DESIGN OF MULTI-PURPOSE FRONT END DEVICE FOR INTEGRATED TRANSPORTATION SYSTEMS

Waskitho Wibisono*, Hideto Ikeda, Nikolaos Vogiatzis*****

* Department of Computer Science, Ritsumeikan University 1-1-1 Noji- Higashi, Kusatsu, Shiga, Japan 525-8577
** Department of Computer Science, Ritsumeikan University 1-1-1 Noji- Higashi, Kusatsu, Shiga, Japan 525-8577
**. Transport Systems Centre University of South Australia BJ2-49 Bonython Jubilee Building, Adelaide SA 5000
E-mail: waswib@cyber.is.ritsumei.ac.jp, hikeda@cyber.is.ritsumei.ac.jp, nikolaos.vogiatzis@postgrads.unisa.edu.au

ABSTRACT

The integration of transportation systems is a very important issue within our society. In order to realize more secure and efficient systems, an integrated system needs to be developed that is decentralized, stable and highly automated. In previous papers we have proposed a three layer object model (3LOM) as the base architecture for the integration of transportation systems. This paper describes the design of the physical input/output sub-systems in the bottom layer of the architecture and how to implement this layer from the technical perspective. This layer is primarily responsible for data acquisition, filtering and transmission of data to the middle layer and related control centers. Local knowledge is organized to be used to perform controlling functions within a localized environment. This paper is an important contribution to design the logical view of the bottom layer as an important step to develop the proposed integrated transportation system as our ultimate aim.

Keywords: Data stream management system, Distributed system, Distributed object system design, intelligent transportation system, and knowledge-based system.

Paper Received [20 January 2004]. Accepted [21 March 2004].

1. INTRODUCTION

The complexities of modern commercial and civilian vehicular unit movement have highlighted a growing disparity between current transport and traffic management systems and the growing needs of society. In order to realize more secure and efficient transport management systems, an integrated system has been proposed by the work of Ikeda, Vogiatzis and others [1] recently, and they have suggested that in order to achieve this goal of transport systems integration it is necessary to design the system to be decentralized, stable and highly automated.

This complexity is especially prevalent in the metropolitan cities like Tokyo, Sydney, Bangkok, London, New York and Jakarta with road-based transport networking, highway management/monitoring, traffic control and in-vehicle route navigation systems are now fundamental functions leading towards better transportation systems in the future.

For those numerous reasons, the integration of road-based transport networks, highway management/monitoring, traffic control and in-vehicle route navigation are fundamental functions for better transportation systems in the future.

In general, traffic control systems are not new issues and there are many high-quality commercially available systems and have a long history of successful implementation. The most notable transportation management systems in the Asia and Pacific are SCATS (Sydney Coordinate Adaptive Traffic System) and SCOOT (Split Cycle Offset Optimizing Technique. The primary purpose of these systems is to manage traffic flow using electronic sensors and video information.

Most of current traffic control methods are centralized systems and are based on feed back algorithms using traffic demand data in the past; the time span between data collection, data transmission, and calculation of parameter settings varies from years to a minimum of 5 minutes. These methods are not optimal where demand changes traffic rapidly within a time interval of 1- 10 minutes, such as in the morning or evening rush hours.

2. ARCHITECTURE FOR INTEGRATED TRANSPORTATION SYSTEMS

For the most part the real time environment for transportation and control system management has a similar structure. First, there are many sensors to monitor and record various information of the environment. Second, the control unit analyzes such information and decides which strategies to implement to adjust the environment. Finally the control part converts such strategies into control commands and therefore manages the environment.

System IMAGINATION (*integrated multi-nodal traffic network system*) proposed by Vogiatzis, Ikeda et al [1] was designed with the objective of integrating transportation systems. The system manages not only various types of vehicles, but also facilities, environment and activities related to transportation. IMAGINATION can navigate drivers efficiently by using various types of information including Origin-Destination (OD) map, Route Choice (RC) information, and knowledge. The authors [2] also show the main architecture of IMAGINATION in Figure 1.

2.1. Three Layers Object Model for Integrated Transportation System

In order to integrate various kinds of transportation systems the authors proposed a layered architecture consisting of three layers. This was proposed to ensure that the architecture is secure and efficient. In order o realize those ideas, it was decided that the system should be decentralized, stable and highly automated.

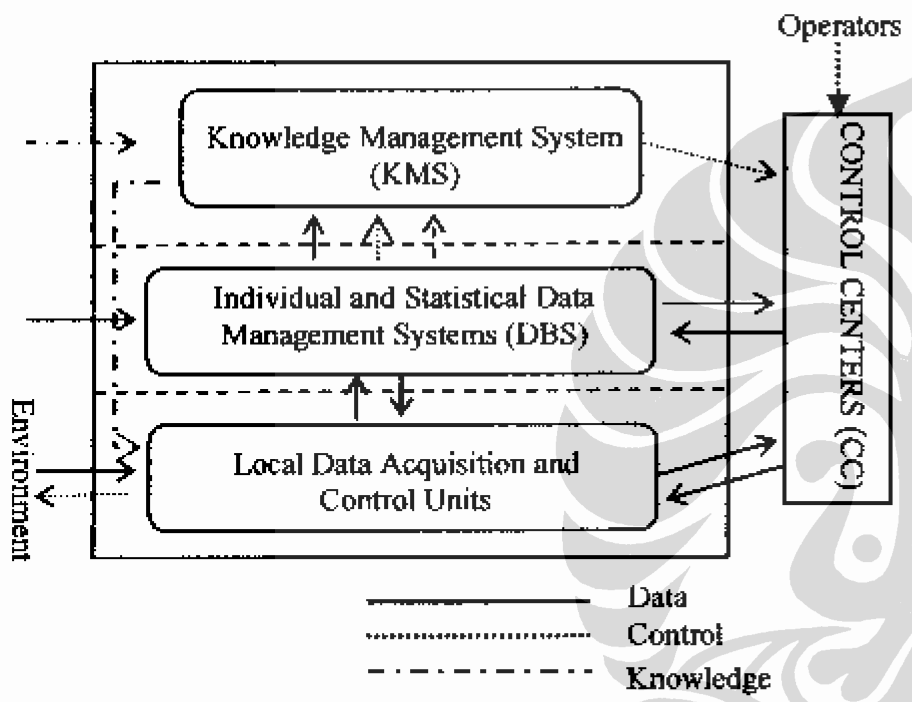

Figure 1. Three layers model of the integrated systems.

The three layers of the model are local units on the bottom layer, individual and statistical data management (DBS) system in the middle layer and knowledge system (KMS) on the top layer and independent control centers (CC) for some specified purposes.

The lowest layer manages specific locations intersection controllers; acquire data from those locations and the control the environment. Data acquired by a local unit is transmitted to the data management system or related control centers. Some data is used specifically for the control of individual locations whereas other data is locally summarized and then transmitted to the middle layer.

The DBS in the middle layer also manages data from individual controller and system-wide data for the purpose of resource management, analysis and transport facilities planning. The data in the middle layer are used for establishing new knowledge rules, supporting activities of managing transport office, registration of vehicles, the administrating of roads, etc.

In the top layer, KMS manages the control/behavioral rules associated with the environment such as intersections, facilities and appropriate rules for some conditions such as accident wardens, traffic jam etc.

Finally, control centers are established for specific purposes such as fire control center, highway monitoring center, road traffic monitoring etc. Each control center connects to the related local units to get appropriate data for its control mechanism.

3. REQUIREMENTS OF THE BOTTOM LAYER

This paper concentrates on the design of multi purpose front-end devices as an implementation of a new type of local control unit for the proposed integrated transportation system. From an implementation perspective for the local units; we need to devise an optimum strategy to integrate them into a networked real-time and distributed system.

Specifically, this unit is independent mode and a synchronized mode. First, we need to define an architecture which synchronizes all the distributed objects within the system.

The database and knowledge management systems are mainly implemented as software whereas other elements are developed as hardware implementations; however the most important of all is the need to establish an appropriate strategy as to standardize the object so as to allow generic unit controllers to be tailored for individual required functions.

3.1. General Requirements

In order to customize local units, their design needs to satisfied some important requirements i.e. they must be multi-purposes, decentralized and robustness.

a. Multi-purpose
   The local unit should be a standard device which can be customized according to its role in a localized environment. It means that the local unit must have all the basic functionality for the device to work in the integrated system and also be easy to customize it as is necessary.
b. Decentralization
   Each local unit should have capabilities to work independently and perform its local functions to manage the environment without need to be managed by the control centers. For some specified cases, the local unit has initiative to alert the control centers and ask for any related data for the controlling decision. This idea will give more responsibility and role to each local unit to control its localized environment.

c. Flexibility
   The local unit should implements the open system standard which enables interoperability among many devices from different manufacturers. As a result it will produce higher flexibility and lower cost for implementation of the system.
d. Scalability
   Scalability is to be understood as the possibilities to extend the system to achieve the appropriate purposes by integration of the local unit with other local units and control centers.
e. Robustness
   Any system that will be located in a hostile environment, such as traffic signals and traffic management controllers need to be a robust device. Furthermore, in case of computer network electrical spiking (a very common occurrence) the units need to be able to function on their own until network communications are re-established.

3.2. Functional Requirements

To make this device work according to our main goals, in this chapter we define also the functional requirements of the local unit. These requirements are designed to achieve to the missions of the local unit in the integrated transportation system. The description of each functional requirement is defined as follows:

a. Data transmitter
   Data processed in the units need to be sent to the middle layer. Data transmitter component plays roles of transmitting the data to the data management system in the middle layer or any connected control centers. This component also plays important function to receive data from the higher layers.

b. Data receiver
   In order to detect various environmental conditions, the local unit should be able to accept real-time signals from various sensor devices. The data receiver in this local unit is a component to handle those purposes.

c. Data filter
   The filtering component receives raw data from each connected sensors. There are instances where the data acquired needs to be transformed in a more meaningful way for use by this system, for example, noise reduction or some aggregation processes.

d. Local data manager
   In this unit, data from sensors usually arrives into the local controller non-linearly and continuously. We need to manage this data and be able to use it for special purposes like condition tracking of specified sensor or knowledge generation as well as routine data requirements. In general, this component has the role of local data management produced by other components in this unit.

e. Local knowledge manager
   To ensure the efficiency of real-time actions, it is necessary that the local unit has sufficient knowledge to be able to work independently.
   Local knowledge manager is a component to manage local knowledge and customized those ones for some special missions related to the real condition of the specified location. Some basic operation like how to add, update or even delete some knowledge is provided through this component.

f. Customization program
   Multi-purpose is the main requirement of this unit. In this case, this unit should have a capability to be modified or customized according to specific role of each local unit plays. For each component in the unit, customization program provide some services to customize those components. Simple user interface is provided for user to customize each component in the unit.
   To provide access to this unit from remote location, customization scripts are also defined. These scripts can be used using both direct remote connection and embedded web server installed in this unit. The purpose of these scripts are not only related to the customization processes but also can be extended to some execution tasks sent by control centers for special purposes.

g. Conditions detector
   In intelligent transportation systems, one of the most critical functions is that of condition detection. These conditions could be accidents, traffic jam, fire, or weather problems etc.
   This component plays an important role in handling and identifying conditions from specific sensors connected to the data receiver component and sends appropriate data to the local knowledge manager. Then the local knowledge data manager will use this data and knowledge to generate some actions.

h. Action controller
   Under coordination with the local knowledge manager, this component performs special tasks to execute tasks related to the conditions which are detected. An example may be the function of a local fire detector unit which may produce an audible alarm followed by information in the electric board about the fire

i. Web server
   Web server is created to give another access to the local unit of each location using web. We can use the

web server to provide some controlling services of the local environment, run customization scripts or browse the current conditions in specified location.

4. IMPLEMENTATION APPROACH

We have described the both general and functional requirements of the local unit before. In this chapter we define our implementation approach to satisfy the requirements we have described before.

4.1. General Architecture of Control Systems

Centralized architectures until recently have been common in control systems and control applications. Sensors and actuators are connected to a sub-panel and then connected to the central controller. The central controller contains a high-performance microprocessor running a custom application program to control all the I/O points connected to it.

One famous commercial product of this architecture is SCADA (Supervisory Control and Data Acquisition). This system is more focus on the supervisory level rather than a full control system [8]. In this centralized architecture, typically sensors and actuators have no internal intelligence and communication capability and connected to the system through specified controllers.

On the other hand there are many incompatibilities and difficulties to interconnect digital controllers from different manufacturers since there are no standard protocols; as a result every system must have custom application programs.

4.2. Local Operating Network

There are rapid changes in many types of system architectures, including control systems. There is a new concept of open architecture for control system called Local Operating Network (LON) that has been established.

A local operating network consists of intelligent devices, or nodes, that are connected by one or more communication media and that communicate with one another using a common protocol [6]. Nodes are programmed to send messages to one another in response to changes in various conditions and to take action in response to messages they receive.

The nodes may be sensors, controllers, human operator interfaces (displays, terminals, keyboards, buttons) actuators and other interfaces. Routers, repeaters and gateways are available for easy interfacing to other networks and architectures. Finally, these open standards give many advantages including lower cost of implementation and operation and also make greater possibility of new and exciting control applications reachable.

4.3. An Overview of LONWORKS Control Technology

A high quality Local Operating Network product available is LONWORKS that has been developed by Echelon Corp. of Palo Alto, California in partnership with Motorola. A LONWORKS network consists of intelligent devices such as sensors, actuators, and controllers—that communicate with each other using a protocol over one or more communications channels.

LONWORKS devices communicate with other nodes in the system using the LonTalk communications protocol on various physical communication lines. Each node has it own simple application program so that the control logic is distributed throughout the system [6].

LONWORKS Network Services (LNS) provides network operating system to control the network in this platform [7]. Its client-server architecture allows software client tools (e.g. human-machine interfaces, supervisory controls, operator stations, etc.) to be attached to the network. LNS also supports services of control systems encountered in different industries applications. The object-oriented approach used by LNS allows all network devices and attributes to be treated as objects with properties and methods.

4.3.1. Advantages of LONWORKS platform

There are some advantages of using LONWORKS platform technology. One important key of success of this approach is that this technology is an open standard platform. This standard allows many manufacturers to produce high volume hardware and software products that are interoperable which enable to the devices to be combined with higher flexibility and reliability.

LonTalk protocol is a collection of services that supports reliable communication among nodes and makes efficient use of the communications medium. A major goal of the LonTalk protocol is to give developers, from the same or different companies, the ability to design products that will be able to interact with one another.

This standard protocol makes possible a new generation of smart, low-cost products that communicate with one another. Using this approach makes it possible to create affordable networks of intelligent devices that sense, process, communicate, and control a multitude of applications ranging from handheld instruments to large process control systems in transportation system.

4.4. Architecture of the Local Unit

Using the LONWORKS approach means that we use the flat architecture. Flatness, in the LONWORKS case implies that the device and tools are at the same logical level, these will provides seamless interoperability between them [7].

The architecture of the local unit and the integration with other layers are illustrated in Figure 2.

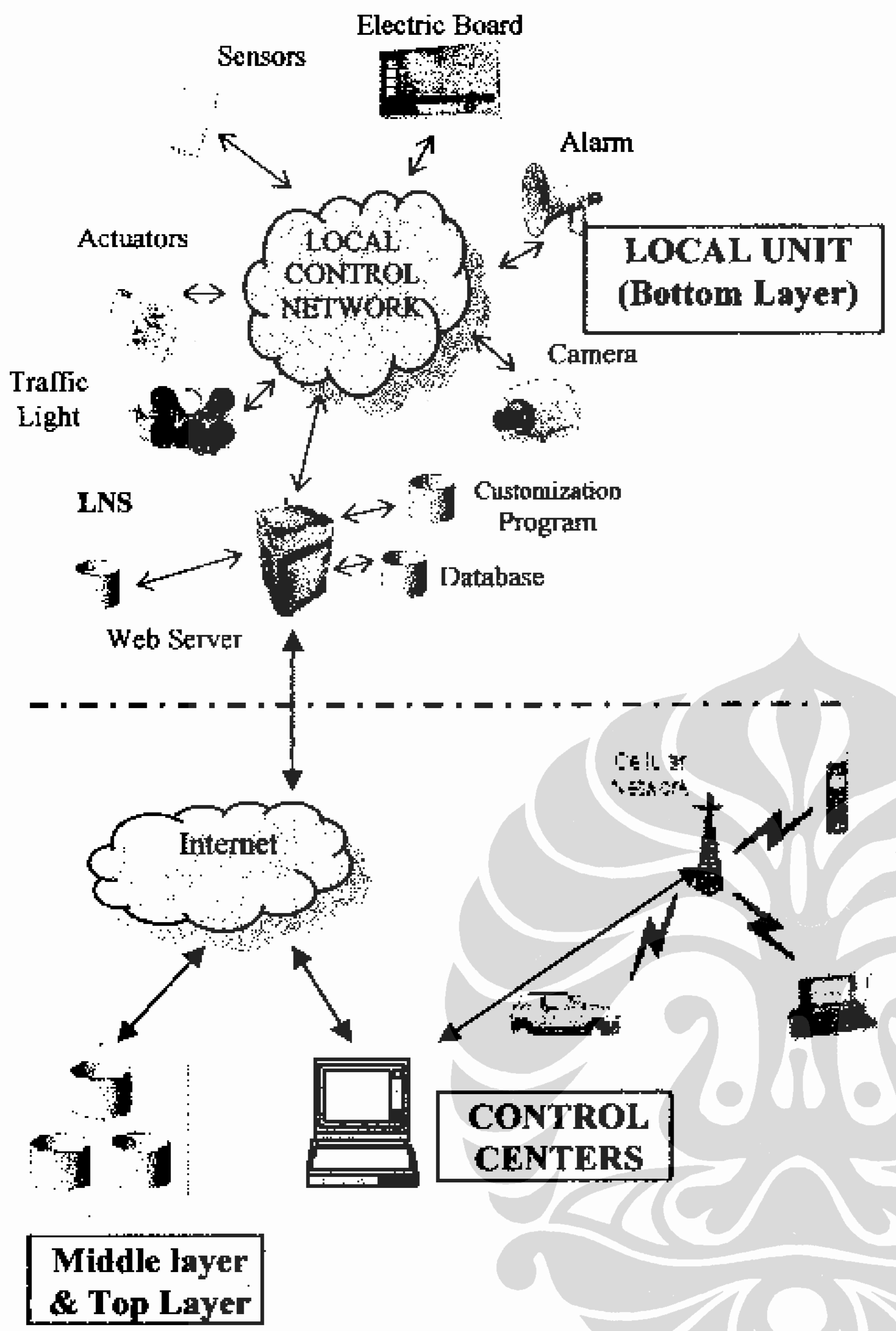

Figure 2. Architecture of the local unit and its integration with other layers and control centers

4.4.1. Functional Components in the Local Unit

In the Figure 3, we define the structure of the functional components of the local unit. This component architecture is designed in order to attain the functional requirements we have defined in the previous chapter.

We implement these functional components as an independent application within the LNS Server in the LONWORK platform. All of the components play an important role within the unit; furthermore it performs the mechanisms of the local unit to perform their controlling system according of each role in specified location.

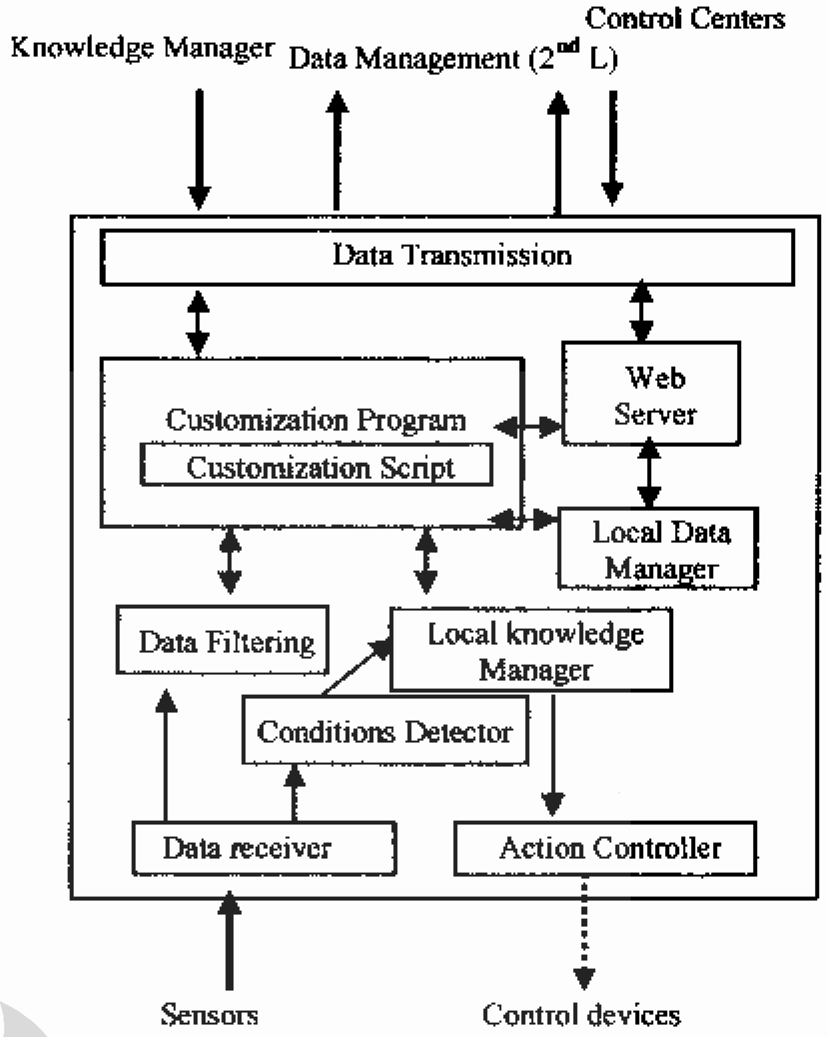

Figure 3. Functional components in the local unit

4.4.2. Local knowledge manager

Local knowledge component is established to achieve the efficiency of real-time performance of some special missions related to the condition in the environment [2]. Knowledge in this layer has one of the following two forms:

If *<condition>* **then force to** do {*<action>*}.
If *<condition!>* **then prohibit from** {*<action>*}.

In the definition form of knowledge, <condition> and <action> are defined as follows;

<condition>:=<simple condition>
| <condition> <logical operator> <condition>
| <negative operator> <condition>
|(<condition>)
<simple operator>:=<object name>.<variable> < comparison operator> < value>
< logical operator>:= **and** | **or**
<negative operator>:=**not**
<comparison operator>:= <|>|=|>=|<=
<variable>:= variable defined in an condition object
<object>:=<object class>|[<object class>.]<object name>
<action>:=<simple action>| <action> <simple action>
| **Repeat** <object>.<action> **until** <condition>;
| **While < condition> do** *<action>*;
<simple action>:=**Do** <object>.<action name>(parameter, parameter, ...);
<action name>:= registered in an action object

The Figure 4 below shows the basic model of the local knowledge

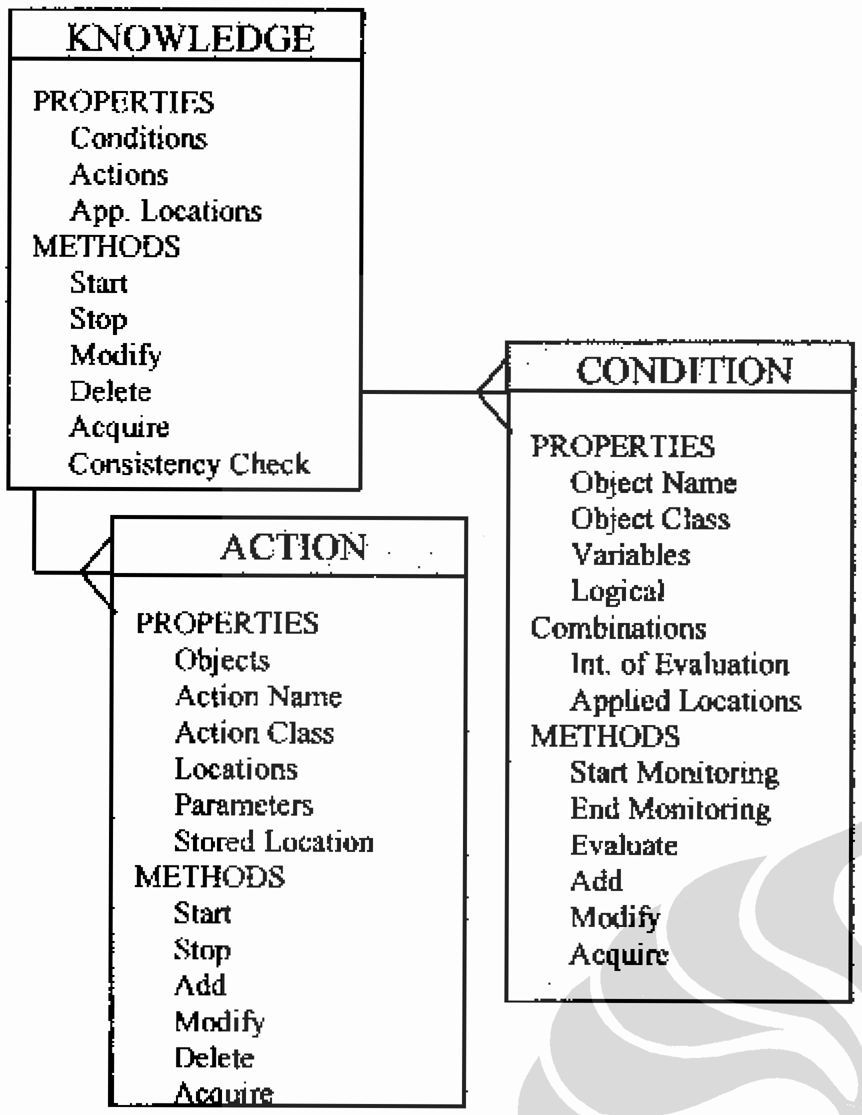

Figure 4. Basic model of the local knowledge

As an example within transportation systems:
"If a traffic accident occurs in some location of highway R1 and local road R2 has light congestion, close the highway from exit P1 to P2 and show information 'Take route R2 from P1 to P2' on electric bulletin board F."
The corresponding knowledge can be expressed as follows;
<object>(type):=R1(highway)|R2(local road)|P1(exit)|P2(exit)|F(bulletin board)

<object>.<variable>:=<value>:=
R1.traffic-condition={"accident","normal"}
| R2.number-of-vehicles ={0,1,2,...}
| F.content-of-bulletin-board= any strings

<condition>:= <heavy congestion>:= number-of-vehicles > 500
<action>:="R1.close(P1,P2) | F.show(<any string>)

Then we can express the knowledge by
If R1.traffic-condition = "accident" **and** R2.trafficcondition= **not** "heavy congestion"
Then force to do

```
{ R1.close;
   F.show("Take route R2 from P1 to P2");
}
```

In reality, there are several thousand such rules required to describe an urban transport network, however, by introducing **abstract objects** we are able to reduce this number significantly.

4.4.3.Customization Program

To establish multi-purpose capability we establish customization program inside a local unit. For each component in the unit, customization program provide some services to customize those components including customization of local database, local knowledge and also filtering variables.

This program will enable the local unit to be customized using installed user interface. To provide accesses of this unit from remote location, customization scripts are also defined. These scripts can be used using both direct remote connection and embedded web server installed in this unit.

Customization scripts are designed to manage all components in this unit. The purpose of these scripts are not only related to the customization processes but also can be extended to some execution tasks sent by any control center for specified purposes.

Some scripts are designed to be related to the mechanisms to set local knowledge or database. For example we can use some scripts such as *add_knowledge(id,knowledge)*, *delete_knowledge(id)* for local knowledge or for local database *addsensor(id,type,location)*, *deletesensor(id)* or *sensor_tracking(idsensor,starttime,endtime)* to get some specific data relating to a sensor in local database.

Other important scripts include scripts to customize and handle each sensor and control device which are connected to the local unit such as ID, port and type of sensor such as *set_sensor(id,port, type,datafreq)* or *set_controller(id,port,type,knowledgeid[])*. Some scripts also designed to customize the data filtering component such as *set_filtering_threshold(idfilter,threshold)* or run script such as *apply_datafiltering(filter_id,parameter)* for specified purpose.

4.4.4. Data Filtering

Most sensors will send the data to the local unit continuously and usually in large stream of incoming data. This is one of characteristics of a data stream system; accordingly this layer should has capabilities to perform filtering activities in order to reduce the large number of incoming streams or noise reduction and provide appropriate mechanisms to process the data related to specified purpose.

Some algorithms of filtering will be applied to achieve those aims. In addition, related to the data transmission component, data acquired needs to be transformed in a more meaningful way for use by this system. On the other hand as some data may need to be aggregated before being transmitted them to the middle layer or use those results for special purpose within this layer.

4.4.5. Local Data Manager

Within the local unit we also designed local database to manage information created by each components in this unit for special purpose for example tracking processes and also related to knowledge management mechanism. The web server also has access to this database and performs some database actions for some missions such as retrieving data through the web.

The local data manager actually is a system within the local unit to handle the local database. This system can be run using installed user interface in the local unit or using some customization scripts or use those scripts through remote connection or the web page.

4.4.6. Conditions Detector

In the local unit, a group of sensors is responsible for collecting data related to real time condition changes in the environment. The conditions detector is an application within the system to analyze the collected data in order to identify if any serious conditions have occurred which either have affected or will affect the environment in real time

This component mainly works to identify important condition and send appropriate parameters to the local knowledge manager. For this purpose we designed a problem detection system that can identify any problems within the traffic network (such as traffic congestion) in an automated way, and then ask the knowledge manager to send appropriate commands to the actions controllers as a respond to the environment.

5. CONCLUSION

The integration of transportation systems using high-tech computer systems based solutions is still in its infancy. This paper needs to be viewed in the context of the overall development of the necessary systems to allow for such integration. This paper is an important step to design the logical view of the bottom layer as an important layer to develop the proposed integrated transportation system as our ultimate aim. As such, the primary conclusion of this paper is that this is only the beginning; however, the proposed solution is based on the rigorous analysis of traffic systems and its implementation forms the next phase of work.

6. REFERENCES

[1] N. Vogiatzis, H. Ikeda, J. Woolley, and Y. He, "Integrated Multi-Nodal Traffic Network Systems" East Asia Society for Transportation Studies Journal, 2003.

[2] H. Ikeda, N. Vogiatzis, W. Wibisono, B. Mojarrabi, and J. Woolley, "Three Layer Object Model for Integrated Transportation System," 1st International Workshop on Object Systems and Systems Architecture, Victor Habor, Australia, 2004.

[3] N. Vogiatzis, "On Intelligent Transport Systems Applications in Public Transport," People for Public Transport - Missed Opportunities and Future Possibilities, Adelaide, South Australia, 2003.

[4] S. Babu, L. Subramanian, and J. Widom, "A Data Stream Management System for Network Traffic Management," presented at In Proceedings of Workshop on Network-Related Data Management (NRDM 2001), 2001.

[5] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and Issues in Data Stream Systems," Proceedings of 21st ACM Symposium on Principles of Database Systems (PODS 2002), 2002.

[6] ECHELON Corporation, "Introduction the the LONWORKS System", http://www.echolon.com/support/documentation/Manuals/078-0183-01A.pdf

[7] Reza S. Raji, ECHELON Corporation "End-to-End Solutions with LONWORKS Control Technology", http://www.lonmark.org/solution/home/end2end.pdf

[8] Axel Daneels, Wayne Salter, "What is SCADA", http://ref.web.cern.ch/ref/CERN/CNL/2000/003/scada