A SURVEY OF TEMPORAL-SPATIAL REASONERS IN THE SEMANTIC WEB.

By

EMMANUEL KORTSIDAKIS

BSC COMPUTER SCIENCE ANGLIA POLYTECHNIC UNIVERSITY

A THESIS

Submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

DEPARTMENT OF APPLIED INFORMATICS
AND MULTIMEDIA

SCHOOL OF APPLIED TECHNOLOGY

TECHNOLOGICAL EDUCATIONAL INSTITUTE OF CRETE

2018

Approved by: Major Professor
NIKOLAOS PAPADAKIS

Table of contents

Περίληψη

Η μελέτη των χωρικών–χρονικών reasoned χαρακτηρίζεται στις μέρες μας σαν πρώτη προτεραιότητα. Στο παρελθόν οι σχεσιακές βάσεις δεδομένων χρησιμοποιήθηκαν για να προσεγγίσουν αυτό τον σκοπό. Το κύριο πρόβλημα με την προαναφερόμενη λύση είναι ότι δεν επιτρέπεται η παροχή συμπεράσματος. Οι χωρικοί-χρονικοί reasoners που έχουν χρησιμοποιηθεί τα τελευταία χρόνια είναι οι CHRONOS,CNTRO,ONCOR,PROTON,SBOX,SNAP και SPAN,SOUPA,SOWL,TACHYON,SPIRIT,OWL-S. Αυτό το Thesis προσφέρει την περιγραφή όλων των προ αναφερόντων reasoner καθώς και των σημαντικότερων χαρακτηριστικών τους. Στο τέλος του thesis προσφέρεται συγκριτικός πίνακας όλων των reasoner και επίσης περιέχεται θεωρητική και πειραματική αξιολόγηση των χωρικών και των χρονικών reasoner ξεχωριστά.

ΓΕΝΙΚΟΙ ΟΡΟΙ

Semantic Web, Ontology, Reasoner.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ

Semantic Web Reasoners, Ontology Reasoners, Spatial Reasoners, Temporal Reasoners, Spatial Ontologies, Temporal Ontologies.

ABSTRACT

The study of spatial-temporal reasoners is labeled nowadays as a great necessity. In the past relational databases were used to approach this task. The main problem with the approach mentioned above is that no inference is allowed. Some of the spatial-temporal reasoners used in the last few years are CHRONOS,CNTRO,ONCOR,PROTON,SBOX,SNAP AND SPAN,SOUPA,SOWL,TACHYON,SPIRIT,OWL-S. This thesis describes these reasoners with their important features which include reasoning method, completeness, soundness etc. At the end of this survey a detailed table depicting the spatial-temporal reasoners as far as their attributes are concerned is presented.

GENERAL TERMS

Semantic Web, Ontology, Reasoner.

KEYWORDS

Semantic Web Reasoners, Ontology Reasoners, Spatial Reasoners, Temporal Reasoners, Spatial Ontologies, Temporal Ontologies.

1. INTRODUCTION

An Ontology can be described as the formulation of domain into a format which can be read by a machine. An ontology has the capability to be utilized in order to manage knowledge, retrieve info, store information and share information. A search engine is provided the capability of syntactic and semantic matching by the ontology. In order for web services and content to be presented metadata is used to mark them in order to manage various ontologies. Vital aspects of an ontology are its quality along with its correctness something which holds an important part as far as semantic representation and knowledge sharing are concerned. The quality of ontologies, is directly related with the inconsistency of real world application ontologies. Once an error is present in an ontology that ontology is labeled as inconsistent and correct ontology concept representation is prevented. As a result knowledge representation and semantic understanding suffer. To avoid this Ontology reasoning is used to reduce the information redundancy situated in the base of information and detects the possible clashes in the information contents. [3]An application which deducts rational penalties commencing from specifics, declarations, and truisms is called a reasoner. A reasoner is able to produce by the use of these deductions automated reasoning. Reasoners are distinguished from their attributes which can be methodology, soundness and completeness. This paper presents temporal and spatial reasoners and offers an analysis of their attributes. Temporal Reasoning offers the ability to define the concept of time and at the same time empowers the representation and reasoning of knowledge’s temporal aspects. Temporal Reasoning is obligated to provide a language extension to enable the representation of knowledge’s temporal aspects. In addition, it must have the ability to tell apart dissimilar types of temporal entities. Temporal Reasoning is mostly present in the frame of reference of a more general reasoner. Representations of time have been developed within some problem solving systems, but the goal of a Temporal Reasoner system should be to embody common temporal knowledge and the set of functionalities presented above. [1]

The strong need to integrate reasoning about space immerged due to the fact that increasingly more data became available on the web related to space such as geographic coordinates. Relations became spatial relations between objects and locations in space. The furthermost significant decisions concerning systems of spatial info is the selection of the appropriate reasoning mechanism. There is a large number of ways to approach spatial reasoning which all present a compromise concerning generality and effectiveness. [2]Spatial–temporal reasoning is a region of A.I. which focus on the depiction and reasoning as far as spatial-temporal knowledge is concerned. The remainder of the survey has the subsequent structure. Section 2 depicts the associated research on spatial and temporal reasoners and the main fields of research, section 3 gives a perspective on the importance of temporal, spatial reasoners, section 4 outlines the attributes of certain temporal, spatial reasoners, section 5 compares the reasoners with respect to their attributes and finally section 6 delivers a short conclusion as well as a feature scope.

2. Related work

2.1 Spatial relations between objects and their importance

In a complex environment such as medical images and remote sensing images , extremely valuable information is extracted from the spatial arrangement of objects in order to complete tasks such as recognition and interpretation.[4,5].It has been proposed a great number of times that relations of topology as well as relations of directional position are of great importance.[5,6,7]Spatial reasoning is concerned with the depiction of spatial knowledge and contains the relationships of spatial entities along with the reasoning which exists as Far as these entities and their relationships are concerned. Spatial reasoning has been the topic of great study and exploration in A.I. by using qualitative representations on logical formality. [8]

2.2 Represented time in ontologies

Changes which come as a result of time in ontologies are represented by two approaches versioning and the perduranlist approach. [9, 10]One of the problems of approaches such as OWL-TIME is that instead of portraying the events and their evolution through time they represent temporal concepts and relations.[11] Versioning which is based on repetition of information is troubled by redundancy. Another drawback is that only by comparison of the past and present changes in time can be observed instead of providing a direct representation in the ontology. These shortcomings are solved by the pendurandist approach. [10]

2.3 Qualitative Spatial and Temporal Reasoning related research

A substantial volume of effort took place in the development of frameworks with the goal to make Spatial and Temporal Reasoning more united as well as approachable. The most distinguished work concerning spatial reasoning as well as temporal reasoning was issued for Sparq [12] and also for the Gqr [13] which is a fast reasoner for qualitative calculations. In addition to the work mentioned above, an application has been developed to empower the development of   Qualitative Spatial and Temporal Reasoners [14]

3. Reasoner Characteristics

This section of the survey focuses and describes the most valuable attributes of ontology reasoners and provides a brief definition of each one .The characteristics of ontology reasoners are categorized in to 3 different types [15].The first category is concerned with the basic features of ontology reasoners which are Methodology, Soundness, Completeness, Expressivity, Computational Complexity, Native profile, Incremental Classification, Rule support, Justification and finally A box reasoning. The second category is concerned with Dimensional Practical usability which basically determines whether the OWL API is implemented by the reasoner, and some additional characteristics such as availability as a Protégé plug in something which is a standard occurrence by developers. Finally the license status of a reasoner is detailed. The third category details the Performance indicators which are used to calculate the performance of a reasoner. This survey provides a description of a large number of temporal and spatial ontology reasoners with a detailed description of their reasoning characteristics and their dimensional practical characteristics. In addition information about the reasoners will be provided concerning the implementation language, availability, inference support, support for changes in the past and support for spatial data.

Methodology

Description logic reasoners have their foundations on hyper tableau calculi [16][17],something which provides them with soundness and completeness, the Methodology designates the algorithm or the procedure which the reasoner uses in order to find solutions for description logic reasoning problems.[18]

Soundness and Completeness

The characteristic of Soundness and Completeness provides the ability to evaluate whether the conclusions derived from the reasoning methods used are sound and complete (All possible conclusion have concluded).In the case that speed of reasoning is of utmost importance Soundness and Completeness can be neglected[19] .In order for a reasoner to be utilized in a realistic environment the necessity to always be Sound and Complete doesn't exist, what is of outmost importance is to always be aware if it is sound and complete or not. It has been found that the majority of the methods determining the reasoners which are examined in our survey are sound and complete.

Expressivity and Computational Complexity

In description logic a well-known compromise exists, there is an analogy between the articulacy of a language and computational convolution. (The greater the expressivity, the greater the computational convolution.)

Native profile

In order for a reasoned to be efficient, expressive power is needed. The native profile points out the logical pieces which offer that expressive power.

Incremental Classification

Once an ontology has received a classification and in a later phase by the use of additions and removals is updated, a reasoned creates a new concept hierarchy by combining the information provided by the previous classification with the updated axioms. The interfaces of OWL API enable incremental reasoning exposal by the reasoners. The reasoner has the ability given to it by the API to detect changes in the ontology and to instantly process them. .The API provides the reasoned with the ability to listen for ontology changes and then provides the reasoned with the choice to immediately processes them or to place the in a queue in order for them to be processed later processed later [20].

Rule support

In order for rules to be combined with ontologies, the facilitation of Rule support is required. A certain number of reasoners support the Semantic Web Rule set which supplements rules similar to Horn rules to OWLs set of axioms.

Platforms

The operating system a reasoned is designed to operate is the concern of the Platform reasoned attribute

Justifications

Once an inconsistency appears in an ontology this specific attribute provides an explanation. In the case of non-satisfied concept OWL API engulfs a method which returns all explanations.

ABOX reasoning

The individual reasoning which encompasses instance checking is called ABOX reasoning. Also ABOX reasoning encompasses query answering along with consistency checking. The basis for query answering is instance checking which tests whether it's included in a knowledge base that an individual is a concepts instance. Query answering is heavily based on instance checking because by repetition of instance checking query answering can be achieved. [21]

OWL API

An Application Programming Interface (API) which is designed to work with OWL ontologies is called Owl API. [20]The OWL API provides support for parsing and writing by using the syntax provided by the standards of OWL 2.Additionally an interface is provided for OWL reasoners in order for an application to be able to engulf different reasoners without the need of implementation changes.

OWL LINK API

The OWL LINK [21] is a protocol which was built with the intention to interact with OWL 2 reasoners without relying on a specific implementation. Client application are empowered to manage reasoners by the assistance of OWL Link. It also provides assistance for accessing reasoning services by using a group of queries.

Protégé Plugin (Protégé support)

Protégé is an ontological editing environment which fits in OWL 2 language specification, is open source and has as its base the OWL API. Reasoner developers are commonly releasing a plugin for Protégé.

Neon support

Neon support is an attribute which provides an indication on whether a reasoners is able to be used with the Neon toolkit.

License

Concerning reasoner licensing the concept of a dual license exists, which means that the developers offer the reasoned either free or with a fee according to where it’s going to be used .Licenses are distinguished as open source or not.

Jenna support

Whether the reasoned can be used with the Jena API is the concern of the Jena support attribute.

Implementation language

The language used for the reasoner implementation is specified by this attribute.

Availability

Whether a reasoned is free and open is indicated by this attribute.

Inference support

Inference proceeds by forward chaining and backward chaining which are the approach of ontology reasoners.[22].In forward chaining the reasoned begins from the existing data and derives valid inferences. In backward chaining the reasoned starts from a specific query in order to confirm it and find all possible outcomes.

Support for changes in the past

Whether the temporal reasoner provides the user with the ability to change a past occurrence.

Support for spatial data

When we refer to spatial data we mean the data which represents information concerning the shape of certain geometric objects and their physical location. By objects we mean a point location, a road, or river. Two spatial data types are the Geometry data type which is used for representing data in a flat coordinate system and the Geography data type which is used for representing data in a global coordinate system.

4. Temporal Reasoners, Spatial Reasoners and Spatial Temporal Reasoners

In the following chapter an analysis of specific Temporal reasoners , Spatial reasoners and spatial temporal reasoners is provided based on the reasoner attributes as they were defined in the previous chapter and additionally a description of the reasoners architecture is provided. The reasoners are categorized in to three distinct categories temporal reasoners, spatial reasoners and spatial-temporal reasoners.

4.1 Chronos

Chronos is an individual Temporal OWL reasoner. Chronos mission is to reason on temporal knowledge situated in ontologies by using OWL. Performance wise Chronos has been found to be more efficient than SOWL. [23]

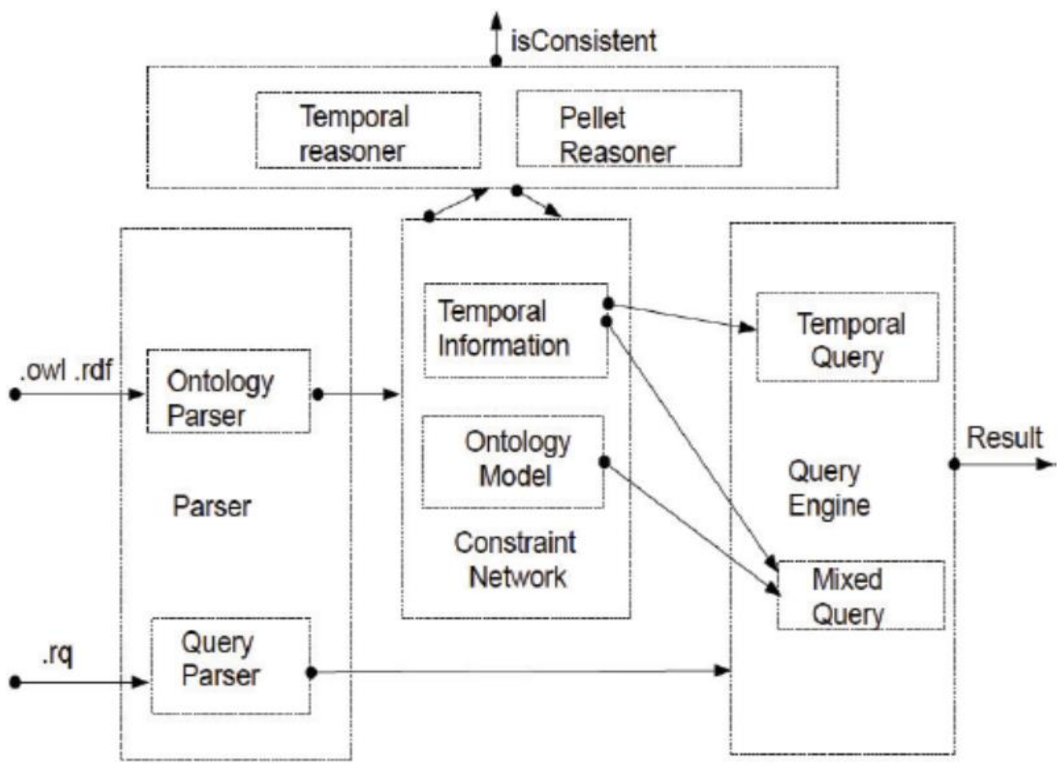

Fig.1 Chronos architecture

Chronos architecture is comprised by a number of modules as is depicted in Fig 1.The main architectural components of Chronos are:

4.1.1 Parser

Chronos forms a Constraint network by parsing the ontology which results to the detection and extraction of the temporal relations among intervals. The implementation of RDF2 and ARQ3 parser is necessary.

The CHRONOS constriction network consists of time-based triplicates in the graph of an ontology and by non-temporal Owl statements cached in the Pellet reasoner Knowledge Base. The user holds the ability to perform tasks such as relation search, determination of the number of temporal and non-temporal relations and in addition determines the amount of consistency in a temporal network by using temporal reasoners composition table. Finally the non-temporal information structure can be examined for consistency with the involvement of Pellet. [46][47]

4.1.2 Reasoner

Chronos uses a reasoned of exclusive use for temporal calculus and discriminates semantic Description Logic reasoning from temporal reasoning. [35][36][37]Temporal reasoning is a type of a constriction gratification problem which is non-deterministic polynomial-time hard. Detectable cases of similar nature problems are managed by Chronos by the limitation of declared temporal relations towards the fundamental Allen relations. The application of the path consistency algorithm leads finally to temporal reasoning. [38][45]

4.1.3 Query Engine

The query functionality of Chronos is similar to the query functionality of Pellet Spatial and Choros with the only difference that in Chronos's case the operators are temporal. So basically it addresses connective queries by the specification of patterns as temporal and non-temporal. The query engine is able to process queries composed in SPARQL if and only if that not any variable is deployed inside the declared location, every assets located inside the declared location an entity or data type and at minimum one of the three has to engulf a temporal object characteristic in the declared spot. [39]The query engine has the ability to process queries of temporal nature which specify temporal assets inside the RDF graph or time-based intermissions which are associated via Allen relations. [41][42]The answering process of a query comprises of two stages. [40][43][44]During the first stage the group of query results produced a time-based query are promoted to the secondary phase where its goal is to assure the satisfaction of a queries non-temporal portion.

4.2 Pellet Spatial

Pellet Spatial is a spatial reasoner which is applied on top of Pellet. The check for Consistency and the answering of queries on spatial data is provided by Pellet Spatial for data which are represented by the Region Connection Calculus. Pellet Spatial provides support for both RCC-8 relations and also provides support for the standard RDF/OWL relations. Mixed SPARQL queries can be addressed over the pre mentioned relation types.[24]Two Region Connection Calculus reasoners are implemented by Pellet Spatial, one based on specific semantics which preserve the conversion of RCC relation to OWL-DL category truisms and one grounded on the configuration table of RCC which executes an algorithm set on path consistency.

4.2.1 Reasoning Engine Architecture

In order for Pellet Spatial to achieve the support as far reasoning operations and the querying operations for spatial RCC-8 relationships in addition to the standard RDF relations are concerned every region is symbolized so that is depicted as an OWL distinct region. An OWL article assets declaration represents a spatial relation between two regions .Relations of non-spatial nature are represented as normal OWL assertions. Pellet Spatial pushes the reasoning capabilities of Pellet so that the semantics of RCC relationships for reasoning are considered.

4.2.2 OWL-DL RCC-8 Reasoning Engine

Description logics and Modal logics are closely related with each other and additionally they is a close relation among Modal logics and RCC-8 formalism. [48] There is the possibility for the translation of RCC-8 relations in to Description logic axioms. The first step is to assign an OWL class CR to every RCC region R. The reader is brought up for the translation particulars. Additionally in order for the condition of region to be regulated the definition of an axiom takes place. The role of this axiom definition is to impact non determinism and the number of eligible ontology quantifiers. [49]The pre mentioned fact greatly affects the performance of the system in such a great way where as a result the system is impractical only in certain sections. [50]Taking under consideration the fact of the semantic translation preservation spatial relations consistency checking is minimized to simply consistency checking. [51]

4.2.3 Hybrid RCC-8 Reasoning Engine

The prementioned implementation faced certain problems as it is defined in the preceding segment, this lead to the quest for a different reasoning approach as far as RCC-8 constraint networks are concerned. Pellet Spatial’s hybrid implementation firmly divides semantic OWL-description logic reasoning from spatial reasoning by the utilization of a dedicated RCC reasoner. The management of spatial relations is accomplished as an RCC constraint network which delivers operability to manage querying and consistency. Relations of non-spatial nature are controlled as a PELLET KB.

The performance of consistency checking in the hybrid application is delivered by a path consistency algorithm which is originated from the configuration table aimed for RCC-8.Path consistency is provided by Pellet Spatial by the way of the 8 base RCC-8 relationships reasoning.

4.2.4 Answering Spatial Queries

Pellet Spatial offers support to a number of basic graph patterns enquiries [52] which are comprised of both spatial and semantic RDF relationships patterns. This kind of enquiring is sub stained for reasoning of both spatial and non-spatial architectures. Spatial querying is inherently supported by Pellet Spatial as it provides processing for spatial and semantic Owl relationships. Hence its applicable make inquiries involving regions in a spatial relation which are attributed semantic RDF relations characteristics. The Pellet Spatial algorithm for responding to queries of spatial nature relies on the fundamental structure where RCC-8 relations are transformed to OWL-DL truisms. As a result enquiries of spatial nature are possible to be transformed to Sparql description logic enquiries which are then replied by Pellet. As far as the query answering algorithm with which we are involved is concerned there is a fact that needs to be mentioned. It’s worth mentioning that a number of the RCC-8 relationships which are converted in to axioms are converted in to axioms with general part limitations. Although Pellet Spatial declines supporting Sparql-description logic query atoms which contain a inconstant filler, there is the possibility when spatial queries are involved that a district which is interpreted in to a role constraint filler to be variable.

4.3 Oncor

Oncor handles the task of building flexible and realistic low load ontologies of areas, instruments and sensors a system that is context aware. Oncor is made from an array of resolver functions which are meant to be used for resolving modified and contextualized information. [25]There are six approaches to reasoning using ontological resolvers.[27] The Closest Common Subsumer (CCS) is a method driven by the objective to determine the most matching location which is consistent with a group of indications. The Granularity Harmonizer (GH) is basically an algorithm which is driven by the mutual contradiction of granularity change met in networks with multiple sensors. The Democratic is the hypothesis that precise data is inclined to be highly reoccurring than noisy data. Point originated from the notion that newer data of a certain location provide more reliability than older ones. In this method the latest piece of information produced by a sensor is chosen. [61]This algorithm is utilized by many area location systems because of its plainness and its computational effectiveness. [62]Time Decay is utilized for the accommodation of imprecise sensor depictions where in order to work a certain level of fuzziness is needed. [63][64] The exponential decay theory is the basis of this algorithm in order for each indication source to be compared in comparison to their newness. A large number of data sources are combined to service the most reliable data source for every location value. [65][66]Bias: a large number of systems utilize an ad-hoc algorithm designed specifically for the sensor framework. It takes under consideration that computer users spent the majority of their time in their offices [67] what the algorithm does is it selects the user's office under the condition it's located in the evidence or else it back to the latest proof origin. It's worth mentioning that the Granularity Harmonizer in addition to the Democratic algorithms are filters and there is no assurance that total conflict resolution will be accomplished, nonetheless certain conflicts are sorted out.[68]All algorithms work in diverse contexts as far as accuracy and precision are concerned. Once precision at a tolerable range is achieved Closest Common Subsumer is a good fit for achieving accuracy. [69]The Granularity Harmonizer is a good choice filter for the sensors which include coinciding areas in granularity alteration. In the case of having false positive sensor evidence the most appropriate choice would be the Democratic algorithm.

4.4 CHOROS

Choros is qualitative spatial reasoner which is executed in java. It offers dependability checking along with query answering as far as spatial data which are depicted by the RCC and CSD is concerned.[26]Choros supports RCC-8 and CSD-9 relationships as well as RDF semantic relations. By taking under consideration the for mentioned details Choros is able to deliver results for diverse SPARQL queries above spatial and non-spatial types. [71]

4.4.1 Choros Reasoning Architecture

Choros represents spatial knowledge by using an OWL ontology. Every region is depicted as an OWL individual, every spatial relation is defined as an OWL entity assets and an OWL entity assets declaration represents a spatial relation between two regions.Choros achieves the separation of spatial reasoning originating form the semantic OWL description logic reasoning by using a unique spatial reasoned component (CT reasoner)

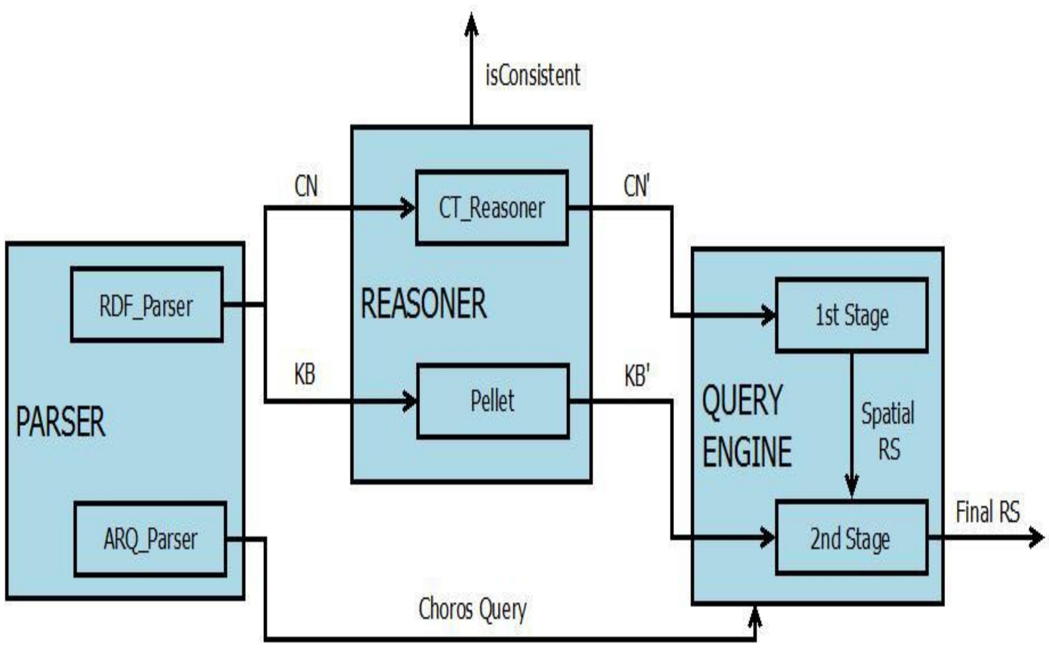

Fig. 2 The architecture of the CHOROS reasoner.

The Ontologies are overloaded into the Parser after authentication. This phase makes sure that all the properties obligate a useable tripartite procedure. Throughout the Parsing taking place by RDF, the three-dimensional property declarations are spaces into a constriction network and declarations of non-spatial nature are cached in an information base. [72]The main part of the arrangement consists of the Reasoner module. The Configuration Table Reasoner is involved in the uniformity concerning a constriction network, and Pellet monitors how consistent the information base is. After the validation follows the loading of the queries and a query structure is formed to classify query atoms to spatial and non-spatial during the loading phase. A double phase query answering procedure is applied by the query engine with the first stage being the return of a set of spatial query results. Secondly the second stage receives the pre mentioned set as input, and applies additional limiting so that the non-spatial query is content. Consequently, the concluding set of query outcomes is formed.

4.4.2 Reasoner

In the center of Choros the resoner component is situated. There is a firm separation in Choros among spatial reasoning and OWL-DL reasoning because of the fact that it one reasoned is used exclusively for every calculus. RCC relationships are handled as being an RCC limited net and respectively CSD relationships are handled similarly by the use of a limited net. In the case of relations of non-spatial nature, these are handled by Pellet as KB. In the first two cases relationships are being articulated on the basis of asset of basic relationships which is shut below a large number of operations. Consequently there is the option to use constraint based methodology in order to reason over the mentioned relations. Hence it’s vital for a composition table to be provided for the basic relation or for all of them accompanied with the procedures needed to compute the configurations of complicated relations. [73]

A definition of a composition table is reached by the use of the official relation semantics. If not then it’s impossible to provide verification for correctness and completeness regarding the interpretations. [74] Another important role of the prementioned relation semantics is that they are vital for finding proficient reasoning algorithms which are needed by the majority of applications. [75] To be able to display whether reasoning is definable over a relation system is impossible without the use of official semantics. [76]

4.4.3 Path-Consistency Algorithm

The path consistency process which is used for inspecting the dependability of a network has the identical implementation as the hybrid approach as Pellet Spatial described earlier in this survey. A constraint network N has for example a group of well-defined RCC-8 relations. There are two ways that the constraint network is consistent, the first is that its empty and the second is when all the relations inside the network are consistent.

4.5 SOWL

SOWL is an ontology which handles spatial and temporal ontological information. In SOWL the 4D fluent representation is extended in a way which it's able to manage quantitative and qualitative spatial and temporal info. [28]Temporal and spatial Allen operators are supported by SOWL and in addition to that it deducts spatial and temporal relations from existing ones by incorporating reasoning rules. Another important aspect of SOWL is query optimization by the way of indexing support on spatial and temporal info, and by extending the reasoning to manage amenable groups of spatial and temporal relations by the utilization of path consistency. [77]

4.5.1 REASONING IN SOWL

In order for SOWL to achieve reasoning it utilizes SWRL rules on spatial relations and Allen rule for stating conditional temporal relations.

Spatial Reasoning

Extra relationships of spatial nature are able to be concluded from preexisting relationships by the use of composition tables which define the spatial relations which are possible among two spatial entities. Spatial reasoning is reached by the application of certain rules which implement the deducted relationships of a composition table. [78]Reasoning rules for RCC-8 relationships and directional relationships by the use of SWRL are implemented by SOWL's spatial representation.

Temporal Reasoning

The conformations of the fundamental Allen relations are the backbone of the temporal reasoning in SOWL. [79]

4.5.2 QUERYING SOWL

SOWL is a newer and improved version of TOQL which is used for querying temporal info in OWL and also is able to easily manage spatial and spatial-temporal information. SOWL reacts to classes and assets ontologies as if they were database tables. SOWL gets to improve by using spatial operators besides the temporal ones which handle spatial and temporal relations, so RCC-8 as well as directional relations are sustained by matching operators. As it was applied in TOQL a similar to SQL dialect and syntax is used by SOWL. (SELECT-FROM-WHERE) and additionally supports operators and concepts like AND, OR, UNION, ANY. In regards to properties of static nature queries are issued as normal queries which are being used on the part of the ontology which remains unchangeable in time.

4.6 PROTON

Proton is system concerned with reasoning OWL temporal ontologies. One of the characteristics that distinguish Proton is the ability to provide answers for queries concerning events which alter in time. [29]

Proton transforms in to triplets a temporal ontology in OWL by using SWI-Prolog. After that the next phase is for the triplets to be transformed into to clauses in Prolog. The implementation of the reasoner is achieved by the use of temporal situation calculus and successively by the utilization of Prolog native mechanisms. [83]

Representing a temporal ontology in to a set of comparable predicates is the idea upon which Proton is founded. The Prolog database receives temporal information queries. Proton has the ability to influence relations among instances and temporal intervals in addition to computing inferences. [84]

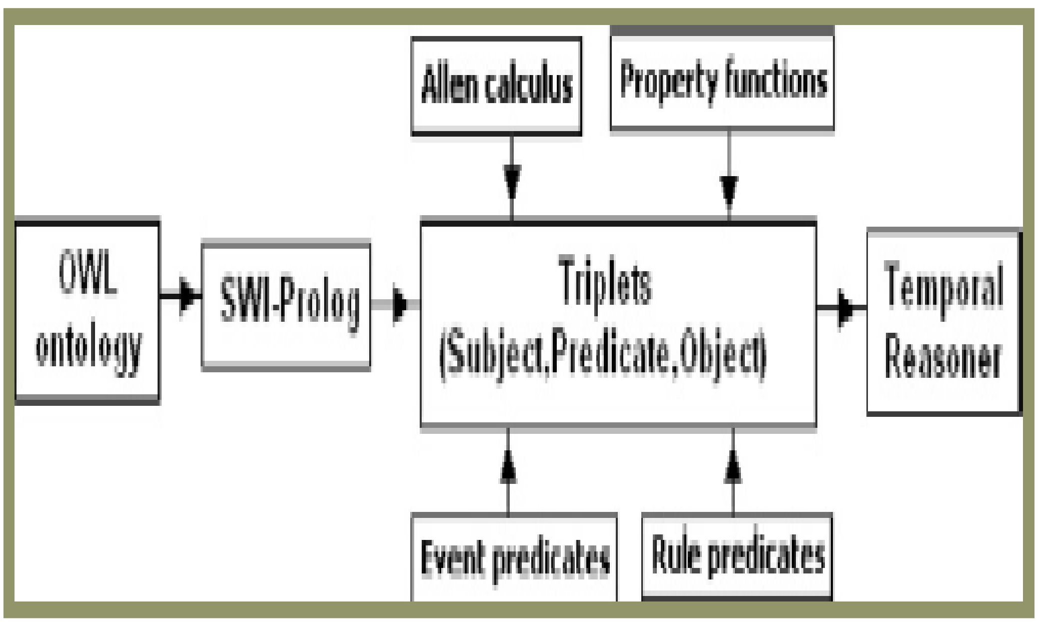

Fig 3 the architecture of PROTON.

4.6.1 PROTONS modules

The following modules are the basic ones Proton's architecture consists of.

1. SWI-Prolog is used for the transformation of temporal OWL conceptions in to prolog measures.

2. The Allen calculus is used for the computation of relationships on time intermissions. [85]

3. A group of functions is used in order to the compute the values of properties at every given temporal occurrence.

4. A group of statements is used for the determination of the instance an event took place.

5. A set of declarations which implement rules is used every time an event occurs or once an alteration in the value of a property takes place.

4.7 Snap

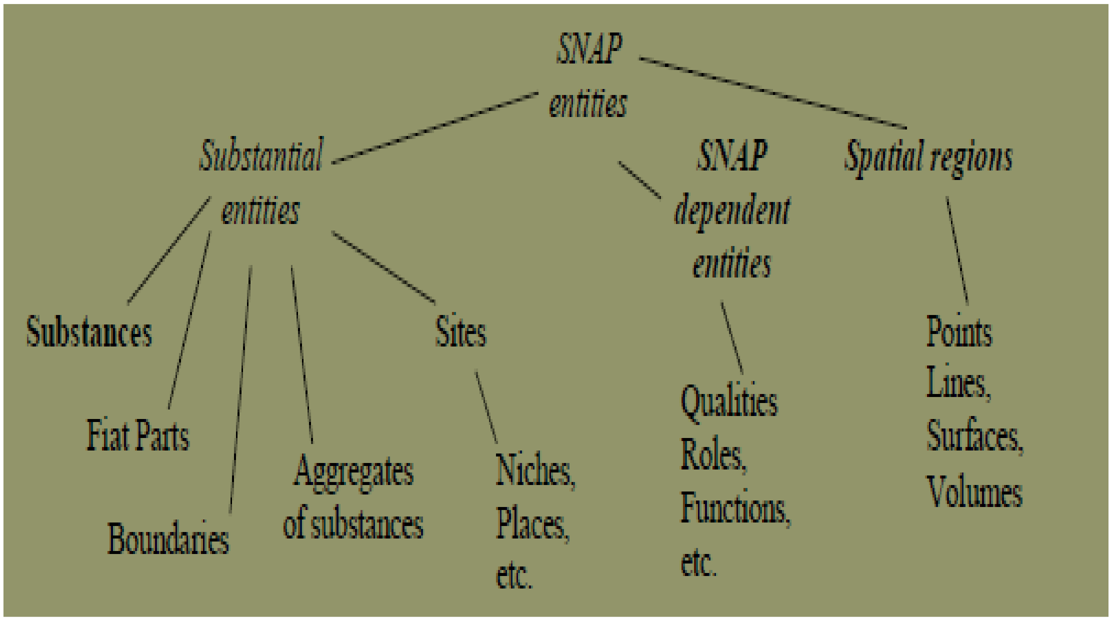

Fig. 4 The architecture of Snap

A SNAP ontology is fashioned via the representation of continuing individuals which exist a specified period, in a specific area, on a specified granularity. [30] The ingredients of a Snap ontology are the SNAP entities. Every Snap ontology is categorized to a certain temporal instant and the main Snap entity is dissective and cumulative simultaneously. [86]The entities which are recognized by a SNAP ontology are the entities which exist during the period of its index, and have usually already existed and will continue to exist in the future. The ingredients of a Snap ontology are the Snap entities. The basic Snap entity is characterized by dissection and collectiveness. Every Snap ontology is listed to a certain time instant. Solely continuing entities are identified by a Snap ontology as they are present at the time of its index, and also most probably existed in the past and will exist in the future. A Snap entities could be accurately described as not an instantaneous entity and definitely cannot be characterized as amounts of instantaneous shares. It is assumed as a variant [87] [88] [89] which grasps the total of entities that occur at the contemporaneous time. Every ontology of Snap is a momentous ontology which is related to a particular instant.

To be a component of Snap ontology equals to be in existence during the index of the ontology.

'ExistsAt(a, t)' stands for: 'the SNAP entity a exists at instant t'. [30]

$$\text{(D15)}\quad \text{ExistsAt}(x, t) \equiv_{\text{def}} \exists \omega\, (\text{Snap}\Omega(\omega) \wedge \text{TemporalIndex}(\omega, t) \wedge \text{Constituent}(x, \omega))$$

Durable entities occur uninterruptedly in great periods of time no matter what alterations they have been going through. Presence throughout a interval of time can be presented by assimilating through the temporal instants in which the Snap ontologies apprehend the endurants.

Exists During is defined by:

$$\text{(D16)}\quad \text{ExistsDuring}(x, y) \equiv_{\text{def}} \text{TimeRegion}(y) \wedge \forall z\, ((\text{TimeInstant}(z) \wedge \text{ProperPart}(z, y)) \rightarrow \text{ExistsAt}(x, z))$$

[30]

4.7.1 Spatial Reasoning

The reasoning which takes place inside a SNAP ontology is spatial reasoning. SPAN considers spatial regions as individual entities. One region which can be distinguished is space which basically is the utmost spatial region. Space contains all the spatial regions.

4.7.2 Substantial Entities

Substantial entities is a generality of the substances category which has as a result to relax the constraints of boundedness and connectedness.

4.7.3 SNAP Dependent Entities

SNAP dependent entities are the individual assets, functions, roles, dispositions, powers, and liabilities. In addition to the above SNAP dependent entities other types exist such as states, forms, plans, standards, tasks, rules. The common feature of all the above entities is their endurance through time and the fact that they require a foundation in SNAP independent entities. Although durability and dependence are basic characteristics of SNAP dependent entities, they are proven to be insufficient conditions. The characteristic that makes the dependent entities distinguishable is that they reside in substances.

4.8 SPAN

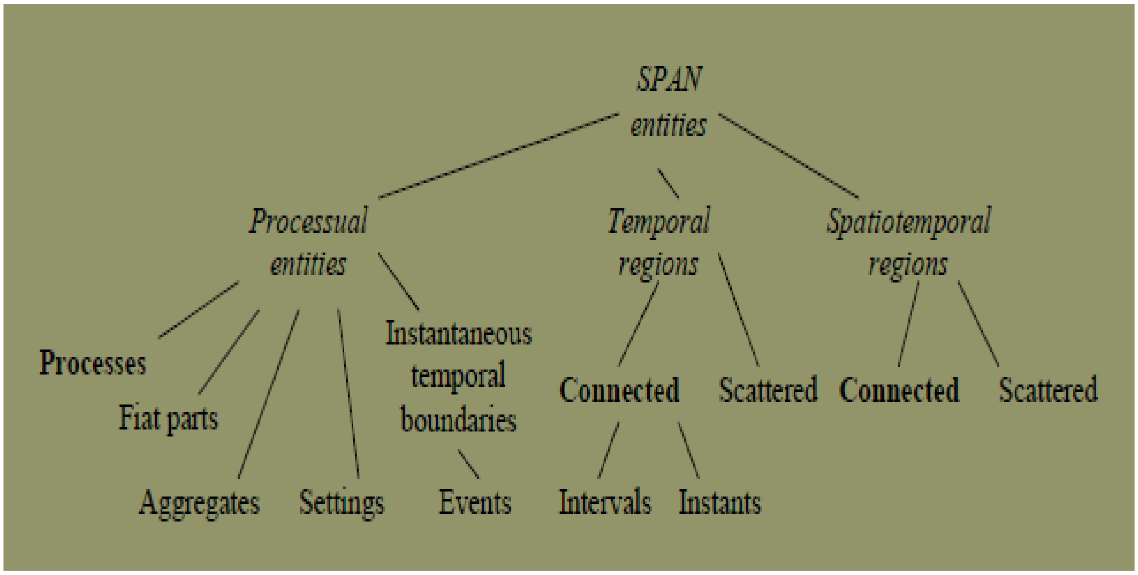

Fig 5. The architecture of SPAN

The entities, the temporal parts of which unfold through time are part of the ontological theory called SPAN.A SPAN ontology is acquired by portraying the truth which is constituted by the entities which develop themselves in a certain subdivision of time, at a certain reality and at a specific level of granularity. [30]SPAN uses a procedural theory of time, which considers the philosophical approach that past, present and future exist on a balance. A global SPAN ontology has time as a basic component. All SPAN ontologies have certain temporal regions as essential components. There is the notion that Span entities are situated in space, this is based on the detail that each area of space may communicate with a specific instant of space-time. Span ontologies understand regions which are extended spatial and temporally and also comprehend the changes which are situated at these regions.

4.8.1 Processual Entities

Processual entities play the role that substantial entities play for substances, to processes. Proccesual entities engulf processes, fiat parts, aggregates and boundaries. [90]

4.8.2 Temporal Regions

As time is a perdurant, the ultimate temporal region is of course A SPAN entity. A temporal location can be assigned to each and every SPAN entity. Each temporal location of every SPAN entity is sole. The way substantial entities can alter their positions in space does not apply to processual entities which don’t change their space location.

Time Region is as seen below:

(D23) $\text{TimeRegion}(x) \equiv_{\text{def}} \text{Part}(x, \textit{time})$ [30]

A time instant is a nil dimensional border of prolonged temporal regions. A specialism of the established Time Region is called Time Instant. A firm directive which stands among two instants of time once the first is former than the second. A temporal location is possible to be allocated to every Span entity. By mentioning Temporal Location it's meant as a original relationship among a region of time and an entity.

('TemporalLocation(a, t)' stands for: 'a is located at region of time t').

$$\text{SpanEntity}(x) \rightarrow \exists y\ (\text{TimeRegion}(y) \wedge \text{TemporalLocation}(x, y))$$ [30]

Every SPAN entity beholds an exclusive location hence the Temporal Location is efficient:

$$(\text{TemporalLocation}(x, y) \wedge \text{TemporalLocation}(x, z)) \rightarrow z = y$$ [30]

A process entity does not have the ability to alter its temporal location just like real life entities alter their space localities.

Instant SPAN entities are sited tie instances.AT Time:

$$\text{AtTime}(x, y) \equiv_{def} (\text{TemporalLocation}(x, y) \wedge \text{TimeInstant}(y))$$ [30]

A number of temporal relationships which hold their corresponding temporal spaces between Span entities ate the exact same region of time:

$$\text{Cotemporal}(x, y) \equiv_{def} \textit{temporal-location}(x) = \textit{temporal-location}(y)$$ [30]

Every fragment of time is the sum of the totality of cotemporal shares of an entity of SPAN which is sited inside a specific region of time. Instant Temporal slices are expressed as:

$$\mathrm{TemporalSlice}(x, y) \equiv \mathrm{def}\ \mathrm{TemporalPart}(x, y) \wedge \exists t\ \mathrm{AtTime}(x, t) \quad [30]$$

Temporal Slice events are situated in an instant of time something which gives the ability to apprehend proceedings of both arbitrary order and authentic kinds.

4.8.3 Spatiotemporal Regions

The entirety of spatiotemporal areas represents the sum of all achievable boundaries of the space-time area. In the eternalism of SPAN, the existence of spatiotemporal regions takes place in harmony with each other inside the spatiotemporal universe.

4.9 SPARQ

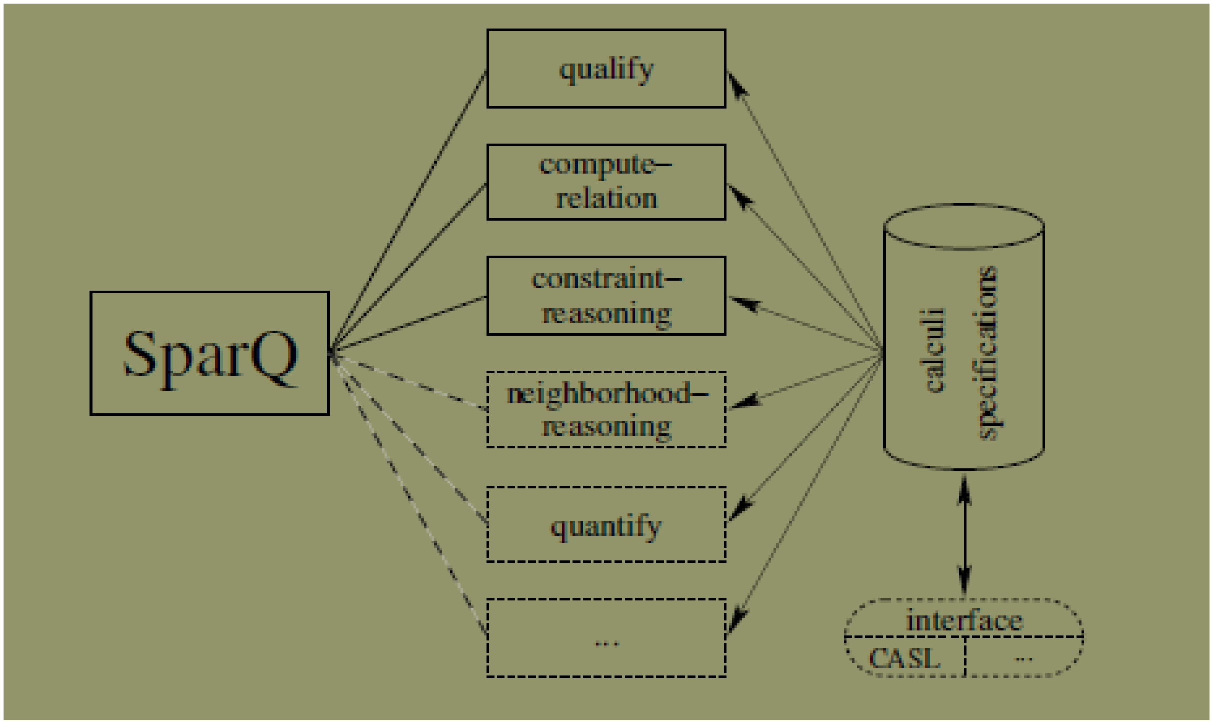

Fig 6. The architecture of SPARQ

The main focus of SPARQ is the calculi originating from reasoning concerned with the orientation of point objects and line sections. Having said that the development and specification of additional calculi is straight forward. [31]SPARQ stands for Spatial Reasoning done qualitatively and has as a goal to offer a toolkit which brings qualitative reasoning and Applications together. The tools contained in SPARQ serve to connect quantitative and qualitative information, binary calculi with qualitative reasoning. [91][92] Additionally SPARQ provides fertile ground for the development of new calculi.

The users of SPARQ have the ability to determine their own custom calculi or alternatively use the expanding calculi depository which accompanies the reasoner. The main objectives of SPARQ are to offer sample applications geared towards spatial calculi originating from QSR. Additionally it's oriented towards the specification and integration of new calculi. SPARQ provides classic procedures in order for the application of QSR to be done conveniently. [93]It also provides a standard interface which makes the interchangeability among calculi easier.

4.9.1 SPARQ modules

A group of modules which deliver distinguished services used for QSR are the structure of SPARQ.A central script holds the modules connected to each other. The SPARQ modules are qualify which converts a quantitative geometric depiction which represents a spatial configuration in to a qualitative depiction which is based on the supported calculi. Another module is computer relation which applies the predefined calculi operations such as union, composition, intersection to a spatial relations set. The mission of the constraint reasoning module is to fulfill constraint network computations. The qualify module supports the goal of transforming a quantitative geometric scene depiction in to a specific calculus, base relation qualitative scene depiction. The presence of qualification is of utmost importance for applications which aim to perform qualitative computations to objects with known geometric data. The compute relation module enables the computation with the operations which are set in the specifications of the calculus. The constraint reasoning module performs a type of consistency check by reading the qualitative scene description (constraint network description) which there is a chance that it contains inconsistencies and disconnections.

4.9.2 Calculi in SPARQ

It's quite straight forward for SPARQ to embrace the majority of calculi as long as the calculus specifications are provided. It is important for the base relations, the diverse operations and the logic of the calculus specifications to be provided. SPARQ has the potentially able to be incorporated in to particular applications as it has the capability to be executed in server mode.

4.10 TACHYON

Tachyon is a constraint based temporal reasoner which was created under the notion that a temporal reasoning system must support a powerful and flexible capability for workflow. [32]Tachyon offers a large number of features which support the above notion. Tachyon is able to handle the fact that it's never easy to guarantee when and how long an event will be. Tachyon has the ability to express qumitative and quantitative constraints between events. Another characteristic of Tachyon is that it can express parameterized qualitative constraints among events. Ease of use is a main goal of Tachyon which achieves it by graphical input and display abilities. Tachyon can be executed individually and as a sub process in applications. The techniques which are used by Tachyon offer operability regardless of the size of application domains. Tachyon finally copes efficiently with the complexity related to disjoint constraints.

4.10.1 Representation

Tachyon symbolizes occurrences by the use of 6 tuples in order to aid in the seizure of uncertainty in an event incidence. An occupancies initial and final probable start and finish times are represented by the 6-tuple.Additionally the constraint expression is spread in order for it to permit quantitative and qualitative relations among events. In order for the identical data to be represented in different models an event need to be distributed to a start and finish event and a duration indicator constraint among the two. A single node is able to depict a real life event and to provide justification for duration and improbability.

The expressiveness for an additional event requires a similar expansion of the constraint model. The networks used by Tachyon incorporate a depiction of spaces with values among events. These quantifiable constraints put arithmetic borders on the temporal relationship. Allen's linguistic relationships can also be utilized in Tachyon by the addition of some relation parameters. The qualitative relationships such as before, overlaps and after have as an option to be parameterized. Two parameters are to be engulfed by each qualitative relationship in order to depict the minimum and maximum range they denote. Caution is needed when handling these parameterized qualitative relationships because the danger of insolvability is lurking. An 8-tuple expresses the edge constraints in Tachyon.

4.10.2 Reasoning with Tachyon

Tachyon uses the Bellman Ford shortest path algorithm modified in way so that it spreads information and stiffens the boundaries of variables in a graph exclusively consistent of convex constraints and it presents differences equated to the Floyd-Warshall algorithm. The Bellman-Ford algorithm as well as the Floyd-Warshall algorithm have O ($n^{3)}$ time complexity and n is the amount of nodes. It is essential to specify problematical constraints when they present the problem of intractability to the reasoning process. Practical planning applications commonly face this problem which is referred to as non-convex constraints. It was proved in testing that the Bellman-Ford algorithm offerd a great increase in performance compared to the Floyd-Warshall algorithm. Tachyon is greatly more capable compared to alternative solutions.

4.10.3 Nonconvex Constraints

Occasionally there is the need for constraint specification which are problematic for the introduction of complexity in the reasoning procedure. These constraints are named disjoint and take place in a great number of scheduling applications.

These disjoint constraints which entail execution of numerous tasks on a specific machine without the need of prioritization among the tasks is a common occurrence in scheduling realms.Tachyon has the ability for disjoined constraint specification and also is capable to resolve such issues. In order for a reasoned to be able to resolve such problems for large systems there is the need for overcoming the big size of consistent solutions. Such a problem requires an enormous amount of time to resolve.

4.11 GQR

GQR stands for Generic Qualitative Reasoner and its purpose is to act as a solving engine for qualitative constraint networks. The way GQR works is that it takes as input a calculus depiction and a constraint network as input and by the use of backtracking and the path consistency methodology attempts to solve the networks. An interesting attribute of GQR which differentiates it from other specialized reasoners is that it offers reasoning amenities for different qualitative calculi, subsequently it’s obvious that these services are not hardcoded in to the GQR.[33]GQR supports RCC family calculi, OPRA family calculi, cardinal direction calculi, and Allen’s interval algebra. GQR is free and it’s a system designed to be generic and extensible while remaining efficient and scalable.

4.11.1 Qualitative Reasoning with GQR

In order to determine if a task to the variables of a constraint network occurs so that all its constraints are satisfied, the constraint satisfaction problem is utilized. GQR solves the constraint satisfaction problem by checking if the constraint graph is stable .Being consistent means that there is a path consistent enhancement originating from the constraint graph where each edge consists a base relation. The above approach has been proven to be more than sufficient for the solution of the satisfaction problem.

The reasoning process in GQR is established exclusively on a qualitative calculi description. [94]The way reasoning problems are expressed in qualitative calculi is called constraint satisfaction. [95]The occurrences these complications face is possible to be depicted as constraint networks which are focused limited graphs, in which every edge is branded from a calculus relationship. These graphs are depicted in GQR in the form of square matrix in which every admission is a bit vector used for encoding the calculi relationship among a pair of nodes in a graph. Afterwards the challenge for constraint restriction is to conclude if an assignment is in existence on its modules in a specified domain named D so that the totality of the network constraints to be true. [96]Furthermore another kind of usual reasoning assignment is to verify if a constraint is included in a constraint network and to calculate a corresponding negligible constraint network. These assignments of reasoning are able to be depicted as equal under the polynomial Turing reductions. [97]The fact the primal models in qualitative reasoning are most of the times inestimable is a significant attribute so in the case of testing SAT of a constraint network it's impossible to estimate the potential tasks to variables inside a model.

4.11.2 Primary algorithms used in GQR

A main way to resolve constriction networks in calculi is the use of the path consistency algorithm. An unsophisticated employment of the pre mentioned procedure would require O (n^5) intersections. In [95] it's mentioned that another approach places the paths which might be touched by certain changes in a queue and then proceeds to run the queue until it becomes empty. This procedure necessitates time of O (n^3) and demands O (n^2) memory.

The pre mentioned method is depicted by Algorithm 1:

Algorithm 1 Path-Consistency(V,C)

Input: A constraint network (V,C)
Output: A constraint network (V,C') that is a (sometimes even path-consistent) refinement of (V,C)

$Q \leftarrow \{(i,j) \mid 1 \leq i < j \leq n\}$ *// Initialize the queue*
while Q is not empty **do**
  select and delete an (i,j) from Q
  for $k \leftarrow 1$ **to** $n, k \neq i$ and $k \neq j$ **do**
    $t \leftarrow C_{ik} \cap (C_{ij} \circ C_{jk})$
    if $t \neq C_{ik}$ **then**
      $C_{ik} \leftarrow t$
      $C_{ki} \leftarrow t^{\smile}$
      $Q \leftarrow Q \cup \{(i,k)\}$
    end if
    $t \leftarrow C_{kj} \cap (C_{ki} \circ C_{ij})$
    if $t \neq C_{kj}$ **then**
      $C_{kj} \leftarrow t$
      $C_{jk} \leftarrow t^{\smile}$
      $Q \leftarrow Q \cup \{(k,j)\}$
    end if
  end for
end while
return (V,c)

Algorithm 1[33]

A vital detail as far as the creation of a new binary calculi is concerned has to with the fact that it must recognize the tractable subclasses. Certain subclasses network which originates form the path consistency is marginally equivalent. In the case that path consistency results to a non-atomic network different manifestations of the constraints must be used. That Algorithm 2 functions is described as follows: in the case that a search ended up in a road block we continue by checking the following likely instantiation of the latest altered constraint. The method described above is named chronological backtracking and right afore every instantiation for the trimming of the search tree the path consistency method is applied. The path consistency method is needed solely to be executed for the tracks which are feasibly afflicted by the previous instauration where it utilizes O (n^2) junctures and configurations.

Algorithm 2 Consistent(V,C)

```
Input: A constraint network (V,C)
Output: A Boolean value that is true if and only if (V,C) is
    satisfiable
Path-Consistency(V,C)
if C contains the empty relation then
    return false
end if
if there are non-basic edges then
    Pick such an edge e = (i, j)
    for all base relations b in the label of e do
        C_ij ← b
        if Consistent(V,C) then
            return true
        end if
    end for
end if
return false
```

Algorithm 2[33]

4.12 Categorization of reviewed reasoners based on their specific use in the industry.

The reasoners reviewed in this thesis besides being categorized as temporal, spatial, spatiotemporal can also be categorized by the use of the comparison table which follows this chapter by comparing the attributes of each one and thus being able to reach to a conclusion as a developer on what to use for a specific project. Another way which is possible to categorize a reasoner is by providing a list of the applications each one is predominately used for in the field.

Starting with the spatial reasoners Pellet spatial is used in small networks as it does not scale up to more than a few thousand relations. Oncor is used for providing an approach for ontology development for lightweight ontologies of spaces and devices. Choros provides a definition of an RDF/OWL vocabulary and provides representation for spatial relationships as assertions of OWL object properties. Snap is ideal for examining georegions in relation to geographical qualities such as for example the depth of a gorge. Sparq is ideal for GIS system developers because it allows them to concentrate on the aspect of core functionality instead of converging on data fabrication. Temporal reasoners such as Proton specialize in providing data for queries which concern proceedings which alter in time. Tachyon offers workflow automation for health services by designing and tracing clinical flow processes. Gqr belongs to the category of reasoners which offer reasoning services for various qualitative calculi instead of being specialized to just one.

	CHRONOS	Pellet Spatial	ONCOR	CHOROS	SOWL	PROTON	SNAP	SPAN	SPARQ	TACHYON	GQR
Methodology											
Soundness	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
Completeness	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
Expressivity	SROIQ	SHIQ	EL+	SROIQ (D	EL+	SHIQ	SROIQ	EL+	SHIQ	SROIQ	SHIQ
Native Profile	DL	DL	DL	DL	DL	DL	DL	DL	DL	DL	DL
Incremental Classification	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
Rule Support	YES(SWRL)	YES(SWRL)	YES(SWRL)	YES(SWRL)	YES(SWRL)	YES(SWRL)	YES(SWRL)	YES(SWRL)	YES(SWRL)	YES(SWRL)	YES(SWRL)
Platforms	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL
Justifications	YES	YES	YES	NO	YES	YES	YES	YES	NO	YES	YES
ABOX reasoning	YES	NO	YES	YES	YES	YES	YES	YES	YES	NO	YES
OWL API	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
OWL LINK API	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
Protégé Plugin	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
License	FREE	FREE	FREE	FREE	FREE	FREE	FREE	FREE	FREE	FREE	FREE
Jenna Support	NO	NO	NO	NO	YES	NO	NO	NO	NO	NO	NO
Implementation Language	JAVA	JAVA	C++	JAVA	C++	PROLOGUE	JAVA	JAVA	JAVA	JAVA	C++
Availability	OPEN SOURCE	OPEN SOURCE	OPEN SOURCE	OPEN SOURCE	OPEN SOURCE	OPEN SOURCE	OPEN SOURCE	OPEN SOURCE	OPEN SOURCE	OPEN SOURCE	OPEN SOURCE
Inference Support	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES	YES
Support for changes in the past	YES	NO	NO	NO	YES	YES	NO	YES	NO	YES	YES
Support for spatial Data	NO	YES	YES	YES	YES	NO	YES	YES	YES	NO	YES

Table 1 Comparison table of Spatial-Temporal reasoners.

5. Spatial and Temporal reasoners theoretical and experimental Evaluation.

This chapter is concerned with the efficiency of the reasoning approaches followed by the reviewed reasoners both temporal and spatial. The efficiency of the reviewed reasoners is going to be assessed both theoretically and experimentally. First we will examine the efficiency of the temporal reasoners under review and then the efficiency of their spatial counterparts.

5.1 Temporal reasoners evaluation

When rules demanded by reasoners apply over relations of temporal nature the goal is to conclude implied relations, inconsistency detection and path consistency preservation. James F. Allen suggested in 1983 basic relations among intervals of time. These Allen relations have three distinctive characteristics which are that the relations are distinct, exhaustive and in addition to the pre mentioned, qualitative. Allen relations are distinct for the reason that a couple of definite intervals can only be termed by the use of one of the available relations. One relation designates any couple of definite intervals hence the characterization exhaustive. There is no consideration of quantitative data so it’s qualitative. A total number of 6 relation couples are converses. Intervals of fixed intervals are portrayed by the Allen basic relations, on the other hand non defined intervals with undefined relations are portrayed by a group of all of Allens basic relations.

Basic relation		Illustration	Endpoints
I precedes J	p	`III`	$I^+ < J^-$
J preceded by I	p$^\smile$	`      JJJ`	
I meets J	m	`IIII`	$I^+ = J^-$
J met by I	m$^\smile$	`    JJJJ`	
I overlaps J	o	`IIII`	$I^- < J^- < I^+$,
J overlapped by I	o$^\smile$	`  JJJJ`	$I^+ < J^+$
I during J	d	`   II`	$I^- > J^-$,
J includes I	d$^\smile$	`JJJJJJ`	$I^+ < J^+$
I starts J	s	`III`	$I^- = J^-$,
J started by I	s$^\smile$	`JJJJJJ`	$I^+ < J^+$
I finishes J	f	`   III`	$I^+ = J^+$,
J finished by I	f$^\smile$	`JJJJJJ`	$I^- > J^-$
I equals J	$\equiv$	`IIII` `JJJJ`	$I^- = J^-$, $I^+ = J^+$

Table 2.The basic relations of Allen's algebra. [54]

In our evaluation we make the hypothesis that only basic Allen rules apply hence our reasoning is ensured polynomial time complexity. Basic Allen relation relations have the characteristic of mutual exclusiveness, maximum $O(n^2)$ relation are possible to be contingent as in n intervals it is known that every time interval is possible to be connected with every intermission on a single basic Allen relation. Additionally for the set of relations which are supported path consistency has $O(n^3)$ utmost complexity.

Experimental Evaluation

The following experiments aim to project the run time efficiency of the temporal reasoners reviewed in this survey. To achieve the performance evaluation we are reaching for several experiments were executed. The tests signified the reasoning run time efficiency of the extent of input information as a function. The running time of CHRONOS, SOWL, PROTON, SPAN, TACHYON, and GQR is compared as an operation of the numeral of time-based statements in the ontology. In order for the execution period effectiveness of our reasoning to be studied an informations group was used covering amid 10 and 100 temporal statements. The pre mentioned statements were derived from real declassified general tourism agency archives. From these archives a number of 200 notes were extracted from different agents. Out of the pre mentioned 200 notes a total of 100 temporal statements was extracted .The statements were used in order to inhabit the CNTRO ontology [97] with arbitrary instances and the reasoning duration is calculated as averages over 10 repetitions. CNTRO which stands for Clinical Narrative Temporal Relation Ontology is basically an ontology which has the capability of modelling temporal info derived from regulated databanks and has additionally the characteristic of deriving information from natural linguistic established reports. Two separate experiments took place the first had to do with the performance as far as the average case is concerned and the second experiment had to do with the worst case. Once less than n^2 entities are concluded as of an input group comprised of n time-based intermissions .The average case experimentation is in the matter of n. We encounter poorest case performances once the number of concluded relations is n^2.

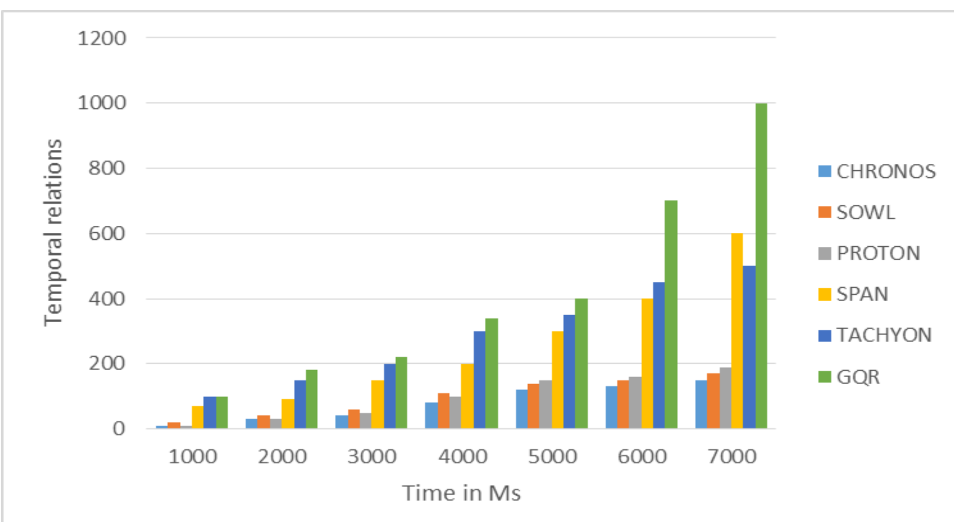

Fig. 7. Average response time (over 10 iterations) of reasoning as a function of the number of temporal assertions in an ontology.

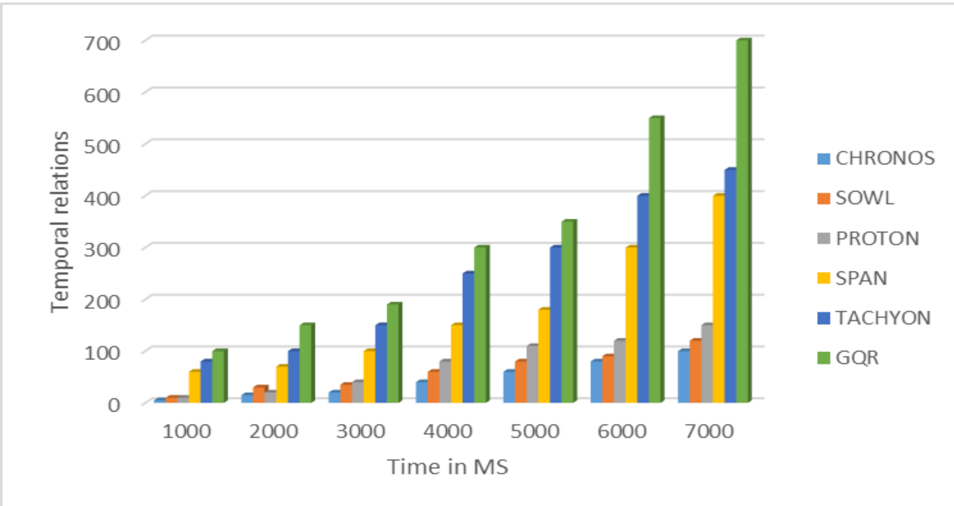

Fig 8 Worst case response time) of reasoning as a function of the number of temporal assertions in an ontology.

In figure 7 the mediocre response time of the CHRONOS, SOWL, PROTON, SPAN, TACHYON, and GQR reasoners is displayed and the results provide sufficient information as long as the capabilities of each temporal reasoner is concerned. In figure 8 the worst case response time of CHRONOS, SOWL, PROTON, SPAN, TACHYON, and GQR is displayed. It is important to declare that the tests of this survey were done on a Windows 10 PC with an I5 processor and 8 gigabytes of Ram. Certain reasoners may be able to perform better on computers with bigger RAM size due to the fact that large data sets cause memory overflow because of the larger number of inferred relations. Out of the six temporal reasoners which where reviewed in this thesis GQR displayed the best results and scales up much better than its counterparts when the bulk of the data groups escalates and will perform much faster than the rest of the temporal reasoners.

5.2 Spatial reasoners evaluation

The objective of the investigational evaluation of the reviewed spatial reasoners which are reviewed in this survey is to demonstrate the performance characteristics of these reasoners. The experiment carried out takes under consideration measurements in the average cases and in the average case portrayal is met when no more than n^2 relationships are contingent orientating from a set of n locations. The kn relations situated in the experimental phase are k=8 intended for RCC-8 and k=9 intended for CSD-9 are declared. In order to reach an objective result as far as the evaluation of the spatial reasoners reviewed in this survey the OSM spatial ontology was used. OSM is a collective venture which aims to provide an open source world map. OSM preserves an assortment of widely used tags generally used for focal map topographies. A map of the island of Crete was used in order to generate and discriminate the relative positions of cities as they are projected on the map (google maps).

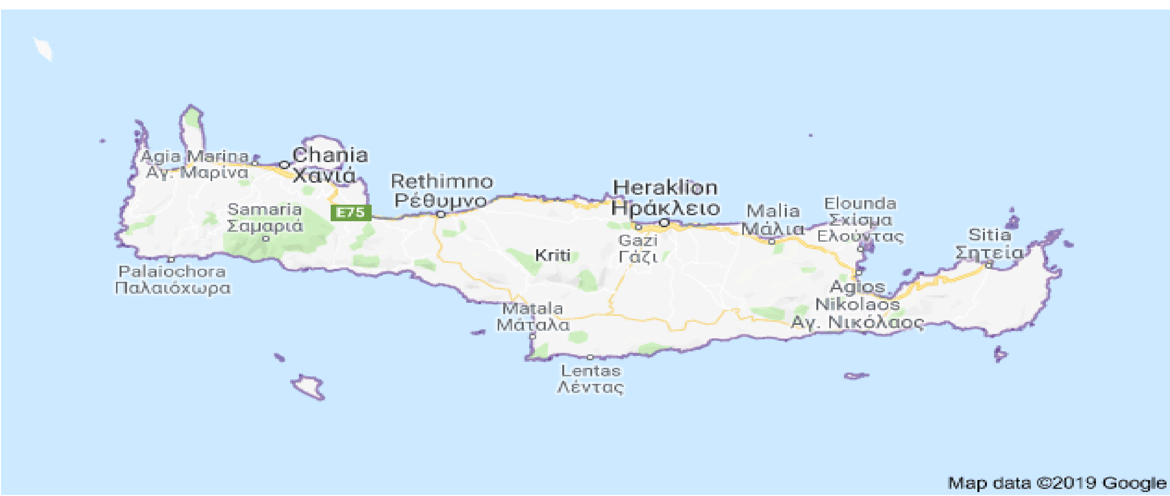

Fig. 9 Map of the island of Crete

Additional instances were generated by the usage of an arbitrary numerical generator in order to achieve a representative spatial ontology comprised of 100 constituencies' over-all. Particularly 70 additional instances were created and each one contained a single CSD—9 relation and a single RCC-8 relation and at the end these instances represented points of cities. Producing arbitrary occurrences and reasoning on the subsequent ontology is reiterated 10 intervals so that the reasoner reaction period conveyed further down is the average in excess of 10 innings. Each particular can only be compared with one other particular. The execution times which are conveyed are norms over 10 spatial ontologies. As a standard the execution time of all reasoned applications was compared as a function of the amount of entities. The equipment which was used for the experiment was a Windows 10 PC with an I5 processor and 8 gigabytes of Ram.

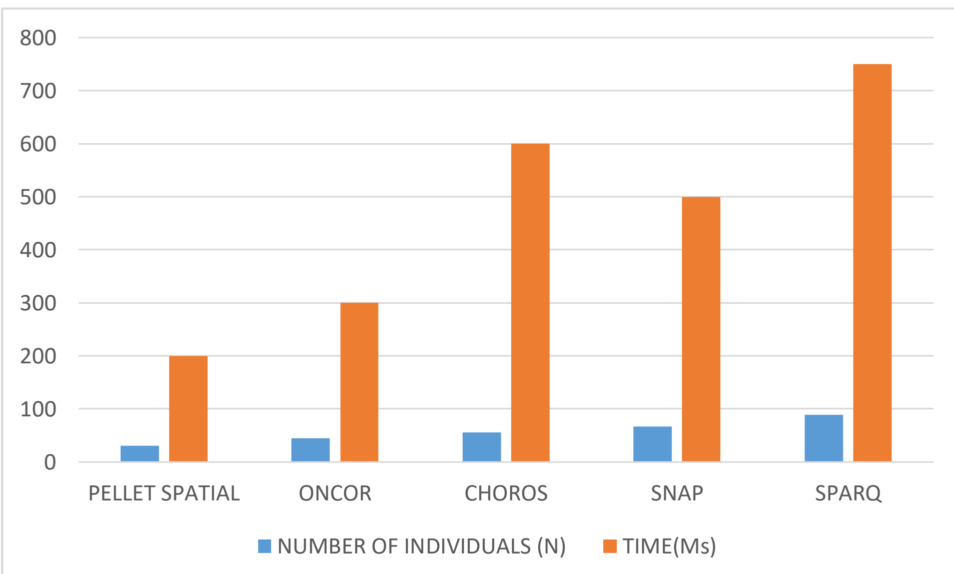

Fig.9 Average case concerning the performance of reasoning on disintegrated CSD relationships.

The average case performance provides a more distinguishing view of the reasoners characteristic performance hence the use of decomposition reasoned in all of our experiments was chosen. The worst case performance measurements demanded the form of the ontology to be such that from n declared relationships we can conclude n^2 relationships. In order to achieve such a result relationships must be declared which firstly must be consistent and secondly the reasoning process over these relationships harvests relationships among the totality of the pairs of entities in the ontology.

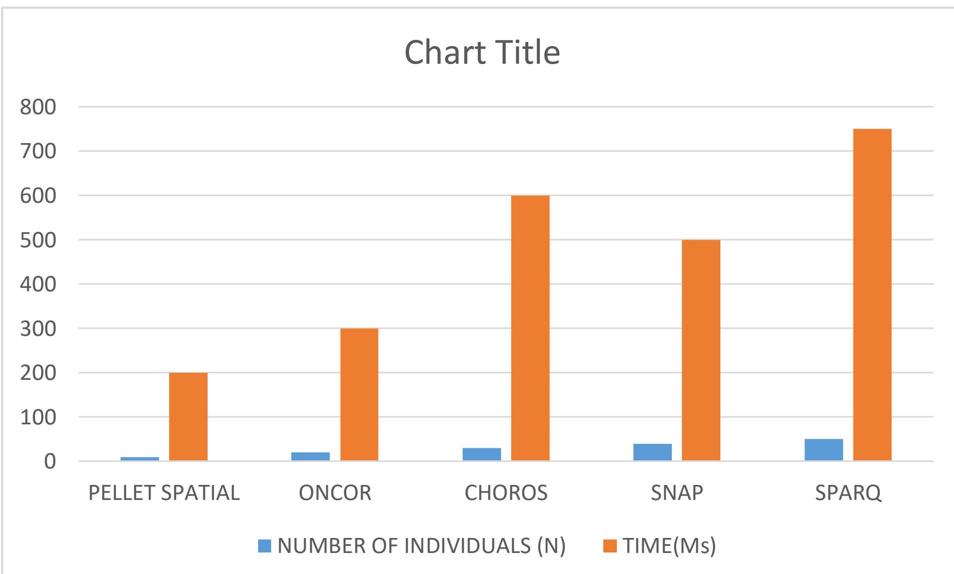

Fig 10. Poorest case performance of reasoning on disintegrated CSD relations.

The worst case enactment takes place when reasoning on CSD-9 and RCC-8 relationship groups has O (n^3) complication.

Assessment outcomes on a representative ontology with 100 states confirmed that spatial reasoning executes virtually in line with the quantity of regions in the contribution ontology in the average case. For an ontology explicitly intended for the poorest case, reasoning is adjacent to the square enactment. Sparq is the best performing spatial reasoner out of the five which were examined in both the average and the worst case scenarios and is going to be subject for further study.

Conclusion and future scope

The goal of this paper was to impact the way a reasoning engine spatial or temporal is chosen for a specific task. Firstly the attributes which are used for the evaluation of reasoners where analyzed and then an extensive description as well as in depth analysis of 11 spatial and temporal reasoners was provided. It was demonstrated clearly that each reasoner differs from another in such a great degree that in order for a user to choose one for a specific task in a real world environment the requirements need to be assessed and evaluated accordingly. In addition to the detailed analysis of the reasoning engine this paper proceeds by providing an evaluation chapter for both spatial and temporal reasoners. By the incorporation of experimental data in the theoretical analysis of the reasoners this paper offers a clear view of the strengths and weaknesses of the spatio-temporal reasoners establishment. The future beholds ambitious plans for the writers of this survey as it’s already established that the creation of two reasoning engines one temporal and one spatial will take place in order for the shortcomings of the reviewed reasoners to be overthrown.

References

[1]L.Vila. A survey on temporal reasoning in Artificial Intelligence. AI Communications 7(1), 4-28, 1994.

[2]Stuckenschmidt, Heiner (2009) Spatial Reasoning for the SemanticWeb - Use Cases and Technological Challenges. In: Generalization of spatial information: Dagstuhl Seminar 09161, 13.04. - 17.04.2009 2009 Wadern [Conference or workshop item]

[3]Sunitha Abburu. Article: A Survey on Ontology Reasoners and Comparison. International Journal of Computer Applications 57(17):33-39, November 2012.

[4]I. Bloch, Fuzzy spatial relationships for image processing and interpretation: a review, Image Vision Comput. 23 (2) (2005) 89–110.

[5]B.J. Kuipers, T.S. Levitt, Navigation and mapping in large-scale space, AI Magazine 9 (2) (1988) 25–43.

[6]J. Freeman, The modelling of spatial relations, Comput. Graphics Image Processing 4 (2) (1975) 156–171.

[7]B. Kuipers, Modeling spatial knowledge, Cognitive Sci. 2 (1978) 129–153.

[8]L. Vieu, Spatial representation and reasoning in artificial intelligence, in: O. Stock (Ed.), Spatial and Temporal Reasoning, Kluwer, Dordrecht,1997, pp. 5–41.

[9]Avery, J., & Yearwood, J. (2003). DOWL: A dynamic ontology language. University ofBallarat.

[10]Welty, C., Fikes, R., & Makarios, S. (2004). A reusable ontology for fluents in OWL,IBM Research paper.

[11]Di Noia, T., Di Sciascio, E., & Donini, F. M. (2007). Semantic matchmaking as nonmonotonic reasoning: A description logic approach. Journal of Artificial Intelligence Research (JAIR), 29, 269–307

[12] Wallgrün J.O., Frommberger L., Wolter D., Dylla F., Freksa C. (2007) Qualitative Spatial Representation and Reasoning in the SparQ-Toolbox. In: Barkowsky T., Knauff M., Ligozat G., Montello D.R. (eds) Spatial Cognition V Reasoning, Action, Interaction. Spatial Cognition 2006. Lecture Notes in Computer Science, vol 4387. Springer, Berlin, Heidelberg

[13] Gantner, Z., Westphal, M., & Wölfl, S. (2008).GQR-a fast reasoner for binary qualitative constraint calculi. In Proceedings of the AAAI'08Workshop on Spatial and Temporal Reasoning (pp. 24-29). Chicago, IL: AAAI.

[14] Bredeweg, B., Bouwer, A., Jellema, J., Bertels,D., Linnebank, F., & Liem, J. (2007). Garp3: A new workbench for qualitative reasoning and modelling. In Proceedings of the 4th International Conference on Knowledge Capture (K-CAP'07),(pp. 183-184). K-CAP.

[15] Kathrin Dentler, Ronald Cornet, Annette ten Teije and Nicolette de Keizer, "Comparison of Reasoners for Large Ontologies in the OWL 2 EL Profile", Semantic Web Journal, 2011, pp.1-5.

[16] I. Horrocks, U. Sattler, and S. Tobies. Practical Reasoning for Very Expressive Description Logics. Logic Journal of the IGPL, 8(3):239–264, 2000.

[17] Y. Kazakov. Consequence-Driven Reasoning for Horn SHIQ Ontologies. In Proceedings of the 21st International Conference on Artificial Intelligence, pages 2040–2045, 2009.

[18] B. Motik, R. Shearer, and I. Horrocks. Optimized Reasoning in Description Logics using Hypertableaux. In Proceedings of the 21st International Conference on Automated Deduction,volume 4603, pages 67–83. Springer, 2007.

[19] S. Rudolph, T. Tserendorj, and P. Hitzler. What Is Approximate Reasoning? Web Reasoning and Rule Systems, 5341:150–164, 2008.

[20] M. Horridge and S. Bechhofer. The OWL API: A Java API for Working with OWL 2 Ontologies. In 6th OWL Experienced and Directions Workshop, 2009.

[21] F. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Reasoning in Description Logics. Principles of knowledge representation, pages 191–236, 1996.T. Liebig, M. Luther, O. Noppens, M. Rodriguez, D. Calvanese,M. Wessel, R. Möller, M. Horridge, S. Bechhofer,D. Tsarkov, and Others. OWLlink: DIG for OWL 2. In 5th OWL Experienced and Directions Workshop, 2008.

[22] Introduction to Semantic Web", available at http://www.mphasis.com/knowledge-center/white-papers-all.asp.

[23] Eleftherios Anagnostopoulos, Sotiris Batsakis, Euripides G.M. Petrakis, CHRONOS: A Reasoning Engine for Qualitative Temporal Information in OWL, Procedia Computer Science, Volume 22, 2013, Pages 70-77, ISSN 18770509,https://doi.org/10.1016/j.procs.2013.09.082

[24] Stocker, M., & Sirin, E. (2009). PelletSpatial: A Hybrid RCC-8 and RDF/OWL Reasoning and Query Engine. OWLED.

[25] Kay, Judy & Niu, William & J. Carmichael, David. (2007). ONCOR: ontology- and evidence-based context reasoner.. 290-293. 10.1145/1216295.1216349.

[26] Christodoulou, George & Petrakis, Euripides & Batsakis, Sotirios. (2012). Qualitative Spatial Reasoning Using Topological and Directional Information in OWL. Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI. 1. 596-602. 10.1109/ICTAI.2012.86.

[27] Niu, W.T., & Kay, J. (2008). Personalised and Scrutable Ontological Reasoning for Pervasive Computing Technical Report 625.

[28] Batsakis, Sotiris & Petrakis, Euripides. (2010). SOWL: Spatio-temporal representation, reasoning and querying over the semantic web. I-SEMANTICS. 10.1145/1839707.1839726.

[29] Nikos Papadakis, Kostas Stravoskoufos, Evdoxios Baratis, Euripides G.M. Petrakis, Dimitris Plexousakis, PROTON: A Prolog Reasoner for Temporal ONtologies in OWL,Expert Systems with Applications, Volume 38, Issue 12,2011,Pages 14660-14667,ISSN 0957-4174,https://doi.org/10.1016/j.eswa.2011.05.017.(http://www.sciencedirect.com/science/article/pii/S0957417411008001)

[30] Grenon, P., & Smith, B. (2004). SNAP and SPAN: Towards Dynamic Spatial Ontology. *Spatial Cognition & Computation*, *4*, 69-104.

[31] Dylla, Frank & Frommberger, Lutz & Wallgrün, Jan Oliver & Wolter, Diedrich. (2006). SparQ: A Toolbox for Qualitative Spatial Representation and Reasoning. 79-90.

[32] Haimowitz I.J., Farley J., Fields G.S., Stillman J., Vivier B. (1996) Temporal reasoning for automated workflow in health care enterprises. In: Adam N.R., Yesha Y. (eds) Electronic Commerce. EC 1994. Lecture Notes in Computer Science, vol 1028. Springer, Berlin, Heidelberg

[33] Gantner, Z., Westphal, M., & Wölfl, S. (2008). GQR – A Fast Reasoner for Binary Qualitative Constraint Calculi.

[34] H. Chen, F. Perich, T. Finin and A. Joshi, "SOUPA: standard ontology for ubiquitous and pervasive applications," *The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004.* Boston, MA, USA, 2004, pp. 258-267.doi: 10.1109/MOBIQ.2004.1331732

[35] N. Noy, and B. Rector. "Defining N-ary Relations on the Semantic Web". W3C Working Group Note 12, April 2006,http://www.w3.org/TR/swbp-n-aryRelations/

[36] S. Batsakis, and E.G.M. Petrakis. "SOWL: Spatio-temporal Representation, Reasoning and Querying over the Semantic Web" 6*th* International Conference on Semantic Systems (I-SEMANTICS 2010), Graz, Austria, 1–3 September 20

[37] C. Welty, and R. Fikes. "A Reusable Ontology for Fluents in OWL". Frontiers in Artificial Intelligence and Applications, 150:226–236, 2006.

[38] G. Christodoulou, E. Petrakis E., and S. Batsakis. "Qualitative Spatial Reasoning using Topological and Directional Information in OWL", 24th International Conference on Tools with Artificial Intelligence (ICTAI 2012), Athens, Greece, November 7-9, 2012.

[39] J. Renz, and B. Nebel. “Qualitative Spatial Reasoning using Constraint Calculi”. In Handbook of Spatial Logics, Springer, Netherlands,pp. 161-215,

[40] B. Parsia, and E. Sirin. “Pellet: An OWL DL reasoner.” Third International Semantic Web Conference (ISWC 2004), 2004.

[41] J. F. Allen. “Maintaining Knowledge about Temporal Intervals”. Communications of the ACM. 26:832-843, 1983.

[42] S. Batsakis, K. Stravoskoufos, and E.G.M. Petrakis. Temporal Reasoning for Supporting Temporal Queries in OWL 2.0. 15th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES 2011), 6881:558–567, 2011.

[43] B. Nebel, and H.J. Burckert “Reasoning about Temporal Relations: A Maximal Tractable Subclass of Allen’s Interval Algebra” Journal of the ACM (JACM), Vol.42 (1), pages: 43–66, 1995.

[44] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean "SWRL: A Semantic Web Rule Language Combining OWL and RuleML." W3C Member submission, 2004. http://www.w3.org/Submission/SWRL/

[45] M. Stocker and E. Sirin. Pelletspatial: A Hybrid RCC-8 and RDF/OWL Reasoning and Query Engine. In CEURWorkshop Proceedings, 529. Citeseer, 2009.

[46] P. van Beek, and R. Cohen "Exact and approximate reasoning about temporal relations." Computational intelligence, Vol 6(3), pp.132–147, 1990.

[47] S. Batsakis, "SOWL: A Framework for Handling Spatio-Temporal Information in OWL". Ph.D. dissertation, Technical Univ. of Crete (TUC), Dept. of Electronics and Comp. Engineering, Chania, Crete, Greece, Dec. 2011. http://www.intelligence.tuc.gr/publications. php?pub author=All&pub type=10&pub subject=All

[48] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider. The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, 2nd Edition, 2007.

[49] B. Bennet. Knowledge Representation and Reasoning: Compositional Reasoning, 2007. Lecture notes, School of Computing, University of Leeds.

[50] Z. Gantner, M. Westphal, and S. W • ol. GQR - A Fast Reasoner for Binary Qualitative Constraint Calculi. In Proc. of AAAI'08 Workshop on Spatial and Temporal Reasoning, Chicago, 2008.

[51] R. Gr • utter and B. Bauer-Messmer. Combining OWL with RCC for Spatioterminological Reasoning on Environmental Data. In Proc. of 3rd OWL: Experiences and Directions (OWLED2007), 2007.

[52] R. Gr • utter and B. Bauer-Messmer. Towards Spatial Reasoning in the Semantic Web: A Hybrid Knowledge Representation System Architecture. In Lecture Notesin Geoinformation and Cartography, Springer, Berlin Heidelberg, 2007.

[53] V. Haarslev, C. Lutz, and R. M • oller. Foundations of Spatioterminological Reasoning with Description Logics. In Proc. of 6th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR98), pages 112{123, 1998.

[54] Y. Katz and B. Cuenca Grau. Representing Qualitative Spatial Information in OWL-DL. In Proc. of 1st OWL: Experiences and Directions Workshop (OWLED2005), Galway, Ireland, 2005.

[55] A. Mackworth. Consistency in Networks of Relations. Articial Intelligence,8(1):99{118, 1977.

[56] W. Nutt. On the translation of qualitative spatial reasoning problems into modal logics. In Proc. of the 23rd Annual German Conf. on AI (KI99), 1999.

[57] B. Parsia and E. Sirin. Pellet: An OWL DL Reasoner. In Proc. of Int. Workshop on Description Logics (DL2004), 2004.

[58] D. Randell, Z. Cui, and A. Cohn. A Spatial Logic based on Regions and Connections. In Proc. of 3rd Principles of Knowledge Representation and Reasoning (KR92), pages 165{176, San Mateo, CA, USA, 1992.

[69] J. Renz and B. Nebel. Ecient Methods for Qualitative Spatial Reasoning. In Proc. of the 13th European Conference on Articial Intelligence (ECAI98), 1998.

[60] D. Tsarkov and I. Horrocks. FaCT++ description logic reasoner: System description. In Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR06), 2006.

[61] Carmichael, D.J., Kay, J., Kummerfeld, B.: Consistentmodelling of users, devices and sensors in a ubiquitous computing environment. User Modeling and User-Adapted Interaction 15 (2005)197–234

[62] Ranganathan, A., McGrath, R.E., Campbell, R.H.,Mickunas, D.: Ontologies in a pervasive computing environment.In: Workshop on Ontologies in Distributed Systems, IJCAI 2003. (2003)

[63] Chen, H., Perich, F., Finin, T., Joshi, A.: SOUPA: Standard ontology for ubiquitous and pervasive applications.In: MOBIQUITOUS 2004. 258–267

[64] Wang, X.H., Zhang, D.Q., Gu, T., Pung, H.K.: Ontology based context modeling and reasoning using owl.In: Proceedings of PERCOMW 2004. (2004)

[65] Heckmann, D.: Ubiquitous User Modeling. PhD thesis,Department of Computer Science, Saarland University,Germany (2005)

[66] de Freitas Bulc˜ao Neto, R., da Grac¸a Campos Pimentel,M.: Toward a domain-independent semantic model for context-aware computing. In: Third Latin American Web Congress, LA-WEB 2005. (2005)

[67] Agostini, A., Bettini, C., Riboni, D.: Experience report:Ontological reasoning for context-aware internet services. In: Proceedings of PERCOMW 2006. (2006)8–12

[68] Noy, N.F., McGuinness, and D.L.: Ontology development 101: A guide to creating your first ontology. Technical Report KSL-01-05, Stanford KSL (2000)

[69] Heflin, J., Hendler, J.A.: Dynamic ontologies on the web. In: Proceedings of AAAI 2000, AAAI Press /the MIT Press (2000) 443–449

[70] Horrocks, I.: DAML+OIL: A reason-able web ontology language. In: EDBT 2002. (2002)

[71] Y. Katz, and B. Grau. "Representing Qualitative Spatial Information in OWL-DL". In Proc. of Int.Workshop: OWL Experiences and Directions, Galway, Ireland, 2005.

[72] J. Renz and B. Nebel. Efficient Methods for Qualitative Spatial Reasoning. In Proc. of the 13th European Conference on Artificial Intelligence (ECAI98), 1998.

[73] B. Bennet. Knowledge Representation and Reasoning: Compositional Reasoning, 2007. Lecture notes, School of Computing, University of Leeds.

[74] J. Renz, and B. Nebel. Qualitative Spatial Reasoning using Constraint Calculi. In Handbook of Spatial Logics, Springer, Netherlands, pp. 161-215, 2007.

[75] B. Nebel, and H.J. Burckert. "Reasoning about Temporal Relations: A Maximal Tractable Subclass of Allen's Interval Algebra" Journal of the ACM (JACM), Vol.42 (1), pages: 4366, 1995.

[76] Boris Motik, Peter F. Patel-Schneider, and Bijan Parsia. OWL 2 web ontology language document overview. URL, 2009.http://www.w3.org/TR/owl2-overview/.

[77] J. F. Allen. \Maintaining Knowledge about Temporal Intervals". Communications of the ACM. 26:832-843, 1983.

[78] A. Cohn, B. Bennett, J. Goodayand and N. Gotts \Qualitative Spatial Representation and Reasoning with the Region Connection Calculus".GeoInformatica, 1(3), pp. 275-316, 1997.

[79] J. Tappolet, and A. Bernstein \Applied Temporal RDF: efficient Temporal Querying of RDF Data with SPARQL." In: Proceedings of the European Semantic Web Conference, LNCS 5554, 308-322, 2009.

[80] H. Chen, F. Perich, T. Finin, and A. Joshi \SOUPA:Standard Ontology for Ubiquitous and Pervasive Applications" International Conference on Mobile and Ubiquitous Systems: Networking and Services, pp.258 {267, 2004.

[81] A. Sotnykova, C. Vangenot, N. Cullot, N. Bennacer and M. Aufaure \Semantic Mappings in Description Logics for Spatio-temporal Database Schema Integration" Journal on Data Semantics III, pp.143-167, 2005.

[82] M. Stocker, and E. Sirin \Pellet Spatial: A Hybrid RCC-8 and RDF/OWL Reasoning and Query Engine"OWLED 2009.

[83]Papadakis, Nikos, & Christodoulou, Yannis (2010). A tool for addressing the ramification problem in spatial databases: A solution implemented in SQL. Expert Systems with Applications, 37(2), 1374–1390.

[84]Nikos Papadakis and Plexousakis Dimitris. Action with Duration and Constraints:The Ramification problem in Temporal Databases. 14th IEEE ICTAI, 2002, Washington D.C.

[85]Papadakis, Nikos, Antoniou, Grigoris, Plexousakis, Dimitris. (2007). the ramification problem in temporal databases: Concurrent execution with conflicting constraints. In 19th IEEE ICTAI, Greece (pp 274–278).

[86]Cohn, A. G. & Varzi, A. C. (2003). Mereotopological connection. Journal of Philosophical Logic, 32, 357–390.

[87]Couclelis, H. (1992). People manipulate objects (but cultivate fields): Beyond the raster-vector debate in GIS. In A. U. Frank, I. Campari, & U. Formentini, (Eds.), Theories and methods of spatio-temporal reasoning in geographic space (pp. 65–77). Berlin: Springer-Verlag.

[88]Egenhofer, M. & Mark, D. (1995) Naive geography. In A. Frank & W. Kuhn(Eds.). Spatial information theory: A theoretical basis for GIS, COSIT' 95,Vol. 988 Lecture Notes in Computer Science, (pp. 1–15). Berlin: Springer.

[89]Frank, A. U. (2003). Ontology for spatio-temporal databases. In M. Koubarakis,et al. (Eds.), Spatiotemporal databases: The chorochronos approach (Lecture Notes in Computer Science 2520), Berlin, Springer-Verlag

[90]Frank, A. U., Raper, J., & Cheylan, J-P. (Eds.) (2001). Life and motion of socioeconomic units (GISDATA Series 8), London: Taylor & Francis.

[91] C. Freksa, M. Knauff, B. Krieg-Br¨uckner, B. Nebel, and T. Barkowsky, editors. Spatial Cognition IV. Reasoning, Action, Interaction: International Conference Spatial Cognition 2004, Lecture Notes in Artificial Intelligence 3343. Springer, Berlin, Heidelberg, 2005.

[92] R. Moratz, F. Dylla, and L. Frommberger. A relative orientation algebra with adjustable

granularity. In Proceedings of the Workshop on Agents in Real-Time and Dynamic Environments (IJCAI 05), 2005.

[93] S. W¨olfl and T. Mossakowski. CASL specifications of qualitative calculi. In Spatial InformationTheory: Cognitive and Computational Foundations, Proceedings of COSIT'05, 2005.

[94]Vilain, M. B., and Kautz, H. A. 1986. Constraint propagation algorithms for temporal reasoning. In Proceedings of the 5th National Conference on Artificial Intelligence. Morgan Kaufmann.377–382.

[95]Wallgr¨un, J. O.; Frommberger, L.; Wolter, D.; Dylla, F.; and Freksa, C. 2006. Qualitative spatial representation and reasoning in the SparQ-toolbox. In Barkowsky, T.; Knauff, M.; Ligozat, G.; and Montello, D. R., eds., Spatial Cognition V, LNCS 4387, 39–58. Springer.

[96] Ligozat, G., and Renz, J. 2004. What is a qualitative calculus?A general framework. In PRICAI 2004: Trends in Artificial Intelligence, 8th Pacific Rim International Conference on Artificial Intelligence, Proceedings, LNCS 3157, 53–64. Springer.

[97]CNTRO: A Semantic Web Ontology for Temporal Relation Inferencing in Clinical Narratives.

(PMID:21347086)Tao C, Wei WQ, Solbrig HR, Savova G, Chute CG.AMIA Annu Symp Proc [2010]

[98]Du, H., Alechina, N., Jackson, M., & Hart, G. (2013). Matching formal and informal geospatial ontologies. In Geographic information science at the heart of europe (pp. 155-171). Springer, Cham.