TOWARDS THE RAILWAY TRAFFIC MANAGEMENT USING MOBILE AGENTS

MUHSINA SHINWARI¹ AND SHER AFZAL KHAN²

¹Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology Islamabad, Pakistan
²Department of Computer Science, Abdul Wali Khan University Mardan, Pakistan

Received May 02, 2013, Revised June 18, 2013

ABSTRACT. Mobile Agent is a chunk of code that travels along a network from one host carrying its state of execution to another provides an execution environment. Mobile agents being programmed with decision making ability which can identify its route along a network. In this paper we present an agent based conceptual model for the railway traffic detection and management. The proposed model attempts to tackle with the bidirectional railway track conflict resolution using mobile agents for messaging and communication. The base system being the Centralized Traffic Control System, can efficiently ensures the smooth flow of traffic. The model also incorporated the accidents scenarios. In the end architectural model for both agent-owner and host are presented based on the given conceptual model.

Keywords: Mobile Agents; Mobile Agent Security; Mobile Agent Applications; Automated Railway Traffic Control.

1. Introduction: Mobile agent technology is based upon the use of mobile agents, which is gaining wide acceptance due to its intelligent nature and code migration capability. Mobile Agent is a chunk of code that travels along a network from one host carrying its state of execution to the another host an in a heterogeneous network. Mobile agents being programmed with decision making ability can identify its route along a network. Code in the mobile agent is made intelligent enough to decide for which next node to choose in the network. In doing so, mobile agent does not have to traverse the whole network to check which node is its destination. Thus, mobile agent provides network bandwidth conservation Not to be confused with client server architecture, mobile agents emerged as a powerful technology for development of complex distributed systems in 1990s[17]. In [1] a conceptual framework is presented for understanding and guiding development of systems incorporating mobile code. Their framework introduces three dimensions for the classification of such systems. Technologies - Mobile Code technologies are the mechanisms present in languages to enable and support code mobility. Design Paradigms - Design paradigms are the architecture styles that enable the technologies to be effectively utilized. Applications - These are the systems that use mobile code. In addition to the above dimensions, four design paradigms are described: The Client-Server paradigm which supports non-mobile distributed computing. The traditional client server model; The Pull Model: Client requests code and executes it at client. The Push Model: Client sends code to be executed to the server e.g. Remote evaluation. The Remote Evaluation & The Code on Demand paradigms which support weak mobility, i.e., where code, but not execution state, can move around a network. The Mobile Agent paradigm which supports strong mobility, i.e., where both code and execution state can move around a network.

Mobile agents while moving along a network, suspends its execution at the source node, and carries the code, data and state of execution while migrating toward the destination node. It resumes its execution at the
destination from the point where it had suspended at the source node and performs the task that is specified in its code. Mobile agents does not need a persistent connection for its complete execution. It carries its executable code along with the state of execution and data to the remote host, access the database or file system, unlike the RPC where a process invokes the procedure of the remote host. The implications of the process migration for mobile agents are common execution language, process persistence, communication mechanism between agents and hosts, and security to protect agents and hosts.

A very critical issue for mobile agents is the security when executable code is transferred across a network. Poorly programmed agents or agent platform/host can lead to unauthorized access or undesired code modification resulting into havoc. Mobile agent applications must be designed in accordance with the security features to ensure safe and smooth execution of the migrant code inside mobile agents. Extreme care must be taken at the time when an agent leaves its host and is exposed to an untrustworthy environment. Code modification of the mobile agent can be prevented by using encryption algorithms, but once the code enters a host and decrypts, it is vulnerable to undesirable changes. In case of mobile agents being a threat to the host environment, digital signatures can be used to authenticate the user and the incoming agent. Such agents may attack the host with loads of commands per unit time, thus using the CPU time and memory to the extent that would damage the agent platform.

Apparently, mobile agents security threats can be classified into two main categories. These are malicious agent attacks, where an unauthorized agent tries to invade a mobile agent platform and malicious host attacks that makes the agent code insecure. Each of these categories is addressed by the researchers and developers to countermeasure these security issues. Initially mobile agent code was given healthy volumes of research against malicious host attacks. Research is today beamed toward securing the agent platform or host against attacks from malicious agents or other non-trustworthy platforms.

Last decade witnessed different many mobile agent applications successfully implemented at industrial scale, giving boost to the mobile agent based technology. This field has been gaining most attention from the researchers and developers because of its features that gives the best possible architecture for services distributed. The development of the mobile agent based applications were entertained by Java which makes them even more acceptable due to java’s platform independence as agent moves in a heterogeneous network. The agent technology can significantly enhance the design and analysis of problem domains under the following three conditions [17]: (1) the problem domain is geographically distributed; (2) the subsystems exist in a dynamic environment; and (3) subsystems need to interact with each other more flexibly. Mobile agents are software components that are able to move between different execution environments [18]. Several benefits that mobile agents provide make mobile agent technology a good programming paradigm for building agile distributed systems [21]. Mobile agents can be created dynamically at runtime and dispatched to source systems to perform tasks with the most updated code and algorithms. Mobility significantly enhances the flexibility and adaptability of large scale distributed systems[16].

This paper presents another system based on mobile agents. An agent based system for railways traffic detection and control that is Railway Traffic Management System using Mobile Agents technology is proposed in this paper. Here, the centralized traffic control system is taken into consideration base system for discussion. Railway tracks are often bidirectional and may share a single track at the same time in opposite directions. Detection of the traffic at a certain coordinates and then scheduling and rescheduling of the trains is proposed to be made efficient and smooth by introducing mobile agents. The rest of the paper is organized in a manner that section 2 gives an overview of the related work done in this area. Section 3 presents the conceptual model of the proposed system. Section 4 discusses the proposed Conceptual model for Railway Traffic Management System using Mobile Agents. Section 5 gives a discussion on the scenarios of the railway traffic conflicts and their management using mobile agents. In the end paper is concluded and future prospects in this area are presented.

2. Related Work. Mobile agent systems also sometimes called middleware provide a framework in which mobile agent applications can be developed and managed. It is the basic execution environment for mobile agents. Need for efficient searching, collecting and retrieval of information over widely expanding internet have made it obvious for the researchers to develop systems for overcoming these issues. Contribution to this field is warmly welcomed in the industry since the earliest mobile agents systems such as Agent TCL/D’Agents, ARA, Concordia, mole, and Tacoma etc [2][3][4][5][6][7] were developed. These systems provided the basic information management and retrieval applications. Currently these applications are implemented in various systems such as Unix Operating systems, network management systems, allowing synchronous and asynchronous communication.

Concordia applications that access multiple databases/sites in dynamic environment. It enables legacy applications that need to be accessible to mobile workers Integrate with distributed objects e.g. CORBA (i.e.
providing Business-Business interfaces for Electronic Commerce). One of the major advantages of Concordia is to enable applications with network latency issues to process data locally on the network and report the results (i.e. wireless communication to client devices with small footprints). It also provides Cross platform operability (i.e. applications that need to access/deliver information via a combination of laptops, PDAs, smartphones, windows, Macs, Email, Voicemail, Pager, or FAX). Not the least, it monitors Network nodes and report results or take action without continue polling (i.e. Network Monitoring Applications) [5]

Mole is the first ever mobile agent system that is developed in java programming language. It is continuously under improvement providing development and execution environment for the mobile agents to be used in the distributed computing environments. [6]

Tacoma is a mobile agent system application mainly designed for the Operating system support for the mobile agents. It focuses on addressing the issues that are traditionally taken care by other distributed computing systems such as client server architecture. Tacoma is used in different industrial projects, one of which is weather update program called StormCast. Supporting agents written in variety of programming languages, Tacoma is installed on UNIX. It was introduced with limited security measures, but recent versions of Tacoma has more focus on security issues related to mobile agent systems along with other services. [7]

Voyager supports both the traditional client-server architecture as well as agent-based architecture. It only supports java which goes against the acceptance of this system commercially. One of its recent implementation is in grid computing where it used for dynamic agent identification through RPC. It provides lots of flexibility and extensibility for the products that are being created with voyager systems. [8]

Agilla is a mobile agent middleware that facilitates the rapid deployment of adaptive applications in wireless sensor networks (WSNs). Agilla allows users to create and inject special programs called mobile agents that coordinate through local tuple spaces, and migrate across the WSN performing application-specific tasks. This fluidity of code and state has the potential to transform a WSN into a shared, general-purpose computing platform capable of running several autonomous applications at a time, allowing us to harness its full potential.[15]

Mobile-C has been used to stimulate highway traffic detection and management. Agent Communication Language(ACL) provides a way for inter platform communication as the agents and agent platform both use the same communication mechanism. C/C++ is used to make efficient the interfacing between the low level software modules and the hardware. Dynamic code deployment and unexpected process control is made possible through agents in this system. Intelligent transportation systems require interoperability and flexibility with distributed computing capabilities. [16]

There are some security applications for mobile agents summarized below:

SeMoA stands for Secure Mobile Agent. It is the best known effort to tackle the issues of malicious host and malicious agents in mobile agent technology. JavaSE is the developing environment of SeMoA. It is represented by an Onion Model, consisting of several layers of protection. The security policy in SeMoA performs, authentication of the incoming agents, digital signature, and permission configuration. Creation and deletion of a mobile agent is under the control of this security policy. It supports agents spawning and is integrated within the middleware; therefore, dedicated and self-managed agents implement anonymity. Policies are defined specifically for a single agent or a specific owner’s agent, which makes it possible to use communication-dedicated agents. Communication-dedicated agents can be started by another trusted owner.[8]

Security threats that mostly retain the researchers’ attention in Mobile agents are compromise on the confidentiality, accountability, availability, integrity and anonymity. Anonymity i.e. hiding the identity of the mobile agents is most tough among all the security requirements for mobile agents. Web browsing makes vulnerable the privacy and anonymity of the mobile agents to the extent of eavesdropping and observation by other non-trusted malicious hosts. This Paper provides two solutions, one of which is based upon the Private Key cryptography that involves the part of the information become part of the decryption key at host site. The other one is Zero-Knowledge-Proof technique which is not elaborated in the paper, instead left for the future work. [10]

Mobile Agent Security with the IPEditor Development Tool and the Mobile UNITY Language

In this paper author has used the IPEditor as the designing tool with visual support for modeling agents, while Mobile UNITY, a formal specification language for mobile agents’ applications. Formal representation of the specification in the IPEditor is translated into Mobile UNITY program using the IPEditor verification facility. Mobile UNITY logic notation is used to describe the security requirements. Verification of the security requirements is a three step process which proves that mobile UNITY program and thus IPEditor model satisfies the Mobile UNITY logic notation. An example of e-catalog is given in the paper which follows the steps of security requirements verification. In the first step the formal representation of the specification is modeled in IPEditor and then translated into the Mobile UNITY program. In the second step the security specification is presented using sign and decipher function and an authentication function verify which satisfies the given Mobile UNITY logic notation.
Security specs are then verified in two parts i.e. verifying the registration of the authorized servers correctly and verifying that data sent by malicious servers is not registered. In the end the technique described in this paper is compared to formal techniques such as Proof Carrying Code PCC, and Ambient Calculus. Mobile UNITY handles mobile agent security issues more efficiently than any other formal specification language.[11]

A Secure Mobile Agent System Model Based on Extended Elementary Object System

This paper addresses the issue of mobile agent platform security in the prospect of formal methods. The solution is based on a security system model developed using Extended Elementary Object System. Five aspects extend the Elementary Object system to make its Object Oriented properties capable of modeling agent systems. Model is presented as a three layered architecture i.e. Agent platform layer, Agent layer and the security layer. Authentication, communication, agent- creation are made secure in the agent platform level; while the mobile agents actions such as reaction to incoming message, self initialization and decisions to move are taken care in mobile agent level. Security Mechanism level uses encryption and decryption techniques to make the mobile agent safe from malicious host attacks. A synchronous firing mechanism is the key element that detects malicious behavior of the agent host. Simulation based results verifies the strong mobility and correction of the model presented. Refinements in the model would do for a trust server which can make trust relationships among the hosts, and agent owners. [12]

Today automated railways are mostly controlled through a signaling system called Centralized Traffic Control[14]. There is a train dispatcher’s office that controls the train switches and signaling which must be obeyed for the smooth and safe traffic across the railroad under the CTC territory. The dispatcher’s office has a graphical depiction of the railroad which is under its control, which helps it to keep track of the train location. The railroads are most of the times bidirectional. There can be two trains, occupying the same railway track at the same time at different locations. The conflict arises when one train is heading at the location where the second train is occupying the same railway track. Absolute permissive block and interlocking towers were the solutions provided at different times to restrict the trains from banging into each other while sharing the same railroad. Problem that arises with the automatic railways traffic control is intertrain conflict that share the same bi-directional railroad. The dispatcher offices are interconnected to each other. In the territories where the railroads are being share by trains coming from opposite directions, there are chances that the train engineers unaware or not timely informed with the offensive moment of the head-on train on the same track. As a result trains would encounter each other head on, and while most trains would end up approaching one another at restricted speed. Trains might get a yellow signal of e.g. “Go”, which assumes that the next block is unoccupied when in fact there is traffic approaching. Mechanism should be introduced to make the communication between the dispatcher office and the train engines safely and smoothly allow the railway traffic without any accident.

**Problem in Existing System:** In the above plotted scenario, the communication between the train and the dispatcher office takes place through signaling.

- Signal dropping
- Longer wait time.
- Waste of energy in applying brakes
  - Energy required for restarting the train engine

**3. Conceptual Model for Proposed Railway Traffic Management using Mobile Agents System.** In the proposed system, mobile agents can be used for the messaging between the train engines and the dispatcher offices to communicate the instructions. Code in the agent dynamic code part would be meant for instructing the train engines about reducing the speed of the train. It will contain the actual speed limit and location of the upcoming train. The dispatcher offices connected to the GPS system locates the current route of the train. Dispatcher offices are in continuous connectivity with the GPS, monitoring the locomotion of the trains.

In case there is a train approaching on the railroad that is already occupied or is the next route of another train, the dispatcher office will send a message-carrying-mobile agent to the conflicting trains, giving the instruction of speed reduction. Mobile agent will carry the speed limit and marshal at the train engine, in response the train engines will have to send an acknowledgement for confirmation of message reception. After that the train speed if reduced would be detected by the Mobile agent system implemented in the train, thus will send another acknowledgment to the dispatcher. Exceptionally, if due to some network issues the message is not properly communicated and the dispatcher office doesn’t receive the acknowledgement then that railroad is considered to be broken or damaged and traffic is halted towards that territory. In this case the dispatcher would send message-carrying-mobile agent to other train approaching the same bi-directional road with instructions of “Stop” or
change of route. An algorithm that supposes share railroads and trains as processors and tasks can be used for scheduling the trains onto the conflicting railroads. Concern of this paper is using mobile agents as the message carrying entity in the proposed system.

3.1. **Figure: Railway Traffic Management using Mobile Agents Technology:** The figure given below depicts a rough sketch of railway traffic control using mobile agent’s technology. There is an agent owner i.e. dispatcher’s office in the given system, and hosts i.e. train engineer. GPS is connected to the hosts and the agent owner so as to keep in synchronization with the location and motion of the trains using the wireless communication medium. It is obvious from the diagram that the dispatcher office sends instructions to the train, in return of which the train would be sending an acknowledgement of the message being received.

![Proposed system Conceptual Model Diagram “Railway Traffic Management Using Mobile Agent Technology”](image)

Dispatcher office continuously monitors the route and scheduling of the trains under its territory. The schedule and route of the trains along a certain territory is predefined and updated regularly in case of any change in the dispatcher’s office. In the predefined schedule and route, there is information about the bidirectional railway line that is going to have traffic in a given frame of time. The density of the traffic i.e. the traffic conditions and occupancy of a certain railroad is also determined from the given route and schedule timetable. Now, the scheduling confirms for the arrival and departure of the trains at a station and not for the route (especially bidirectional railroads). Bi-directional railroads if not monitored and controlled properly might results in irrecoverable damages.

3.2. **Real world Scenario:** In a territory, two scheduled trains Train A and Train B have to cross a bi-directional railroad at the same time in opposite directions. Both the trains leave their stations at scheduled time and move in the predefined speed configured in their systems. There is a time period when both the trains have to share a single railroad. Now, the dispatcher office monitoring the train locations and movement are well aware of the fact that if both the trains kept on moving with the same speed, they can head on to each other any time. This moment the dispatcher office will send signals to the train A to “stop” at a location some let’s suppose 20 miles away from the intersection point. While at the same time, send a message to the train B to increase the speed and move in the same direction. The train A responds to the signal from the dispatcher office by applying brakes and stopping at the determined location. The train B will respond to the signal from the dispatcher by increasing the speed and moving ahead. The train A waits at some remote location X for the next command from dispatcher office. The dispatcher office keeps track of the train B to pass and swap the railroad after crossing the intersection point. Once the shared railroad is not occupied anymore, the dispatcher office sends a command to the train A to start its motion again towards its predefined destination.
4. Architectural Model for Railway Traffic Management System using Mobile Agents. In our proposed framework mobile agents technology is introduced into the centralized traffic control systems for smooth and uninterrupted communication of the messages between the dispatcher office and the train engineers. This gives a viable solution for the issue that arises from conflict over a bi-directional shared railroad due to untimely communication. We use mobile agents as the message carrying entity between the dispatcher office and the train. Mobile Agent systems installed at the dispatcher’s office and the train will provide the execution environment for the mobile agents. It is made sure that mobile agents move in a trusted environment and that their security is not compromised. Security for the safe migration of the agents between the agencies that is agent platform at dispatcher office and train is incorporated in the proposed framework.

Architecture of the proposed system comprises of Agent Platforms installed at Dispatcher’s Offices and Train engines.

4.1 Host Location Store : At the Dispatcher’s Office GPS device installed populates raw data into the Host Locations Store from the GPS Satellite.

4.2 Agency

Agent Platform : It is the main agent platform installed at both the dispatcher’s offices as well as the trains engines. It contains the basic components required for managing, creation, authentication, persistence of the agents. These components are as following

![Diagram of Agent Platform at Dispatcher Office](https://via.placeholder.com/150)

Figure 4.1 Architectural view of “Railway Traffic Management System using Mobile Agent” : The Agent Platform at Dispatcher Office

4.2.1 Agent Manager : This component create, registers, authenticate, migrate the mobile agent. It manages the whole life cycle of the mobile agent. An agent must be registered at this component otherwise it cannot migrate within an agent platform or other agencies.

4.2.2 Security: Security component which ensures defense against malicious host attacks or malicious agent attacks. It contains the basic security policies for the mobile agents and agent platform. An incoming mobile agent or out going mobile agent must pass through here to either communicate with the other components within the platform or going outside the Agent platform.

4.2.3 Agent Communication Channel: This component is an intermediary step between the internal entities communication of the agent platform and the world outside agency. Once an agent passed the security barrier, it can communicate with the agent execution environment only after going through the Agent Communication channel.

4.2.4 Agent Execution Environment: Agents after passing through all the intermediate components finally reaches the Agent Execution Environment which provides the platform for running the agents code. It is
here when mobile agents code is executed and decides intelligently what and when to communicate with the system. The architecture of the proposed system applies for both the dispatcher’s offices and trains with only difference that the GPS device installed at the trains would send its coordinates to the GPS satellite and will not store any location information unlike dispatcher’s office.

5. Discussion. As described earlier, dispatcher’s office acts as mobile agent owner. Hosts are the trains moving along the predefined territory monitored by the dispatcher office. Wireless communication takes place between the GPs, hosts and the owner. By the time, two scheduled trains are approaching in not less than 20 minutes on a bi-directional railroad; some mathematical calculation is done at the dispatcher’s office. Here the dispatcher would send an agent at the host location in this case train A for example to reduce the speed at a certain level. And at the same time send an agent to train B (host) informing about the increase in speed command. Train A lowers its speed and doesn’t stop thus, saving energy that would be wasted in case it stopped and restarted. Meanwhile train B increases its speed as directed and move fast across the point of intersection on the bidirectional railroad. Once the train B has moved away from the shared railroad, the dispatcher office receives an acknowledgement of the obeyed command.

Now, a command carrying mobile agent is send to the train A which is now moving with a lower speed, to increase the speed of the train to a certain level to cross the point of conflict before more traffic heads to that location. Acknowledgements are sent back to the dispatcher’s office (agent Owner) after every mobile agent received at the host. Mobile agent system is connected to the hardware i.e. wheels or axels. As soon as the speed of the train is reduced, the mobile agent dynamic code is automatically modified, which now carries the confirmation code that the command has been fulfilled.
If not send due to some network traffic or other dead links, the dispatcher office assumes that railroad to be damaged and out of order. It can send rescue team at the required location to check what went wrong. At the same time the dispatcher sends message to other dispatcher’s offices connected to it to halt the traffic that is heading towards that conflicted region.

**Step1.** Mathematical calculation is done for the time that train A and train B will take in approaching the Point of intersection.

**Step1a.** Their location is identified from the communication with the GPS system, which would give a numeric value of the distance between the two trains.

**Step1b.** Now, the speed with which both the trains are moving is determined by calculating simple \( V=S/t \) formula.

**Step1c.** With the above calculated speed and location, the system would be able to now calculate in what time would e.g. train A reach the intersection point. Time for reaching an intersecting point X is calculated by following formula

\[
\text{Suppose, Location of train A}= AX
\]

\[
\text{Now, we have the distance and the speed with which we can calculate the time that train A would take to reach the intersection point } X.
\]

\[
\text{Distance between } AX \text{ and } X \text{ is given by}
\]

\[
\text{Then Difference of location of train A and intersection Point } X
\]

\[
i.e. \ AX - X = Y
\]

**Step1d.** Same is repeated for the train B.

**Step2.** Real-time calculation of the time that each train would take to reach at a conflict time is now used to calculate the new speed for each of the train. For example, train A is at a location whose difference with the intersecting point is greater than that worked out for train B. In this case the train A is sent a mobile agent carrying the command to limit the speed to a given value. Here it will decrease the speed.

**Step3.** Train A must send an acknowledgment to the dispatcher office confirming that message has been received and command is obeyed.

**Step4.** At the same time the train B receives a mobile agent that carries the command for increasing the speed e.g. doubles the speed. This will make the train B move faster to cross the conflicting region before the arrival of train A.

**Step5.** Train B also sends an acknowledgment to the dispatcher’s office confirming that message has been received and command has been acted upon.

**Step6.** Acknowledgment is important for the dispatcher’s office (agent owner) so as to keep track of trains and their speed and to avoid any incident resulting from conflict over a shared railroad.

**Step7.** When the train B acknowledgment (mobile agent carrying the acknowledgment) is received which guarantees its locomotion across the intersection point X, the dispatcher’s office sends a mobile agent that carries the command for train A. This command will instruct the train A, to get back to the original speed or increase its speed to double for crossing the same point X before the rising of a new conflict.

It is to be noted that this formal model can be applied to any particular interlocking after a further analysis. This is because we have modeled the system and defined the properties based on the requirements of like a real system.

6. **Conclusion.** This article presents a mobile agent based system for the detection and control of the railway traffic. It gives an overview of the present industrial applications of the mobile agents along with the countermeasures taken so far to handle the security issues of the mobile agent systems. Studies have shown that mobile agents are mostly used in the complex systems where systems are geographically distributed. High traffic control systems and centralized traffic control systems for Trains are being overviewed in the paper. The problems of the message communication between the stations and the trains can be hazardous in situation like sharing a bidirectional railway line. Using mobile agents for the communication between the dispatcher offices and the trains can result in smooth and safe routing and scheduling of the trains. The architecture of the proposed system is made in compliance with the FIPA standardized components and communication language ACL.

**Future Work**

This paper skips the simulation of the proposed system which can prove the correctness and completeness of
the Railway Traffic Management Using Mobile Agents. The system completion and validation can be done in formal methods. The specification of the system can be done using Z/EVES or any other formal language. And simulation generated can be a future research topic. Also the architecture of the given system can be reviewed for further component addition and improvement of the present components.

REFERENCES


