CHOROS 2: Improving the Performance of Qualitative Spatial Reasoning in OWL

Nikolaos Mainas, Euripides G.M. Petrakis
School of Electronic and Computer Engineering
Technical University of Crete (TUC)
Chania, Crete, Greece, GR-73100
Email: {nmainas, petrakis}@intelligence.tuc.gr

***Abstract*—We investigate on potential improvements to reasoning methods for topological and directional spatial information in OWL. Building upon path consistency, the new reasoner design, referred to as CHOROS 2, suggests several optimizations for reducing the number of compositions of basic relations and for speeding-up its run-time performance. CHOROS 2 serves also as a framework for a realistic evaluation of several alternative reasoner designs discussed in this paper. Perhaps, one of the most interesting alternatives that deserves further investigation relies on the idea of decomposing directional relations into two smaller sets of basic relations yielding fewer compositions. CHOROS 2 infers all implied relations and detects inconsistencies while retaining soundness, completeness and tractability over the supported relations sets. Experimental results demonstrate that all variants of CHOROS 2 run up to several times faster than both CHOROS 1 (its previous implementation) and SOWL, a spatial reasoner implemented in SWRL which runs under Protégé.**

Keywords*—*Spatial Ontology; Qualitative Spatial Reasoning; Performance;

I. INTRODUCTION

Formal spatial representations have been studied extensively in the Database, Knowledge Representation, Geographic Information Systems (GIS) and, recently the Semantic Web literature [1]–[3]. Spatial entities (e.g., objects, regions) in classic database systems are represented using points, lines (polygonal lines) or Minimum Bounding Rectangles (MBRs) enclosing objects or regions and their relationships. Relations between spatial entities can be topological, orientation or distance-based relations. Furthermore, spatial relations can be partitioned into qualitative (i.e., relations described using lexical terms such as "Into", "South" etc.) and quantitative (i.e., relations described using numerical values such as "10 Km away", "45 degrees North" etc.). The motivation for using a qualitative approach is that it is considered to be closer to the way humans represent spatial knowledge. Another motivation is that it is possible to deal with incomplete knowledge. Nevertheless, it is not always possible to directly encode the semantics of spatial relations in OWL and DL (the description logic underlying OWL). There might be inconsistencies within a set of spatial relations that will not be detected by an OWL reasoner or, an OWL reasoner might not compute all spatial inferences.

Topological relations between regions represent the relative position of regions in the plane. The most widespread formalism for representing such relations is the so called Region Connection Calculus (RCC) formalism [4]. The most commonly used form of this calculus is referred to as RCC-8 calculus and specifies the 8 mutually exclusive relations between region pairs (DC, EC, EQ, NTPP, NTPPi, TTP, TPPi, PO) which are shown in Fig. 1.

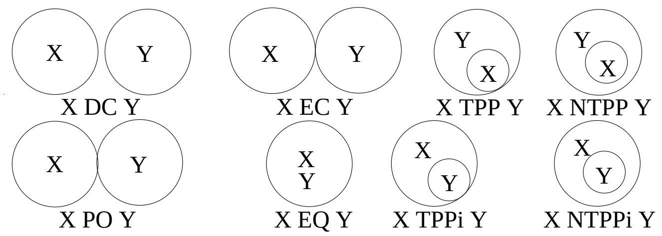


Fig. 1. The set of RCC-8 topological relations.

Cone-Shaped Directional (CSD) relations are defined based on cone-shaped areas [1], [5]. As shown in Fig. 2, eight directional relations (nine with the addition of the *identicalTo* relation for objects sharing the same position) can be identified namely, North (N), North East (NE), East (E), South East (SE), South (S), South West (SW), West (W) and North West (NW). Each relation covers a part of the 360 degrees range and all relations taken together cover the entire 360 degrees range.

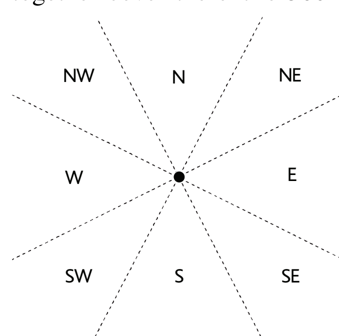


Fig. 2. The set of cone-shaped directional relations.

Choosing either representation is a design decision that depends mainly on the application. However, both RCC and CSD expressions in OWL may co-exist within the same ontology together with standard OWL semantic relations.

Spatial reasoning is a general term that refers to a variety of techniques that allow inference of new facts from a set of given facts as well as checking for their consistency. The most popular reasoning methods are constraint-based techniques [6], [7]. Reasoning applies on sets of qualitative spatial relations which are jointly exhaustive and pairwise disjoint (i.e., between any pairs of spatial entities exactly one of the basic relations holds). Typically, this mechanism is realized by means of path consistency algorithm [8] which applies iteratively over existing and inferred facts until no new facts can be inferred or an inconsistency is detected. The run time performance of reasoning is constrained by the run-time complexity of path consistency (typically polynomial in the number of individuals of the input data set). Path consistency when applied on a set of assertions containing only basic relations retains tractability and guarantees soundness and completeness of reasoning.

SparQ [9] and GQR [10] provide optimized solutions to the problem of reasoning for general qualitative spatial calculi and information expressed in text or XML. Recently, Semantic Web has motivated attempts to provide practical solutions to the problem of reasoning for spatial information expressed in OWL DL. Taking advantage of OWL semantics, spatial reasoning in separated from standard OWL DL reasoning which is handled by Pellet [1]. The emphasis is on reducing the run-time complexity of reasoning while retaining soundness and completeness. Along these lines, PelletSpatial [11] is a solution for topological RCC-8 representations based on a variant of the path consistency algorithm by Renz and Nebel [8].

Building upon PelletSpatial, CHOROS 1 [12] supports reasoning over directional CSD-9 or RCC-8 relations or a mixture of RCC-8 and CSD-9 relations (rather than merely RCC-8 relations as PelletSpatial does). Optimizations of CHOROS 1 include a multi-threading (faster) implementation enabling the parallel execution of CSD and RCC reasoning. A limitation of CHOROS 1 (and also of PelletSpatial) is that the ontology is not updated with reasoning results (i.e., the inferred relations are not added to the ontology). Finally, the SOWL framework [13] supports representation of spatio-temporal information in OWL and includes an implementation of both reasoners in SWRL[2] that runs with Pellet.

The new reasoner design, referred to as CHOROS 2, inherits all features of CHOROS 1 and implements several optimizations to the representation and reasoning mechanism for improving its run-time performance. First, reasoning is speeded-up by reducing the number of 8 basic RCC relations and of the 9 basic CSD relations to 7 and 8 respectively, by replacing in the ontology every instance of RCC *EQuals* relation and every instance of the point identity relation of CSD-9 by the OWL *SameAs* axiom. This reduces the number of possible disjuncitons of composed basic relations by an order of magnitude and results in faster computation of path consistency. Then, path-consistency is further optimized by computing these disjunctions on the fly (i.e., at run time) rather than using large composition tables for all possible disjuncitons of the basic relations sets.

In recent work by Batsakis [14], reasoning over CSD-9 directional relations relies on a decomposition of basic relations into two smaller relation sets. Reasoning uses two smaller composition tables (rather than one bigger one) yielding fewer compositions of relations and finally, resulting in faster response times. In this work, we show that this method may fail to compute all inferred relations in certain cases and we introduce a new decomposition schema that fixes the problem.

Finally, CHOROS 2 updates the ontology with all logical inferences of spatial reasoning, a feature that is not supported by all previous implementations of the spatial reasoner such as PelletSpatial or CHOROS 1. To show proof of concept, the performance of CHOROS 2 is assessed both theoretically and experimentally and compared with that of CHOROS 1 and SOWL.

II. SOWL

SOWL [13] is an ontology for representing and reasoning over spatio-temporal information in OWL. Building-upon well-established standards of the semantic Web (OWL 2.0, SWRL) SOWL enables representation of static as well as of dynamic (temporal) information. Both RCC-8 topological and CSD-9 relational calculi are supported. Handling both, qualitative temporal and spatial information (i.e., information of temporal or spatial extents are unknown, such as "left-of" for spatial and "before" for temporal relations) in addition to quantitative information (i.e., where temporal and spatial information is defined precisely) is a distinctive feature of SOWL. In SOWL, path consistency is implemented using SWRL rules defining compositions and intersections of supported relations. Reasoners that support DL-safe rules such as Pellet can be used for inference and consistency checking over spatio-temporal relations.

SOWL reasoner is capable of inferring new relations and checking for their consistency, while retaining soundness, completeness and tractability over the supported sets of relations. Notice that, using the full set of relations (e.g., $2^8 - 1$ relations in the case of RCC-8 model) leads to intractability since this set cannot be decided by path consistency. Tractable subsets of the full set are known to exist [6], [15]. In SOWL, for the RCC-8 and CSD-9 relations sets, the minimal tractable sets containing the basic relations for the two models consist of 49 and 33 relations respectively [13]. In contrast to PelletSpatial and CHOROS 1, reasoning is part of the ontology (rather than a separate system), so that maintenance of the ontology requires that changes are applied only to the ontology and not to the system.

A. Decomposition of Directional Relations

An interesting approach for improving the run-time performance of reasoning suggests reducing the number of basic relations and of their composition sets. Batsakis [14] proposed that reasoning over Cone-Shaped Directional (CSD) relations be based on a decomposition of basic relations into two (smaller) relation sets, one for the $East - West$ axes (horizontal) and one for the $North - South$ axes (vertical) illustrated in Fig. 3. For each pair of objects, two new relations, one for each axes are derived. For example, if object A is $North - East$ of object B, the basic relations on $North - South$ axes are $North, South, Equal - Horizontal, Identical - Horizontal$. The relations on $East - West$ axes are $East, West, Equal - Vertical, Identical - Vertical$.

[1] http://clarkparsia.com/pellet/

[2] http://www.w3.org/Submission/SWRL

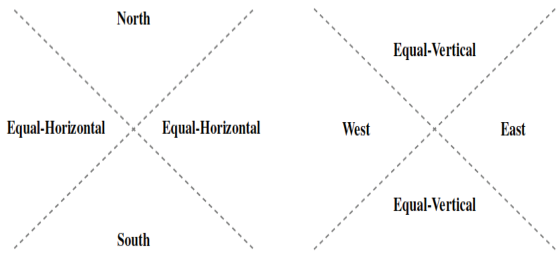


Fig. 3. $North - South$ (left) and $East - West$ (right) relations

The decomposition of CSD-9 relations is defined as follows:

$$\begin{aligned}
N &\equiv North \wedge Equal - Vertical \\
NE &\equiv North \wedge East \\
E &\equiv Equal - Horizontal \wedge East) \\
SE &\equiv South \wedge East \\
S &\equiv South \wedge Equal - Vertical \\
SW &\equiv South \wedge West \\
W &\equiv Equal - Horizontal \wedge West \\
NW &\equiv North \wedge West \\
Identical &\equiv Identical - Horizontal \wedge Identical - Vertical
\end{aligned}$$

Path consistency is realized by introducing two sets of SWRL rules operating on the decomposed relation sets (relations on each axes set are jointly exclusive and pairwise disjoint). Finally, CSD-9 relations are recomposed taking as input the inferred relations from both axes sets (by applying inverses of the above decomposition rules). Table I represents the compositions of relations of $North - South$ set. Relations $North$, $South$, $Equal - Horizontal$ and $Identical - Horizontal$, are denoted by *N*, *S*, *EqH*, *IdH* respectively.

	N	**S**	**EqH**	**IdH**
N	N	N, S, EqH, IdH	N, EqH	N
S	N, S, EqH, IdH	S	S, EqH	S
EqH	N, EqH	S, EqH	N, S, EqH, IdH	EqH
IdH	N	S	EqH	IdH

TABLE I: Composition table for $North - South$ directional relations

Table II represents the compositions of relations of $East - West$ set. Relations *East*, *West*, $Equal - Vertical$ and $Identical - Vertical$, are denoted by *E*, *W*, *EqV*, *IdV* respectively.

	E	**W**	**EqV**	**IdV**
E	E	E, W, EqV, IdV	E, EqV	E
W	E, W, EqV, IdV	W	W, EqV	W
EqV	E, EqV	W, EqV	E, W, EqV, IdV	EqV
IdV	E	W	EqV	IdV

TABLE II: Composition table for $East - West$ directional relations

The following example shows that this method may fail to yield exactly the same relations as the reasoner applying on the original CSD-9 relations set. Consider the following relations between 4 objects: "Object1" *SE* "Object2", "Object2" *SW* "Object3", "Object3" *E* "Object4". Then, reasoning according to CSD-9 model, infers the following relations: "Object1" (*SE, SW, S*) "Object3", "Object2" (*S, SE, SW, E*) "Object4", "Object1" (*S, SE, SW, E*) "Object4".

Reasoning according to [14] takes as input the following $North - South$ relations: "Object1" *S* "Object2", "Object2" *S* "Object3", "Object3" *EqH* "Object4" and also, the following $East - West$ relations: "Object1" *E* "Object2", "Object2" *W* "Object3", "Object3" *E* "Object4", and infers the following relations: "Object1" *S* "Object3", "Object2" (*S, EqH*) "Object4", "Object1" (*S, EqH*) "Object4". Taking these as input, the re-constructed CSD-9 relations are: "Object1" (*SE, SW, S*) "Object3", "Object2" (*S, SE, SW*) "Object4", "Object1" (*S, SE, SW*) "Object4". Notice that, relation *E* between pairs "Object2", "Object4"and "Object1" and "Object4" cannot be re-constructed. For details the reader is referred to [16].

III. CHOROS 2

CHOROS 2 inherits all advantages of its previous implementation [12] while incorporating several improvements and new features. Similarly to CHOROS 1, it works with RCC-8 and CSD-9 relations, as well as with RDF/OWL relations which are handled by Pellet. CHOROS 2.0 updates the ontology with inferred facts so that, these can be re-used or queried using SPARQL[3]. This feature is not supported by CHOROS 1 or PelletSpatial.

CHOROS 2.0 architecture (Fig. 4) consists of several modules, the most important of them being the *Parser* which loads and processes the ontologies into memory, the *Constraint Network* for storing spatial property assertions, the *Reasoner* which runs path consistency and, the *Re-constructor* which updates the ontology with new spatial inferences.

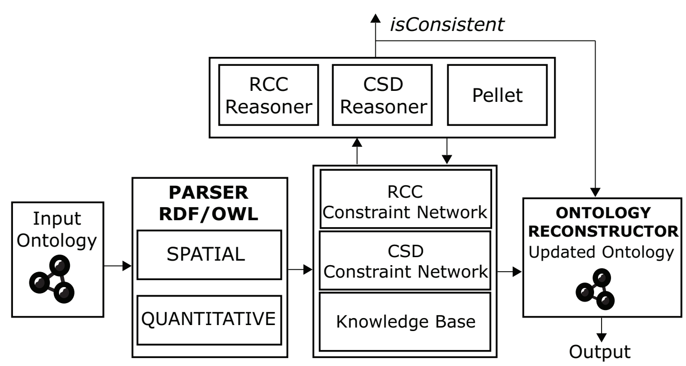


Fig. 4. CHOROS 2 architecture.

The *Parser* handles ontologies and spatial queries (e.g., in SPARQL). It implements a) *Spatial Parser* that separates spatial relations from non-spatial OWL ones using vocabularies with name definitions of CSD-8 and RCC-9 relations. Spatial relations are removed from the RDF graph and are stored in their corresponding RCC or CSD constraint network. The rest of the graph, contains only non-spatial OWL assertions which are stored in Pellet's KB and are handled by Pellet. *Quantitative Parser* which handles quantitative spatial information and

[3]http://www.w3.org/TR/rdf-sparql-query/

computes all RCC-8 and CSD-9 relations from objects whose spatial location is expressed using numerical values (e.g., X, Y coordinates). It is implemented using Jena [4]. All directional and topological relations are computed and are asserted into their corresponding constraint network. This is also a unique feature of CHOROS 2.

A *Constraint Network* (CN) is a set of variables together with a set of constraints. CHOROS 2 implements two such networks, one for RCC-8 and one for CSD-9 relations. Non-spatial OWL assertions are stored in Pellet's KB whose structure is similar to the constraint network. Pellet's KB consists of an assertions box (that contains assertions about individuals) and a terminological box (that contains axioms about classes). This KB is used for consistency checking and inference on non-spatial information.

CHOROS 2 handles separately spatial from semantic DL reasoning. Because compositions of basic relations for the two spatial models are mutually exclusive, CHOROS 2.0 separates reasoning for each model: The relations of each model are hold in separate CNs and reasoning is executed separately using separate threads.

The *Re-constructor* updates the ontology with information inferred by the reasoner. If no inconsistency is detected in any CN, the re-constructor updates the output ontology with new inferred spatial relations. Spatial relations from both CSD-9 and RCC-8 CNs are asserted into the ontology, as OWL object property assertions.

A. CHOROS 2 Reasoning and Optimizations

Spatial reasoning for each spatial calculus is achieved by applying path consistency [11] separately for each calculus. Path consistency, computes all inferred relations using compositions of existing relations until a fixed point is reached or until an inconsistency is detected (i.e., $\emptyset$ is produced as a result). The possible compositions of basic RCC-8 and CSD-9 relations are stored in composition tables defined by Cohn et.al. [17] and by Renz and Mitra [5] respectively.

Path consistency is implemented by applying the rule:

$$\forall x, y, k\; R_s(x, y) \leftarrow R_i(x, y) \cap (R_j(x, k) \circ R_k(k, y))$$

representing intersection of compositions of relations with existing relations (symbol $\cap$ denotes intersection, symbol $\circ$ denotes composition and R_i, R_j, R_k, R_s denote spatial relations). The formula is applied until a fixed point is reached (i.e., the consecutive application of the rules above doesn't yield new inferences) or until the empty set is reached, implying that the ontology is inconsistent.

1) CHOROS 2 optimizations:

a) Composing disjunctions of relations: The composition of basic relations may infer disjunctions of such relations because disjunctive entries exist in the composition table (i.e., not all compositions yield a unique relation as a result). For example, the composition of relations $NTPP$ and EC returns relation DC, while compositions of relations EC and PO returns five relations namely, (DC, EC, PO, TPP, $NTPP$). Disjunctions of relations are represented using new relations, whose compositions must also be defined and asserted into the KB.

The composition of disjunctions of relations can be computed in two ways: (a) By pre-computing the composition of disjunctions and storing the result in a table and (b) By computing the composition of disjunctions "on the fly" (i.e., at run-time). CHOROS 1 adopted the first approach. For topological relations the full composition table has $2^8 \times 2^8$ entries (i.e., up to 2^8 disjunctions can appear). Similarly, for directional relations, the composition table has $2^9 \times 2^9$ entries. This method takes memory to store the tables and more time to compute. Following the second approach in practice results in much less combinations of disjunctions of basic relations and each one involves a simple look-up operation in the 8×8 RCC-8 composition table or the 9×9 CSD-9 table. This not only saves memory but also (as will be shown in the experiments) computes faster.

For example, the composition between the disjunction of relations (DC, EC, $TPPi$) and relation $TPPi$ is computed as follows:

$$(DC \cup EC \cup TPPi) \circ TPPi$$
$$\rightarrow (DC \circ TPPi) \cup (EC \circ TPPi) \cup (TPPi \circ TPPi)$$
$$\rightarrow DC \cup (DC \cup EC) \cup (TPPi \cup NTPPi)$$
$$\rightarrow DC \cup EC \cup TPPi \cup NTPPi$$

b) Reducing the number of basic relations: The number of possible compositions over all disjunctions is big and effects the performance reasoning even for small data sets. To deal this problem CHOROS 2 reduces the number of basic CSD relations from 9 to 8 (the size of possible compositions is then $2^8 \times 2^8$), by replacing the directional relation *identicalTo* with OWL axiom *sameAs*. Similarly, the number of basic RCC basic relations is reduced from 8 to 7 by replacing the topological relation *EQ* by OWL axiom *sameAs*. These two relations are asserted into the Pellet's KB and are treated as OWL object properties.

c) Decomposing CSD Relations: CHOROS 2 also implements reasoning based on the decomposition of basic CSD relations as in [14]. However, as shown in Sec. II-A, this approach may fail to compute all inferred relations (as reasoning on the original CSD-9 relations does). To fix this problem, a new decomposition framework (and new decomposition and recomposition rule sets) is proposed [16]. The axes of Fig. 3 are replaced by those of Fig. 5. The following rules apply for the decomposition of CSD-9 relations:

$$\begin{aligned}
N &\equiv North \wedge Vertical - North\\
NE &\equiv North \wedge East\\
E &\equiv Horizontal - East \wedge East\\
SE &\equiv South \wedge East\\
S &\equiv South \wedge Vertical - South\\
SW &\equiv South \wedge West\\
W &\equiv Horizontal - West \wedge West\\
NW &\equiv North \wedge West\\
Identical &\equiv Identical - Horizontal \wedge Identical - Vertical
\end{aligned}$$

[4] http://jena.apache.org/

This is implemented in the *Parser* component. Reasoning is now realized by means a $Equal - Vertical$ and and $Equal - Horizontal$ reasoner operating on their corresponding CNs.

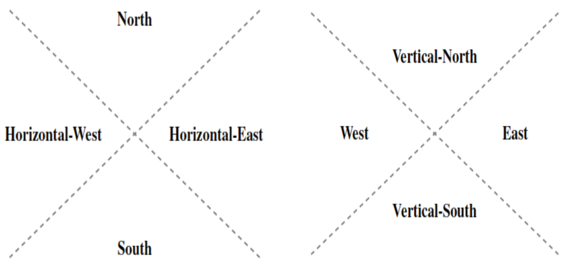


Fig. 5. Proposed $North - South$ (left) and $East - West$ (right) relations.

Table III illustrates the composition table of $East - West$ relations set. Relations $Vertical - North$, $Vertical - South$ are denoted by *VerN*, *VerS* respectively. Table IV is the composition table of the new $North - South$ relations set. Relations $Horizontal - East$, $Horizontal - West$ are denoted by ***HorE***, ***HorW*** respectively.

	E	**W**	**VerN**	**VerS**	**IdV**
E	E	E, W, VerN, VerS, IdV	E, VerN	E, VerS	E
W	E, W, VerN, VerS, IdV	W	W, VerN	W, VerS	W
VerN	E, VerN	W, VerN	VerN	E, W, VerN, VerS, IdV	VerN
VerS	E, VerS	W, VerS	E, W, VerN, VerS, IdV	VerS	EqV
IdV	E	W	VerN	VerS	IdV

TABLE III: Composition table for the new $East - West$ directional relations

	N	**S**	**HorE**	**HorW**	**IdH**
N	N	N, S, HorE, HorW, IdH	N, HorE	N, HorW	N
S	N, S, HorE, HorW, IdH	S	S, HorE	S, HorW	S
HorE	N, HorE	S, HorE	HorE	N, S, HorE, HorW, IdH	HorE
HorW	N, HorW	S, HorW	N, S, HorE, HorW, IdH	HorW	HorW
IdH	N	S	HorE	HorW	IdH

TABLE IV: Composition table for the new $North - South$ directional relations

The following re-composition rules are the inverse of the decomposition rules described above, enriched with rules handling relations between spatial entities of either the $East - West$ or the $North - South$ CN. For details the reader is referred to [16].

For each relation $relNS(x, y)$ in *North-South* network the following rules are applied:

If a relation $relEW(x, y)$ exists in $East - West$ Network then:

- if (*North* $\in relNS(x, y)$ AND $Vertical - North \in relEW(x, y)$)add in ontology $North_{CSD9}(x, y)$;
- if (*South* $\in relNS(x, y)$ AND $Vertical - South \in relEW(x, y)$) add in ontology $South_{CSD9}(x, y)$;
- if ($Horizontal - East \in relNS(x, y)$ AND *East* $\in relEW(x, y)$) add in ontology $East_{CSD9}(x, y)$;
- if ($mathitHorizontal - West \in relNS(x, y)$ AND *West* $\in relEW(x, y)$) add in ontology $West_{CSD9}(x, y)$;
- if (*North* $\in relNS(x, y)$ AND *East* $\in relEW(x, y)$)) add in ontology $NorthEast_{CSD9}(x, y)$;
- if (*North* $\in relNS(x, y)$ AND *West* $\in relEW(x, y)$) add in ontology $NorthWest_{CSD9}(x, y)$;
- if (*South* $\in relNS(x, y)$ AND *East* $\in relEW(x, y)$) add in ontology $SouthEast_{CSD9}(x, y)$;
- if (*South* $\in relNS(x, y)$ AND *West* $\in relEW(x, y)$) add in ontology $SouthWest_{CSD9}(x, y)$;
- if ($Identical - Horizontal \in relNS(x, y)$ AND $Identical - Vertical \in relEW(x, y)$) add in ontology $Identical_{CSD9}(x, y)$;

else:

- if (*North* $\in relNS(x, y)$) add in ontology: $North_{CSD9}(x, y)$, $NorthEast_{CSD9}(x, y)$, $NorthWest_{CSD9}(x, y)$;
- if (*South* $\in relNS(x, y)$) add in ontology: $South_{CSD9}(x, y)$, $SouthEast_{CSD9}(x, y)$, $SouthWest_{CSD9}(x, y)$.
- if ($Horizontal - East \in relNS(x, y)$) add in ontology: $East_{CSD9}(x, y)$.
- if ($Horizontal - West \in relNS(x, y)$) add in ontology: $West_{CSD9}(x, y)$.

For each relation $relEW(x, y)$ in $East - West$ network apply the following rules:

If a relation $relNS(x, y)$ exists in $North - South$ Network then:

- if (*East* $\in relEW(x, y)$ AND $Horizontal - East \in relNS(x, y)$) add in ontology $East_{CSD9}(x, y)$;
- if (*West* $\in relEW(x, y)$ AND $Horizontal - West \in relNS(x, y)$) add in ontology $West_{CSD9}(x, y)$;
- if ($Vertical - North \in relEW(x, y)$ AND *North* $\in relNS(x, y)$) add in ontology $North_{CSD9}(x, y)$;

- if ($Vertical-South \in relEW(x,y)$ AND *South* $\in relNS(x,y)$) add in ontology $South_{CSD9}(x,y)$;
- if (*East* $\in relEW(x,y)$ AND *North* $\in relNS(x,y)$) add in ontology $NorthEast_{CSD9}(x,y)$;
- if (*East* $\in relEW(x,y)$ AND *South* $\in relNS(x,y)$) add in ontology $SouthEast_{CSD9}(x,y)$;
- if (*West* $\in relEW(x,y)$ AND *North* $\in relNS(x,y)$) add in ontology $NorthWest_{CSD9}(x,y)$;
- if (*West* $\in relEW(x,y)$ AND *South* $\in relNS(x,y)$) add in ontology $SouthWest_{CSD9}(x,y)$;
- if ($Identical-Vertical \in relEW(x,y)$ AND *Identical-Horizontal* $\in relNS(x,y)$) add in ontology $Identical_{CSD9}(x,y)$;

else:

- if (*East* $\in relEW(x,y)$) add in ontology: $East_{CSD9}(x,y)$, $NorthEast_{CSD9}(x,y)$, $SouthEast_{CSD9}(x,y)$;
- if (*West* $\in relEW(x,y)$) add in ontology: $West_{CSD9}(x,y)$, $NorthWest_{CSD9}(x,y)$, $SouthWest_{CSD9}(x,y)$;
- if ($Vertical-North \in relEW(x,y)$) add in ontology: $North_{CSD9}(x,y)$;
- if ($Vertical-South \in relEW(x,y)$) add in ontology: $South_{CSD9}(x,y)$;

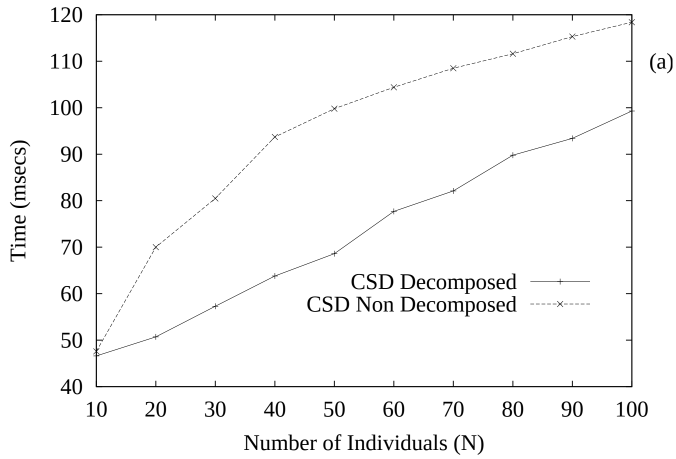
By applying the new model to the example of Sec. II-A, the generated $North-South$ relations are [16]: "Object1" *S* "Object2", "Object2" *S* "Object3", "Object3" *HorE* "Object4". The generated $East-West$ relations are: "Object1" *E* "Object2", "Object2" *W* "Object3", "Object3" *E* "Object4". Reasoning is applied on each set separately and infers the following relations: "Object1" *S* "Object3", "Object2" (*S, HorE*) "Object4", "Object1" (*S, HorE*) "Object4". The *Reconstructor* produces the following output: "Object1" (*SE, SW, S*) "Object3","Object2" (*S, SE, SW, E*) "Object4", "Object1" (*S, SE, SW, E*) "Object4" (i.e., exactly the same output with the original CSD reasoner).

IV. Evaluation

The purpose of the experimental evaluation is to demonstrate the superior performance of CHOROS 2 over a) PelletSpatial [11] and CHOROS 1 [12] and b) SOWL [13] a spatial reasoner implemented in SWRL. We carried-out two different sets of experiments corresponding to measurements of performance in the average and the worst cases. The average case performance is encountered when less than n^2 relations are inferred from an input set of n locations. In our experiments, kn relations ($k=8$ for RCC-8 and $k=9$ for CSD-9) are asserted. This is for example the case of a random set of objects. Accordingly, the worst case performance is encountered when the number of asserted relations are in the order of n^2. This is for example the case of objects given in a certain arrangement (i.e., each one is *North* of another or contained in each other).

The data sets are simple ontologies containing between 10 and 100 random spatial individuals (regions or locations). Every individual can be related with only one individual. All running times reported are averages over 10 ontologies. In all experiments we compare the running time of all reasoner implementations as a function of the number of individuals. All experiments were carried-out on a Windows PC, 2.60GHz, 4 Gb RAM.

In CHOROS 2, similarly to PelletSpatial, path consistency (and hence reasoning over CSD-9 or RCC-8 relation sets) has $O(n^3)$ complexity in the worst case. This is a pessimistic upper bound since, the overall number of iterations of path-consistency algorithm may be lower than $O(n^2)$ because an inconsistency detection may terminate the reasoning process early, or the asserted relations may yield a smaller number of inferences.



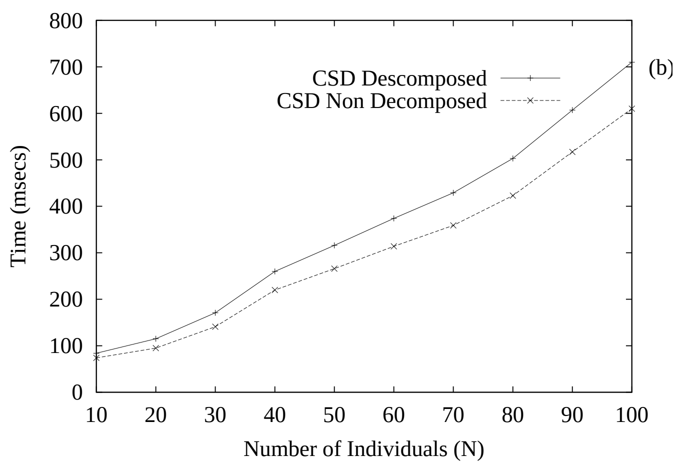


Fig. 6. Average (a) and Worst (b) case performance of reasoning on decomposed CSD relations

Fig. 6 shows that a CSD reasoner running the decomposition scheme of Sec. III-A outperforms a standard CSD reasoner in the average (but not in the worst) case. The new reasoner applies path-consistency on twice as many relations which is fast for small relation sets, such as in the average case (i.e., number of individuals and of relations are in the order of n). In the worst case, decomposition produces many relations (i.e., in the order of n^2) and path consistency runs slower than for the standard reasoner. Nevertheless, because the average case is more characteristic of the typical performance of a reasoner, we have chosen to use the decomposition reasoner in all experiments.

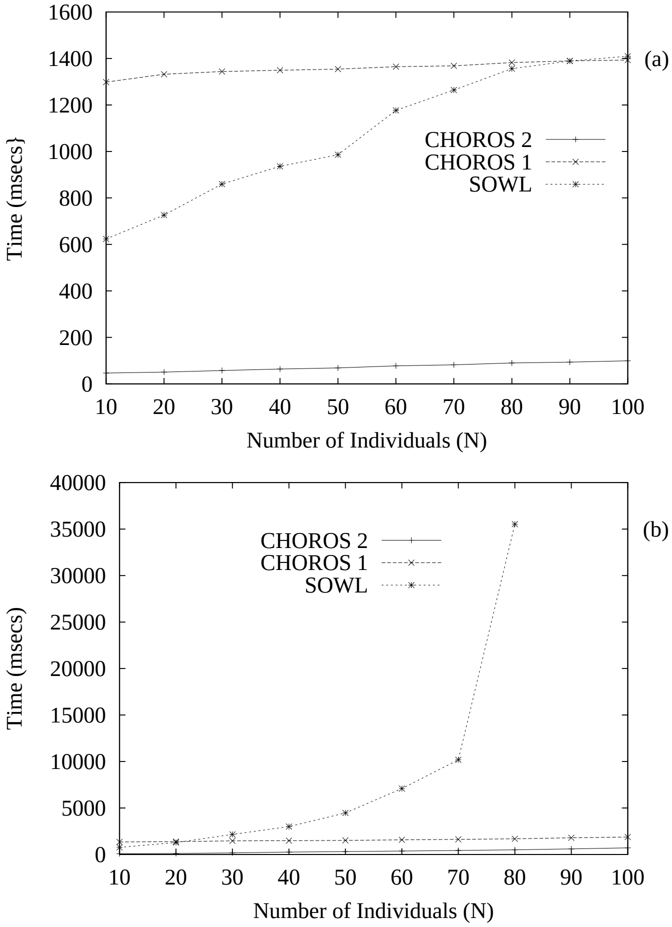


Fig. 7. Average (a) and worst case (b) performance of reasoning over CSD relation sets.

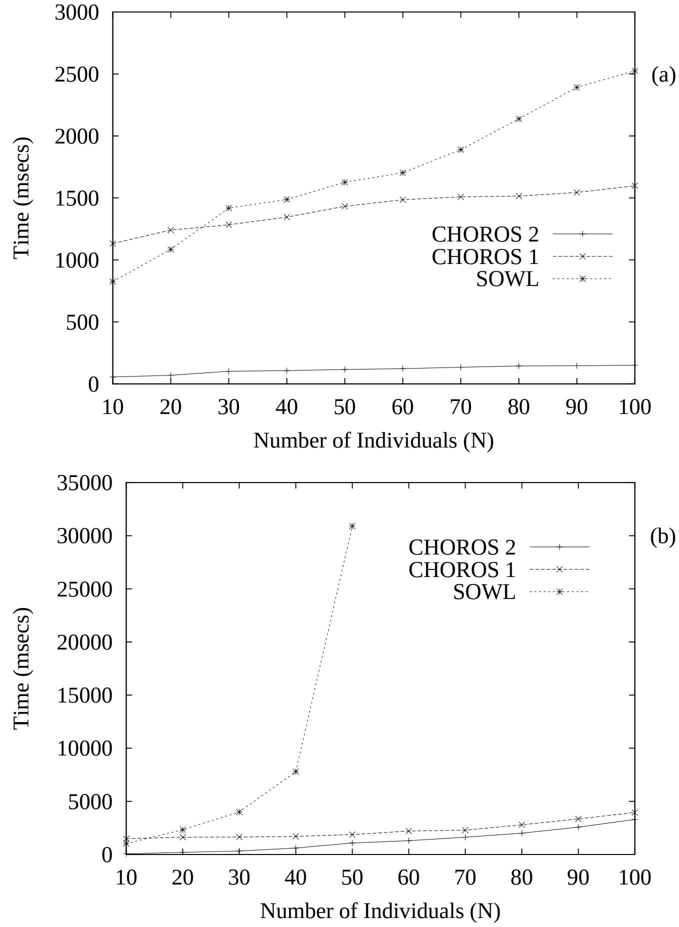


Fig. 8. Average (a) and worst case (b) performance of Reasoning over RCC relation sets.

Fig. 7 illustrates the average (left) and worst case (right) performance of reasoning over CSD relations. Clearly, CHOROS 2 outperforms all its competitors. CHOROS 2 owes its improved performance to the computation of compositions on the fly (i.e., at run time) and to the reduction of spatial relations from 9 to 8 (which leads to smaller composition relation sets).

The run-time performance of SOWL declined drastically for larger data sets in the worst case, as the large number of inferred relations caused memory overflow. Although SOWL may perform better on computers with more memory, CHOROS 2 scales-up much better than SOWL with the size of the input (i.e., the performance gap between the two reasoners increases with the size of the data set) and will run much faster than SOWL for large data sets even on average computers.

Fig.8 illustrates that CHOROS 2 is again the faster implementation in the case of RCC relations. Justification is similar to the previous case. Notice though that reasoning over CSD relations runs significantly faster compared to reasoning over RCC relations (at least in the average case). The justification for this behavior lies in the inherent characteristic of the CSD model, where every relation has an inverse relation.

Composing the inverse relations yields all possible relations. These are skipped by the algorithm (i.e., as they provide no new information) which proceeds with the remaining (fewer) relations. The RCC model does not feature the same characteristic; only relations *NTPP* and *TPP* can be regarded as the inverse relations of *NTPPi* and *TPPi* respectively. However, compositions of these inverse relations don't yield all possible relations as in the CSD model. As a result. more relations are inserted into the CN.

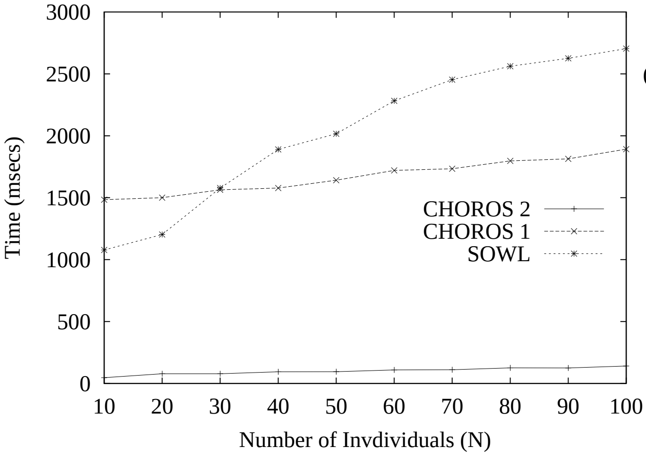
Fig. 9 illustrate the performance comparison of all competing implementations on mixed data sets containing both CSD and RCC relations. CHOROS 2 clearly outperforms any other implementation. Each reasoner is implemented as a "thread" which enables concurrent execution.

A. *TUC Spatial Ontology*

As a case study and to objectively assess the performance of reasoning in the average case we applied CHOROS 2 on the "TUC spatial ontology" [12] with 60 spatial entities of the University campus of Technical University of Crete (TUC). Reasoning times are shown in Table V. Obviously CHOROS 2 outperforms any other reasoner.

V. Conclusions and Future Work

CHOROS 2 is a new spatial reasoner for topological and directional information. It incorporates optimizations and improvements to the spatial representation and path consistency checking mechanism including, a new decomposition model



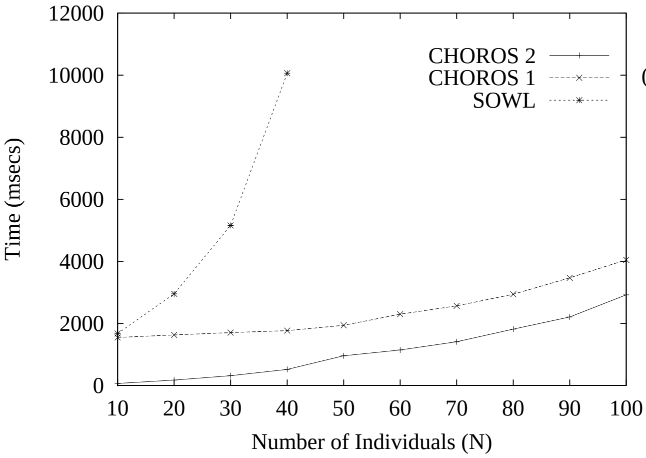


Fig. 9. Average (a) and worst case (b) performance of reasoning on combined RCC and CSD relations sets.

	SOWL SWRL	**CHOROS 1**	**CHOROS 2**
Time (msecs)	8,313	2,312	407

TABLE V: Response time of reasoning techniques on the "TUC spatial ontology".

for CSD-9 directional relations. Evaluation results demonstrate significant performance improvements over existing reasoner implementations such as, CHOROS 1 [12] in both the average and worst cases. Proving completeness and improving the performance of the proposed decomposition scheme, investigating on more effective reasoners utilizing smaller tractable relation sets [13] and examining their performance on real applications are issues for future research.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Unions Seventh Framework Programme (FP7/2007-2013) under grant agreement no 604691 (project FI-STAR).

REFERENCES

[1] A. Frank, "Qualitative Spatial Reasoning About Distances and Directions in Geographic Space," *Visual Languages and Computing*, vol. 3, pp. 343–371, 1992.

[2] R. H. Güting, "An Introduction to Spatial Database Systems." *The VLDB Journal*, vol. 3, no. 4, pp. 357–399, Oct. 1994. [Online]. Available: http://dl.acm.org/citation.cfm?id=615204.615206

[3] I. B. Arpinar, A. P. Sheth, C. Ramakrishnan, E. L. Usery, M. Azami, and M.-P. Kwan, "Geospatial Ontology Development and Semantic Analytics." *T. GIS*, vol. 10, no. 4, pp. 551–575, 2006. [Online]. Available: http://dblp.uni-trier.de/db/journals/tgis/tgis10.html#ArpinarSRUAK06

[4] D. A. Randell, Z. Cui, and A. Cohn, "A Spatial Logic Based on Regions and Connection," in *KR'92. Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference*, B. Nebel, C. Rich, and W. Swartout, Eds. San Mateo, California: Morgan Kaufmann, 1992, pp. 165–176. [Online]. Available: http://citeseer.ist.psu.edu/randell92spatial.html

[5] J. Renz and D. Mitra, "Qualitative Direction Calculi with Arbitrary Granularity." in *PRICAI*, ser. Lecture Notes in Computer Science, C. Zhang, H. W. Guesgen, and W.-K. Yeap, Eds., vol. 3157. Springer, 2004, pp. 65–74. [Online]. Available: http://dblp.uni-trier.de/db/conf/pricai/pricai2004.html#RenzM04

[6] J. Renz and B. Nebel, "Qualitative Spatial Reasoning Using Constraint Calculi." in *Handbook of Spatial Logics*, M. Aiello, I. Pratt-Hartmann, and J. van Benthem, Eds. Springer, 2007, pp. 161–215. [Online]. Available: http://dblp.uni-trier.de/db/reference/spatial/spatial2007.html#RenzN07

[7] B. Nebel and H. J. Bürckert, "Reasoning about Temporal Relations: a Maximal Tractable Subclass of Allen's Interval Algebra," *Journal of the ACM*, vol. 42, no. 1, pp. 43–66, 1995.

[8] J. Renz and B. Nebel, "Efficient Methods for Qualitative Spatial Reasoning," in *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI 1998)*. John Wiley & Sons, 1998.

[9] D. Wolter, "SparQ - A Spatial Reasoning Toolbox," in *AAAI Spring Symposium: Benchmarking of Qualitative Spatial and Temporal Reasoning Systems*. AAAI, 2009, pp. 53–54. [Online]. Available: http://dblp.uni-trier.de/db/conf/aaaiss/aaaiss2009-2.html#Wolter09

[10] M. Westphal, S. Wlfl, and Z. Gantner, "GQR: A Fast Solver for Binary Qualitative Constraint Networks," in *AAAI Spring Symposium: Benchmarking of Qualitative Spatial and Temporal Reasoning Systems*. AAAI, 2009, pp. 51–52. [Online]. Available: http://dblp.uni-trier.de/db/conf/aaaiss/aaaiss2009-2.html#WestphalWG09

[11] M. Stocker and E. Sirin, "PelletSpatial: A Hybrid RCC-8 and RDF/OWL Reasoning and Query Engine." in *6^{th} Intern. Workshop on OWL: Experiences and Directions (OWLED 2009)*. Springer-Verlag New York, Inc., Aug. 2009, pp. 2–31. [Online]. Available: http://www.webont.org/owled/2009/papers/owled2009_submission_20.pdf

[12] G. Christodoulou, E. Petrakis, and S. Batsakis, "Qualitative spatial reasoning using topological and directional information in owl," in *Tools with Artificial Intelligence (ICTAI), 2012 IEEE 24th Intern. Conference on*, vol. 1, Nov 2012, pp. 596–602.

[13] S. Batsakis and E. G. M. Petrakis, "SOWL: A Framework for Handling Spatio-temporal Information in OWL 2.0." in *RuleML Europe*, ser. Lecture Notes in Computer Science, N. Bassiliades, G. Governatori, and A. Paschke, Eds., vol. 6826. Springer, 2011, pp. 242–249. [Online]. Available: http://dblp.uni-trier.de/db/conf/ruleml/ruleml2011e.html#BatsakisP11

[14] S. Batsakis, "Reasoning over 2D and 3D Directional Relations in OWL: A Rule-based Approach," in *Proc. of the 7th Intern. Conf. on Theory, Practice, and Applications of Rules on the Web*, ser. RuleML'13. Berlin, Heidelberg: Springer-Verlag, 2013, pp. 37–51. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-39617-5_8

[15] J. Renz, "Maximal Tractable Fragments of the Region Connection Calculus: A Complete Analysis," *Artificial Intelligence*, vol. 108, pp. 69–123, 1999.

[16] N. Mainas, "CHOROS 2.0: Improving the Performance of Spatial Reasoning in OWL," Nov. 2013. [Online]. Available: http://www.intelligence.tuc.gr/publications.php?pub_author=268&pub_type=All&pub_subject=All

[17] A. G. Cohn, B. Bennett, J. Gooday, and N. M. Gotts, "Qualitative Spatial Representation and Reasoning with the Region Connection Calculus," *GeoInformatica*, vol. 1, no. 3, pp. 275–316, 1997. [Online]. Available: http://dblp.uni-trier.de/db/journals/geoinformatica/geoinformatica1.html#CohnBGG97