Semantic Mappings in Description Logics for Spatio-Temporal Database Schema Integration*

A. Sotnykova[1], C. Vangenot[1], N. Cullot[2], N. Bennacer[3], and M-A. Aufaure[3]

[1] École Polytechnique Fédérale de Lausanne, Database Laboratory, CH-1015 Lausanne, Switzerland, {Anastasiya.Sotnykova,Christelle.Vangenot}@epfl.ch
[2] Université de Bourgogne, Laboratoire LE2I, F-21078 Dijon Cedex, Nadine.Cullot@u-bourgogne.fr
[3] École Supérieure d' Électricité, F-91192 Gif-Sur-Yvette Cedex, France, {Nacera.Bennacer, Marie-Aude.Aufaure}@supelec.fr

Abstract. The interoperability problem arises in heterogeneous systems where different data sources coexist and there is a need for meaningful information sharing. One of the most representive realms of diversity of data representation is the spatio-temporal domain. Spatio-temporal data are most often described according to multiple and greatly diverse perceptions or viewpoints, using different terms and with heterogeneous levels of detail. Reconciling this heterogeneity to build a fully integrated database is known to be a complex and currently unresolved problem, and few formal approaches exist for the integration of spatio-temporal databases. The paper discusses the interoperation issue in the context of conceptual schema integration. Our proposal relies on two well-known formalisms: conceptual models and description logics. The MADS conceptual model with its multiple representation capabilities allows to fully describe semantics of the initial and integrated spatio-temporal schemas. Description logics are used to express the set of inter-schema mappings. Inference mechanisms of description logics allow us to check the compatibility of the semantic mappings and to propose different structural solutions for the integrated schema.

1 Introduction

Information sharing between heterogeneous information sources is a significant challenge, which has been the focus of much research but remains an open problem. Enabling the cooperation of heterogeneous information systems is not easy to achieve because related knowledge is most likely described in different terms and using different assumptions and different data structures. Heterogeneity may arise from syntactic, structural and semantic differences in the data sources.

* The work presented in this paper was partly carried out in the framework of the EPFL Center for Global Computing and supported by the Swiss National Funding Agency OFES as part of the European projects KnowledgeWeb (FP6-507482) and DIP (FP6-507483).

Syntactic heterogeneity is due to the use of diverse database models (e.g. object oriented vs. relational), *structural heterogeneity* arises from different conceptual choices during the database modeling phase (e.g., modeling as an object, as a relationship, or as an attribute), and *semantic heterogeneity* comes from differences between the terms used to represent information and their intended meaning. Heterogeneity is accentuated for spatio-temporal data, due to the existence of two very different paradigms for data representation, known as the raster mode (space is represented through images) and the vector mode (space is represented as sets of localized objects). Moreover, spatio-temporal data can be represented at different granularities or levels of detail for the spatial and/or temporal features. Also, we have to consider topological relationships between objects, temporal evolution and synchronization relationships.

Two main categories of frameworks have been proposed for the co-operative information systems: *federation of information systems* [1] and *mediation* which relies on the definition of *wrappers* and *mediators* [2]. Wrappers are used to access local sources from the mediation layer and mediators provide a transparent access to the information from the cooperative layer. Mediation-based architectures facilitate evolution through the addition of new data sources. They support cooperation of large information systems and thus they are more suitable in a web environment. Federation-based architectures are best suited for small-scale cooperation. In all these approaches, information sharing can be done either through the definition of direct mappings between the source data sets, or through the definition of an integrated schema together with associated mappings supporting access to the existing data instances.

Irrespectively of the system architecture, a fundamental task in integration is the ability to recognize corresponding information in heterogeneous data sets and to describe the mappings between them. A large number of papers have investigated various facets of mappings, such as mapping discovery, mapping definition or mappings usage.

Mapping discovery. Surveys originating from two different communities, database and ontology, analyse various propositions from different points of view: database integration [3] and ontology mapping [4]. Work on mapping discovery aims at providing heuristics to find corresponding elements in different information systems, and basically relies on similarity measures. In the survey on automatic schema matching, Rahm and Bernstein in [3] propose a classification of the matching approaches. They distinguish the shema-level and the instance-level matchers. The methods for matching discovery are classified as element-level or structure-level with linguistic or constraint-based heuristics. Automatic mapping discovery became particularly important for ontology cooperation due to the large number of concepts in an ontology. Ehrig and Sure in [5] propose a methodology combining different similarity measures for identifying mappings between two ontologies. Doan et al. in [6] and [7] propose a system, GLUE, that apply machine learning techniques to improve the mapping discovering process. However, complex mappings have proven difficult to extract and the mapping discovery procedure certainly requires human feedback. Dhamankar in

[8] presents a promising system, iMAP, for the discovery of complex mappings between database schemas. However, mapping discovery between heterogeneous schemas describing spatio-temporal data still remains an open issue. In the works on ontology mapping described in [4], the sets of mapping operators used in different mapping methods are inferior to the one we use for spatio-temporal schemas mapping; the algorithms used for initial mappings proposition are not designed to capture ontologies spatio-temporal features.

Inter-schema correspondences. Complementary to the above approaches, other research works ([9], [10], [11], [12]) focus on formalisms to specify and use inter-schema knowledge. From a conceptual perspective, inter-schema knowledge identifies elements (or sets of elements) in two schemas that describe the same (or related) facts in the real world, and specifies to what extent the data instances and their type definitions relate to each other (i.e., what is identical, what is similar, what is different). This inter-schema knowledge can then be used to build the integrated schema and to provide for an integrated access to the data sources. The four works presented below follow this objective using different languages. A formalism relying on a logic-based language is proposed by Catarci and Lenzerini in [9]. The language they propose is used to describe both schemas and inter-schema knowledge. The reasoning mechanism of the language can then be used to check inter-schema consistency (i.e., the correctness of the cooperative information system) and to support integrated access to data. Calvanese et al. [10] present an architecture for information integration. A Description Logic called $\mathcal{DLR}$, which includes concepts and n-ary relationships, is used to describe the database schemas, to specify inter-schema knowledge; reasoning services are used during the integration process. The same language, $\mathcal{DLR}$, is proposed by Calvanese et al. in [11] to define mappings in a general framework for ontology integration. These mappings allow the mapping of a concept in one ontology to a view, i.e., a query in another ontology. Finally, Devogele et al. [12] propose a complete methodology for spatial database integration based on three phases: schema preparation, correspondence investigation, and integration. The authors also provide an algebraic data manipulation language (algebra for complex objects) to describe inter-schema correspondences that fully supports the description of correspondences between the spatial features of data.

Querying. Once mappings are formally defined, one should be able to use them for query answering and reasoning [13]. Calvanese et al. [11] discuss various approaches for specifying mappings (global- and local-centric approaches) and, for each approach, analyze the complexity of query answering. The authors conclude that mappings should be defined using suitable mechanisms based on query languages. In [14], Halevy et al. express mappings between data sources on a pairwise basis and define inclusion and equivalence relationships between views of each schema. An algorithm enabling queries to go through mappings in order to find data is also proposed.

Semantic enrichment. In order to reconcile semantic heterogeneity more semantic information about data is needed. Various proposed approaches add extra information to data either through the specification of meta-data, or through the explanation of the context of data or more generally, by using descriptions stored in ontologies. *Meta-data* describe the content of the underlying data in an easily understandable way. *Contexts* are more complex descriptions specifying the domain of source data. *Ontologies*, by definition, provide an encoded representation of a shared understanding of terms and concepts in a given domain and community. They serve as semantic references for users or applications that accept to align their interpretations of the semantics of their data to the interpretation stored in the ontology. Ontologies are actually extensively proposed as a means to overcome interoperability problems [15]. This is the focus of the work of Fonseca et al. in [16]. In their framework, conceptual schemas of geographical databases are mapped to spatial ontologies that are considered as the formal representation of the spatial semantics. The objective in describing such mappings is to enrich the conceptual schema descriptions and thus, to improve the integration of database conceptual schemas. Hakimpour and Geppert in [17] propose a database integration approach that employs formal ontologies merging. Source ontologies (one per database source) are merged by a reasoning system that finds semantic similarity relations between the various definitions used for each concept. An ontology-based schema integrator builds the global schema of the integrated database using the source schemas and the mappings found during the ontology merging process. Fonseca et al. in [18] propose an Ontology-Driven GIS system which plays the role of a system integrator. The idea is to provide access to data by browsing through ontologies. The architecture is based on four main components namely the *ontology server*, the *ontologies*, *mediators* and *applications* that give access to the information sources. The ontology server is the central component providing the connection between the ontologies, the applications and the information sources. The integration is partly realized by the mediators: when the information system is queried, the mediators extract parts of information necessary to generate a complete instance from the ontologies and the information sources.

Our proposal focuses on the co-operation of spatio-temporal databases. In this respect, our objective is to propose a complete methodology for the integration of spatio-temporal conceptual schemas. Our approach relies on two well-known formalisms: conceptual models and description logics. Spatio-temporal conceptual schemas to be integrated are specified using the MADS conceptual data model [19], which can represent rich spatio-temporal semantics. Reasoning services of description logics are then used to check the consistency of the mappings that guide the construction of the integrated schema.

Compared to the papers presented above, our proposal falls within the scope of approaches that aim at defining a formalism or methodology to specify and use inter-schema knowledge. We do not tackle the issue of mapping discovery, as we assume that a set of inter-schema correspondences given by the designer is

completed by an inference engine, nor do we consider the subject of query rewriting, which is out of the scope of this work. However, the proposal contributes to the area of research on the following original topics:

- the proposed methodology, based on description logics reasoning mechanisms, conceptual modeling and integrity constraints, is hybrid and thus innovative;
- we are dealing with spatio-temporal data which, to the best of our knowledge, has not yet been attacked;
- we are using reasoning mechanisms of description logics in order to validate the set of inter-schema mappings against the source schemas.

The paper is structured as follows: Section 2 briefly describes the MADS model and introduces two schemas that are used as a running example throughout the paper. Section 3 introduces description logics: The $\mathcal{SHIQ}$ description logic is used to describe source schemas and inter-schema mappings without any spatial and temporal features. The spatial and temporal aspects are specified using an extension of $\mathcal{ALC}$, $\mathcal{ALCRP}(\mathcal{D})$, that provides concrete domain with space and time. Section 4 presents our integration methodology through its various parts: specification of the inter-schema mappings, validation, and generation of an integrated schema. Finally, Section 5 concludes the paper.

2 The MADS model

MADS [19] is an object+relationship spatio-temporal conceptual data model. In this model, we assume that the real world of interest that is to be represented in the database is composed of complex objects and relationships between them; both characterized by properties (attributes and methods), and both may be involved in a generalization hierarchy (is-a links). To further illustrate our proposal, we will use two MADS schemas shown in Figures 1 and 2. These schemas are designed for two tourist offices describing the same geographical area, the city of Paris. The purpose of the schema T_2 is to provide tourists with information on the closest to the tourist sites boat, bus, metro, and tram stops. Schema T_1 is more general and describes the transport means and the tourist sites of the same city. Both schemas illustrate structural, spatial, and temporal MADS modeling capabilities.

Data structuring capabilities of MADS are orthogonally complemented with space and time modeling concepts, i.e., spatiality and temporality may be associated at the various structural levels: object, attribute, and relationship. The spatiality of an object conveys information about its location and its extent; the temporality describes its lifecycle. For instance in Fig. 1, the object type TouristPlace has both a spatiality (an area) and a temporality (an interval). Attributes may have spatial (e.g. the attribute Start of the object type Walk in Fig. 2) or temporal (e.g. the attribute Season of the object type Theatre) domains of values. A set of predefined spatial and temporal abstract data types is used to describe the spatial and temporal extents of data. The abstract data types

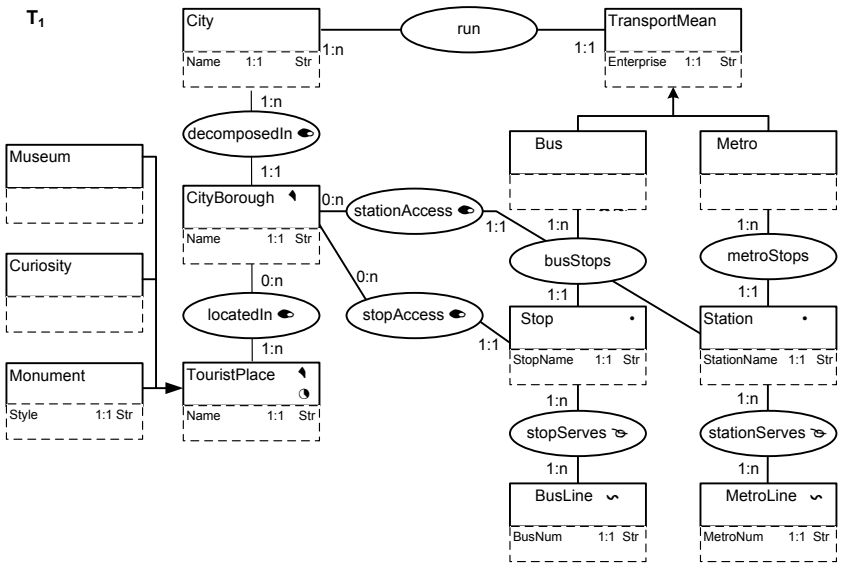

Fig. 1. Schema T_1.

are organized in a generalization hierarchy where generic data types are used to describe domains whose values may be of different, more specific types, e.g., small rivers may be described as lines, bigger ones as areas, hence, their domain should be of the generic type Geo. Attributes may also be space- or time-varying, supporting in this way the continuous view of space. For instance, the attribute nbLanes of Road in Fig. 2 whose value is changing according to the considered road section is a space-varying attribute.

Relationships are either classical n-ary relationships among individual objects or n-ary relationships among sets of objects (multi-association). Relationships may be enhanced with one or several specific semantics, such as aggregation, topological, synchronization, and inter-representation semantics. Topological and synchronization semantics define constraints between spatial and temporal objects respectively. The relationship along between the object types Stop and TransportLine in Fig. 2, holds a topological semantics of intersection.

Multi-representation has been added in MADS as an additional orthogonal dimension. Multi-representation allows the definition in the same schema of several representations for the same real world objects. Those multiple representations may be the consequence of diverging requirements during the database design phase or, in the particular context of spatial data, of the description of data at various levels of detail. The MADS multi-representation feature may also be used in the context of spatial database integration where the full integration, possibly based on different levels of detail, is not possible [12].

To allow users to retrieve specific representations from the set of existing ones, these representations have to be distinguishable and denotable. To this extent,

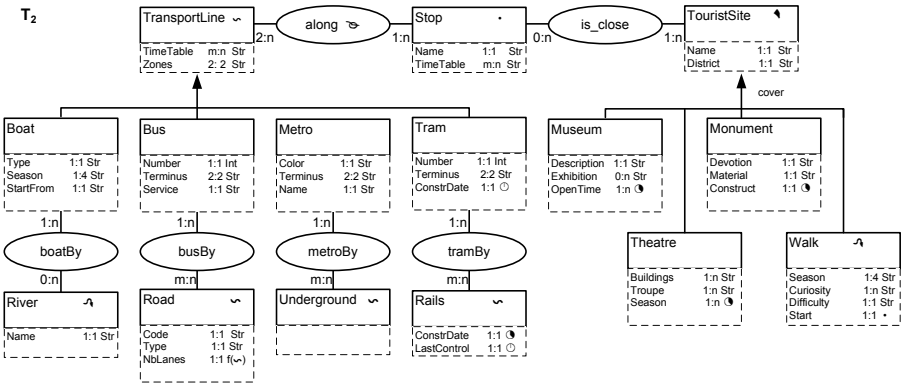

Fig. 2. Schema T_2.

representation stamps are added on data, whether they are object type instances or attribute values, and on meta-data, object and relationship type definitions or attribute definitions. Stamps are vectors of values characterizing the context of each representation (e.g. spatial resolution, viewpoint, ...). Object and relationship types may be representation-varying types and thus have a different set of attributes according to the considered representation. For instance in Fig. 7.d, the object type Monument is a multi-representation type with two definitions, one for stamp t1 with the attribute Style and one for stamp t2 with the attributes Devotion, Material and Construct. Attributes of such types may have several definitions (different cardinalities and/or value domains) and/or several values (the notation of such an attribute is f(t1, t2) to state that the value is function of the stamp). For instance, the attribute Name of District has a representation-varying definition, i.e., it is a multi-valued attribute for the stamp t1 and a monovalued attribute for the stamp t2. Relationship types may hold several different semantics according to the representation and, for instance, be a topological relationship in one representation and a synchronization in another. We also propose a specific inter-representation semantics that may be applied to both associations and multi-associations to denote that the linked instances are different representations of the same real world object. Actually this inter-representation semantics does not induce any constraints between the linked objects. It denotes paths in the schema that are likely to support consistency checks and update propagation rules. For instance, the correspond relationship in Fig. 7.c holds a multi-representation semantics which states that the instances of TouristSite and TouristPlace linked through this relationship are two representations of the same real world object. For data manipulation, we have defined an algebraic language that provides formal support for manipulating multi-represented data. Concerning multi-representation, users may specify one or several stamps that delimit the subset of the database they will be working on.

3 Description Logics

Description Logics are a family of terminological formalisms with formal logic semantics and designed for representing knowledge and for reasoning about it. Basic elements in a description logic are primitive concepts, primitive roles, the universal concept $\top$ and the bottom concept $\bot$. Complex concepts and roles can be built from primitive ones using the considered description logic constructors. The terminology defines relevant concepts of the domain and their properties. Then individuals occurring in the domain are described using this terminology [20].

Basic description logic constructor, as found in $\mathcal{ALC}$ are: $\neg C$ (negation), $C \sqcap D$ (conjunction),$\forall R.C$ (value restriction) and $\exists R.\top$ (limited existential quantification) where C and D are concepts and R is a role. The $\mathcal{ALC}_{R+}$ description logic is an extension of $\mathcal{ALC}$ with transitive roles. The description logic $\mathcal{SHIQ}$ [21] extends $\mathcal{ALC}_{R+}$ with inverse roles, role hierarchies and qualified number restrictions ($\geq n\ R.C$ and $\leq n\ R.C$). Qualified number restrictions play an important role for representing and for reasoning about conceptual models because they add the ability to model cardinalities of relationships [22]. The expressiveness of $\mathcal{SHIQ}$ is rich and allows encoding of database schemas but it is insufficient to describe spatio-temporal objects.

For representing and reasoning about spatial objects, spatial description logics have been proposed in the literature. Qualitative spatial reasoning in description logic is based on topological relationships [23],[24]. These are known as the set of the $\mathcal{RCC}_8$ relations : Equal (EQ), Disconnect (DC), Externally Connected (EC), Partial Overlap (PO), Tangential Proper Part (TPP), Non-Tangential Proper Part (NTPP) and the inverses of TPP and NTPP : TPPI and NTPPI. A family of description logics called $\mathcal{ALCI}_{\mathcal{RCC}}$ suitable for qualitative spatial reasoning on various granularity is discussed in [25]. The satisfiability problem of these logics is addressed considering the role axioms derived from the $\mathcal{RCC}$ composition tables. Inverse and disjoint roles are also needed to capture the semantics of theses relationships.

Recent work [26] has been proposed to find a way to combine available knowledge representation and reasoning formalisms suitable to consider different real aspects of the world such as time and space. An $\mathcal{E}$-connection is defined in terms of abstract description systems (ADSs), and is a combination of description logics, numerous logics of time and space, and modal and epistemic logics. Link relationships are introduced to combine the formalisms while keeping their domains disjoint. One of the main contributions of the work in [26] is the study on the decidability of the $\mathcal{E}$-connections; it is shown that the $\mathcal{E}$-connections are decidable even with expressive link operators like boolean combinations of link relations.

Extending descriptions logics with concrete domains is a way to introduce new data types such as integer or rational, or to deal with specific dimensions of objects such as spatial or temporal features. The $\mathcal{ALC}(\mathcal{D})$ [27] description logic extends the $\mathcal{ALC}$ DL by adding a new concept-forming predicate operator. $\mathcal{ALC}(\mathcal{D})$ divides the set of objects into two disjoint sets, the abstract and the

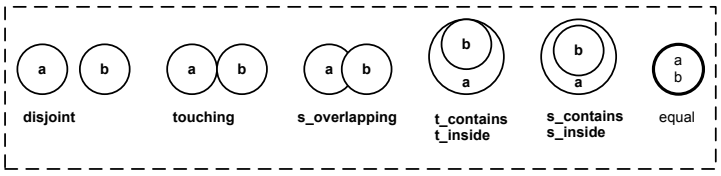

Fig. 3. Topological relationships.

concrete objects such as numbers, strings and in particular spatial and temporal objects. Abstract objects can be related to abstract objects via abstract roles and to concrete objects via concrete roles. The relationships between concrete objects are described with a set of domain specific predicates. The pair consisting of a set of concrete objects and the set of predicates forms the concrete domain. Concrete domains increase the expressive power of an extended description logic and allow reasoning on these new features.

The $\mathcal{ALCRP}(\mathcal{D})$ DL proposed by V. Haarslev [28] extends $\mathcal{ALC}(\mathcal{D})$ to build complex roles based on a role-forming predicate operator [29]. In particular, an appropriate concrete domain $\mathcal{S}_2$ is defined for polygons using $\mathcal{RCC}_8$ relations as basic predicates of concrete domain as shown in Fig. 3 (disjoint stands for the DC $\mathcal{RCC}_8$ relationship, touching for EC, s_overlapping for PO, t_inside (t_contains) for TPP, s_inside (s_contains) for NTPP, and equal for EQ). For temporal aspect, the concrete domain $\mathcal{T}$ is a set of time intervals and the 13 Allen relationships (before, after, meets, met-by, overlaps, overlapped-by, during, contains, starts, started-by, finishes, finished-by, equal) are used as basic predicates describing the relationships between intervals. The combination of $\mathcal{S}_2$ and $\mathcal{T}$, $\mathcal{S}_2 \oplus \mathcal{T}$, defines a spatio-temporal concrete domain.

For our purpose, we exploit the $\mathcal{ALCRP}(\mathcal{S}_2 \oplus \mathcal{T})$ expressive power to describe source spatio-temporal schemas and inter-schema mappings. Moreover, the underlying theory allows to detect both inconsistencies and implicit information in the integration process. Using $\mathcal{ALCRP}(\mathcal{S}_2 \oplus \mathcal{T})$, we can define a concept that has a geometry with a specific concrete spatial role called hasArea. Further, using the hasArea feature, we can specify topological relationships between spatial concepts. To define a concept as a temporal concept we can use a specific concrete temporal role called hasDuration. Through this role, temporal relationships between concepts can then be defined. For example, elements of the schema T_1 in Fig. 1 can be described as follows.

A city has a name, it is decomposed in districts and it runs transport means:

$$\begin{aligned}\mathsf{City} \sqsubseteq\ & \forall \mathsf{Name.String} \\ & \sqcap \geq 1\mathsf{Name} \sqcap \leq 1\mathsf{Name} \\ & \sqcap \forall \mathsf{decomposedIn.District} \\ & \sqcap \forall \mathsf{run.TransportMean}\ ;\end{aligned}$$

A tourist place has a name, it is a spatio-temporal object thus, it has a ge-

ometry and a temporality which are respectively specified by the concrete roles hasArea of the domain Polygon and hasDuration of the domain Interval:

$$\begin{aligned}&\mathsf{TouristPlace} \sqsubseteq \forall \mathsf{Name.String}\\ &\quad \sqcap \geq 1\mathsf{Name} \sqcap \leq 1\mathsf{Name}\\ &\quad \sqcap \exists \mathsf{hasArea.Polygon}\\ &\quad \sqcap \exists \mathsf{hasDuration.Interval}\ ;\end{aligned}$$

Museums are tourist places:

$$\mathsf{Museum} \sqsubseteq \mathsf{TouristPlace}\ ;$$

Monuments are tourist places having a specific feature expressing their style, with the cardinality stating that a monument has exactly one style:

$$\begin{aligned}&\mathsf{Monument} \sqsubseteq \mathsf{TouristPlace}\\ &\quad \sqcap \forall \mathsf{Style.String}\\ &\quad \sqcap \geq 1\mathsf{Style} \sqcap \leq 1\mathsf{Style}\ ;\end{aligned}$$

Where, the object types City, Museum, Monument, TouristPlace, District, and TransportMean are modeled as abstract concepts; the relationships decomposedIn, run, and attributes Name, and Style are modeled as roles. Inverse roles can also be defined, for example $\mathsf{isRun} \equiv \mathsf{run}^{-1}$. It is also possible to define a contemporary museum as a museum which has at least 10 contemporary paintings:

$$\mathsf{ContemporaryMuseum} \equiv \mathsf{Museum} \sqcap \geq 10\ \mathsf{expose.ContemporaryPainting}\ ;$$

To define museums that are spatially connected to some monuments and whose opening times overlap, we first define a spatial predicate connected as the disjunction of elementary predicates, a spatial role spatial_connected based on the previously defined connected predicate, and a temporal role duration_overlaps. The role spatial_connected (respectively duration_overlaps) may be used to link couples of objects whose spatiality (respectively lifecycle) satisfy the connected (respectively overlaps) predicate. Then with these roles, we define such museums, MuseumMonument, as follows:

$$\begin{aligned}&\mathsf{connected} \equiv \mathsf{touching} \vee \mathsf{s\_overlapping} \vee \mathsf{t\_contains} \vee \mathsf{t\_inside} \vee \mathsf{s\_contains} \vee\\ &\quad \mathsf{s\_inside} \vee \mathsf{equal}\ ;\\ &\mathsf{spatial\_connected} \equiv \exists(\mathsf{hasArea})(\mathsf{hasArea}).\mathsf{connected}\ ;\\ &\mathsf{duration\_overlaps} \equiv \exists(\mathsf{hasDuration})(\mathsf{hasDuration}).\mathsf{overlaps}\ ;\\ &\\ &\mathsf{MuseumMonument} \sqsubseteq \mathsf{Museum}\\ &\quad \sqcap \exists \mathsf{spatial\_connected.Monument}\\ &\quad \sqcap \exists \mathsf{duration\_overlaps.Monument}\ ;\end{aligned}$$

These descriptions combine not only abstract and concrete objects but also the spatial and temporal concrete domains. This aspect ensures that a GIS

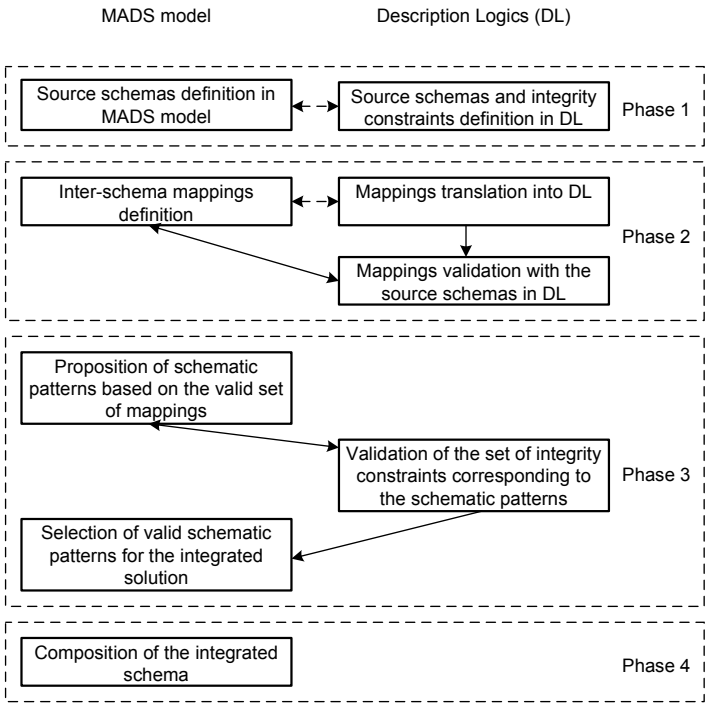

Fig. 4. Integration phases.

system reasoning can be achieved according to the intended semantics of spatio-temporal objects.

4 Integration methodology

Our integration methodology uses two modeling approaches: database conceptual modeling and modeling in description logics. Fig. 4 shows phases that compose our integration methodology. In the scope of this paper we assume that the source database schemas are expressed in the MADS data model, which has been introduced in Sect. 2. The MADS model has a rich spatio-temporal semantics that is easily understood by a wide circle of designers and users. Contrary to the proposals with rather weak data models enhanced with additional mechanisms for mappings discovery, e.g., as in [8] and [6], we adhere to a different approach, where at the very first phase of the integration procedure, the data are modeled with a very expressive conceptual model. The expressiveness of MADS on one hand greatly simplifies manual mapping discovery, and on the other hand it makes the issue of implementation of mapping discovery algorithms less important in the scope of our integration methodology. Such techniques for MADS model would require very sophisticated algorithms because besides the

structural dependencies, i.e., subsumption, there are three more dimensions - spatial, temporal, and multi-representational to be encoded together with their semantics.

MADS was conceived as a conceptual database model, and thus, it lacks reasoning services for the schema integration processes. Defining inter-schema mappings is an error-prone task done manually by the integrated schema designer. Therefore, the compatibility of the set of mappings has to be checked, and to do this task we employ the DL reasoning capabilities. As inter-schema mappings should be validated against the schemas, the source schemas are translated in description logics (*Phase 1* in Fig. 4). Then the set of inter-schema mappings is translated into description logics (*Phase 2* in Fig. 4).

In our method we differentiate several kinds of inter-schema mappings that are detailed in Sect. 4.1. The set of inter-schema mappings conditions changes that can be potentially applied to the source schemas to construct the final integrated schema. Reasoning services of the DL are used to validate these changes by checking the compatibility of the integrity constraints associated with them (*Phase 3* in Fig. 4).

At the final phase of our method the schema designer is presented with a set of valid schematic patterns that can be used to design the integrated schema (*Phase 4* in Fig. 4). For the running example we present possible structural solutions in Sect. 4.2.

4.1 Inter-schema mappings

The inter-schema mappings are initially formulated in the MADS language, for details of the language the interested reader may refer to [30]. We distinguish several types of MADS inter-schema mappings. Firstly, there are mappings that express the relationships between populations of schema elements that are intentionally related. We use terms intentionally and extensionally in the same sense as in [9], i.e., intentionally related object types share the schema level representation; extensionally related object types share parts of their populations[4]. We call these mappings *Schema population Correspondences* or SCs. For the population correspondences we apply the set operators shown in Fig. 5. Intuitively, if an SC is asserted, then the intentional equivalence is assumed, and the extensional relationship is defined by the operator. If the operator is disjoint, there are no common instances in the two populations.

Further, another set of mappings describes the intentional relationships between the descriptions of the schema elements involved in an SC. By the description we assume the attributes, including identifiers, and relationships. Since, the attributes and relationships in MADS can have spatial/topological and(or) temporal/synchronization semantics, the set of operators for the set of *Property semantic Correspondences* or PCs includes spatial operators (Fig. 3) and a subset

[4] These relations are orthogonal, i.e., two object types can be intentionally equal (the same schema representation) but extensionally disjoint (no common instances) or vice versa.

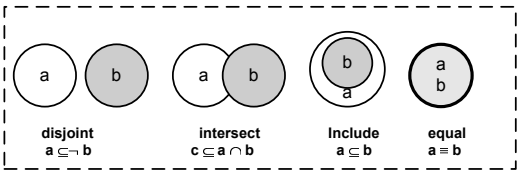

Fig. 5. Population relationships and corresponding DL expressions.

of Allen operators mentioned in Sect. 3. A subset of PCs that involves identifier attributes is called *Matching Rules* or MR. In case of a non-disjoint operator in the SC, the MRs are used to match identical instances. This set of mappings, formulated for all intentionally and extensionally related schema elements is used then for defining possible schematic patterns for an integrated schema.

For the purpose of this paper we will give the inter-schema mappings already translated in DL as presented in Sect. 3. The intention of this translation is to validate the set of inter-schema mappings using inference mechanisms of DL. For this purpose we will weaken some semantic of the mappings, e.g., we use the same syntax to state relationships between key and non-key attributes for the validation procedure, but for the clarity of the paper we keep the notion of *Matching Rules* in further discussions.

Schema population Correspondences. For our running example (schemas T_1 and T_2 in Figures 1 and 2 respectively) the relationship between the population of the object type $\mathsf{TouristPlace}_{\mathsf{T}_1}$ and $\mathsf{TouristSite}_{\mathsf{T}_2}$ is not disjoint, as we assume some museums and monuments are represented in both databases. The type of the relationship between these object types is intersection because the subtypes of $\mathsf{TouristSite}_{\mathsf{T}_2}$, Theatre and Walk, are not modeled in T_1. And, there is a subtype of $\mathsf{TouristPlace}_{\mathsf{T}_1}$ for which there is no corresponding subtype in $\mathsf{TouristSite}_{\mathsf{T}_2}$, i.e., the Curiosity subtype. The DL expression stating the intersection of the populations of $\mathsf{TouristPlace}_{\mathsf{T}_1}$ and $\mathsf{TouristSite}_{\mathsf{T}_2}$ is $\mathsf{SharedTouristSite} \sqsubseteq \mathsf{TouristPlace}_{\mathsf{T}_1} \sqcap \mathsf{TouristSite}_{\mathsf{T}_2}$. The populations of $\mathsf{Museum}_{\mathsf{T}_1}$ and $\mathsf{Museum}_{\mathsf{T}_2}$, $\mathsf{Monument}_{\mathsf{T}_1}$ and $\mathsf{Monument}_{\mathsf{T}_2}$ are included in each other, i.e., in description logics - $\mathsf{Museum}_{\mathsf{T}_2} \sqsubseteq \mathsf{Museum}_{\mathsf{T}_1}$, and $\mathsf{Monument}_{\mathsf{T}_2} \sqsubseteq \mathsf{Monument}_{\mathsf{T}_1}$.

Already at the level of the *Schema population Correspondences* (SCs), which are the most general correspondences, we can define a set of possible schematic solutions for the integrated schema. Structural patterns that potentially can be applied for constructing the integrated schema, correspond to two decision types that can be taken by the integrated schema designer. The first decision could be to merge overlapping populations. For this decision the inter-schema mappings should be formulated for all related elements and possible structural patterns should be verified. For the second type of decision, where the populations are not merged and therefore, there are no structural transformations to be done, the schema designer is provided with the multi-representation solution. In this case the related populations should be correctly stamped, each instance

with the stamp(s) representing the source(s) database(s) they come from; formulated inter-schema mappings become the integrity constraints for the integrated schema. For example, the SC $\mathsf{Monument}_{\mathsf{T}_2} \sqsubseteq \mathsf{Monument}_{\mathsf{T}_1}$ would constraint the insert operation by inserting the same instance of Monument in T_1 if an instance of Monument is inserted in T_2.

With the decision required structural changes, the structural pattern cannot be chosen based only upon the relationship between populations, the next essential factor is the *integratability* of the related populations. In other words, the possibility to formulate a valid mapping rule for each related representation in the local schemas. In the set of inter-schema mappings there are two types of correspondences that we call *Property semantic Correspondences* and *Matching Rules* that together with the integrity constraints are meant to assess the integratability of the source schemas. In the next subsection we discuss in more detail *Property semantic Correspondences* and *Matching Rules*.

Property semantic Correspondences and Matching Rules. With the *Property semantic Correspondences* (PCs), the schema designer states the relationships between different representations (or part of representations) of the intentionally or extensionally same object types. These correspondences are formulated for all the types of the *Schema population Correspondences* (SCs) including disjoint ones, because the PCs relate intentional representations of the object types. The alphabet of the language for the PCs consists in the attribute names of the schema elements involved in the SCs, in other words, the PCs unfold the SCs expressions.

The temporality (lifecycle) of an object is translated in DL by using the predefined role hasDuration and the spatiality by using the role hasArea as defined in [28]. We assume that museums in T_1 have the role hasDuration. In T_2, the temporality is defined through a temporal attribute openTime. Thus in T_2, the museums have a role openTime whose domain is a temporal domain. To express the constraint that says that $\mathsf{openTime}_{\mathsf{T}_2}$ of $\mathsf{Museum}_{\mathsf{T}_2}$ is temporally equal to the temporality of $\mathsf{Museum}_{\mathsf{T}_1}$ we first have to define two roles based on temporal predicates as in [28] :

$\mathsf{museum\_equal}_1 \equiv \exists(\mathsf{openTime}_{\mathsf{T}_2})(\mathsf{hasDuration}).\mathsf{equal}$;
$\mathsf{museum\_equal}_2 \equiv \exists(\mathsf{hasDuration})(\mathsf{openTime}_{\mathsf{T}_2}).\mathsf{equal}$;

Then the constraint is defined as:

$\mathsf{Museum}_{\mathsf{T}_2} \sqcap \mathsf{Museum}_{\mathsf{T}_1} \sqsubseteq \exists \mathsf{museum\_equal}_2.\mathsf{Museum}_{\mathsf{T}_2} \sqcap$
$\exists \mathsf{museum\_equal}_1.\mathsf{Museum}_{\mathsf{T}_1}$;

To express spatial equality of $\mathsf{TouristPlace}_{\mathsf{T}_1}$ and $\mathsf{TouristSite}_{\mathsf{T}_2}$ - both object types have spatial extensions, we state the following expression in DL:

$\mathsf{area\_equal} \equiv \exists(\mathsf{hasArea})(\mathsf{hasArea}).\mathsf{equal}$;
$\mathsf{TouristPlace}_{\mathsf{T}_1} \sqcap \mathsf{TouristSite}_{\mathsf{T}_2} \sqsubseteq$
$\exists \mathsf{area\_equal}.\mathsf{TouristPlace}_{\mathsf{T}_1} \sqcap \exists \mathsf{area\_equal}.\mathsf{TouristSite}_{\mathsf{T}_2}$;

The rest of the PCs for attributes of $\mathsf{TouristPlace}_{\mathsf{T}_1}$ and $\mathsf{TouristSite}_{\mathsf{T}_1}$ are listed below:

$\mathsf{monument\_equal}_1 \equiv \exists(\mathsf{hasDuration})(\mathsf{construct}_{\mathsf{T}_2}).\mathsf{equal}$;
$\mathsf{monument\_equal}_2 \equiv \exists(\mathsf{construct}_{\mathsf{T}_2})(\mathsf{hasDuration}).\mathsf{equal}$;
$\mathsf{Monument}_{\mathsf{T}_2} \sqcap \mathsf{Monument}_{\mathsf{T}_1} \sqsubseteq \exists \mathsf{monument\_equal}_2.\mathsf{Monument}_{\mathsf{T}_2} \sqcap$
$\exists \mathsf{monument\_equal}_1.\mathsf{Monument}_{\mathsf{T}_1}$;
$\forall \mathsf{name}^{-1}_{\mathsf{T}_1}.\mathsf{CityBorough}_{\mathsf{T}_1} \equiv \forall \mathsf{district}^{-1}_{\mathsf{T}_2}.\mathsf{TouristSite}_{\mathsf{T}_2}$;
$\forall \mathsf{name}^{-1}_{\mathsf{T}_1}.\mathsf{TouristPlace}_{\mathsf{T}_1} \equiv \forall \mathsf{name}^{-1}_{\mathsf{T}_2}.\mathsf{TouristSite}_{\mathsf{T}_2}$;

The set of the PCs is complete if for all the SCs stated for the source schemas, all the pairs of elements (attributes) of object and relationship types involved, are examined for the existence of a PC between them. Completeness is ensured by the DL reasoning service. To complete the set of PCs that are initially proposed by the integrated schema designer, an additional set based on the source schema descriptions and the set of the inter-schema mappings, is deduced by the reasoner. Completeness of reasoning means in this context that no valid deduction is left out by the inference engine. In the complete set of the PCs the designer can now state a subset called *Matching Rules* (MRs). The MRs are the rules that state the correspondences between instances that are represented differently in source schemas. These rules involve identifier attributes. Matching rules are useful in order to find corresponding data during the data integration process. For our example, the MRs between $\mathsf{TouristPlace}_{\mathsf{T}_1}$ and $\mathsf{TouristSite}_{\mathsf{T}_2}$ are those involving $\mathsf{Name}_{\mathsf{T}_2}$, $\mathsf{Name}_{\mathsf{T}_1}$ attributes and the spatiality of the object types.

$\mathsf{TouristPlace}_{\mathsf{T}_1} \sqcap \mathsf{TouristSite}_{\mathsf{T}_2} \sqsubseteq$
$\quad \exists \mathsf{area\_equal}.\mathsf{TouristPlace}_{\mathsf{T}_1} \sqcap \exists \mathsf{area\_equal}.\mathsf{TouristPlace}_{\mathsf{T}_2}$;
$\forall \mathsf{Name}^{-1}_{\mathsf{T}_1}.\mathsf{TouristPlace} \equiv \forall \mathsf{Name}^{-1}_{\mathsf{T}_2}.\mathsf{TouristSite}_{\mathsf{T}_2}$;

Validation in DL. As it was mentioned above, we use DL reasoning services to check the satisfiability of our DL model, i.e., the compatibility of the two source schemas, and the set of inter-schema mappings expressed in DL. If our model is found to be unsatisfiable, then the set of the inter-schema mappings should be reconsidered for unsatisfied objects (*Phase 2* in Fig. 4). Unsatisfiability means that there are some concepts that describe an empty set of instances. For our example an unsatisfiable model would be detected for the following set of definitions:

$\mathsf{Stop}_{\mathsf{T}_1}$'s are spatially connected to $\mathsf{BusLine}_{\mathsf{T}_1}$'s:

$\mathsf{Stop}_{\mathsf{T}_1} \sqsubseteq \exists \mathsf{stopServes}_{\mathsf{T}_1}.\mathsf{BusLine}_{\mathsf{T}_1}$;
$\mathsf{stopServes}_{\mathsf{T}_1} \sqsubseteq \mathsf{connected}$;

$\mathsf{Stop}_{\mathsf{T}_2}$'s are spatially connected to $\mathsf{TransportLine}_{\mathsf{T}_2}$'s:

$\mathsf{Stop}_{\mathsf{T}_2} \sqsubseteq \exists \mathsf{along}_{\mathsf{T}_2}.\mathsf{TransportLine}_{\mathsf{T}_2}$;

$\mathsf{along}_{\mathsf{T}_2} \sqsubseteq \mathsf{connected}$;

Some stops are represented in both databases described by T_1 and T_2:
$\mathsf{Stop}_{\mathsf{T}_1} \sqsubseteq \mathsf{Stop}_{\mathsf{T}_2}$;

There is no $\mathsf{TransportLine}_{\mathsf{T}_2}$ that is spatially connected to a $\mathsf{BusLine}_{\mathsf{T}_1}$:
$\mathsf{area\_disjoint} \equiv \exists(\mathsf{hasArea})(\mathsf{hasArea}).\mathsf{disjoint}$;
$\mathsf{TransportLine}_{\mathsf{T}_2} \sqsubseteq \exists \mathsf{area\_disjoint}.\mathsf{BusLine}_{\mathsf{T}_1}$;

This model will be invalidated by the reasoner based on the following inferences: firstly, since BusLine and Stop from schema T_1 are spatially connected then, TransportLine and Stop from schema T_2 are also spatially connected. Furthemore, some Stops from T_1 and T_2 are the same, and consequently, some of the $\mathsf{BusLine}_{\mathsf{T}_1}$ are spatially connected to $\mathsf{TransportLine}_{\mathsf{T}_2}$. This last deduction of the reasoner contradicts the last expression of the model above.

Upon completion of this phase, the schema designer will have in hand a complete and valid set of inter-schema mappings. We are now able to define a set of possible structural solutions for the integrated schema from *Schema population Correspondences*. In the next phase (*Phase 3* in Fig. 4), different schematic patterns will be validated against the compatibility of integrity constraints for the integrated solutions.

4.2 Structural solution for the integrated schema

Proposed schematic patterns for the integrated schema suggest application of a particular structural transformation of the schema elements involved in the inter-schema mappings. These structural transformations should be validated for the integrity of the resulting schema. The question to be answered is, whether these transformations would lead to a violation of the integrity constraints imposed on one or several schemas' elements. If the planned structural transformation is not valid for the given integrity constraints, then the integrity constraints are weakened, or another structural solution is proposed, and the check is run again. To ensure the meaningful integrated solution even for the cases of greatly diverse representations of related data we employ the multi-representation solution consistently preserving the initial representations on the integrated level.

In the following sections we will consider the integration of two object types, $\mathsf{TouristPlace}_{\mathsf{T}_1}$, and $\mathsf{TouristSite}_{\mathsf{T}_2}$. We assume the following set of correspondences is stated (as explained in Sect. 4.1):
$\mathsf{area\_equal} \equiv \exists(\mathsf{hasArea})(\mathsf{hasArea}).\mathsf{equal}$;
$\mathsf{museum\_equal}_1 \equiv \exists(\mathsf{openTime}_{\mathsf{T}_2})(\mathsf{hasDuration}).\mathsf{equal}$;
$\mathsf{museum\_equal}_2 \equiv \exists(\mathsf{hasDuration})(\mathsf{openTime}_{\mathsf{T}_2}).\mathsf{equal}$;
$\mathsf{monument\_equal}_1 \equiv \exists(\mathsf{hasDuration})(\mathsf{construct}_{\mathsf{T}_2}).\mathsf{equal}$;
$\mathsf{monument\_equal}_2 \equiv \exists(\mathsf{construct}_{\mathsf{T}_2})(\mathsf{hasDuration}).\mathsf{equal}$;

Schema population correspondences:
(1) $\mathsf{SharedTouristSite} \sqsubseteq \mathsf{TouristPlace}_{\mathsf{T}_1} \sqcap \mathsf{TouristSite}_{\mathsf{T}_2}$;

(2) $\mathsf{Museum}_{\mathsf{T}_2} \sqsubseteq \mathsf{Museum}_{\mathsf{T}_1}$;
(3) $\mathsf{Monument}_{\mathsf{T}_2} \sqsubseteq \mathsf{Monument}_{\mathsf{T}_1}$;

Property semantic correspondences:

(4) $\mathsf{Museum}_{\mathsf{T}_2} \sqcap \mathsf{Museum}_{\mathsf{T}_1} \sqsubseteq \exists \mathsf{museum\_equal}_2.\mathsf{Museum}_{\mathsf{T}_2} \sqcap \exists \mathsf{museum\_equal}_1.\mathsf{Museum}_{\mathsf{T}_1}$;
(5) $\mathsf{Monument}_{\mathsf{T}_2} \sqcap \mathsf{Monument}_{\mathsf{T}_1} \sqsubseteq \exists \mathsf{monument\_equal}_2.\mathsf{Monument}_{\mathsf{T}_2} \sqcap \exists \mathsf{monument\_equal}_1.\mathsf{Monument}_{\mathsf{T}_1}$;
(6) $\forall \mathsf{name}_{\mathsf{T}_1}^{-1}.\mathsf{CityBorough}_{\mathsf{T}_1} \equiv \forall \mathsf{district}_{\mathsf{T}_2}^{-1}.\mathsf{TouristSite}_{\mathsf{T}_2}$;

Matching Rules within the *Property semantic correspondences:*

(7) $\mathsf{TouristPlace}_{\mathsf{T}_1} \sqcap \mathsf{TouristSite}_{\mathsf{T}_2} \sqsubseteq \exists \mathsf{area\_equal}.\mathsf{TouristPlace}_{\mathsf{T}_1} \sqcap \exists \mathsf{area\_equal}.\mathsf{TouristSite}_{\mathsf{T}_2}$;
(8) $\forall \mathsf{name}_{\mathsf{T}_1}^{-1}.\mathsf{TouristPlace}_{\mathsf{T}_1} \equiv \forall \mathsf{name}_{\mathsf{T}_2}^{-1}.\mathsf{TouristSite}_{\mathsf{T}_2}$;

Schematic Patterns. The set of possible schematic patterns depends on the type of the SCs between the related representations. From the spectrum of the structural patterns [31], the integrated schema designer is provided with several patterns for validation. For the context of this paper we chose four structural patterns: fusion - the one resulting in the least number of schema elements for the integrated schema; generalization-partition - the one that produces the most detailed integrated schema; and two types of multi-representations that relate source schemas without changing their structures. For our example schemas, the population correspondence on TouristSite and TouristPlace is intersect (as per assertion (1)), and hence the designer will be provided with all four patterns. The set of available patterns would be different for diferent operators in the population correspondence expression. For example, with the disjoint operator, the generalization-partition pattern is excluded as its application requires common instances in related populations.

The first solution (Fig. 6.a) is to extract the overlapping part of the populations and model it as the subtype of the two source populations. This policy uses the multi-inheritance paradigm of the MADS data model. This pattern is called **generalization-partition**. With this structural pattern, the population of the $\mathsf{SharedTouristSite}_{\mathsf{T}_{int}}$ is $\mathsf{TouristPlace}_{\mathsf{T}_1} \sqcap \mathsf{TouristSite}_{\mathsf{T}_2}$. The population of the $\mathsf{SharedTouristSite}_{\mathsf{T}_{int}}$ are those tourist sites (only of subtypes Museum and Monument) that are close to a public transport stop, i.e., accessible by the public transport.

According to the schema population correspondences (1), (2) and (3), we have an integrated representation for common entities SharedMonument and SharedMuseum for schemas T_1 and T_2. The subtype Curiosity (as well as Theatre and Walk) is not present as a subtype of $\mathsf{SharedTouristSite}_{\mathsf{T}_{int}}$ because there is no entities of this type neither in $\mathsf{Curiosity}_{\mathsf{T}_1} \sqcap \mathsf{TouristSite}_{\mathsf{T}_2}$ nor in the $\mathsf{TouristSite}_{\mathsf{T}_2} \sqcap \neg \mathsf{Curiosity}_{\mathsf{T}_1}$.

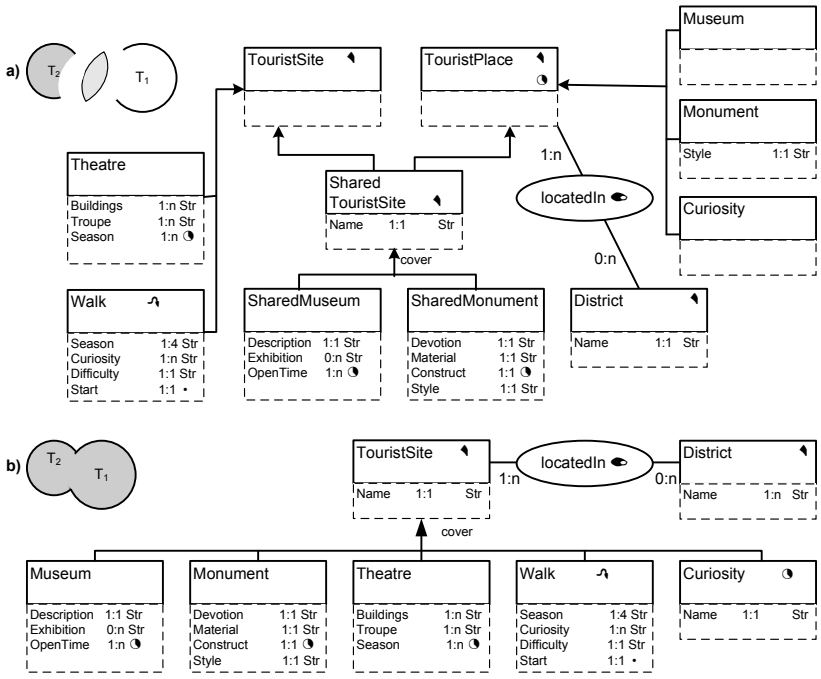

Fig. 6. Schematic solutions under the intersection relation between the populations of the source schemas for integrated schema T_{int}.

From the correspondence assertion stating that the name of the city borough is equal to the district of the TouristSite (assertion (6)), the designer should decide whether he chooses to keep the modeling solution of T_1 - with an object type CityBorough or District, or the modeling solution of T_2 - with an attribute district. In Fig. 6.a, city boroughs are modeled by a spatial object type District with Name attribute. The cardinality of the locatedIn relationship is preserved as it is in schema T_1 - a tourist site can be located in several city boroughs. Such a cardinality would be required for example for the Opera de Paris theatre, that has two buildings, one is in the 9^{eme} city borough, and another in the 12^{eme}. In schema T_2 the cardinality of the District attribute was 1:1, but preserving this cardinality would invalidate the extension (population) of T_1. Another solution for CityBorough would be to keep it as a multivalued attribute of TouristSite (as it is in T_2), but since in T_1 there are relationships attached to the CityBorough object type, the designer should adhere to the pattern shown in Fig. 6.a. where the relationship locatedIn is linked to TouristSite (as in the source schema T_1 for the object type TouristPlace) and attribute District is removed from TouristSite.

Finally, we have to consider the correspondence assertions (4) and (5) about the temporality of TouristPlace and the temporal attributes of Museum and Monument. To be consistent with the schema T_1, the temporality of TouristPlace should be preserved. Considering the MADS model, several solutions are possi-

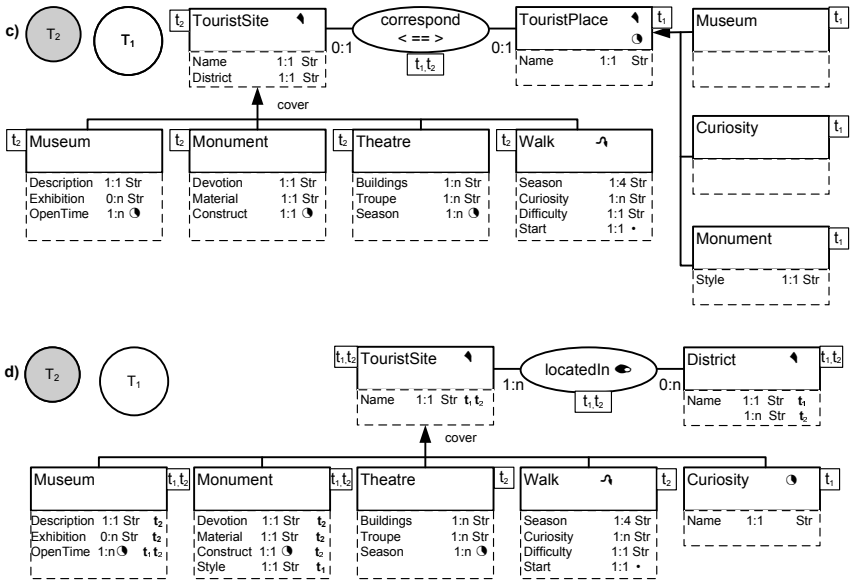

Fig. 7. Multi-representation solutions under the intersection relation between the populations of the source schemas for integrated schema T_{int}.

ble for the temporal attributes openTime and construct : we could either remove them as the temporality of TouristPlace will be inherited in Museum and Monument, or define them as derived attributes (derived from the inherited temporality) or finally, keep them and add an integrity constraint. In Fig. 6.a, we choose to present the second possibility with derived attributes to keep the resulting schema more detailed.

The second possible structural pattern (Fig. 6.b) is **fusion** where the populations of the source schemas are merged. As previously, before giving the final schema, the designer has to consider the correspondence assertions (6), (4) and (5). The proposed solution for the CityBorough is the same as in the first pattern for the same reasons. Considering the temporality of the TouristSite object type, the situation and the proposed pattern is different from above: the temporality of $\mathsf{TouristPlace}_{\mathsf{T}_1}$ is migrated one level down (TouristPlace no longer has a temporality but all its subtypes have one), because in T_2 there are more subtypes for TouristSite and not all of them have temporal attributes. In addition, usage of the redefined temporal attributes $\mathsf{OpenTime}_{\mathsf{T}_{int}}$ and $\mathsf{Construct}_{\mathsf{T}_{int}}$ is more expressive for the schema user than would be the inherited temporality (as it was in the source schema T_1).

Finally, the third and fourth possible structural solutions are the **multi-representations** shown in Fig. 7, where the initial representations and local integrity constraints are preserved and no structural transformation is done. This pattern can be applied in the situation where all other proposed patterns are invalidated.

Two possible modeling solutions may be considered: the designer could either choose to link the different object types under consideration with a link holding the specific inter-representation semantics (as in Fig. 7.c) or integrate the different representations in a multi-representation object type (as in Fig. 7.d). The last solution is structurally the same as the fusion but all the schema elements hold the stamps characterizing the schema from where they come: t1 for elements described in T_1, and t2 for elements from T_2. Thus, the object type TouristSite holds the stamps t1, t2 as it is defined in both schemas (with a different name but the same semantic) whereas Curiosity bears only the stamp t1 as it is only described in T1. When considering the object Monument stamped t1, t2, its attributes Devotion and Material are stamped t2 as these attributes are only described in the schema T_2, Style is stamped t1 and finally Construct is defined in both schemas thus stamped t1, t2. Moreover, the object District is stamped t1, t2 and its attribute Name has a representation-varying definition: for t1 it is a monovalued attribute and for t2 it is a multi-valued attribute.

Validation in DL. The compatibility of integrity constraints is checked if the object types under constraints are involved in a SC and according to the chosen structural pattern, the representations of the two concepts are merged (totally or partially). The component ICs must be checked to deduce a common, *global* ICs guaranteeing validity of the resulting *global* ICs. The result of the validation procedure determines if we can define valid integrity constraints for merged object types and consequently for the whole integrated schema.

For our example schemas, assume that a schematic pattern **fusion** proposed for the object types $\mathsf{TransportLine}_{\mathsf{T}_2}$ and $\mathsf{BusLine}_{\mathsf{T}_1}$ with integrated object type is $\mathsf{TransportLine}_{\mathsf{T}_{int}}$ which is defined as $\mathsf{TransportLine}_{\mathsf{T}_{int}} \sqsubseteq \mathsf{BusLine}_{\mathsf{T}_1} \sqcap \mathsf{TransportLine}_{\mathsf{T}_2}$. From the definition of the schema T_1 the reasoner would find that $\mathsf{BusLine}_{\mathsf{T}_1}$ has a role $\mathsf{stopServes}_{\mathsf{T}_1}$ (Fig. 1) with the cardinalities $\geq 1\mathsf{stopServes}^{-1}_{\mathsf{T}_1}.\mathsf{Stop}_{\mathsf{T}_1}$ and $\geq 1\mathsf{stopServes}_{\mathsf{T}_1}.\mathsf{BusLine}_{\mathsf{T}_1}$. On the other side, from the definition of the schema T_2 (Fig. 2) the reasoner would find that $\mathsf{TransportLine}_{\mathsf{T}_2}$ has a role $\mathsf{along}_{\mathsf{T}_2}$ with the cardinalities $\geq 1\mathsf{along}^{-1}_{\mathsf{T}_2}.\mathsf{Stop}_{\mathsf{T}_2}$ and $\geq 2\mathsf{along}_{\mathsf{T}_2}.\mathsf{TransportLine}_{\mathsf{T}_2}$. Thus, the definition for the integrated concepts $\mathsf{Stop}_{\mathsf{T}_{int}}$ and $\mathsf{TransportLine}_{\mathsf{T}_{int}}$ are the following:

$$\begin{aligned}
&\mathsf{TransportLine}_{\mathsf{T}_{int}} \sqsubseteq \\
&\quad \forall \mathsf{along}^{-1}_{\mathsf{T}_{int}}.\mathsf{Stop}_{\mathsf{T}_{int}} \\
&\quad \sqcap \geq 1\mathsf{along}^{-1}_{\mathsf{T}_{int}}.\mathsf{Stop}_{\mathsf{T}_{int}} \\
&\quad \sqcap \geq 2\mathsf{along}^{-1}_{\mathsf{T}_{int}}.\mathsf{Stop}_{\mathsf{T}_{int}} \ ; \\
&\mathsf{Stop}_{\mathsf{T}_{int}} \sqsubseteq \\
&\quad \forall \mathsf{along}_{\mathsf{T}_{int}}.\mathsf{TransportLine}_{\mathsf{T}_{int}} \\
&\quad \sqcap \geq 1\mathsf{along}_{\mathsf{T}_{int}}.\mathsf{TransportLine}_{\mathsf{T}_{int}} \\
&\quad \sqcap \geq 1\mathsf{along}_{\mathsf{T}_{int}}.\mathsf{TransportLine}_{\mathsf{T}_{int}} \ ;
\end{aligned}$$

As the resulting integrated cardinality for $\mathsf{along}_{\mathsf{T}_{\mathsf{int}}}$ the reasoner would propose $\geq 2\mathsf{along}^{-1}_{\mathsf{T}_{\mathsf{int}}}.\mathsf{Stop}_{\mathsf{T}_{\mathsf{int}}}$. The choice of this cardinality as the global one may invalidate a part of the population of $\mathsf{BusLine}_{\mathsf{T}_1}$ because the cardinality of the $\mathsf{along}^{-1}_{\mathsf{T}_1}$ role was 1:n. To meet the cardinality 2:n for all instances of $\mathsf{T}_{\mathsf{int}}$, designer could formulate and execute a query that would fill the reference attribute for $\mathsf{TransportLine}_{\mathsf{T}_{\mathsf{int}}}$ with at least 2 values.

As well, the designer could choose to impose 1:n as the global cardinality of the $\mathsf{along}^{-1}_{\mathsf{T}_{\mathsf{int}}}.\mathsf{Stop}_{\mathsf{T}_{\mathsf{int}}}$. In this case, no population is invalidated, but the semantics of 2 terminus stops is lost. To keep this information for the integrated schema, the designer can formulate a query that finds the 2 terminus stops for each instance of the $\mathsf{TransportLine}_{\mathsf{T}_{\mathsf{int}}}$, so every time the user wants to find two terminus stops for a given line, he/she would execute this query.

4.3 Composing Integrated Schema

By the completion of the validation procedure for the DL integrated schema descriptions, the designer of the integrated schema has valid structural solutions for the related representations assured by the reasoning engine; associated integrity constraints; and mappings for all related elements of the schemas provided by the complete set of inter-schema mappings. In this last phase of the integration process the designer can choose the integrated solutions for each related element of the source schemas and compose the resulting integrated schema.

As it was shown in Sect. 4.2, for each set of mappings, a designer is provided with one or more valid structural solutions (Figures 6 and 7 show possible structural solutions for TouristPlace and TouristSite object types). For the final integrated schema, for each set of mappings for (at least) intentionally related object types, schema designer can choose one of the solutions following a criterion. This criterion is application dependent and could be for example, the complexity of the structural solution, or the type of links used, or the types of queries that will be processed by the information system under development. Considering the structural solutions shown in Figures 6 and 7, the designer could choose the **fusion** as the least complex one, i.e., the one with the least number of elements. As the solution for the bus lines and bus stops representation, the designer could choose to adhere to the **multi-representation** solution as shown in Fig. 8, to avoid the invalidation of the population of $\mathsf{BusLine}_{\mathsf{T}_1}$ object type (cf. the validation section above).

Let us now demonstrate how the mappings can be used in an integrated database. For the part of the schema shown in Fig. 8 the following mappings for the Bus and BusLine object types are relevant:

Schema population correspondences:
(1) $\mathsf{Bus}_{\mathsf{T}_2} \sqsubseteq \mathsf{BusLine}_{\mathsf{T}_1}$;
Matching Rules within the *Property semantic correspondences:*
(2) $\forall \mathsf{number}^{-1}.\mathsf{Bus}_{\mathsf{T}_2} \equiv \forall \mathsf{busNum}^{-1}.\mathsf{BusLine}_{\mathsf{T}_1}$;

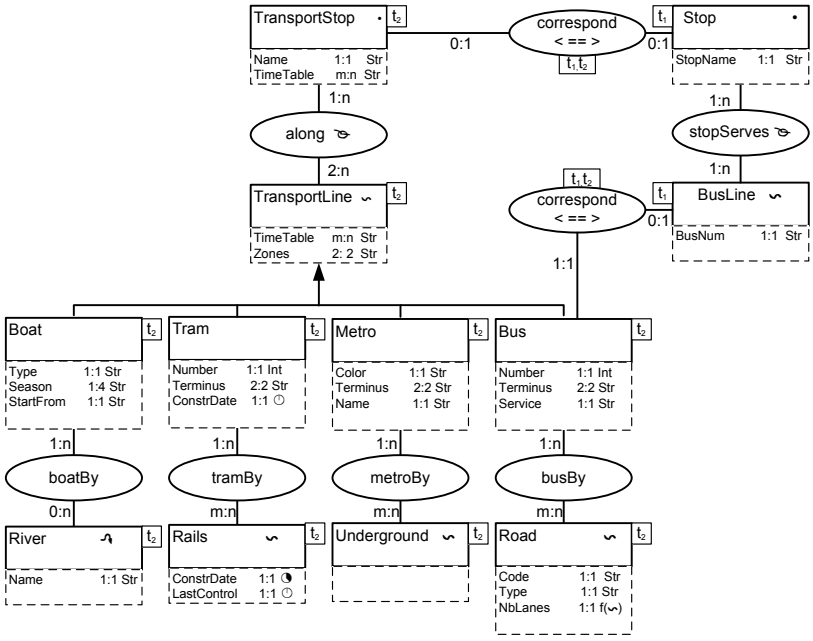

Fig. 8. Multi-representation solution for bus lines and bus stops in an T_{int}.

Let us assume that we use an SQL like query language to query and maintain the database. Mapping (1) requires insertions of an equal instance in the BusLine table every time a new instance is inserted in the Bus table. The equality between the instances of Bus and BusLine is defined by the mapping (2), i.e., the value of the attribute busNum in BusLine must be equal to the value of the attribute number in Bus. Then, the following code will be added to keep the integrated database valid:

```
CREATE TRIGGER busLine
AFTER INSERT ON Bus
   BEGIN
     SELECT busNum FROM BusLine WHERE busNum = NEW.number
       IF SQL%NOTFOUND THEN
       INSERT INTO BusLine VALUES(NEW.number) ;
       ENDIF ;
   END ;
```

Now, every time an insert operation is executed on the Bus table, the trigger will be fired to check the existence of an equal instance in the table BusLine, if it is not found, an insert operation will be executed on the BusLine table, with

the value of the busNum attribute equal to the last inserted value of the number attribute.

5 Conclusion

Database integration has been and continues to be the focus of many research efforts, and is a task much harder by the presence of spatio-temporal aspects. Very few formal approaches have been reported which deal specifically with spatio-temporal databases.

In this paper we propose an approach to integrate spatio-temporal database schemas relying on two well-known formalisms: conceptual models and description logics. We use MADS, an object+relationship conceptual model, intended to describe spatio-temporal application data. A peculiar feature of MADS that is of interest in a data integration environment is that it includes specific concepts to describe multiple representations of data. Indeed, as stated in [12], full integration of spatial database requires a powerful data model for the integrated schema in order not to loose the semantics of the original schemas. Description logics are a family of knowledge representation formalisms, with special support for the definition of terminologies. The first phase of our methodology, Figure 4, consists of defining the source schemas using the MADS conceptual model. Inter-schema mappings are then defined between the source schemas. Since defining inter-schema mappings is an error-prone activity, we need to check the compatibility of the mappings. In *Phase 2*, mappings are thus expressed in Description Logics whose inference mechanisms are used for satisfiability checking. From those validated mappings, integration patterns are proposed in *Phase 3*, and their compatibility against integrity constraints is checked. The designer is subsequently provided with a set of valid patterns, to be used in defining the integrated schema (*Phase 4*).

To further this work we plan to design a framework in which one would be able to follow our methodology and to realize its schema integration task in an assisted way. Our framework will combine existing tools like MADS schema editor [19] to design source schemas enhanced with capabilities to define inter-schema mappings; an automatic translator from MADS to a Description logic formalism; and finally, with a DL reasoner like Racer [32] enhanced with spatio-temporal semantics in order to validate the mappings and the integrated schema.

References

1. Sheth, A., Larson, J.: Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Computer Surveys (1990) 183–236
2. Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J., Widom, J.: Integrating and accessing heterogeneous information sources in TSIMMIS. In: Proc. of AAAI Spring Symposium on Information Gathering, Stanford, California (1995)
3. E.Rahm, Bernstein, P.: A survey of approaches to automatic schema matching. The VLDB Journal (2001) 334–350

4. Kalfoglou, Y., Schorlemmer, M.: Ontology mapping: the state of the art. The Knowledge Engineering Review **18** (2003) 1–31
5. Ehrig, M., Sure, Y.: Ontology mapping - an integrated approach. In: Proc. of the ESWS 2004, Heraklion, Crete, Greece (2004) 76–91
6. Doan, A., Madhavan, J., Domingos, P., Halevy, A.: Ontology matching: A machine learning approach. In Staab, S., Studer, R., eds.: Handbook on Ontologies in Information Systems. Springer-Velag (2004) 397–416
7. Doan, A., Madhavan, J., Domingos, P., Halevy, A.: Learning to map between ontologies on the Semantic Web. In: Proc. of International WWW conference, WWW'02, Honolulu, Hawaii, USA (2002)
8. Dhamankar, R., Lee, Y., Doan, A., Halevy, A., Domingos, P.: iMAP: Discovering Complex Semantic Matches between Database Schemas. In: Proc. of the 2004 ACM SIGMOD International Conference on Management of Data, Paris, France, ACM Press (2004) 383–394
9. Catarci, T., Lenzerini, M.: Representing and using interschema knowledge in cooperative information systems. Journal of Intelligent and Cooperative Information Systems **2** (1993) 375–398
10. Calvanese, D., de Giacomo, G., Lenzerini, M., Nardi, D., Rosati, R.: Information integration: Conceptual modeling and reasoning support. In: CoopIS 1998. (1998) 280–291
11. Calvanese, D., de Giacomo, G., Lenzerini, M.: A Framework for Ontology Integration. In: Proceedings of SWWS'01, The First Semantic Web Working Symposium Stanford University, California, USA (2001) 303–306
12. Devogele, T., Parent, C., Spaccapietra, S.: On spatial database integration. International Journal of Geographic Information Systems, Special Issue on System Integration **3** (1998) 335–352
13. Madhavan, J., Bernstein, P., Domingos, P., Halevy, A.: Representing and reasoning about mappings between domain models. In: Proc. of the AAAI Eighteenth National Conference on Artificial Intelligence, Edmonton, Alberta, Canada, American Association for Artificial Intelligence (2002) 80–86
14. Halevy, A.Y., Ives, Z.G., Mork, P., Tatarinov, I.: Piazza: Data Management Infrastructure for Semantic Web Applications. In: Proc. of International WWW Conference, WWW'03. (2003) 556–567
15. Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., Hübner, S.: Ontology-Based Integration of Information - A Survey of Existing Approaches. In: Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing, Seattle, USA (2001)
16. Fonseca, F., Davis, C., Câmara, C.: Bridging ontologies and conceptual schema in geographical information integration. Geoinformatica **7** (2003) 355–378
17. Hakimpour, F., Geppert, A.: Global schema generation using formal ontologies. In: Proc. of the ER2002. LNCS 2503 (2002) 307–321
18. Fonseca, F., Egenhofer, M., Agouri, P., Câmara, C.: Using ontologies for integrated geographic information systems. Transactions in GIS **6** (2002) 231–257
19. MurMur project: MurMur consortium: Multi-representations and Multiple resolution in geographic databases. http://lbdwww.epfl.ch/e/MurMur (2002)
20. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P., eds.: The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
21. Horrocks, I., Sattler, U., Tobies, S.: Practical reasoning for expressive description logics. Logic Journal of the IGPL **8** (2000) 161–180

22. Berardi, D., Calvanese, D., de Giacomo, G.: Reasoning on UML class diagrams using description logic based systems. In: Proc. of the KI'2001 Workshop on Applications of Description Logics. (2001) CEUR Electronic Workshop Proceedings, http://ceur-ws.org/Vol-44/.
23. Cohn, A.G., Hazarika, S.M.: Qualitative spatial representation and reasoning: An Overview. Fundamenta Informaticae **46** (2001) 1–29
24. Gerevini, A., Nebel, B.: Qualitative spatio-temporal reasoning with RCC-8 and Allen's interval calculus: Computational complexity. In: Proceedings of ECAI'2002, IOS Press (2002) 312–316
25. Wessel, M.: On spatial reasoning with description logics - position paper. In: Proc. of the International Workshop in Description Logics 2002 (DL2002), Touluse, France (2002)
26. Kutz, O., Lutz, C., Wolter, F., Zakharyaschev, M.: $\mathcal{E}$-connections of abstract description systems. Artif. Intell. **156** (2004) 1–73
27. Baader, F., Hanschke, P.: A scheme for integrating concrete domains into concept languages. In: Proc. of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI), Sydney, Australia (1991) 452–457
28. Haarslev, V., Lutz, C., Möller, R.: A description logic with concrete domains and a role-forming predicate operator. Journal of Logic and Computation **9** (1999)
29. Lutz, C.: Description logics with concrete domains—a survey. In: Advances in Modal Logics Volume 4, King's College Publications (2003)
30. Sotnykova, A., Monties, S., Spaccapietra, S.: Semantic integration in MADS conceptual model. In Bestougeff, H., Thuraisingham, B., eds.: Heterogeneous Information Exchange and Organizational Hubs. Kluwer (2002)
31. Dupont, Y.: Resolving Fragmentation Conflicts in Schema Integration. In Loucopoulos, P., ed.: 13th International Conference on the Entity-Relationship Approach, ER'94. Volume 881 of LNCS., Manchester, U.K., Springer (1994) 513–532
32. Haarslev, V., Möller, R.: Racer system description. In: Proc. of International Joint Conference on Automated Reasoning, IJCAR 2001, Springer-Verlag (2001) 701–705