HORUS: a Semantic Broker for GIS Interoperability

Eric LECLERCQ, Djamal BENSLIMANE, KokouYÉTONGNON
LE2I - Équipe Ingénierie Informatique - Université de Bourgogne
9, rue Alain Savary - BP 400
21011 DIJON Cedex - FRANCE
e-mail : {Eric.Leclercq,Djamal.Benslimane,Kokou.Yetongnon}@u-bourgogne.fr

Abstract

The diversity of spatial information systems promotes the need to integrate heterogeneous spatial or geographic information systems (GIS) in a cooperative environment. This paper describes HORUS (an Heterogeneous Object Request Uniformization System), a part of the research project ISIS (Interoperable Spatial Information System) which adapts semantic mediation approach to GIS interoperability. Its key characteristic is a dynamic resolution of semantic conflicts which is adequate for achieving autonomy, flexibility and extensibility required in WEB environments. We propose a loosely coupled architecture based on distributed object paradigm, to share spatial information and services. We describe a spatial OO model that provides tools to model distribution and to resolve semantic heterogeneities according to the mediation model defined.

1 A Introduction

GIS are used in various type of organizations to provide support for spatial or geo-referenced data processing applications ranging from natural resource management, urban planning, to traffic control or environment service management. Many of these systems are ad hoc information systems designed for specific purposes. They represent large repositories of spatial data which are stored in different formats. With the rapid development of networking technologies and WEB-type environments, a growing number of heterogeneous and autonomous spatial data sources are becoming avalaible to users. GIS interoperability is making an important impact on the acheivement of complex spatial applications by allowing the design of information systems in which collections of autonomous information sources interoperate or cooperate to carry out tasks. The motivation for GIS interoperability is the need 1) to reduce high spatial data acquisition costs, 2) to avoid creation of redundant data repository intailing inconsistences and 3) to share and exchange spatial data among systems.

To match the requirements of emerging data intensive applications, methodologies and tools are needed to allow uniform, transparent and integrated access to multiple data sources. For example, consider an urban planning management system that is used to select a location for building a new commercial mall. This choice can be made by using a decision support system to combine or exchange information from several information sources including 1) a GIS that contains road and traffic information on the location of the shopping center, 2) an information system that includes population distribution in areas next to the selected location and 3) a database consisting of the results of financial analysis and marketing research carried out in the selected areas.

The primary intent of GIS interoperability is to provide functionalities to allow transparent and integrated sharing among systems by addressing issues related to spatial data heterogeneity, conflicts and contexts (to interpret data from other systems). Exchanging and sharing information among heterogeneous and autonomous GIS may be hindered by differences in 1) the spatial data models (raster, spaghetti, network, geometric...) and data formats (DEM, Tiger, SDTS, ...) used to represent the data and 2) specific geoprocessing functions (shortest path, map overlay, ...) supported by systems. Furthermore, users of spatial application work under different semantic assumptions. They have different views on their working environments and they use different representations. Thus, different semantic contexts (interpretation of the real world) may be associated with each spatial information systems.

In this paper, we present HORUS, a part of an ongoing research project called ISIS. The main contribution of the project is a semantic mediation approach to support GIS interoperability. The focus of the project is not on using static integration methodologies in which export schema are integrated to resolve semantic conflicts, but rather on a dynamic resolution of semantic conflicts which is more adequate for achieving autonomy, flexibility and extensibility. We address several key issues related to 1) how semantic contexts can be represented and used to capture the semantics of concepts from different sources, and 2) how semantic similarities between objects can be detected and used to reconcile discrepancies among cooperating systems.

HORUS proposes a semantic extension of the distributed object paradigm which handles both the semantic mediation process and the mediation architecture. The key features of the methodology are embodied in a multi-level data model AMUN which provides a set of concepts used for 1) representing both traditional textual (thematic properties) and spatial information, 2) defining semantic contexts, 3) providing a foundation for the resolution of semantic differences among different contexts and 4) converting and transferring data objects between systems. The mediation architecture consists of a set of software services built over distributed object technology which provides autonomy of both information sources and consumers, flexibility of the global system and a dynamic interconnexion of participating systems. The list of data providers capable of cooperating in a query processing step is calculated at execution time.

The remainder of the paper is organized as follows. Section 2 is devoted to a brief description of issues and solutions to the problem of GIS interoperability. Section 3 presents an overview of ISIS's mediation approach and HORUS brokering process. Section 4 presents the AMUN data model. Finally section 5 concludes the paper.

2 GIS Interoperability

In this section, we briefly discuss several issues related to interoperability of information systems, including GIS. Then, we present a background of approaches for GIS interoperability.

2.1 Some issues related to interoperability

To achieve GIS interoperability, several issues must be addressed. Spatial data combine several characteristics that are problematic from an interoperability point of view.

First, resolving heterogeneity conflicts among systems is a major issue. Different heterogeneities have been identified, including schematic conflicts which arise when different data sources use different data models to represent information and semantic conflicts which arise when the same concept or entity is assigned to different meaning in different sources. In GIS, specific spatial conflicts must also be taken into account. They range from spatial representation, spatial scale, spatial fragmentation aggregation, entity classification, fragmentation, geometric to coordinate systems and spatio-temporal differences [24, 19, 32]. See [17] for a detailed description of some the discrepancies related to spatial data processing.

Next, 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.

Finally, 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 increases. Composability relates to requirements for incremental design and construction of interoperation. These issues are particularly important in dynamic environment like WWW where the set of sites that may cooperate to process a task may vary in terms both of 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

The research addressing the above issues is mostly concerned with the integration and the interoperation of traditional information systems such as databases, knowledge based, or file based systems and has identified three major approaches. The database federation approach uses schema integration techniques to reconcile discrepancies among component systems [3, 26]. Two types of federation have been distinguished. Tightly coupled federations include a *global federated schema* that encompasses all participant systems while loosely coupled federations contain *local federated schemas* that combine information from subsets of the participants. The second approach is the multi-database 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 to combine information from different sources [31, 27, 22].

Recently, concern over GIS interoperability has been the focus of several investigations [5, 23, 13, 30, 24, 19, 18, 20]. Ken Gardels [13] 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 and resources in a network of systems.

The OpenGIS consortium has defined a generic framework and guidelines for extending classical open systems principles to GIS. The goals are to allow sharing of data, resources and system services among GIS applications; to facilitate exchange of information 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 the process of abstracting from real world objects [9] [13]. 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 [8]. Finally, the OpenGIS service model defines functions for assembling spatial objects and building complex spatial applications. Agnès Voisard et al. [30] propose a multi layer decomposition approach based on the above guidelines. It consists of four levels including application, abstract services, concrete services and data access levels. Their methodology is primarily intended for designing extensible GIS which allow the combination of different subsystems and services. However their solution can provide a basis for interoperating multiple systems.

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

Platform level interoperability 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 metadata description of available information sources. 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 file formats and invoke the appropriate converters on the transferred files. For example, the GeoWeb [25] 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. In GIS-WWW Gateway project [6] users can access different GIS one at a 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.

Syntactic level interoperability provides functionalities and tools to define persistent and uniform views over multiple heterogeneous spatial data sources. Access to remote data sets is done via either a high level language or an application interface. Typically, there is no support for the unification of component systems or for reconciling semantic differences. Some solutions try to unify the structure by using common data model or exchange format. Other solutions, which are comparable to the multi-database language approach, allow users to connect to remote GIS to submit queries using their own language.

The OGDI [7] project uses the Transient Data Model, which is derived from the DIGEST [11] model, to allow users to access spatial data through an API developed in C. In [29], Včkovski define the Virtual Data Set model to handle field data type (raster). A VDS encapsulates the field data behavior and its original representation in an object. From a user's point of view a VDS is visible through a standard interface which provides access to the original data. Methods are a persistent part of a VDS interface whereas the values are virtual in the sense that they are derived on demand. A VDS is also able to produce various views of the field depending on the data requirement of potential applications. This approach supports a common interface implemented in Java for accessing distributed data. The GEO2DIS project [14] allow users to query the global system by using OQL-like queries. With a client software, the user first queries a catalog meta-data. Then, the client sends GeOQL queries to a server that translated it to the local GIS hosting the data.

Application level interoperability is concerned with providing seamless system interoperation in which users can access multiple GIS as if they were centralized or integrated spatial system without having to worry about data models, data location, or the semantics associated with the data. Three major approaches can be distinguished:

a) 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. [10] present an overview of database integration schemes applied to spatial databases. They discuss techniques for identifying interschema correspondence, and conflicts that may arise when different assumptions (different scale, generalizations, etc.) are used in the design of spatial databases. Others propose spatial data model or data transformation techniques that can be used to construct integrated schemas [28, 8].

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

b) Schema mediation approach in the GIS realm has been the focus of several projects. Solutions which are based on the wrapper/mediators architecture, aim at extending many functionalities including common data models to incorporate spatial data types. Amann in [2] describes 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_2, Postgres and mSQL. The OASIS project is based on mediation and 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 [23] which is based on the OpenGIS specifications. The 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 [12] project extends the schema mediation approach by adding semantic information represented in meta-data catalog to facilitate the discovery of information. The user queries are based on a profile which is constructed incrementally by browsing meta-data.

c) The context mediation approach is explored by Bishr in [4, 5]. The SEMWEB project is based on an explicit representation of contextual information which is not described by a schema. It provides a representation of the semantics of spatial data 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 translators which enable users to query remote objects without knowing their semantics, localization or representation.

3 An overview of HORUS

This section presents an overview of the ISIS project. In our view the objectives of interoperability are to define a set of functionalities and tools to allow a process (a user query or an application) on a given site: 1) to discover information from other systems, 2) to interpret the semantics associated with remote information in terms of the semantics of its local context, and 3) to transform and retrieve spatial data object from remote sources.

We present two key aspects of ISIS. First, the semantic brokering defines the functional features of ISIS. Next, we describe the architecture HORUS. Distributed object paradigm is used to carry out the dynamic mediation approach and to execute the various mediation steps. The abstractions defined by the semantic brokering framework are embodied in the multi-level data model AMUN which is presented in section 4.

3.1 The semantic brokering approach

Figure 1 depicts the brokering process which may help to meet some of the requirements of GIS interoperability. It consists of components which are grouped in three main levels.

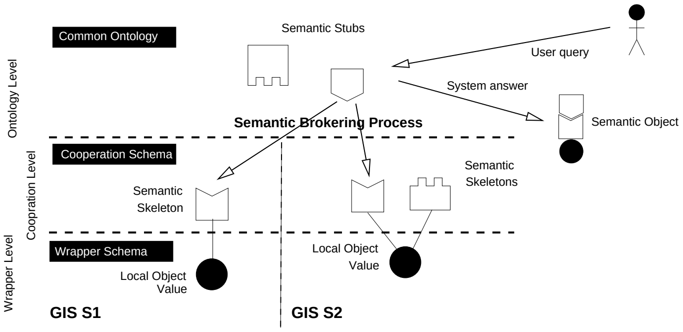

Figure 1: Semantic Brokering Process

- The top level, called ontology level, is used to define the semantics of an application domain and to resolve semantic conflicts. It includes a set of semantic stubs organized as a lattice of classes that represents common concepts.

- The middle level, called cooperation level, provides services to facilitate semantic resolution and query processing. These services embody the dynamic aspect of the cooperative system, including information source discovery based on the semantics of user queries, information brokering and query execution. It includes sets of cooperative schemas that act as a mediator between a site and other data sources. A cooperative schema contains a set of triple objects which contains: 1) a semantic skeleton used to represent the semantic of the object in term of a semantic stub, 2) a reference to local object and 3) context transformation that link local object to their semantic skeleton (semantic representation).

- 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 an export schema described with AMUN's concepts.

The semantic brokering process allows users to retrieve objects matching from the semantics of their queries, i.e. objects which semantic stub and skeleton are equivalent (specialization or generalization).

While the semantic brokering process is architecture oriented the semantic mediation approach is the theroritical basis of ISIS. The semantic mediation includes a reference context which is used to describe common mediation classes, cooperation context which define cooperation objects and schema, and context transformations which define translation mappings between local contexts. Figure 2 depicts the semantic meditation principle. Each notion is formalized in the AMUN model which is presented in section 4.

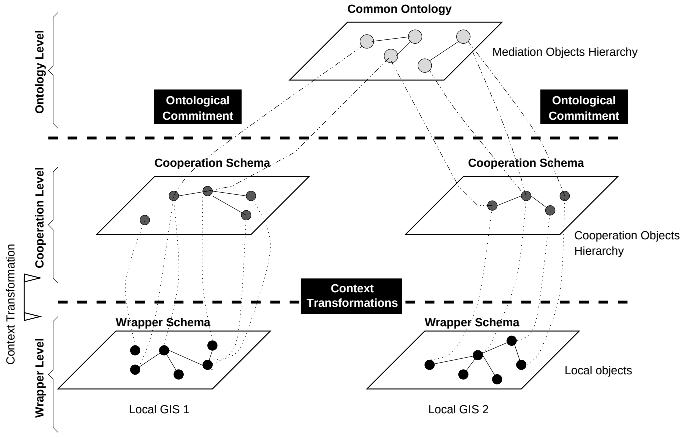

Figure 2: ISIS Semantic Mediation Principle

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 from the underlying syntactic representations of the 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) or description logic model. In our approach, the ontology is viewed as a hierarchy of mediation classes described in the 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 model through which local sites define cooperation classes by selectively *agreeing on the descriptions and the semantics 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 the different roles played by an object. Thus, a semantic 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 described below. In our approach, cooperative classes are defined by specifying *ontological agreements* on common ontological concepts (mediation class). Ontological agreements play a key role in query processing. They are used to bind informations sources that can cooperate on a user query at execution time. An ontological concept can be partially agreed on 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.

Context transformation is used to relate cooperate contexts, which contain the ontological objects accepted by a site, to the 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.

3.2 The ISIS's architecture

As depicted in figure 3, the architecture of ISIS consists of the following types of services: wrapper, cooperation service, ontology server, global query processor, semantic router and user interface. Each service is described by its general functionality in the system, the services it provides, the data and knowledge it owns.

Wrapper service: it is used to encapsulate a local GIS in a generic spatial object server which has the capability of processing data retrieval request concerning the local GIS. The main role of a wrapper is to process OQL queries originating from its corresponding cooperation service. This consists of several steps: 1) translation of the OQL query/subquery to target local query/subquery language, 2) execution of the target query on a local GIS and 3) forwarding local results to cooperation service. To hide schematic heterogeneities, local schema associated with wrapper are represented by AMUN objects using the core concepts.

Cooperation service: it coordinates high-level query processing. It has knowledge about only one local GIS. This knowledge constitutes a cooperation schema (cooperation objects) which describes the semantics of objects contained in the wrapper schema. In order to create it, the cooperation services send messages to the semantic router. A cooperation service processes queries initiated by itself or sub-queries submitted by other cooperation agents. The queries are translated from the cooperation context to the local context by using definitions of cooperation objects to rewrite the query in term of local objects. The translated query is sent to the local wrapper to retrieve data from the local GIS. The reverse query transformation is performed when a cooperation service receive queries from the local users.

Ontology server: it allows communication among different type of services. In order to exchange queries without operating on global schema, all the services use a common ontology which provides mutual understanding of the concepts used in submitted queries. The link between the ontological concepts and the cooperation objects are created by ontological agreements.

Semantic router: its goal is to provide to a cooperation service the identity and location of other cooperation services which can participate in the execution of a query. It stores ontological agreements which are represented by semantic stubs containing the name of the agreed concept, its structure and the name of the cooperation service which can provide objects corresponding to the semantics defined in the stub. To execute the query, the initiating cooperation service sender must contact each cooperation involved in the execution to obtain the definition of cooperation objects.

Global query processor: it takes as an input a query originating from a cooperation service expressed in terms of ontological objects schema and uses ontological agreements to: 1) identify relevant information sources and to 2) create an execution plan which is sent back to the cooperation service.

User interface: it is an interface between a user and a cooperation service. Its role is to receive a query from a user, send the query to cooperation service and to deliver query result to a user. Each user interface is connected to only one cooperation service. The user interacts directly with only one interface even if query processing step required more than one cooperation service.

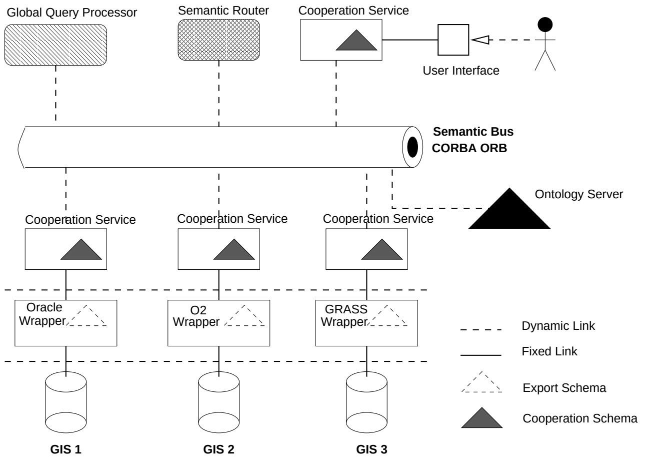

Figure 3: ISIS Mediation Architecture

The architecture of ISIS is based on CORBA and Java. An implementation of a prototype is underway. We have realize different types of wrappers (O2, Access, Postgres) which provides basic access to different spatial databases.

4 The AMUN data model

We have pointed out the important role played by semantic considerations in the cooperation of GIS. In this section we present the data model AMUN that can be used to represent information at both wrapper, cooperation and ontology 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.

To illustrate our approach and the different concepts defined in AMUN, we will use the following example. It consists of three spatial databases S_1, S_2 and S_3 that model information on three different sites.

Example :

• S_1 is an OO GIS which records information on parcels and farm workers for a farm land application. The textual information of interest are parcels number, farmer name, type of crop (culture). Spatial information are used to represent the shape of a parcel which is modeled as a polygon. For farm worker entity only textual information is stored. They represent first name, last name, address birth date, sex and status (full-time, seasoned worker, or farmer). Some of the farm workers are farmers and exploit a farming. GIS S_1 uses an ODMG-ODL like model, the interfaces for objects classes `Parcel` and `FarmWorker` are represented below.

```
interface Parcel{               interface FarmWorker{
 attribute                        attribute
   long Parcel#;                   string FirstName;
   FarmWorker Farmer;              string LastName;
   string CropType;                tuple Address
    polygon Geo;                    [string City,
                                     string Street];
 methods                           date BirthDate;
  real Surface(void);              string Status;
  void WFarmerName(string Name);   char Sex;
  string RFarmerName(void);... }  methods
                                   short Age(void); ...}
```

• S_2 is a government agency GIS containing cadastral information based on the concept of traditional European cadastre. The system is used for land registration and land management. It contains two fundamental elements, cadastral parcels and owners. The following information is recorded: parcel number (unique cadastral parcel identifier), owner name, address, parcel type, shape and surface. GIS S_2 uses a relational data model. Table `Parcel` is shown below: It contains two external references, the attribute `Type` references the `ParcelType` table which is used to represent the different types of parcels (plant, house, crop, etc...); and the attribute `Shape` models the geometry of the parcel as a reference to a polygon identifier.

```
table Parcel(CPI int, OwnerName char[50],
             Address char[200], Type int,
             Shape int, Surface float);
```

• S_3 is an OO GIS of a pollution control agency. Its goal is to better identify water quality problem. It works cooperatively with the different communities (S_1 and S_2 in our example) to develop polluant reduction strategies, to identify type of polluant in relation to crop type and industrial plant and to focus pollution sources by using values collected by different station in watershed basin. Each station has an identifier and is associated to a set of measures which are couples of values consisting of a date and a concentration of nitrate. GIS S_3 use an object data model to represent informations concerning rivers and stations.

```
class River{            class Station{
 public :                 public :
  string Name;             string Id;
  list<Line> Geometry;     Measure: list
}                          <tuple (int Value, date Date)>;
                           Position:
                            tuple (real x, real y);}
```

■

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 [8, 9]. OpenGIS spatial types are described as Well Known Structures which are rigorously defined data types in terms of a sequence of coordinates. Furthermore, OpenGIS defines two basic geodata types: features and coverage. A feature type is used to represent real world entities, and a coverage type represents associations 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, we only consider geometric data types (feature types). Figure 4 shows the hierarchy of spatial data type used in the AMUN data model. `Geometry` which is the highest spatial type in the hierarchy represents general geometric information. `CoordinateGeometry` is a subtype of `Geometry`. It is used to model spatial objects that contain coordinate information. The lowest level of the hierarchy contains the basic spatial data types: `Point`, `LineString`, `Polygon`, ...

4.1.2 Core concepts

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

Figure 4: Type hierarchy of AMUN

thematic or spatial types can be created by the usual set or tuple constructors. The values of attributes can be simple values, complex values or references to other objects. We denote the set of all the types by $\mathcal{T}$ and the set of all the objects by $\mathcal{O}$. An object $o \in \mathcal{O}$ is formally defined by the 3-tuple $o =< oid, Val, MethList >$ 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 is 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 $c =< Name, AttList, MethList >$ where $Name$ is the name of c, $AttList$ and $MethList$ are respectively the list of the attributes and methods belonging to c. The instance relationship defines for each object o the class of which o is a member. The function $pop(c)$ defines for each class c, 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' and $AttList(c)$ is contained in $AttList(c')$.

Example:

In the example GIS S_1 above, the entities farm worker and parcel can be represented at the wrapper level by the following classes: class `Parcel` contains the special spatial type Geo.

```
<Parcel,                         <FarmWorker,
  Parcel#: integer,                FirstName: string,
  Farmer: FarmWorker,              LastName: string,
  CropType: string,                Address: [City: string,
  Geo: POLYGON,                        Street: string],
 { Surface(): real,                BirthDate: date,
  WriteOwnerName(Name: string),    Status: string,
  ReadOwnerName(): string ... }>   Sex: char,
                                   { Age():integer ...}>
```

■

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

Let $\mathcal{OP}$ denote the set of operations. It includes the following operations which can be used for creating virtual classes: Select, Extend, Project, Union, and Join.

The derivation of virtual classes can be carried out using 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 satisfying 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(c_i) is used to merge the population of different classes c_i. The derived class has only the common attributes and methods of the source classes.

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

Example:

The following examples define two virtual classes at the wrapper level in S_2. First of all the local schema of S_2 is translated in AMUN and take place in the wrapper.

```
<Parcel,
  CPI: integer, OwnerName: string, Address: string,
  Type: string, Geo: POLYGON, Surface: real>
```

• a virtual class `LandOwner` can be defined from the class `Parcel`. The `LandOwner` class has the attribute `OwnerName` and `Address`.

```
<LandOwner, OwnerName: string, Address: string>
```

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

```
LandOwner=Project(Parcel,{OwnerName, Address}, {})
```

• a virtual class `SmallParcel` is defined to model parcels which have a surface less than 50 acres. `SmallParcel` is a specialization of `Parcel`. They share 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,Parcel.Surface<50)
```

■

4.2 Ontology and Cooperation layers

The cooperation layer is devoted to the resolution of semantic discrepancies among heterogeneous GIS. To achieve coordination between different schemas with heterogeneous semantics, we introduce the concept of context which can be used to express semantic information contained in schemas, and to record the assumptions under which schemas are designed. Three types of contexts are defined in the ISIS architecture: 1) reference contexts model common semantic 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 three types of descriptions: static properties, behavior (list of methods) and semantics. The semantics 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` of a `Parcel` can be precisely 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, or to express that the nitrate concentration is expressed in milligrams per litre.
- 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 $\mathcal{MC}$ be the set of all the mediation classes. Let $\mathcal{MN}$ be the set of method names, and $\mathcal{PN}$ the set of parameter names. A class $mc \in \mathcal{MC}$ is a tuple $mc =< Name, AttList, MethList >$ where:

- $Name(mc)$ is the name of the mediation class cm.
- $AttList(mc) = \{A_i, A_i : T_{A_i} \vee A_i : T_{A_i}$ **domval** $d \vee A_i : T_{A_i}(B_{j_i} : d)\}$, $i = 1..n$, $j_i = 1..m_i$, $A_i \in Att_{name}$, $B_{j_i} \in Att_{name}$, $T_{A_i} \in T$. **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}.
- $MethList(mc) = \{M_j, M_j : \{p_{k_j} : T_{k_j}\} \vee M_j : \{p_{k_j} : T_{k_j}\} : T_{res}\}$, $j = 1..m$, $k_j = 1..q_j$, $M_j \in \mathcal{MN}$, $T_{k_j} \in \mathcal{T}$, $T_{res} \in \mathcal{T}$, $p_{k_j} \in \mathcal{PN}$ is the set of methods belonging to mc.
- $pop(mc)$ function compute the extent of mc
- $mr(mc), mr : mc \rightarrow \{cc_1, cc_2, \ldots, cc_n\}$ function gives the list of cooperation classes which define a role defined as mc

□

Note that a mediation class defines a virtual class since it has no actual instance. It 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. Virtual extensions consisting of instances defined at the local information sources can be associated with a mediation class. When needed, these extension can be computed by merging the extensions of the corresponding cooperation classes. Moreover, the resulting calculated extensions are composed of different information sources, and thus may have different structures.

Example:

The following mediation classes describe a part of an agricultural ontology: 1) a class `Person` to describe a person by name, birth name, sex, birth date. It includes a method age. A person is a male or a female so the domain associated with attribute sex has a type string restricted to an enumerated set that contains two possible values and 2) a class `Parcel` to describe cadastral parcels with parcel identifier, owner and address.

```
<Person,                                 <Parcel,
  Name: string,                             PID: integer,
  BirthName: string,                        Owner: Person,
  Sex: string                               Address: string,
    domval {"male","female"},               Surface: integer,
  BDate: date,{Age():integer}>              Geo: POLYGON>
```

Two sub-classes of `Parcel` are defined: `CerealParcel` is a specialization of `Parcel`, it inherits all the attributes of `Parcel` and add a crop type, `IndustrialParcel` is a `Parcel` and add an attribute to record type of plant.

```
<CerealParcel                          <IndustrialParcel,
   PID: integer,                          PID: integer,
   Owner: Person,                         Owner: Person,
   Address: string,                       Address: string,
   Surface: integer,                      Surface: integer,
   CropType: string>                      PlantType: string>
```

■

4.2.2 Cooperation context

On a site, a cooperation context acts as a mediator between a 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, participant GIS need a set of commonly understood objects which are used 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 5 shows a mediation class and the corresponding roles which are defined in cooperation classes $CC_1, CC_2, \ldots, CC_n$. A mediation role is defined as follows: 1) describe the subset of attributes (of a mediation class) on which the local site agrees, 2) use 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 doesn't create or delete any objects. Formally, a mediation role is defined as follows.

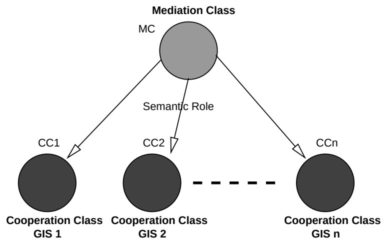

Figure 5: Semantic roles played by a mediation class

Definition 2 *(Mediation Role)*
Let $\mathcal{MR}$ be the set of all the mediation role. A mediation role $mr \in \mathcal{MR}$ is a tuple $mr =< mc, AttList, MethList, Q >$ where:

- $mc \in \mathcal{MT}$ is a mediation class whose interpretation is mr

- $AttList(mr) \subseteq AttList(mc)$ is a subset of attributes of mc built by using algebraic operations Select and Project
- $MethList(mr) \subseteq MethList(mc)$ is a subset of mc methods
- 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.

□

mc is the role player of mr. Role players can be shared by objects of different types. For example both `LandOwner` and `Farmer` defined roles played by the mediation class `Person`.

Example:

This example shows two mediation roles `LandOwner` and `FarmWorker` corresponding to `Person`. They express that `LandOwner` in S_2 and `Farmworker` in S_1 have the semantics of `Person` described in the ontology.

```
r1=<Person, Name: string>   /*LandOwner role on S2*/

r2=<Person,                 /*FarmWorker role on S1*/
 Name: string, BirthName: string,
 Sex: string, BirthDate: date,
 {Age():integer},
 DateN>"01/01/1928">
```

They model two different local interpretations of the mediation class person in the GIS S1. For `r1` role corresponding to `LandOwner`, only `Name` attribute can be given. For `r2` role corresponding to `FarmWorker`, all the information in mediation class is supplied. Furthermore, the qualification associated with `DateN` is used to state the fact that age of the farmers in the GIS S_1 are less than 70. ■

B. Cooperation objects and classes

A cooperation class incorporates a mediation role, a view defined by a virtual class which links the mediation role to local objects, and context transformations which are used to transform object description from one cooperative context to another (see figure 6). 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. Cooperative 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.

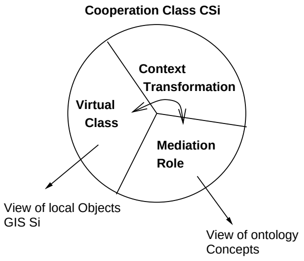

Figure 6: A cooperation class of a site Si

Definitions 3 and 4 show respectively a formal definition of cooperation object and cooperation class.

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:
$co =< oid, cc, val, MethList, context_o >$ where :

- $oid \in OID$ is the oid of co
- $cc \in \mathcal{CC}$ is the cooperation class of which co is an instance
- $val \in \mathbf{D}$ is the value of the cooperation objet, val is also named local value of co
- $MethList(co)$ is the set of methods binded to co
- $context_o(co) = \{< val(mr), \mathcal{CTF} >\}$ where $mr \in \mathcal{MR}$ is a mediation role, $val(mr)$ is its value and $\mathcal{CTF}$ is the set of context transformations which convert a local value of co to its value for mr.

□

Definition 4 *(Cooperation Class)*
A cooperation class $cc \in \mathcal{CC}$ is a tuple
$cc =< Name, AttList, MethList, cv, context_C >$ 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
- $context_C = \{< mr, \mathcal{CTF} >\}$ defines the context of cc. It is a set of tuples where mc is a mediation class such as $\exists mr \in \mathcal{MR} mr.c = mc, \mathcal{CTF}$ is a set of context transformations.
- cv is a virtual class encapsulated by cc such that $AttList(cc) \subseteq AttList(cv)$ and $AttList(mr.c) \subseteq AttList(cv)$ and $MethList(cc) \subseteq MethList(cv)$ and $MethList(mr.c) \subseteq MethList(cv)$.

□

C. Context transformations

A context transformation is a function which establishes a mapping from one value domain (local) to another (cooperation). A context transformation is associated with each mediation role to allow objects to migrate from a local context to a cooperation context. Figure 7 shows a context conversion between two GIS. A semantic translation process allows objects defined in a 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.

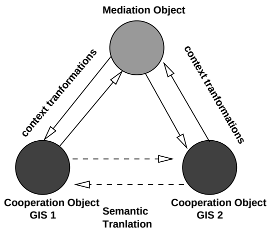

Figure 7: Context Transformations

Definition 5 *(Context Transformation)*
$\mathcal{CTF}$ is the set of the context transformation: $\mathcal{CTF} = \{\uparrow_o^{mr}\} \cup \{\downarrow_o^{mr}\}$. They are defined by:
- the function type $\uparrow_o^{mr}$, is used to translate a local value of an object into its value for the mediation role mr.

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

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

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

□

Example:

The following examples depict context transformations between the mediation role `Person` and the class `FarmWorker` of S_1.

```
↑_o^{mr}Person.Name(FarmWorker.FirstName)={
                   return(FarmWorker.FirstName)}
↓_o^{mr}Farmworker.FirstName(Person.Name)={
                   return(Person.Name)}
```

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

```
↑_o^{mr}Person.Sex(FarmWorker.Sex)=
                   {if FarmWorker.Sex="M" return("Male")
                     else return("Female")}
↓_o^{mr}FarmWorker.Sex(Person.Sex)=
                   {if Person.Sex="Male" return("M")
                     else return("Female")}
```

The cooperation class `CCFarmWorker` encapsulates both the virtual class `CVFarmworker` and its context (mediation role, context transformations and qualification). The virtual class `CVFarmWorker` and the cooperation class `CCFarmWorker` are defined below :

```
<CVFarmWorker, FirstName: string,
   LastName: string, Sex: char,
   Address: [city: string, street: string],
   BirthDate: date, Status: string,
  {Age(): integer...}>

<CCFarmWorker, FirstName: string,
   LastName: string, Sex: char,
   Address: [city: string, street: string],
   BirthDate: date, Status: string,
  {Age(): integer...}, CVFarmer, {<
  <Person, Name: string, BirthName: string,
    Sex: string, BDate: date, {...},DateN>"01/01/1928">,<...
   ↑_o^{mr}Person.Surname(CVFarmeworker.FirstName)={
                   return(CVFarmWorker.FirstName)}
   ↓_o^{mr}CVFarmWorker.FirstName(Person.Name)={
                   return(Person.Name)}...>>}>
```

■

4.3 Mediation example

Suppose that in S_3 a station gives a high nitrate concentration level. In order to investigate the potential pollution source, S_3 tries to contact the persons which are responsible (owner, farmer or manager) for the parcel which contains the station.

To satisfy this query S_3, has to define two cooperation classes related to the concept of person and parcel. This allows users to query `Person` or `Parcel` as they are recorded in S_3.

For `Person` class only `Name` is relevant, so the cooperation class `CCPerson` in S_3 just contains the attribute `Name`. Users query `CCPerson` through a virtual class `CVPerson`. `CCPerson` is defined as follows:

```
<CCPerson, Name: string,
  CVPerson, {<Person, Name: string, {...}>,<...
   ↑_o^{mr}Person.Name(CVPerson.Name)={
                   return(CVPerson.Name)}
   ↓_o^{mr}CVPerson.Name(Person.Name)={
                   return(Person.Name)}...>}>
```

For `Parcel`, information of interest are `Owner`, `Address` and `Geo`. Cooperation class `CCParcel` and virtual class `CVParcel` are given below:

```
<CCParcel, Name: string,
  CVParcel, {<
  <Parcel, PID: integer, Owner: Person,
    Address: string, Geo: POLYGON {...}>,<...
   ..>}>
```

The query expressed by user is the following:

```
Select CVPerson.Name, CVParcel.Address
   From CVPerson, CVParcel, Station
   Where Station.Position in CVParcel.Geo
     and CVParcel.Owner.Name=CVPerson.Name
```

The query processing step is as follows: 1) CA sends the query to QPA 2) QPA sends a request to SRA to identify sites relevant to the query i.e. sites that agree `Person` and `Parcel` concepts 3) QPA decomposes the query and sends sub-queries to S_1 and S_2 4) each CA (CA$_{S_1}$ and CA$_{S_2}$) uses cooperation classes and context transformation to return `Parcel` and `Person` in the common context, 5) QPA joins CA$_{S_1}$ and CA$_{S_2}$ answers and sends them back to CA$_{S_3}$ which uses `CCPerson` and `CCParcel` to produce results in the user context.

5 Conclusion

In this paper, we have focused on fundamental issues relevant to the design of interoperable GIS. We argue that resolution of semantic differences among various systems must be based on the representation of context informations that can be used to capture the semantics of various systems. We introduce a distributed spatial object model AMUN that has the following main characteristics:

- definition of a set of concepts to deal with distribution and heterogeneities: virtual classes, mediation class, mediation role, context transformation and cooperation class;
- spatial data types to deal with spatial objects;

We also present a mediation based architecture which extends the distributed object paradigm to allow sharing spatial data using their semantics. It's major components are: 1) wrappers (one per local GIS) to resolve heterogeneity in the cooperative environment, 2) mediators for coordinating cooperative tasks such as services communications or query dispatching. The main advantage of the architecture is that 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 without 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.L. Tan, and S. Huat 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, 1997. ftp://sikkim.cnam.fr/pub/Reports/GISMed.ps.gz.

[3] C. Batini, M. Lenzerini, and S. B. Navathe. A comparitive analysis of methodolgies 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. OGDI: toward interoperability among geospatial databases. *ACM SIGMOD Record*, pages 18–23, 1997.

[8] Open GIS Consortium and The OGIS Technical Committee. The Open GIS abstract specification. Technical report, 1996.

[9] Open GIS Consortium and The OGIS Technical Committee. The Open GIS Guide Part I: Introduction to interoperable geoprocessing. Technical report, 1996.

[10] T. devogele, C. Parent, and S. Spaccapietra. On spatial database integration. *Int. Journal of Geographical Information Science*, 12:335–352, 1998.

[11] Digital Geographic Information Working Group DGIWG. The digital geographic information exchange standard (digest 2.0). Technical report, Directorate of Geomatics Department of National Defence, Canada, June 1997.

[12] 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.

[13] K. Gardels. Opengis and on-line environmental libraries. *ACM SIGMOD Record*, 26:32–38, March 1997.

[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 incorparating gis, the world wide web and CORBA. In National Center For Geographic Information and Analisys (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 geoographic objects. In *Proc. of Parallel and Distributed Comuting 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 multi-database continuity and indexing: a step toward seamless gis data interoperability. *Int. Journal of Geographical Information Science*, 12:373–402, 1998.

[20] E. Leclercq, D. Bensliman, 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] S. Nittel, R. Muntz, and E. Mesrobian. geoPOM: a heterogeneous geoscientific persistent object system. In *Statistical and Scientific Database Management (SSDBM)*, 1997.

[24] C. Parent, S. Spaccapietra, and T. Devogele. Conflicts in spatial database integration. In *Proc. of Parallel and Distributed Comuting Systems*, pages 772–778. Dijon, France, 1996.

[25] 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.

[26] A. Sheth and J. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22:183–236, September 1990.

[27] 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.

[28] 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.

[29] A. Včkovski. Java as a software system for distributed and interoperable geoprocessing. In *Proc. of Parallel and Distributed Comuting Systems*, pages 779–783. Dijon, France, 1996.

[30] A. Voisard and H. Schweppe. A multilayer approach to the open gis design problem. In *Second International ACM GIS Workshop*, pages 23–29, 1994.

[31] G. Wiederhod. Mediators in the architecture of future information systems. *IEEE Computer Magazine*, 25(3):38–49, March 1992.

[32] M. Worboys and S. Deen. Semantic heterogeneity in distributed geographic databases. *ACM SIGMOD Record*, 20(4):30–34, 1991.