AN AUGMENTED REALITY DEMO ENVIRONMENT FOR INTELLIGENT TRANSPORTATION SYSTEMS

Pearce, Hammond
University of Auckland
38 Princes Street, Auckland, 1011
hpea485@aucklanduni.ac.nz
+64 27 352 3897

Sinha, Roopak
Auckland University of Technology
cnr Rutland and Wakefield Streets
rsinha@aut.ac.nz
+64 9 921 9999

Hikmet, Mahmood
University of Auckland
38 Princes Street, Auckland, 1011
mhik003@aucklanduni.ac.nz
+64 21 065 9167

Roop, Partha
University of Auckland
38 Princes Street, Auckland, 1011
p.roop@auckland.ac.nz
+64 9 373 7599 extn: 85583

ABSTRACT

Co-Operative Intelligent transportation systems (C-ITS) incorporate distributed control and coordination to ensure better and safer transportation networks, reduced carbon emissions, and improved traffic flows. In particular, ITS may benefit greatly from the synergy of technologies for traffic simulation, real-time control, and communication networks.

In this paper, we develop an augmented reality demo environment for ITS by combining traffic micro-simulation with a physical scaled intersection model containing toy vehicles controlled by embedded controllers. We use the framework to develop an application illustrating the use of vehicle-to-vehicle (V2V) communications between vehicles to facilitate safe driving. Vehicles maintain appropriate distance and use variable acceleration and smart lane-changing to ensure safety. We simulated V2V communications between autonomous vehicles by modifying the on-board electronics on the vehicles and by using the XBee wireless standard for communication.

1 INTRODUCTION

As wireless networks become safer and more robust, they find increasing use in next-generation co-operative intelligent transportation systems (C-ITSs). C-ITSs are aimed at creating safer, faster, and energy efficient vehicular movement on existing infrastructure using vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) wireless communications [1]. Wireless vehicular communications standards such as WAVE/802.11p [2] have now become the backbone of radical C-ITSs such as Virtual Traffic Lights (VTL) [3] and VTL+ [4] where vehicles use V2V and V2I communications to dynamically negotiate traffic plans at intersections. Such ITSs have been shown to have the potential to significantly decrease greenhouse gas emissions while increasing traffic flows [3].

Currently, micro-simulators like Aimsun [5] and SUMO [6] are used for extensively testing ITSs before deployment. Unfortunately, while they provide excellent models of vehicles, traffic and road networks, most micro-simulators do not provide adequate modelling for V2V and V2I communications. While some studies [7] have used micro-simulators in conjunction with network simulators like NS-3, a comprehensive approach for ensuring the safety of ITS calls for testing ITSs using real V2V/V2I communications before deployment.

In this paper, we propose an augmented reality demonstration environment (ARDE) for testing ITSs in which scaled embedded models of vehicles, traffic signals, and V2V/V2I communications are used in tandem with a micro-simulator. The micro-simulator brings several benefits, such as varying traffic scenarios, different vehicle types, and road network models. On the other hand, embedded models provide an avenue for observing and measuring the performance of V2V/V2I enabled traffic more realistically. Specifically, we use the SUMO micro-simulator [6] and use suitably modified toy vehicles, custom-built LED panels, and XBee [8] controllers suitable for modelling vehicles, traffic signals, and wireless communications. However, our ARDE framework allows using other alternative embedded models.

In addition, we use a *model-driven engineering* (MDE) [9] process to design the control software for the embedded devices. MDE is used to model and analyse safety-critical control systems such as aircraft software [10]. In our setting, we use the BlokIDE [11] software to visually design the control software for the various embedded devices used in ARDE. MDE allows us to refine and reuse software blocks in a visual manner while low-level program code is generated automatically.

The main contributions of this article are:

1. An extensible augmented reality demo environment (ARDE) for rigorously testing ITSs.
2. A model-driven engineering (MDE) framework to program embedded devices for ARDE and a framework to observe and capture real-time data on embedded devices during simulation.
3. The creation of suitably modified, scaled-down embedded models of vehicles, traffic signals and V2V/V2I communications.

The rest of this paper is organised as follows. Section**Error! Reference source not found.** 2 describes in detail the various parts of the ARDE. Section 3 provides initial results, while concluding remarks and future directions appear in section 4.

2 METHODOLOGY

2.0 OVERALL STRATEGY

This project aimed to bridge the gap between the virtual and the physical worlds of traffic simulation; simulated traffic environments found in software, and physical autonomous vehicles beginning to be found around the world. By integrating the ideas of both, an Intelligent Transportation System (ITS) Augmented Reality Demonstration Environment (ARDE) can be constructed allowing for a demonstration of the interaction between those intelligent vehicles found on the roads and the more common ordinary vehicle simply controlled by the driver.

For the demonstration, a single four-lane intersection, with the aim of allowing future development in scaling the environment to achieve larger and more complex systems as needed, would be built. The architecture to achieve this goal was composed of a number of elements: *physical* components like robotic, semi-autonomous vehicles and traffic signals, and *software* components for recognising and controlling the physical devices, and for the co-simulation of the augmented reality set up. Physical vehicles and simulation software interacted using bidirectional wireless personal area network (PAN), and a camera-based visual recognition algorithm was used to monitor the physical vehicles.

Figure 1 describes the various parts of the framework, which we discuss in the following sub-sections. The various blocks of the figure are labelled with the sub-section numbers in which they are described.

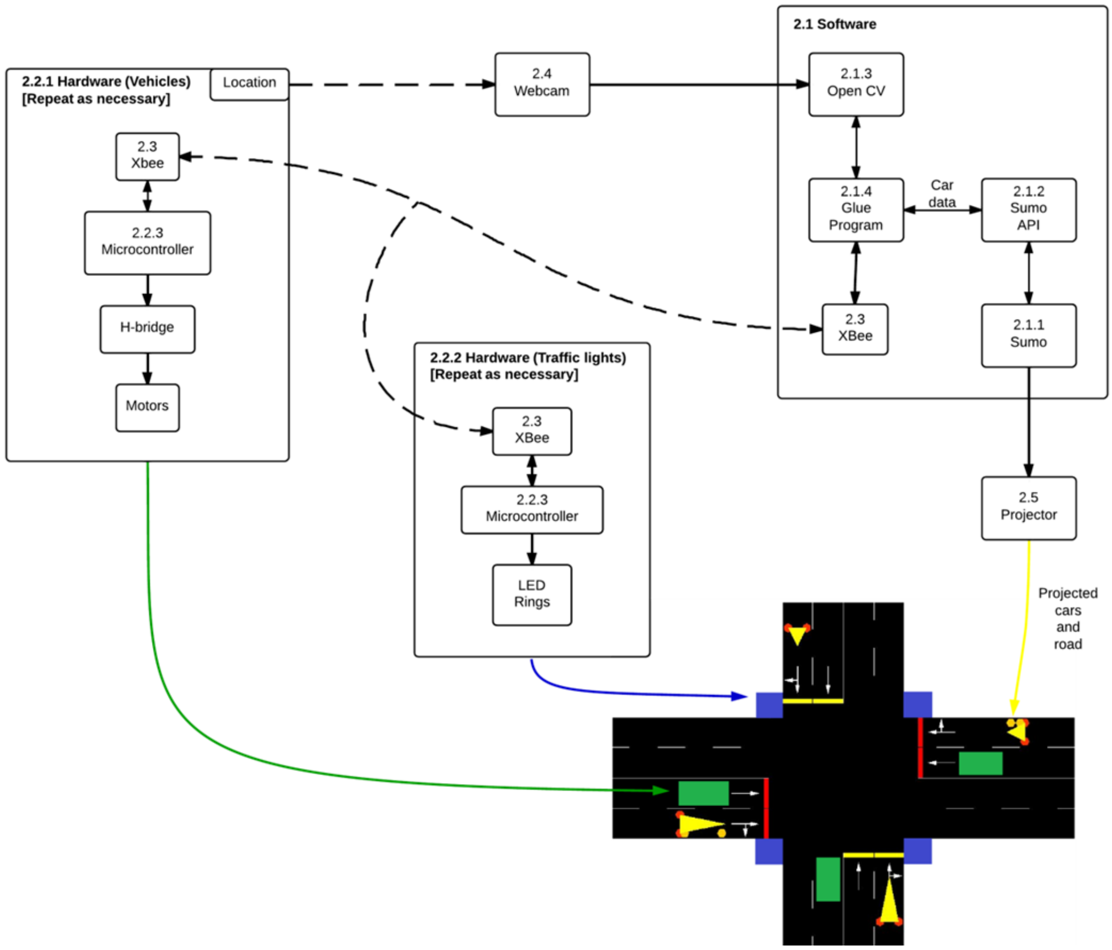

Figure 1: ITS Demo Environment System Diagram and Physical Setup

2.1 SOFTWARE

2.1.1 SUMO

We chose the open source SUMO [6] to control the simulation. SUMO is an open-source 2D road traffic simulation package with a strong following in the research and academic community [12, 13]. SUMO has many desirable features, such as a programmable interface (section 2.1.1), robust models of vehicles, fast execution time, and an easy interface to build road networks that make it a suitable choice for this application. However, it does not have a programmable network model preventing us from realistically simulating the performance of ITSs that use specific V2V communication protocols.

2.1.2 SUMO API

The SUMO API is a flexible way of controlling, analysing, and examining the simulation environment using external software. It allows for external applications (in this project, this was the "Glue program", see Section 2.1.3) to connect to and update the location of the vehicles in SUMO, as well as extract information about individuals vehicles during simulation. The API is critical for ARDE as it allows for the instructions for the movement of the physical semi-autonomous vehicles to come directly from the traffic simulator itself.

2.1.3 OpenCV

For the ARDE, we decided to eschew traditional electronic sensory apparatus on the vehicles (such as infrared/ultrasonic distance meters, accelerometers, and speedometers) for a single camera (See Figure 2 and Section 2.4), positioned above the intersection (described in Section 2.4) which can provide images of the environment for processing by suitable tools.

Figure 2 Roof-mounted Webcam and Projector

OpenCV (Open Source Computer Vision Library) [14] is a popular open-source computer vision and machine learning software library. In our setting, OpenCV reads the data via a web camera mounted above the intersection, and calculates the position, speed, and angle of physical vehicles. This information is fed back to the control software of the vehicles (for correcting their motion) and to SUMO. This provides pseudo-GPS functionality to the vehicles.

The *template matching* feature of OpenCV allows us to place identifiable markers (in this case, a basic but unique pattern) on the roof of each vehicle, allowing the software to individually track each vehicle. This is extremely important in this application as accurate tracking of each autonomous vehicle is paramount to their successful participation in the simulation. An example of the output from the image processing when using the positioning pattern from the vehicles is depicted in Figure 3.

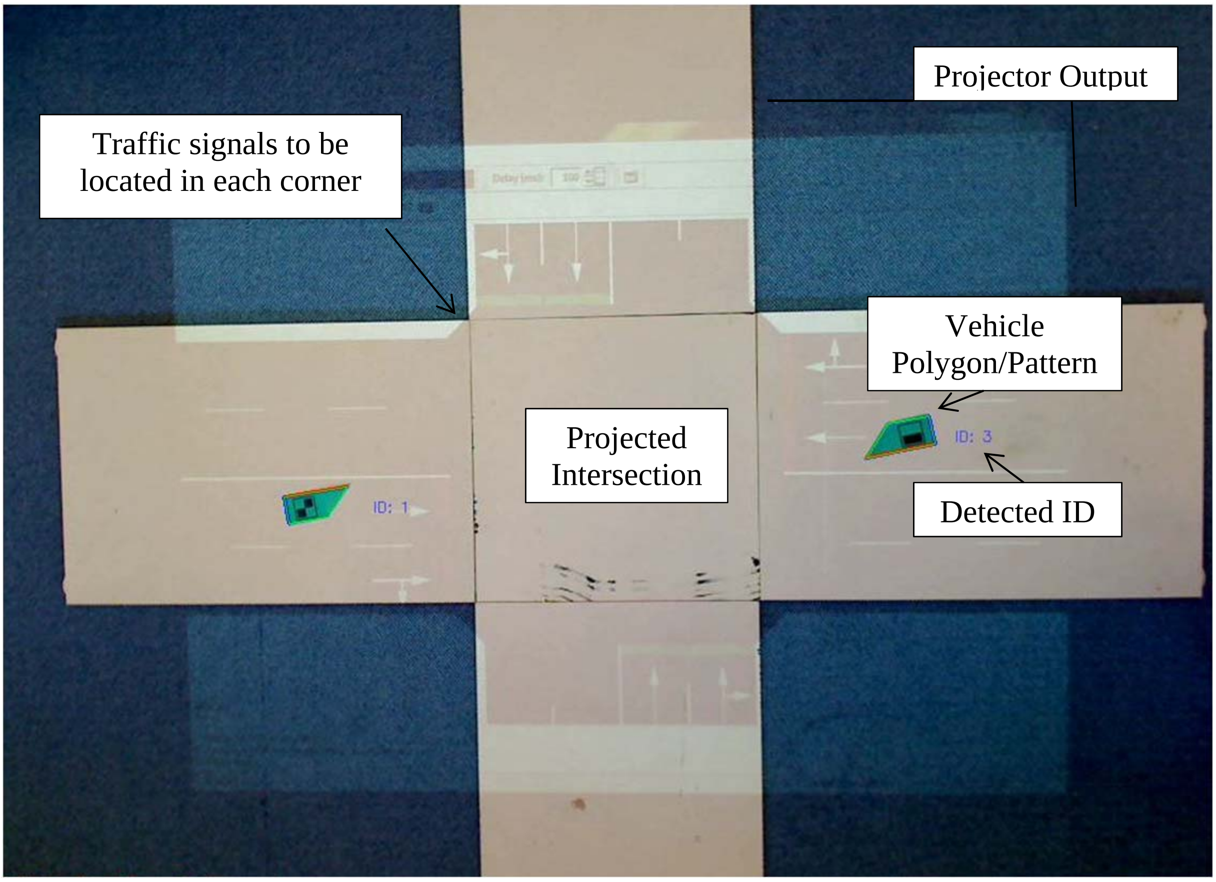

Figure 3 OpenCV Image Processing of the ARDE

2.1.4 The "Glue" program

As the name suggests, the "Glue" program connects the various parts of the ARDE. It acts as the middleware for the entire software system, and allows the flow of control instructions and data (vehicle positions) between SUMO and the physical devices (vehicles and traffic signals). The glue program uses an OpenCV image library to read and process the web camera images, and issues instructions to vehicles via the XBee network (see Section 2.3), and connects to the SUMO API to read commands for the physical devices from SUMO.

In each simulation step, a large number of tasks need to take place. These tasks are repeated 10 times per second in the following order:

1. Webcam captures a frame of the environment.
2. OpenCV processes the frame and determines orientation and location of each physical vehicle in the view.
3. This data is then used to update the physical vehicle locations in SUMO.
4. SUMO performs a simulation step.
5. The physical traffic signals are updated to reflect the virtual ones.
6. The physical vehicles are instructed on what they need to do (e.g. drive forward, backward, turn left) to get to the new/current SUMO coordinates for them.

2.2 HARDWARE, TRAFFIC SIGNALS AND VEHICLES

The physical setup of this demo environment (visible in Figure 3) consists of five large squares of plywood (the "road") arranged in a "plus" formation with a roof-mounted projector and camera aimed at them. The projector displays the lanes and virtual vehicles on these squares of plywood, and the camera reads back the positions of the “real” vehicles, described in section 2.2.0. Also arranged around these wooden squares are four traffic signals, which are controlled wirelessly over the same XBee network as the vehicles.

The hardware consists of the traffic signals and the semi-autonomous vehicles. The project is intended to be highly portable and easy to set up, so wireless connectivity and self-powered units were an obvious choice for both the lights and the vehicles.

Both the lights and the vehicles have very similar embedded control systems as shown in Figure 1. Each consists of a wireless XBee module, a PIC microcontroller, a battery, and components interfaced to the general purpose input/output pins of the microcontroller.

2.2.0 Semi-autonomous Vehicles
For this application several remote-controlled toy vehicles were acquired and then modified, disabling/removing their existing circuitry and control logic and replacing it with our own PIC microcontroller and XBee module, as shown in Figure 4.

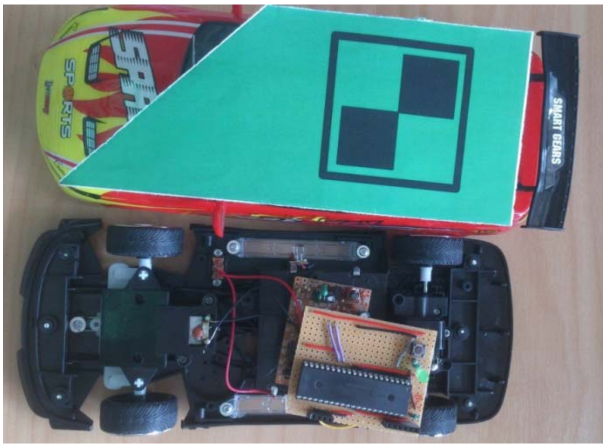

Figure 4 Modified RC vehicle with polygon and pattern

These vehicles do not determine their own routing decisions, but instead receive instructions to move wirelessly from SUMO. They receive all of their locational data from the OpenCV image library as described in section 2.1.1.

To allow for simpler recognition and tracking of each vehicle, a green polygon with a unique pattern is placed on each roof . OpenCV is capable of identifying the orientation and location of these shapes. The determined position and orientation is then sent to each respective vehicle.

2.2.1 Traffic Signals
The traffic signals, depicted in Figure 5, consist of a circuit board made up of an industry-standard PIC microcontroller [15], a wireless XBee module, a 9 volt battery, and components suitable for driving the rings of LED lights that make up the traffic signal.

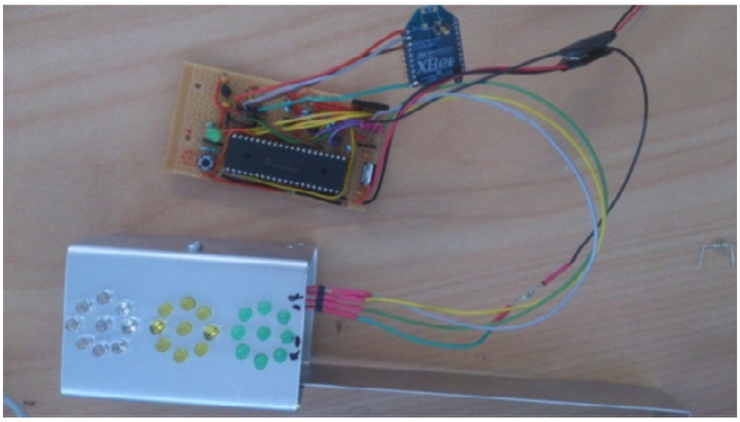

Figure 5 Traffic Signal (Showing Internals)

The traffic signals receive a command about which state to enter (Red, Green, or Yellow) via the XBee interface, transition to that state, and then acknowledge completion by sending another Xigbee signal back to the sender.

2.2.2 Microcontroller software

The software running on the PIC micro-controller was created using a model-driven approach (MDE) in the BlokIDE software. BlokIDE allows system-level design using a visual block-oriented language. Designers can focus on higher-level design while lower-level code is generated automatically by the BlokIDE compiler. We compile the BlokIDE program for the light and vehicle controllers to C code. The C code was then deployed on the PIC micro-controllers that control the physical lights and vehicles.

Figure 6 BlokIDE to PIC process

2.3 XBEE COMMUNICATION

The ARDE achieves seamless interaction between the physical components and the SUMO micro-simulator using XBee wireless communications. We use appropriate network modules on the computer that runs SUMO, and each of the individual physical components. A wireless personal area network (PAN) based on the 802.15.4 standard allows these modules to communicate using point-to-point messages.

2.4 WEBCAM

As described in Section 2.1.2, the web camera (Figure 2) captures bird-eye images of the physical environment. These images are then processed by the OpenCV library to ascertain vehicle positions. This simplistic yet elegant approach does not require any additional sensors, keeping the set up easy to maintain and upgrade. However, a single-sensor set up also means that the webcam resolution affects the size of the physical layout. A very wide layout will require a very high resolution camera to accurately track vehicles.

2.5 PROJECTOR

As shown in Figures 2 and 3, a roof-mounted projector displays the virtual vehicles and the road network (from SUMO) onto the physical layout. A single projector, like a single webcam, restricts the size of the physical set up due to limited resolution.

3 RESULTS

The benefits of this project are not directly quantifiable, as the scope of the project was not to perform experiments ourselves; rather it was to create an ARDE which allows for other experimentation to take place. This completed ARDE is visible in Figure 7, and as designed, it is a capable tool for ITS experimentation. For example, different V2V and V2I communication set-ups (separate to those presented) can be modelled and monitored. The embedded controllers on the physical vehicles can monitor and compute total travel time, delays, and SUMO can estimate green-house gas emissions. The “glue” program can be tailored to provide information about how many messages are sent per unit of time (e.g. per minute) in order to predict network congestion thresholds. The traffic signal controllers could separately compute average cycle times and phase splits on-board. The ARDE provides a capable, scalable test-bed for use in testing ITS systems, as the developed ARDE framework is easy to extend with more physical vehicles and intersections, requiring very little modifications to the OpenCV library to sense the positions of the added vehicles.

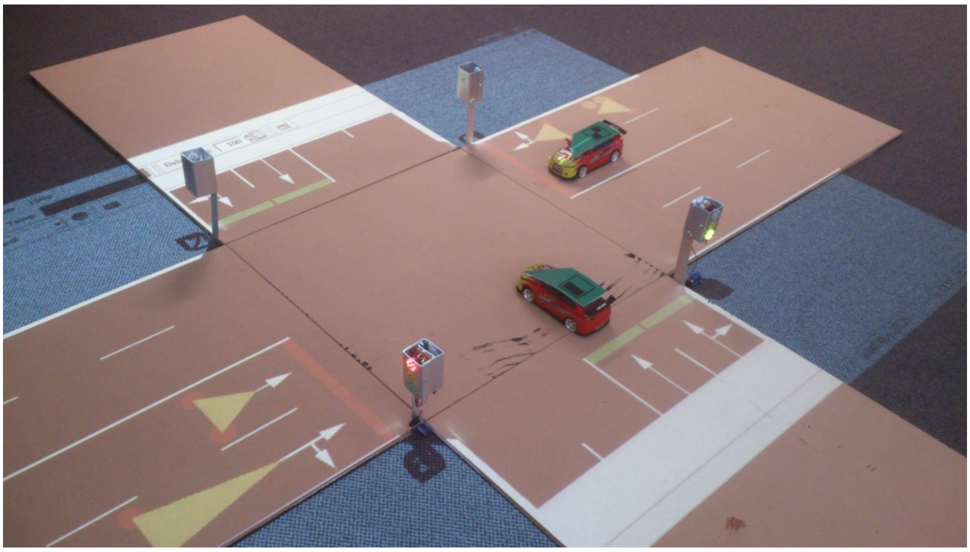

Figure 7 Complete ARDE setup

4 CONCLUSIONS AND FUTURE DIRECTIONS

This paper presents an augmented reality demonstration environment (ARDE) for intelligent transportation systems (ITS). The ARDE is geared towards simulating the behaviour of ITS requiring wireless V2V and V2I communications. This is achieved by using a micro-simulator (SUMO) along with physical components like semi-autonomous vehicles and traffic signals interconnected using embedded wireless modules. The ARDE allows observations of the behaviour of scaled vehicle models in a cost-effective set up. In addition, the physical embedded devices can monitor and compute key parameters on-board, thereby enriching the simulation and reducing the computation load on the micro-simulator.

The ARDE is an extensible environment. We envisage adding more sensors like scaled induction loops to ARDE for testing and modelling existing traffic control algorithms such as SCATS [16]. Similarly, we can add other sensors and different vehicle types to create a richer augmented-reality demonstration environment.

REFERENCES

[1] P. Fernandes and U. Nunes, "Vehicle communications: A short survey," *IADIS Telecommunications, Networks and Systems, Lisboa, Portugal,* 2007.

[2] D. Jiang and L. Delgrossi, "IEEE 802.11 p: Towards an international standard for wireless access in vehicular environments," in *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE,* 2008, pp. 2036-2040.

[3] M. Ferreira and P. M. d'Orey, "On the impact of virtual traffic lights on carbon emissions mitigation," *Intelligent Transportation Systems, IEEE Transactions on,* vol. 13, pp. 284-295, 2012.

[4] R. Sinha, P. S. Roop, and P. Ranjitkar, "Virtual Traffic Lights+ (VTL+): A Robust, Practical, and Functionally-Safe Intelligent Transportation System," in *Transportation Research Board 92nd Annual Meeting*, 2013.

[5] J. Barceló and J. Casas, "Dynamic network simulation with AIMSUN," *Simulation Approaches in Transportation Analysis,* pp. 57-98, 2005.

[6] D. Krajzewicz, M. Bonert, and P. Wagner, "The open source traffic simulation package SUMO," *RoboCup 2006 Infrastructure Simulation Competition,* vol. 1, pp. 1-5, 2006.

[7] C. Sommer, Z. Yao, R. German, and F. Dressler, "On the need for bidirectional coupling of road traffic microsimulation and network simulation," in *Proceedings of the 1st ACM SIGMOBILE workshop on Mobility models,* 2008, pp. 41-48.

[8] Z. Alliance, "XBee specification," *Document 053474r06, Version,* vol. 1, 2006.

[9] S. Kent, "Model driven engineering," in *Integrated Formal Methods,* 2002, pp. 286-298.

[10] R. France and B. Rumpe, "Model-driven development of complex software: A research roadmap," in *2007 Future of Software Engineering,* 2007, pp. 37-54.

[11] K. Nicholas, Z. E. Bhatti, and P. S. Roop, "Model-driven development of industrial embedded systems: Challenges faced and lessons learnt," in *Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference on*, 2012, pp. 1-4.

[12] D. Krajzewicz, E. Brockfeld, J. Mikat, J. Ringel, C. Rössel, W. Tuchscheerer, P. Wagner, and R. Wösler, "Simulation of modern traffic lights control systems using the open source traffic simulation SUMO," in *Proceedings of the 3rd industrial simulation conference*, 2005.

[13] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *Mobile Computing, IEEE Transactions on,* vol. 10, pp. 3-15, 2011.

[14] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*: O'reilly, 2008.

[15] J. Iovine, *PIC microcontroller project book*: McGraw-Hill, Inc., 2004.

[16] P. Lowrie, "SCATS, Sydney Co-Ordinated Adaptive Traffic System: A Traffic Responsive Method of Controlling Urban Traffic," 1990.