Semantic Heterogeneity in Global Information Systems: The Role of Metadata, Context and Ontologies

Vipul Kashyap[1,2] and Amit Sheth[1]
[1]LSDIS, Dept. of Computer Science, Univ. of Georgia, Athens, GA 30602
[2]Dept. of Computer Science, Rutgers University, New Brunswick, NJ 08903

June 12, 1996

Abstract

Semantic heterogeneity has been identified as one of the most important and toughest problems when dealing with interoperability and cooperation among multiple databases. It was earlier studied in the context of exchanging, sharing and integrating data, especially during the schema/view analysis phase of schema or view integration, or when writing a view or query using a multidatabase language. With the advent of global interconnectivity, we now need to deal with more heterogeneous information resources consisting of a variety of digital data, and the scale of the problem has changed from a few databases to millions of information resources, thus making it more important than ever to address this problem. It is also recognized that the problem has only become harder and that simplistic solutions involving only representational or structural components of data will not work beyond a very restricted set of cases.

In this chapter, we explore approaches to tackle the semantic heterogeneity problem in the context of Global Information Systems (GIS) which are systems geared to handle information requests on the Global Information Infrastructure (GII). These approaches are based on the capture and representation of metadata, contexts and ontologies. In order to handle *information overload*, it would be advantageous to abstract out the representational details of the underlying data and capture the information content by using *domain specific metadata*. The next important step is that of understanding the context of the query, using metadata to construct the context and identifying the relevant data in that context. Another critical issue that arises here is that of *different vocabularies* used to characterize similar information. We present an approach to deal with this problem at the metadata/context level by using terms from *domain specific ontologies* to construct metadata/context. We deal with semantic heterogeneity at this level and propose an approach using *terminological relationships* to achieve semantic interoperability.

1 Introduction

Many organizations face the challenge of interoperating among multiple independently developed database systems to perform critical functions. Three of the best known approaches to deal with multiple databases are tightly-coupled federation, loosely-coupled federation, and interdependent data management [SL90][She91]. A critical task in creating a tightly-coupled federation is that of schema integration (e.g., [DH84]). A critical task in accessing data in a loosely-coupled federation [LA86, HM85] is to define a view over multiple databases or to define a query using a multidatabase language. The problem of semantic heterogeneity, which is defined in [SK92] as identification of semantically related objects in different databases and the resolution of schematic differences among them, is a critical issue in any of the above three tasks.

However, with global interconnectivity we now need to deal with more heterogeneous information resources consisting of a variety of digital data. Huge amounts of digital data in a variety of structured (e.g. relational databases), semi-structured (e.g. e-mail messages) and unstructured

Appears as a chapter in *Cooperative Information Systems: Current Trends and Directions*, M. Papazoglou and G. Schlageter (editors)

(e.g. image data) formats have been collected and stored in thousands of autonomous repositories and CD-ROMs. Affordable multimedia systems allow creation of multimedia data and support access and presentation of such data. These digital repositories are increasingly being made available on the fast evolving GII of which the World Wide Web [BL+92] is an oft-cited and popular example. A GIS now has to deal with millions of information resources (as opposed to a few databases in a multidatabase federation), and simplistic solutions involving only representational or structural components of data will not work beyond a very restricted set of cases.

In this chapter we explore approaches that use *metadata, context* and *ontologies* to handle the semantic heterogeneity problem in a GIS. Two basic components of these approaches are (Figure 1):

- Use of metadata to capture the *information content* of the data in the underlying repositories. Intensional descriptions constructed from metadata and termed as *metadata contexts (m-contexts)* are used to abstract from the structure and organization of the individual repositories.
- Terms (concepts, roles) in domain specific ontologies are used to characterize contextual descriptions and are called *conceptual contexts (c-contexts).* Semantic interoperability is achieved by using terminological relationships between terms across ontologies.

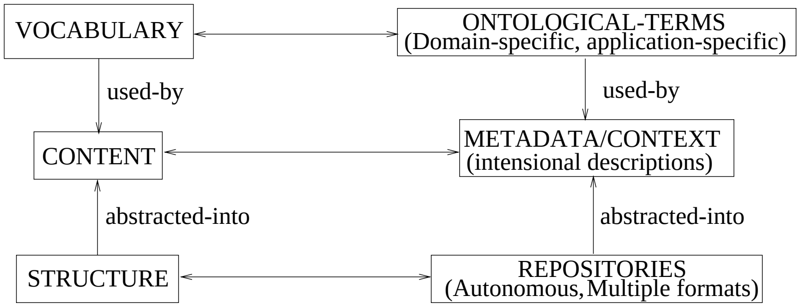


Figure 1: The basic components of our approaches

The key objective of such approaches should be to reduce the problem of knowing the contents and the structure of each of the huge number of information repositories to the significantly smaller problem of knowing the contents of the domain specific ontologies, which a user familiar with the domain is likely to know or easily understand. In this chapter we demonstrate the need for techniques which go beyond the structural and representational components of data and focus on the application of those techniques to structured databases.

Different types of metadata may be stored in the system (e.g., indices, schema information). The Rufus [SLS+93] and the InfoHarness [SSKS95] systems use automatically generated metadata to access and retrieve heterogeneous information independent of type, representation and location. In Section 2, we discuss the different kinds of metadata and present an informal classification. We identify and propose *domain specific metadata* as the key for solving the semantic heterogeneity problem [KSS95].

Section 3 discusses the construction of c-contexts from domain specific ontologies and their representation in a formalism that can be easily mapped [KS96] to a description logic (DL) expression. Issues of language and ontology involved in the above are also discussed. These c-contexts are used to represent *extra knowledge* about the information content of the database which may not be represented in the schema of the database. A user query can also be represented as a c-context. *Schema correspondences* [KS96] that capture the associations between c-contexts and the underlying data are also discussed.

The key to interoperability is *vocabulary sharing* among the intensional m-context and c-context descriptions associated with the various databases. Different concepts may be used to design contextual descriptions for different databases. We assume the existence of *application* and *domain specific ontologies* describing the information content of the various databases from which contextual expressions may be constructed. In fact, ontologies are viewed in our approach as a special case of *domain specific metadata.* In Section 4 we present an approach for semantic interoperability using terminological relationships across ontologies. We discuss the OBSERVER prototype [MKSI96] which demonstrates the use of *synonym* relationships to achieve semantic interoperability. Extensions of the above using *hyponym* and *hypernym* relationships [MKIS96] are also discussed. Section 5 presents future and ongoing work and our conclusions.

2 What is Metadata?

Figure 1 illustrates the two components of our approach for addressing the information overload problem in the GII. Metadata is the pivotal idea on which both the components depend. The function of the metadata descriptions is to be able to abstract out and capture the essential information in the underlying data *independent of representational details.* This represents the first step in reduction of the information overload as metadata descriptions are in general an order of magnitude less in size than the underlying data. In this section we discuss in detail our notion of metadata, the various types of metadata and the information they capture.

Metadata in its most general sense is defined as data or information about data. For structured databases, the most common example of metadata is the schema of the database. However with the proliferation of various types of multimedia data on the GII, we shall refer to an expanded notion of metadata of which the schema of structured databases is a (small) part. We use metadata to store derived properties of media useful in information access or retrieval. They may describe or be a *summary* of the information content of the data described in an intensional manner. They may also be used to represent properties of or relationships between individual objects of heterogeneous types and media. We now discuss a classification of the different types of metadata and characterize the amount of information content they capture. We also identify the types of metadata which will play a key role in enabling *semantic interoperability.*

2.1 A Classification of Metadata

We now present a classification of the various types of metadata used by various researchers to capture the information content represented in the various types of digital data (Table 1).

Content Independent Metadata This type of metadata captures information that does not depend on the content of the document with which it is associated. Examples of this type of metadata are *location, modification-date* of a document and *type-of-sensor* used to record a photographic image. There is no information content captured by these metadata but these might still be useful for retrieval of documents from their actual physical locations and for checking whether the information is upto date or not.

Content Dependent Metadata This type of metadata depends on the content of the document it is associated with. Examples of content dependent metadata are *size* of a document, *max-colors, number-of-rows, number-of-columns* of an image. We now present a categorization of content dependent metadata.

Direct Content-based Metadata This type of metadata is based directly on the contents of a document. A popular example of this is full-text indices based on the text of the documents. *Inverted tree* and *document vectors* are examples of this type of metadata.

Content-descriptive Metadata This type of metadata describes the contents of a document without direct utilization of the contents of the document. An example of this type of metadata is textual annotations describing the contents of an image. This type of metadata comes in two flavors:

Domain Independent Metadata These metadata capture information present in the document independent of the application or subject domain of the information. Examples of these are the *C/C++ parse trees* and *HTML/SGML document type definitions.*

Domain Specific Metadata Metadata of this type is described in a manner specific to the application or subject domain of information. Issues of vocabulary become very important in this case as the terms have to be chosen in a domain specific manner. Examples of such metadata are *relief, land-cover* from the GIS domain and *area, population* from the Census domain. In the case of structured data, the database schema is an example of such metadata. Another interesting example is *domain specific ontologies*, terms from which may be used as vocabulary to construct metadata specific to that domain.

Metadata	**Media Type**	**Metadata Type**
Q-Features [JH]	Image, Video	Domain Specific
R-Features [JH]	Image, Video	Domain Independent
R-Features [JH]	Image, Video	Content Independent
Impression Vector [KKH94]	Image	Content Descriptive
NDVI, Spatial Registration [AS]	Image	Domain Specific
Speech feature index [GSW]	Audio	Direct Content-based
Topic change indices [CHK^{+}]	Audio	Direct Content-based
Document Vectors [DDF^{+}90]	Text	Direct Content-based
Inverted Indices [KM91]	Text	Direct Content-based
Content Classification Metadata [BR]	MultiMedia	Domain Specific
Document Composition Metadata [BR]	MultiMedia	Domain Independent
Metadata Templates [OM93]	Media Independent	Domain Specific
Land-Cover, Relief [SK96]	Media Independent	Domain Specific
Parent-Child Relationships [SSKS95]	Text	Domain Independent
Contexts [SSR92, KS94a]	Structured Databases	Domain Specific
Concepts from Cyc [CHS91]	Structured Databases	Domain Specific
User's Data Attributes [SLS^{+}93]	Text, Structured Databases	Domain Specific
Domain Specific Ontologies [MKSI96]	Media-Independent	Domain Specific

Table 1: Metadata for Digital Media

2.2 Metadata: A means for capturing information content

In this section we discuss the information content captured by the various types of metadata enumerated in the previous section. We shall also identify the level (Figure 1) at which this metadata may be used.

Content Independent Information This type of information is captured by Content Independent metadata and helps in the encapsulation of information into units of interest and may be represented as objects in a data model.

Capturing Representational Information This type of information is typically captured by Content Dependent Metadata described in the previous section. This along with Domain Independent Metadata (which primarily captures structural organization of the data) enables interoperability via navigational and browsing approaches which depend on representational details of the data.

Capturing Information Content Information Content is typically captured to various degrees by various types of Content Dependent Metadata. Direct Content-based Metadata lies in a grey area in the sense that it is not entirely divorced from the representational details. However, the metadata which helps abstract out representational details and capture information meaningful to a particular application or subject domain is Domain Specific Metadata.

Vocabulary for Information Content Characterization Domain Specific Metadata can be constructed from terms in a domain specific ontology or concept libraries describing information in an application or subject domain. Thus we view Ontologies as Metadata which themselves can be viewed as a vocabulary of terms for construction of more domain specific metadata descriptions. Semantic interoperability at the vocabulary level is achieved with the help of terminological relationships.

The above discussion suggests that *domain specific metadata* capture information which is more meaningful *wrt* a specific application or a domain. The information captured by the other types of metadata primarily reflect the format and organization of the underlying data. This leads us to propose domain specific metadata as the most appropriate for dealing with issues related to semantic heterogeneity.

Constructing Intensional Descriptions from Domain Specific Metadata

Domain specific metadata can be used to construct intensional descriptions which capture the information content of the underlying data. We categorize these intensional descriptions as follows:

Metadata Contexts (m-contexts) These descriptions primarily serve to abstract the representational details in the underlying data and may be viewed as *boolean combinations* of the individual metadatum. These contexts are typically populated before hand by processing the underlying data. They may also be computed at run-time by using *parameterized routines.* Examples of this type of metadata and how they may be used to interoperate across multimedia data are illustrated in [SK96].

Conceptual Contexts (c-contexts) These descriptions primarily serve to capture *domain knowledge* and help impose a conceptual semantic view on the underlying data. C-contexts are constructed from terms (concepts, roles) in domain specific ontologies. The terms used in construction of the c-contexts might be interrelated to each other via relationships viz. terminological, domain/range constraints on roles.

In the rest of the chapter, we focus on the structured data and the use of c-contexts constructed from domain specific ontologies to capture the information content. The relationships between terms in the ontologies enable the representation of *extra knowledge* not represented in the database schema. We shall also discuss the cases where c-contexts may be constructed from different domain specific ontologies.

3 Constructing c-contexts from Ontological Terms

In Figure 1, we have identified metadata as the pivotal idea on which our approaches to address the information overload problem in the GII are based. In the previous section we discussed the various types of metadata and identified domain specific metadata as the most appropriate for handling semantic heterogeneity. One approach to construct metadata which capture meaningful information *wrt* an application domain is to use terms from domain specific ontologies as the vocabulary to characterize the information. We have identified such metadata descriptions as **c-contexts** in the previous section, and in this section we present a discussion of issues related to their representation and use.

We discuss the inadequacies of purely structural and mapping based methods in representing object relationships and discuss the advantages of representing c-contexts. We shall discuss a partial representation of c-contexts and equivalent description logic expressions. We shall also discuss operations for automatic ways of comparing and manipulating c-contexts and illustrate with the help of examples how they maybe used to achieve interoperation across information sources. A brief discussion of issues relating to the language for representing c-contexts and the ontologies from which the c-contexts may be constructed is also given. We shall refer to c-contexts as contexts unless otherwise specified in the rest of the chapter.

3.1 Rationale for Context representation

In characterizing the similarity between objects based on the semantics associated with them we have to consider the real world semantics (RWS) of an object. It is not possible to completely define what an object denotes or means in the model world [SG89]. We propose the **context** of an object as the primary vehicle to capture the RWS of the object. We argue for the need for representing context by showing the inadequacy of purely structural representations. We also discuss the computational benefits of representing context.

3.1.1 Inadequacy of purely Structural Representations

It has been suggested by Sheth and Gala/Kashyap [SG89][SK92] and Fankhauser et al. [FKN91] that the ability to represent the structure of an object does not help capture the real world semantics of the object. It is not possible to provide a structural and hence a mathematical definition of the complex notion of real world semantics. In [LNE89], a one-to-one mapping is assumed between the attribute definition and the attribute's real world semantics. They define an attribute in terms of fixed descriptors such as *Uniqueness, Lower/Upper Bound, Domain, Scale* etc. which are used to generate mappings between two attributes. They are also used to determine the equivalence of attributes. However what they establish is the structural equivalence of these attributes which is necessary, but not sufficient, to determine the semantic equivalence of the attributes.

Consider two attributes, *person-name* and *department-name*. We may be able to define a mapping between the value domains of these two attributes, but we know that they are not semantically equivalent. In order to be able to capture this lack of equivalence, we propose the mappings between the domains of the attributes be made *wrt* a context. We define two objects to be semantically equivalent if it is possible to define mappings *wrt* all known and coherent contexts, and the definition contexts of the objects should be coherent *wrt* each other. Definition contexts and the notion of coherence are discussed later in this section. Since the definition contexts of *person-name* and *department-name* are not coherent (one identifies an animate and the other identifies an inanimate object), they are not defined to be equivalent.

3.1.2 Computational benefits of representing context

Shoham [Sho91] has discussed the computational benefits that might accrue in modeling and representing context in AI and Knowledge-Based systems. We believe that some of those reasons are very relevant in the presence of information overload in the GII and suggest the identification and representation of context.

Economy of representation: In a manner akin to database views, contexts can act as a *focusing mechanism* when accessing the component databases on the GII. They can be a *semantic summary* of the information in a database or group of databases and maybe able to capture semantic information not expressed in the database schema(s). Thus unnecessary details can be abstracted from the user.

Economy of reasoning: Instead of reasoning with the information present in the database as a whole, reasoning can be performed with the context associated with a database or a group of databases. This approach has been used in [KS94a] for information resource discovery and query processing in Multidatabases.

Managing Inconsistent Information: In the GII, where databases are designed and developed independently, it is not uncommon to have information in one database inconsistent with information in another. As long as information is consistent within the context of the query of the user, inconsistency in information from different databases may be allowed. This has been discussed with the help of an example in [KS96].

Flexible semantics: An important consequence of associating abstractions/mappings with the context is that the same two objects can be related to each other differently in two different contexts. This is because two objects might be semantically closer to each other in one context as compared to the other.

3.2 A partial Context representation

There have been attempts to represent the similarity between objects in different databases. In the previous section, we showed with the help of an example how a fixed set of descriptors used in [LNE89] do not guarantee semantic similarity. Thus, any representation of context which can be described by a fixed set of descriptors is not appropriate.

The descriptors, called meta-attributes or contextual coordinates, are not fixed but are dynamically chosen to model the characteristics of the application domain in question. It is not possible apriori to determine all possible contextual coordinates which would completely characterize the semantics of the application domain. This leads to a *partial* representation of context as a collection of contextual coordinates:

Context = <(C$_1$, V$_1$) (C$_2$, V$_2$) ... (C$_k$, V$_k$) >

Table 2 shows how our context descriptions can be mapped to expressions in CLASSIC [BBMR89], a DL system. Using CLASSIC[1], it is possible to define primitive classes and in addition specify classes using intensional descriptions phrased in terms of necessary and sufficient properties that must be satisfied by their instances. The intensional descriptions may be used to express the collection of constraints that make up a context. Also, each C$_i$ roughly corresponds to a role and each V$_i$ roughly corresponds to fillers for the role the object must have. We shall also explain the meaning of the symbols C$_i$ and V$_i$ by using examples and by enumerating the corresponding CLASSIC expressions.

- C$_i$, $1 \leq$ i $\leq$ k, is a contextual coordinate denoting an aspect of a context.
- C$_i$ may model some characteristic of the subject domain and may be obtained from a domain specific ontology (discussed later in this section).
- C$_i$ may model an implicit assumption in the design of a database.
- C$_i$ may or may not be associated with an attribute A$_j$ of an object O in the database.

Contextual coordinates and Values	**CLASSIC descriptions**
<(C$_1$, V$_1$) ... (C$_k$, V$_k$)>	(**AND** O (**ALL** C$_1$ V$_1$) ... (**ALL** C$_k$, V$_k$))
<(C$_i$, O$_i$$\circ$ <(C$_j$, V$_j$)>)>	(**AND** O (**ALL** C$_i$ (**AND** O$_j$ (**ALL** C$_j$ V$_j$))))
<(C$_i$, X) (C$_j$, X)>	[C$_i$] for (**SAME-AS** C$_i$ C$_j$)
<(C$_i$, X$\circ$ <(C$_j$, V$_j$)>)>	[C$_i$] for (**FILLS** C$_i$ (**ALL** C$_j$ V$_j$))

Table 2: Contextual coordinate, value pairs and the corresponding CLASSIC expressions

The value V$_i$ of a contextual coordinate C$_i$ can be represented in the following manner:

- V$_i$ can be a variable.
  - It can be unified (in the sense of Prolog) with another variable, a set of symbols, an object or type defined in the database or another variable.
  - It can be unified with another variable associated with a context.
  - It can be used as a place holder to elicit answers from the databases and impose constraints on them.

[1] We have proposed a minor addition [<role-set>] for <classic-expression> to CLASSIC expressions [MKSI96] to enable retrieval of object properties.

Example:
Suppose we are interested in people who are authors and who hold a post. We can represent the query context C_q (discussed later in this section) as follows:
C_q = <(author, X) (designee, X)>
The same thing can be expressed in a Description Logic (DL) as follows:
C_q = [author] for (**SAME-AS** author designee)
The terms *author* and *designee* may be roles chosen from a domain specific ontology.

- V_i can be a set.
  - The set may be an enumeration of symbols from a domain specific ontology.
  - The set may be defined as the extension of an object or as elements from the domain of a type defined in the database.
  - The set may be defined by posing constraints on pre-existing sets.

Example:
Suppose we want to represent the assumptions implicit in the design of the object EMPLOYEE in a database. We can represent this as the definition context of EMPLOYEE, C_{def}(EMPLOYEE) as follows:
C_{def}(EMPLOYEE) = <(employer, [**Deptypes** ∪ {restypes}])(article, PUBLICATION)>
The same thing can be expressed in a DL as follows:
C_{def}(EMPLOYEE) = (**AND** EMPLOYEE (**ALL** article PUBLICATION)
(**ALL** employer **Deptypes** ∪ {restypes}))

Deptypes is a type defined in the database. The symbols *restypes, employer* and *article* may be chosen from a domain specific ontology. The symbols *employer* and *article* may be related to attributes associated with the underlying database objects. The symbol *restypes* acts as a role filler and may be mapped to a data value in the database. The definition context expresses an association between the objects EMPLOYEE and PUBLICATION which may not be captured in the database schema.

- V_i can be a variable associated with a context.
  - This can be used to express constraints which the result of a query should obey and is called the constraint context.
  - The constraints would apply to the set, type or object the variable X would unify with.

Example:
Suppose we want all articles whose titles contain the substring "abortion" in them. This can be expressed in the following query context:
C_q = <(article, X∘ <(title, {y|substring(y) = "abortion"})>)>
where ∘ denotes *association* of a context (<(title, {y|substring(y) = "abortion"})>) with a variable (X) and ensures that the answer satisfies the constraints expressed in the context. The same thing can be expressed in a DL as follows:
C_q = [article] for (**ALL** title {y|substring(y) = "abortion"})

- V_i can be a set, type or an object associated with a context. This is called the association context and may be used to express semantic dependencies between objects which may not be modeled in the database schema.

Example:
Suppose we want to represent information relating publications to employees in a database. Let PUBLICATION and EMPLOYEE be objects in a database. The definition context of HAS-PUBLICATION can be defined as:
C_{def}(HAS-PUBLICATION) = <(article, PUBLICATION)
(author, EMPLOYEE∘ <(affiliation, {research})>)>

where ∘ denotes association of a context with an object (EMPLOYEE) and a context (<(affiliation, {research})>).

Association of a context with an object is similar to defining a view on the object extensions such that only those instances satisfying the constraints defined in the context are exported to the GIS. The symbols used as contextual coordinates, e.g., *article, author* and *affiliation* are obtained from a domain specific ontology and may be mapped to attributes of database objects. The relationships between the database objects EMPLOYEE, PUBLICATION and HAS-PUBLICATION captured in the contextual description are not modeled in the database schema. The same thing can be expressed in a DL as follows:

$$\begin{array}{ll} \mathrm{C}_{def}(\text{HAS-PUBLICATION}) = & (\textbf{AND}\ \text{HAS-PUBLICATION} \\ & \quad (\textbf{ALL}\ \text{article PUBLICATION}) \\ & \quad (\textbf{ALL}\ \text{author}\ (\textbf{AND}\ \text{EMPLOYEE} \\ & \qquad\qquad (\textbf{ALL}\ \text{affiliation}\ (\textbf{ONE-OF}\ \{\text{research}\}))))) \end{array}$$

3.3 Reasoning about and manipulation of contexts

We have proposed a partial representation of context in the previous section. This can be used to abstract out the information content of the underlying data and help reduce the information overload in the GII. The next step is to use these representations meaningfully to enable a GIS to focus on relevant information and to correlate information from the various information sources on the GII. In order to achieve this, the following need to be precisely defined [KS96]:

Specificity The most common relationship between contexts is the "specificity" relationship. Given two contexts C_1 and C_2, $C_1 \leq C_2$ iff C_1 is at least as specific as C_2. This is useful when objects defined in a particular context have to transcend [McC93] to a more specific or general context and is discussed in detail with examples in [KS96].

Organization in a Lattice Structure It is possible that two contexts may not be comparable to each other, i.e. it may not be possible to decide whether one is more specific than the other. Thus, the specificity relationship gives us a partial order. The following useful operations on the context lattice can be defined:

overlap(Cntxt$_1$, Cntxt$_2$) This is the common set of contextual attributes present in the contextual descriptions.

coherent(Cntxt$_1$, Cntxt$_2$) This operator determines whether the constraints determined by the values of the contextual coordinates are consistent.
Example:
Let Cntxt$_1$ = <(salary, {x| $x \leq 10000$})>
Cntxt$_2$ = <(salary, {x| $x > 10000$})>
Thus, coherent(Cntxt$_1$, Cntxt$_2$) = FALSE

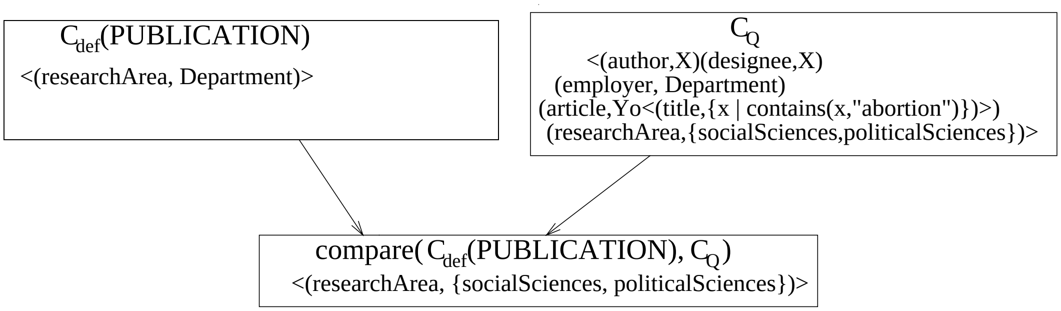
greatest lower bound (glb) of two contexts The contexts can be organized in a special kind of lattice structure called a *meet semi-lattice* in which every pair of contexts has a greatest lower bound. Intuitively the *glb* computes the conjunction of constraints expressed in the contextual descriptions.

Inferences using Contextual Descriptions

We now illustrate how reasoning with contextual descriptions can help enable semantic interoperability across different databases on the GII. The interoperability is achieved *wrt* the query which is represented as a context and known as the query context C_Q. The definition contexts of the various objects in the underlying databases enable the (partial) capture and representation of the information content in the databases. The query context is compared with the definition contexts and this can be easily implemented as a combination of the *glb* and *overlap* operations discussed above.

A critical assumption made in the examples illustrated below is that query and definition contexts are constructed from a common ontology. This is a very *un-scalable* assumption in the context of a GIS. One way of enhancing the scalability is to support the use of pre-existing and independently developed (often ad-hoc) domain ontologies. This requires mechanisms for comparing terms across ontologies at run-time, which is the subject of discussion of the next section. Issues of language to represent the contextual descriptions and ontologies are discussed later in this section.

Consider the comparison of the query context C_Q and the definition context C_{def}(PUBLICATION) illustrated in Figure 2.

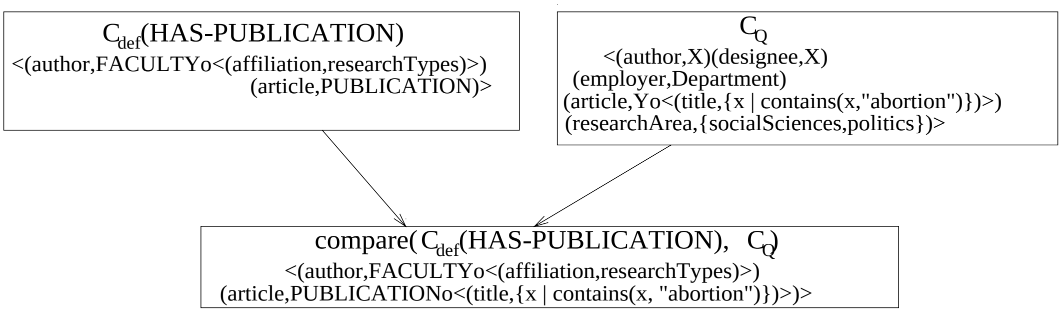


Cdef(PUBLICATION)
<(researchArea, Department)>
CQ
<(author,X)(designee,X)
(employer, Department)
(article,Yo<(title,{x | contains(x,"abortion")})>)
(researchArea,{socialSciences,politicalSciences})>
compare(Cdef(PUBLICATION), CQ)
<(researchArea, {socialSciences, politicalSciences})>

Figure 2: Comparison of contextual descriptions: Identifying the relevant publications

The instances of the PUBLICATION object identified as belonging to the research areas *socialSciences* and *politicalSciences* are determined to be relevant to the user query. This is an example of using contextual expressions for determining information relevant to a query.

In the next example, we illustrate how constraints in a query can be applied to information in a database to determine the relevant answers. Consider the query context C_Q and the definition context C_{def}(HAS-PUBLICATION) illustrated in Figure 3.



Cdef(HAS-PUBLICATION)
<(author,FACULTYo<(affiliation,researchTypes)>)
(article,PUBLICATION)>
CQ
<(author,X)(designee,X)
(employer,Department)
(article,Yo<(title,{x | contains(x,"abortion")})>)
(researchArea,{socialSciences,politics})>
compare(Cdef(HAS-PUBLICATION), CQ)
<(author,FACULTYo<(affiliation,researchTypes)>)
(article,PUBLICATIONo<(title,{x | contains(x, "abortion")})>)>

Figure 3: Comparison of contextual descriptions: Incorporating a constraint from the query

The constraint in the query, requiring the article titles to contain the word "abortion", is incorporated in the contextual descriptions describing the information content of the database and propagated to the object PUBLICATION. The modified contextual description thus characterizes only those instances of the object PUBLICATION which contain the word "abortion" in their titles.

Another interesting use of contextual descriptions is to rule out the possibility of a database having information relevant to a query. Suppose we are interested in all authors having a salary > \$200,000. Suppose all the faculty members in the university database are represented as having a salary $\leq$ \$150,000. Consider the following contextual descriptions.

C_{def}(FACULTY) = <(salary, {x| x $\leq$ \$150,000})>
C_Q = <(author, X) (salary, {x| x > \$200,000})>
compare(C_{def}(FACULTY),C_Q) $\Rightarrow$ inconsistent(x $\leq$ \$150,000, x > \$200,000)
$\Rightarrow$ The university database is not relevant for the query Q.

3.4 Mapping Contextual descriptions to the Database Schema

As discussed earlier, the contextual descriptions serve to abstract out the underlying representational details and capture the information content. However once the relevant high-level contextual descriptions have been identified, there is a need to retrieve the relevant data and display it to the user. In [KS96], we propose a uniform formalism used to map contextual descriptions to underlying data. Work on mapping intensional descriptions to SQL queries is reported in [BB93]. Collet et al. [CHS91] have used articulation axioms to relate object classes in databases to concepts in the Cyc ontology. Our approach is similar to the above but we have also defined an algebra in [KS95] to keep track of the changes in the mappings when the associated contextual descriptions change.

Each information system exports a global object O_G corresponding to the objects O it manages to the GIS. The objects O_G are obtained by applying the constraints in the definition context $C_{def}(O)$ to the object O. The user sees only the exported objects. The contextual coordinates C_i of the $C_{def}(O)$ act as the attributes of O_G. The exported objects O_G are associated with the objects and types defined in the database. This association might be implemented in different ways by various component systems. We use schema correspondences defined as follows to express these associations (Figure 4).

$\mathbf{schCor(O_G,O) = <O_G,\{C_i| \ C_i \in C_{def}(O)\},O,attr(O),M>}$

- O_G is the exported GIS object of an object O or type T defined in the database.
- The attributes of the object O_G are the contextual coordinates of the definition context $C_{def}(O)$.
- The mapping operation $\mathbf{map_O(C_i,A_i)}$ stores the association between contextual coordinate C_i and attribute A_i of object O whenever there exists one.
- The mapping M between O_G and O can be evaluated using the projection rules enumerated and illustrated in [KS96].

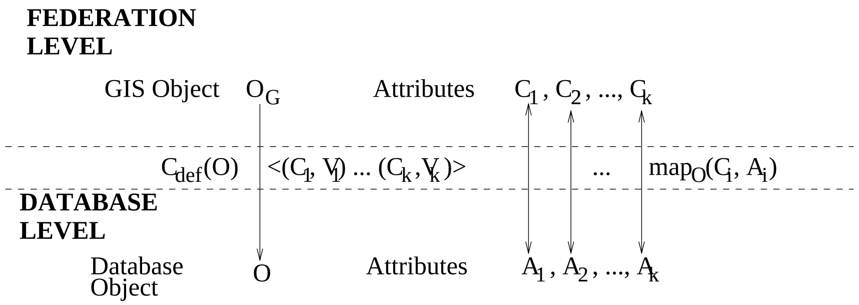


Figure 4: Schema Correspondences: Mapping contextual expressions to underlying data

We have discussed in [KS96] a set of *projection rules* which map a contextual expression to underlying database objects. We now discuss two examples which illustrate how *extra information* may be represented using contextual expressions.

3.4.1 Representing relationships between objects

We illustrate a case where the definition context of the object HAS-PUBLICATION captures its relationships with another database object EMPLOYEE in an intensional manner. These relationships are *not stored* in the database and mapping the contextual description results in *extra information* being associated with the GIS object HAS-PUBLICATION$_G$. A naive user will ordinarily not be aware of this relationship. The detailed mapping of this relationship has been illustrated in [KS96].

Example:
Consider objects EMPLOYEE and PUBLICATION defined earlier and an object HAS-PUBLICATION(SS#, Id) in the same database which represents a relationship between employees and the publications they write.
C_{def}(HAS-PUBLICATION) = <(author,EMPLOYEE∘ <(affiliation,{research})>)>
HAS-PUBLICATION$_G$ = Join((SS# = SS#), HAS-PUBLICATION,
Select(Affiliation ∈ {research}, EMPLOYEE))

This results in only those objects being exported to the GIS which satisfy the constraints specified in the contextual descriptions. The user thus does not have to keep track or know the relationships between the various objects in the database.

3.4.2 Using terminological relationships in Ontology to represent extra information

In this section, we illustrate an example in which terminological relationships obtained from an ontology are used to represent *extra information*. In the example illustrated below, the contextual coordinate *researchInfo* is a composition of two contextual coordinates (*researchArea* and *journalTitle*) and is obtained from the ontology of the domain. This is then used to correlate information between the objects PUBLICATION and JOURNAL. However, the contextual coordinate researchArea has not been modeled for the object PUBLICATION. Thus, this results in *extra information* about the relevant journals and research areas being associated with the object PUBLICATION, even though no information about research areas is modeled for PUBLICATION.
Example:
Consider a database containing the following objects:
PUBLICATION(Id, Title, Journal), where
C_{def}(PUBLICATION)
= <(researchInfo,JOURNAL∘ <(researchArea,Deptypes)(journalTitle,JournalTypes)>)>
JOURNAL(Title, Area), where C_{def}(JOURNAL) = <>

The mapping expression is given as follows (see [KS96] for details):
PUBLICATION$_G$ = Join((researchArea=Area)∧(Title=Journal), PUBLCATION,
Select((Area∈Deptypes)∧(Title∈JournalTypes),JOURNAL))

- Only journals belonging to the research areas corresponding to the departments are selected (Select((Area IN Deptypes) AND ... ,JOURNAL)).
- The join condition (Title = Journal) ensures that only those articles which are from the research areas corresponding to the departments are exported to the GIS (Join((researchArea=Area) AND (Title = Journal), ...)).
- This is achieved even though the attribute Area is not modeled for PUBLICATION. Thus there is extra information in terms of association of Deptypes with PUBLICATION through the join condition.

3.5 Issues of language and ontology in context representation

In this section we discuss the issues of a language in which the explicit context representation discussed in Section 3.2 can be best expressed. Besides, as discussed earlier, we use terms from domain specific ontologies as vocabulary to characterize domain specific information. We also discuss in this section issues of ontology, i.e. the vocabulary used by the language to represent the contexts.

3.5.1 Language for context representation

In Section 3.2 we have represented context as a collection of contextual coordinates and their values. The values themselves may have contexts associated with them. In this section, we enumerate the properties desired of a language to express the context representation.

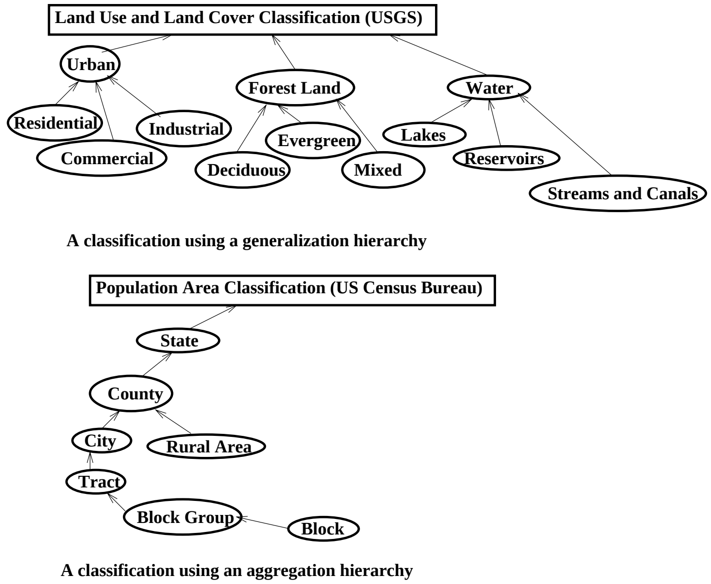
- The language should be declarative in nature as the context will typically be used to express constraints on objects in an intensional manner. Besides, the declarative nature of the language will make it easier to perform inferences on the context.
- The language should be able to express the context as a collection of contextual coordinates, each describing a specific aspect of information present in the database or requested by a query.
- The language should have primitives (for determining the subtype of two types, pattern matching, etc.) in the model world, which might be useful in comparing and manipulating context representations.
- The language should have primitives to perform navigation in the ontology to identify the abstractions related to the ontological objects in the query context or the definition contexts of objects in the databases.

3.5.2 The Ontology Problem

The choice of the contextual coordinates (C_is) and the values assigned to them (V_is) is very important in constructing the contexts. There should be *ontological commitments* that imply agreements about the ontological objects used between the users and the information system designers. In our case this corresponds to an agreement on the terms and the values used for the contextual coordinates by both a user in formulating the query context, and a database administrator for formulating the definition and association contexts. In the example in Section 3.2, we have defined C_{def}(EMPLOYEE) by making use of symbols like *employer, affiliation* and *reimbursement* from the ontology for contextual coordinates, and *research, teaching,* etc. for the values of the contextual coordinates.

We assume that each database has available to it an ontology corresponding to a specific domain. The definition and association contexts of the objects take their terms and values from this ontology. However in designing the definition contexts and the query context, the issues of combining the various ontologies arise. We now enumerate various approaches one might take in building ontologies for a GIS comprising of numerous information sources. Other than the ontological commitment, a critical issue in designing ontologies is the **scalability** of the ontology as more information sources enter the federation. Two approaches are discussed next.

- **The Common Ontology approach:**
  - One approach has been to build an extensive global ontology. A notable example of global ontology is Cyc [LG90] consisting of around 30,000 objects. In Cyc, the mapping between each individual information resource and global ontology is accomplished by a set of *articulation axioms* which are used to map the entities of an information resource to the concepts (such as frames and slots) in Cyc's existing ontology [CHS91].
  - Another approach has been to exploit the semantics of a single problem domain (e.g., transportation planning) [ACHK93]. The domain model is a declarative description of the objects and activities possible in the application domain as viewed by a typical user. The user formulates queries using terms from the application domain.
- **Re-use of Existing Ontologies/Classifications:** We expect that there will be numerous information systems participating in the GIS. In this context, it is unrealistic to expect any one existing ontology or classification to suffice. We believe that the re-use of various existing classifications such as ISBN classification for publications, botanical classification for plants is a very attractive alternative. An example of such a classification is illustrated in Figure 5. These ontologies can then be combined in different ways and made available to the GIS.
  - A critical issue in combining the various ontologies is determining the overlap between them. One possibility is to define the "intersection" and "mutual exclusion" points between the various ontologies [Wie94].



A classification using a generalization hierarchy

A classification using an aggregation hierarchy

Figure 5: Examples of Generalization and Aggregation hierarchies for Ontology construction

- Another approach has been adopted in [MS95]. The types determined to be similar by a sharing advisor are classified into a collection called *concept*. A *concept hierarchy* is thus generated based on the superconcept-subconcept relationship. These types may be from different databases and their similarity or dissimilarity is based on heuristics with user input as required.

4 Semantic Interoperability using Terminological Relationships

In Figure 1, we illustrated how terms from domain specific ontologies can be used as vocabularies to characterize domain specific information. This is an essential component of the approaches to enable tackling the semantic heterogeneity problem on the GII. In the previous section, we discussed how terms from an ontology may be used to construct contextual expressions and how terminological relationships result in the representation of extra information not represented in the database schema. However there was an implicit assumption of a common ontology behind the construction of the contextual expressions. As discussed earlier, this is a very *un-scalable* assumption. In this section we discuss the issues involved when contextual descriptions may be constructed from different domain specific ontologies. We discuss how *semantic interoperability* may be achieved by interoperation across these domain specific ontologies. We now discuss approaches to achieve interoperation across ontologies using terminological relationships like *synonyms, hyponyms* and *hypernyms.*

4.1 Using synonyms to interoperate across ontologies

In this section we propose an approach to interoperate across ontologies which have been expressed using a description logic system like CLASSIC [BBMR89]. We have illustrated how contextual expressions may be represented using description logic expressions. We now discuss our work in the OBSERVER[2] [MKSI96] system which enables interoperation across various independent pre-existing ontologies based on synonym relationships across terms in different ontologies.

[2] *Ontology Based System Enhanced with Relationships for Vocabulary hEterogeneity Resolution*

4.1.1 An architecture for interoperation

In this section we discuss an architecture for interoperation across domain specific ontologies (Figure 6).

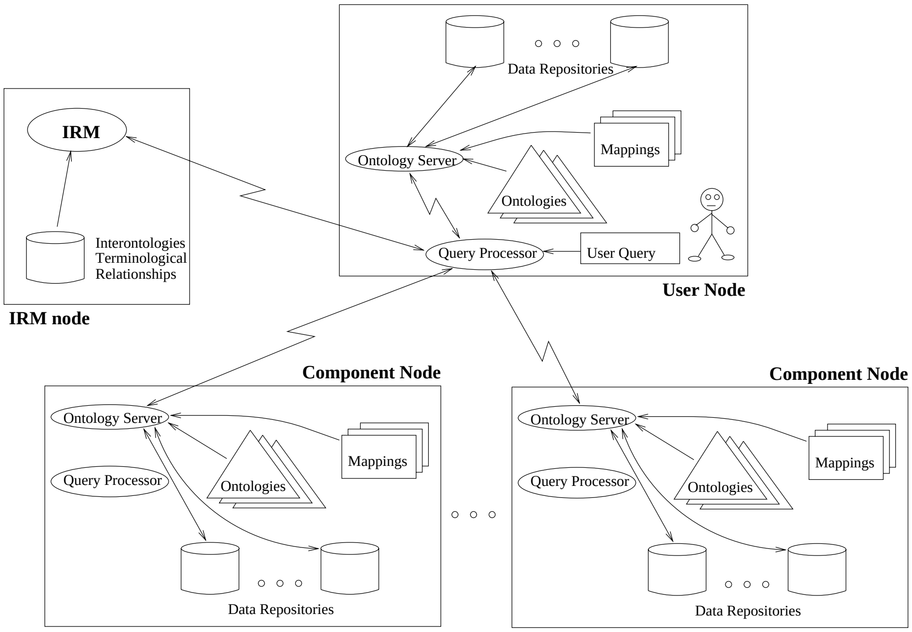


Figure 6: OBSERVER: An architecture to support interoperation across ontologies

Query Processor. This component takes as input a user query expressed in DLs using terms from a chosen *user ontology*. It then navigates other component ontologies of the Global Information System and translates terms in the user query into the component ontologies preserving the semantics of the user query. This may result in a partial translation of the query at a component ontology. It also combines the partial translations at the present ontology with those determined at previous ontologies such that all constraints in the user query are translated.

Ontology Server. The Ontology Server provides information about ontologies to the Query Processor. It provides the definitions of the terms in the ontology and retrieves data underlying the ontology. It is responsible for evaluating the mappings of the contextual expressions to the underlying data and retrieving the data which satisfies the constraints in the user query.

Interontologies Relationships Manager (IRM). Synonym relationships relating the terms in various ontologies are represented in a declarative manner in an independent repository. This enables interoperation across the various ontologies.

Ontologies. Each Ontology is a set of terms of interest in a particular information domain, expressed using DLs in our work. They are organized as a lattice and may be considered as semantically rich metadata capturing the information content of the underlying data repositories. The various ontologies used in OBSERVER are illustrated in the Appendix.

4.1.2 The Interontologies Relationship Manager (IRM)

The IRM is the critical component which supports ontology-based interoperation. It also enhances the **scalability** of the query processing strategy by avoiding the need for: (a) designing a common global ontology containing all the relevant terms in the Global Information System; and (b) investing time and energy for the development of an ontology specific for your needs when "similar" ontologies are available. Relationships between terms across ontologies that capture the overlapping of domains are stored in a repository managed by the IRM. The repository also includes information about transformer functions which can transform values (or role-fillers) from a domain in one ontology to another. The main assumption behind the IRM is that the number of relationships between terms across ontologies is an order of magnitude smaller than the number of all the terms relevant to the system.

Hammer and McLeod [HM93] have suggested a set of relationship descriptors to capture relationships between terms across different (locally developed) ontologies. A set of terminological relationships has been proposed in [Mil95]. In the OBSERVER system we discuss an approach using *synonym* relationships. We will discuss extensions to the OBSERVER system for using *hyponyms* and *hypernyms* in the next section.

4.1.3 Query Processing in OBSERVER

We now discuss a query processing approach that involves the *re-use of pre-existing ontologies* and interoperation across them. The query processor performs the following important steps:

1. Translation of terms in the query into terms in each component ontology. The query processor obtains information from the IRM (discussed in Section 4.1.2) and the Ontology Server.

2. Combining the partial translations in such a way that the semantics of the user query is preserved.

3. Accessing the Ontology Server to obtain the data under the component ontology that satisfy the translated query. This basically amounts to the evaluation of the mappings of the contextual expressions to the underlying database schema and has been discussed in the previous section.

4. Correlation of the objects retrieved from the various data repositories/ontologies.

We illustrate steps 1, 2 and 4 using an example in [MKSI96]. A detailed discussion of the query processing strategy is described in the same paper. Consider a contextual expression represented in CLASSIC used for the following query:

'Get the titles, authors, documents and the number of pages of doctoral theses dealing with "metadata" and that have been published at least once.'

Let us assume that there are 4 ontologies (described in detail in [MKSI96]) as discussed below:

- **Stanford-II** This ontology is a subset of the Bibliographic Data Ontology [Gru94] developed as a part of the ARPA Knowledge Sharing Effort (http://www-ksl.stanford.edu/knowledge-sharing). It corresponds to the sub-tree under the concept 'reference' of the Bibliographic Data Ontology and is illustrated in Appendix D.

- **Stanford-I** This ontology is also a subset of the Bibliographic Data Ontology and corresponds to the rest of the ontology. It is illustrated in Appendix C.

- **WN** This ontology was built by re-using a part of the WordNet 1.5 ontology [Mil95]. The concepts in the WN ontology are a subset of terms in the hyponym tree of the noun "print-media". It is illustrated in Appendix B.

- **LSDIS** This ontology is a local "home-grown" ontology which represents our view of our Lab's publications and is illustrated in Appendix A.

The query can be constructed from the concepts in Stanford-II (denoted as the user ontology) and represented in CLASSIC as follows:

[title author document pages] for (**AND** doctoral-thesis-ref (**FILLS** keywords "metadata")
(**ATLEAST** 1 publisher))

We now enumerate the translations of the query into the ontologies discussed above and identify the translated and non-translated parts:

Stanford-II The query always represents a full translation into the user ontology.

Stanford-I There is a **partial translation** of the query at this ontology.
Translated Part [title author NULL number-of-pages] for
(**AND** doctoral-thesis (**ATLEAST** 1 publisher))
Non-translated Part (**FILLS** keywords "metadata")

WN Terms in the query are substituted by their definitions in the ontology from which they are chosen (Stanford-II) to obtain a complete translation into WN.
doctoral-thesis-ref ≡ (**AND** thesis-ref (**FILLS** type-of-work "doctoral"))
thesis-ref ≡ (**AND** publication-ref (**FILLS** type-of-work "thesis"))
Translated Part [name creator NULL pages] for (**AND** print-media
(**FILLS** content "thesis" "doctoral") (**ATLEAST** 1 publisher)
(**FILLS** general-topics "metadata"))

LSDIS There is a **partial translation** at this ontology where the value of the role-filler of the role keywords is transformed by the transformer function between the roles keywords (Stanford-II) and subject (LSDIS).
Translated Part [title authors location-document NULL] for (**AND** publications
(**FILLS** type "doctoral" "thesis") (**FILLS** subject "METADATA"))
Non-translated Part (**ATLEAST** 1 publisher)

Consider the partial translations of the user query at the ontologies Stanford-I and LSDIS. As the intersection of the non-translated parts of the partial translations into Stanford-I and LSDIS is empty, then the intersection of both partial answers must satisfy all the constraints in the query. Intuitively:

- From Stanford-I, doctoral theses about any subject which have been published at least once will be retrieved;

- From LSDIS, documents about metadata which may not have been published will be retrieved.

- The intersection of the above will be those documents classified as doctoral theses about metadata and have been published at least once, which is exactly the user query.

After obtaining the corresponding data for each ontology involved in the user query, that data must be combined to give an answer to the user. For each answer (represented as a relation), the Query Processor will transform the values in the format of the user ontology by invoking the appropriate transformer functions obtained from the IRM. After this initial step, the different partial answers can be correlated since all of them are expressed in the *language* of the user ontology. The correlation plan corresponding to the translations illustrated above is:

User_Query_Objects = Objects('[self title author document pages] for (**AND** doctoral-thesis-ref (**FILLS** keywords "metadata") (**ATLEAST** 1 publisher))')
Stanford-II_Objects = Objects('[self title author document pages] for (**AND** doctoral-thesis-ref (**FILLS** keywords "metadata") (**ATLEAST** 1 publisher))', Stanford-II)
Stanford-I_Objects = Objects('[self title author NULL number-of-pages] for (**AND** doctoral-thesis (**ATLEAST** 1 publisher))', Stanford-I)
WN_Objects = Objects('[self name creator NULL pages] for (**AND** print-media (**FILLS** content "thesis" content "doctoral") (**FILLS** general-topics "metadata"))', WN)
LSDIS_Objects = Objects('[self title authors location-document NULL] for(**AND** publications (**FILLS** type "doctoral" "thesis") (**FILLS** subject "METADATA")', LSDIS))

Based on the combination of partial translations the data retrieved from the repositories underlying the ontologies can be combined as follows:

$$\begin{aligned}\textbf{User\_Query\_Objects} &= \textbf{Stanford-II\_Objects} \cup \textbf{WN\_Objects} \\ &\cup [\ \textbf{Stanford-I\_Objects} \cap \textbf{LSDIS\_Objects}\ ]\end{aligned}$$

4.2 Using hyponyms and hypernyms to interoperate across ontologies

Synonym relationships between terms in independent developed ontologies are very infrequent. On the contrary, and real examples confirm it, hierarchical relationships like *hyponyms* and *hypernyms* are found more frequently. The substitution of a term by its hypernyms or hyponyms changes the semantics of the query. We try to translate the non-translated terms in the user ontology into terms (which are not its synonyms) in a target component ontology.

We substitute a non-translated term by the intersection of its immediate parents or the union of its immediate children. The loss of information is measured in both cases and translation with less loss of information is chosen. This method is applied recursively until a full translation of the conflicting term is obtained. Using hyponym and hypernym relationships as described above can result in several possible translations of a non-translated term into a target ontology. Very simple intuitive measures depending on the extensions of the terms in the underlying ontologies may help in choosing the translations and minimizing the loss of information.

In order to obtain the immediate parents and children of a term in the target ontology, two different kinds of relationships related to the conflicting term must be used:

1. Synonyms, hyponyms and hypernyms between terms in the user and target ontology.

2. Synonyms, hyponyms and hypernyms in the user ontology.

The first three types of relationships are stored in the IRM repository. The second are relationships between terms in the same ontology; synonyms are equivalent terms, hyponyms are those terms subsumed by the non-translated term and hypernyms are those terms that subsume the conflicting term.

The task of getting the immediate parents/children is not easy to perform. To obtain the parents/children within the user ontology, the corresponding functions (e.g., subsumption) of the DL systems can be used. But we must combine that answer with the immediate parents/children in the target ontology. Taking into account that some relationships stored in the IRM can be redundant (they were independently defined by different ontologies administrators) such a task can be quite difficult. We would need a DL system dealing with "distributed" ontologies.

In Figure 7, we show two ontologies with some relationships between them (arrows are hyponyms relationships, double arrows are synonyms, and dashed lines are interontology relationships) and with the integrated ontology (synonyms are grouped into one term) on the right. We can see that obtaining the immediate parents is not evident; for instance to get the immediate parents of B4 we must deduce that A1 is a child of B1. There are also redundant relationships like the one between A2 and B2.

To work with the above relationships in a homogeneous way, an approach is to integrate the user and the target ontologies, and to use the deductive power of the DL system to obtain the immediate parents/children of a term in the target ontology [BIGP94]. The properties between terms in the different ontologies are exactly the interontology relationships stored in the IRM, so no intervention of the user is needed. Although some of the previous relationships can be redundant, the DL system will classify the terms in the right place in the ontology. To know if the resulting terms of the integrated ontology are *primitive* or *defined* (depending on A and B) the rules described in [BIG94] can be used.

5 Conclusions

We have discussed in this chapter the implications of the exponential growth of the information on the GII on the *semantic heterogeneity* problem and explored new techniques to enable a solution

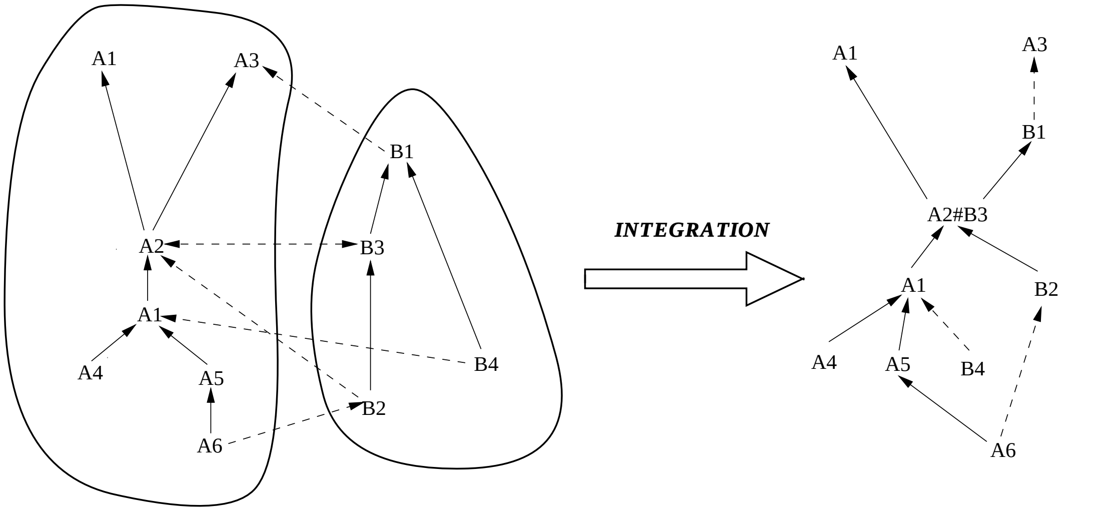


Figure 7: Integrating two ontologies

to the same. Information overload which arises as a consequence of the heterogeneity of the digital data and media types is identified as the first problem. We explore an approach whereby metadata descriptions are used to abstract out the representational details and characterize the information content. An informal classification of the various types of metadata used to handle the wide variety of digital data was presented in Section 2. The amount of information content captured by each is identified and *domain specific metadata* are identified as critical to the semantic heterogeneity problem.

We then discuss how approaches dependent on representational or structural components are inadequate and argue the need for representation of contextual expressions in Section 3. We discussed the representation of these expressions using description logics and propose operations to reason with contextual expressions. We show how *extra information* which may not be represented in the database schema may be represented using contextual descriptions. We illustrated how contextual expressions may be constructed from domain specific ontologies and how terminological relationships between concepts in an ontology enable representation of extra information.

We have recognized the problem of *vocabulary sharing* as the most critical problem in construction of contextual descriptions. We propose approaches to tackle the semantic heterogeneity (as opposed to representational heterogeneity) at this level in Section 4. Semantic interoperability across ontologies is enabled by utilizing terminological relationships like *synonyms, hypernyms* and *hyponyms*.

We have thus explored various approaches based on metadata, context and ontologies which we believe are important and provide the required capabilities to handle the semantic heterogeneity problem in the context of the GII. This research is a part of the InfoQuilt project within the theme of *Enabling Infocosm [KS94b, Fer95]* at the Large Scale Distributed Information Systems Laboratory (http://lsdis.cs.uga.edu/) at the University of Georgia. Some of the interesting research topics that are being investigated further in this this theme are as follows:

- Use of *domain specific metadata* to enable correlation of information across image and structured data. A future extension of this project will be to look into use of metadata standards such as FGDC, OGIS and domain specific ontologies to describe multimedia data.

- Extending the OBSERVER system to enable support for *hyponyms* and *hypernyms*.

- Measures to characterize the loss of information accrued when a term is replaced by expressions with differing semantics. These measures are being developed and experimented within extended OBSERVER system.

- Providing a metadata-based reference link <A MREF ... > as an alternative to the physical reference link <A HREF ... >. This is being implemented as an extension to HTML on the WWW [SK96]. This enables the publisher of an HTML document to specify domain specific metadata which are then mapped to the underlying multimedia data by the enhanced server. This would enable a higher-level metadata based meta-structure over the current WWW.

References

[ACHK93] Y. Arens, C. Chee, C. Hsu, and C. Knoblock. Retrieving and Integrating Data from Multiple Information Sources. *International Journal of Intelligent and Cooperative Information Systems*, 2(2), June 1993.

[AS] J. Anderson and M. Stonebraker. Sequoia 2000 Metadata Schema for Satellite Images, in [KS94c].

[BB93] A. Borgida and R. Brachman. Loading Data into Description Reasoners. In *Proceedings of 1993 ACM SIGMOD*, May 1993.

[BBMR89] A. Borgida, R. Brachman, D. McGuinness, and L. Resnick. CLASSIC: A structural data model for objects. In *Proceedings of ACM SIGMOD-89*, 1989.

[BIG94] J.M. Blanco, A. Illarramendi, and A. Goñi. Building a Federated Database System: An approach using a Knowledge Based System. *International Journal on Intelligent and Cooperative Information Systems*, 3(4):415–455, December 1994.

[BIGP94] J. Blanco, A. Illarramendi, A. Goñi, and J. Perez. Using a terminological system to integrate relational databases. *Information Systems Design and Hypermedia, Cepadues-Editions*, 1994.

[BL$^+$92] T. Berners-Lee et al. World-Wide Web: The Information Universe. *Electronic Networking: Research, Applications and Policy*, 1(2), 1992.

[BR] K. Bohm and T. Rakow. Metadata for Multimedia Documents, in [KS94c].

[CHK$^+$] F. Chen, M. Hearst, J. Kupiec, J. Pederson, and L. Wilcox. Metadata for Mixed-Media Access, in [KS94c].

[CHS91] C. Collet, M. Huhns, and W. Shen. Resource Integration using a Large Knowledge Base in Carnot. *IEEE Computer*, December 1991.

[DDF$^+$90] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Hashman. Indexing by Latent Semantic Indexing. *Journal of the American Society for Information Science*, 41(6), 1990.

[DH84] U. Dayal and H. Hwang. View definition and Generalization for Database Integration of a Multidatabase System. *IEEE Transactions on Software Engineering*, 10(6), November 1984.

[Fer95] C. Ferguson. Into the infocosm. *Computerworld Leadership Series*, 1(6), July 1995.

[FKN91] P. Fankhauser, M. Kracker, and E. Neuhold. Semantic vs. Structural resemblance of Classes. *SIGMOD Record, special issue on Semantic Issues in Multidatabases*, A. Sheth, ed., 20(4), December 1991.

[Gru94] T. Gruber. Theory BIBLIOGRAPHIC-DATA, September 1994. http://www-ksl.stanford.edu/knowledge-sharing/ontologies/html/bibliographic-data/index.html.

[GSW] U. Glavitsch, P. Schauble, and M. Wechsler. Metadata for Integrating Speech Documents in a Text Retrieval System, in [KS94c].

[HM85] D. Heimbigner and D. McLeod. A federated architecture for Information Systems. *ACM Transactions on Office Information Systems*, 3,3, 1985.

[HM93] J. Hammer and D. McLeod. An approach to resolving Semantic Heterogeneity in a Federation of Autonomous, Heterogeneous, Database Systems. *International Journal of Intelligent and Cooperative Information Systems.*, March 1993.

[JH] R. Jain and A. Hampapuram. Representations of Video Databases, in [KS94c].

[KKH94] Y. Kiyoki, T. Kitagawa, and T. Hayama. A meta-database System for Semantic Image Search by a Mathematical Model of Meaning. *SIGMOD Record, special issue on Metadata for Digital Media*, W. Klaus, A. Sheth, eds., 23(4), December 1994.

[KM91] B. Kahle and A. Medlar. An Information System for Corporate Users : Wide Area Information Servers. *Connexions - The Interoperability Report*, 5(11), November 1991.

[KS94a] V. Kashyap and A. Sheth. Semantics-based Information Brokering. In *Proceedings of the Third International Conference on Information and Knowledge Management (CIKM)*, November 1994.

[KS94b] V. Kashyap and A. Sheth. Semantics-based Information Brokering: A step towards realizingthe Infocosm. Technical Report DCS-TR-307, Department of Computer Science, Rutgers University, March 1994.

[KS94c] W. Klaus and A. Sheth. Metadata for digital media. *SIGMOD Record, special issue on Metadata for Digital Media*, W. Klaus, A. Sheth, eds., 23(4), December 1994.

[KS95] V. Kashyap and A. Sheth. Schematic and Semantic Similarities between Database Objects: A Context-based Approach. Technical Report TR-CS-95-001, LSDIS Lab, University of Georgia, January 1995. Available at http://lsdis.cs.uga.edu/˜amit/66-context-algebra.ps; An abridged version [KS96] appears in the VLDB Journal.

[KS96] V. Kashyap and A. Sheth. Semantic and Schematic Similarities between Databases Objects: A Context-based approach. *The VLDB Journal*, 5(4), October 1996. To appear; http://www.cs.uga.edu/LSDIS/˜amit/66b-VLDB.ps.

[KSS95] V. Kashyap, K. Shah, and A. Sheth. Metadata for building the MultiMedia Patch Quilt. In S. Jajodia and V. Subrahmanian, editors, *MultiMedia Database Systems: Issues and Research Directions*. Springer Verlag, 1995.

[LA86] W. Litwin and A. Abdellatif. Multidatabase Interoperability. *IEEE Computer*, 19(12), December 1986.

[LG90] D. Lenat and R. V. Guha. *Building Large Knowledge Based Systems : Representation and Inference in the Cyc Project*. Addison-Wesley Publishing Company Inc, 1990.

[LNE89] J. Larson, S. Navathe, and R. Elmasri. A Theory of Attribute Equivalence in Databases with Application to Schema Integration. *IEEE Transactions on Software Engineering*, 15(4), 1989.

[McC93] J. McCarthy. Notes on formalizing Context. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1993.

[Mil95] G. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11), November 1995.

[MKIS96] E. Mena, V. Kashyap, A. Illarramendi, and A. Sheth. Managing Multiple Information Sources through Ontologies: Relationship between Vocabulary Heterogeneity and Loss of Information. In *Proceedings of the workshop on Knowledge Representation meets Databases in conjunction with European Conference on Artificial Intelligence*, August 1996.

[MKSI96] E. Mena, V. Kashyap, A. Sheth, and A. Illarramendi. OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies. In *Proceedings of the First IFCIS International Conference on Co-operative Information Systems (CoopIS '96)*, June 1996.

[MS95] D. McLeod and A. Si. The Design and Experimental Evaluation of an Information Discovery Mechanism for Networks of Autonomous Database Systems. In *Proceedings of the 11th IEEE Conference on Data Engineering*, February 1995.

[OM93] J. Ordille and B. Miller. Distributed Active Catalogs and Meta-Data Caching in Descriptive Name Services. In *Proceedings of the 13th International Conference on Distributed Computing Systems*, May 1993.

[SG89] A. Sheth and S. Gala. Attribute relationships : An impediment in automating Schema Integration. In *Proceedings of the NSF Workshop on Heterogeneous Databases*, December 1989.

[She91] A. Sheth. Federated Database Systems for managing Distributed, Heterogeneous, and Autonomous Databases. *Tutorial Notes - the 17th VLDB Conference*, September 1991.

[Sho91] Y. Shoham. Varieties of Context, 1991.

[SK92] A. Sheth and V. Kashyap. So Far (Schematically), yet So Near (Semantically). *Invited paper in Proceedings of the IFIP TC2/WG2.6 Conference on Semantics of Interoperable Database Systems, DS-5*, November 1992. In IFIP Transactions A-25, North Holland, 1993.

[SK96] A. Sheth and V. Kashyap. Media-independent Correlation of Information. What? How? In *Proceedings of the First IEEE Metadata Conference*, April 1996. http://lsdis.cs.uga.edu/~kashyap/IEEEpaper.

[SL90] A. Sheth and J. Larson. Federated Database Systems for managing Distributed, Heterogeneous and Autonomous Databases. *ACM Computing Surveys*, 22(3), September 1990.

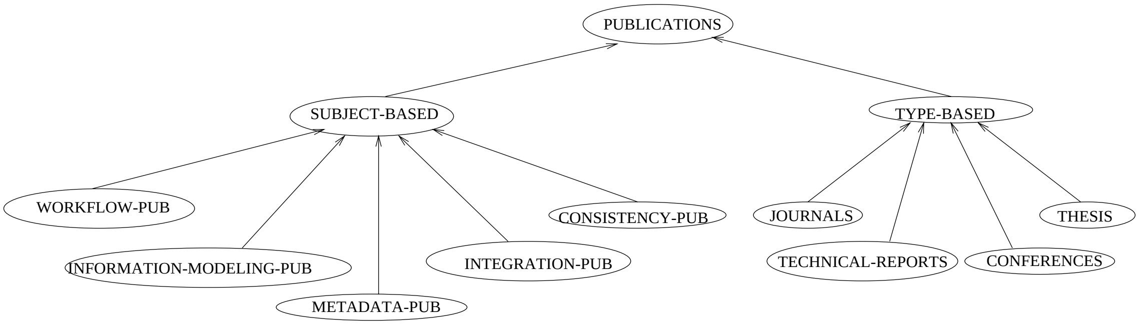
[SLS+93] K. Shoens, A. Luniewski, P. Schwartz, J. Stamos, and J. Thomas. The Rufus System: Information Organization for Semi-Structured Data. In *Proceedings of the 19th VLDB Conference*, September 1993.

[SSKS95] L. Shklar, A. Sheth, V. Kashyap, and K. Shah. Infoharness: Use of Automatically Generated Metadata for Search and Retrieval of Heterogeneous Information. In *Proceedings of CAiSE '95*, June 1995. Lecture Notes in Computer Science, #932.

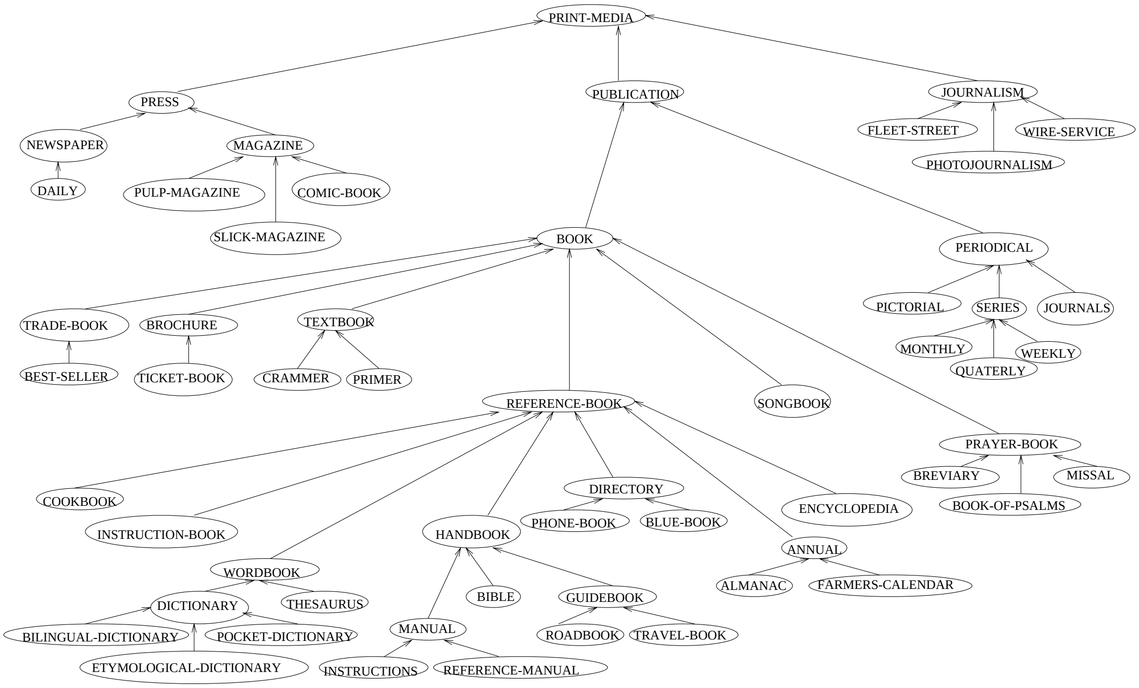
[SSR92] E. Sciore, M. Siegel, and A. Rosenthal. Context Interchange using Meta-Attributes. In *Proceedings of the CIKM*, 1992.

[Wie94] G. Wiederhold. Interoperation, Mediation and Ontologies. In *FGCS Workshop on Heterogeneous Cooperative Knowledge-Bases*, December 1994.

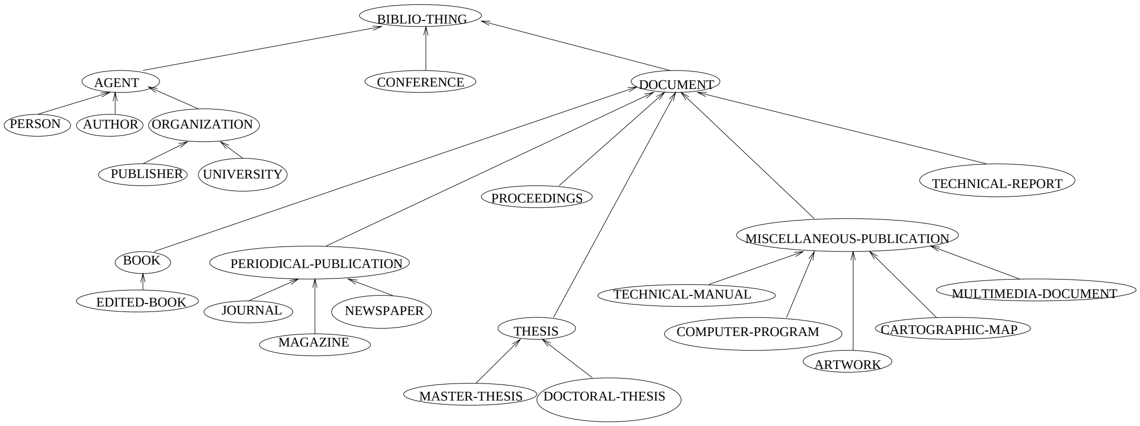
A The LSDIS ontology



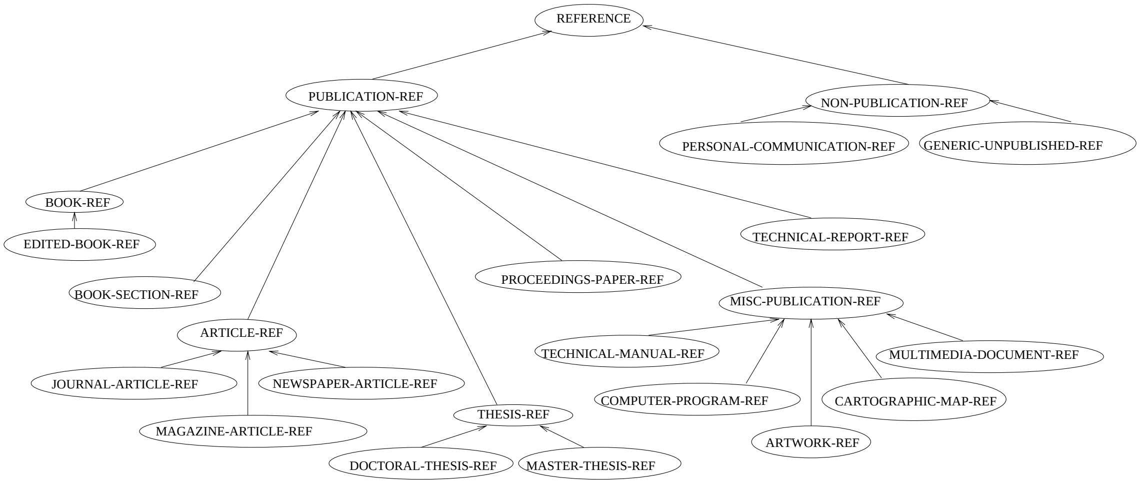
B WN: A subset of the WordNet 1.5 ontology



C Stanford-I: A subset of the Bibliographic-Data ontology



D Stanford-II: A subset of the Bibliographic-Data ontology


REFERENCE
PUBLICATION-REF
NON-PUBLICATION-REF
PERSONAL-COMMUNICATION-REF
GENERIC-UNPUBLISHED-REF
BOOK-REF
EDITED-BOOK-REF
TECHNICAL-REPORT-REF
BOOK-SECTION-REF
PROCEEDINGS-PAPER-REF
MISC-PUBLICATION-REF
ARTICLE-REF
TECHNICAL-MANUAL-REF
MULTIMEDIA-DOCUMENT-REF
JOURNAL-ARTICLE-REF
NEWSPAPER-ARTICLE-REF
COMPUTER-PROGRAM-REF
CARTOGRAPHIC-MAP-REF
THESIS-REF
MAGAZINE-ARTICLE-REF
ARTWORK-REF
DOCTORAL-THESIS-REF
MASTER-THESIS-REF