

# Semantic Mediation For Cooperative Spatial Information Systems: The AMUN Data Model

Eric LECLERCQ, Djamel BENSLIMANE, Kokou YÉTONGNON  
LE2I - Équipe Ingénierie Informatique - Université de Bourgogne  
9, rue Alain Savary - BP 400  
21011 DIJON Cedex - FRANCE

e-mail : {Eric.Leclercq,Djamel.Benslimane,Kokou.Yetongnon}@u-bourgogne.fr

## Abstract

*We present the on-going research project ISIS (Interoperable Spatial Information Systems) which is a semantic mediation approach to support geographical spatial information systems interoperability. The diversity of spatial information systems (GIS) and data models has created a need for tools and methodologies to allow cooperation or inter-operation among GIS. In particular, mediation based approaches, which aim at achieving autonomy, extensibility and flexibility of participant systems, will become increasingly popular in emerging WEB-based processing environments. The underlying approach of ISIS combines techniques from the domains of traditional interoperable information systems, spatial data modeling and multi-agents systems. In this paper we describe the AMUN data model which provides a foundation for resolving semantic discrepancies among systems and for modeling cooperative GIS.*

## 1. Introduction

In the last ten years or so, a significant amount of research has been directed towards designing interoperable systems in which collections of autonomous and heterogeneous information systems can cooperate to carry out tasks [21, 29, 11]. The main thrust of this effort has been on issues related to the integration and interoperation of traditional information systems such as databases, knowledge based, or file based systems. Three major approaches have been identified. The database federation approach uses schema integration techniques to reconcile discrepancies among component systems [3, 29]. Two types of federation have been distinguished. Tightly coupled federations include *global federated schema* that encompass all participants systems while loosely coupled federations contain *local federated schema* that combine information from subsets of partici-

pant. The second approach is the multidatabase language approach in which extended SQL like query languages are used to connect to remote information sources, allowing users to access and manipulate remote data [21]. The third approach is a dynamic mediation in which mediator components are used to provide functionalities or services for combining information from different sources [34, 30, 22].

More recently, some research efforts have been focused on interoperability of geographic information systems. Many GIS applications or systems have been designed as ad-hoc solutions for specific purposes, creating large amounts of high cost spatial data sets stored in various formats. To reduce the high costs incurred by spatial data acquisition, it may be necessary to share data among different systems. For example, the selection of a location for a new commercial mall may require a decision support system that consolidates information from several heterogeneous sources: 1) a GIS which contains roads and traffic information on new the shopping center's location, 2) an information system which provides information population distribution in areas next to the selected location and 3) a database which contains the results of financial analysis and marketing research in the neighborhood of the candidate location. Interoperability among GIS can be hindered by many factors including a diversity of spatial data models (raster, spaghetti, network, geometric, ...), a variety of data formats (DEM, Tiger, SDTS, ...) and differences in type and support for geoprocessing functions (shortest path, map overlay, ...).

A major goal of GIS interoperability is to allow transparent and integrated sharing among systems. To achieve this goal several issues related to heterogeneity, conflicts and common contexts (for interpreting data from other systems) must be properly addressed. In this paper, we present the on-going research project ISIS (Interoperable Spatial Information System) based on a semantic mediation approach which aims to support GIS interoperability. In ISIS,

the emphasis is not on static integration methodologies in which export schema are integrated to resolve semantic conflicts, but rather on a mediation solution in which semantic conflicts can be resolved dynamically by using multi-agent techniques which rely on a set of contexts to carry out semantic correlations or agreements among various systems. We address several key issues regarding how contexts can be represented and used to capture semantics of concepts of different sources, and how semantic similarities between objects can be detected and used to reconcile discrepancies among cooperating systems.

The underlying approach of ISIS combines techniques from the domains of traditional interoperable information systems, spatial data modeling and multi-agents systems. ISIS is based on two key elements. First, a multi-level data model AMUN that provides a set of concepts 1) to represent both textual information (thematic properties) and spatial information, 2) to define semantic contexts, 3) to provide a foundation for the resolution of semantic differences among different contexts and 4) to convert and transfer data objects between systems. Second, a loosely couple agent-based architecture which preserves the autonomy of both information sources and consumers. It comprises a set of agents components to help users to discover information sources relevant to their queries, to access data from multiple independent sources and to identify and exploit the processing capabilities of different sites. The list of data providers that can participate in a query processing step is determined dynamically. Multi-agents technology is used to support autonomous evolution and composability of individual components (sources and receivers).

The remainder of the paper is organized as follows. Section 2 is devoted to a brief description of issues and solutions to GIS interoperability problem. Section 3 presents an overview of ISIS's architecture. Section 4 describes the AMUN data model. Section 5 describes each type of agent. Finally, section 6 concludes the paper.

## 2. GIS Interoperability

In this section, we briefly discuss several issues related to the interoperability of GIS and present recent works that address some of these issues.

### 2.1. Issues related to interoperability

To achieve GIS interoperability, several issues must be addressed. Resolving heterogeneity conflicts among systems is a major issue. Different heterogeneities have been identified. They include 1) schematic conflicts which occur when different data sources use different data models to represent information and 2) semantic conflicts which

arise when the same concept or entity is assigned to different meaning in different data sources. Moreover, in GIS specific spatial conflicts ranging from spatial data representation, spatial scale, spatial fragmentation/aggregation, entity classification, fragmentation, to geometric coordinate systems and spatio-temporal differences [27, 19, 35] must be taken in account. See for example [17] for a detailed description of some discrepancies related spatial data processing. Another important issue is how to represent context information and use it to define common understanding among different systems. To cooperate or share information and services, participating GIS must have reference contexts which can be used to capture the meaning or the usage of concepts. Other issues may include extensibility and composability. Extensibility is the ability to cope with problems that may arise when the number of available data sources increase, and composability relates to requirements for incremental design and construction of interoperation. This is particularly important in dynamic environment where composition, i.e. the set of sites that may cooperate to process a task, may vary in terms of both number and capabilities. Query processing and query optimization in interoperable systems is another key issue which is beyond the scope of this paper.

### 2.2. Background

Recently, GIS interoperability has been the focus of several investigations [5, 24, 12, 33, 27, 19, 18, 20]. In [12], Ken Gardels defines several fundamental requirements of GIS interoperability including 1) generic models to support various GIS functions and capabilities, 2) specific tools or functions to process user applications, and 3) methods and interface to discover and access spatial information resources in a network of systems.

The OpenGIS consortium has defined a generic framework and guidelines for extending classical distributed principles to GIS. The goals are to allow sharing of data, resources and system services among GIS applications, to facilitate information exchanges of among heterogeneous systems, to enable the reuse of software components and to permit the design of extensible systems. The guidelines consist of three interoperation models. The essential model describes abstracting process from real world objects to computer representation [26] [12]. The Open Geodata Model (OGM) provides geographic formalism such as types and schema that can be used to define behavior or methods for geographic elements, to specify a catalog of meta information and to represent spatial reference systems [25]. Finally, the OpenGIS service model defines functions for assembling spatial objects and building complex spatial applications. Agnès Voisard et al. [33] suggest a multi layer decomposition approach based on above guidelines. It con-

sists of four levels including application, abstract services, concrete services and data access levels. Their methodology is primarily intended for designing extensible GIS by allowing combination of different subsystems and services. However their solution can provide a basis for interoperating multiple systems.

In [4], Yaser Bishr et al. describe six different levels of GIS interoperability ranging from network protocols to application semantic. Below, we present a different classification which consists of three levels of interoperability corresponding to the top four levels described by Bishr et al..

### 2.2.1. Platform level interoperability

This level is concerned with hardware, operating systems and network protocols. Generally, these systems are gateways that allow one system to access data from other systems by providing support for the transfer of flat structure files between systems. Some systems provide catalogues containing meta-data description of available information sources. However, there is no attempt to unify descriptions and semantics of the underlying systems. The major drawback is that users must have a-priori knowledge of remote files formats and invoke appropriate converters on transferred files. For example, the GeoWeb [28] project provides a browser and data clearing house for retrieving data sets from remote spatial data servers. The spatial data clearing house contains meta information for locating spatial data servers. Another example is the GIS-WWW gateway project [6] in which users can access to different GIS one at the time by using a browser, a switch and a map converter. The browser is used to query the global system, the switch is used to dispatch and rewrite queries on target data sources, and finally the map converter serves to produce results in picture formats.

### 2.2.2. Syntactic level interoperability

Generally, this level provides functionalities and tools for defining persistent and uniform views over multiple heterogeneous spatial data sources. Access to remote data sets is done via either a high level query language or an application interface. Typically, there is no support for unifying components systems or for reconciling semantic differences. Some solutions, which are comparable to the multidatabase language approach, allow users to connect to remote GIS to submit queries using their own language. Others approaches are based on data exchange format or common data model, and define software tools to convert structure between pairs of GIS.

Včkovski defines the Virtual Data Set model (VDS) [32] to handle field data type (raster). A VDS encapsulates in an object the behavior and representation of field data type.

From a user's point of view, a VDS is visible through a standard interface which provides access to original data. Methods are a persistent part of VDS interface whereas values are virtual in the sense that they are derived on demand. Moreover, VDS can create various views of a field depending on the requirements of potential applications. Finally, VDS supports a common interface implemented in Java for accessing distributed data.

The OGD1 [7] project uses the Transient Data Model, which is derived from the DIGEST [9] model, to allow users to access spatial data through an API developed in C.

The GEO2DIS project [14] allows users to query the global system by using GeOQL query language which is a spatial extension of OQL. With the client software, user first submit queries on a catalog meta-data, then the client software system sends GeOQL queries to a server that translates them into the model of the local GIS which contains the data.

### 2.2.3. Application level interoperability

This level aims at defining seamless system interoperation in which users can access multiple GIS as if they were centralized or integrated spatial systems. Users do not have to have knowledge of data models, data location or the semantics associated with the data. Three major approaches can be distinguished:

- **The federated database approach** focuses on providing integrated global views over information systems, constructing integrated schemas to combine the information contents of component systems. Several authors have discussed extensions of traditional integration to handle spatial heterogeneities. Devogele et al. [8] present an overview of database integration schemes as applied to spatial databases. They discuss techniques for identifying inter-schema correspondence and conflicts that may arise when different criteria or assumptions (different scale, generalizations, etc.) are used in the design of different spatial databases. Others propose dedicated spatial data model or data transformation techniques that can be used to construct integrated schemas [31, 25].

Some recent works have focused on building federations over distributed processing functionalities. Abel et al. in [1] describe a federation architecture based on CORBA. Koschel et al. in [16] develop a web oriented federation approach in which system services are organized in two levels: horizontal services which concern access to spatial data and HTML pages construction and vertical services are dedicated to users.

- **The schema mediation approach** has been the focus of several projects in the GIS realm. These solutions which are based on wrapper/mediators architecture, aim at extending many functionalities including common data models

to incorporate spatial data types. Amann in [2] details a schema mediation approach that uses ODGM 93 as a common object model extended with spatial data types. This solution uses CORBA to connect different spatial servers and defines wrappers for O<sub>2</sub>, Postgres and mSQL. The OASIS project [23], based on mediation, uses a persistent object approach in which each GIS or data repository is seen as a persistent store for spatial objects described by a common data model [24] based on the OpenGIS specifications. Localization of spatial objects is transparent to end users. An object environment and related tools are defined to allow reusability of the functionalities of the participating systems. The GeoChange [10] project extends the schema mediation approach by adding semantic information and by using a meta-data catalog to facilitate discovery of informations. User queries are based on a profile which is constructed incrementally by browsing meta-data.

- **The context mediation approach** is explored by Y. Bishr in [4, 5]. The SEMWEB project is based on an explicit representation of contextual informations which are not described by schema. It provides a representation spatial data semantic through the notion of context which is described by a set of rules and constraints attached to object definitions. The concept of proxy context is used to mediate between two local contexts. Context comparison is achieved by semantic translator which enable users to query remote objects without knowing their semantic, localization or representation.

### 3. An overview of ISIS mediation approach

Figure 1 depicts the functional architecture of ISIS which may help meeting some of the requirements of GIS interoperability. It consists of components which are grouped in two main levels.

- The bottom level, called *wrapper level*, consists of information providers which may use different spatial data models. Each repository is associated with a wrapper whose main task is to facilitate external accesses to the spatial repository by providing export schema described with AMUN's concepts.

- The second level is a *cooperation level* which provides services and functionalities to facilitate semantic resolution and query processing. These services embody the dynamic aspect of the cooperative system, including information source discovery, conflicts resolution, and query execution. The mediation level includes three major features: 1) a mediation context that contains common concepts or objects, representing a common ontology for an applications, i.e. a common understanding without which information sharing is a very complex task, 2) cooperative schemas that act as a mediator between a site and other data sources, and 3) context transformations that link local objects to on-

tological concepts.

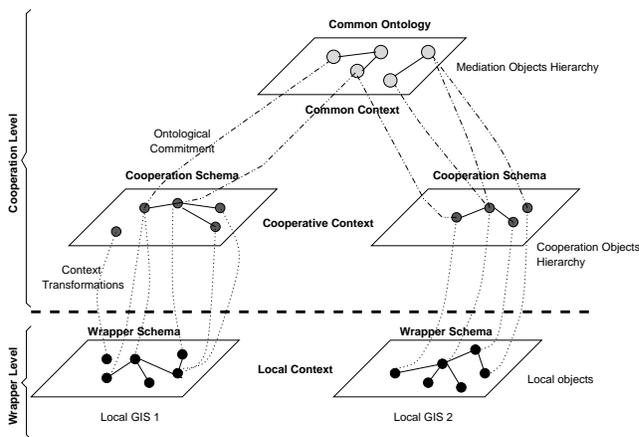


Figure 1. ISIS Functional Architecture

A **Common Ontology** is used to capture the semantic of an application domain and to define a semantic framework that gives concise descriptions of semantic information that are independent of the underlying syntactic representations of local data. A global ontology is often defined as a common vocabulary for a shared domain of discourse [15], thus allowing dialogue and exchange among different sites. Ontologies can be expressed in logical models such as KIF (Knowledge Interface Format) [13] or description logic model. In our approach, the ontology is viewed as a hierarchy of mediation classes described in AMUN model. The description of a mediation class consists of a set of generic properties that can be inherited by cooperation classes, logic rules and constraints which clarify their semantics. Mediation classes form a mediation context or a template models through which local sites define cooperation classes by selectively *agreeing on the descriptions and the semantic of classes*. Within this semantic framework, a mediation class can be used: 1) to carry out semantic similarities among objects from different local contexts, 2) to convert and transfer objects from one information system to another and 3) to represent a virtual (not materialized) extension that may contain semantically similar objects that originate from different classes, and thus may have different descriptions.

**Cooperative schemas** are composed of cooperative classes which represent local semantic interpretation of one mediation class, defining different aspects or facets of ontological concepts. As such, they are comparable to the concept of role used in OO models to represent different roles played by an object. Thus, a mediation class of the common ontology acquires a new role, i.e. a new semantic interpretation, when it is used in the definition of a cooperative class. A cooperative class is a modeling construct that encapsulates three concepts: 1) a mediation role corresponding to a semantic concept, 2) a virtual class (a view), defined on

a set of objects of the local information source. The virtual class implements, in term of the local context, the semantic associated with role and 3) a set of context transformation functions which are describe below.

In our approach, cooperative classes are defined by specifying *ontological agreements on common ontological concepts*. Ontological agreements play a key role in query processing. They are used to discover information sources that can cooperate on the query. An ontological concept can be partially agreed if only a subset of its properties or its ontological constraints are accepted by a site. Otherwise, the agreement is said to be totally agreed on. Ontological commitments are expressed by built-in predicates provided in AMUN model.

**Context transformations** are used to relate cooperate contexts, which contain ontological objects accepted by a site, to local contexts of information sources. Context transformations are defined by mapping functions and are encapsulated into cooperative classes. They are used to convert local objects to ontological properties which can be mapped to remote properties and semantics when an object is transferred from one site to another.

## 4. The AMUN data model

We have pointed out the important role played by semantic considerations in interoperability of GIS. In this section we present the data model AMUN that can be used to represent information at both the wrapper and cooperation levels. The primary intent of AMUN is to provide a set of concepts 1) to represent traditional textual information (thematic properties) and spatial information, 2) to define semantic contexts, 3) to provide a foundation for the resolution of semantic differences among different contexts and 4) to convert and transfer data objects between systems.

### Example :

To illustrate our approach and the different concepts defined in this section, we will use the following example. It consists of two spatial databases  $S_1$  and  $S_2$  that model information on two different sites.

- $S_1$  is a GIS which records information on parcels and farmers for a farm land application. The textual information of interest are: parcel number, owner name, type of crop (culture) and surface. The spatial information is given by an attribute shape which represent the geometry property of parcel. For the farmers entities, the following textual informations are represented: name and first name, address, birth date and status (full-time, seasoned worker, etc.).

- $S_2$  is a government agency's GIS containing informations about workers. It is used to produce statistical maps that show distribution of farmers population over global population in a selected area. The following informations

are recorded on workers: social security number, name and address.

## 4.1. Wrapper layer

The wrapper layer comprises a set of core concepts which are used to represent real world entities, including spatial data types and object oriented core concepts.

### 4.1.1. Spatial Data Types

The predefined spatial data types provided by AMUN are based on a subset of the spatial types of the OpenGIS specifications [25, 26]. OpenGIS spatial types are described by the *Well Known Structures* which are defined in term of coordinates sequences. Furthermore, OpenGIS defines two basic geodata types: features and coverage. A feature type is used to represent real world entities, and a coverage type represent association between points or polygons with a value (for example depth of a lake, wind speed over an area).

In the current version of the project ISIS, only geometric data types (feature types) are included in the model. Figure 2 shows the hierarchy of spatial data type used in the data model AMUN. Geometry which is the highest spatial type in the hierarchy represents general geometric information. Subtype *CoordinateGeometry* is used to model spatial objects that contain coordinate informations. The lowest level of the hierarchy contains the basic spatial data types: *Point*, *LineString*, *Polygon*, ...

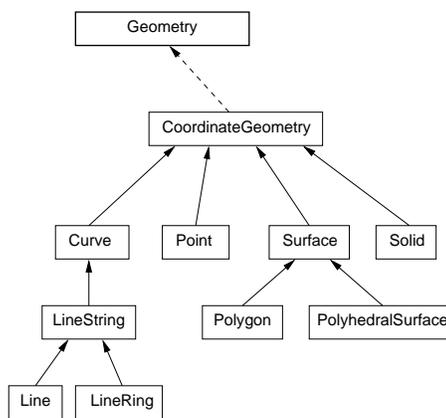


Figure 2. Spatial Types hierarchy of AMUN

### 4.1.2. Core concepts

An object comprises a structure which is defined by attributes, a behavior which is defined by a set of methods and a state which is defined by values taken by its attributes. The attributes of an object can be of thematic or spatial types.

Complex thematic or spatial types can be created by the usual `set` or `tuple` constructors. Attributes can have simple value, complex values or can be references to other objects.

We denote the set of types by  $\mathcal{T}$  and the set of objects by  $\mathcal{O}$ . An object  $o \in \mathcal{O}$  is formally defined by the 3-tuple  $o = \langle oid, Val, MethList \rangle$  where  $oid$  is an identifier which uniquely identifies  $o$ ,  $Val$  is the value of  $o$ ,  $MethList$  is the set of methods of  $o$ .

The specific spatial attribute `Geo` can be included in the description of an object to model the spatial characteristics of the object. It can be an aggregation of features (`set` or `tuple`). For example a lake can have different geometric shapes, one form for each season.

Object classes organize objects into sets of similar entities that share the same structure and behavior. Let  $\mathcal{C}$  denote the set of all the classes. A class  $c \in \mathcal{C}$  is a tuple  $c = \langle Name, AttList, MethList \rangle$  where  $Name$ ,  $AttList$  and  $MethList$  are respectively the name, the list of the attributes and methods belonging to  $c$ . The function  $pop(c)$  defines the set of the objects belonging to  $c$ . The IS-A (subclass) relationship is an acyclic relationship between classes. It states that if a class  $c$  is a subclass of another class  $c'$  then all the instances of  $c$  must also belong to  $c'$ ,  $AttList(c)$  is contained in  $AttList(c')$  and  $MethList(c)$  is contained in  $MethList(c')$ .

#### Example:

In the example GIS S1 above, the entities `Farmer` and `Parcel` can be represented by the following classes: class `Parcel` contains the spatial attribute `Geo`.

```
C1=<Parcel,
  Parcel#: integer,
  OwnerName: string,
  CropType: string,
  Surface: real,
  Geo: POLYGON,
  { WriteOwnerName(Name: string),
    ReadOwnerName(): string ... }>
```

```
C2=<Farmer,
  Name: string,
  Firstname: string,
  Address: [city: string, street: string],
  BirthDate: date,
  Status: string,
  { Age():integer ...}>
```

■ In AMUN, virtual classes which represent (non materialized) views over one or more existing classes can be used 1) to restructure the values of objects, thus allowing multiple representations derived or calculated from the values of an object and 2) to allow aggregation of informations spread over different classes. As will be discussed in detail below, this done by incorporating virtual classes in the definition of cooperation classes.

The following operations can be used for creating virtual classes: `Select`, `Extend`, `Project`, `Union` and `Join`.

The derivation of virtual classes can be carried out by three different processes. First, a specialization process is an abstraction that defines a sub-class of a super-class. A sub-class shares attributes and methods with the super-class and can have additional attributes. The sub-class can be defined by the algebraic operators `Select` and `Extend`. Operation `Select(c,Pred)` restricts the objects instance of a class by selecting objects that satisfy the predicate `Pred`. The `Extend(c,Att)` operation adds the attribute `Att` to the class  $c$ .

Second, a generalization process abstracts common attributes and methods from different classes into a single super-class. The super-class is derived by the algebraic operations `Project` and `Union`. `Project(c,AttList,MethList)` is used to select a subset of attributes and methods of  $c$ . The `Union(ci)` is used to merge the population of different classes  $c_i$ . The derived structure of the class comprise attributes and methods of the source classes.

Finally, an aggregation process defines a complex class from a set of classes. The `Join(c1,c2,Pred)` build a virtual class by assembling component classes  $c_1$  and  $c_2$  into a complex class according to a combination predicate.

#### Example:

The following definition represents two virtual classes for representing land owner and small parcels in  $S_1$ .

- a virtual class `LandOwner` is defined from the class `Parcel`. The `LandOwner` class consist of attribute `OwnerName` and two methods inherited from the class `Parcel`.

```
C3=<LandOwner, OwnerName: string,
  { WriteOwnerName(string Name),
    ReadOwnerName(): string }>
```

`LandOwner` is defined using the `Project` operation as follows:

```
LandOwner=Project(Parcel,OwnerName,
  {WriteOwnerName, ReadOwnerName})
```

- virtual class `SmallParcel` that models parcels whose surface is less than 50 acres. It is a specialization of `Parcel`, with which it shares the same set of attributes and methods. But the population of `SmallParcel` is the subset of the population of `Parcel` satisfying the predicate `Surface < 50`. The class `SmallParcel` can be derived using the `Select` operation.

```
SmallParcel=Select(Parcel,Surface<50)
```

■

## 4.2. Cooperation layer

The cooperation layer is devoted to the resolution of semantic discrepancies among heterogeneous GIS. To achieve this goal we introduce the concept of context which can be used to express semantic informations contained in schemas and to record the assumptions under which a schema is designed. Three types of contexts are defined in the ISIS architecture: 1) reference context model common semantics of an application domain, 2) cooperation contexts are used to interpret the common reference context in terms of concepts or objects of sites and 3) local contexts depict the semantics of local information sources.

### 4.2.1. Reference Context

The reference context serves as a common vocabulary [15] (ontology), identifying and recording informations relevant to a particular application domain. It contains mediation classes which are defined by: static properties, behavior (list of methods) and semantic. The semantic associated with a mediation class is value oriented, i.e. it is used to specify constraints or precise knowledge about possible values taken by an attribute. It can be:

- a domain value (an enumerated type) which spells out the set of values allowed for an attribute. For example, the type of attribute `CropType` can be specified by `{wheat, corn}`.
- a semantic value which is used to express the meaning of an attribute. Typically, a semantic value describes units, coordinate systems or other quality or properties of an attribute. For example, a semantic value `acre` may be associated with the attribute `Surface of Parcel` to state that the surface is measured in acres.
- a logic expression that represents knowledge assertion or a constraint. For example, to state that parcels cultivated with wheat are cereal parcels, a semantic rule is defined: `CerealParcel(X) => Parcel(X)` and `(X.CropType="wheat")`.

A mediation class is formally defined as follows.

#### Definition 1 (Mediation Class)

Let  $\mathcal{MC}$ ,  $\mathcal{MN}$  and  $\mathcal{PN}$  be the set of mediation classes, the set of methods names, the set of parameter names respectively. A class  $mc \in \mathcal{MC}$  is a tuple

$mc = \langle Name, AttList, MethList \rangle$  where:

- $Name(mc)$  is the name of the mediation class  $mc$ .
- $AttList(mc)$  represent the list of attributes of  $mc$ . It is defined by:  $AttList(mc) = \{A_i, A_i : T_{A_i} \vee A_i : T_{A_i} \text{ domval } d \vee A_i : T_{A_i}(B_{j_i} : d)\}$ ,  $i = 1..n$ ,  $j_i = 1..m_i$ ,  $A_i \in Att_{name}$ ,  $B_{j_i} \in Att_{name}$ ,  $T_{A_i} \in T$ .  $\text{domval } d$  specify the domain of the attribute  $A_i$  by enumerating the values (simple or composed) allowed for  $A_i$ .

The expression  $A_i : T_{A_i}(B_{j_i} : d)$  defines the semantic value of  $A_i$  by adding the meta-attribute  $B_{j_i}$ .

- $MethList(mc)$  is the list of methods of  $mc$ . It is defined by:  $MethList(mc) = \{M_j, M_j : \{p_{k_j} : T_{k_j}\} \vee M_j : \{p_{k_j} : T_{k_j}\} : T_{res}\}$ ,  $j = 1..m$ ,  $k_j = 1..q_j$ ,  $M_j \in \mathcal{MN}$ ,  $T_{k_j} \in \mathcal{T}$ ,  $T_{res} \in \mathcal{T}$ ,  $p_{k_j} \in \mathcal{PN}$ .

- function  $pop(mc)$  computes the extent of  $mc$
- $mr(mc)$  gives the list  $\{cc_1, cc_2, \dots, cc_n\}$  of cooperation classes which defines roles played by  $mc$

□

Note that a mediation class defines a virtual class and has no actual instances. Virtual extensions consisting of instances defined at the local information sources can be associated with a mediation class. When needed, these extensions can be computed by merging the extensions of the corresponding cooperation classes. Moreover, the resulting calculated extensions are composed of different informations sources, and thus may have different structures. A mediation class represents a semantic description or a semantic pattern that provides a foundation for defining mediation roles which are used in the representation of cooperation classes. Mediation roles are presented in the next section.

#### Example:

The following mediation class specifies a simple type `Person` to describe a person by name, birth name, sex, and birth date. It includes a method `age`. To state the fact that a person is male or female, the domain of the attribute `sex` is an enumerated set that contains two possible string values: `male` and `female`.

```
<Person, SurName: string, BirthName: string,
  Sex: string domval {"male","female"},
  BDate: date, {Age():integer}>
```

■

### 4.2.2. Cooperation context

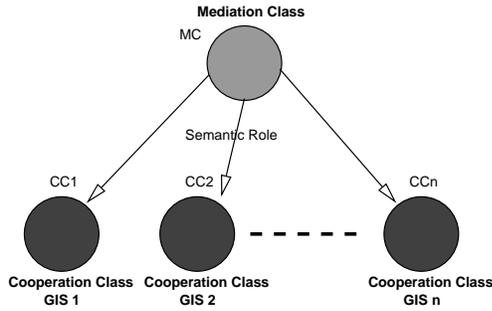
On a site, a cooperation context acts as a mediator between the reference context and the local data context. It consists of cooperation classes which are used to express local interpretations (mediation roles), i.e. local agreement or acceptance, of mediation classes.

#### A. Mediation roles

As stated above, to cooperate and reconcile semantic differences, participants GIS need a set of commonly understood objects to interpret data from other sites. Interaction between sites will be done through different perspectives of the commonly agreed on objects. In our approach, the common objects are represented by mediation classes and the different interpretations are different roles played by the

mediation classes on different sites. Figure 3 shows a mediation class and corresponding roles defined in cooperation classes  $CC_1, CC_2, \dots, CC_n$ . A mediation role is defined by: 1) describing the subset of attributes (of a mediation class) on which the local site agrees on, 2) using a qualification to restrict the properties or semantics of the objects that plays the role.

Like a class, a mediation role has a set of attributes and methods which define its properties and behavior. But unlike class it does not create or delete any objects. Formally, a mediation role is defined as follows.



**Figure 3. Semantic roles played by a mediation class**

**Definition 2 (Mediation Role)**

Let  $\mathcal{MR}$  be the set of mediation roles. A mediation role  $mr \in \mathcal{MR}$  is a tuple

$mr = \langle mc, AttList, MethList, Q \rangle$  where:

- $mc \in \mathcal{MC}$  is a mediation class whose interpretation is  $mr$
- $AttList(mr) \subseteq AttList(mc)$  is a subset of attributes of  $mc$  selected by using algebraic operations **Select** and **Project**
- $MethList(mr) \subseteq MethList(mc)$  is a subset of the methods of  $mc$
- $Q$  is a logic formula (qualification formula) associated with  $mr$ . It can be used to specify a constraint on the objects that play the role  $mr$ .

□

Roles can be shared by objects, for example both `LandOwner` and `Farmer` defined roles played by the mediation class `Person`.

**Example:**

This example represents two mediation roles `LandOwner` and `Farmer` corresponding to mediation class `Person`.

```
r1=<Person,
    Name: string>
r2=<Person,
    Name: string,
```

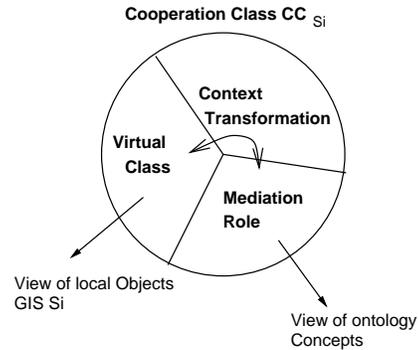
```
    BirthName: string,
    Sex: string,
    BDate: date,
    {Age():integer},
    BDate>"01/01/1928">
```

They model two local interpretations of the mediation class `Person` in the GIS  $S_1$ . For role `r1` corresponding to `LandOwner`, only the `Name` is given. For role `r2` corresponding to `Farmer`, all information in the mediation class are supplied. Furthermore, the qualification associated with `BDate` is used to state the fact that ages of the farmers in the GIS  $S_1$  are less than 70.

■

**B. Cooperation objects and classes**

A cooperation class incorporates three components: a mediation role, i.e. a view defined by a virtual class which links the mediation role to local objects, and context transformations which map objects description from one cooperative context to another (see figure 4). In addition to the descriptions defined by mediation roles, cooperative classes can have specific attributes, methods or semantic constraints. Furthermore, mediation roles can be inherited from super cooperative class to sub cooperative class. Cooperation classes have the following characteristics: 1) cooperation classes are the means by which local objects are shared between GIS, 2) cooperation classes from different GIS are semantically equivalent if they are defined using the same term of the ontology, 3) instances of cooperation classes can be complex objects if they are aggregated by virtual classes.



**Figure 4. A cooperation class of a site  $S_i$**

The formal definition of cooperation objects and cooperation classes are given below in definition 3 and definition 4 respectively.

**Definition 3 (Cooperation Object)**

Let  $\mathcal{CO}$  be the set of cooperation objects and  $\mathcal{CC}$  the set of cooperation classes.

A cooperation object  $co \in \mathcal{CO}$  is a tuple:

$co = \langle oid, cc, val, MethList, context_o \rangle$  where :

- $oid \in OID$  is the oid of  $co$
- $cc \in \mathcal{CC}$  is the cooperation class  $co$  belongs to
- $val$  is the value of the cooperation objet,  $val$  is also called local value of  $co$
- $MethList(co)$  is the set of methods binded to  $co$
- $context(co) = \{ \langle val(mr), CTF \rangle \}$  where  $mr \in MR$  is a mediation role,  $val(mr)$  its value and  $CTF$  is the set of context transformations which convert a local value of  $co$  to the corresponding value for  $mr$ .

□

#### Definition 4 (Cooperation Class)

A cooperation class  $cc \in \mathcal{CC}$  is a tuple

$cc = \langle Name, AttList, MethList, cv, context \rangle$  where:

- $Name(cc)$  is the name of the cooperation class
- $AttList(cc) = \{A_i : T_{A_i}\}$ ,  $i = 1..n$ ,  $A_i \in Att_{name}$ ,  $T_{A_i} \in \mathcal{T}$  is the set of attributes of  $cc$
- $MethList(cc) = \{M_j : \{p_k : T_{p_k}\} | M_j : \{p_k : T_{p_k}\} : T_{res}\}$ ,  $j = 1..m$ ,  $k = 1..q$ ,  $M_j \in \mathcal{MN}$ ,  $T_{p_k} \in \mathcal{T}$ ,  $T_{res} \in \mathcal{T}$ ,  $p_k \in \mathcal{PN}$  is the set of the methods attached to  $cc$
- $context(cc) = \{ \langle mr, CTF \rangle \}$  defines the context of  $cc$ . It is a set of tuples where  $mc$  is a mediation class such as  $\exists mr \in MR$   $mr.cc = mc$ ,  $CTF$  is a set of context transformations.

- $cv$  is a virtual class encapsulated by  $cc$  such that:  
 $AttList(cc) \subseteq AttList(cv)$  and  
 $AttList(mr.cc) \subseteq AttList(cv)$  and  
 $MethList(cc) \subseteq MethList(cv)$  and  
 $MethList(mr.cc) \subseteq MethList(cv)$ .

□

### C. Context transformations

A context transformation is a function which establishes a mapping between on local value domain to a cooperation value domain. A context transformation is associated with each mediation role to allow objects to migrate from a local context to a cooperation context. Figure 5 shows a context conversion between two GIS. A semantic translation process allows objects defined in context  $C_1$  to be used in a context  $C_2$ . It consists of a sequence of two partial context transformations: from  $C_1$  to the reference context then from the reference context to  $C_2$ . This take place by using cooperation objects and mediation objects.

#### Definition 5 (Context Transformation)

$CTF$  is the set of the context transformations:

$CTF = \{ \uparrow_o^{mr} \} \cup \{ \downarrow_o^{mr} \}$ . They are defined by:

- the functions type  $\uparrow_o^{mr}$ , are used to translate a local value of an object to its value for the mediation role  $mr$ .

$\uparrow_o^{mr}: dom(A_1) \times \dots \times dom(A_n) \rightarrow dom(A'_i)$ ,  
 $A_1, \dots, A_n \in AttList(c)$ ,  $A'_i \in AttList(mr)$

- the functions type  $\downarrow_o^{mr}$  are used to translate a mediation role value of an object to its local value.

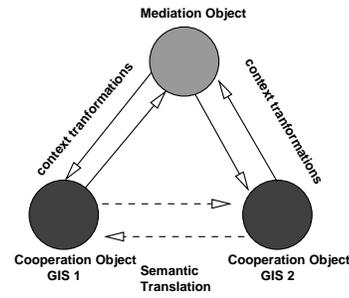


Figure 5. Context Transformations

$\downarrow_o^{mr}: dom(A'_1) \times \dots \times dom(A'_k) \rightarrow dom(A_l)$   
 $A'_1, \dots, A'_k \in AttList(mr)$ ,  $A_l \in AttList(c)$

□

#### Example:

The following examples depict context transformations between the mediation role Person and the class Farmer.

```

 $\uparrow_o^{mr}$  Person.Surname(Farmer.Name)={
    return(Farmer.Name)}
 $\downarrow_o^{mr}$  Farmer.Name(Person.Surname)={
    return(Person.Surname)}

```

In GIS  $S_1$  the attribute sex of Farmer is coded by a single character (M or F) while in the mediation class Person this attribute is coded by male or female.

```

 $\uparrow_o^{mr}$  Person.Sex(Farmer.Sex)=
    {if Farmer.Sex="M" return("Male")
     else return("Female")}
 $\downarrow_o^{mr}$  Farmer.Sex(Person.Sex)=
    {if Person.Sex="Male" return("M")
     else return("Female")}

```

The cooperation class CCFarmer encapsulates both the virtual class CVFarmer and its context (mediation role, context transformations and qualification). The virtual class CVFarmer and the cooperation class CCFarmer are defined below :

```

<CVFarmer, Name: string, First-
name: string, Sex: string,
  Address: [city: string, street: string],
  BirthDate: date, Status: string,
  {Age(): integer...}>

<CCFarmer, Name: string,
  Firstname: string,
  Sex: string,
  Address: [city: string, street: string],
  BirthDate: date,
  Status: string,
  {Age(): integer...}, CVFarmer, {<
<Person, Surname: string, BirthName: string,
  Sex: string, BDate: date, {...}>,<...
 $\uparrow_o^{mr}$  Person.Surname(Farmer.Name)={
    return(Farmer.Name)}
 $\downarrow_o^{mr}$  Farmer.Name(Person.Surname)={
    return(Person.Surname)}...>,
  BDate>"01/01/1928">>

```

Site  $S_2$  only records information about workers and not farmers. In a first step, it is interesting to obtain global informations about persons living in a county without considering their profession. So,  $S_2$  defines a cooperation class `CCPerson` linked with a virtual class `CVPerson` defined with local class `Worker` and having a mediation role `Person`. When  $S_2$  is queried for `Person` the mediation role is used and objects corresponding to person on  $S_1$  are also acceded.

■

In the next section we will show how the concepts introduced in the previous section will take place in the agent architecture.

## 5. ISIS agent based architecture

In the ISIS project, we have proposed a set of architectural components to aid users 1) to locate information sources and data relevant to their queries, 2) to access data from multiple independent information sources and 3) to identify and exploit processing capabilities of different sites. Figure 6 shows the proposed architecture. It includes six types of agent which are described below. For each agent type, we describe its general objective, the services it provides, the data and knowledge it contains and the different agents it communicates with.

**A Wrapper Agent (WA)** is used to submit elementary queries to the underlying local GIS. It encapsulates the local GIS in a generic spatial object server capable of accessing and retrieving local data. Each WA is associated with a single cooperation agent from which it receives and processes queries. Query processing is decomposed into the following stages: 1) translation of the OQL query to target local query language, 2) execution of the target query on a local GIS and 3) transfer of local results to the corresponding cooperation agent. To hide schematic heterogeneities of the local systems, the schema of WA agents are represented using the core concepts of the AMUN data model. WA are reactive agents, thus they are activated only when messages are sent from cooperation agents.

**A Cooperation Agent (CA)** seeks and coordinates the assistance of other agents to process high-level queries. It contains knowledge representing the cooperation context. As discussed earlier, this knowledge consists of a cooperation schema (cooperation objects) that is used to relate the contents of a wrapper's schema to a set of concept agreements that correspond to the semantics of accepted ontological concepts. To create the agreements, a CA communicate with a semantic router agent to discover information. A CA can receive queries either from a user, or from another cooperation agent. It translates queries from the cooperation context to the local context by using transformation functions to rewrite queries on cooperation objects to equivalent

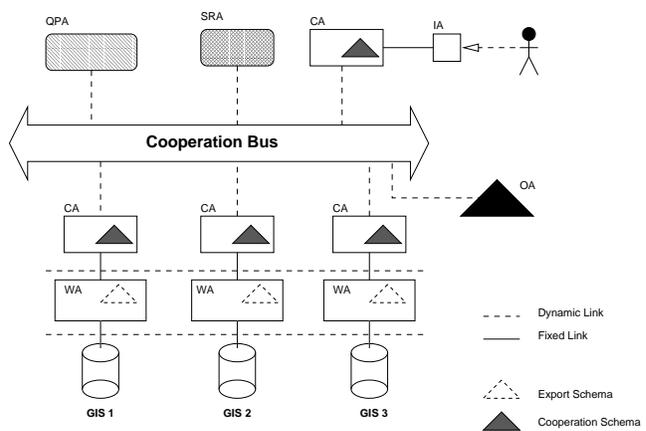
queries on local objects which are submitted to the wrapper agent. Note that since cooperation agents use the same model, subqueries which are sent to other CA do not require translation.

**An Ontology Agent (OA)** allows communication among different agents to answer a query. To exchange queries without using global schema, to span multiple sources, agents use a common ontology which provides a mutual understanding of the query. The ontology agent provides definitions of the common terms which are used by the query processor, the semantic router and the cooperative agents. Interoperability relies on the ability of the OA to essentially create ontological commitments which can be total or partial agreements on the shared semantic. Agreements are stored in the semantic router agent. OA uses these ontological commitments to determine agents capabilities. A cooperation agent is not required to answer all queries that can be formulated in the shared vocabulary.

**A Semantic Router Agent (SRA)** is used to provide name service or discovery service to cooperation agents. When a CA receives a query it calls the SRA agent to discover the identities of other agents (CA) which contains information relevant to the query and can contribute to its execution. For each agreed upon concept, the SRA agent maintains a list of predicates to specify the names of CA.

**An Interface Agent (IA)** is an intermediary agent which receives queries from users, send the queries to the associated cooperation agent and deliver the results to users. Each IA is connected to only one cooperation agent, and thus users can directly interact with only one IA even if the execution of the submitted query may involve several CA.

**A Query Processor Agent (QPA)** takes as input a query expressed over an ontological schema and uses a semantic router to 1) identify relevant informations sources and 2) create an execution plan.



**Figure 6. ISIS Agent Based Mediation Architecture**

Using multi-agents concepts to define mediation among

multiple information systems requires 1) protocols for implementing both communications among agents and 2) a distributed architecture to support messages passing. ISIS uses a subset of KQML and implements a subset of performatives (ask, tell, register, etc.). The architecture is based on CORBA and Java. We have developed several types of wrapper (O2, Access, Postgres) and a generic Java class which implement the generic structure of an agent (e.g. communication module, event engine).

## 6. Conclusion

In this paper, we have focused on key issues relevant to the design of interoperable GIS. We argue that resolution of semantic differences among various systems must be based on context informations that can be used to capture the semantics of various systems. To achieve this, we define a reference context which is composed of a common ontology that defines a semantic framework shared by the participating information sources. To provide a foundation for specifying the semantics and properties of shared data, we introduce a distributed spatial data model AMUN that includes:

- a set of concepts for handling distribution and heterogeneities: virtual classes, mediation class, mediation role, context transformation and cooperation class;
- spatial data types for specifying spatial objects;

Furthermore, we propose an agent based mediation architecture to allow spatial data sharing and cooperative query processing. It's major components are: 1) wrappers (one per local GIS) for resolving heterogeneity in the cooperative environment, 2) mediators for coordinating cooperative tasks such as communications service or query dispatching. The main advantage of the architecture is it brings core software components to be used in different contexts and thus it allows flexible and extensible cooperation in different environments such as WEB [4], federated GIS whitout using global schema.

The initial stages of our project are devoted to the definition of the architecture and the data model. Our future work will focus on spatial query processing to handle the distribution and sharing not only of spatial objects but also of specialized spatial operators.

## References

- [1] D. Abel, B. Ooi, K. Tan, and S. H. Tan. Towards integrated geographical information geoprocessing. *Int. Journal of Geographical Information Science*, 12:353–371, 1998.
- [2] B. Amann. Integration GIS components with mediators and CORBA. Technical report, CEDRIC CNAM, 1997. <ftp://sikkim.cnam.fr/pub/Reports/GISMed.ps.gz>.
- [3] C. Batini, M. Lenzerini, and S. B. Navathe. A comparative analysis of methodologies for database schema integration. *ACM Computing Surveys*, 18:323–364, December 1986.
- [4] Y. Bishr. *Semantic Aspects of interoperable GIS*. PhD thesis, ITC Enschede NL, 1997.
- [5] Y. Bishr. Overcoming the semantic and other barriers to GIS interoperability. *Int. Journal of Geographical Information Science*, 12:299–314, 1998.
- [6] C. Boehner, P. Haastrup, and A. Reggiori. The GIS-WWW gateway. In *Proc. of the Third Joint European Conference and Exhibition on Geographical Information*, pages 125–134. Vienna, Austria, 1997.
- [7] G. Clement, C. Larouche, D. Goiun, P. Morin, and H. Kucera. OGD: toward interoperability among geospatial databases. *ACM SIGMOD Record*, pages 18–23, 1997.
- [8] T. Devoegele, C. Parent, and S. Spaccapietra. On spatial database integration. *Int. Journal of Geographical Information Science*, 12:335–352, 1998.
- [9] D. DGIWG. The digital geographic information exchange standard (Digest 2.0). Technical report, Directorate of Geomatics Department of National Defence, Canada, June 1997.
- [10] P. Drew and J. Ying. GeoChange: an experiment in wide-area database services for geographic information exchange. In *Proceedings of the Third Forum on Research and Technology Advances in Digital Library, ADL '96*, pages 14–23. IEEE Computer Society Press, 1996.
- [11] O. E. Pitoura and A. Elmagarmid. Object orientation in multidatabase systems. *ACM Computing Surveys*, 27:141–195, jun 1995.
- [12] K. Gardels. OpenGIS and on-line environmental libraries. *ACM SIGMOD Record*, 26:32–38, March 1997.
- [13] M. Genesereth and R. Fikes. Knowledge interchange format version 3.0 reference manual. Technical Report 92-1, Logic Group, Computer Science Department Stanford University CA, June 1992.
- [14] GEO2DIS. GEO2DIS: a client-server architecture on internet to document and access geodata stored on heterogeneous GISs. Technical report, 1997. <http://www.pisa.intecs.it/project/GEO2DIS>.
- [15] T. Gruber. Towards principles for the design of ontologies used for knowledge sharing. *Int. Journal of Human-Computer Studies*, 43:907–928, jun 1995.
- [16] A. Koschel, R. Kramer, R. Nikolai, W. Hagg, J. Wiesel, and H. Jacobs. A federation architecture for an environmental information system incorporating gis, the world wide web and CORBA. In N. C. F. G. Information and A. (NCGIA), editors, *Third International Conference Workshop on Integrating GIS and Environmental Modeling*, Santa Fe, New Mexico USA, January 1996.
- [17] R. Laurini. Spatial multidatabase indexing and topological continuity of fragmented geographic objects. In *Proc. of Parallel and Distributed Computing Systems*, pages 784–787. Dijon, France, 1996.
- [18] R. Laurini. La télégéomatique : problématiques et perspectives. *Revue internationale de géomatique*, 8(1-2):27–58, June 1998.
- [19] R. Laurini. Spatial multidatabase continuity and indexing: a step toward seamless gis data interoperability. *Int. Journal of Geographical Information Science*, 12:373–402, 1998.

- [20] E. Leclercq, D. Benslimane, and K. Yétongnon. Amun: An object-oriented model for cooperative spatial information systems. In *IEEE Knowledge and Data Engineering Exchange Workshop, Newport Beach, USA*, pages 73–80, 1997.
- [21] W. Litwin, L. Mark, and N. Roussopoulos. Interoperability of multiple autonomous databases. *ACM Computing Surveys*, 22:265–293, jun 1990.
- [22] L. Liu and C. Pu. The distributed interoperable object model and its application to large-scale interoperable database systems. In *International Conference on Information and Knowledge Management (CIKM)*, 1995.
- [23] E. Mesrobian, R. Muntz, E. Shek, S. Nittel, M. L. Rouche, and M. Kriguer. OASIS: An open architecture scientific information system. In *Sixth International Workshop on Research Issues in Data Engineering - Interoperability of Non-traditional Database Systems*, pages 107–116, 1996.
- [24] S. Nittel, R. Muntz, and E. Mesrobian. geoPOM: a heterogeneous geoscientific persistent object system. In *Statistical and Scientific Database Management (SSDBM)*, 1997.
- [25] Open GIS Consortium and The OGIS Technical Committee. The Open GIS abstract specification. Technical report, 1996.
- [26] Open GIS Consortium and The OGIS Technical Committee. The Open GIS Guide Part I: Introduction to interoperable geoprocessing. Technical report, 1996.
- [27] C. Parent, S. Spaccapietra, and T. Devogele. Conflicts in spatial database integration. In *Proc. of Parallel and Distributed Computing Systems*, pages 772–778. Dijon, France, 1996.
- [28] B. Plewe. The GeoWeb project : Using WAIS and the world wide web to aid location od distributeddata sets. Technical report, State University of New York at Buffalo, 1994.
- [29] A. Sheth and J. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22:183–236, September 1990.
- [30] A. Tomasic, L. Raschid, and P. Valduriez. Scaling heterogeneous databases and the design of Disco. In *Proceedings of the 16th International Conference on Distributed Computing Systems*, pages 449–457. IEEE Computer Society., May 1996.
- [31] N. Tryfona and J. Sharma. On information modeling to support interoperable spatial databases. In *Advances Information System Engineering, 8th International Conference, CAiSE'96, Heraklion, Crete, Greece, May 20-24*, pages 210–221, 1996.
- [32] A. Včkovski. Java as a software system for distributed and interoperable geoprocessing. In *Proc. of Parallel and Distributed Computing Systems*, pages 779–783. Dijon, France, 1996.
- [33] A. Voisard and H. Schweppe. A multilayer approach to the open gis design problem. In *Second International ACM GIS Workshop*, pages 23–29, 1994.
- [34] G. Wiederhod. Mediators in the architecture of future information systems. *IEEE Computer Magazine*, 25(3):38–49, March 1992.
- [35] M. Worboys and S. Deen. Semantic heterogeneity in distributed geographic databases. *ACM SIGMOD Record*, 20(4):30–34, 1991.