Mcity OS:
New Cloud-Based Tools Simplify Complex and Repeatable Automated Vehicle Testing

TYLER WORMAN
Manager, Software Engineering, Mcity

ED SERZO
Software Engineer, Mcity

GREG MCGUIRE
Associate Director, Mcity

Contents

INTRODUCTION

New software and hardware developed by a team of engineers at the University of Michigan's Mcity automated vehicle research center now make it possible for researchers to create and execute complex, sophisticated, and easily repeatable tests for connected vehicles, automated vehicles, and connected and automated vehicles, potentially saving testing time and costs, and accelerating product development.

The new Mcity OS is a cloud-based operating system that runs on any internet-enabled device, including vehicle computer platforms, laptops, tablets, and smartphones to control all the features of the Mcity Test Facility. Mcity OS gives users point-and-click control over vehicle interactions at intersections, train crossings, crosswalks, and other test facility features. Using a simple visible programming language, Mcity OS makes it possible to drag and drop instructions to control all testing features and create easily repeatable multi-step test scenarios. Mcity OS can be integrated at other test facilities and in real-world environments as well. The American Center for Mobility in Ypsilanti, Michigan, is the first facility to license Mcity OS through the University of Michigan's Office of Technology Transfer. Mcity OS also provides researchers access to connected vehicle and intersection data in Ann Arbor, Michigan.

The software was developed by Tyler Worman, software engineering manager for Mcity, and Greg McGuire, associate director of Mcity. Supporting hardware was built by Ed Serzo, a software engineer for Mcity. Mcity OS also helps bridge the gap between software and hardware simulation and real-world testing by providing a consistent, simple interface, repeatable test scenarios, and data collection to enable fully virtual control as well as augmented reality, blending both the real and simulated environments.

"Vehicles of the future are going to be magnitudes more capable and complex in their behaviors than vehicles today," McGuire said. "Test facilities have to come along with them in terms of their capabilities. You can't have a dumb test facility testing a smart car."

DEFINING THE CHALLENGE

The demand for autonomous vehicles to be thoroughly tested can be seen in the development of Mcity's test track in Ann Arbor. The Mcity Test Facility includes dozens of traffic lights, blind curves, traffic circles, a railroad crossing, an underpass, a construction zone, four different road surfaces, urban and rural environments, plus robotic cyclists, pedestrians, and deer. Creating a scenario that tests how a vehicle reacts and performs in even just a few of these combined driving situations was once a nearly impossible task.

There simply was no way to work with existing system controls so that researchers could integrate all the many elements of the facility into their tests. When a more complicated test scenario was needed, engineers would be brought in to specifically design and code individual pieces of software, an expensive and time-consuming process. Synchronizing existing capabilities of the test facility with customized test scenarios needed a lot of setup and were very hard to change, adapt or update.

Consider what was needed when researchers wanted to test a relatively simple and very common driving situation in an autonomous vehicle – approaching a yellow traffic signal.

"Before, in really simple tests involving one piece of infrastructure, like a traffic signal, all the timing would have been done manually by someone with a radio," Worman explained. "In that case, you're hoping the car comes in at the right speed, you're hoping someone presses the button at the right time, and you're hoping that the traffic light gets to the same phase at the right time. And all that data would have had to have been recorded manually, or you wouldn't know if you really did the test right or how you could repeat it."

In a more complicated test involving just three interactions, the testing would have been incredibly difficult.

"I don't know how you would have executed a multi-part test with any type of reliability," Worman added. "Take a situation where you want the car to come into an intersection at 40 mph, then you want to roll the pedestrian robot out, change the traffic light, and trigger the crosswalk warning. You would need to have a lot of people who were all watching a stopwatch to try and get the timing right."

The result was that researchers using the Mcity Test Facility could not take advantage of all the infrastructure in place. Those who were using segments of the infrastructure either had a background in using that specific piece of hardware, or Mcity staff specifically trained them on it. The ability to use all the test track hardware at once would require a team with vast knowledge of each piece in place on the track. Mcity staff realized that for the test facility to truly support users and deliver all of its potential value, researchers would need to be able to easily access all the Mcity infrastructure, with the shortest learning curve possible.

A PROPOSED SOLUTION

Because the concept of creating easily repeatable tests aligned with Mcity's research goals, the effort to create a software solution began. Within five months, software was developed that controlled the railroad gate and all of the test facility's traffic lights. During the past two-and-a-half years, the developers have added more components and capabilities to the operating system—eventually supporting more than 100 different types of interactions—and will continue to add new features, attributes and capabilities.

Mcity OS is made up of several components, including Octane, a software interface, and Skyline, a web application. Octane allows users to catalog, query, control, and collect data from devices such as traffic signals, rail crossings, crosswalks, cameras, and other visual sensors, as well as lighting, robotic test platforms and more. Octane easily integrates with many common traffic control devices and simplifies working with complicated devices, such as traffic controllers. While Mcity OS is available by license, the specification for the interface is open, making it easily extendable in order to quickly build support for existing custom hardware. The open interface also makes it easy to incorporate support for Mcity OS in third party test tools, such as those by Foretellix and dRisk.

Skyline uses Octane to provide real-time updates as well as point-and-click control of the features of the test facility, including intersections, rail crossings, crosswalks, and gates (Figure 1). Skyline also collects data from infrastructure sensors, records weather conditions and real-time kinematic (RTK) positioning to precisely record test vehicle position.

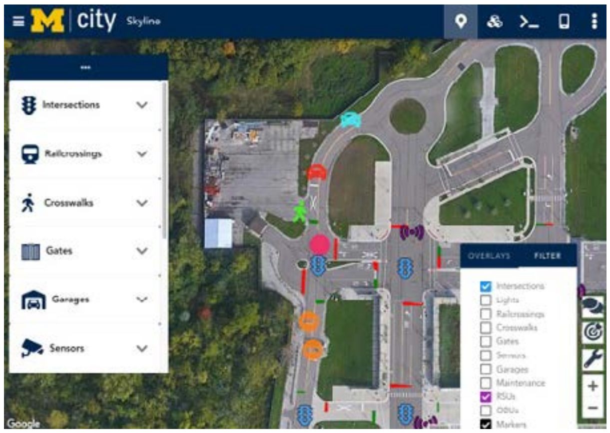

Figure 1: Mcity OS Skyline app

Skyline gives users the ability to build test scenarios using a simple visible programming language to drag and drop individual segments of code containing instructions and controls (Figure 2). Those modules act as building blocks that can be combined to create large, complex, and easily repeatable test scenarios, while also providing real-time data collection from sensors. Test scenarios can be triggered by any monitored condition or event, including trip wires, vehicle position, timing, the press of a crosswalk button, or even through other scenarios. Once entered, new variations of a test can be quickly performed, recorded and compared.

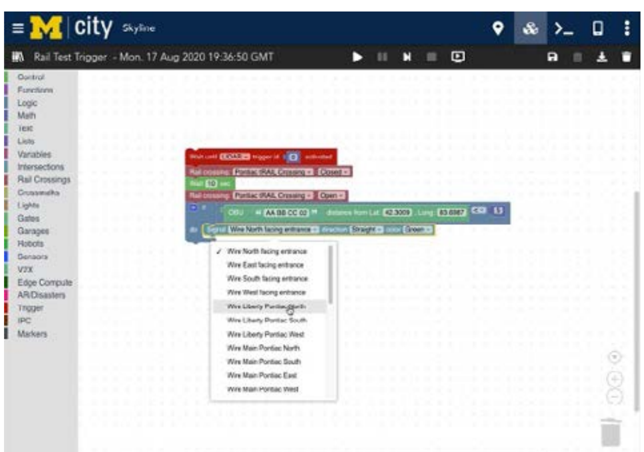

Figure 2: Tool to build test scenarios in Mcity OS Skyline app.

Beyond the software, Mcity engineers also created inexpensive ($5-$50) hardware that can be used to integrate non-standard devices or infrastructure into Octane, so that those devices can be controlled, monitored and triggered using Mcity OS.

“The hardware Mcity has built comes pre-integrated into Mcity OS,” Serzo said. “It is designed to be simple. One of our goals is to abstract out the detail required for test setup, so that individuals using Mcity OS can focus on developing repeatable tests.”

Mcity OS is fully integrated with the 16.5-acre Mcity Test Facility in Ann Arbor, Michigan, to fully synchronize each piece of infrastructure, putting Mcity well ahead of other testing sites that are just beginning to add controllable infrastructure and haven’t reached the point where it all needs to be synchronized. While some manufacturers are building systems that integrate their suite of hardware with software, they often lack the ability to synchronize with other systems and infrastructures (Figure 3).

Figure 3: The Mcity Test Facility at the University of Michigan

Mcity OS was designed with deployment to other test facilities and smart cities in mind. It has already been licensed by The American Center for Mobility in Ypsilanti, Michigan, and by several companies. Test scenarios and data collected by Mcity OS are easily downloaded and can be shared with other users to promote collaboration and development. The software is also in place to compile data from the Ann Arbor Connected Environment, which involves streaming data to Mcity from 70 intersections on the streets of Ann Arbor, and about 2,000 vehicles or more.

Beyond simulation and testing, Mcity OS offers the ability to trigger data collection at specific points and based on specific events. The tools were designed to be deployed in the real world where systems can be tasked to record naturalistic data, such as how people move through streets and intersections. In this way, Mcity OS can form the framework for mobility research and test development under different sets of safety guidelines.

Because most new development of vehicle technology is likely to begin with software simulation, Mcity OS is fully capable of running simulated scenarios as well as controlling real test track features. This gives developers the opportunity to create, edit, and refine tests away from testing facilities, then run those scenarios at Mcity or other locations where Mcity OS is available to check and validate the simulated tests and data. Mcity developers project that the bulk of autonomous vehicle testing will need to be handled in simulation, with test facilities providing a backstop and feedback loop.

CONCLUSION

Mcity OS solves the problem of how researchers can create the kinds of intricate interactions that advanced vehicles—connected, automated, and both connected and automated—will need to safely negotiate public roads. Mcity OS also reports all data from each test, which can include not only the details of the test and the vehicle response, but also lighting, weather, and other test variables. Advanced test scenarios created with Mcity OS can be accurately repeated an endless number of times with split-second precision, while also allowing researchers to update or change scenarios on the fly using nothing more than a smartphone.

The end result is better, more accurate vehicle testing that costs less to develop and requires much less time to create. This is a crucial factor in developing Level 3 automated vehicles, where the car or truck can take over all driving functions under certain limited circumstances, and Level 4 vehicles, which operate with no input at all from the driver under select conditions and in certain defined locations, such as a ride-hailing or public transit application in an urban center.

The immediate result of deploying Mcity OS is that it expands the capabilities of the Mcity Test Facility, opening the test track and its capabilities to a much wider audience of vehicle developers, researchers, and engineers.

"Now you don't need to have a mechanical engineering degree just to run a test," McGuire said.

Worman noted that, “When we have people who are testing a vehicle, they don’t have to be software developers anymore. Testing can now be available to everyone.”

LOOKING AHEAD

Mcity OS is just the latest innovation produced by Mcity to expand and accelerate testing for connected vehicles, automated vehicles, and connected and automated vehicles while reducing the time and expense of conducting tests. Before consumers will embrace automated vehicles—especially cars with no driver controls at all—the people who will buy and ride in these “vehicles of the future,” or share the road with them, will need assurance that the vehicles are reliable and safe.

Mcity OS is part of the ongoing development of the Mcity Test Facility which, since 2015, has evolved to make all major transportation infrastructure connected and controllable, along with its digital twin versions, offering faster and less expensive test verification and validation.

Going forward, Mcity OS developers plan to expand Mcity OS to handle an increasing number of variables, and integrate a great deal of hardware to add more triggers, including additional types of robots and sensors, to provide more realistic environments for testing.

“The key thing we’re excited about is that the vast, vast majority of tests are going to be run in some kind of simulation,” McGuire said. “Using Mcity OS, users can take the results of their tests at Mcity, go back into their simulation and recreate everything. Then they can come back when they’ve made changes and be sure that cycle is the same every time. If you can’t set the test up at Mcity to be the same as last time, then you won’t have confidence the software changes you made are working.

“I think this technique will become crucial for rapid, safe development by industry, and it fits squarely within Mcity’s mission.”

RESOURCES

Mcity OS Demonstration Video

First license for AV testing software developed by Mcity goes to American Center for Mobility

Mcity offers cloud C/AV solution to ACM (ITS International)

Open-source Octane specification

Sample scripts/tutorials for working with Octane without using scenario blocks

About Mcity

Mcity at the University of Michigan is leading the transition to connected and automated vehicles. Home to world-renowned researchers, a one-of-a-kind test facility, and on-road deployments, Mcity brings together industry, government, and academia to improve transportation safety, sustainability, and accessibility for the benefit of society.