The KBMS Project and Beyond

Vinay K. Chaudhri[1], Igor Jurisica[2], Manolis Koubarakis[3], Dimitris Plexousakis[4], and Thodoros Topaloglou[5]

[1] SRI International, Menlo Park, CA, USA
[2] Ontario Cancer Institute, University Health Network, Toronto, Canada,
[3] National and Kapodistrian University of Athens, Athens, Greece
[4] University of Crete and FORTH-ICS, Heraklion, Crete, Greece
[5] McGill University and Genome Quebec Innovation Center, Montreal, Canada

Abstract. The Knowledge Base Management Systems (KBMS) Project at the University of Toronto (1985-1995) was inspired by a need for advanced knowledge representation applications that require knowledge bases containing hundreds of thousands or even millions of knowledge units. The knowledge representation language Telos provided a framework for the project. The key results included conceptual modeling innovations in the use of semantic abstractions, representations of time and space, and implementation techniques for storage management, query processing, rule management, and concurrency control. In this paper, we review the key ideas introduced in the KBMS project, and connect them to some of the work since the conclusion of the project that is either closely related to or directly inspired by it.

1 Introduction

In the early nineties, the emergence of advanced applications such as Computer-Aided Design (CAD), software engineering, real-time control systems, and grand challenge projects such as the DARPA knowledge sharing project [1] and the Human Genome project [2] required technologies for creating and managing data representations with rich structure and ability for inference. The goal of the Knowledge Base Management Systems (KBMS) project, led by Prof. John Mylopoulos, was to develop the technology for construction and efficient access of large and shared knowledge bases (KBs). At that time, the state of the art in KB implementations was to use expert system shells or programming languages such as Lisp or Prolog. In the KBMS project, we investigated the use of database technology as a source for techniques for advancing the state of the art in constructing large knowledge base systems [3].

The knowledge representation language Telos provided the focal point of research in the KBMS project, and a framework for five Ph.D. [4–8] and three Masters [9, 10] theses. We begin this paper with an overview of Telos and the key technical results of the project. We then discuss the current state of development of knowledge base systems most closely related to the problems investigated in the KBMS project.

2 The Knowledge Representation Language Telos

Since a detailed description of Telos is available in published papers [11], we will highlight here only its salient features.

Propositions: A proposition is the fundamental unit in a KB. A proposition is a triple, and can represent an individual or an attribute.

Structural Knowledge: The propositions in a KB are organized along three dimensions: aggregation, classification, and instantiation. The aggregation represents structured objects, the classification represents the class-subclass relationship, and the instantiation represents the class-instance relationship.

Classes can be instances of other classes, thus allowing meta-classes. Meta-classes are a mechanism for extending the representation model. Similarly, attributes can be instances of classes, called *attribute classes*, which provide a mechanism to impose constraints or specific semantics to attributes.

Temporal Knowledge: Every Telos proposition has an associated history time and a belief time. The history time of a proposition represents the lifetime of a proposition in the application domain (i.e., the lifetime of an entity or a relationship). A proposition's belief time, on the other hand, refers to the time when the proposition is believed by the KB, i.e., the interval between the moment the proposition is added to the KB and the time when its belief is terminated. In Telos, time is modeled using intervals that can be related using the relations in Allen's Interval Algebra [12]. The combination of constraint-based temporal representations and nontemporal ones was fully explored in the Ph.D. thesis of Koubarakis [4].

Assertional Knowledge: Telos provides an assertion language for the expression of deductive rules and integrity constraints. The assertion language is a first-order language with equality. Telos supports both static constraints (that apply to all states of a KB) and dynamic constraints (that apply to those temporal states that satisfy specific temporal predicates or to the transition between temporal states). Deductive rules are also explicitly associated with history and belief time intervals.

Telos supports primitive KB operations such as *Tell*, *Untell*, *Retell*, and *Ask* that can be used to query and update the KB. A possible-worlds semantics for Telos was defined in the Master's thesis of Plexousakis [10]. The semantics of Telos include an ontology of objects based on the property of existence, and proofs for the soundness, consistency, and completeness of a Telos KB.

Spatial Knowledge: The representation for the spatial knowledge in Telos was defined in the Ph.D. thesis of Topaloglou [7]. The spatial representation in Telos was accomplished through a library of meta-classes and meta-attributes that capture the semantics of spatial features of physical objects. The objects with a spatial extension can be *placed in* spaces, called *maps*, at variable scales and related to other spatial objects or constants through qualitative or quantitative relationships.

2.1 Telos Implementations

There have been four implementations of Telos. The first implementation was done as part of two Master's theses [9], carried out at the University of Toronto and ICS-FORTH, and covered all the features of Telos including reasoning with incomplete temporal information. The implementation language was Prolog with the temporal reasoning module implemented in C for efficiency. The temporal reasoning module implemented constraint satisfaction for a subset of Allen's Interval Algebra so that consistency checking remains polynomial. Nevertheless, the data complexity of query answering for the query language used is at least NP-hard [13].

The second implementation of Telos was done by an ICS-FORTH team led by Martin Doerr. The implementation was done in C++ and covered only the structural knowledge, and provided no support for assertional or temporal knowledge. This Telos prototype was used in the implementation of the Software Information Base [14].

The third implementation of Telos was based on the dialect O-Telos defined in the Ph.D. thesis of Manfred Jeusfeld at the University of Passau, and implemented in the ConceptBase system [15]. O-Telos omitted the history time component of Telos and its facilities for reasoning with incomplete information, and implemented only the structural and assertional knowledge features of Telos. Since 1995, ConceptBase has been continuously developed and it is freely available.[6] ConceptBase is currently applied at more than 500 sites worldwide for research as well as teaching and is the most complete implementation of Telos.

The fourth implementation, called Common Knowledge Base (CKB), was done by Bryan Kramer and Martin Stanley at the University of Toronto in the context of the APACS project. This implementation was done in C++ and a commercial object-oriented database management system, Versant, and covered only the structural knowledge [16].

2.2 Research on Implementation Techniques

The implementations of Telos mentioned above were done using the technology available at that time, did not support all the features of Telos, and, with the exception of ConceptBase, were not designed with scalability in mind. A significant effort in the KBMS project was devoted to addressing these limitations. Specifically, we investigated database techniques for implementing Telos, and subjected those techniques to rigorous performance evaluation [3]. We summarize here the techniques that were developed during the KBMS project.

Storage Management: The research goal of this task was to efficiently store on disk a KB that was too large to fit in the main memory. We developed a scheme called *Controlled Decomposition Model*, that could map the structural knowledge of a KB to a set of relations in a way that the information that was

[6] `http://www-i5.informatik.rwth-aachen.de/CBdoc/`

likely to be accessed together was stored together for efficient access, while the rest was split across multiple relations to support efficient storage and updates [17]. We extended the database techniques based on join indices to deal with the temporal dimension of a KB.

Query Processing: The research goal of this task was to develop semantic query optimization techniques for processing Telos queries [18]. We developed query simplification techniques for temporal knowledge, syntactic simplification techniques that apply knowledge of the class hierarchy, and techniques for generating query evaluation plans. We also developed a cost model for optimizing path queries in knowledge bases with structural and temporal knowledge. [19].

Concurrency Control: The goal of this task was to develop techniques to allow multi-user updates to a KB and was done as part of the Ph.D. thesis of Chaudhri [5]. The key result was a new locking protocol, called the *Dynamic Directed Graph policy*, that improved performance over the traditional two phase locking protocol by taking advantage of the assertional component of Telos.

Integrity Constraints: The research goal of this task was to develop an algorithm for efficient checking of integrity constraints for a Telos KB. This research was done as a part of the Ph.D. thesis of Plexousakis [6]. We developed techniques for compiling, simplifying and checking the violation of integrity constraints when there are updates to a KB. The technique took into account the temporal knowledge of Telos as well as potential interactions between rules and integrity constraints.

Case-based Reasoning: The research goal of this task was to develop a methodology for building large and complex case-based reasoning systems, and it was done as part of the Ph.D. thesis of Jurisica [8]. To achieve flexibility without reducing performance, we adapted an incremental view maintenance algorithm from database management systems [20] into a reasoning system called $\mathcal{TA}$**3**, and successfully applied it to several biomedical domains [21–23, 22, 24, 25]. The key components of $\mathcal{TA}$**3** included case representation in Telos, incremental query relaxation, modified k-nearest neighbor algorithm, and anytime retrieval algorithm.

3 Evolution of Knowledge Systems Since the KBMS Project

It is helpful to think of problems in the general area of constructing large knowledge systems in three broad classes: Content Modeling, Implemented Systems and Measurement and Evaluation. We now consider each of these subtopics in detail concentrating on research that took place after the KBMS project, and is either closely related to it or directly inspired by it. We notice that much of this work is at the core of research areas that have received much attention recently such as scientific data management or the Semantic Web.

3.1 Content Modeling

There are two aspects of content modeling: the knowledge representation language itself, and the KB content. In Telos, the temporal and spatial representations were considered part of the language itself. This is not the case in systems such as Cyc [26] in which the temporal knowledge and spatial knowledge are considered as subtheories in a KB - known as an upper ontology, and independent of the representation language. Here, we review the research on KR language features that were the subject of inquiry in Telos.

Structural Knowledge: In the mid-nineties, several KR research groups investigated languages such as CLASSIC [27], LOOM [28], and Ontolingua [29] that are in the tradition of KL-ONE [30]. In an effort to synthesize among the best features of various languages, Peter Karp at SRI International did a survey of frame-based knowledge representation languages prevalent at that time that led to the development of the Generic Frame Protocol [31] followed by the Open Knowledge Base Connectivity Model [32].

The World-Wide Web (WWW) gave rise to a whole new group of languages that are based on semantic abstractions and were designed as a foundation for the Semantic Web [33]. Building on the WWW technologies (for example, Uniform Resource Indicators) and the past research on semantic networks and frame-based knowledge representation languages, Resource Description Framework (RDF) was created [34] and became a standard of the World Wide Web Consortium (W3C) in 2004.[7] With a similar emphasis, the DARPA Agent Markup Language program[8] (DAML) focused on standardizing expressive knowledge representation languages for the WWW. At the same time, another group of researchers, mainly based in Europe, developed a competitor ontology language called the Ontology Interchange Language (OIL) [35]. The interaction among DAML and OIL led to the language DAML+OIL, which eventually became the Ontology Web Language or OWL that became a W3C recommendation in 2004.[9]

Looking back at the gradual development of the languages mentioned above, it is easy to discern the influence of Telos and the KL-ONE tradition in RDF and the languages building on them. RDF triples bear a close resemblance to Telos propositions. The abstraction mechanisms adopted in these languages had already been incorporated as modeling primitives in the representation framework of Telos since the late 1980s.

[7] `http://www.w3.org/RDF/`

[8] `http://www.daml.org`

[9] `http://www.w3.org/2004/OWL/`

Assertional Knowledge: Early efforts to combine an assertion language with structural knowledge predate Telos and the KBMS project (see, e.g., KRYPTON [36]); such languages are now part of several implemented systems [29, 26]. In parallel to work on Telos, there have also been various other proposals of languages combining an assertional and structural knowledge, many of them coming under the label "deductive and object-oriented" languages. Perhaps the most theoretically elegant language in this family is F-Logic [37]. F-Logic, like all other deductive and object-oriented languages, originated from the database and logic programming traditions, and accounts in a clean and declarative fashion for most of the structural aspects of object-oriented and frame-based languages.

F-Logic, however, does not allow for temporal or spatial knowledge in its assertion language in the way that Telos supports. An additional distinguishing feature of the assertional component of Telos, not present in related languages, is the coupling of the integrity constraints and deductive rules with the structural knowledge. In Telos, rules and constraints are treated uniformly and can be attached to classes at any level of the class hierarchy.

More recently, there have been several proposals for rule languages for the Semantic Web. For example, Semantic Web Rule Language (SWRL) deals with issues such as rule definition based on OWL.[10] Semantic Inferencing for Large Knowledge or SILK is an effort to define an expressive rule language that addresses issues such as reasoning with processes and defaults.[11]. The Rule Markup Language (RuleML) initiative aims at defining a common rule interchange and markup language based on XML and RDF.[12] The problems studied in the context of the specification, semantics and use of the assertion language of Telos are thus recast and expanded in the context of reasoning on the Semantic Web.

Temporal Knowledge: The rich temporal knowledge representation concepts of Telos (history time, belief time, interval-based incompleteness of historical knowledge) did not find any followers in the deductive and object-oriented (or frame-based) families of languages. There has been interesting related work in other areas.

The distinction between history and belief time had already been made explicitly in the area of temporal relational databases [38, 39] before Telos was put forward (the corresponding terms used were *valid time* and *transaction time*) and Telos had benefited from this work. The area of temporal databases saw an explosion of activity culminating in the specification of the query language TSQL2 [40], a temporal extension of the SQL-92 standard. In the deductive database and logic programming community, the most comprehensive proposal to introduce valid and transaction time using event calculus is by Sripada [41].

Gutierrez and colleagues [42–44] have proposed to extend RDF triples of the form (s, p, o) with an additional temporal label t that is a natural number. The resulting quad (s, p, o, t) is called a *temporal RDF triple* and denotes the fact

[10] `http://www.w3.org/Submission/SWRL/`

[11] `http://silk.projects.semwebcentral.org/`

[12] `http://www.ruleml.org/`

that the triple (s, p, o) is valid at time t. Based on this definition, [42–44] define a *temporal RDF graph* as a set of temporal RDF triples, and study problems of semantics and computational complexity of query answering. This representation directly maps to the representation of propositions in Telos.

The temporal description logics introduce a *concrete domain* to model time together with appropriate definitions for concepts, roles and features [45]. Work here has concentrated mostly on issues of semantics and reasoning in these logics with little emphasis on implementations.

Starting with Cyc,[13] there has also been work on ontologies of time [14]. The time ontology in Cyc is one of the most comprehensive representations of time available today and has been reused outside the context for which it was originally created [46].

Spatial Knowledge: Numerous spatial representations and calculi to support efficient reasoning with spatial knowledge are now available that were not available at the time of the KBMS project [47]. The most comprehensive representation of space in an implemented system is in the Cyc KB. The Cyc KB provides a first-order axiomatization of basic spatial representational primitives. It includes axiomatization of more than 65 spatial predicates, which include 15 different kinds of containment and 7 different kinds of covering. It also supports representations for shapes, boundaries, regions, and convex hulls [48].

The Semantic Web language RDF has recently been extended to represent spatial knowledge. In the system SPAUK [49], geometric attributes of a resource (e.g., location of a gas station) are represented in RDF by introducing a blank node for the geometry, specifying the geometry using the Geography Markup Language,[15] and associating the blank node with the resource using Geography Encoded Objects for RSS feeds.[16] Queries in SPAUK are expressed in the SPARQL query language utilizing geometric vocabularies and ontologies [47]. In a similar spirit, Perry defines an extension of SPARQL, called SPARQL-ST, that allows one to query spatial and nonspatial data with a time dimension [50].

3.2 Implemented Systems

We review here the implemented systems to which the graduates of the KBMS project have contributed. These systems are closely related to the spirit of the KBMS project.

The PERK System: The PERK (or Persistent Knowledge) system at SRI International was implemented to enable collaborative construction of KBs [51]. It addressed the issues of storage management, concurrency control, graphical

[13] `http://www.opencyc.org/`

[14] `http://www.w3.org/TR/2006/WD-owl-time-20060927/`

[15] `http://www.opengeospatial.org/standards/gml`

[16] `http://georss.org/`

editing, and application programming interfaces. The PERK system supports only structural knowledge, and has no support for assertional, temporal or spatial knowledge. We consider here the topics of storage management and concurrency control since they are most closely related to the implementation techniques of Telos KBMS.

The storage system in PERK is aimed at allowing incremental loading and saving of a large KB into the main memory. It achieves this by submerging the commercial Oracle database management system in an existing Lisp-based frame-based representation system. Due to impedance mismatch between the in-memory representation of Lisp objects and a relational database, and to facilitate evolution of the schema, PERK represents the content of a frame using a compressed string representation. Since a string representation is not in the first normal form, it is unable to support query processing directly by the database. All the query processing is then done in memory using the frame system. To facilitate fast retrieval, PERK adds indices for frequently accessed slots. The spirit of this design is very close to the controlled decomposition model used in the Telos KBMS.

The concurrency control scheme in PERK is based on an optimistic concurrency control approach: the users are allowed to make divergent updates to the KB; and once they are satisfied with their changes, they attempt to commit their updates; any conflicting updates are detected and they must resolve the conflicts. This model seemed preferable to locking-based approach advocated in the Telos KBMS because it allowed greater collaboration among the contributors. The conflict detection is done based on a change log that is organized at the granularity of knowledge editing operations and not low-level storage operations in a database. The PERK system is in extensive use as the back end of SRI's EcoCyc system [52].

The OPM Tools Suite: The Object Protocol Model (OPM) project [53] at the Lawrence Berkeley National Laboratory utilizes a semantic data model to describe the semantics of the entities in biological databases and build advanced query mechanisms for biological data. The OPM project shares significant similarities with the Telos knowledge management project.

OPM is an object-based data model that is very similar to Telos and is used to describe a database at a conceptual level in terms of classes, attributes, and relationships. By representing the semantics of the data, it enables scientific users to create, query, and integrate databases without being bogged down by implementation and system-level details. The gap between the user view and the implementation view of a database is bridged by data management tools that are driven by rich meta-data represented in OPM. The OPM suite includes tools for creating and querying databases, adding a query interface to an existing database, querying multiple databases, and extending databases with application-specific data types and methods. Forward development of an OPM database results in the creation of a relational database and object-relational (O-R) mapping meta-data that are used to reformulate object queries to relational ones, and reconstruct objects from the results of a relational query plan. The O-R mapping of OPM shares striking similarities to the controlled decomposition model, although they have been developed independently.

The mechanism for model extensibility in OPM, needed to support methods and application-specific types, such as DNA sequences and images, was realized following an approach similar to the Telos approach, i.e., adding a new meta-class at the language level, and complemented by a robust integration framework and implementation infrastructure [54].

The RDF Suite: Due to the expansion of the WWW, there is a significant need to enable querying over heterogeneous information sources. One effort to meet this need is the ICS-FORTH RDFSuite [55], which provides services for loading, parsing, schema-aware storage, querying, viewing, and updating of RDF/S resource descriptions and schemas. The design of RDFSuite was influenced by the design decisions of the Telos KBMS, in particular with respect to storage management and querying. RDFSuite comprises (a) the Validating RDF Parser (VRP): a parser supporting semantic validation of both RDF/S resource descriptions and schemas, (b) the RDF Schema Specific DataBase (RSSDB): a store exploiting a variety of Object-Relational (SQL3) representations to manage RDF/S resource descriptions and schemas, (c) the RDF Query Language (RQL): a declarative language for uniformly querying RDF/S resource descriptions and schemata, (d) the RDF View Language (RVL): the first declarative language for creating virtual RDF/S resource descriptions and schemas, and (e) the RQL Graphical Query Generator (GRQL): a user interface generating minimal declarative queries by taking into account the browsing actions in an RDF/S schema during a user navigation session.

The $\mathcal{TA}$3 System: The $\mathcal{TA}$**3** system was developed as part of the Ph.D. thesis of Jurisica [8] and has undergone significant evolution. The $\mathcal{TA}$**3** system has been expanded by using incremental query relaxation and anytime retrieval algorithm [20], reimplementing the system in Java, and using the IBM DB2 relational database for persistent storage [23]. Further expansion covered support to handle images [22, 56, 24, 57], improved performance with data mining [25, 58], and improved accuracy with an ensemble of classifiers [59].

The $\mathcal{TA}$**3** system currently handles the scale and the size of data for which the KBMS project was envisioned. As a specific example, it is being used to store and analyze protein crystallization experiments [60, 23, 58]. In its current use, there are 12,000 protein crystallization experiments, each of which has 9,216 attributes, and the data is derived from 110,592,000 images [61, 62]. The repository grows at a rate of more than 200 experiments each month. Before $\mathcal{TA}$**3** can suggest crystallization optimization strategies for a novel protein, we have to compute 12,375 image features and automatically classify images into 10 possible categories [56, 63, 61, 62].

The key to scaling the techniques that were developed in the KBMS project for this new application has been the use of a scalable computational infrastructure. Although the $\mathcal{TA}\mathbf{3}$ system can run on a regular Unix or Windows server, the image processing runs on the World Community Grid[17] and the post-processing is done on a 1,344-core Linux cluster.

Another large-scale application of the $\mathcal{TA}\mathbf{3}$ system involves estimating a job runtime [64]. The application domain considered scheduling Functional Regression Tests (FRT) for the IBM DB2 Universal Database product Version 8.2 (DB2 UDB) [65]. A job runtime can be affected by a large number of both job and machine characteristics. Scalable performance is critical, as there are more than 50,000 jobs to test for each version of DB2 UDB in a grid, which comprises about 300 machines with different configurations. A case is represented as a record that includes job information, machine information, and runtime information. Case retrieval and adaptation uses the priorities of job and both static and dynamic machine characteristics. Our experimental results show that for more than 90% of jobs, the estimation error is 45% or less, and the average estimation error is at most 22%. Applying the system to FRT, we achieved average performance improvements of 20% to 50%. In the worst case, we still achieved a 13% to 36% performance improvement [65].

3.3 Measurement and Evaluation

Performance evaluation of implementation techniques was a central methodology in the KBMS project. We discuss here innovations in measurement and evaluation of knowledge-based systems.

Metrics on KB Size: In the world of databases, the size of the data measured in records or bytes is a very good indicator of the scale of the problem. But, a similar measure such as "millions of knowledge units" is not always meaningful in the context of KBs. For example, a KB containing millions of simple facts may have less knowledge content than a KB with a small number of axioms. As an approximate measure, one can use the number of axioms in a KB as one measure of competence. competence.

One possible approach to measuring the size was investigated in DARPA's High Performance KBs (HPKB) project in which the measure of axiom counts was refined to include axiom categories [66]:

1. *Constants* are any names in the KB, whether an individual, class, relation, function, or a KB module.
2. *Structural statements* are ground statements about the semantic abstractions in a KB, for example, subclass-of, instance-of, domain, and range assertions.
3. *Ground facts* are any statement without a variable.
4. *Implications* include any nonground statement that has an *implies* (a ground statement that contains an *implication* is counted as a ground statement).
5. *Non ground, non implications* are statements that contain variables but not an implication.

[17] http://www.worldcommunitygrid.org

While axiom categories are an improvement over measuring the size in terms of *knowledge units*, they are still imperfect; a larger number of axioms in a category does not alway imply a greater amount of knowledge. As a methodological improvement, Vulcan's Project Halo measures the size of a KB in terms of the questions it is able to answer on a standardized college-level test [67]. While the approach of using a standardized test is a significant improvement, we believe new ways of measuring the KB content and quality are needed that are applicable to more general classes of systems.

Knowledge Reuse Metrics: A central claim in building the content for large KBs is that as we add more content, things get easier, as the content added later builds on what already exists. This is a claim that makes intuitive sense because when new knowledge is to be added to a KB, relevant terms may already exist and some knowledge may be available through inheritance. One innovative way to test this claim is to study how knowledge is reused in a large KB [68] that was tried in DARPA's HPKB project.

The knowledge reuse metric can be defined as follows. Suppose one wishes to add a new piece of knowledge to a KB. Every item i one wishes to add to the KB contains $n(i)$ terms, $k(i)$ of which are already in the KB, and support is $s(i) = k(i)/n(i)$. Adding new terms to a KB changes the size of the KB, and the support offered by the KB for future axioms might be higher because new terms were added. Thus, support is indexed by versions of the KB: $s(i, j) = k(i, j)/n(i)$ is the support provided by version j of the KB for concept i.

Although the idea of knowledge sharing has been in the literature for many years [1], the reuse metric was one of the first attempts to empirically study the claim. The results in using this metric suggested that the answer depends on the kind of prior knowledge, who is using it, and what it is used for. There is still lot of room for further understanding of the KB construction process: How long will a knowledge engineer hunt for a relevant term or axiom in a prior ontology? How rapidly do KBs diverge from available ontologies if knowledge engineers do not find the terms they need in the ontologies? By what process does a knowledge engineer reuse not an individual term but a larger fragment of an ontology, including axioms? How does a very general ontology inform the design of KBs, and what factors affect whether knowledge engineers take advantage of the ontology? Why do prior ontologies apparently provide less support for encoding axioms than for encoding test questions?

Benchmarks for Knowledge Base Systems: At the time of the KBMS project, hardly any data sets were available for testing tools and algorithms that we were investigating. In recent years, some progress has been made on this front, and we review two such efforts.

The Lehigh University Benchmark (LUBM) is a benchmark for large OWL KBs [69]. The LUBM features an ontology for the university domain, synthetic OWL data scalable to an arbitrary size, fourteen extensional queries representing a variety of properties, and several performance metrics. The LUBM can be used to evaluate systems with different reasoning capabilities and storage mechanisms. It has been used for memory-based systems as well as systems with persistent storage.

The OpenRuleBench[18] is a suite of benchmarks for analyzing the performance and scalability of different rule engines. It has been tested on five different technologies: Prolog, deductive databases, production rules, triple engines, and general KBs. It examines how different systems scale for a number of common problem sets that involve reasoning such as recursion and negation. It also examines, by interpolation, how these technologies or their successors might perform on the WWW scale.

There is no existing benchmark to characterize the performance of temporal and spatial reasoning systems, but there is recent interest in the research community to define such a benchmark [70].

We believe that further work on benchmarking and evaluating the implementations of KB systems is essential to continued science and engineering of KBMS implementations.

4 Summary and Conclusions

The Telos KBMS project attempted an extensive experimentation of applying database management techniques to implementation of KBs. The project fulfilled its goals by at least two measures. First, it generated five Ph.D. theses, and three Master's theses, and trained highly qualified personnel who later contributed to a number of visible projects. Second, it emphasized key themes in knowledge base management that have sustained the test of time. The key findings of the project were the following:

1. A rich representation language that combines semantic abstractions of classification, instantiation, aggregation with deductive rules, integrity constraints with temporal and spatial representation defined as part of the language is essential for constructing large knowledge bases.
2. To achieve efficient implementation of large knowledge bases, we will need to rely on implementation techniques from database management systems such as storage management, query processing, concurrency control, rule management, and view maintenence.
3. Performance evaluation is core methodology for testing the implementations for large KB systems.

The semantic abstractions considered in the Telos knowledge representation language are at the core of most modern representation systems. There are languages and systems that specialize in various advanced features such as object-oriented and deductive representations (for example, F-Logic), rule and constraint management (for example, relational and deductive databases) or temporal reasoning (for example, Allen's calculus), but there is no unified

[18] `http://rulebench.projects.semwebcentral.org/`.

representation and reasoning theory that combines all the features that were considered in Telos into one language.

The adoption of systems such as PERK, OPM, and $\mathcal{TA}\mathbf{3}$ suggests that the bio-medical research community has been the earliest adopter of expressive representation languages and the database techniques of the sort investigated in the Telos KBMS project. The recent work on the Semantic Web is leveraging the same ideas and technology - the theory and practice behind the RDF framework (e.g., as realized in the RDFSuite) is just one such instance.

Performance evaluation of knowledge base systems has now become a central methodology, especially in government funded projects in the United States as exemplified by the HPKB project. The research community has started to define benchmarks for characterizing the performance for isolated systems features (for example, LUBM and OpenRuleBench), and there is growing need and acceptance for benchmarking and performance evaluation of knowledge base systems.

Reflecting on our work on the KBMS project, we can say that due to the WWW and the proliferation of scientific data, the knowledge management challenges today are even more real and significant than twenty years ago. As a result, the concepts, techniques, and methods investigated in the KBMS project will continue to have very high relevance for many years to come.

References

[1] Neches, R., Fikes, R., Finin, T.W., Gruber, T.R., Patil, R.S., Senator, T.E., Swartout, W.R.: Enabling technology for knowledge sharing. AI Magazine **12**(3) (1991) 36–56

[2] Frenkel, K.A.: The Human Genome Project and informatics. Communications of the ACM **34**(11) (1991) 41–51

[3] Mylopoulos, J., Chaudhri, V.K., Plexousakis, D., Shrufi, A., Topaloglou, T.: Building knowledge base management systems. The VLDB Journal **5**(4) (1996) 238–263

[4] Koubarakis, M.: Foundations of Temporal Constraint Databases. PhD thesis, Computer Science Division, Dept. of Electrical and Computer Engineering, National Technical University of Athens (February 1994)

[5] Chaudhri, V.K.: Transaction Synchronization in Knowledge Bases: Concepts, Realization and Quantitative Evaluation. PhD thesis, University of Toronto, Toronto (January 1995)

[6] Plexousakis, D.: Integrity Constraint and Rule Maintenence in Temporal Deductive Knowledge Bases. PhD thesis, University of Toronto, Toronto (1996)

[7] Topalogou, T.: On the Representation of Spatial Knowledge in Knowledge Bases. PhD thesis, University of Toronto, Toronto (1996)

[8] Jurisica, I.: TA3: Theory, Implementation, and Applications of Similarity-Based Retrieval for Case-Based Reasoning. PhD thesis, University of Toronto, Department of Computer Science, Toronto, Ontario (1998)

[9] Topaloglou, T., Koubarakis, M.: Implementation of Telos: Problems and Solutions. Technical Report KRR-TR-89-8, Dept. of Computer Science, University of Toronto (1989)

[10] Plexousakis, D.: An Ontology and a Possible-Worlds Semantics for Telos. Master's thesis, Dept. of Computer Science, University of Toronto (1990)

[11] Mylopoulos, J., Borgida, A., Jarke, M., Koubarakis, M.: Telos: A language for representing knowledge about information systems. ACM Transactions on Information Systems **8**(4) (October 1990) 325–362

[12] Allen, J.: Maintaining knowledge about temporal intervals. Communications of the ACM **26**(11) (November 1983) 832–843

[13] Koubarakis, M.: The complexity of query evaluation in indefinite temporal constraint databases. Theoretical Computer Science **171** (January 1997) 25–60 Special Issue on Uncertainty in Databases and Deductive Systems, Editor: L.V.S. Lakshmanan.

[14] Constantopoulos, P., Doerr, M., Vassiliou, Y.: Repositories for software reuse: The software information base. In: Information System Development Process. (1993) 285–307

[15] Jarke, M., Gallersdörfer, R., Jeusfeld, M.A., Staudt, M.: ConceptBase - A deductive object base for meta data management. Journal of Intelligent Information Systems **4**(2) (1995) 167–192

[16] Wang, H., Mylopoulos, J., Kusniruk, A., Kramer, B., Stanley, M.: KNOWBEL: New tools for expert system development. In Bourbakis, N.G., ed.: Developement of Knowledge-Based Shells. Advanced Series on Artificial Intelligence, World Scientific (1993)

[17] Topaloglou, T.: Storage management for knowledge bases. In: CIKM '93: Proceedings of the Second International Conference on Information and Knowledge Management, New York, NY, USA, ACM (1993) 95–104

[18] Topaloglou, T., Illarramendi, A., Sbattella, L.: Query optimization for KBMSs: Temporal, syntactic and semantic transformantions. In Golshani, F., ed.: Proceedings of the Eighth International Conference on Data Engineering, February 3-7, 1992, Tempe, Arizona, IEEE Computer Society (1992) 310–319

[19] Shrufi, A., Topaloglou, T.: Query processing for knowledge bases using join indices. In: Proceedings of the 4th International Conference on Information and Knowledge Management, Baltimore (November 1995)

[20] Jurisica, I., Glasgow, J., Mylopoulos, J.: Incremental iterative retrieval and browsing for efficient conversational CBR systems. International Journal of Applied Intelligence **12**(3) (2000) 251–268

[21] Jurisica, I., Mylopoulos, J., Glasgow, J., Shapiro, H., Casper, R.F.: Case-based reasoning in IVF: Prediction and knowledge mining. Artif Intell Med **12**(1) (1998) 1–24

[22] Jurisica, I., Rogers, P., Glasgow, J., Collins, R., Wolfley, J., Luft, J., DeTitta, G.: Improving objectivity and scalability in protein crystallization: Integrating image analysis with knowledge discovery. IEEE Intelligent Systems Journal, Special Issue on Intelligent Systems in Biology (Novem-

ber/December) (2001) 26–34

[23] Jurisica, I., Rogers, P., Glasgow, J., Fortier, S., Luft, J., Wolfley, J., Bianca, M., Weeks, D., DeTitta, G.T.: Intelligent decision support for protein crystal growth. IBM Systems Journal, Special Issue on Deep Computing for Life Sciences **40**(2) (2001) 394–409

[24] Jurisica, I., Glasgow, J.: Application of case-based reasoning in molecular biology. Artificial Intelligence Magazine, Special issue on Bioinformatics **25**(1) (2004) 85–95

[25] Arshadi, N., Jurisica, I.: Integrating case-based reasoning systems with data mining techniques for discovering and using disease biomarkers. IEEE Transactions on Knowledge and Data Engineering. Special Issue on Mining Biological Data **17**(8) (2005) 1127–1137

[26] Lenat, D.B., Guha, R.V., Pittman, K., Pratt, D., Shepherd, M.: Cyc: Toward programs with common sense. Commun. ACM **33**(8) (1990) 30–49

[27] Borgida, A., Brachman, R., McGuiness, D., L., R.: CLASSIC: A structural data model for objects. In: Proceedings of ACM SIGMOD International Conference on Management of Data. (1989) 58–67

[28] MacGregor, R.M., Brill, D.: Recognition algorithms for the LOOM classifier. In: Proceedings of the National Conference on Artificial Intelligence (AAAI). (1992) 774–779

[29] Farquhar, A., Fikes, R., Rice, J.: The ontolingua server: a tool for collaborative ontology construction. Int. J. Hum.-Comput. Stud. **46**(6) (1997) 707–727

[30] Brachman, R., Schmolze, J.: An overview of the KL-ONE knowledge representation system. Cognitive Science **9**(2) (1985) 171–216

[31] Karp, P.D., Myers, K.L., Gruber, T.R.: The generic frame protocol. In: IJCAI (1). (1995) 768–774

[32] Chaudhri, V.K., Farquhar, A., Fikes, R., Karp, P.D., Rice, J.: OKBC: A programmatic foundation for knowledge base interoperability. In: AAAI/IAAI. (1998) 600–607

[33] Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American **284**(5) (May 2001) 34–43

[34] Lassila, O.: The resource description framework. IEEE Intelligent Systems **15**(6) (2000) 67–69

[35] Fensel, D., van Harmelen, F., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F.: OIL: an ontology infrastructure for the semantic web. IEEE Intelligent Systems **16**(2) (2001) 38–45

[36] Brachman, R.J., Levesque, H.J., Fikes, R.: Krypton: Integrating terminology and assertion. In: AAAI. (1983) 31–35

[37] Kifer, M., Lausen, G.: F-Logic: A higher-order language for reasoning about objects, inheritance, and scheme. In: Proceedings of ACM SIGMOD International Conference on Management of Data. (1989) 134–146

[38] Snodgrass, R., Ahn, I.: A taxonomy of time in databases. In: Proceedings of ACM SIGMOD International Conference on Management of Data. (1985) 236–246

[39] Snodgrass, R.: The temporal query language TQuel. ACM Transcactions on Database Systems **12**(2) (June 1987) 247–298

[40] Snodgrass, R.T., ed.: The TSQL2 Temporal Query Language. Kluwer (1995)

[41] Sripada, S.M.: A logical framework for temporal deductive databases. In Bancilhon, F., DeWitt, D.J., eds.: Fourteenth International Conference on Very Large Data Bases, August 29 - September 1, 1988, Los Angeles, California, USA, Proceedings, Morgan Kaufmann (1988) 171–182

[42] Gutierrez, C., Hurtado, C., Vaisman, A.: Introducing time into RDF. IEEE Transactions on Knowledge and Data Engineering **19**(2) (2007) 207–218

[43] Gutierrez, C., Hurtado, C., Vaisman, R.: Temporal RDF. In: European Conference on the Semantic Web. (2005) 93–107

[44] Hurtado, C., Vaisman, A.: Reasoning with temporal constraints in RDF. In: Principles and Practice of Semantic Web Reasoning. Springer Verlag (2006) 164–178

[45] Lutz, C., Milicic, M.: A tableau algorithm for description logics with concrete domains and general tboxes. J. Autom. Reasoning **38**(1-3) (2007) 227–259

[46] Vinay K. Chaudhri and Mark E. Stickel and Jerome F. Thomere and Richard J. Waldinger: Reusing prior knowledge: Problems and solutions. In: Proceedings of the AAAI Conference on Artificial Intelligence. (2000)

[47] Cohn, A.G., Bennett, B., Gooday, J.M., Gotts, N.: RCC: A calculus for region based qualitative spatial reasoning. GeoInformatica (1997) 275–316

[48] Uribe, T.E., Chaudhri, V.K., Hayes, P.J., Stickel, M.E.: Qualitative spatial reasoning for question-answering: Axiom reuse and algebraic methods. In: Proceedings of the AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases. (2002)

[49] Kolas, D., Self, T.: Spatially augmented knowledge base. In: Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference (ISWC/ASWC2007). Volume 4825., Busan, South Korea, Springer Verlag (November 2007) 785–794

[50] Perry, M.: A Framework to Support Spatial, Temporal and Thematic Analytics over Semantic Web Data. PhD thesis, Wright State University (2008)

[51] Karp, P.D., Chaudhri, V.K., Paley, S.M.: A collaborative environment for authoring large knowledge bases. J. Intell. Inf. Syst. **13**(3) (1999) 155–194

[52] Karp, P.D., Riley, M., Saier, M., Paulsen, I.T., Collado-Vides, J., Paley, S., Pellegrini-Toole, A., Bonavides, C., Gama-Castro, S.: The EcoCyc database. Nucleic Acids Research **30**(1) (2002) 56–58

[53] Chen, I.M.A., Kosky, A., Markowitz, V.M., Szeto, E., Topaloglou, T.: Advanced query mechanisms for biological databases. In: ISMB '98: Proceedings of the 6th International Conference on Intelligent Systems for Molecular Biology, AAAI Press (1998) 43–51

[54] Topaloglou, T., Kosky, A., Markowitz, V.M.: Seamless integration of biological applications within a database framework. In: Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Bi-

ology, AAAI Press (1999) 272–281

[55] Alexaki, S., Christophides, V., Karvounarakis, G., Plexousakis, D., Tolle, K.: The ICS-FORTH RDF Suite: Managing voluminous RDF description bases. In: Proceedings of the 2nd International Workshop on the Semantic Web. (2001)

[56] Cumbaa, C.A., Lauricella, A., Fehrman, N., Veatch, C., Collins, R., Luft, J., DeTitta, G., Jurisica, I.: Automatic classification of sub-microlitre protein-crystallization trials in 1536-well plates. Acta Crystallogr D Biol Crystallogr **59**(Pt 9) (2003) 1619–27 22805983 0907-4449 Journal Article.

[57] Acton, B.M., Jurisicova, A., Jurisica, I., Casper, R.F.: Alterations in mitochondrial membrane potential during preimplantation stages of mouse and human embryo development. Mol Hum Reprod **10**(1) (2004) 23–32

[58] Jurisica, I., Wigle, D.A.: Knowledge Discovery in Proteomics. Mathematical Biology and Medicine. Chapman and Hall/CRC Press (2006)

[59] Arshadi, N., Jurisica, I.: An ensemble of case-based classifiers for high-dimensional biological domains. In: International Conference on Case-Based Reasoning, Springer-Verlag Press (2005) 21–34

[60] Jurisica, I., Rogers, P., Glasgow, J., Fortier, S., Collins, R., Wolfley, J., Luft, J., DeTitta, G.T.: High throughput macromolecular crystallization: An application of case-based reasoning and data mining. In Johnson, L., Turk, D., eds.: Methods in Macromolecular Crystallography. Kluwer Academic Press (2000)

[61] Snell, E., Lauricella, A., Potter, S., Luft, J., Gulde, S., Collins, R., Franks, G., Malkowski, M., Cumbaa, C., Jurisica, I., DeTitta, G.T.: Establishing a training set through the visual analysis of crystallization trials Part II: Crystal examples. Acta Crystallographica D (2008)

[62] Snell, E., Luft, J., Potter, S., Lauricella, A., Gulde, S., Malkowski, M., Koszelak-Rosenblum, M., Said, M., Smith, J., Veatch, C., Collins, R., Franks, G., Thayer, M., Cumbaa, C., Jurisica, I., DeTitta, G.T.: Establishing a training set through the visual analysis of crystallization trials Part I: 150,000 images. Acta Crystallographica D (2008)

[63] Cumbaa, C., Jurisica, I.: Automatic classification and pattern discovery in high-throughput protein crystallization trials. J Struct Funct Genomics **6**(2-3) (2005) 195–202

[64] Xia, E., Jurisica, I., Waterhouse, J., Sloan, V.: Runtime estimation using the case-based reasoning approach for scheduling in a grid environment. J ACM submitted.

[65] Xia, E., Jurisica, I., Waterhouse, J., Sloan, V.: Runtime estimation using a case-based reasoning system for scheduling in a grid environment. IBM Invention Disclosure (2007)

[66] Pease, A., Chaudhri, V.K., Lehman, F., Farquhar, A.: Practical Knowlege Representation and the DARPA High Performance Knowledge Base Project. In: Seventh International Conference on Principles of Knowledge Representation and Reasoning, Breckenridge, CO (2000)

[67] Friedland, N., Allen, P., Mathews, G., Whitbrock, M., Baxter, D., Curts, J., Shepard, B., Miraglia, P., Angele, J., Staab, S., Moench, E., Opperman, H., Wenke, D., Israel, D., Chaudhri, V., Porter, B., Barker, K., Fan, J., Chaw, S.Y., Yeh, P., Tecuci, D., Clark, P.: Project Halo: Towards a Digital Aristotle. The AI Magazine (2004)

[68] Cohen, P., Chaudhri, V.K., Pease, A., Schrag, B.: Does prior knowledge facilitate the development of knowledge-based systems. In: Proceedings of the AAAI-99. (1999) 221–226

[69] Guo, Y., Pan, Z., Heflin, J.: LUBM: A benchmark for OWL knowledge base systems. The Journal of Web Semantics **3**(2) (2005) 158–182

[70] Nebel, B.: Benchmarking of qualitative temporal and spatial reasoning systems. In: AAAI Spring Symposium. AAAI Press, Menlo Park, CA (2009)