On Languages for the Specification of Integrity Constraints in Spatial Conceptual Models

Mehrdad Salehi [1,2], Yvan Bédard [1,2], Mir Abolfazl Mostafavi [2], Jean Brodeur [2,3]

[1]Canada NSERC Industrial Research Chair in Geospatial Databases for Decision Support
[2]Center for Research in Geomatics, Department of Geomatics Sciences, Laval University, Quebec City, Canada, G1K 7P4
[3]Center for Topographic Information, Natural Resources Canada, Sherbrooke, Canada, J1J 2E8
mehrdad.salehi.1@ulaval.ca, yvan.bedard@scg.ulaval.ca, mir-abolfazl.mostafavi@scg.ulaval.ca, brodeur@nrcan.gc.ca

Abstract. This paper intends to classify, review, and compare the existing languages for the specification of spatial integrity constraints at the conceptual level. We classify these languages into natural, visual, first-order logic, and hybrid and review their syntax and semantics. We compare these language categories based on expressiveness and pragmatics qualities. The results of this study show that controlled natural languages and natural hybrid languages with pictograms are good candidates for expressing spatial integrity constraints at the conceptual level of spatial databases. At the end, we identify new research challenges that should be addressed in this domain in order to facilitate spatial integrity constraint specification and representation.

1 Introduction

A database design process typically starts with a *conceptual model* and moves towards development based on an *implementation model*. A Conceptual model concerns clients' views of the data of interest. In this paper, we consider its abstraction level the same as the Computation Independent Model (CIM) in the Object Management Group (OMG)'s Model-Driven Architecture (MDA) [1]. An implementation model, however, serves as a developer's view of the data for the implementation on a given family of platforms (e.g., transactional DBMS, OLAP servers) or for a specific software package. These respectively correspond to the Platform Independent Model (PIM) and Platform Specific Model (PSM) in MDA. A conceptual model includes two components: (1) *a schema* which shows how the data are structured, and (2) a *data dictionary* which includes additional information about the data [2].

Typically, integrity constraints (ICs) are defined at the CIM level with the client's point of view, and then translated for the PIM and PSM levels. Integrity Constraints describe semantically valid states of the data and preserve database integrity [3]. It is emphasized that ICs convey important semantic information about the database application domain [4]. Hence, it is necessary to specify ICs at the conceptual model

of applications (i.e., CIM) and to make them amenable to implementation via the PIM and PSM. Each of these levels may require a different language to express these ICs. In this paper, we are interested in the languages used for the definition of ICs in spatial conceptual level.

In spatial conceptual models, additional types of ICs, known as Spatial ICs (SICs), must be specified. Examples of these ICs are *topological ICs* restricting topological relationships, e.g., two building do not *overlap* and *non-topological ICs* restricting non-topological relationships (such as metric), e.g., the *distance* between a residential area and a gas station must be more than *30 meters*.

Different kinds of languages for the specification of SICs are proposed in the literature. Despite the existence of research surveys on modeling approaches for spatial database applications (e.g., [5],[6]), there is a gap concerning the investigation of the proposed languages for expressing SICs. This paper intends to review and classify the state-of-the-art SICs specification languages at the conceptual level. In addition, it compares the languages and identifies their strengths and weaknesses which contributes to developing future IC specification languages, e.g., for spatial multidimensional models (datacubes).

To this end, Section 2 reviews, classifies, and discusses the state-of-the-art languages for the specification of SICs. Section 3 compares these categories of languages. Final conclusions and future research requirements are given in Section 4.

2 Classifying and Reviewing the Languages

Various kinds of languages have been developed for the specification of SICs at different abstraction levels. For example, an ontology-driven language named Semantic Web Rule Language (SWRL) is used by [7] for expressing SICs within ontologies. At the implementation level, languages like Spatial SQL [8] are proposed. However, in this paper we focus on the languages used at the *conceptual level* within the spatial database community, i.e., natural languages [9] [10], a visual language [11], a hybrid natural language [12], spatial OCL [13], spatio-temporal modeling languages [14] [4], and first-order logic [15]. In spite of the capability of spatio-temporal conceptual modeling languages to express SICs directly in the conceptual schema, more specific languages, called Integrity Constraint Specification Languages (ICSL) have been introduced. These languages specify SICs in a data dictionary since they cannot be expressed efficiently in a schema. We classify all these possibilities into four main categories, i.e., natural languages, visual languages, first-order logic language, and hybrid languages.

2.1 Natural Languages

People use natural languages (e.g., English and French) for their daily communications. These languages are the simplest and easiest languages for the clients to specify SICs. We classify natural languages into two categories: (1) *Free natural language*, and (2) *Controlled natural language* as described below.

2.1.1 Free Natural Languages

A free natural language is a natural language without implication of any limit on syntax and semantics of the language. *Syntax* of a language describes the way language's symbols should be combined to create well-formed sentences. However, *semantics* reveals the meaning of syntactically correct strings in a language [16]. SICs can be defined using a free natural language. For example, in English: "A road does not cross a building" is the simplest form of specifying SICs. Free natural languages support a rich vocabulary. However, they are sometimes ambiguous or used too loosely. In such languages, several words may bear the same semantics and a word may have several meanings depending on the context. Nevertheless, they remain today's most widely used language for expressing SICs.

2.1.2 Controlled Natural Languages

In order to overcome the ambiguity of free natural languages, controlled natural languages have been proposed. Controlled natural languages are subsets of natural languages whose syntax and semantics have been restricted [17].

For the specification of topological ICs, [10] proposes a controlled natural language in a form similar to predicates. Within this language, the syntax of a topological IC consists of entity class 1, a topological relationship between entity classes characterized by extended 9-intersection model [18] (e.g., inside, cross, join), entity class 2, and a multiplicity quantifier which can be one of the followings: forbidden, at least *n* times, at most *n* times, and exactly *n* times:

```
(Entity class1, Relation, Entity class2, Quantifier)
```

For instance, the following SIC in free natural language "Road cannot cross a Building" is specified using this controlled natural language as follows:

```
(Road, Cross, Building, forbidden)
```

Similarly, [9] introduces a controlled natural language for the definition of topological ICs. In this language, the syntax of a topological IC is composed of object class 1, one of the eight topological relationships *equal*, *disjoint*, *intersect*, *touch*, *cross*, *within*, *contain*, *overlap* (of ISO19107:2003 – Geographic information – Spatial schema) extended by three notions *tangent*, *border*, and *strict*, object class 2, and cardinality of the relationship. For example, in this language, "each road segment should be end-to-end connected to at least one or at most two road segments" is expressed as:

```
Road Segment Touch-Tangent Road Segment [1,2]
```

Where `[-,-]` is the cardinality of the topological IC indicating minimum and maximum number of instances of the two classes defined in the relationship.

2.2 Visual Languages

A visual language employs graphical and image notations to communicate the information with the user. Various visual languages for different purposes have been

proposed; for a survey of these languages refer to [19]. In this section, by visual languages, we mean the languages that use only graphical or image notations. We will study hybrid languages, which combine visual and other languages, in Section 2.4.

For the specification of SICs, [11] present a model and a visual language based on depicted pictures. The pictures show unacceptable database states, termed "constraint pictures". For instance, Figure 1 expresses topological IC "cars and people cannot be in a crosswalk simultaneously". In this figure, people, cars, and crosswalks are represented by asterisks, small boxes, and dashed lines respectively.

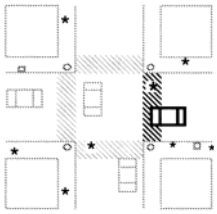

Fig. 1. A visual language which highlights SICs: "cars and people cannot be inside a crosswalk simultaneously" [11].

The aim of visual languages is to create an easy to use and perceive language [20]. However, representing all the information of SICs using a visual language necessitates clients to learn the meaning of every visual construct of the language to be able to use it, and to understand very well the very specific context of it usage, which requires a big effort. In spite of such efforts, several ambiguities and unintended meanings can emerge from such representations.

2.3 First-order Logic Language

First-order logic language is a common formal language for representing ICs in databases and knowledge bases. First-order logic allows quantification on objects, i.e., the first-order entities exist in the real world.

Hadzilacos and Tryfona [15] describe a formal model with first-order logic language for expressing topological ICs at the conceptual level. In this language, the syntax of a topological sentence is built of atomic topological formulae with negation ($\neg$), conjunction ($\wedge$), disjunction ($\vee$), and universal ($\forall$) and existential ($\exists$) quantification. Atomic topological formulae consist of topological relations between objects based on binary topological relationships [21], geometric operators over objects, and comparison between attributes of objects. As an example, topological IC "roads and buildings are disjoint" in first-order logic is:

$$\forall(r,b)[\text{disjoint}(r,b)]$$

Where r and b stand for "road" and "building" respectively, and *disjoint* is a topological relationship.

First-order language supports precise semantics and syntax which avoids error in interpretation. Nevertheless, understanding this language requires a mathematical background. Clients do not necessarily have such a background.

2.4 Hybrid Languages

A number of languages used for the specification of SICs are not purely natural, visual, or logical and can be best described as a combination of them. We call such languages *hybrid languages*. Depending on the dominant component of a language, a hybrid language can be *visual hybrid* or *natural hybrid*.

2.4.1 Visual Hybrid Languages

The main part of visual hybrid languages consists of visual symbols, which are enriched by a natural language. As a limited number of visual constructs are easy to perceive [20], visual hybrid languages combine this advantage with the richness of natural languages. Unified Modeling Language (UML) is such a language.

As previously stated, a conceptual model consists of conceptual schema and data dictionary. We did not find any visual hybrid ICSL for defining SICs in data dictionary. However, spatio-temporal conceptual modeling languages, such as Perceptory [14] and MADS [4], are among visual hybrid languages that can express a number of ICs (e.g., topological and synchronization constraints) in a conceptual schema. Most of these modeling languages provide spatial and temporal data types or object stereotypes with associated icons. While some of these languages provide a number of spatial and temporal topological relationships (e.g., disjoint, overlap, starts, finish), others leave it up to the client to select the most appropriate relationships based on their field vocabulary (e.g., homemade 3D set of relationships, spatio-temporal relationships). These constructs can be employed for defining SICs.

For instance, a conceptual schema for a road network application is likely to consist of object classes such as “route” and “roundabout” classes, their attributes, and the relationships between classes, e.g., each roundabout is crossed by at least one route. This schema relationship “cross” between “roundabout” and “route” expresses a SIC. The expression of this SIC using a visual hybrid modeling language Perceptory [14] is shown in Figure 2. In this figure, “route” class and “roundabout” class hold geometric attribute of type line and polygon. The SIC between two classes is expressed by the relationship between two classes.

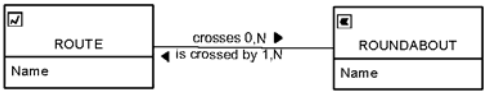

Fig. 2. SIC using visual hybrid modeling language Perceptory at the conceptual schema

The elements of schema are necessary for maintaining database consistency. However, they cannot express all the possible SICs (e.g., the minimum allowable distance between two entities) as simply, easily, and shortly as ICSL which are designed specially for this purpose. Moreover, SICs are detailed information and may end up over hundreds of times the number of object classes in a schema [14]. Hence, systematically representing them in a conceptual schema contradicts the rule that conceptual schema should be concise enough to remain readable. Thus, visual hybrid modeling languages are not sufficient to be used alone for defining all SICs in a practical way.

2.4.2 Natural Hybrid Languages

A natural hybrid language consists of a natural language as well as pictograms or symbols, but the natural language part is dominant.

2.4.2.1 Natural hybrid language with pictograms

A natural hybrid language with pictograms consists of a natural language which is enriched by pictograms [22]. In the spatial database community, pictograms are intuitive symbols employed for the representation of objects' geometries. Normand [12] proposes a natural hybrid language with a limited number of pictograms to overcome the limitations of visual hybrid modeling languages in the specification of SICs. This language can specify all types of SICs between object classes, including the SICs that depend on specific attribute values or on objects' geometries (when alternate or multiple geometries are involved). In this language, class names, topological relationships, and cardinality of relationships are specified by controlled natural language. Additionally, pictograms ⊡ ◪ ⧈ etc., stand for point, line, polygon, and more complex geometries (see [14] for all supported cases) of the classes. The language uses a table for each object class to express its SICs with regards to all other object classes (the class name is at the top of the table). The table consists of six columns: "Dimension Object 1" shows the geometry of object 1 (there maybe several in case of complex geometries), "Operator" describes logical operator (AND, OR, XOR) when more than one spatial relationship is possible between two classes, "Relations" addresses the spatial relationship between two classes based on 9 intersection model [23] or on metric constraints (e.g., maximum distance), "Cardinality" expresses the minimum and maximum occurrences of this relationship, "Object 2" represents the name of object class 2 (there may be several), and "Dimension Object 2" holds the geometry of object 2 (there may also be several in case of complex geometries for object 2). This method shows all its power when complex SICs or several SICs are involved for an object class. For instance, Figure 3 shows a simple SIC between "Canal" and "Watercourse" classes which support multiple representation for "Watercourse" class.

Canal

Dimension Object 1	Operator	Relations	Cardinality	Object 2	Dimension Object 2
◪	OR	Disjoint	-	Watercourse	◪
		Adjacent	1-2		
	OR	Disjoint	-		⧈
		Adjacent	1-2		

Fig. 3. Specification of a topological IC using Normand's method [12].

Figure 3 describes that a linear canal must be disjoint from a linear watercourse or must be adjacent to one or at last two linear watercourses. In addition, a linear canal must be disjoint from a polygonal watercourse or must be adjacent to one or at last two polygonal watercourses.

2.4.2.2 Natural hybrid language with symbols

It is possible to add a number of specific symbols to a controlled natural language to make it more abstract than a natural language for knowledge representation. Object Constraint Language (OCL), which is used along UML and developed by the OMG,

is such a language. This language allows expressing constraints and queries on UML class diagrams. OCL is a formal language designed to overcome the ambiguity of natural languages and the disadvantage of formal languages which are usable only for individuals with a mathematical background [24].

In order to precisely define topological ICs along with UML for conceptual modeling of spatial databases, [13] proposes a spatial extension of OCL. The authors in [13] add new basic geometric primitives, i.e. `point`, `polyline`, and `polygon`, generalized by `BasicGeoType` to the OCL meta-model. Furthermore, for describing topological ICs based on 9 intersection method [23], the following new OCL operators: `disjoint`, `meet`, `contains`, `covers`, `inside`, `coveredBy`, `equal`, and `overlap` are defined and added to OCL. For specifying the SIC "all buildings are disjoint from all roads" one can write:

```
context Building inv:
Road.allInstances()->forAll(R|R.geometry->Disjoint self.geometry))
```

Where "context" specifies the model entity for which the OCL expression is defined; `allInstances` and `forAll` are universal and existential operators respectively. "R" refers to one instance of Road class. The keyword `self` returns an instance of the class declared in the "context." Although less abstract than the mathematically formal languages, it remains a language with a low-level of readability from the clients' point of view.

3 Comparing Languages for the Specification of SICs

Inspired by the work of [16], [25], and [26], by 20 years of research and experimentation of our team in real projects, and by the final objective of our research, we analyzed the languages discussed in Section 2 by focusing on the two aspects: expressiveness and pragmatics.

Expressiveness takes into consideration semantic quality (the degree of correspondence between the concepts that a language supports and the meaning of SICs in the real world), syntactic quality (the degree to which rules govern the structure of expressions), richness (the capability to express clearly the needed elements of SICs), and inherence (the precision of a language to be straight to the point and focus on essential aspects of SICs).

Pragmatics, in the context of conceptual modeling, refers to two qualities. First, it refers to the usability of the language by those who must express the domain-related constraints that are needed for the system under development, and also who must validate the database designer's understanding of these constraints, that is the clients (i.e., the end-users of the system, not the computer). Second, pragmatics also refers to the facility to translate (by the database designer when necessary) these ICs in a technical language amenable to computerization. In our context of expressing ICs for conceptual modeling, the former pragmatic quality has priority over the latter since experience has taught us that we have higher chances of success when we ask the database designer to translate users requirements into technical languages for implementation than when we ask the clients to learn technical languages to express

and validate themselves their database requirements. In other words, we consider that a good pragmatic quality lets the client focus on the ICs, not on learning a new language (N.B. the client's involvement into database development is a *ad hoc* activity taking place only once or very few times in a career). Using OMG's MDA vocabulary, we are targeting the CIM and PIM levels rather than the PSM level.

As this research is in progress, in this paper we compare the categories of the languages instead of their instances. In addition, since we are not aiming at finding "the best" language (if such a thing exists) but rather at summarizing potential avenues for our research on SICs for spatial datacubes, we focused on expressiveness and pragmatics as two general indicators rather than considering a larger series of detailed criteria for a very precise comparison. Thus, this research is not an end *per se* but a mean to better attack ICSL for spatial datacubes.

Table 1 shows the evaluation results of the languages' categories with respect to expressiveness and pragmatics, and ranks them by "good", "medium", and "weak". The values in this table represent our opinion resulting from a study of the literature and of 20 years of experience in several industrial projects involving spatial databases, spatio-temporal databases, and spatio-temporal datacubes in diverse fields and implementations with very diverse technologies.

Table 1. Our opinion about the categories of languages for the specification of SICs.

Language		***Expressiveness***	***Pragmatics***
Natural	Free	Medium	Good
	Controlled	Good	Good
Visual		Weak	Medium
First-Order Logic		Good	Weak
Visual Hybrid		Good	Medium
Natural Hybrid	With Pictograms	Good	Good
	With Symbols	Good	Medium

Considering the above table, controlled natural languages and natural hybrid languages with pictograms seem more suitable ICSL for the conceptual level. Although visual hybrid languages could be considered good candidates, the existing proposed visual hybrid languages typically cope with SICs in the conceptual schema. Consequently, due to the large number of ICs and their level of complexity, [14] suggested defining ICs in data dictionary in a complementary manner to those defined in the schema in order to prevent the schemas from becoming too complex and unreadable. Thus, conceptual modeling languages can be used to a certain degree if completed by detailed SICs described in another language.

Finally, the pragmatics values change according to the target audience. For example, when moving from conceptual to implementation modeling, different suggestions come out in such cases. Therefore, one will typically have to deal with more than one ICSL to completely cover the development process from needs analysis to design, implementation, operation, and updating of a database.

4 Conclusions and Future Research

In this paper, we categorized the languages for the specification of SICs at the conceptual level into: natural (free and controlled), visual, first-order logic, and hybrid (visual, natural with pictograms, and natural with symbols) languages. We discussed on state-of-the-art languages and compared the categories of languages based on their expressiveness and pragmatic qualities. The results indicate that controlled natural languages and natural hybrid languages with pictograms are the best choices as ICSL at the conceptual level. One may also expect to see a different result for the implementation level and have to deal with more than one ICSL.

The existing spatial ICSL presented in this paper have been developed for transactional databases. However, to specify ICs of spatial datacubes, additional syntax and semantics concerning multidimensional conceptual models should be taken into account [27]. Our future research is focused on developing ICSLs for spatial multidimensional databases at the conceptual level. This ICSL can be an extension to a discussed language in this paper. Therefore, our next step is to compare into more details the quality of the proposed languages' instances.

Acknowledgments. The authors would like to acknowledge the financial support from the Canada NSERC Industrial Research Chair in Geospatial Databases for Decision Support. We are also grateful to the three anonymous reviewers for their useful comments on this paper.

References

1. Miller, J., Mukerji, J. : MDA Guide Version 1.0, OMG Document (2003)
2. Brodeur, J., Bédard, Y., Proulx, M.J.: Modeling Geospatial Application Databases using UML-based Repositories Aligned with International Standards in Geomatics. In: Proceedings of Eighth ACM Symposium on Advances in Geographic Information Systems (ACMGIS), Washington D.C, pp. 39-46 (2000)
3. Godfrey, P., Grant, J., Gryz, J., Minker, J.: Integrity Constraints: Semantics and Applications. In: Logics for Databases and Information Systems, Kluwer. 265-306 (1997)
4. Parent, C., Spaccapietra, S., Zimányi, E.: Conceptual Modeling for Traditional and Spatio-temporal Applications: The MADS Approach. Springer-Verlag pp. 466 (2006)
5. Miralles, A.: Ingénierie des Modèles pour les Applications Environnementales. Ph.D. Thesis, Université de Montpellier II (2006)
6. Parent, C.: A Framework for Characterizing Spatio-temporal Data Models. In: Advances in Multimedia and Databases for the New Century, 89-97, World Scientific (2000).
7. Mas, S., Fei, W., Wolfgang, R.: Using Ontologies for Integrity Constraint Definition. ISSDQ05. Beijing, China (2005)
8. Egenhofer, M.J.: Spatial SQL: A Query and Presentation Language, IEEE Transactions on Knowledge and Data Engineering 6 (1): 86-95, (1994)
9. Vallières, S., Brodeur, J., Pilon, D.: Spatial Integrity Constraints: A Tool for Improving the Internal Quality of Spatial Data. In: Devillers, R., Jeansoulin, R., eds, Fundamentals of Spatial Data Quality, Great Britain, ISTE Ltd., pp. 161-177 (2006)
10. Ubeda, T., Egenhofer, M.J.: Topological Error Correcting in GIS, SSD '97, LNCS, Vol. 1262, Springer-Verlag, 283-297 (1997)

11. Pizano, A., Klinger, A., Cardenas, A.: Specification of Spatial Integrity Constraints in Pictorial Databases, Computer, Vol. 22 (12), 59-71, (1989)
12. Normand, P.: Modélisation des contraintes d'intégrité spatiale, théorie et exemples d'applications, M.Sc. Thesis, Laval University, Quebec, Canada (1999)
13. Kang, M., Pinet, F., Schneider, M., Chanet, J., Vigier, F.: How to Design Geographic Databases? Specific UML Profile and Spatial OCL Applied to Wireless Ad Hoc Networks, In: Proceedings of 7th AGILE Conference on Geographic Information Science (2004)
14. Bédard, Y., Larrivée, S., Proulx, M.J., Nadeau, M.: Modeling Geospatial Databases with Plug-Ins for Visual Languages: A Pragmatic Approach and the Impacts of 16 Years of Research and Experimentations on Perceptory, CoMoGIS04, LNCS 3289, pp. 17-30 (2004)
15. Hadzilacos, T., Tryfona, N.: A Model for Expressing Topological Integrity Constraints in Geographic Databases, LNCS, Vol. 639, 252-268, Italy (1992)
16. Slonneger, K., Barry, L.K.: Formal Syntax and Semantics of Programming Languages: A Laboratory Based Approach, Addison-Wesley, pp. 637, (1995)
17. Schwitter, R.: Controlled Natural Language as Interface Language to the Semantic Web, 2nd Indian International Conference on Artificial Intelligence, pp.1699-1718 (2005)
18. Clementini, E., Di Felice, P.: A Comparison Method for Representing Topological Relationships, Information Systems, Vol. 80, pp. 1-31 (1994)
19. Chang S.: Visual languages: A Tutorial and Survey, Software, Vol. 4(1), pp. 29-39 (1987)
20. Boursier, P., Mainguenaud, M.: Langages de Requêtes Spatiales : SQL Étendu vs. Langage Visuel vs. Hypermédias, SIGAS, Vol. 2 (1), pp.37-51, (1992)
21. Egenhofer, M., Herring, J.: Pint-set topological spatial relations, IJGIS,Vol. 5(2),pp.133-152
22. Bédard, Y., Larrivee, S.: Spatial Database Modeling With Pictrogrammic Languages. In: Encyclopedia of GIS., S. Shekhar & Hui Xiong (Editors), Springer, NY, in press, 14 p.
23. Egenhofer, M., Herring, J.: Characterizing Binary Topological Relations between Regions, Lines, and Points in Geographic Databases. NCGIA Technical Report 94-1 (1994)
24. UML 2.0 OCL Specification, pp. 170 (2003)
25. Lindland, O.I., Guttorm, S., Solvberg, A.: Understanding Quality in Conceptual Modeling, IEEE Software, Vol. 11 (2), pp. 42-49 (1994)
26. Teeuw, W.B., Van den Berg, H.: On the Quality of Conceptual Models, In: Behavioral Models and Design Transformations: Issues in Conceptual Modeling, ER 97 (1997)
27. Salehi, M., Bédard, Y., Mostafavi, M.A., Brodeur, J.: Towards Integrity Constraints of Spatial Datacubes, ISSDQ07, the Netherlands (2007)