Abstract

Dynamic Traffic Management Systems (DTMS) are intended to operate within Traffic Management Centers (TMC) to provide pro-active route guidance and traffic control support. The integration of multiple DTMS software systems requires the modification of the structure and design of the TMCs where they will be integrated. An open, scalable and parallel system architecture that allows the integration of multiple DTMS servers at minimum development cost is presented in the current research. The core of the architecture provides: a generic distribution mechanism that extends the Common Object Request Broker Architecture (CORBA); a generic creation mechanism based on the Abstract Factory pattern that permits an anonymous use of any DTMS within TMCs; and a generic naming mechanism (Registry) that allows the TMC to locate the DTMS servers in remote hosts without using any vendor specific mechanism. Finally, the architecture implements a Publisher/Subscriber pattern to provide parallel programming on top of the CORBA's basic synchronous communication paradigm.

This system architecture is used to propose TMC application designs. The system architecture was validated in a case study that showed the integration of DynaMIT, a prediction-based real-time route guidance system with MITSIMLab, a laboratory for the evaluation of Dynamic Traffic Management Systems. MITSIMLab includes a Traffic Management Simulator (TMS) that emulates the TMC operations. DynaMIT was integrated within TMS using the proposed system architecture.

The core of the system architecture was distributed under CORBA using IONA Technologies Orbix 2.0 Object Request Broker, and it was implemented in C++ using the object-oriented paradigm.

Chapter 1

Introduction

In the last decade, research in Intelligent Transportation Systems (ITS) has led to the development of various tools for the optimization of transportation systems. ITS has concentrated some of its efforts on the study of road traffic. Currently, several ITS technologies intended to achieve efficiency in the management of traffic operations are being developed. These technologies are based on complex software systems implemented in the Traffic Management Centers (TMCs). Advanced Traffic Management Systems (ATMSs) and Advanced Traveler Information Systems (ATISs) are the two main ITS technologies currently being developed. The use of both technologies in TMCs will lead to advanced traffic management with dynamic route guidance and traffic control in the future. However, the introduction of such diverse software systems poses difficult problems for issues of system integration, data communication, interfacing , and synchronization, among others. The study of these difficulties and the possible solutions is the basis of this work.

1.1 Dynamic Traffic Management Systems

Dynamic Traffic Management Systems (DTMSs) are designed to support the operations of TMCs. They are the latest generation of support systems and all internal operations are generally performed in a completely automated way. DTMS are dynamic: the management of the traffic network is based on proactive strategies as opposed to reactive strategies. Predicted conditions constitute the basis upon which these proactive systems generate strategies. Predicted conditions are used to generate route guidance and traffic control.

There is a very strong coupling between traffic control and route guidance, both in the modeling level as well as the physical level. The modeling issues are partially reviewed in 1.4. The physical issues include communication networks, system interfaces, database management, etc.

DTMSs may have varying levels of automation. At the most automated level, all operations (data collection, data processing, decision support, and data dissemination and execution) are performed by a group of software elements. DTMS must interact in real-time among themselves and with the other systems in the TMC. This requires the TMC to provide an open communication architecture that allows the integration of all core systems and DTMSs.

1.2 TMC overview

A TMC consists of multiple ITS systems: some core systems (e.g. the *Surveillance System*), and some interfaced systems (e.g. a Route Guidance System). The core systems provide the basic set of functionality needed to operate a TMC (collecting surveillance data, controlling signals, and coordinating incident response). The core systems are usually *legacy*[1] systems designed for a custom TMC. The TMC basic functionality is enhanced with the addition of interfaced systems (e.g. real-time adaptive traffic control, route guidance, etc.) The TMC must provide the necessary interfacing capabilities. Namely, the TMC must provide:

- A system architecture that allows the integration of all systems. The architecture must encapsulate the custom nature of the TMC's core systems, so that any additional system can be easily plugged in.

- Interfaces to other ITS components, such as Advanced Traveler Information Systems (ATIS).
- Interfaces to non-ITS components, such as police, fire and local organization.
- A unified database structure to support the integration, operating efficiency, and interface to other systems.

The fact that different TMCs may have different number of core systems suggests that the design of a generic TMC system architecture that allows the integration of any DTMS should provide encapsulation mechanisms to hide the custom nature of the core systems. These encapsulation mechanisms are provided through a series of interfaces. To obtain the maximum flexibility out of the system architecture, every system should have an interface.

A *thin interface* locates all the functionality in the subsystems, leaving the clients with a minimum shell, most of the times a Graphical User Interface. This provides freedom and expandability to the system interaction, because the functionality is finely distributed among many small systems. A *fat interface* locates more functionality in the clients and somehow reliefs the load of the server systems. This limits expandability, but it is easier and less risky to develop. The ideal interface will be somewhere in between: it is determined by finding the minimum functionality that the system must provide to the TMC, and then finding a compromise in terms of development risk and system expandability.

In order to design the required interfaces it is therefore important to identify and describe the functionality of the subsystems the TMC may consist of.

FHWA (1993) reviewed the state-of-the-practice of several TMCs and identified some of the desirable objectives:

- Collection of real-time traffic data and area-wide surveillance and detection.
- Integrated management of various functions including demand management, toll collection, signal control, and ramp metering.
- Rapid response to incidents and collaborative action on the part of various transportation management organizations to provide integrated responses.

- Proactive traffic management strategies including route guidance and pre-trip planning.

Of these issues, the first has already been deployed in TMCs; while the latter three have been implemented but not yet deployed.

1.2.1 Surveillance Systems

Traffic surveillance is an essential TMC system. Traffic information is collected using a variety of technologies: loop detectors, video detection, infrared sensors, vehicle probes, aerial surveillance, etc. The information is captured by the sensors, processed locally, and aggregated for transmission to the TMC. Regular scanning frequencies are 1/240s and broadcasting frequencies are in the order of 1 or 2 seconds.

1.2.2 Control Systems

One of the fundamental properties of a TMC is the capability for controlling a traffic network. Control may be centralized, distributed, or hierarchical. In a centralized environment, a central facility collects traffic status data and makes traffic control decisions. In a distributed environment, control is performed locally, generally at the intersection level. A hierarchical control configuration is a hybrid between central and distributed control. In this architecture, control is generally performed locally; control decisions are monitored by a central facility that may override local control to achieve optimized traffic flow on a sub-region basis.

Regardless of the specific architecture, the vision is for integrated, proactive control rather than reactive control. Proactive control requires the support of a prediction system. A wide range of options is available for the control, including real-time traffic adaptive signal control, adaptive freeway control including ramp metering, transit and emergency vehicle preferential treatment, and lane usage control.

For most control system configurations, the control software will reside within the TMC as one of the support systems. In the case of distributed control, the TMC functions as a supervisory node with control override capabilities. The design and redesign of TMCs needs to be sufficiently robust to accommodate variations of the architecture.

1.2.3 Incident Management Systems

One of the primary goals of ITS is to reduce congestion. Since a significant portion of traffic congestion is due to traffic accidents and other incidents, a primary functional requirement of a TMC is the detection and management of incidents. Because of the importance of this function, it is generally treated separately, although it involves elements of surveillance, monitoring, control, and decision support.

Certain information processing capabilities are required for the TMC to provide incident management. These include: integrated data management, real-time traffic model execution, image processing for area-wide surveillance and incident detection, and man/machine interfaces providing transparent access to needed information.

1.2.4 Decision Support Systems

Decision Support Systems extend the TMC's traffic control capabilities to support the decisions operators at TMCs have to make. All of the traffic models and simulations used in the TMC reside in the Decision Support System. Online models are used for developing response strategies. They must execute faster than real-time so that their results can affect decision-making processes occurring in real-time.

FHWA (1993) interviewed the managers from the most important TMCs in the U.S. The requirements for Decision Support Systems that the managers suggested included the following:

- Expert systems to aid in incident detection and management.
- Access to traffic simulation models.
- Evaluation models to support route diversion and route guidance.

Thus, existing traffic simulation models are not used online for three reasons: most models are extremely data-input intensive; the data structures in the models do not correspond to the structures in the TMC databases; some of the network models cannot be executed and analyzed within the time frame available for decision making. Currently there are no methods for the real-time evaluation of route diversion and route guidance strategies.

The overall scarcity of online models results from the fact that existing models have been developed for offline use and are difficult to integrate within an online environment.

1.2.5 Other Systems

ATMS is the core of ITS. As such, it is the integrating agent for both ITS and non-ITS systems. Within the TMC, interfaces are required to all other ITS systems including ATIS, APTS (Advanced Public Transportation Systems), CVO (Commercial Vehicle Operations) and AVCS (Automatic Vehicle Control Systems). Of these, the strongest coupling is between ATMS and ATIS. Centralized route guidance will require high performance processors and efficient algorithms to be resident in the TMC. The communications load will vary depending on whether vehicles communicate directly with the TMC or through a roadside processor.

1.2.6 Computing Environments

The typical TMC configuration uses a mini-computer (e.g., Concurrent, Perkin-Elmer, Modcomp) to operate the control system software, and a networked group of PCs to provide all other TMC functions. The dominance of PCs in the TMC primarily reflects the fact that historically PCs were cheaper than workstations. Additionally, much of the traffic engineering sofware was originally written under Microsoft's DOS. The prevalence of personal computers has created a de facto standard operating system in TMCs: MS-DOS. Many of the managers interviewed by FHWA (1993) expressed an interest in moving towards UNIX environments. Due to the date of this report, it is likely that the operating system conditions have changed. In fact, many TMCs are now updating their systems to UNIX platforms.

As an example of a recently updated TMC computing environment, Anaheim, CA's traffic control center is designed to manage the surface street network. Their systems run on UNIX workstations. The database is managed by a commercial database manager (Oracle^{TM}). The communications network is based on an ATM infrastructure, designed to be compatible with the existing Teleos ISDN PRI Network established by the Caltrans WAN (Wide Area Network) for video-teleconferencing. The ATM Internet-working infrastructure is linked with the Caltrans District 12 TMC and the City of Irvine ITRAC via an OC3 155Mbps SONET fiber optics network, and with the City of Anaheim TMC via ATM T-1 using T-1 facilities provided by PacBell. The system also includes MPEG 1 video transmission system between the UCI ATMS Laboratories and the Caltrans District 12 TMC, allowing for selection and display of freeway video surveillance cameras within District 12. The TMC is distributed using CORBA (Common Object Request Broker Architecture) to provide to external agents the following services: real-time data (VDS, RMS, CMS), CCTV switching, ramp meter control, and raw field device data.

1.3 Research Objective

DTMS are TMC-independent software systems intended to run in all configurations of TMCs. To accomplish this, the TMC must comply with an adequate generic system architecture that allows an easy integration/interface of both internal systems and external systems. Previous efforts have mainly focused on the development of custom ATMS inside the TMCs, or on stand-alone DTMS. However, there have been no efforts to develop an open standard architecture that allows the integration of DTMS within TMCs.

DTMS work independently of one another and no assumption should be made about the possibility of recoding them (or the TMC) for integration. Aicher et al. (1991) proposed four different levels of integration:

- Coexistence, which is in fact no integration. The subsystems operate independently.

- Unidirectional cooperation, which means a data flow only in one direction, e.g. from traffic control to route guidance. The sending system is active and is operating independently. The receiving system is performing an adaptive strategy based on the current control strategy.
- Multidirectional cooperation, which means a multi-way data exchange among the subsystems, e.g. to and from prediction, route guidance and route control. All subsystems have full knowledge of the current status of the other system but an optimization is performed independently.
- Full integration, which means solution of the control problem within one overall strategy. All models are known by all systems. This means that no cooperation rules are needed. The full integration is practically unfeasible because it requires recoding of all the subsystems to know each others logic.

There are logical and physical architectural problems that affect the integration of DTMSs into TMCs. The system architecture defines the physical means of interaction among different platforms (combinations of hardware and software), and deals with issues such as distribution, communications, location services, and concurrent programming. The application design defines the logical relations among subsystems and the work-flow of the system as a whole.

The objectives of a system architecture are to control the complexity and to define the means of distributing the work among the different subsystems. The distribution of the system into subsystems increases modularity and isolates implementation dependencies (Yourdon et. al. (1995)).

In this research, a system architecture is developed that allows the integration of various Dynamic Traffic Management Systems under multidirectional cooperation is developed. The architecture is *distributed, parallel, decentralized* and *open*. Distribution mechanisms will be assured using the Common Object Request Broker Architecture (CORBA). Parallelism is assured through a custom mechanism based on the *push model*[2] that extends the CORBA basic synchronous communication paradigm. The architecture allows the design of decentralized systems thanks to the interaction of selected *design patterns*[3]. Finally, the core of the architecture is generic, and permits integration of any DTMS by the development of *adaptors* between the DTMS and the core of the architecture. The main contribution of this research is a core system architecture that allows the integration of different DTMS into Traffic Management Centers. The core system architecture is the result of the integration of design patterns for *location* (Registry), *creation* (Abstract Factory), and *parallelism* (Push).

1.4 Literature review

As seen in section 1.2, Traffic Management Centers are responsible for the management of the operations of one or more regions of the traffic network. These operations include collecting data, activating sensors under a certain strategy, providing guidance to travelers, generating reaction plans to incidents, etc.

There is a lack of literature regarding real-time integration of dynamic traffic control and dynamic route guidance. Current research focuses mainly on the design of independent software solutions (either ATMS or ATIS), and most of the times these have been only tested off-line. There is extensive literature detailing individual models for route guidance and dynamic traffic control.

We will start by reviewing existing traffic control and travel information systems. Then we will analyze previous experiences in terms of combined ATMS/ATIS. We will finish by reviewing some proposed integrated DTMS frameworks. The purpose of this review is to identify the internal subsystems for each of the existing software systems, and to learn from existing frameworks. The literature review focuses on the following elements:

- Dynamic traffic control systems.
- Dynamic traffic assignment for ATIS.