Semantic Integration in MADS Conceptual Model

Anastasiya Sotnykova, Sophie Monties, and Stefano Spaccapietra

Swiss Federal Institute of Technology in Lausanne, Database Laboratory,
1015 Lausanne, Switzerland
{Anastasiya.Sotnykova,Sophie.Monties,Stefano.Spaccapietra}@epfl.ch

Abstract. Our vision of a viable way for transparent and meaningful processing of heterogeneous spatio-temporal data is to put data semantics in the foundation of an integration process. We present and correlate means of integration as components of the mediation level of an interoperable system. For our domain of interest we present MADS domain ontologies and MADS conceptual data model dedicated to modeling of spatio-temporal data. Using as example two MADS schemas we outline an integration methodology based on semantic interschema correspondence assertions and integration goals.

1 Introduction

The interoperability problem arises in heterogeneous systems where different data resources coexist and there is a need for meaningful information sharing in the system. The heterogeneity of the data can be originated by semantic, syntactic, and structural differences of the data sources. One of the demonstrative realms of diversity of data representation is the spatio-temporal domain. In spatio-temporal domain the same objects can be represented (and are represented) from multiple and greatly diverse points of view. For example, a building can be represented from four different points of view as shown in Table 1.

In contrast to thematic data representation, spatio-temporal data heterogeneity largely lies in the semantic of the data. Thus, it is definitely insufficient to establish a correspondence between attribute value domains, for example. An adequate amount of integration work has to be done till we can establish a correspondence on the attribute domain level. Interoperable system that operates spatio-temporal data should be based, first and foremost, on the semantic information as the core of such a system. As it is illustrated by example in Table 1, to propose rules by which it can be inferred that the two or more different data representations portray the same object from the real world is a challenge. Such rules or correspondence assertions is a viable way to express the fact of common population, spatial, and/or temporal features of objects from different applications. Derivation of semantically driven assertions is feasible if this process is founded on an equally semantically expressive data model for the application domain. This implies that the application data should be remodeled

Table 1. How a 'building' object can be represented.

Purpose of representation	*User*
Architectural style and fitting in the neighborhood environment	Architect department of a city administration
Robustness of the construction of the building and the materials it is built of	Rescue crew of the city
Condition of the building and suitability for living in it	Renovation construction company
Location and dimensions of the building	Cadastral department of the city administration

or pre-integrated at the conceptual level[1] in a semantically rich model. Such a model is called Canonical Data Model (CDM). A CDM should have a minimal number of concepts while being sufficient to capture the semantics of the application domain and tasks to which it is dedicated. In the paper we give our view of the appropriate architecture for an interoperable system, CDM, correspondence assertions expressions, and integration methodology for spatio-temporal domain.

We begin our paper by presenting a generic view on the interoperable systems and proceeding by refining the scope of possible architectures to agent-based and mediator-based systems as mostly wide approved by the research community. In Sect. 2 we discuss main features that can be accomplished within each architecture and point the one which is more suitable for our domain of interest. Within this architecture in Sect. 3.1 we define the system component for which we contribute some proposals of our own. Section 3.2 presents the conceptual data model which we use as the CDM for pre-integration of the spatio-temporal data. A provisional integration methodology is presented in Sect. 4 and illustrated by an example introduced in Sect. 3.3. Section 5 concludes our paper.

2 Interoperable System Components

Generally, an interoperable system consists of three main components as shown in Fig. 1. At the foundation level there are heterogeneous legacy data sources. The mediation level supports exchange of queries and results between legacy data sources and applications. At the application level the interaction with the users is carried out [13].

Without the 'Value-Added Services', the structure presented would be an ordinary information system architecture designed for a particular group of users operating a specific set of data sources. Nowadays, when modern information systems increasingly address the information and knowledge acquisition issues over

[1] as the implementation independent level

heterogeneous data sources [9, 19], this is no longer an answer for an information system architecture. An information system with an intermediate level between 'USERS' and 'SOURCES' levels is called mediated. The mediation level that provides the users with services based on the data collected and operated previously with other purposes, and within other information systems, would allow for defining such a system as interoperable. In the literature it can be found many different implementations of the mediation level [18, 1, 3, 5, 2]. Among these, the components distilled by the practice are the following:

- *application ontology* - a dictionary containing all the concepts and their hierarchy for the application domain;
- *agents* - intelligent components that can serve different purposes in the system, for instance, location of the data sources in a distributed system, matching user requests with the services available;
- *translators* - translate user queries to a CDM;
- *wrappers* - translate the heterogeneous source data to a CDM;
- *integrators* - perform integration of heterogeneous data sources based for example on an ontology, or a CDM;
- *mediator* - a complex component which provides transparent access and processing at the application level over a set of heterogeneous source data.

The mediation level can incorporate a set of different components. The choosing of these components and functionality of the mediation level is dictated by the intended objective of the system. In the sequel we will present two mostly distant in the functionality system architectures: mediator-based [1] and agent-based [8].

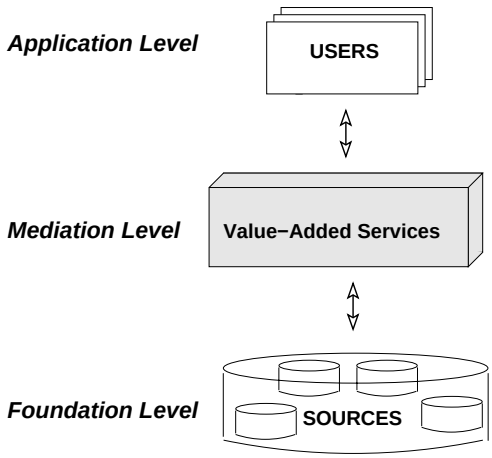


Fig. 1. Generic Structure of an Interoperable System.

Mediator-based systems. In a mediator-based system it is assumed that there is a component to which all the user's queries are addressed, where these queries are processed, and where the results of these queries are sent. This component plays the mediation role between the users and data sources and maintains the global vision of the system [22]. A mediator-based system that we have chosen as the illustrative example is presented in [1].

Figure 2 shows a simplified architecture of the system. As the basis for data integration, a CDM was used. The authors have chosen the object-oriented data model whose capabilities in modeling semantics and relationships were suitable for the application area.

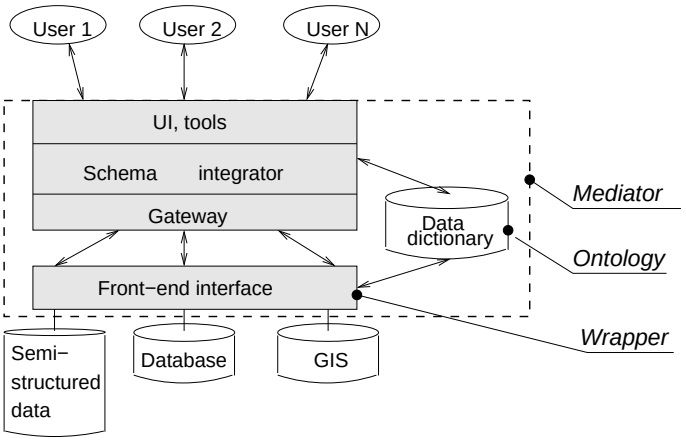


Fig. 2. Mediator-based approach.

The local schemas of component databases are translated into the CDM and are enriched semantically if necessary. The federated schema is a schema constructed in CDM based on the user specifications on the subset of data of their interest. Thus, the users view the system as a single database containing the data they requested. User queries are directed to the mediator component of the system where the queries are decomposed and then translated to the local schema query languages. Although the technique described in the paper suits the application requirements, the authors do not address issues such as semantic conflicts resolution, integrity constraints management. In addition the disadvantage of the system is that the component databases are not operable locally and that the data sources updates are done as well globally. Presenting the system capabilities the authors mention that:

> *. . . the schema integrator provides facilities for integrating the schema exported from the component databases into the federated schema. It needs to generate mapping between the exported and federated schemas*

and must have a reasonable capability for detecting conflicts between data...

However, no real example of the schema integrator functionality is given in the paper. The authors propose the use of a mapping table[2] for object representations matching.

Agent-based systems. As an example of an agent-based architecture we consider the InfoSleuth system presented in [8]. InfoSleuth is a distributed system where the data sources and the users reside on different sites and are connected by sets of different agents. System agents communicate on the base of system ontology which is the only global component of an agent based system. *Ontology* is a specification of how to represent the objects, concepts and other entities that are assumed to exist in some domain of interest and the relationships that hold among them [6]. Ontology does not represent a global structural vision of the system data sources but only the set of terms the system is aware of.

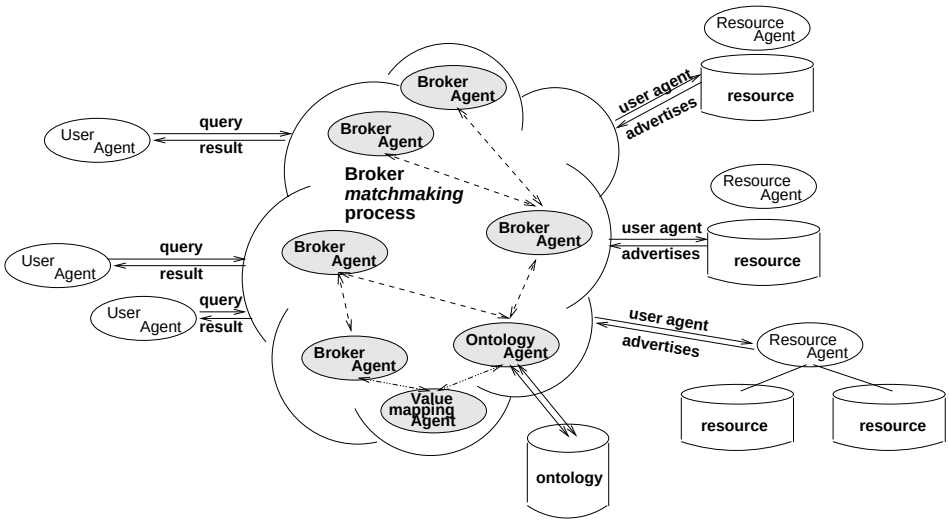


Fig. 3. Agent-based approach.

In an agent system the main three agent types can be pointed out [8]:

- *User* agent - maintains the user state and provides the system interface that enables the user to communicate with the system independently of the user location.
- *Resource* agents - translate queries and data stored in some external data repository between their local forms and their representation (data model) in the system.

[2] which can be seen as a simplified ontology

- *Broker* agents - match requests for services from user agents with resource agents that can provide them.
- *Resource*,*Ontology*, and *Value Mapping* agents - are the means of interoperability of the system.

All the types of heterogeneity, i.e., structural, syntactic and semantic, are solved by resource agents and value mapping agents in InfoSleuth based on the developed ontology. The value mapping agents map query terms to and from the canonical value domain which is defined by the system ontology. The canonical value domain reduces heterogeneity only in terms of allowed attribute values but not in terms of data representation. Users query and view data in whichever value domain they prefer, and their user agents perform the value mapping necessary to communicate with other agents in the canonical value domain. To perform a value mapping a user agent contact a value mapping agent. Thus, all the operations related to data interoperability are done through consultation with the ontology of the system. Referring to Fig. 3, it can be seen, that the functionality of the *user* and *value mapping* agents is similar to that of the *query translator*. On the other hand, a resemblance can be found in *wrapper*'s in Fig. 2 and the *resource* agent functionalities.

The partial knowledge of the available data and its location is maintained by broker agents. The information stored by broker agents is partitioned in a way that the whole set of broker agents 'knows' about all the data available in the system[3]. The system ontology stores the hierarchy of the data the system is aware of. When the user queries the system, it is the broker agent(s) functionality to determine whether the data requested can be found in the system data sources. The user does not have a global view of the system data and does not know whether his/her request can be met. The partition of the knowledge and communication between the broker agents ensures that the user agents and the resource agents are fully connected, e.g., any user can potentially reach any resource.

As it follows from the system description the users are assumed to pose `SELECT`-type queries. This system does not allow `UPDATE`-type queries. The broker agents are oriented towards locating requested data, matching the semantic and syntactic information of the user agent and a data source. Done automatically, the matchmaking process restricts the amount of semantic information that agents can operate. The last observation together with absence of a global vision of the system limits the application area of the agent-based systems.

Comparison. Comparing the approaches presented above, we bear in mind the following characteristics of an intended interoperable system:

- at the foundation level there are heterogeneous spatio-temporal data sources;
- at the application level there are users with their vision of the universe of discourse;

[3] Depending on the system design redundancy may be allowed or even required.

– users expect transparent operations on geodata stored in different formats, with different resolutions, and for different purposes.

Let us first briefly summarize the pivotal characteristics of the two approaches to make our reasoning about their applicability to spatio-temporal domain more clear.

– *Agent-based*
  - system ontology is the only global component in the system,
  - the users do not have a global view of the system,
  - updates are allowed on the local level and may not reflect the system ontology.
– *Mediator-based*
  - a mediator stores the schemas of component databases and the relationships between them or a federated schema of the system depending on the implementation,
  - users have a schematic partial or global view of the system,
  - updates are theoretically allowed from both the global and local levels, but with no clear methodology for updates propagation, the global consistency of the system is an open question.
– *Desired interoperable system characteristics*
  - a global schema is constructed based on semantic and syntactic information, conflicts are resolved at the global level;
  - consistency of the data is ensured during integration process;
  - updates are allowed from the global level as well as from the local, consistency of the component databases is ensured by an updates propagation mechanism.

The agent-based system is hardly a viable way to accomplish such a task. Agents are more oriented towards determining the location of data sources in a distributed system: the data sources available are heterogeneous in the sense that different objects are stored in different locations. Whereas, dealing with the spatio-temporal data it is also likely that the same phenomena would be presented in different ways. Consequently, to establish a relation between different objects, a semantically based methodology should be employed which requires semantically rich integration platform. Semantic information carried by agents in the InfoSleuth system is not sufficient for integration of spatio-temporal object.

Moreover, in the spatio-temporal domain the global vision of the data and data structure are indispensable properties of the system. An attempt to augment the broker agents with more semantic information would lead to an increase in the system response time and therefore to a decline in the system performance. Another potentially incompatible with spatio-temporal integration process feature of the agent-based system architecture is that the matching process is automatic. This advantageous feature of the agent-based architecture is not yet applicable to the spatio-temporal domain. To the best of our knowledge, there is no methodology which would support an automatic matching process for spatio-temporal objects modeled within different applications.

More features that can be adopted in the spatio-temporal domain are borne by the mediator-based approach. The example given in Table 1 suggests that for integration purposes the data sources should be implemented or translated (on the conceptual or physical level) into a model which allow to express diverse semantic aspects of the data objects. In addition there should be an expressive language which allows to define interrelations of spatio-temporal object types. In [16] the authors provide a picture of what are the integration approaches leading towards system interoperability. In Sect. 4 we present our integration methodology in more detail.

One of the common system components in Fig. 2 and Fig. 3 is the system ontology. The notion of ontology in not unambiguously perceived by the database community, whereas ontology plays a key role and for particular implementations is the only mean of integration of the system data sources. In the following section we present a notion of ontology and conceptual models as the next level of abstraction of an application area.

3 Interoperability: Ontologies and Conceptual Models

Guarino in [10] distinguishes several levels of the ontology, as shown in Fig. 4. Top-level ontology is a representation of the 'truth', i.e., the representation of the real world without any inference services in mind. The top-level ontology is the most generic type of ontology where the concepts like space, time, matter, object are presented. The taxonomy of top-level ontologies is the simplest from the structural point of view. The only relation that is used for top-level ontologies is subsumption. Thus, a top-level ontology has a non-cyclic tree structure without multi inheritance. An example of such ontology can be found in [12].

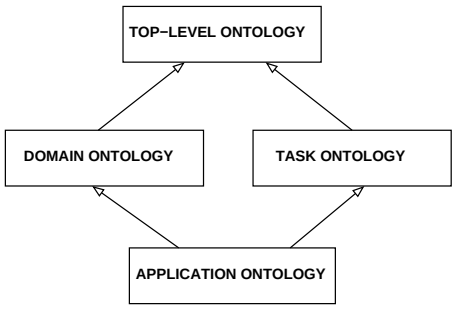


Fig. 4. Kinds of ontology from [10].

The reasoning behind constructing a top-level ontology lies in the four meta-properties borne by the things of the real world. These meta-properties are: *rigidity*, *identity*, *unity*, and *dependence* described in details in [10] and [11]. In brief, identity is related to the problem of distinguishing a specific instance of a

certain class from other instances by means of a characteristic property, which is unique for it. A rigid property is a property that is hold for all the instances of a class. For example, imagine two classes PERSON and STUDENT. From the point of view of rigidity, PERSON is rigid - all the instances of this class are of PERSON type, on the other hand, STUDENT is not rigid - the same individual can be STUDENT in one context, and non-STUDENT in another. From the previous example we could conclude that, rigid classes supply the identity, and non-rigid ones just carry an identity. Unity is related to the problem of distinguishing the parts of an instance from the rest of the world by means of a unifying relation that binds them together. An example identity query would be 'What is this country?' and a unity query would be 'Does this canton belong to this country?'. If existence of an instance of a class implies a necessary existence of an instance in another class, then the former possesses the dependence meta-property. An example would be a CANTON class that implies existence of a COUNTRY class. Ascription of those properties to classes imposes certain constraints on the positional relationship of these classes in the top-level ontology. For example, one of the imposed structural constraints is that a dependent class cannot subsume the class it depends on. The role of the top-level ontology is to formalize the real world in a widely sharable, multidisciplinary way to be used further as the pattern for different domain and task ontologies. For the spatio-temporal domain, the top-level ontology is one of the subjects of research and agreement of the OpenGis consortium [15].

When an ontology contains some domain specific concepts or concepts related to general features of an application we step down the ontology hierarchy. Such an ontology is a subjective, refined representation of the same concepts as in the top-level ontology, but domain and task ontologies already can be used in the integration process.

3.1 Ontology and Conceptual Schemas

Domain and task ontologies contain the classes that are further used in the conceptual schemas. When we start to model the roles of the domain ontology classes while performing certain activity we are at the level of application ontology, the most specific and application dependent type of ontology. The main thread through-passing the structure shown in Fig. 4, is that domain, task, and application ontologies are structurally compliant with the top-level ontology. In the light of the above presentation, we believe that an ontology notion in each particular utilization should be differentiated and clearly distinguished from the next level of knowledge presentation, namely conceptual modeling.

The objective of conceptual modeling is to represent application data together with the rules of the application domain, in other words, conceptual models allow to represent the user understanding of the universe of discourse. The task of designing a modern information system becomes more complex with the progress of information technology and with the users becoming more demanding for the functionality of the information systems. In such circumstances

conceptual modeling gains in importance, as it is the starting point for understanding of the user needs. The most important properties that a conceptual model should have are: abstraction, non-ambiguity, ease of understanding and verification, and implementation independence [14]. The last property implies that a conceptual model should be expressive enough so that, the same conceptual schema would be valid even when software paradigms were upgraded or replaced. Conceptual schemas being the representation of the user perception of the application domain, would constitute the basis for integration of heterogeneous domain sources in an interoperable system. In the integration approach presented in Sect. 4, we chose a conceptual representation as the starting point for corresponding heterogeneous data sources. The choice of an appropriate conceptual model depends on the completeness of representation allowed by it, formal semantics and simplicity of use and interpretation.

In this section we described two levels of metadata representation that are components of the mediation level of an interoperable system. An ontology is the representation of real world without bearing in mind any application of this representation; conceptual schemas are implementation independent representations of the application area, containing users vision of the application domain. The link between an ontology and a conceptual schema is that the conceptual schema of an application domain should be structurally compliant with the ontology for the same domain. Nevertheless, in our research we base our approach only on domain ontologies and conceptual schemas without making any reference to a top-level ontology as we are not aware of existence of an approved top-level spatio-temporal ontology. In the following section we reason about our choice of the spatio-temporal canonical data model.

3.2 MADS Conceptual Model as a Canonical Data Model

Applications manipulating geodata are difficult to model due to the particularity and complexity of the spatial and temporal components. More facets of real-world entities have to be considered, e.g., location, form, size, time validity; more links are relevant, e.g., spatial, temporal links; several spatial abstraction levels often need to be represented. Thus, modeling spatio-temporal databases requires advanced facilities [20], such as the following.

- Objects with complex structure (e.g., composition/aggregation links, generalization links), at least equivalent to those supported by current object-oriented models. This should achieve full representational power in terms of data structures;
- Alternative geometry features to support both discrete and continuous views of space, representations at different scale/precision, multiple viewpoints from different users;
- Spatial objects with one or several geometries associated to different resolutions or user points of views;
- Temporal objects with complex life-cycles that allow users to create, suspend, reactivate, and eventually delete objects;

- Timestamped attributes that record their past, present, and future values;
- Spatio-temporal concepts for describing moving and deforming objects;
- Explicit relationships to describe structural links as well as spatial (such as adjacency, inclusion, spatial aggregation) and synchronization links (such as before, during). The knowledge of the topological links between real-world entities is an essential requirement for applications.
- Causal relationships describing the causes and effects of changes that happen in the real world.

The model must also allow defining schemas that are readable and easy to understand. A key element for achieving this double objective is the orthogonality of the structural, temporal, and spatial dimensions of the model (and more generally of the concepts of the model). Thus, whatever the concept of the model, e.g., object, relationship, attribute, aggregation, the spatial and temporal dimensions may be associated to it.

In our research we employ MADS conceptual data model [17] as the CDM. MADS stands for Modeling of Application Data with Spatio-temporal features. MADS model was specially designed to fill the niche of conceptual data models for spatio-temporal applications. In [17] the authors analyze different spatio-temporal data models along the axes of expressiveness, simplicity and comprehensiveness, formalism, and user friendliness, making the conclusion that none of the existing models satisfied all the demanding criteria. MADS includes a set of predefined spatial and temporal Abstract Data Types (ADTs) that are used for describing the spatial and temporal extents of the spatio-temporal elements of schemas. Spatiality and temporality may be associated to object and relationship types, aggregation links, and attributes. MADS structural, spatial, and temporal domains are orthogonal meaning that spatial and temporal features can be freely added to any schema designed in MADS.

MADS structural dimension. Structurally MADS is an object+relationship data model. Thus, it allows to represent basic concepts from entity-relationship model, e.g., entity type, relationship type, *IsA* link, as well as more complex structural components, e.g., aggregation links, multi-inheritance, complex attributes. The MADS data structure notation is shown in Fig. 5.

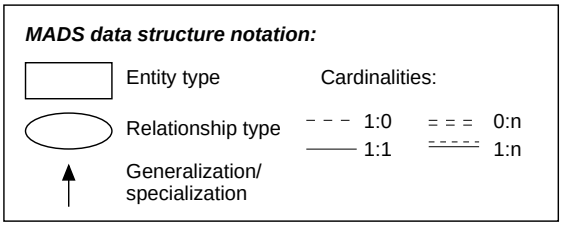


Fig. 5. MADS data structure notation.

MADS spatial dimension. MADS predefined Spatial ADTs (SADTs) are: point, line, oriented line, simple area, simple geo, point set, line set, oriented line set, complex area, complex geo, geo. The spatial domain ontology is shown in Fig. 6. Figure 6 also shows the icons denoting each SADT. The most generic SADT is

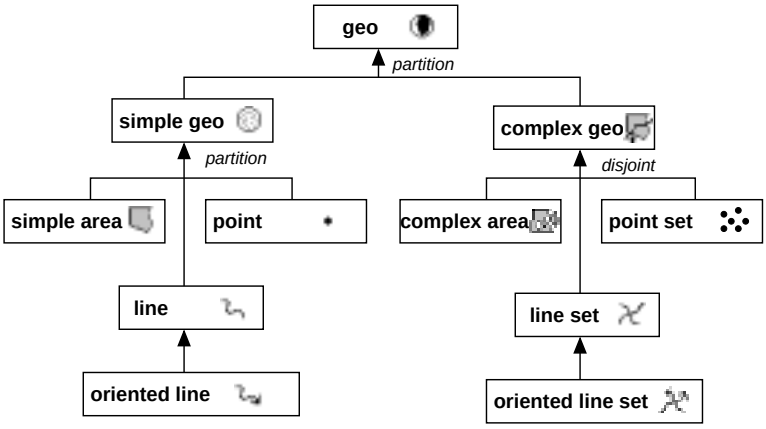


Fig. 6. MADS basic hierarchy of spatial abstract data types.

geo which generalizes the simple-geo and complex-geo SADTs, and whose semantics is 'this element has a spatial extent' without any commitment to a specific SADT. These three SADT are abstract and they are never instantiated. The spatiality of an element can be either defined precisely e.g., point, oriented line, or left undetermined, e.g., geo.

MADS temporal dimension. Temporal ADTs (TADTs) support timestamping, i.e., associating a timeframe to a fact. Timestamping is the traditional way of modeling so-called temporal information. Timestamped attribute values allow expressing when a value was, is, or will be holding in the real world as perceived by the application (valid time) or when it was known in the database (transaction time). Timestamped objects and relationships expresses information on their life cycle: when an object or relationship was created, suspended, reactivated, or deleted. Object and relationship timestamps are also based on either valid time or transaction time. Currently MADS supports valid time. Figure 7 shows MADS hierarchy of temporal data types.

The spatiality/temporality of an application is reflected by the existence of spatial/temporal entities, but also by the existence of space- and time-related relationships between these entities. Is important to be able to explicitly describe space-related relationships in conceptual schemas. This enriches the schema, allowing these relationships to be named and described with attributes and methods.

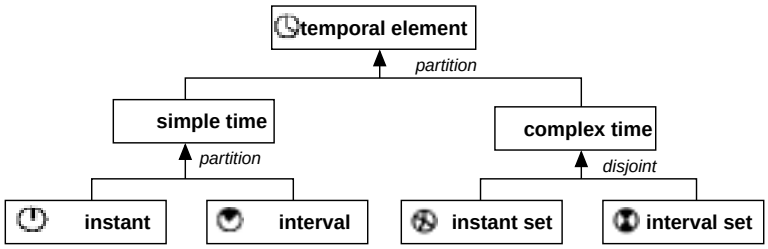


Fig. 7. MADS basic hierarchy of temporal abstract data types.

MADS constrained relationships. MADS constrained relationship types are relationship types that convey spatial and temporal constraints on the objects they link. MADS includes topological and synchronization relationships as built-in constrained relationship types. For example, a topological relationship type inside may be defined to link object types Canton and Country, expressing that the geometry of a canton is within the geometry of the related country. The list of MADS predefined topological relationship types and the associated icons is shown in Table 2. Every MADS topological relationship type is characterized by its spatial type, which is visually represented by an icon. Although these icons represent surface objects, these symbols are valid for every spatial object type.

Table 2. MADS topological relationships.

Spatial type	*Icon*	*Definition*
disjunction	○●	the linked objects have spatially disjoint geometries
adjacency	○●	geometry sharing without common interior
crossing	S	sharing of some part of interior such that, the dimension of the shared part is strictly inferior to the higher dimension of the linked objects
overlapping	◔	sharing of some part of interior such that, the dimension of the shared part is equal to the dimension of the linked objects
inclusion	◉	the whole interior of one object is part of the interior of other object
equality	●	sharing of the whole interior and of the whole envelope (for spatial objects of the same dimension)

Synchronization relationships enable specifying constraints on the life cycles of the participating objects. They convey useful information even if the related objects are not timestamped. They allow in particular to express constraints on schedules of processes. MADS built-in synchronization relationships are shown in Table 3.

Table 3. MADS temporal relationships.

Temporal type	*Icon*	*Temporal type*	*Icon*
equal		during	
meets		starts	
overlaps		finishes	
before			

3.3 Motivating Example

Figures 8 and 9 show two example schemas that will be used throughout the rest of the paper to illustrate the integration methodology we present in Sect. 4. Schema S_1 is a part of a hypothetical schema for a park administration of a city. The objects this park administration is interested in are the green plantations within the city area, their geometries, types, e.g., flower bed, park, field. As well there are bordering objects included in the schema, e.g., crossroads, build-up areas. For these objects, the park administration merely needs to know their name, e.g., crossroad, or their geometry, e.g., road, water body, build-up area. Schema S_2 is a part of a hypothetical road network schema for road management department of the same city. The focus of this schema is the detailed representation of road network elements, their classification and the relationships among them. Both schemas model real world elements geographically located in the same area - a city, thus, the populations of these schemas have some common instances providing an integration ground.

There are some concepts used in these schemas which are peculiar to MADS data model. For example, the Park entity type being non-temporal can have temporal attributes, illustrating the concept of orthogonality of the structural, temporal, and spatial dimensions of MADS data model. Flower Bed entity type has multivalued attribute FlowerType with $1 : n$ cardinality. From the point of view of MADS spatial domain ontology, the IsA hierarchy of spatial entity types is coherent with it.

We presented the basic features[4] of the MADS model, which is complete, to the best of our knowledge, in terms of semantic modeling of spatio-temporal data. The following section presents a provisory integration methodology based on MADS model. We believe that the model which possesses such features can be considered as a firm basis for fully-sound integration.

4 Provisional Integration Methodology

As described in Sect. 1, there are three essential types of conflicts to be resolved during the integration: syntactic, semantic, and structural conflicts. For each type, there are several possible ways to resolve the conflicts. When a methodology for integration is proposed it should include solutions for all the types

[4] A more detailed specification of the MADS model can be found in [17].

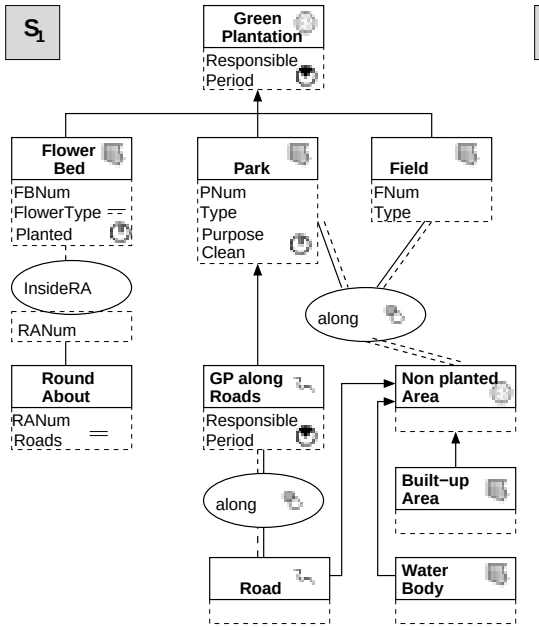


Fig. 8. Park Administration.

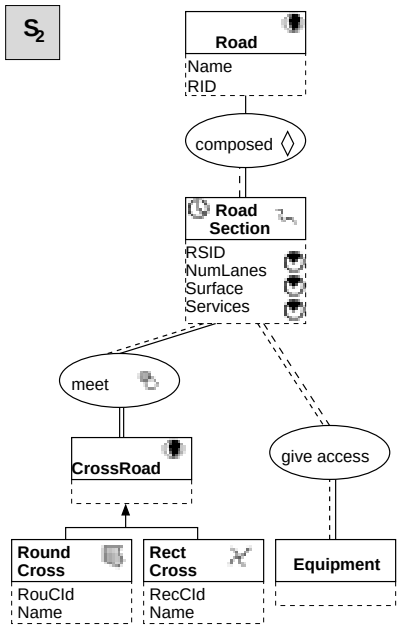


Fig. 9. Road Administration.

of conflicts. Table 4 briefly presents the integration phases, conflicts and corresponding solutions.

Table 4. Integration: Phases, Conflicts, Resolution.

Integration phases	*Conflicts*	*Resolution*
Pre-integration	Syntactic Conflicts	Modeling in MADS
ICAs formulation	Semantic Conflicts	Semantic correspondences
Integrated schema generation	Structural Conflicts	Set of possible structural solutions

Following the three-step integration process described in the literature, we realized that we could propose definite, application independent solutions only for the first two phases. For syntactic conflicts, i.e., when data involved in the integration process are logically designed with different approaches, e.g., relational, object-oriented, we propose to remodel the application in MADS conceptual model. For semantic conflicts, i.e., when the same real world facts serve different purposes for different disciplines and they have nothing in common or when the same objects are called differently in different disciplines, resolution in our methodology is done by establishing semantic correspondences or Inter-schema Correspondence Assertions (ICAs). In more details ICAs are presented in Sect. 4.1. Structural solutions taken during the third phase are dependent on the application, on the designer perception of the result of integration. In addition,

different structural solutions have their merits and shortcomings which might not be clear to the designer. Therefore, there is a need of an intermediate step in which the designer is guided through different possible structural solutions. Generally, having two entity types A and B independently on the semantic correspondences existing between them, the resulting structural solutions can be different, thus, adding another dimension to the set of the decisions to be taken during the integration process. In Sect. 4.2 we present in more details this intermediate phase called *'Choosing an integration goal'*. The results of this phase would allow to resolve structural conflicts of the next integration phase.

4.1 Interschema Correspondence Assertions

For semantic conflicts resolution we propose an integration language similar but more comprehensive and expressive than those presented in [21] and [4]. The language allows to formulate correspondences between different database schemas, the correspondences are called Interschema Correspondence Assertions. The correspondences are defined in four levels. Semantic Correspondences (SC) state the fact that there is a correspondence between two object populations. At the next level there are Property Semantic Correspondences (PSC) where the domain mismatches are resolved. The Matching Rules (MR) uniquely identify the same object instances represented diversely in several schemas. The Integrity Constraints (IC) inherited from the component schemas complete the set of correspondences and allow to deduce those that are not inferable from the component conceptual schemas.

SCs are the most general notions stating that there is something in common between the real-world objects modeled in the databases. The syntax of the SC is the following:

`EntityPath Operator EntityPath;`,

where `EntityPath` is composed of the name of a schema and the name of an entity type, `Operator` is one of the following:

- set operator - $\{\cap, \subset$ or $\sigma, \equiv\}$.

Set operators associate the populations of the entity types involved in the assertion. The choice of $\subset$ or σ operators depends on whether it is possible to state a condition for selecting a subset of an entity population. If it is the case, then the inclusion operator can be replaced with the more precise unary operator σ. Refining inclusion in such a way allows to establish a clearer correspondence between integrating entities' populations.

Example. Population links between the schemas shown in Fig. 8 and 9 exist between crossroads, e.g., RoundAbout and RoundCross entity types; and roads modeled as Roads entity types. The SCs are the following:

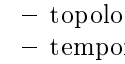
S_1.`RoundAbout` $\subset$ S_2.`RoundCross`;
S_1.`Road` $\subset$ S_2.`Road`;

Here we use the $\subset$ operator, because population of S_1.RoundAbout is only those round crossroads that contain a flower bed inside, where as population of S_2.`RoundCross` is all the round crossroads in the city. The same is true for road sections modeled by the schemas.$\diamond$

Property Semantic Correspondences precise the semantic correspondences defined by SC assertions. The syntax of a PSC is the following:

`[Function]AttributePath Operator [Function]AttributePath;`,

where `Function` is a pre-defined or user-defined function over an attribute domain, `AttributePath` is composed of the name of a schema, the name of an entity type, and the attribute or attributes' name(s)[5], and `Operator` is one of the following:

- equality of the domain values - $\{=\}$,
- user-defined correspondence of the domain values - $\{\leftrightarrow\}$,
- topological relationship -  ,
- temporal relationship -  .

PSCs assertions state general correspondences between two value domains. They establish a translation function between the two value domains and are employed in the case of reversible integration to restore the attribute values for component schemas. Existence of a PSC via the $=$ or $\leftrightarrow$ operator implies that there is a population intersection within the entities involved in this PSC. If the entity types corresponded by a PSC possess spatial features then they could be related by a topological relationship. The same is true for temporal relationships, i.e., a property semantic correspondence involving a temporal relationship can exist between these entity types. The three mentioned types of semantic correspondences are not interdependent. An example could be a basement of a building from one schema and a building from another schema: they have the same geometry, but they are not the same objects, i.e., there is a spatial relationship but no population relationship. Generally, if a spatial or temporal PSC is caused by the spatiality or temporality of the entity types involved, there may be no population intersection for the same entity types. On the other hand, if a spatial or temporal PSC relates spatial or temporal attributes of two entity types[6], then there is a population intersection between these entity types.

Example. For our example schemas there are PSCs that are caused by existence of SC, and there are those that are due to overlay of the location or time dependence of the objects modeled by the schemas.

[5] in the case of complex attributes
[6] not necessary spatial or temporal since MADS supports orthogonal concepts

S_1.RoundAbout.Roads = S_2.Road.Name;
S_1.Road ● S_2.RoadSection;
S_1.FlowerBed ◉ S_2.RoundCross;
S_1.Park ⊢⊣ S_2.RoundCross;

The last two PSCs illustrate the situation when there is no population link between two entity types but there is a link between spatial and temporal attributes of these entity types. Spatial relationship between the FlowerBed and the RoundCross indicates that a flower bed can lay inside a round about. The condition under which this assertion is true is defined by a corresponding matching rule shown in the example hereafter. The last temporal relationship would correspond to a situation when the road administration for security reasons decides to reconstruct a park surrounded by roads to a round about, as less dangerous then the previous layout or a crossroad. ⋄

On the next layer of the ICAs there are Matching Rules that allow to determine exactly which instances represent the same real-world objects via their key attribute values. The syntax of the MRs is the following:

`[Function]AttributePath Operator [Function]AttributePath;`,

where `Function` is a pre-defined or user-defined function over the key attribute domain, `AttributePath` is composed of the name of a schema, the name of an entity type, and the attribute or attributes' name(s), and `Operator` is one of the following:

- equality of the domain values - {=},
- user-defined correspondence of the domain values - {↔},
- spatial ● or temporal ⊢⊣ equality operator.

The difference between PSCs and MRs assertions is that the MRs relate only key attributes whereas PSCs may deal with any attribute. For MRs it is also necessary to presume that the equality can not always be established directly with the = operator, e.g., for different measurement systems a correspondence table must be used, for different attribute value types a function can be defined, for ad-hoc correspondence some heuristics can be used. The correspondence operator ↔ is used for such situations when the equality is not the equality in the mathematical sense. Spatial or temporal equality is used in the case when there is no other identification attribute than geometry, location, or time.

Example. The MRs corresponding to the PSC are the following:

S_1.RoundAbout.RANum ↔ S_2.RoundCross.RouCId;
S_1.Road ● S_2.RoadSection;
S_1.InsideRA.RANum ↔ S_2.RoundCross.RouCId;

S_1.Park ● S_2.RoundCross;

For Roads from S_1 and RoadSections from S_2 we do not have any other means for matching instances than to compare their geometry. The last relationship says that if the geometry of a park is equal to that of a round crossroad, then, according to the PCS stated for temporal attributes of these two instances, the park was reconstructed to the round crossroad.◇

4.2 Integration Goals

As we mentioned above in this section, we consider important to add one more step to the integration procedure: choosing an integration goal. The integration goal imposes application of an integration technique for schema elements involved in the semantic correspondences. The choice is based on permissible characteristics of the resulting integrated schema element(s). There are three possible types of losses as the consequence of integration [7]:

- loss of information;
- loss of precision;
- reversibility of the integrated schema, meaning that all the information stored in component schemas is deducible from the integrated one.

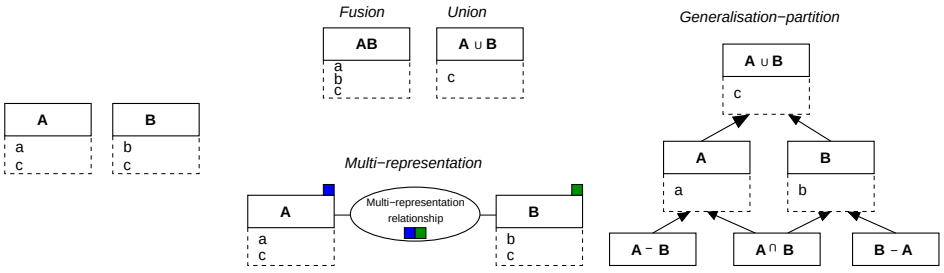


Fig. 10. Sample integration patterns.

Figure 10 shows several ways to integrate two entity types A and B with one common attribute c. The *Fusion* approach preserves the information, e.g., neither attribute values nor instances are lost. The cardinalities of the attributes a and b are set to be minimal, e.g., if an attribute was multivalued, its domain is reduced to a single value. Depending on the application this might be counted as loss of precision. Concerning the reversibility of the fused entity type, it is reversible if there exists a one-to-one attribute value mapping function. Under the *Union* approach, information is not preserved because only the common attribute is retained in the resulting entity type. Obviously integration under this

approach is not reversible, the values of the attributes a and b cannot be reconstructed. The last approach, *Generalization-partition*, preserves the information, as well as precision and it is reversible. Integration techniques such as those presented in Fig. 10 can be applied to a whole schema, i.e., all the schema elements are integrated according to the chosen technique or, for each schema element a particular, the most suitable integration technique is applied.

Example. Using our example and assuming that we need to make an integrated schema based on the two input schemas we can obtain significantly different results. They depend on the purpose of usage of the integrated schema. If the integrated view is created for park administration, we might keep minimal information about the crossroads and the resulting entity type would be modeled as shown in Fig. 11. This entity type is obtained with the fusion technique, with loss of information, i.e., the spatial features are dropped; with loss of precision, i.e., now the cardinality of the link between RoundAbout and FlowerBed is $0:1$; and finally with no precise way to reconstruct geometry of crossroads, i.e., geometry can be approximately derived form the geometry of flower beds. But, still, such a representation suits the user needs. If, on the other hand, the integrated

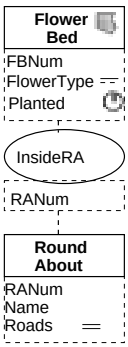


Fig. 11. Fusion technique applied.

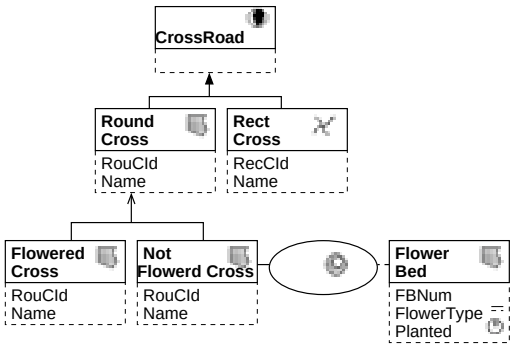


Fig. 12. Generalization-partition technique applied.

schema would be used by road administration or for both the divisions, we might want to preserve all the information and maybe enrich a resulting schema with new entity or relationship types. Figure 12 shows the result obtained with the generalization-partition technique. Regarding the temporal PSC between Park and RoundCross, it can be modeled with additional temporal transition relationship between these entity types. We did not present this type of relationship in the paper, but it is supported in MADS data model.$\diamond$

We believe that it is important to present clearly to the integrated schema designer the possible structural solutions together with the features, or loss of those, that the resulting schema will possess. Formalization of the possible struc-

tural solutions, on one hand limits the designer in the choice of the structural solutions[7] by those that are proposed to him/her. On the other hand, a formal definition of the structural solutions makes the goal of designing a semi-automatic integration tool close to be claimed as practically feasible.

5 Summary and Future Developments

In this paper we deductively presented the notion of interoperability in application to the spatio-temporal domain. As the most general means of interoperability we presented a generic architecture of an interoperable system. Then we specialized the structure of the mediation component that is intended to provide the interoperable functionality of the system. Considering our domain of interest we presented MADS conceptual data model and MADS spatial and temporal domain ontologies. We then considered the conceptual level of data representation, for which we presented a provisory integration methodology comprising four phases: *pre-integration*, *correspondence formulation*, *choosing an integration goal* and *generating an integrated schema*. Finally, we gave a preliminary syntax for formulating the interschema correspondence assertions which are basically the rules defining the common elements found in heterogeneous spatio-temporal data sources.

Our ambitions are to formalize the syntax of inter-schema correspondence assertions for spatio-temporal domain, to add a viable way to manage heterogeneous integrity constraints, and finally to design a tool which would realize our research proposals.

References

1. David Abel, Beng Chin Ooi, Kian-Lee Tan, and Soon Huat Tan. Towards integrated geographical information processing. *International Journal of Geographical Information Science*, 12(4):353–371, June 1998.
2. Goksel Aslan and Dennis McLeod. Semantic heterogeneity resolution in federated databases by metadata implantation and stepwise evolution. *VLDB Journal*, (8):120–132, 1999.
3. Oleg T. Balovnev, Andreas Bergman, Martin Breunig, Armin B. Gremers, and Serge Shumilov. A corba-based approach to data and system integration for 3d geoscientific application. In *Proceedings of the 8th International Conference on Spatial data Handling (SDH'98)*, pages 396–407, Vancouver, Canada, July 1998.
4. Sonia Bergamaschi, Silvana Castano, Maurizio Vincini, and Domanico Benevantano. Semantic integration of heterogeneous information sources. *Data & Knowledge Engineering*, (36):215–249, March 2001.
5. Jan Chomicki and Peter Z. Revesz. Constraint-based interoperability of spatio temporal databases. *GeoInformatica*, 3(3):211–244, September 1999.
6. Online dictionary of computer science. http://burks.bton.ac.uk/burks/foldoc/.

[7] for federating two entity types there exists at least 15 choices, see [7], we believe that a designer would hardly come up by him/herself with that many variants.

7. Yann Dupont. *Une méthode flexible pour l'intégration de schémas dans les bases de données á objets complexes.* PhD thesis, École polytechnique fédérale de Lausanne, 1996.
8. Jerry Fowler, Brad Perry, Marian Nodine, and Bruce Bargmeyer. Agent-based semantic interoperability in infosleuth. *ACM SIGMOD Records*, 28(1):60–67, March 1999.
9. Michael Goodchild, Max Egenhofer, Robin Fegeas, and Cliff Kottman, editors. *Interoperating Geographic Information Systems.* Kluwer Academic Publishers, 1999.
10. Nicola Guarino. *Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology*, chapter Semantic Matching: Formal Ontological Distinctions for Information Organisation, Extraction, and Integration, pages 139–170. Springer-Verlag, 1998. M.T. Pazienza.
11. Nicola Guarino and Christopher Welty. Ontological analysis of taxonomic relationships. In A. Lander and V. Storey, editors, *Proceedings of ER-2000: The 19th International Conference on Conceptual Modeling.*, LNCS. Springer-Verlag, October 2000.
12. Nicola Guarino and Christopher Welty. Towards a methodology for ontology based model engineering. In J. Bezivin and J. Ernst, editors, *Proceedings of the ECOOP-2000 Workshop on Model Engineering*, June 2000.
13. Amarnath Gupta, Richard Marciano, Iliya Zaslavsky, and Chaitanya Baru. Integrating gis and imagery through xml-based information mediator. In P. Agouris and A. Stefanidis, editors, *International Workshop on Integrated Spatial Databases: Digital Images and GIS (ISD'99)*, volume 1737 of *Lecture Notes in Computer Science*. Springer, Portland, Maine, USA, June 1999.
14. Natalia Juristo and Ana M. Moreno. Introductory paper: Reflection on conceptual modeling. *Data & Knowledge Engineering*, (33):103–117, July 2000.
15. Opengis consorcium. http://www.opengis.org.
16. Michael P. Papazoglu, Stefano Spaccapietra, and Zahir Tari, editors. *Advances in Object-Oriented Modeling*, chapter Database Integration: The Key to Data Interoperability. The MIT Press, 2000.
17. Cristine Parent, Stefano Spaccapietra, and Esteban Zimanyi. Spatio-temporal conceptual models: Data structures + space + time. In *7th ACM Symposium on Advances in GIS, Kansas City, Kansas*, Kansas City, Kansas, November 5-6 1999.
18. Xiaobei Qian and Teresa F. Lunt. Semantic interoperation: A query mediation approach. Technical Report SRI-CSL-94-02, Computer Science Laboratory, SRI International, April 1994.
19. Amit P. Sheth. *Changing Focus on Interoperability in Information Systems: from System, Syntax, Structure to Semantics*, chapter in [9]. Kluwer Academic Publishers, 1999.
20. Stefano Spaccapietra, Cristine Parent, and Cristelle Vangenot. Gis databases: From multiscale to multirepresentation. In B.Y.Choueiry and T.Walsh, editors, *Proceedings 4th InternationalSymposium, SARA-2000*, volume 1864 of *LNAI*, Horseshoe Bay, Texas, USA, July 26-29 2000. Springer-Verlag.
21. J. Tan, A. Zaslavsky, and A. Bond. Meta object approach to database schema integration. In *Proceedings of the International Symposium on Distributed Objects and Applications*, 2000.
22. Gio Wiederhold. Mediators is the architecture of future information systems. *The IEEE Computer Magazine*, March 1992.