contain integrity constraints, but is unable to enforce them outside the boundaries of the central system. In essence, a participating information system must agree to enforce integrity constraints in the global schema, but there is no way of checking whether they have done so.

The MIPS system is a standard language community, with the standard language given a priori by the global schema. Each participating service must resolve fundamental semantic heterogeneity before joining, and is responsible for constructing the views and rules to overcome structural semantic heterogeneity.

This second example achieves tight coupling by expanding the standard language community. This is done commonly in areas where a number of autonomous organisations must make standardised reports to an external body: for example company reports and taxation returns. In this case the standard language is arrived at by a combination of agreement among the accounting professional bodies and regulations by government agencies. Expansion of the standard language community is also the basis of EDI, in which autonomous organisations do business by exchanging electronic documents which are constructed in standard ways from standard vocabularies.

Note that one of the goals of the FAST project is to encourage the industry to develop such standards, both for nomenclature and for EDI. It does not, however, aim to be able to achieve tight coupling among parts catalogues. Its system is expected to be semi-automatic for the foreseeable future.

Once the standard language community has been expanded, then interoperability can proceed on a WSLC basis, with structural semantic heterogeneity resolved by view definitions and rules, held in appropriate components in the architecture. However, it is important to recognize that expansion of the standard language community almost certainly requires changes to the participating information systems.

Standardisation is not primarily a computing activity. First, the standard language derives from the application domain, and must be defined by agreement among members of that domain. Secondly, effective standardisation requires an ongoing process of audit, to ensure that the information exported by participating organisations accurately represents the agreed-upon meanings. Auditing company reports and taxation returns consumes very significant resources. Thirdly, the standards must be enforced, either through incentives or penalties. This is an extremely difficult area. The taxation office has a wide range of enforcement powers for making sure that taxation returns agree with standard definitions. Stock exchanges have a range of penalties for failing to adhere to reporting standards, including de-listing. Standardisation, audit and enforcement may employ information systems in themselves, but they are primarily human activities, not computer applications.

The argument of this paper so far has been that in general interoperation of databases is dominated by fundamental semantic heterogeneity. In general, therefore, autonomy-respecting tight coupling of federated databases is not achievable. Further, in general first-order query languages such as SQL are not unrestrictedly useable even in a loosely coupled environment.

Two databases are based on two different social realities. Not only are the semiotic systems by which the speech act recorded in each database different, but also the framing and the social systems maintaining the stability of framing supporting each system of record do not extend to the other.

The only remaining support for the expectation of interoperability between the two is the implicit convention that only the information recorded exists, which is the fundamental basis for the expectation of exact matching, and therefore for the definition of interoperability. However, part of the framing of the genre of speech acts declaring equality of semiotic elements is the sameness of framing support for the speech acts recorded by each system, in addition to the resolvable structural relationships between the two semiotic systems. If the two databases are supported by different framing conventions, then the speech acts declaring equality across databases cannot be framed, and therefore cannot be performed. The special factors distinguishing structured databases from text databases do not exist between databases. Interoperability would therefore not be expected.

In general, therefore, the technologies employed in autonomy-preserving interoperability would be dominated by considerations similar to those used in text databases. Limited precision and recall would be the norm, and the user's perception of relevance would have to be taken into account by exercise of human judgment at many stages in the construction and processing of a query.

Sciore *et al.* (1994)'s system for interoperability is based on a system of contexts, as is Garcia-Solaco *et al.* (1996). Contexts have been studied by many researchers, notably McCarthy (1993). It is generally accepted that it is neither possible nor even desirable to make all aspects of context explicit. The expedient generally adopted is a variation on use of systems of opaque context designators which guide interoperation among contextualised theories. The contexts are axiomatised only with respect to particular classes of problems, and always within a most general context. Context systems generally, and Sciore *et al.* (1994) in particular, are therefore WSLC systems. Outside the most general context, or for problems not considered in developing the context system, no formal reasoning can be done. For (a somewhat light-hearted) example, a system of interoperation set up to convert currencies may enounter a system based on exchange of goods and services made to maximise happiness among the participants at each exchange. Since it is well known that money can't buy happiness, each system would probably appear to be completely anarchic according to the semiotic rules of the other, and tight coupling would very likely be infeasible.

A tight coupling is feasible only if there is some special circumstance whereby fundamental semantic heterogeneity is not present. One such circumstance is discussed in the next section.

Limited tight coupling

Standard language communities are not generally completely self-contained. Most organisations interact with others in well-defined ways: commonly by buying and selling or more generally by exchange of contract, all of which are speech acts which are in the repertoire of both organisations, and which must have complementary framing rules. The language of these exchanges is therefore a common sublanguage of both standard languages, regulated by the law of contract and enforced by the civil courts. Many organisations achieve interoperation with selected partners using these shared sublanguages (e.g. Johnston and Vitale, 1988; Konsynski and McFarlan, 1990).

The information contained in these exchanges is often complex. Often, also, there are a large number of similar exchanges. The exchanges can be described in a standard language, so can have a schema. The aggregate of instances of exchanges can be thought of as a population of that schema. Since the messages are stored in both systems, this schema must be equivalent to a subschema in the information system of each organization, possibly requiring structural transformation. Systems of interoperability such as that of Sciore *et al.* (1993) can be used to formalise the transformations.

Common subschemas are not sufficient for interoperation. A natural join is null unless the two relations joined have common instances of the join attributes. In order to interoperate based on these exchanges, the population of the *exchange schema* must exist in both systems, at least transiently. The two organisations can agree on a policy for retention of these populations. The retained populations can now be seen as a collection of subtypes of entities and relationships in each information system.

Even though the overall schemas of the two information systems are otherwise disjoint, this common subschema with corresponding common subtypes can form the basis for a limited form of tight coupling, since natural joins can be reliably constructed. Note that due to the way this limited tight coupling is constructed, a network of organisations can have many disjoint limited tight couplings without any notion of a common global schema.

For example, a manufacturer might exchange orders with several suppliers, and also with several of its customers. It might also exchange messages with a shipping company which moves inventory from factories to warehouses. An order to a supplier might contain the manufacturer's identifier for the assembly to which the part ordered will contribute. If both the manufacturer and supplier retain the order for an extended period, the exchange population could form the basis for a query by the supplier on the manufacturer's planned production schedule for assemblies to which that supplier contributes parts, and by the manufacturer on the supplier's production schedules for parts which the manufacturer orders. Each party would supply to the other definitions of the schemas accessible via the common subtypes, so that the precision and recall of interorganisational queries would be increased. Each party would become a partial member of the other's standard language community.

Similarly, the manufacturer might be able to access a customer's inventory level for products it sells, the customer might be able to access the manufacturer's production schedules, and this relationship might exist between several manufacturers and several customers.

Moreover, although there is a relationship in the manufacturer's information system between the products it ships and the parts it buys, if these relationships are not made visible to its exchange partners by inclusion in the manufacturer's export schemas, customers and suppliers are insulated from each other.

On the other hand, the manufacturer's exchange with its shipping companies might be made visible to its customers, so that customers could get a better idea of expected delivery times or the effect of delays at ports. In this case, the manufacturer's understanding of the language used by the shipper is passed on to the manufacturer's customers. The necessity for a common population for a query, however, would restrict a customer to shipping data related only to products it orders. Also, the customer's understanding of the shipper's standard language is probably less than the manufacturer's, so that precision and recall would likely be less for each stage of distance from one information system to another.

Each participant therefore has a view of several other participants' information systems, but no participant has a view of the entire network of exchanges. The network of partial tight couplings includes many pairwise partial integrations of standard language communities, and even some pairwise second and higher order partial integrations, but is not necessarily a standard language community in itself.

This kind of common population can be used by several suppliers and several manufacturers without any possibility of overlap since the common subtypes are disjoint, even though the supertype schemas may be the same. A single supplier could construct a query which integrated information from several manufacturers, and a single manufacturer could construct a query which integrated information from several suppliers, but there is no global schema. Each party can be thought of as having an extended schema which imports views of the export schemas of its partners, integrated via the exchange schemas.

Also, each exchange of partial membership in standard language communities is distinct from all others. A wide-scale standard language community□is not a necessary requirement, although there might be a tendency for one to evolve, especially if the network were dominated, for example, by a single large manufacturer or retailer.

Conclusion

This paper has analysed semantic heterogeneity from the point of view of constructing a tightly coupled interoperating federation of information systems, with a main focus on fundamental semantic heterogeneity. Its main conclusion is that unless the organisations participating in a federation exchange messages among themselves, or alternatively all provide audited standard reports to a common centre, fundamental semantic heterogeneity is likely to be present and tight coupling extremely difficult to achieve. Organisations exchanging messages in their ordinary course of business can achieve a tight coupling, but it is likely to be limited to the content of the messages and to objects in the respective information systems functionally dependent on data in the messages.

These results have implications for less ambitious interoperation, where the information systems are more or less loosely coupled. Interoperation following interorganisational communication is likely to be feasibly based on structured database technology, since the main semantic heterogeneity encountered is likely to be structural. Interoperation not following interorganisational communication is likely to dominated by fundamental semantic heterogeneity. In these situations, the architecture must be based on considerations of precision and recall, and involve a high degree of participation by the customers in the construction and evaluation of responses combining data from several sources, and therefore must be based on considerations underlying text database technology.

Acknowledgments

This work has been greatly clarified by discussions with Nirad Sharma, Geraldine Fitzpatrick and Anne Freadman, who introduced me to semiotics.

References

Austin, J.L. (1962) *How to Do Things with Words* Cambridge, Mass, Harvard U.P.

Austin, J.L. (1978) "Truth" *Philosophical Papers*, Oxford University Press, pp 117-133.

Austin, W.J., Hutchinson, E.K., Kalmus, J.R., MacKinnon, L.M., Jeffery, K.G., Marwick, D.H., Williams, M.H. and Wilson, M.D. (1994). Processing Travel Queries in a Multimedia Information System. In W. Schertler, B. Schmid, A.M. Tyon and H. Werther (Eds.)*Information and Communication Technologies in Tourism*. Proceedings First ENTER Conference, Innsbruck, Austria. Berlin: Springer-Verlag.

Bakhtin, M. (1986) "The Problem of Speech Genres", *Speech Genres and Other Late Essays* Austin: University of Texas Press, pp 60-102.

Benveniste, E. (1981) "The Semiology of Language", *Semiotica* Special Supplement, 5-23.

Colomb, R.M. and Orlowska, M.E. (1995) "Interoperability in Information Systems" *Information Systems Journal* , 5(1) 37ff.

Eco, U. (1990)*The Limits of Interpretation*, Indiana University Press, Bloomington.

Elmagarmid, A.K., Chen, J. and Bukhres, O.A. (1993). Remote System Interfaces: an Approach to Overcoming the Heterogeneity Barrier and Retaining Local Autonomy in the Integration of Heterogeneous Systems *Int. J. of Intelligent and Cooperative Information Systems* 2(1) 1-22.

Elmasri, R. & Navathe, S.B. (1994) *Fundamentals of Database Systems, Second Edition* Benjamin Cummings.

Erickson, T. (1997) Social interaction on the Net: virtual community as participatory genre *Proceedings of the Thirtieth Annual Hawaii International Conference on System Science* Vol VI, R.H. Sprague, jr (ed) 13-21.

Fang, D. and McLeod, D. (1992). Seamless Interconnection in Federated Database Systems. In Y. Kambayashi (Ed.) *Database Systems for Next Generation Applications: Principles and Practice* ed Singapore: World Scientific.

Frankhauser, P. and Neuhold, E.J. (1993). Knowledge Based Integration of Heterogeneous Databases. In D.K. Hsiao, E.J. Neuhold and R. Sacks-Davis (Eds.)*Interoperable Database Systems (DS-5)* Amsterdam: North-Holland 150-170.

Freadman, A.(1987) "Anyone for Tennis?", in Ian Reid (ed.) *The Place of Genre in Learning: Current Debates* Typereader Publications, pp. 91-124.

Garcia-Colaco, M., Saltor, F. and Castellanos, M. (1996). Semantic Heterogeneity in Multidatabase Systems, in Bukhres, O. and Elmagarmid, A (eds) *Object-Oriented Multidatabase Systems* Prentice Hall, 129-202.

Halpin, T. A. (1994) *Conceptual Schema and Relational Database Design, 2nd edition* Prentice-Hall.

Hammer, J. and McLeod, D. (1993). An Approach to Resolving Semantic Heterogeneity in a Federation of Autonomous, Heterogeneous Database Systems, *Int. J. of Intelligent and Cooperative Information Systems* 2(1) 51-83.

Johnston, H.R. and Vitale, M.R. (1988). Creating Competitive Advantage With Interorganizational Information Systems, *MIS Quarterly*, 12(2) 153-165.

Kim, W. and Seo, J. (1991) Classifying schematic and data heterogeneity in multidatabase systems *Computer* 24 (12) 12-18.

Konsynski, B.R. and McFarlan, F.W. (1990). Information Partnerships- Shared Data, Shared Scale, *Harvard Business Review* 68(5) 114-120.

Krovets, R. and Croft, W.B. (1992) Lexical ambiguity and Information Retrieval *ACM TOIS* 10(2) 115-141.

Lee J. and Malone, T.W. (1990) Partially shared views: a scheme for communicating among groups that use different type hierarchies, *ACM Transactions on Information Systems*, 8(1) 1-26.

Lim, E., Srivastava, J., Prabhakar, S. and Richardson, J. (1993) Entity identification in database integration, IEEE International Conference on Data Engineering, 294-301.

McCarthy, J. (1993) A note on formalizing context. *Proceedings Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93)* Morgan Kaufmann, pp. 535-560.

Merz, U. and King, R. (1994) DIRECT: a query facility for multiple databases *ACM TOIS* 12(4) 339-359.

Neches, A-L (1991). *FAST Acquisition*, Final Technical Report, Information Sciences Institute, University of Southern California, USA, December 1991.

Neches, A-L (1993). *Government Application of FAST Technology, Tasks 1, 3 & 4* Semiannual Technical Report, Information Sciences Institute, University of Southern California, USA, April 1993.

Peirce, C.S. (1932) *Elements of Logic* Collected Papers, Volume 2, Harvard University Press, Cambridge MA.

Reddy, M.P., Prasad, B.E., Reddy, P.G. and Compton, A. (1994) A methodology for integration of heterogeneous databases *IEEE Transactions on Knowledge and Data Engineering* 6(6) 920-933.

Rorty, R. (1992) The pragmatist's progress, in S. Collini (ed.) *Interpretation and Overinterpretation* Cambridge University Press, Cambridge, 89-108.

Saltor, F, Castellanos, M.G. and Garcia-Solaco, M. (1993) "Overcoming Schematic Discrepancies in Interoperable Databases", in Hsaio, D., Neuhond, E. and Sacks-Davis, R. (eds) *Interoperable Database Systems (DS-5)(A-25)* Elsevier, pp. 191-205.

Salton, G. (1989). *Automatic Text Processing : The Transformation, Analysis, and Retrieval of Information by Computer* Reading MA: Addison-Wesley.

Saussure, F. de (1983) Course in General Linguistics, Duckworth : London.

Sciore, E., Siegel, M. and Rosenthal, A. (1994) Using Semantic Values to Facilitate Interoperability among Heterogeneous Information Systems *ACM TODS* Vol 19, No. 2, pp. 254-290.

Sheth, A.P. and Larsen, J. (1990). Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Databases, *ACM Computing Surveys: Special Issue on Heterogeneous Databases* 22(3) 183-236.

Soergel, D. (1985). *Organizing Information: Principles of Data Base and Retrieval Systems* New York: Academic Press.

Stamper, R. (1985). Management Epistemology: Garbage in Garbage Out. In L. R. Methlie and R. H. Sprague (Eds.) *Knowledge Representation for Decision Support Systems* Amsterdam: Elsevier Science Publishers B.V. 55-77.

van Rijsbergen, C.J. (1979). *Information Retrieval* 2nd ed. London: Butterworths.

Walter, D.O. and Bellman, K. (1990). Some Issues in Model Integration. In W. Webster and R. Uttamsingh (Eds,) *AI and Simulation* . Simulation Series Vol 22/3, San Diego: Society for Computer Simulation.

Wang, Y. and Madnick, S. (1989) The interdatabase instance identification problem in integrating autonomous systems, IEEE International Conference on Data Engineering, 46-55.