ASTra - API for Simulation of Traffic

Rémi Domingues
Département informatique, INSA de Lyon, Villeurbanne, France
and Lero@UCD, School of Computer Science and informatics Dublin, Ireland
remidomingues@live.fr

Anthony Labaere
École Centrale de Nantes, France
and Lero@UCD, School of Computer Science and informatics Dublin, Ireland
anthony.labaere@eleves.ec-nantes.fr

ABSTRACT

SUMO is a powerful urban traffic simulator that has the potential to help road authorities to make their transportation systems 'smarter' (i.e., better and at a lower cost). However, SUMO is also a little complex, and integrating it to existing systems can be tricky. This paper presents ASTra, an API built in Python, that allows developers of transportation systems to interact with an instance of SUMO remotely (using sockets). Using ASTra, it is possible to run a SUMO instance, add or remove vehicles, process routing requests and manage traffic lights, without expert knowledge of SUMO.

Keywords

SUMO; TraCI; Web Service

1. INTRODUCTION

SUMO [3] and TraCI [5] work together to offer a dynamic traffic simulation: SUMO creates and runs a simulation environment, while TraCI allows users to interact at runtime with the simulation engine: adding cars, modifying traffic lights. These two tools have proven to be helpful in research for many purposes [2], such as evaluating Inter-Vehicle Communication (IVC) protocols [1] or Vehicular Ad Hoc Networks (VANETs) [4]. The Performance Engineering Laboratory[1] of UCD is working on a traffic management system: SUMO is used extensively to simulate some vehicular traffic and user trips, and TraCI is used to configure the simulations during runtime.

Two observations that were made from using these tools include: (i) most of the features offered by TraCI are very low-level; and (ii) TraCI requires the use of Python to interact with it. These limitations motivated the development of a *high-level, language independent, library for interacting with TraCI and running traffic simulations.* This paper includes a presentation of ASTra in some details (Section 2 and 3), which is then compared to a recent webservice approach of TraCI (Section 4).

[1] http://pel.ucd.ie

2. OVERVIEW

ASTra[2] is a multithreaded Python process that uses TCP/IP connections, based on sockets, for each of the main functionalities (except the simulator). Each thread waits for a request which is processed and followed by a response, if relevant. All threads can be enabled or disabled and come with a high reliability - with automatic restarts for the whole API and simulation if a serious issue occurs.

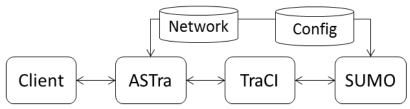

Figure 1: Communication diagram

Figure 1 illustrates ASTra's communication. As depicted in the figure, ASTra manages every TraCI and SUMO interaction itself, and only needs the SUMO network and a few configuration files. Every interaction between ASTra and the remote client will be done using socket messages.

3. ASTRA'S ARCHITECTURE

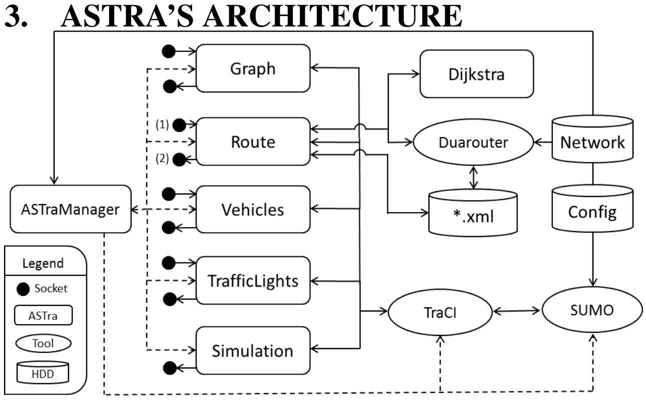

Figure 2: Detailed architecture

(1) Socket requests:
GET <algorithm>[<vehicleId> <edgeIdSrc> <edgeIdDest>]*
GET <algorithm>[<vehicleId> <srcCoords> <destCoords>]*
(2) Socket reponses:
ROU <vehicleId> <edgeId1> ... <edgeIdN>
ERR <vehicleId> <errorCode>

[2] https://github.com/remidomingues/ASTra

3.1 ASTraManager

ASTra's main thread, the ASTraManager builds the data structures needed by the graph and traffic Light threads from the Network file. It also launches a SUMO subprocess, connects to TraCI, waits for the distant socket connections, then starts the threads described below. It is also responsible for the whole API restart if SUMO encounters an unexpected error.

3.2 Graph

The Graph thread provides enough information for rebuilding the network with an edge scale accuracy. The distant process will be able to access edges geographic coordinates, length, occupancy and successors. An edge can be partially or totally blocked by a call, caused by stopped vehicles on the lanes of the specified edge. Building and displaying the graph and studying the impact of an incident on the traffic congestion will be easier with these functionalities.

3.3 Route

The Route thread is responsible for routing requests; received calls (See Figure 2: (1)) must contain the routing algorithm chosen. Using Duarouter[3], the network has to be loaded for each request, resulting in a long processing time. Extending TraCI calls, ASTra build its own graph, providing a SUMO compatible routing functionality using a Dijkstra algorithm. The computation time between these algorithms is major, Duarouter requires important hard drive accesses for each routing requests set. This important issue is the main motivation for the development of a custom routing feature.

Crossing points can also be requested for both routings, based on geographic coordinates or edges IDs. The response contains the requested route, defined by a list of edge IDs, or an error code.

3.4 Vehicles

The Vehicle thread is used to add and remove vehicles to/from the simulation. The added vehicle can be described by a route or it can accept a random one. Neverthless, one of ASTra's main contributions is the priority vehicle functionality, managed by the simulation thread. Designed for ambulances, fire trucks and police cars, its main objective is to improve the traffic on the roads (the vehicle will use) by managing traffic lights phases.

3.5 Traffic lights

The Traffic Lights thread is used to access traffic lights details and geographic coordinates. Details are defined by a SUMO screenshot of the specified junction, a current phase index, the next switch time and for each phase, its state and duration. These details can also be updated by a remote procedure call.

3.6 Simulation

The Simulation thread is responsible for the SUMO simulation steps; this thread sends periodic messages containing vehicles geographic coordinates and arrived vehicles' IDs. One thing to note, given that ASTra knows each vehicle's route, it can detect every junction with a traffic light that a priority vehicle will cross. When the vehicle is close enough, an 'orange' temporary phase is first set. This temporary phase is then removed and an existing phase, which has a green traffic light for the selected lane, is identified and set.

[3] `http://sourceforge.net/apps/mediawiki/sumo/index.php?title=DUAROUTER`

4. RELATED WORK

TraaS[4] is a Web service developed in Java which, aims at interacting with TraCI using the SOAP protocol. Both ASTra and TraCI provide a TCP/IP connection for access to the simulation at runtime, however, ASTra distinguishes itself by its high level calls and new features such as a routing service and a priority vehicles management. As opposed to the SOAP technology, a string, lighter-weight, data-interchange format was chosen.

5. DEMONSTRATION

During ASTra's presentation at the Irish SUMO User Workshop, some of the main elements of ASTra will be demonstrated: routing, vehicule management, traffic light management. The demonstration will also be used to showcase how easy and powerful ASTra is when developers want to simulate some traffic for their traffic management system.

6. CONCLUSION AND FUTURE WORK

Designed for microscopic simulation requiring an at runtime remote access, ASTra offers a user-friendly interface which free the user from programming languages constraints and extends SUMO and TraCI functionalities. The feature developed can be improved and new ones will be developped, such as running multiple SUMO simulations, managing detailed traffic and CO2 probes, etc. This work provides a valuable contribution to the SUMO users community, and could potentially lead to new research and software development.

7. REFERENCES

[1] Stefan Joerer, Christoph Sommer, and Falko Dressler. Toward reproducibility and comparability of ivc simulation studies: a literature survey. *Communications Magazine, IEEE*, 50(10):82–88, 2012.

[2] Daniel Krajzewicz. Summary on publications citing sumo, 2002-2012. In *the first SUMO Conference.*

[3] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138, December 2012.

[4] Michal Piorkowski, Maxim Raya, A Lezama Lugo, Panagiotis Papadimitratos, Matthias Grossglauser, and J-P Hubaux. Trans: realistic joint traffic and network simulator for vanets. *ACM SIGMOBILE Mobile Computing and Communications Review*, 12(1):31–33, 2008.

[5] Axel Wegener, Michał Piórkowski, Maxim Raya, Horst Hellbrück, Stefan Fischer, and Jean-Pierre Hubaux. Traci: an interface for coupling road traffic and network simulators. In *Proceedings of the 11th communications and networking simulation symposium*, pages 155–163. ACM, 2008.

[4] http://traas.sourceforge.net/cms