Database-Centered Architecture for Traffic Incident Detection, Management, and Analysis

*Shailendra Bhonsle, Mohan Trivedi, and Amarnath Gupta**
Department of Electrical and Computer Engineering,
*San Diego Supercomputing Center,
University of California, San Diego, USA.
Email: trivedi@ece.ucsd.edu

Abstract

Reliable and efficient means for traffic incident detection and management are very important contributors to make the traffic flow smoother and safer. This paper presents various issues related to the development of an integrated software architecture for a traffic incident monitoring, mitigation, and analysis system. The novel concept of using a set of distributed databases, having many different functions and types, is proposed for distributed coordination of sensing, control, and analysis algorithms. This coordination paradigm using databases makes the whole architecture robust by providing means to efficiently manage current and past states of the monitored environment and the monitoring system. Use of a semantic event/activity database in the integrated architecture also provides high level abstractions through its query language to model traffic incidents and traffic behaviors. We also present experimental results of the use of a concrete database-centered architecture and algorithms in identifying important traffic flow events (such as tail-gating, exit from a ramp etc.).

Keywords: Incident detection and mitigation, semantic databases, video databases, traffic analysis.

1. Introduction

Safer, smoother, and congestion free traffic flow is indeed an important element of an enhanced quality of life in the modern world. Intelligent Transportation Systems need to provide solutions to make traffic flows smoother and safer. Our team, in partnership with the California Department of Transportation, has initiated a research project to develop novel integrated systems for traffic incident detection, verification, and mitigation. This project called ATON (Autonomous Transportation agents for On-scene Networked incident management) system has a broad scope and research agenda. At the core of this is the issue of the software architecure that provides robust interaction between various distributed detection, control, and analysis algorithms. Incorporating a collection of databases of right types in the integrated architecture provides for such robust interaction. The discussion of the database-centered abstract and concrete architectures for ATON is the primary focus of this paper.

The goal of the (ATON) system is to automatically detect, verify, and manage traffic incidents. A traffic incident is defined as "an event that causes blockage of traffic lanes or any kind of restriction of the free movement of traffic" [8]. To achieve this goal, ATON uses clusters of video and acoustic sensors, mobile robotic agents, and interactive multimedia workstations and interfaces, all connected using high-speed, high-bandwidth communication links. Figure 1 shows a scenario for traffic incident detection, verification, and management. The various steps taken in this scenario include automatic detection and reporting of incidents by sensor clusters [11], dispatch of a robotic *mothership* [1, 9] or remote agents to the scene of the incident, dismounting of a team of little robotic agents from the *mothership* to form a safe zone in a coordinated way, and assistance offered by the mother ship through tele-existence terminals [10]. The project is divided into five thematic areas of robotics agents with coordinated behavior, distributed multi-sensory networks, ubiquitous tele-existence, distributed virtual environments, and distributed databases. The use of distributed databases of various types and functions in the ATON architecture is discussed in the following sections of the paper.

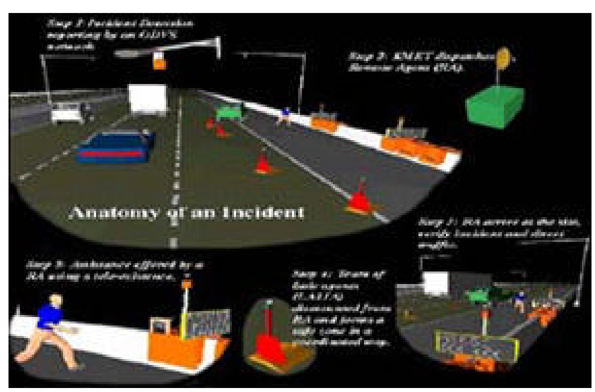


Figure 1: Scenario depicting steps taken for traffic incident detection, verification, and management.

2. Database-centered Architecture for Multi-sensory Monitoring Systems

This paper addresses the issues in the design of an integrated software architecture for coordination of many distributed control and analysis algorithms present in various thematic areas. The novel concept of using a set of distributed databases, having many different functions and types, is proposed for distributed coordination of these algorithms. This coordination paradigm using databases makes the whole architecture robust by providing means to efficiently manage current and past *states* of the *system* and the *monitored environment*. Such *states* are typically large in multi-sensory environments where some of the sensors could be as complex as omni-directional video sensors (ODVS). The database-centered architecture is shown in figure 2. The representation depicted in figure 2 is a functional representation of the abstract architecture, showing the abstract functions of sensor and actuator control, algorithmic control, environment analysis, state abstraction and management by databases, and system management. This figure hides a number of important architectural features that are elaborated below.

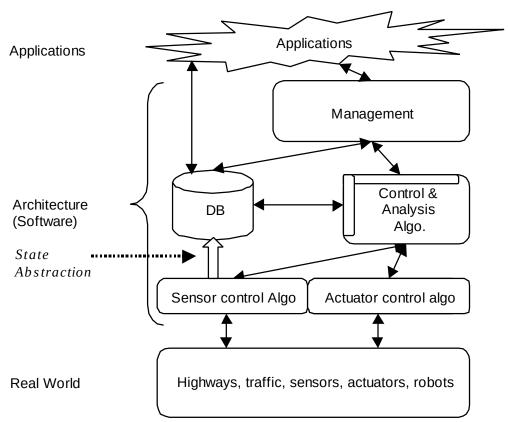


Figure 2: Database-centered abstract architecture for multi-sensory information management systems.

Distributed architecture: Every functional component of the architecture is logically distributed. The logical distribution of functions is mapped to a physical configuration of sensor and actuator clusters and to clusters of workstations. For example, the sensor control algorithms are distributed hierarchically to control clusters of physical sensor elements. Similarly, database is logically distributed in a hierarchical fashion, and any specific type of distribution is dictated by needs of storing raw or *aggregated* data and their types.

Control, analysis, and management algorithms: In addition to low-level (in the sense of being able to control sensors at a specific physical location) sensor and actuator control algorithms, there are distributed control algorithms. These collaborate (to improve *coverage*) or compete (to improve *reliability*) either to fuse data from physically distributed clusters of sensors or to co-ordinate the operations of physically distributed sensor clusters. These distributed control algorithms achieve their functions by interacting with the underlying distributed database through their query interfaces.

An important function of the architecture is to provide interfaces for incorporation of algorithms for the analysis of the monitored environment. These analysis algorithms are typically distributed and use the distributed database query interfaces. Typical functions provided by analysis algorithms are classification of objects, clustering of usual traffic related *events* and *behaviors*, causality detection etc. Such analysis is important in the ATON project, or in any multi-sensory monitoring environment, because they dynamically ***refine*** the notions of regions and objects of interest, traffic events and behavior, unusual traffic behavior, etc. For example, an analysis algorithm may continuously build clusters of usual traffic behavior for a particular time of the day. The unusual behavior can only be detected with respect to the current set of usual behavior clusters that continuously evolves.

The management function of the architecture typically provides functions to query the *state of the system* and the *monitored environment*. It can also be used to interactively *update* parameters of various control and analysis algorithms used in the architecture. *Safety* and *security* of the operation of the overall system is an important issue that can be addressed in this functional module. Safety refers to the *integrity constraints* that exist between different components of the system so that for proper operation only specific configuration of sensors, actuators, and algorithmic parameters are possible. Such integrity constraints are defined in and enforced by this module. Security refers to *access rights* of different application instances to different components of the system. For example, ATON applications may be used by police, ambulatory services, commuters, etc., each having access rights to only certain parts or *domains* of the overall system. The management module defines *access rights and domains* for various classes of applications for secure updates to sensor, actuator, and algorithmic parameters. Additionally, the management module can be used for performance and fault management of the overall system.

World and system databases: Many different types of databases are used in the ATON architecture for storage of the abstracted state. These databases can be broadly divided into two classes of *world* and *system* databases. The world databases deal with the abstracted state of the *monitored environment*, whereas the system databases deal with the abstracted state of the *monitoring* *system*. System databases become increasingly important as the complexity of the distributed monitoring system, which involves multiple clusters of sensors, actuators, and control and analysis algorithms, increases. These databases also help to appropriately implement safety and security related functions of the management module. For a scalable architecture, it is important to use a number of appropriately defined distributed system databases. On the other hand, world databases provide representations and retrieval mechanisms for the abstracted state of the monitored environment. These databases help make distributed control and analysis algorithms robust for two main reasons. Firstly, the abstracted state is typically spatio-temporally extensive, for example when visual or omni-directional [11] sensors are used. Specialized techniques need to be used to represent and store the current and past states. Databases take the onus of such representation away from the distributed control and analysis algorithms. Secondly, distributed algorithms can directly operate upon the stored aggregated *global states* (of the algorithm and of the monitored environment) which are physically distributed across multiple hosts, but the distributed database *makes it appear* as if this state is locally available and the distributed algorithm component only needs to operate locally.

Specialized semantic event/activity databases: Specialized databases are also used in ATON to provide high level semantic abstractions for flexible definition and detection of traffic incidents. These abstractions are those of semantic events, activities, and behaviors [2, 3]. **Events** are atomic semantic units that are detected over a bounded temporal extent by fusion of multi-sensory information. **Activities**, which have much higher semantic content than events, are spatio-temporal compositions of events. A **behavior** is defined to be a set of activities. A traffic **incident** has a very high level of semantic content, and is usually modelled using abstractions of *events*, *activities*, and *behaviors*. Apart from providing a direct semantic link to the notion of a traffic incident, semantic abstractions are also required by many distributed control algorithms to control sensors and robotic actuators in the real world. The computation of these abstractions from raw sensory data has prohibitive computational cost, and when the amount of sensory information produced is very large, it is best to delegate the role of these computations to a database. Incorporation of such specialized semantic databases in the architecture of a multi-sensory monitoring and control system, for example in ATON, is a novel architectural contribution of this research project.

In section 3, many types of databases in the concrete ATON architecture are briefly described. The semantic event/activity database is discussed in section 4, and some preliminary results of experimental studies of semantic activity detection are described in section 5.

3. Databases in the ATON architecture

There are many functional classes of databases that are required for distributed decision making to detect incidence occurrences. In this subsection, we only concentrate on *world databases* or databases that deal with the *state* of the monitored environments. Figure 3 shows some examples of these databases and functions that they provide. The distributed nature of these databases is not shown in this figure.

At the core representation level, the information gathered by multi-sensory environment can be treated as a huge multi-sensory database. A part of this database may consist of the video of the segment of highway being monitored. Another part of this database may consist of just the average intensity of light at the monitored site sensed by an appropriate sensor. Using appropriate sensor fusion algorithms, many functional classes of databases are derived from this raw multi-sensory database.

Spatial databases provide information about the spatial structure of the monitored environment like its division into regions corresponding to traffic lanes, curb, exit lane, etc. The **environment** databases provide static or dynamic environmental conditions like fogginess, rain intensity, luminance, etc. Spatial and environmental databases depend only on the site being monitored and conditions that are independent of the actual traffic related facts. As pointed out later, the semantic databases make use of these databases for detection of *useful* events and activities.

Traffic parameter databases provide a number of traffic related parameters like counters that indicate the flow of traffic, parameters to indicate the *burstiness* of the traffic etc. These databases make use of spatial and environment databases for computations of these parameters. These databases are typically temporally versioned.

There are two special classes of databases that deal with the semantics associated with various observations: the **feature** and the **event/activity** databases. The feature databases deal with features of either the mobile objects or the spatial regions. For example, these features could relate to color, size, centroid etc. of monitored mobile objects or to some statistical properties of the backgroud region. Features are the basic domain-dependent semantic information associated with monitored mobile objects or spatial regions. Events usually represent the state-transition of observed mobile objects or spatial regions. They represent a form of atomic semantic units, and the monitored environment can be described as a collection of these events. In the next section, we describe a specific semantic event database that stores events and their temporal structures.

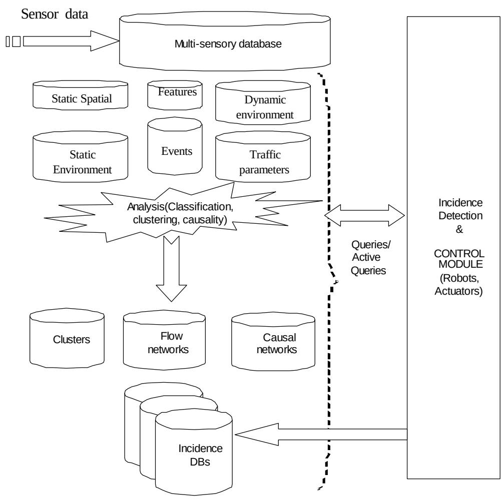


Figure 3: Multitude of databases of various types and functions used for incidence detection and management in ATON.

The need for behavior recognition “databases” in multi-sensory monitoring systems has lately been recognized [2, 3, 5]. Multi-sensory environments may include sensors as complex as visual sensors. Typically these sensors produce signals from which feature-specific data having limited semantic content is extracted. On the other hand a high-level behavior is a fusion of multiple such feature specific data with much more semantic content. Such high level behavior is compositional when viewed from a specific level of semantic granularity. Depending on the rules of composition from primitive feature data, different behavior can be defined. The goal of a behavior recognition database is to provide for the flexible identification of different patterns of behavior using the data stored in the database. Typically, data stored in such databases has higher semantic content than those contained in data produced by “signal processing” components of the monitoring system. Other usual goal of such databases is to provide mechanisms for efficient handling of huge amounts of data produced in such environments. The cited references [3,4,5,6,7] provide a good overview of issues in dealing with processing of semantic information within the context of multi-sensory information systems, and image and video databases.

In the above, a set of databases was listed that provide basic information about the monitored objects and space. The data in these databases is queried and processed to provide the information required by ATON distributed decision making algorithms for incidence detection and management. These data are also used by distributed analysis algorithms. For example, traffic flow analysis algorithms use these databases to produce flow networks. Many types and time-stamped versions of such flow networks need to be kept in a separate database. Such specialized databases are our next level of databases providing a very high level semantic view of the physically distributed monitored environment.

The ***type*** of a database refers to its underlying data model. The range of database types that provide above mentioned functions varies from the simplest and widely used ***relational databases*** to database models that can store and retrieve ***complex spatial and temporal structures***. In the following section, we describe one such complex database used in the ATON architecture.

4. Semantic event/activity database

As mentioned above, semantic activities are complex spatio-temporal compositions of semantic events. The event/activity database stores events and their spatio-temporal inter-relations. These events are detected by a transducer using sensory information, spatial information stored in a surrogate spatial database, and feature data of detected objects. The event/activity database query language provides high level semantic activity abstractions through its set of operations. These queries are executed against a database instance to detect activities. Also, there is provision for ***active queries*** that query abstractions similar to those provided by the activity query language, but it executes in real-time and can notify different entities of the ATON architecture as soon as a high level activity is detected. Many components of such event/activity databases together with their interactions with the sensor processing elements and the event detection transducer are depicted in figure 4.

Design of such semantic event/activity databases depends on many factors like whether events represent state or state-transition of objects, whether spatial, temporal, or spatio-temporal aspect is emphasized, how spatial and temporal uncertainties in event occurrences are represented, whether the model of composition of activities from events is statistical or combinatorial, etc.. For our modeling purposes, we defined events to represent state-transitions of observed objects of the system. These objects may be the mobile objects like moving cars in a highway segment, or spatial regions like a specified lane of the monitored highway.

We focused on the temporal composition of activities from events, and an important consideration was to take into account the ***temporal uncertainties*** in event occurrences. Assuming that such uncertainties for a given set of events is bounded by a constant, we obtained a model of composition of semantic activities from events that can suitably represent the ***concurrent occurrences of events*** in the presence of such temporal uncertainties. The concurrency could be between events associated with

multiple objects, or it could also manifest itself between many events associated with a ***single agent***. Such concurrency in the presence of temporal uncertainties is *not modeled* by ubiquitous sequential temporal composition rules or by its simple extensions.

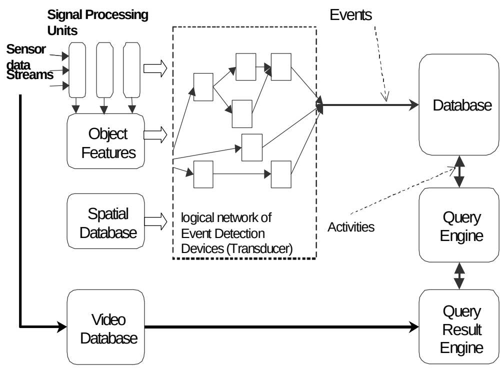


Figure 4: Detection, representation, and storage of semantic events and retrieval of semantic activities.

The combinatorial structure obtained above is a semiorder (a proper subclass of partial orders) based on the binary relation of *<precede,D>* such that *event x <precede, D> event y* if and only if the occurrence time of *y* is greater than the occurrence time of *x plus D*, where *D* is the fixed temporal uncertainty interval. Semiorders based temporal compositions represent a natural evolution of sequential composition rules. We designed a semiorder based data model and a corresponding query language that also embeds a semiorder pattern definition language.

System design and implementation: A prototype semiorder database was built using Java programming language and a native object-oriented database system. Many reusable modules for query, schema, and update engines were developed. Many algorithmic and architectural issues associated with the design and implementation of this database is discussed in [3]. In the following section, we briefly describe a set of simple queries to illustrate our use of this database in the ATON architecture.

5. Experimental Studies

A prototype of the semantic event/activity database was designed, developed, and used for detection of many complex traffic related *activities.* A set of traffic related atomic events like startLeftTurn, endLeftTurn, enterRegion, exitRegion, etc. were defined and the corresponding modules were inserted in the event detection transducer. Many atomic events are defined only with respect to the specific *spatial structure* of the monitored highway segment described below.

The spatial structure, depicted in figure 5, is stored in a surrogate spatial database. The highway segment is divided into many layers, where each layer consists of non-overlapping regions like ExitLane, Lane1, Lane2, Curb, etc. The event detection system queries this spatial database to appropriately detect atomic events like LaneChange by a car. The event/activity database stores the detected events and their spatial and temporal attributes.

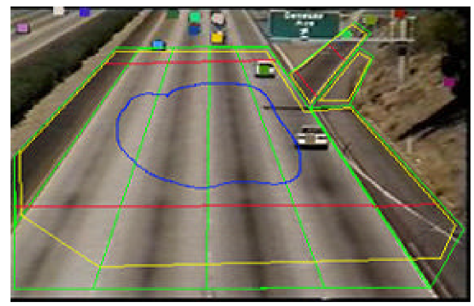


Figure 5: Spatial structure showing layers of regions of interest of the highway segment being monitored.

The dynamically changing monitored environment is represented as a collection of occurred events, their temporal structure, and their parameters in the database. Finally, complex activities were detected using the activity query language. Complex temporal structures over a set of events is flexibly defined using a *semiorder* pattern, which in turn is used by the query language. The execution of the query extracts the set of matching activities from the environment representation. Although the prototype supports very complex queries, for brevity, we only discuss a few simple representative queries. In our experiments, the Δ uncertainty bound was set to 100 ms.

Query 1: (***Exit***) Find white cars that entered lane1, then entered the exit lane, and finally exited from the exit lane.

Query 2: (***Tailgate***) Find a set of cars that entered lane 1 almost simultaneously.

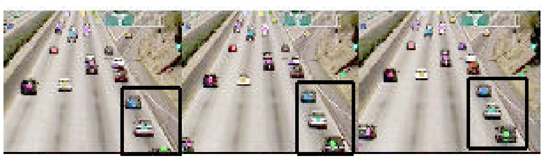
Query 3: (***Simultaneous***) Find an object that *while* entering lane 3 was also turning left.

The first query illustrates the sequential composition of an activity from atomic semantic events. The important point to be emphasized here is that *exponentially* many such *activities* can be defined on the same base set of *events*. The use of the database paradigm provides a tool to the user for ***flexible*** definition of what one needs. This compares well to the currently used alternative of fixing a set of high level activities and then detecting them in an ad-hoc but efficient manner. One of the results produced by this query is shown in figure 6.

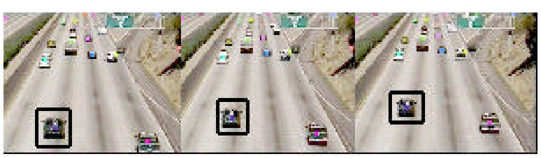
The second query shows a semantic activity that is ***multi-agent*** and involves the specification of

concurrency. The third query involves ***multiple concurrent events*** associated with a ***single agent***. Some of the results produced by these queries are depicted in figure 7. Being able to handle the requirement of modeling concurrency of events in the presence of temporal uncertainties is an important strength of our database model.

Figure 6: A result of the sequentially composed *Exit* query showing a car entering lane 1, then entering exit lane, and eventually exiting the environment.



a. Tailgating



b. Simultaneous

Figure 7: Results of (a) *Tailgate* and (b) *Simultaneous* queries. These queries illustrate the use of event detection uncertainty interval and the resulting multi-modal and uni-modal concurrency associated with monitored mobile objects.

These simple queries demonstrate the usefulness of the event/activity database model in ATON architecture. Combinations of the essential elements of these queries specify semantic activities with very complex structures that the query language is able to deal with.

6. Concluding Remarks

The role of databases in the integrated ATON architecture was described. It was pointed out that for detection and management of traffic incidents a number of databases having different *functions* and *types* are required. One of these many types of databases is the semantic event/activity database. These databases provide high level of abstraction for *flexible* definition and *detection* of semantic activities. These high level abstractions in turn help in the flexible definition and detection of *incidents*, or *conditions* that lead to traffic incidents.

On the other hand, databases also provide a new paradigm for the integration of various distributed control and analysis algorithms present in many thematic areas of ATON. This is especially true in a multi-sensory monitoring environment where the "global and local states" of the observed environment have large spatio-temporal extents and these need to be made persistent for robust interactions between many distributed algorithms.

Acknowledgements

Our research is supported in part by the California Digital Media Innovation Program in partnership with the California Department of Transportation (Caltrans). We would like to thank Mr. Ramez Gerges of Caltrans for his support and valuable inputs. We also wish to thank our colleagues from the Computer Vision and Robotics Research Laboratory who are also involved in related research activities.

References

[1]. M. M. Trivedi, K. C. Ng, N. Lassiter, and R. Capella, "New generation of multirobot systems", IEEE Int. Conf. on Systems, Man, and Cybernatics, San Diego, California, Oct. 1998.

[2]. S. K. Bhonsle, et.al., "Complex visual activity recognition using a temporally ordered database", 3rd Int. Conf. Visual Information Management, Amsterdam, June 1999.

[3]. S. K. Bhonsle, "Semiorder Model for Temporal Composition of Activities from Events in Multi-Sensory Environments", Ph.D. dissertation, Dept. of Computer Science and Engineering, Univ. of Cal. San Diego, Winter 2000.

[4]. W. I. Grosky, "Managing multimedia information in database systems", Comm. ACM, Vol. 40, No. 12, December 1997.

[5]. A. Gupta, S. Santini, and R. Jain, "In search of information in visual media", Comm. ACM, Vol. 40, No. 12, December 1997.

[6]. B. Yao, and B. Liu, "Rapid scene analysis on compressed videos", IEEE Trans. Circuits Sys. Video Tech., Vol. 5, No. 6, pp. 533-544, December 1995.

[7]. A. Del Bimbo et. al., "Symbolic description and visual querying of image sequences using spatio-temporal logic", IEEE TKDE, Vol. 7, No. 4, August 1994.

[8]. K. Ozbay and P. Kachroo, "Incident Management in Intelligent Transportation Systems", Artech House, 1999.

[9]. M. M. Trivedi, S. Roche, and B. Hall, "Teleautonomous Mobile Probes in Distributed Intelligent Environments," IEEE Int. Systems, Man, and Cybernetics Conference, October 2000.

[10]. H. Ishiguroa and M. M. Trivedi, "Integrating a Perceptual Information Infrastructure with Robotic Avatars: A Framework for Tele-Existence," IEEE/RSJ Intelligent Robotic Systems Conference, Korea, Oct. 1999.

[11]. K. C. Ng, H. Ishiguro, M. M. Trivedi, and T. Sogo, "Monitoring Dynamically Changing Environments by Ubiquitous Vision System," IEEE Int. Workshop on Visual Surveillance, June 1999.