N88-17278

AN EFFICIENT REPRESENTATION OF SPATIAL INFORMATION FOR EXPERT REASONING IN ROBOTIC VEHICLES

by Steven Scott
and Mark Interrante

Texas Instruments
Corporate Artificial Intelligence Lab
Dallas, Texas

ABSTRACT

The previous generation of robotic vehicles and drones were designed for a specific task, with limited flexibility in executing their mission. This limited flexibility arises because the robotic vehicles do not possess the intelligence and knowledge upon which to make significant tactical decisions. Current development of robotic vehicles is toward increased intelligence and capabilities, adapting to a changing environment and altering mission objectives. The latest techniques in Artificial Intelligence (AI) are being employed to increase the robotic vehicle's intelligent decision making capabilities. This document describes the design of the SARA spatial database tool at Texas Instruments, which is composed of request parser, reasoning, computational, and database modules that collectively manage and derive information useful for robotic vehicles.

1. INTRODUCTION

In order for future autonomous systems to efficiently reason about their environment it is necessary for them to maintain a consistent world model. Such models impose increasing demands for the supply, storage, and maintenance of large knowledge or information bases. While conventional database management systems (DBMS) perform much of the storage and retrieval of information, they are generally not powerful enough to deal with multi-dimensional (N-dimensional) world data or intelligently reason about the data they contain.

In support of Texas Instrument's work in avionic expert systems, a tool is being built to support the construction, maintainence, and usage of world models. The SARA spatial database retrieves, analyzes, abstracts, and maintains large multi-dimensional information bases that require efficient retrieval based of the spatial location of the data objects. The system manipulates both factual data, such as "the location of an airport", and derived abstractions, such as "areas of danger to low flying slow aircraft". Also provided is inferencing based on changing database values and the ability to perform complex algorithmic computations on the data. As the SARA tool continues to develop, the bonds between databases, algorithmic computations, and expert system reasoning will certainly be strengthened and more closely unified.

The examples are from a sample application of SARA in the airspace environment. The system is quite modular and flexible, allowing easy customization to specific applications.

2. SYSTEM ARCHITECTURE OVERVIEW

The SARA tool is composed of four modules: the request parser, triggering, algorithmic computation, and database. Each of these modules contain components for module communication and request management. Figure-1 shows the four SARA modules and how application programs using SARA communicate directly and only with the request parser.

2.1. REQUEST PROCESSING

The SARA system has an integrated syntax with nesting of event triggering, computation, and database operations. Each module has an associated request queue manager that is responsible for regulating its work flow and keeping track of paperwork. The module itself is allowed to process operations uninterrupted by the hassles of task management. This flexibility has led to the design of the SARA system shown in figure-2.

In the SARA system, requests arrive from the application program and conform to the SARA grammar. Requests enter the system through the request parser. After the request parser validates the syntax of the request, it ships the request to its request queue manager. The request queue manager then splits the requests into tasks for specific modules to handle. Tasks may be nested, requiring the services of another module. When the request queue manager of the intended module (i.e. the triggering module) receives tasks from the request queue manager of another module (i.e. the request parser), it schedules time with its module for completion of the task. At this point, the task does not contain uncompleted tasks for other modules. This task is atomic for the module and referred to as an operation.

2.1.1 REQUEST PARSER MODULE

The request parser module verifies the request syntax , initially splitting the request into tasks, and forwarding the tasks to the appropriate module for processing. After validating request syntax and contracting out the tasks to the other modules, the request parser continues with the next request. When the final result from the tasks is produced, the request parser responds to the application program with the result.

2.2 ALGORITHMIC COMPUTATION MODULE

The computation module is designed to contain compute bound algorithms which may benefit from the use of special numerical hardware. Depending on the application program, these algorithms could be short-term or background low priority processing. Since the SARA system is highly modular, the computation module could reside on a different machine running whatever language is available for the machine. Additionally, multiple computation modules could be established, each performing a specific range of tasks and thus giving the user the ability to configure a SARA system to solve the application problem.

2.3 SPATIAL DATABASE MODULE

The spatial database (SDB) provides rapid processing of multi-dimensional data within a relational framework. Applications needing spatial retrieval of information range from VLSI computer aided design to robotics. The SDB module accepts a SQL-like grammar sentence as input and translates it into syntax for the Texas Instrument's Relational Table Management System (RTMS). The SARA SDB approach is based on manipulating entire objects rather than breaking the spatial representation down, as in quad-trees. This processing is supported by the spatial index, which is a variation of the R-tree [Interrante 87]. The spatial index provides for efficient retrieval of information or objects possessing clustering characteristics that, in theory, would yeild less page faults and better performance than traditional database indexes. The SDB module consists of four components: the machine specific database tool (RTMS), the SQL to RTMS syntax translator, boundary data types and operators, and the spatial index.

2.3.1 RELATIONAL TABLE MANAGEMENT SYSTEM

The Texas Instrument's Relational Table Management System (RTMS) is a relational database for the Explorer lisp machine. Since RTMS is built on top of the lisp environment, any predicate lisp expression can be used in the where clause of a query as a condition for tuple selection. RTMS has been enhanced with special purpose data types and predicates for the spatial database. The following RTMS query produces the airplanes at Ohara airport. In other words, retrieve the airplanes whose position is enclosed by the boundaries of Ohara airport.

```
(RETRIEVE airplanes WHERE
   (enclose
      (RETRIEVE airports
         PROJECT (boundaries)
         WHERE (= airport-id "Ohara"))
      aircraft-position))
```

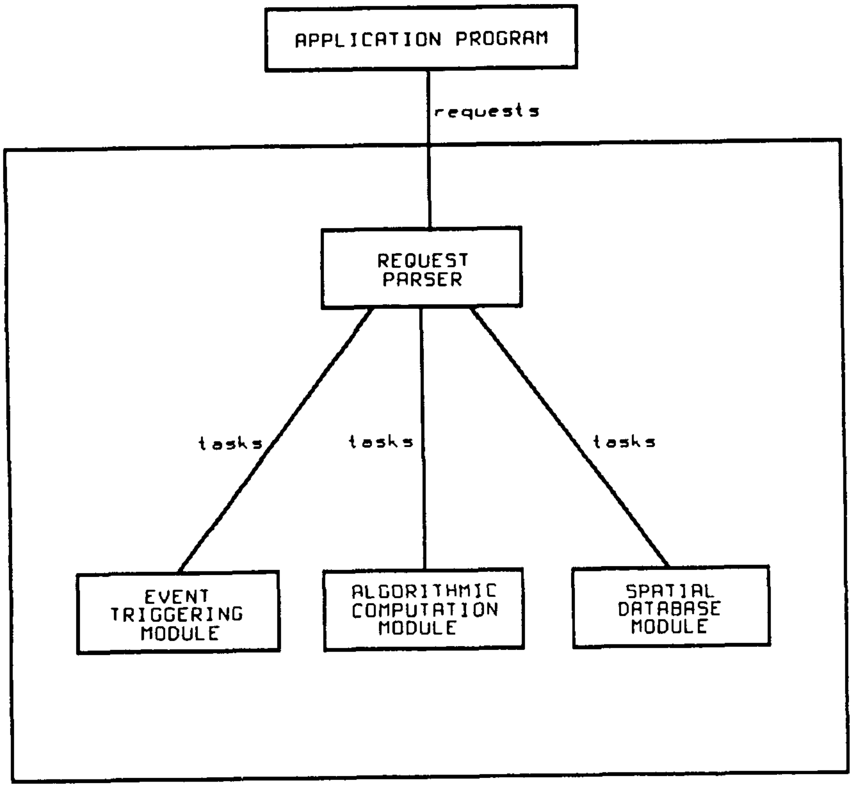

FIGURE 1. Overview of the SARA System Architecture.

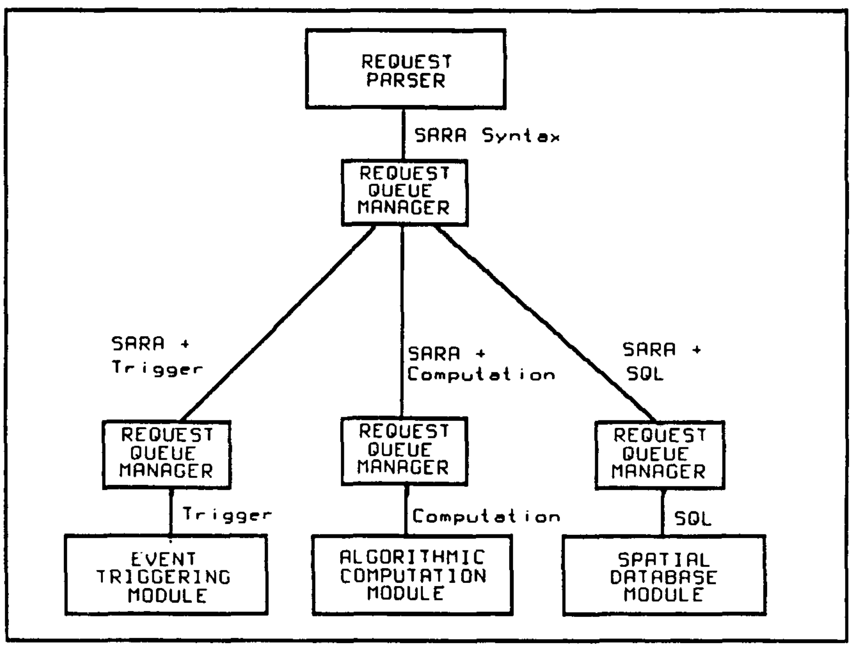

FIGURE 2. Internal Components of SARA Showing How the Sentence Composition Changes During the Life of a Request.

2.3.2 BOUNDARY DATA TYPES AND OPERATORS

In addition to the the usual database data types, a set of data types representing the boundaries of multi-dimensional objects have been defined specifically for the spatial database. The currently defined boundary data types are: rectangular solid, point, and cylinder. Future releases of the SARA system plan on supporting trajectories, cones, polygons, and ellipsoids. The operators UNION, ENCLOSE and OVERLAP are defined on the boundary data types. These operators form the basic level of a spatial database query understood by SARA.

2.3.3 SPATIAL INDEX

The SARA database provides faster access of spatial objects through the spatial index, which clusters objects based on locality. The spatial index used is a modified R-Tree, which is comparable to a multi-dimensional B-Tree. The R-tree differs from a Quad tree in that, with the R-Tree, individual objects are not decomposed into smaller components [Guttman 83]. In this implementation the actual shape of the object must be approximated by using one of the boundary data types. Support for modeling of objects at various levels of detail may be possible using persistent objects to organize the objects into a hierarchy [Thatte 86].

The spatial index provides partitioning mechanisms, allowing the relations to be clustered together based on similiar characteristics (i.e. static/dynamic and objects/regions). These partitions, defined by the database administrator (DBA) when the database is created, only influence speed and not functionality. Partition can be based on how often an object is updated (static or dynamic) or how large the object boundaries are (objects or regions). The static, dynamic, object, and region classifications are only examples of the partitioning advantages, the definition and use being under full user control.

2.4 EVENT TRIGGERING MODULE

The event triggering module is responsible for setting up, maintaining, and checking conditions on which the application program has asked to be notified. The action taken when the event or condition is triggered can be a notification to the application program or some SARA task to perform. One action might be to reanalyze certain database information and derive data abstractions needed some time in the near future. Other conditions could require the application program to be periodically or continuously notified of the event. The triggering module is sufficiently general as to suit most event recognition situations.

Triggers can be set to fire in four different ways. They can fire the first time a condition becomes true or every time the condition becomes true. In addition, triggers can fire continuously when a condition is true or they can fire only when the boolean condition changes value (i.e. from true to false).

Event triggering is implemented in a forward chaining rule based system with some support from the spatial database. Triggers are written as rules and may have embedded database and computation requests. Triggers that depend on spatial location are highly efficient because the database selectively postpones evaluation of the condition until it could possibly become true, thus eliminating unnecessary evaluations.

Two of the main uses of triggers are in notifying the application user of significant changing events and the creation of data abstraction levels in the database. The creation of abstraction levels is a powerful aspect of triggers as new tables and objects in the database can be built with derived relationships and groupings of objects. Triggers can also update the abstractions as the significant events change.

3. SUMMARY

This paper described the SARA spatial database tool which is composed of request parser, database, reasoning, and computational modules that collectively manage and derive large multi-dimensional information bases. Efficient retrieval of objects by spatial location is provided in the database. The system manipulates both the factual data and derived abstractions. Also provided is inferencing based on changing database values and the ability to perform complex algorithmic computations on the data.

ACKNOWLEDGEMENTS

The authors would like to thank all of those who have contributed to the SARA project at the Texas Instruments Computer Science Center, especially Karl Schricker, Trish Dutton and Mike Ernst.

REFERENCES

[Bentley79] Bentley, J. L., Friedman, J. H., Data Structures for Range Searching. Computing Surveys 11,4 (December 1979)

[Donelson 78] Donelson, W. C. "Spatial Management of Information", ACM SIGGRAPH (1978)

[Guttman83] Guttman, A., Stonebraker, M., R-trees: A Dynamic Index Structure for Spatial Searching. Memorandum No. UCB/ERL M83/64 Unviersity of California at Berkley (1983)

[Herot 80] Herot, Christopher F., "Spatial Management of Data", ACM Transactions on Database Systems, Vol. 5 No. 4, (December 1980)

[McKeown83] McKoewn, D. M. Jr., MAPS: The Organization of a Spatial Database using Imagery, Terrain and Map Data. Technical Report CMU-CS-83-136, Carnegie-Mellon University (July 1983)

[Interrante87] Interrante M., A Real-Time Spatial Index for a 3-dimensional Airspace Database. Proceedings from the National Aerospace and Electronics Conference. p. 1331 (1987)

[Orenstein86] Orenstein, Jack A., "Spatial Query Processing in an Object Oriented Database System", ACM SIGMOD, pp. 326-333 (1986)

[Reuss77] Reuss J. L., Chang, McCormick, "Paging Techniques for a Pictorial Database", Technical Report. Knowledge Systems Laboratory, University of Illinois at Chicago Circle, (June 1977)

[Scott87] Scott S., A Generalized Airspace Expert System Proceedings from the National Aerospace and Aeronautics Conference. p. 1315 (1987)

[Thatte86] Thatte S., Persistent Memory: A Storage Architecture for Object Oriented Database Systems. Proceedings from the International Workshop on Object-Oriented Database Systems. p.148 (1986)