

Mobile Agent Approach for Traffic Load Balancing using Sensors

T. Karthikeyan, PhD.
Associate Professor
Department of Computer
Science
PSG College of Arts &
Science, Coimbatore, India

S. Sujatha
Associate Professor
School of IT & Science
Dr.G.R.Damodaran College of
Science, Coimbatore, India

N. Sudha Bhuvaneshwari
Associate Professor
School of IT & Science
Dr.G.R.Damodaran College of
Science, Coimbatore, India

ABSTRACT

The MATLB (Mobile Agent for Traffic Load Balancing) architecture proposed here is a conceptual service-oriented architecture supporting interoperability of systems implementing Ambient Intelligence (AmI) environment. The objective of the paper Mobile Agent Approach for Traffic Load Balancing using Sensors focuses on building an environment equipped with ambient intelligence that can control the traffic in metropolitan cities without crippling the mobility of users.

The growing advancement in Mobile Technology and its supportive tools has a greater impact on the social living of human beings. This agent based architecture with sensors, effectors, filters and middleware is one such approach that balances the traffic load and reduces the risks associated with congestion.

Keywords

AmI, SALSA, sensors, effectors, MATLB

1. INTRODUCTION

Ambient intelligence is a process of building smart functionalities in the physical world that we live. It helps in integration of sensing actuating, computing and communicating components [12]. Ambient Intelligence is based on key technologies like Ubiquitous Computing, Ubiquitous Communication and Intelligent User Interfaces. Due to advancements in technologies developing and deploying ambient services with flexibility, adaptability and situation handling still remains a challenge. This challenge ahead in the open system of AmI can be better handled with agent architecture [5].

Agents are autonomous software components, able to sense and to affect their operational context, and typically interacting with each other to orchestrate the achievement of goals beyond their individual capabilities[7]

2. AGENT-BASED AMBIENT INTELLIGENCE SYSTEM

The new paradigm of Agent technology provides flexible and portable solutions to Ambient Intelligence platform. Agents are amiable for building user-friendly and environment friendly systems with dynamic services are its features of mobility and adaptability. The agent based AmI system has an important role in servicing user and these services must be accessible via the mobile agent architecture. It is the responsibility of the agent to access information provided on real-time by sensors and effectors on vehicles and must act whenever an abnormal reading is recorded. [13]

AmI environment requires hardware support, interacting and interoperating elements represented by mobile agents for efficient functioning. Hardware component includes computer-based devices, sensors, effectors and filters. A backbone architecture is required to capture properties of the physical world or convert them into accessible digital format, or to transform digital commands into specific physical actions. Thus, the proposed Agent based AmI infrastructure maps into agent system environment, acting as the medium supported by middleware via which interactions with the world can be enforced[5].

To provide user level services and to control all the activities happening in the surrounding it is necessary to have software service components that may be in the form of protocols or queries. These protocols or queries acts as the middleware in the architecture.

The noted feature of an autonomous agent is its mobility. Applying this feature in AmI environment mobile agents can migrate among the hardware components like sensors, effectors and filters to obtain the local data. It is necessary that the entire architecture should work without any interruption and this is made possible with updated mobile agents that help in uninterrupted execution of the entire system [10]. As an additional consideration, embedding AmI services into agents enables also a good modularity: providing new services means that new agents must be “installed” into the environment, without affecting the older ones. Please note that a service could not require only a single agent, but several agents, producing a multi agent organization.

3. AGENT-BASED AMBIENT INTELLIGENCE IN MATLB

The objective of the prescribed proposal of MATLB (Mobile Agent for Traffic Load Balancing) architecture is to implement an AmI environment that controls traffic congestion in Metropolitan cities using agent based service oriented approach.

3.1 The Matlb Infrastructure

The MATLB scenario involves several components, that includes the use of traffic lights, video cameras, motion sensors and dark vision sensors (e.g., based on infrared or thermal) and actuators.

All the data and streams produced by the devices are captured and sent to a centralized database that logs all available information. The middleware is responsible for capturing the devices and to convert them into the appropriate data format.

The operating central takes charge of the entire MATLAB environment and in producing non-sensible information publicly available through the Web portal like smileys before further processing.

All the distributed sensors of the MATLAB infrastructure are either computer-based or coupled with an embedded PC, and thus can locally host the execution of data processing algorithms, Aml middleware components, and associated agents.

The preliminary processes begins with data sensing from the devices and the relevant information is extracted to detect or identify vehicles to detect or identify vehicles, their type, their registration number details at a particular location or street[5]. These sensed data, as well as raw data streams, are then made available to the middleware[8].

The middleware is a merger that merges heterogeneous components of the environment and additionally performing the function of converting data. It also supports agent execution in the form of a coordination logic, protocols, queries and discovery of event services.

The agents, that forms the backbone of the architecture is classified into sensor agents and effector agents. Sensor agents are responsible for processing and managing information acquired from devices and effector agents are responsible for performing actions in the ambient environment they live.

Sensor agents are directly connected with the environmental devices like sensors to execute the computation in the device itself. It can be done in the proposed MATLAB environment since this environment has a separate computational unit that supports agent execution. The effector agents are distributed within the environment and consumes data issued by sensor agents.

Sensor agents are attached to sensor devices and effector agents are independent in their environment merely consuming the data issued by the infrastructure. All the agents in the environment can interact with each other to co-ordinate the entire infrastructure.

3.2 MATLAB Approach

The Middleware is the building block of the environment working to support the system in the form of software developed using Java and J2EE technologies.

The middleware should take the responsibility of not only dealing with heterogeneous data components but should also provide basic support required for agent execution.

The choice of implementing middleware over the existing services is highly recommended because it makes the agent model light-weight reactive. First, to make use of a very light-weight reactive agent model. The second is the complexity of properly capturing and analyzing a large amount heterogeneous data streams and this is overcome by designing a customized kernel and middleware for agents rather than integrating complex data analysis algorithms into the existing system [17].

3.3 Structure And Functioning Of Agent Environment In MATLAB

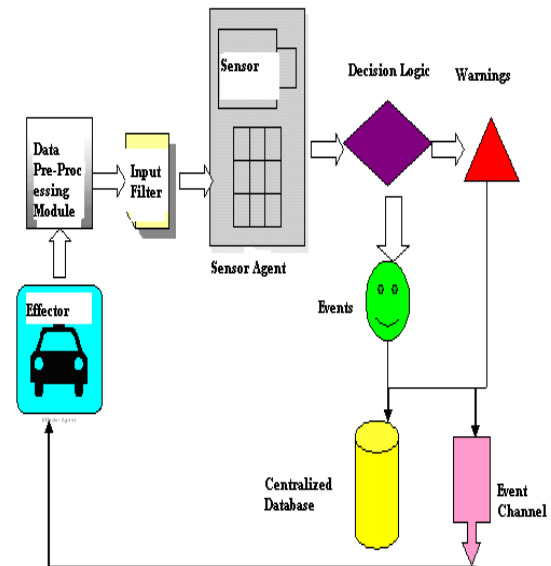


Fig 1: MATLAB Architecture[17]

The proposed architecture includes components like sensor devices, the middleware and the processing procedure. The data acquired by the sensor device is handed over for pre-processing. The pre-processing module applies algorithms to capture the image and identify the type of image. Then the heterogeneous raw data is handed over to the sensor agent.

The filter parses the heterogeneous raw data and converts it into the format recognised by the sensor agent. The data from the sensor agent is then handed over to decision logic that makes a decision on the situation that is being analyzed.

The output of the analysis is taken to the event generator, that prepares the data for the Aml environment. The MATLAB architecture identifies events and warnings. Events represent normal conditions, warnings represent anomalies in the Aml environment. These captured Events and warnings are then issued to the Aml environment to update information on the happenings in the environment. The sensor agents can directly control the devices through events and warnings require set of events to identify what the critical conditions is.

Based on the decision logic events and warnings are given by the sensor agent that can be either passed to the storage module or event channel in the MATLAB architecture. The MATLAB storage module stores the event into the centralized database for future reference and the MATLAB event channel notifies the event happenings. In the MATLAB Aml environment, each agent interested in the data issued by a sensor agent must register to the corresponding event channel. A notification is sent to the registered agent on event happening to analyze information. Agents can register or undo registration from event channel, thereby making reconfiguration dynamic. The synchronization and discovery module in MATLAB architecture make the availability of Sensor agents.

The sensor agents collaborates sensors and Effector agents collaborates user devices and infrastructure. Therefore the SALSA middleware, sensor agents, Effector agents do not represent all the intelligence in MATLAB environment.

4. MATLB AT WORK

Services and applications within MATLB can be developed as a set of effector agents that exploit the services provided by the sensor agents, or can be directly embedded into the decision logics of sensor agents. The following scenario is the work that is under consideration of MATLB proposal.

4.1 Scenario

The metropolitan cities in India is drastically improving, attracting people towards employment opportunities .But still these cities lacks in transportation infrastructure. As a result of huge population congestion is a major problem to the public. This proposed MATLB architecture provides solution using agents and middleware service.

4.2 Mobile Agent for Traffic Load Balancing(MATLB)

The Traffic Load balancing needs a balance between the sensor and effector agents. Sensor agents are stationary located in the environment monitoring images and the effector agents are non stationary in the form mobile phones, PDAs, Laptops, Pagers etc.,

Taking into consideration a busy six lane road, it is the work of the sensor device to scan the traffic and store the image files. The image files are processed to identify the vehicle type and handed over to the SALSA middleware.[17]. The threshold value is computed by the middleware logic based on historical data. The initial step starts with vehicle identification and count of each type. Then a comparison is made based on the count and threshold value by the middleware. The event is declared as normal if the count is less than the threshold value else warning is generated. The output of processing is distributed to the other agents in the environment. In case of warning There should be co operation between sensor agents in managing congestion duly informing the control room to initiate action.

4.3 Functioning Of MATLB

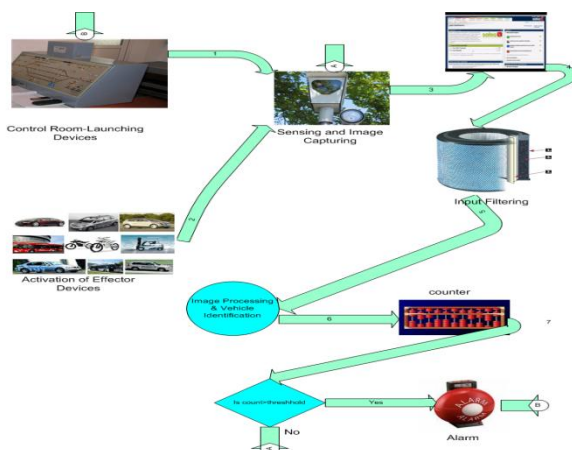


Fig 2: Functioning of MATLB

MATLB is a friendly agent that controls the traffic in metropolitan cities with the help of SALSA middleware.

The design of MATLB indicates the activities that include

initialization of Control-Room Launching devices which controls the operating devices like Sensors, Effectors, RFID Cameras, Filters, Counters, indicators. The working principle of MATLB starts with the initialization of Sensors and Camera controlled by a control room. The Sensors activates the effectors by capturing Images that is sent for filtering and the filter initiates vehicle identification by generation group tags. A counter is maintained simultaneously that keeps count of identified vehicles. This counter value is compared with the threshold value that is estimated based on load and available bandwidth. If the current counter value exceeds the threshold value then the alarm is sent to the control room and the action will be taken accordingly. If not then agent keeps continuing. The SALSA middleware is introduced between the between Sensors, Effectors and Filters, Counters in order to control the entire functioning of MATLB.

4.4 Role Of SALSA In MATLB

SALSA is the central management middleware for automatic networking system. SALSA brings the power of mobile networking architecture to the user through automation, distributed processing and web-based interfaces. Access to SALSA is performed through a simple web interface.

SALSA plays a major role in the MATLB architecture, where this middleware periodically collects information regarding the vehicle identification and movements from the control room. This information is consolidated into synchronous multi-dimensional table that contains details such as vehicle identity, time, location, sensor details, traffic volume, load capacity and estimates a threshold value. Threshold value is estimated based on speed limit which is normally 10 percent over the speed limit plus 2kmph. For example if the speed limit is 40kmph, the threshold is (speedlimit+10% over speed limit+2 kmph=40+4+2=46kmph). The traffic intensity is calculated based on the factors like type of road, speed limit, traffic density, atmospheric conditions, number of lanes, railway crossings, safety installations, road dimensions etc., These computed information are given as inputs for filters, counters to check whether the traffic density is within the stipulated range or not.

Table 1. Multidimensional Table for Vehicle Identification

Sensor Identification	SMPTS192I
Road Layout	Highway
Load Capacity	220
Length	100 kms
Breadth	180 ft
Number of Lanes	6
Elevations	3
Speed Limit	60 km/h

The current traffic intensity is calculated as follows:

Vehicle Id	Effector Id	Image Id	Vehicle Type	Driving Speed
TN38A B1213	SMPTE 292M	IMV0 04	Four Wheeler	60 km/h
TN38A S5143	SMPTE 456M	IMV3 02	Two Wheeler	40 km/h

Speed Limit = (Length of Road*width of Road)-(Number of Railway Crossings + Number of Safety Installations)

Traffic density= Type of Road+ Speed Limit+(Length of Road*width of Road)

Lane density=Traffic Density/Number of Lanes

$$CurrentTI(\sigma) = \sum \sqrt{(X) - (\bar{X})} / N$$

Where , TI stands for Traffic Intensity, X-Speed Limit, X Bar-Mean Speed

N –Number of Lanes and

Type of Roads include with their volume represented as integer

Highway-300

Middleway-200

Motorway-150

Subway-75

5. PERFORMANCE EVALUATION

Based on the study conducted using SALSA architecture in handling traffic density in Metropolitan cities using mobile agent MATLAB the above formulae are derived and a threshold value is estimated. Considering the speed limit as the major factor and based on the current traffic intensity and estimated threshold value an analysis is done to categorize the traffic intensity and speed limit into acceptable, acceptable risk, tolerant risk and unacceptable as shown in the graph.

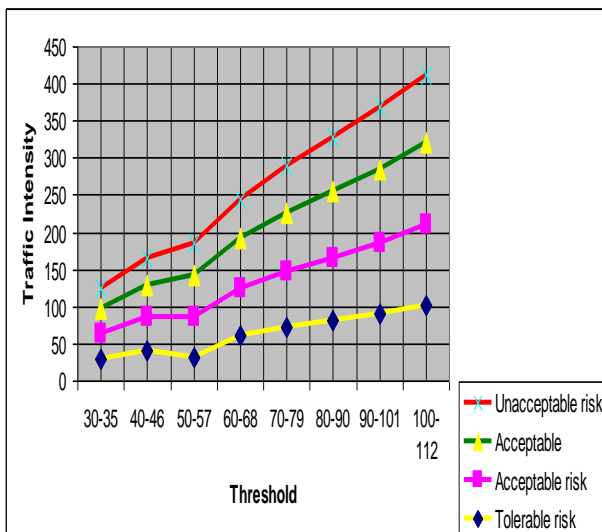


Fig 3: MATLAB Performance Evaluation based on threshold value

6. RELATED WORK

The agent based approach has taken its path in AmI environment with varied technologies. The major drawback being the co-ordination between the sensor and the effector agents.

The queue model approach for urban traffic control proposed by Grether et.al. simulates the interplay between traveller's and traffic signal agents in transport environment using a multiagent modelling approach[7].

According to Bo Chen and Harry Cheng after a decade of investigation on agent based approach in various traffic and transportation areas is still immature and need to be further explored to get ideas on integration of new mobile agent technologies.[3]

The Agentfly environment proposed by Michal Pechoucek and David Sislak simulates collision-free operations for numerous aircraft and to detect possible collisions using multi-agent environment.[11]

Satoh's approach[16] exploits mobile agents capable of interacting with an environment enriched with RFID tags and embedded devices to localize users and devices and helps in location-aware computing.

KIMURA[4] is an improved office environment that aims at monitoring and supporting human activities and collaborations through a set of agents. Agents, via sensors and monitoring of computer actions, can recognize user activities, which are then represented in a digital form as working contexts. These contexts, other than being possibly graphically rendered on users' displays, can act as the basis of inter-agent interactions. Via a blackboard schema, agents can push and pop messages and information, communicating to each other.

The SALSA middleware has been proposed in a healthcare scenario for supporting AmI through autonomous mobile agents[14]. This approach, by exploiting agents as entities capable of interacting with the devices of the AmI infrastructure and with each other, but the focus is mostly in providing context-aware information to visitors and alerts to doctors, rather than in providing coordinated services.

The GAIA[15] middleware is a complete framework for ubiquitous computing applications. Relying on active blackboards associated to specific locations of an environment, it can be exploited in AmI applications by acting as a media to coordinate the activities of various applications agents and devices.

7. CONCLUSION

This paper highlights on the application of agent technology in ambient intelligence environment, considering a real time traffic issues in transportation service offered in Metropolitan cities. This paper proposes an agent based model MATLAB for traffic control that functions with the support of SALSA middleware. The SALSA middleware exploits the distributed sensor agents to be aware of everything is happening in the AmI environment. The current MATLAB environment has few limitations with the use of effector devices and agents which could be of future interest provided that all the metropolitan cities are enriched with sensor networks, computer actuated traffic lights, Wi-Fi meshes, and RFIDs.

8. REFERENCES

- [1] André Bottaro, Remco Poortinga, Context Management and Semantic Modelling for Ambient Intelligence, Future

- Research Challenges for Software and Services, Telematica Institute
- [2] Belifemmine., A. Poggi, G. Rimassa, 2001, JADE - A FIPA2000 Compliant Agent Development Environment, Proc. of the 4th ACM, Conference on Autonomous Agents, ACM Press, June 2001.
- [3] Bo Chen, Harry H. Cheng, 2010, A Review of the Applications of Agent Technology in Traffic and Transportation Systems, IEEE Transaction on Intelligent Transportation System, Volume 11, No.2.
- [4] Christensen et al., 2004 H. B. Architecture Presentation: Experiences from Pervasive Computer Projects, Computer Science Department, University of Aarhus, Center for Pervasive Computing, Report Series, CøPC-2004-PB-1, revision 1.4, 2004.
- [5] Cucchiara, A. Prati, R. Vezzani, 2005, Ambient Intelligence for Security in Public Parks: the LAICA Project, Proceedings of IEEE International Symposium on Imaging for Crime Detection and Prevention, London, UK, pp. 139-144, June 2005.
- [6] Estrin, D. Culler, K. Pister, G. Sukjatme, 2002, Connecting the Physical World with Pervasive Networks, IEEE Pervasive Computing, 1(1):59-69, Jan. 2002.
- [7] D.Grether, A.Neumann, K.Nagel, 2012, Simulation of Urban Traffic Control: A Queue Model Approach, The 1st International workshop on Agent-based mobility, Traffic and transportation models, Methodologies and Applications.
- [8] Henrik Petander, 2007, A Netowrk Mobility Management Architecture for a Heterogeneous Network Environment, Helsinki University of Technology.
- [9] Jennings, 2001, An agent-based approach for building complex software systems, Comm. of the ACM, 44(4):35-41, April 2001.
- [10] Luca Ferrari, Giacomo Cabri, Franco Zambonelli, Agents and Ambient Intelligence: the LAICA Experience,
- [11] Michal Pechoucek and David Sislak, 2009, Agent-Based Approach to Free-Flight Planning, Control, and Simulation, IEEE Intelligent Systems.
- [12] Nikolaos I. Spanoudakis, Pavlos Moraitis, 2006, Agent based Architecture in an ambient intelligence context
- [13] Reza et.al, 2006, Ambiance: A Moile Agent Platform for End_user Programmable Ambient Systems
- [14] Agent Middleware for Supporting Ambient Intelligence in Healthcare, Workshop on Agents Applied in Health Care, Valencia (E), August 2004.
- [15] Roman, C. Hess, R. Cerqueira, A. Ranganathan, R. H. Campbell, K. Nahrstedt, 2002, A Middleware Infrastructure for Active Spaces, IEEE Pervasive Computing, 1(4), Oct-Dec 2002.
- [16] Satoh, 2005, Mobile Agents for Ambient Intelligence, Postproceedings of International Workshop on Massively Multi-Agent Systems, Lecture Notes in Computer Science (LNCS), Springer, 2005.
- [17] N.Sudha Bhuvaneswari, S.Sujatha, 2010, Vibrant Ambient Intelligent System for traffic congestion control in Coimbatore city(VAISTC4), Proceedings of the 12th International Conference on Networking, VLSI and Signal Processing, Pg.No. 288-293.