CENTER FOR CONNECTED AND AUTOMATED TRANSPORTATION

UMTRI-2023-4
March 2023

Mcity Augmented Reality Testing Environment Development – Phase II

Dr. Henry Liu
Dr. Yiheng Feng

UMTRI

CENTER FOR CONNECTED AND AUTOMATED TRANSPORTATION

Report No. UMTRI 2023-4
Project Start Date: October, 2017
Project End Date: March, 2021

April 2019

Mcity Augmented Reality Testing Environment Development – Phase II

By
Henry Liu, Professor
Yiheng Feng, Assistant Research Scientist
University of Michigan

CENTER FOR CONNECTED AND AUTOMATED TRANSPORTATION

DISCLAIMER

Funding for this research was provided by the Center for Connected and Automated Transportation under Grant No. 69A3551747105 of the U.S. Department of Transportation, Office of the Assistant Secretary for Research and Technology (OST-R), University Transportation Centers Program. The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

Suggested APA Format Citation:

Feng, Y., Bao, S., & Liu, H.X. (2019). Connected and Automated Vehicle (CAV) Testing Scenario Design and Implementation Using Naturalistic Driving Data and Augmented Reality. Final Report.
UMTRI-2023-4.
DOI: 10.7302/7016

Contacts

For more information:

Dr. Henry X. Liu
University of Michigan
2350 Hayward, Ann Arbor, MI, 48109
Phone: (734) 647-4796
Email: henryliu@umich.edu

Dr. Yiheng Feng
University of Michigan
2901 Baxter Rd, Ann Arbor, MI, 48109
Phone: (734) 936-1052
Email: yhfeng@umich.edu

Center for Connected and Automated Transportation
University of Michigan Transportation Research Institute
2901 Baxter Road
Ann Arbor, MI 48152
umtri-ccat@umich.edu
ccat.umtri.umich.edu
(734) 763-2498

CENTER FOR CONNECTED AND AUTOMATED TRANSPORTATION

Technical Report Documentation Page

<table>
<tr><td>1. Report No.
UMTRI-2023-4</td><td>2. Government Accession No.</td><td>3. Recipient's Catalog No.</td></tr>
<tr><td colspan="2" rowspan="2">4. Title and Subtitle
Mcity Augmented Reality Testing Environment Development – Phase II
DOI: 10.7302/7016</td><td>5. Report Date
April 2019</td></tr>
<tr><td>6. Performing Organization Code</td></tr>
<tr><td colspan="2">7. Author(s)
Feng, Yiheng, Ph.D., https://orcid.org/0000-0001-5656-3222
Liu, Henry, Ph.D., https://orcid.org/0000-0002-3685-9920</td><td>8. Performing Organization Report No.</td></tr>
<tr><td colspan="2" rowspan="2">9. Performing Organization Name and Address
UMTRI
2901 Baxter Road
Ann Arbor, MI 48109</td><td>10. Work Unit No.</td></tr>
<tr><td>11. Contract or Grant No.
Contract No. 69A3551747105</td></tr>
<tr><td colspan="2" rowspan="2">12. Sponsoring Agency Name and Address
Center for Connected and Automated Transportation
University of Michigan Transportation Research Institute
2901 Baxter Road
Ann Arbor, MI 48109</td><td>13. Type of Report and Period Covered
Final Report
October 2017 – March 2019</td></tr>
<tr><td>14. Sponsoring Agency Code</td></tr>
<tr><td colspan="3">15. Supplementary Notes
Conducted under the U.S. DOT Office of the Assistant Secretary for Research and Technology's (OST-R) University Transportation Centers (UTC) program.</td></tr>
<tr><td colspan="3">16. Abstract
An augmented reality (AR) testing environment for connected and automated vehicle (CAV) testing and evaluation has been developed and used for different demos and projects since 2017. In order to fully integrate the AR testing environment with Mcity test facility, we further upgrade the system to be compatible with the new vehicle development and communication standard (i.e., SAE J2735), including new functionalities, designing new testing scenarios, and supporting Mcity's ABC CAV testing procedures.
Mcity has installed four new RSUs which are compatible with 2016 revision of the standard, we upgraded the AR system to the 2016 standard. Both the encoder and decoder of BSM and SPaT messages need to be upgraded, due to the different encoding rule (i.e., UPER) and data structure applied in the 2016 standard.
A scenario library was built to facilitate Mcity's ABC CAV testing procedure. The library now has 6 scenarios including basic interactions at intersections, red-light violations, interactions with virtual train, and newly developed highway cursing, car following, roundabout mering, and signal priority for the CAV.
The previous version of AR system only supports a fixed route of the CAV. In this project the newly surveyed high-definition map by CAV was used to rebuild the Mcity road network in the simulation so that the CAV supported flexible routing.
Finally, the upgraded AR system was integrated with Mcity API and became one of the services that Mcity can provide to the customers. After the integration, the third parties were able to start, stop the AR system and check the running status through Mcity's API. A new demo video includes all scenarios with a 3D view of traffic simulator, human machine interface (HMI) in the CAV, inside view of the CAV, front view of the CAV from the dash camera was recorded and created.</td></tr>
<tr><td>17. Key Words
Connected vehicles, automated vehicles, augmented reality</td><td colspan="2">18. Distribution Statement
No restrictions.</td></tr>
<tr><td>19. Security Classif. (of this report)
Unclassified</td><td>20. Security Classif. (of this page)
Unclassified</td><td>21. No. of Pages
XXXXXXX

22. Price</td></tr>
</table>

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

UMTRI | The University of Akron | ILLINOIS CENTER FOR TRANSPORTATION | PURDUE UNIVERSITY | Washtenaw Community College | Central State University 1887

Table of Contents

Project Summary

An augmented reality (AR) testing environment for connected and automated vehicle (CAV) testing and evaluation has been developed and used for different demos and projects since 2017. In order to fully integrate the AR testing environment with Mcity test facility, we further upgrade the system to be compatible with the new vehicle development and communication standard (i.e., SAE J2735), including new functionalities, designing new testing scenarios, and supporting Mcity's ABC CAV testing procedures.

Mcity has installed four new RSUs which are compatible with 2016 revision of the standard, we upgraded the AR system to the 2016 standard. Both the encoder and decoder of BSM and SPaT messages need to be upgraded, due to the different encoding rule (i.e., UPER) and data structure applied in the 2016 standard.

A scenario library was built to facilitate Mcity's ABC CAV testing procedure. The library now has 6 scenarios including basic interactions at intersections, red-light violations, interactions with virtual train, and newly developed highway cursing, car following, roundabout mering, and signal priority for the CAV.

The previous version of AR system only supports a fixed route of the CAV. In this project the newly surveyed high-definition map by CAV was used to rebuild the Mcity road network in the simulation so that the CAV supported flexible routing.

Finally, the upgraded AR system was integrated with Mcity API and became one of the services that Mcity can provide to the customers. After the integration, the third parties were able to start, stop the AR system and check the running status through Mcity's API. A new demo video includes all scenarios with a 3D view of traffic simulator, human machine interface (HMI) in the CAV, inside view of the CAV, front view of the CAV from the dash camera was recorded and created.

1. Introduction

An augmented reality (AR) testing environment for connected and automated vehicle (CAV) testing and evaluation has been developed with the support of USDOT Center for Connected and Automated Transportation (CCAT) and has been demonstrated in Mcity since 2017. This environment combines the real-world testing facility (Mcity) with a simulation platform which generates virtual vehicles as background traffic to help design different CAV testing scenarios. The information of virtual traffic is encoded into Basic Safety Messages (BSMs) and broadcasts to testing CAVs by the Road Side Units (RSUs) through the Dedicated Short Range Communication (DSRC) network. Meanwhile, BSMs generated from testing CAVs are also transmitted through the RSU to the simulation platform. As a result, the testing CAVs can interact with virtual vehicle as if they exist on the roadway in real-time. Moreover, all traffic signals in the real-world and simulation platform are synchronized and broadcast through Signal Phasing and Timing (SpaT) messages, so that both virtual and real vehicles can react to the same signal indications. Two initial testing scenarios have been designed (red light running and railway crossing) which shows that the new platform is able to provide a safe, cost-effective testing environment.

This platform has been used for different demos and projects since 2017. The objective of the Phase 2 development is to fully integrate the AR testing environment with Mcity test facility, further upgrade the system to be compatible with the new vehicle development and communication standard (i.e., SAE J2735), include new functionalities, design new testing scenarios, and support Mcity's ABC CAV testing procedures.

2. Communication Protocol Upgrade

The current AR platform still adopts SAE J2735-2009 version of standard for DSRC messages. With the four new RSUs Mcity has installed which are compatible with 2016 revision of the standard, in this project, we upgraded the AR system to support the 2016 standard. The upgrade consists of two parts: the MKZ vehicle and the AR platform. Both the encoder and decoder of BSM and SPaT messages need to be upgraded, due to the different encoding rule (i.e., UPER) and data structure applied in the 2016 standard.
The DSRC messages are defined in ASN.1 format, which is an interface description language for defining data structure that can be serialized and de-serialized in a standard. To interpret the ASN.1 definition into corresponding programming languages, an open source ASN.1 complier called ASNAC is used (https://github.com/vlm/asn1c). The ASN.1 compiler is a tool for converting the specifications in ASN.1 notation into some other language such as C or C++. The compiler creates codes which allows automatic serialization and deserialization of these data structures using several standardized encoding rules. The generated C++ compatible C source code can be used to serialize the native C structures into compact and unambiguous BER/OER/PER/XER-based data files, and de-serialize the files back. **Error! Reference source not found.** shows the flow chart of the compiler application.

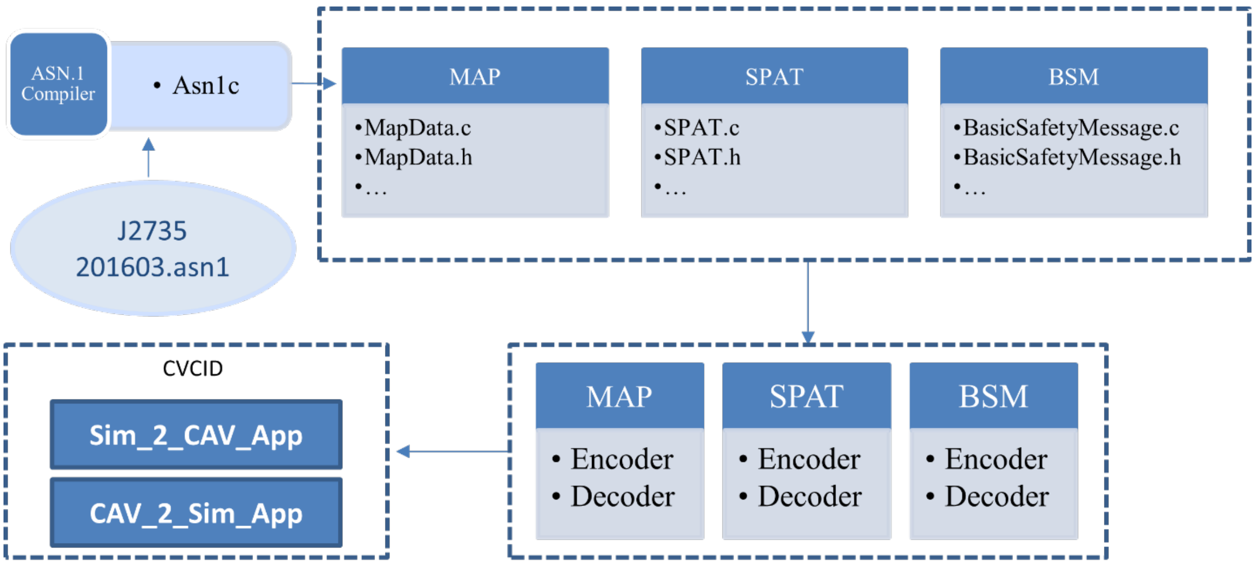

Figure 1 Flow Chart of the Compiler Application

First, the ASN.1 compiler needs to be installed in a Linux operating system (e.g., Ubuntu). It converts the J2735_201603.ans1 file, which includes all data elements defined in the J2735 standard, to the corresponding header and source files of each data structure in C/C++ as shown in **Error! Reference source not found.**.

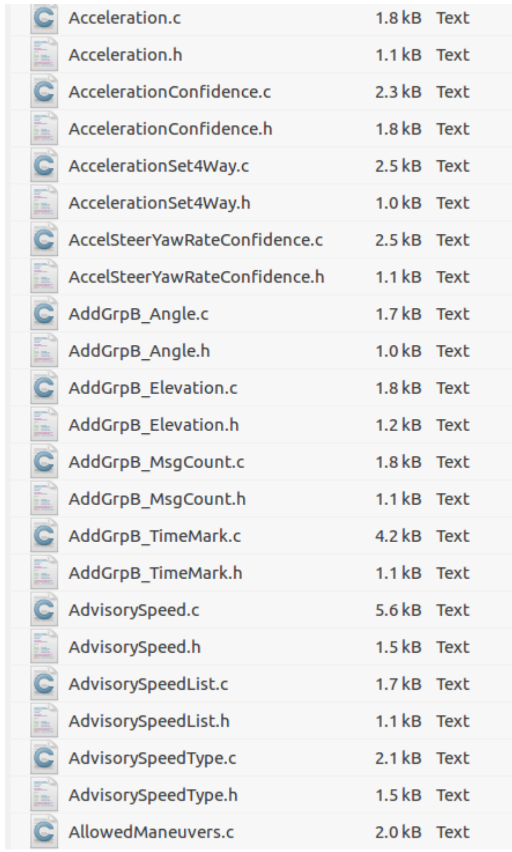

Figure 2 Generated C/C++ source and header files

We generate different projects for different types of messages. For testing purposes, fake data are generated and populated into the data structure. Then the encoder and decoder functions are called to see whether the data remain consistent before and after the encoding and decoding process. **Error! Reference source not found.** shows an example of the encoded hex buffer and decoded message content from a BSM message.

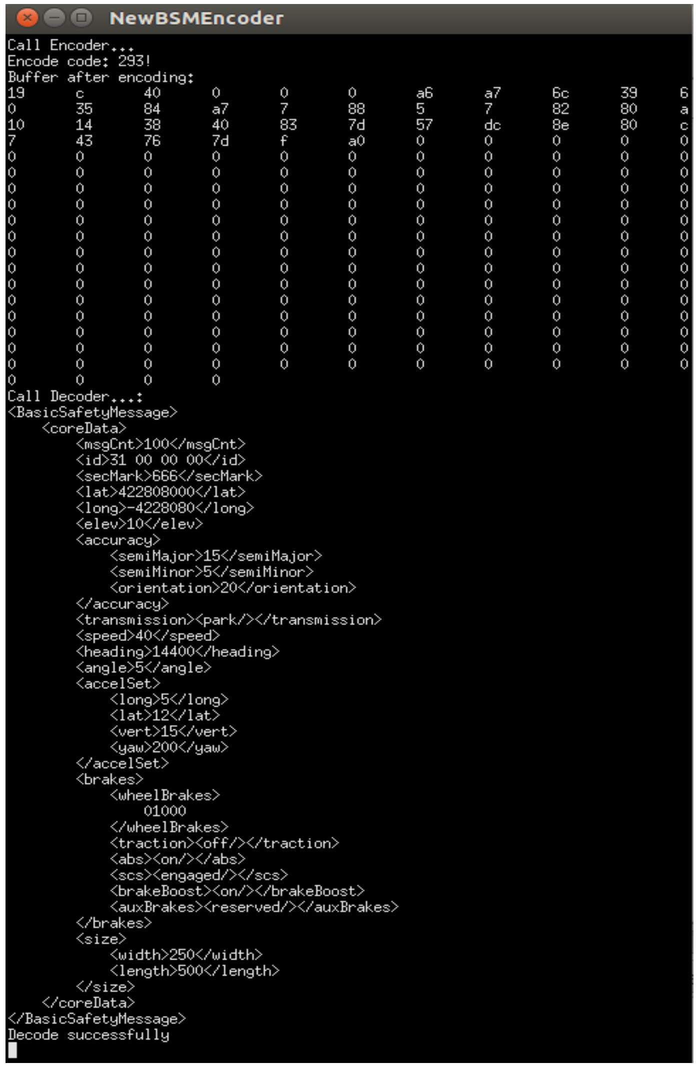

Figure 3 The encoding and decoding process.

Finally, to validate whether the encoding/decoding tool is interoperable with other industrial companies, two validation tests are conducted:

An online decoding tool provided by Marben (https://www.marben-products.com/decoder-asn1-automotive/) is used to validate whether our encoded hex string is correct.

We called decoding function after encoding process and compare the original value of each data element with the value parsed from the decoding process.

After the validation tests, the J2735-201603 encoder/decoder were integrated in the applications that running in the CVCID box and CAV to replace the original J2735-2009 encoder/decoder.

3. Flexible Routing and Traffic Signal Priority

Flexible routing

The previous version of AR system only supports a fixed route of the CAV. In this project, we used the newly surveyed high-definition map by CAV to rebuild the Mcity road network in the simulation to support flexible routing of CAV. A high-definition map of the testing facility for both testing CAVs and the AR environment was used offline to build the simulated road network. In order to synchronize the vehicle's location in the real world with that in the simulation precisely, the HD map running in the CAV should also be matched with HD map in simulation. The HD map of the testing facility includes information about road segments, lanes and connectors.

A road segment consists of one or multiple lanes from the same travel direction. A lane has a starting point, an end point, and several intermediate points. The end point of an upstream lane is connected to a downstream lane via a connector, which can also consist of multiple way points. Connectors between segments are used to model possible turning movements, merging, and diverging in the road geometry. The detail information of each object of the HD map is listed below.

Road segment: road segment ID, number of lanes.

Lane: Road segment ID, Lane ID, lane width, pairs of waypoints including latitude and longitude that can describe the roadway geometric. Resolution: at 10^{-8} m.

Connector: Connector ID, connecting road segment and lane IDs, waypoint ID and its latitude and longitude for the staring points and ending points of the connector.

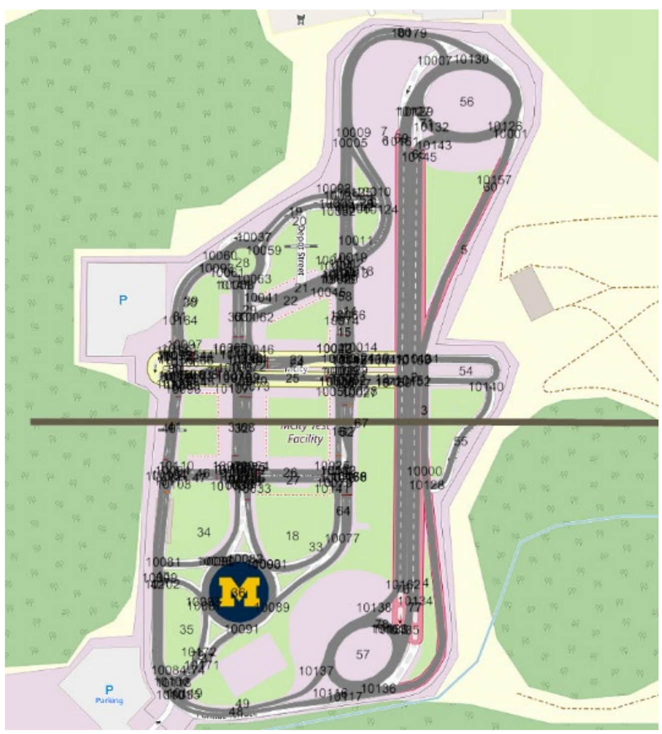

Figure 4 HD MAP in Vissim simulation

```xml
<?xml version="1.0" encoding="utf-8"?>
<mapping>
    &lt;segment id="1"&gt;
        <lane id="1" linkID="1" laneIndex="1"/>
        <lane id="2" linkID="1" laneIndex="2"/>
        <lane id="3" linkID="2" laneIndex="2"/>
        <lane id="4" linkID="2" laneIndex="1"/>
        <lane id="5" linkID="3" laneIndex="1"/>
    &lt;/segment&gt;
    &lt;segment id="2"&gt;
        <lane id="1" linkID="4" laneIndex="1"/>
    &lt;/segment&gt;

    &lt;segment id="3"&gt;
        <lane id="1" linkID="5" laneIndex="1"/>
    &lt;/segment&gt;

    &lt;segment id="4"&gt;
        <lane id="1" linkID="6" laneIndex="1"/>
        <lane id="2" linkID="7" laneIndex="1"/>
    &lt;/segment&gt;
    &lt;segment id="5"&gt;
        <lane id="1" linkID="8" laneIndex="1"/>
        <lane id="2" linkID="9" laneIndex="1"/>
    &lt;/segment&gt;
    &lt;segment id="6"&gt;
        <lane id="1" linkID="10" laneIndex="1"/>
        <lane id="2" linkID="11" laneIndex="1"/>
        <lane id="3" linkID="12" laneIndex="1"/>
        <lane id="4" linkID="13" laneIndex="1"/>
    &lt;/segment&gt;
    &lt;segment id="7"&gt;
        <lane id="1" linkID="14" laneIndex="1"/>
        <lane id="2" linkID="58" laneIndex="1"/>
        <lane id="3" linkID="15" laneIndex="1"/>
        <lane id="4" linkID="59" laneIndex="1"/>
    &lt;/segment&gt;
    &lt;segment id="8"&gt;
        <lane id="1" linkID="16" laneIndex="1"/>
        <lane id="2" linkID="62" laneIndex="1"/>
        <lane id="3" linkID="17" laneIndex="1"/>
    &lt;/segment&gt;
    &lt;segment id="9"&gt;
        <lane id="1" linkID="18" laneIndex="1"/>
    &lt;/segment&gt;
    &lt;segment id="10"&gt;
        <lane id="1" linkID="19" laneIndex="1"/>
```


Figure 5 Transition Table of the HD Map

Figure 6 HD MAP for CAV

A transition table is used to convert the real CAV route information to that in simulation with a different data structure to support flexible routing. Meanwhile, the method of synchronizing CAV in simulation and communication protocol was upgraded to include more information such as past, current and future link, and lane information to support this new feature. The traffic assignment was modified to create more interactions between the testing CAV and the virtual vehicles (e.g., highway merge, roundabout merge, lane change).

Traffic Signal priority

Signal priority for the CAV was added into the AR system as a new demo, which involves control of Mcity's infrastructure. Communication protocols between testing CAV and RSUs was modified to include a priority request field. A signal priority application at the infrastructure side was written to control the traffic signal, when the testing CAV sends a priority request to a certain traffic signal phase. The signal priority function for the six intersections in the downtown area was implemented.

The process of the signal priority for the CAV can be categorized into 4 main steps as shown in **Error! Reference source not found.** In the first step, when the CAV approached an intersection and want to pass the intersection with priority, the signal priority request will be sent. The signal priority request that includes the desired intersection id and phase id is contained in the field of data accuracy in BSM. Then, after the RSU received the message sent from the OBU, it will be immediately forwarded to the CVCID. In the second step, the CVCID was instantly requesting the SPaT information from the controller. With the infrastructure information and vehicle information, in the third step, the CVCID will generate an optimal signal control plan which guarantees the shortest delay for the CAV based on the signal settings of the intersection (e.g., maximal and minimal green time for each phase), SPaT information (current phase and timing), and the vehicle state (e.g., the estimated time of arrival of the vehicle calculated based on the speed and position). The control strategy which might include a combination of forcing-off, extending, and omitting of different phases. Then in the last step, the CVCID will send the control command to the traffic signal controller to implement the new plan accordingly. Moreover, the system will continuously monitor the status of the CAV and update the signal control plan when needed, i.e., the priority of the CAV will be maintained until it passes the intersection, after which the signal control plan will return to its normal setting.

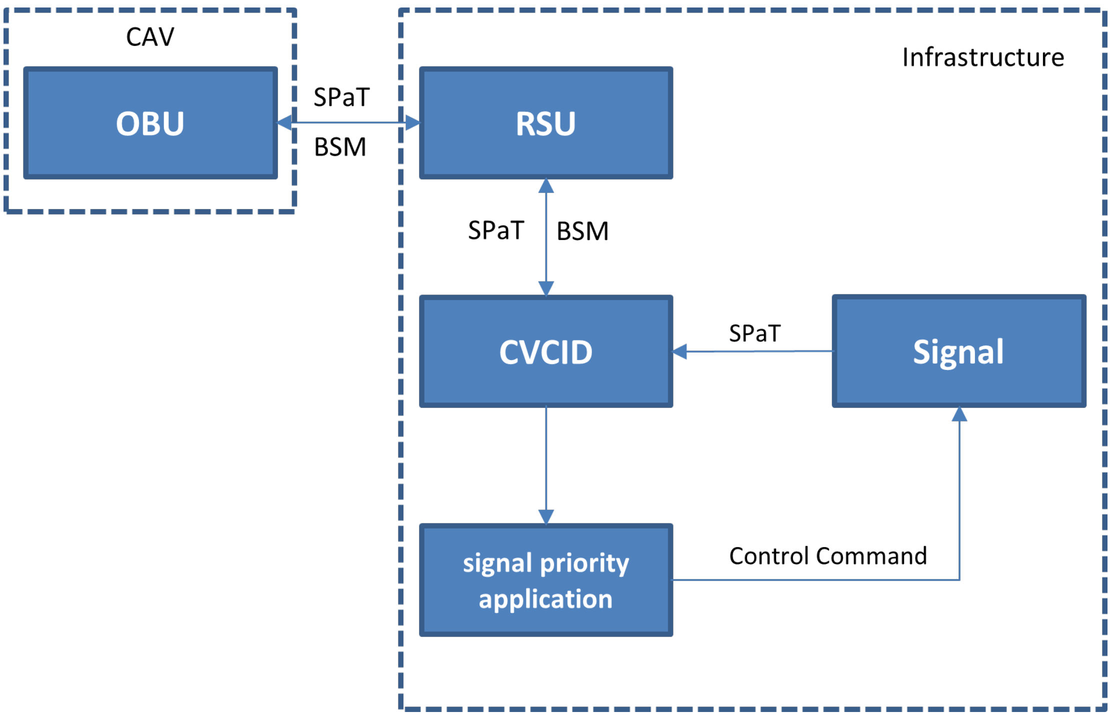

Figure 7 Signal Priority Architecture

4. Design and Implement New Testing Scenarios

A scenario library was built to facilitate Mcity's ABC CAV testing procedure. A few examples from the behavior competence scenarios were selected based on Mcity's interests and created in the AR platform, especially the ones that requires interactions with other vehicles. The library now has 6 scenarios including 3 old scenarios and 3 new designed scenarios. The old ones are basic interactions at intersections, red-light violations, interactions with virtual train and 3 new ones are highway cursing and car following, roundabout mering, signal priority for the CAV.

Highway cursing and car following:

The CAV can maintain high speed cruising along the highway of Mcity. The virtual CAV will be synchronized precisely in the simulation. When the CAV enters the highway, a relative lower speed platoon will be generated at the highway in the simulation. When the CAV approaches the platoon, the CAV will slow down and performs the car-following.

Signal priority for the CAV

When the CAV approaches to an intersection and encounter a red light in front. The CAV can send a signal priority request to pass the intersection with obtaining the green phase. The control strategy might include a combination of forcing-off, extending, and omitting of different phases based on the signal settings of the intersection (e.g., maximal and minimal green time for each phase), SPaT information (current phase and timing), and the vehicle state (e.g., the estimated time of arrival of the vehicle calculated based on the speed and position). The system will continuously monitor the status of the CAV and update the signal control plan when needed, i.e., the priority of the CAV will be maintained until it passes the intersection, after which the signal control plan will return to its normal setting.

Roundabout mering

Roundabout mering is one of the most challenging tasks for the CAV. In this scenario, a constant traffic was generated in the simulation at the roundabout for the CAV to perform the cut in. The CAV will pick an appropriate gap to cut in to avoid the conflicts with the other vehicles in the roundabout. When the CAV leaves the exit at the roundabout, a traffic flow will be generated at the same entrance to wait as well until there is an appropriate gap for them to merge into the roundabout.

5. Integrate with Mcity's API

Mcity has developed an API that provides a user interface for all services to the customers. The upgraded AR system was integrated with the Mcity API and became one of the services. After the integration, Mcity's API was able to start, stop the AR system and check the running status.

The system architecture consists of a server (CVCID) which synchronize the traffic between the simulation and the real world. The new AR environment including the traffic simulator, and new testing scenarios is called ARMcity. It was deployed in a local server at Mcity. The ARMcity control software is stopped/started by a WSGI server (waitress) running a Flask application. The flask application has several endpoints for launching, terminating, and restarting the simulation. The web server is hosted on port 8443 and the API is accessible by going to /apidocs/ Comunication with the port is open to all, but the user must have a valid key to perform any action. The server has been installed as a Service on the local machine, it is launched on startup. The project is checked into a GitHub.

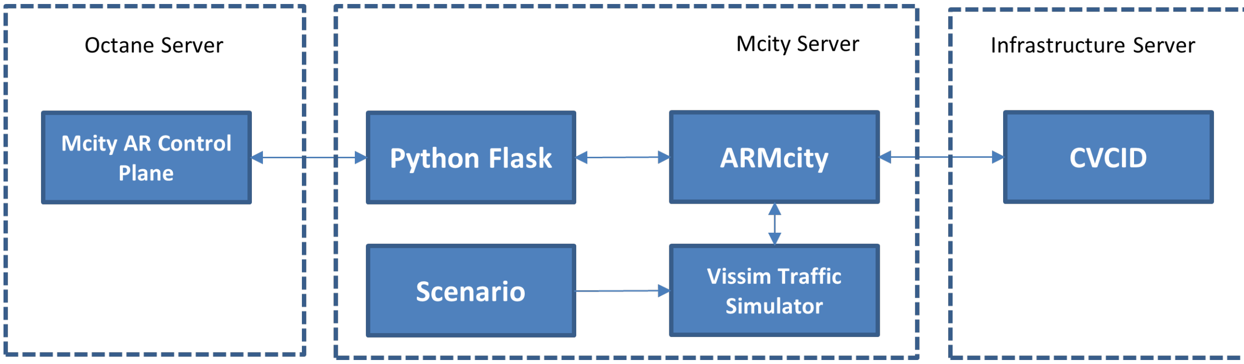

Figure 8 Architecture of integration with Mcity API

6. Demo Video

A demo video was also recorded and created. It includes all scenarios with a 3D view of traffic simulator, human machine interface (HMI) in the CAV, inside view of the CAV, front view of the CAV from the dash camera. All windows of each view are synchronized. The video can be viewed from here https://www.youtube.com/watch?v=dIvJ66Jak4g.

7. Outcomes and Impacts

In this project, we further developed the AR testing environment so that it is fully integrated with Mcity test facility. We also further upgrade the system to be compatible with the new vehicle development and communication standard (i.e., SAE J2735), include new functionalities such as flexible routing and signal priority for the CAV. We also designed new testing scenarios, and support Mcity's ABC CAV testing procedures. Finally, the upgraded AR system was integrated with the Mcity API and became one of the services. After the integration, the third testing parties can start, stop the AR system through the Mcity's API. A demo video was also recorded and created which includes all scenarios with a 3D view of traffic simulator, human machine interface (HMI) in the CAV, inside view of the CAV, front view of the CAV from the dash camera.