This document is published in:

Molina, José M., et al. (eds.) (2011) *User-Centric Technologies and Applications: Proceedings of the CONTEXTS 2011.* (Advances in Intelligent and Soft Computing, 94). Springer, pp. 69-77.
DOI: http://www.dx.doi.org/10.1007/978-3-642-19908-0_80_8

Automatically Updating a Dynamic Region Connection Calculus for Topological Reasoning

Miguel A. Serrano, Miguel A. Patricio, Jesús García, and José M. Molina

Abstract. During the last years ontology-based applications have been thought without taking in account their limitations in terms of upgradeability. In parallel, new capabilities such as topological sorting of instances with spatial characteristics have been developed. Both facts may lead to a collapse in the operational capacity of this kind of applications. This paper presents an ontology-centric architecture to solve the topological relationships between spatial objects automatically. The capability for automatic assertion is given by an object model based on geometries. The object model seeks to prioritize the optimization using a dynamic data structure of spatial data. The ultimate goal of this architecture is the automatic storage of the spatial relationships without a noticeable loss of efficiency.

Keywords: RCC Automatic Assertion, Dynamic Topological Relationships, Ontology-based Application, Ontology-centric Application.

1 Introduction

Knowledge approaches have always used ontologies to conceptualize and organize interpretations of the real world. Since its conception ontologies were designed for reasoning with preset information. This information also had the condition that must not undergo intensive updates. In the last years, researchers have been developing ideas which assumes that ontology applications are prepared to automatically populate a knowledge base or to bring about changes in a large amount of data at the same time. Nowadays none of these functions can be done maintaining an acceptable performance from a given number of updates.

Lately, interest and necessity to carry out the semantics of the representation between domain objects with spatial characteristics is taking a greater presence in knowledge applications. Spatial representations are limited by the description

Miguel A. Serrano · Miguel A. Patricio · Jesús García · José M. Molina
GIAA Computer Science Departament, Universidad Carlos III de Madrid, Avda. de la Universidad Carlos III, 22, 28270 Colmenarejo, Madrid, Spain
e-mail: {miguel.serrano, jesus.garcia, miguelangel.patricio, josemanuel.molina}@uc3m.es

logic standard used in ontologies because the definition of the spatial logic is very wide and complex and also is a not very efficient alternative in terms of calculation time. To overcome this limitation spatial representations are achieved through the Region Connection Calculus (RCC) logic formalism. RCC is a theory [1] for qualitative spatial representation and reasoning. This theory provides a formalism which allows inferring implicit knowledge which is hidden but is deductible from explicit knowledge declared in concrete statements and a set of qualitative spatial relationships to describe the relative location of spatial objects to one another.

The loss of performance when updates occur intensively and the increased use of RCC could result in a collapse in ontology applications. This paper presents an architectural model for solutions which combines dynamic spatial relations with the need of automated RCC assertions and updates. As was previously introduce this kind of approach needs a compromise between the automatic storage of dynamic RCC data with the minimal loss of performance in terms of time.

To obtain automatic functionality the spatial individuals represented as geometric entities have to be instantiated in a Euclidean planar linear geometry model. This model must discover, maintain and provide the relationships between these geometries. All these tasks are carried out through an analysis performed after each update of the geometries in the model. Spatial relationships are only updated when the geospatial situation of the objects causes changes in the general object organization.

To keep the spatial relationships updated is necessary to check all the instantiated objects pairwise. To accomplish this, it would have to perform a comparison of quadratic complexity at every moment. This task may cast down a good overall performance. For example in a video tracking system could be necessary check, at every frame, the number of tracked and static geometries relationships. To resolve this kind of situations we will show an approach which uses an auxiliary data structure capable to reduce the difficulty of the search to logarithmic complexity.

Spatial relationships from the geometric model are finally stored to a RCC module as qualitative spatial relationships. Knowledge is shared between the knowledge base and the RCC. RCC module contains the topology relationships of the spatial individuals who belong to the knowledge base. Access to knowledge is guaranteed since knowledge base individuals can be easily identified in the RCC module.

The paper is organized as follows. In Section 2 approaches related with the RCC are studied briefly; Section 3 the dynamic RCC automatically update overall architecture is presented; Section 4 shows an application example to solve a typical tracking problem applying our solution; Section 5 explains the conclusions obtained and the future work.

2 Spatial, Temporal and Topological Approaches

The problem of the spatial relationships between objects is very common in the spatial databases field. Unfortunately, there are no designed architectures for the automatic and optimized assertion of spatial information in the knowledge bases field.

Spatial databases follow a similar approach; receive and store spatial data in a tree trying to optimize the objects searches. Spatial databases have the disadvantage that they cannot do reasoning with objects; however knowledge approaches are thought to do it.

In recent years the joint use of the spatial and temporal dimensions combined with ontology reasoning is growing. Nowadays there are ontology-based proposals which include spatial reasoning with dynamic geometries [2]. There are also trends which combines knowledge system architecture with a representation of the RCC family tree in the Ontology Web Language (OWL) [3]. Even there are architectures with reasoning systems for visual information fusion [7].

The most remarkable standards in the topology field are developed by the Open Geospatial Consortium previously known as the Open GIS Consortium. These are the result of an agreement process to develop publicity available interface standards. OpenGIS standards support interoperable solution ranging from web wireless to the location-based services and mainstream IT [12].

RCC theory can be seen as a topology standard in the knowledge bases field. RCC spatial representation uses topological relations which are compliant with the OpenGIS standard [4]. There are publications dealing with the mapping of existing relationships in the OpenGIS standard to the RCC theory [1]. The RCC is an axiomatization of certain spatial concepts and relations in first order logic. The basic theory assumes just one primitive dyadic relation: C(*x*, *y*) read as "*x* connects with *y*". Individuals (*x*, *y*) can be interpreted as denoting spatial regions. The relation C(*x*, *y*) is reflexive and symmetric.

Of the defined relations, Disconnected (DC), Externally Connected (EC), Partially Overlaps (PO), Equal (EQ), Tangential Proper Part (TPP), Non Tangential Proper Part (NTPP), Tangential Proper Part Inverse (TPPi) and Non Tangential Proper Part Inverse (NTPPi) have been proven to form a jointly exhaustive and pairwise disjoint set, which is known as RCC-8. Similar sets of one, two, three and five relations are known as RCC-1, RCC-2, RCC-3 and RCC-5.

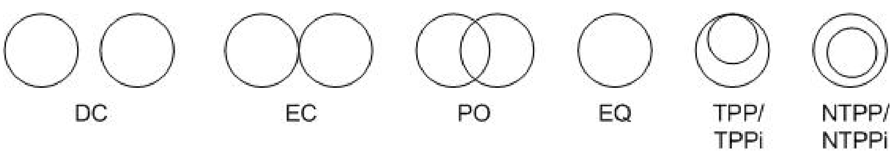

Fig. 1 RCC-8 relations.

Currently there are implementations of this theory for two of the most powerful systems of reasoning based on ontologies, Pellet [5] and RACER.

3 Overall Architecture

Dynamic RCC system presented in this paper is based on three modules; a container with individuals with geometric features which provides the entry data to the main module, a synergistic module composed of an object model and an auxiliary data structure, and a Region Connection Calculus where the qualitative spatial relationships between the objects are stored. The overall architecture of the proposed framework is illustrated in Fig. 1.

Since this architecture is thought for ontology-based systems, geometric individuals are initially stored in a knowledge base. This knowledge base contains static objects whose position does not change and dynamic objects whose position is altered over time. All the geometric representations of these objects are used as input to the main module and more specifically to the sub-module that contains the object model. The geometric representations are instantiated in the object model in two cases; if they do not exist previously in the model or if they were already instantiated but its position or size has been modified regarding the last update. As a result of these changes it is necessary to perform a full topological analysis of the relations between the new instantiated geometries with the existing geometries. This type of analysis has a quadratic complexity, since it is necessary to compare each geometry with the others.

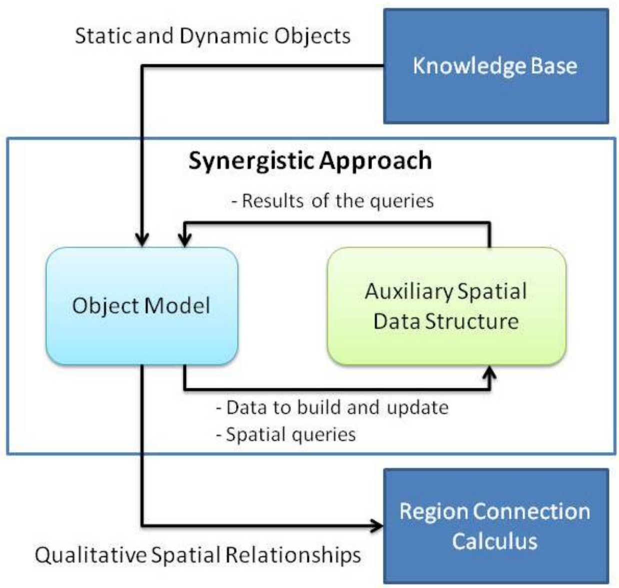

Fig. 2 Overall Architecture.

This difficulty can be translated into a decrease of the system efficiency. To avoid this problem, the number of checks between geometries has to be reduced. To achieve this is necessary to reduce the number of geometries selecting only those are candidates to modify the spatial relations of each geometry. By applying an auxiliary data structure, these candidates, who make up a subgroup inside of the geometries total group, can be determined.

Once the auxiliary data structure returns the candidates, the object model can be fully updated. The qualitative spatial relationships which have changed from the previous state are then updated on the RCC system. The RCC characteristics of the objects are independent of the knowledge base therefore it is not necessary to make changes in the instances of the ontologies.

3.1 Object Model

Synergistic approach is composed of two sub-modules that work in a complementary way; an object model and an auxiliary data structure. The object model is a system that represents spatial objects in a Euclidean plane and is capable of obtaining quickly 2D spatial relationships between objects. Besides implementing the geometric model its operations could define the OpenGIS Simple Features standard [13]. This standard specifies digital storage of geographical data with spatial and non spatial attributes and defines a set of spatial operators.

Sometimes, building spatial data structures is necessary to know the objects speed and the smallest enclosing rectangle for their geometric representations. Object speed data can come from the knowledge base or from the object model. Although the speed may not be considered a spatial data, the OpenGIS standard implemented by the object model ensures the possibility of include non-spatial attributes in its model. To find the smallest rectangle enclosing just need to find the minimum area which limit points of the geometry.

OpenGIS spatial representations and RCC are not compatible. The output from the object model must be mapped from the standard features of the OpenGIS to the RCC-8 base relations. This kind of translation has been previously proposed in [8].

As mentioned earlier the object model automatically instantiate all the new or updated geometries corresponding to the spatial representations of objects from the knowledge base. To discover the spatial relationships between geometries should be carried out a topological analysis of these new instances. This analysis leads to a quadratic order check because it is necessary to check the new items with everyone else in every moment of time. The exactly number of total checks is $N*(N-1)/2$.

In a scenario in which mobile objects move in a consistent manner, spatial relationships change between objects that are close in consecutive time instants. For this reason, it is necessary to do the topological analysis to geometries that are physically close to the geometry whose relationships are being analyzed. A structured topological hierarchy that can change over time may therefore improve the performance of this approach. Starting from basic information on the situation of the geometries it is possible to build a spatial data structure that maintains a hierarchical topological sort on the Euclidean space and supports spatial queries. Performing spatial queries on this structure is possible to reduce the amount of geometries involved in the object model analysis. To do so each query has to return for each geometry which candidate geometries can change their spatial relationships. The results of the queries to the spatial data structure are the candidate geometries who were next to the geometry whose relations were evaluated in time when the query was performed.

3.2 Auxiliary Spatial Data Structure

This spatial data structure is not predefined by the architecture, however it is recommended that follow several conditions; must be capable of defining a spatial hierarchy throughout the time, should be able to handle the overlap between objects and its operations must have a complexity lower than the quadratic.

It is strongly recommend that search, deletion, insertion and update operations do not carry a high overhead because the spatial data structure will have to be updated every time an object is created or its position is changed. Unfortunately spatial data structure approaches without quadratic complexity in their operations do not make an optimal sort of the space.

The spatial data structure is created and updated while the object model is being updated by the knowledge base. Ideally, the smallest rectangle which enclose each geometry are included in different areas in which the plane is divided. Assuming that consistency between objects exists, the spatial relationships can only change from one to another instant only with the object belonging to the same areas or adjacent areas. With this type of management is very simple and quick to refer the candidate geometries which reduce the number of test in the object model.

Many structures can be used depending on the conditions of each application. Applications with few dynamic objects that are not changing constantly its area may use tree structures. R-Tree [11] or R* [10] approaches could be very proper. In the same way there are also previous and well known structures, like quad trees and *k*-dimensional trees, these structures are very useful for static object applications. *k*d-tree approaches can be combined with dynamic objects, however if the

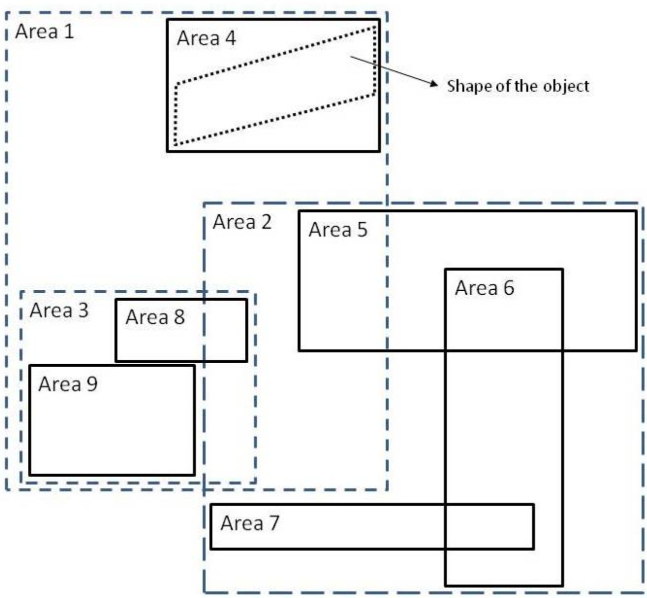

Fig. 3 Example of structured spatial information.

number of dynamic objects grows too this would force to rebuild the tree at every moment and this would not be efficient. If a big quantity of dynamic and static objects is needed, it would be interesting to have another type of structure. [9] is an example of approach with logarithmic complexity search and non-quadratic complexity insertion operations with suboptimal solutions.

4 Case Study: Video Tracking

This type of architecture has multiple areas of application. A direct application area is tracking systems. This type of systems have many needs that this architecture can cover, such as, moving objects over the time, tasks which include spatial and semantic interpretation, good prospects for processing time performance, etc.

Imagine that exist a tracking system which stores all its information on a knowledge base to carry out semantic and spatial tasks for error corrections. These semantic tasks can treat, for example, the problem of loss of consistency in the size of a track when it crosses with another track. The aim of this error correction is to keep the size of the tracks constant during the overlap.

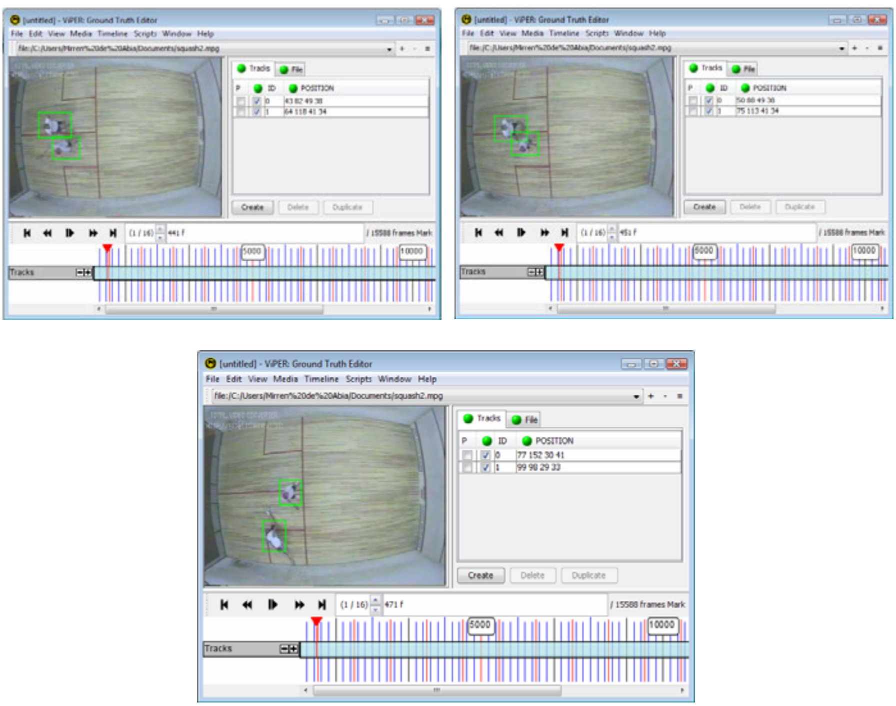

Fig. 4 Error correction in a tracking system. 4A shows the beginning of the error treatment. 4B shows no changes in the size of the tracks during the time of overlap. 4C shows the end of the error treatment when the tracks can change its size.

Table 1 Relationship between events and actions during an example of execution of the system.

Fig	Event	Actions per frame	Topology Relationships
-	Track1 and Track2 are in an approximation state.	1) Track1 and Track2 positions are updated at the spatial data structure. 2) Object model does not detect changes between the geometric relationships.	DC(Track1, Track2)
3A	Track1 and Track2 intersect.	1) Track 1 and Track 2 positions are updated at the spatial data structure. 2) Object model automatically detect the intersection between Track1 and Track2. 3) RCC topology relations change from Disconnected to Externally Connected.	EC(Track1, Track2)
3B	Track1 and Track2 overlap.	1) Track 1 and Track 2 positions are updated at the spatial data structure. 2) Object model automatically detect the overlap between Track1 and Track2. 3) RCC topology relations change from Externally Connected to Partially Overlaps.	PO(Track1, Track2)
3C	Track1 and Track2 are in a withdrawal state.	1) Track 1 and Track 2 positions are updated at the spatial data structure. 2) Object model automatically detect the absence of relationship between Track1 and Track2. 3) RCC topology relations change from Partially Overlaps to Disconnected.	DC(Track1, Track2)

To make this control is necessary to complete the task in several phases; one must first know the positioning of the tracks at time. According to what has been specified, that information must be provided by the knowledge base. Secondly the system has to know when tracks enter in an overlap state. This task is carried out jointly by the two modules belonging to the synergistic approach. The object model automatically discovers this new spatial relationship between the tracks and stores it in as a RCC relationship. Finally the system should keep the size of both fixed during the overlap duration. This condition must be sent from the knowledge base to the tracking system as a recommendation. In the same way, when the tracks are no longer overlapping, the object model has to modify the RCC relationship between them and the submission of recommendations has to be stopped. Figure 3 shows a sequence of three pictures illustrating an example case of automatic error correction with a tracking tool. Table 1 shows the event-action response of the system to the sequence in Figure 4.

5 Conclusion and Future Work

We have presented a novel architecture to address a dual problem; the automatic assertion of dynamic spatial objects and the overhead that these updates may cause on a knowledge base. The architecture has been designed to be embedded in any ontology-based application but some recommendations about possible standards to implement it in a real system have been done.

Future works will include the use of the temporal properties of RCC taking advantage of space operations as operations between intervals. Thus a time interval can be asserted to be included in another one, consecutive, partially coincident, etc. [6] This architecture will be implemented in a semantic activity recognition system similar to that have been proposed in the example application.

Acknowledgments. This work was supported in part by Projects CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02-02/TEC, CAM CONTEXTS (S2009/ TIC-1485) and DPS2008-07029-C02-02.

References

[1] Randell, D.A., Cui, Z., Cohn, A.G.: A Spatial Logic based on Regions and Connection. In: Proceedings 3rd International Conference on Knowledge Representation and Reasoning (1992)

[2] Grenon, P., Smith, B.: SNAP and SPAN: Prolegomenon to Geodynamic Ontology. IFOMIS Technical Report, University of Leipzig

[3] Grütter, R., Bauer-Messmer, B.: Combining OWL with RCC for Spatioterminological Reasoning on Environmental. In: Proceedings of 3rd OWL: Experiences and Directions, OWLED 2007 (2007)

[4] Grütter, R., Scharrenbach, T., Bauer-Messmer, B.: Improving an RCC-Derived Geospatial Approximation by OWL Axioms. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, pp. 293–306. Springer, Heidelberg (2008)

[5] Stocker, M., Sirin, E.P.: A Hybrid RCC-8 and RDF/OWL Reasoning and Query Engine. In: Proceedings of OWL: Experiences and Directions 2009 (OWLED 2009). Sixth International Workshop, vol. 529 (2009)

[6] Gómez-Romero, J., Patricio, M.A., Garcia, J., Molina, J.M.: Towards the Implementation of an Ontology-Based Reasoning System for Visual Information Fusion. Springer, Heidelberg (2009)

[7] Gómez-Romero, J., Patricio, M.A., Garcia, J., Molina, J.M.: Ontology-based context representation and reasoning for object tracking, Colmenarejo (2009)

[8] Schuele, M., Karaenke, P.: Qualitative spatial reasoning with topological information in BDI agents (2010)

[9] Hadjieleftheriou, M., Kollios, G., Tsotras, V.J., Gunopulos, D.: Efficient Indexing of Spatiotemporal Objects. In: Jensen, C.S., Jeffery, K., Pokorný, J., Šaltenis, S., Hwang, J., Böhm, K., Jarke, M. (eds.) EDBT 2002. LNCS, vol. 2287, p. 251. Springer, Heidelberg (2002)

[10] Beckmann, N., Kriegel, H.-P., Schneider, R., Seeger, B.: The R*-tree: An Efficient and Robust Access Method for Points and Rectangles. In: Proceedings ACM SIGMOD International Conference on Management of Data, Atlantic City, pp. 322–331 (1990)

[11] Guttman, A.: R-trees a dynamic mdex structure for spatial searching. In: Proceedings ACM SIGMOD International COnference on Management of Data (1984)

[12] Open Geospatial Consortium, `http://www.opengeospatial.org/standards`

[13] OpenGIS Simple Features Specification for SQL, `http://www.opengeospatial.org/standards/sfs`