1 LiSuM: Design and Development of a Middleware to couple Virtual LISA+ TLS Controller and SUMO Simulation

Maximiliano Bottazzi, Louis C. Touko Tcheumadjeu, Jan Trumpold, Jakob Erdmann, Robert Oertel; German Aerospace Center (DLR)
{maximiliano.bottazzi, louis.toukotcheumadjeu, jan.trumpold, jakob.erdmann, robert.oertel}@DLR.de

1.1 Abstract

Traffic signal control logic programs are analyzed and tested in traffic flow simulators before being put into operation in real traffic road intersections. LISA+ is a proprietary software tool used to plan and evaluate complex intersections and the control logic programs created with it can be directly uploaded to real controllers or tested with VISSIM, a proprietary microscopic 3D traffic simulator. On the other hand, Simulation of Urban Mobility (SUMO) is a free and open traffic simulation tool suite that facilitates simulation of traffic and the evaluation of infrastructure changes as well as policy changes before implementing them on the road. As mentioned above, LISA+ control logic programs can be currently tested only with proprietary software and a free and open alternative, like SUMO, is desirable. To close this gap this paper presents the design and development of a middleware called LiSuM that enables the communication and interaction between the virtual LISA+ traffic light signal (TLS) controller and SUMO. Furthermore LiSuM provides also a friendly graphical user interface (GUI), which allows managing all aspects related to the interaction between LISA+ and SUMO. The main features of the LiSuM Middleware as well as the provided interfaces are also described. The usage of LiSuM is demonstrated for a single intersection simulated with SUMO and controlled by LISA+.

Keywords: SUMO, LISA+, traffic simulation, virtual TLS controller, detector data, Middleware

1.2 Motivation

So far, most traffic-actuated controls for traffic lights can only be simulated with self-written scripts in SUMO [2] [5]. The import of proprietary formats of complex signal programs such as those used with real traffic light controllers was not possible. To allow testing of LISA+ control algorithms within a SUMO simulation, the realization of a middleware that interfaces these programs is required. Such communication interfaces [4] already exist between the VISSIM simulation software and the virtual LISA+ controller (see Figure 1.1). The aim of these interfaces is to exchange the traffic demand data obtained from loop detector from the simulation (e.g. SUMO or VISSIM) and the switch commands for the signal heads (e.g. red, yellow, green) obtained from the virtual TLS controller (e.g. LISA+ controller) so that the simulation can transfer the obtained signal heads to the traffic light operating into the simulation. How the simulated controller processes the data and reacts depends on the traffic-actuated control logic planed with LISA+ OTMC (Open Method of Traffic Control) editor [4] (see Figure 1.1) before. Our software solution which realizes the communication interface between

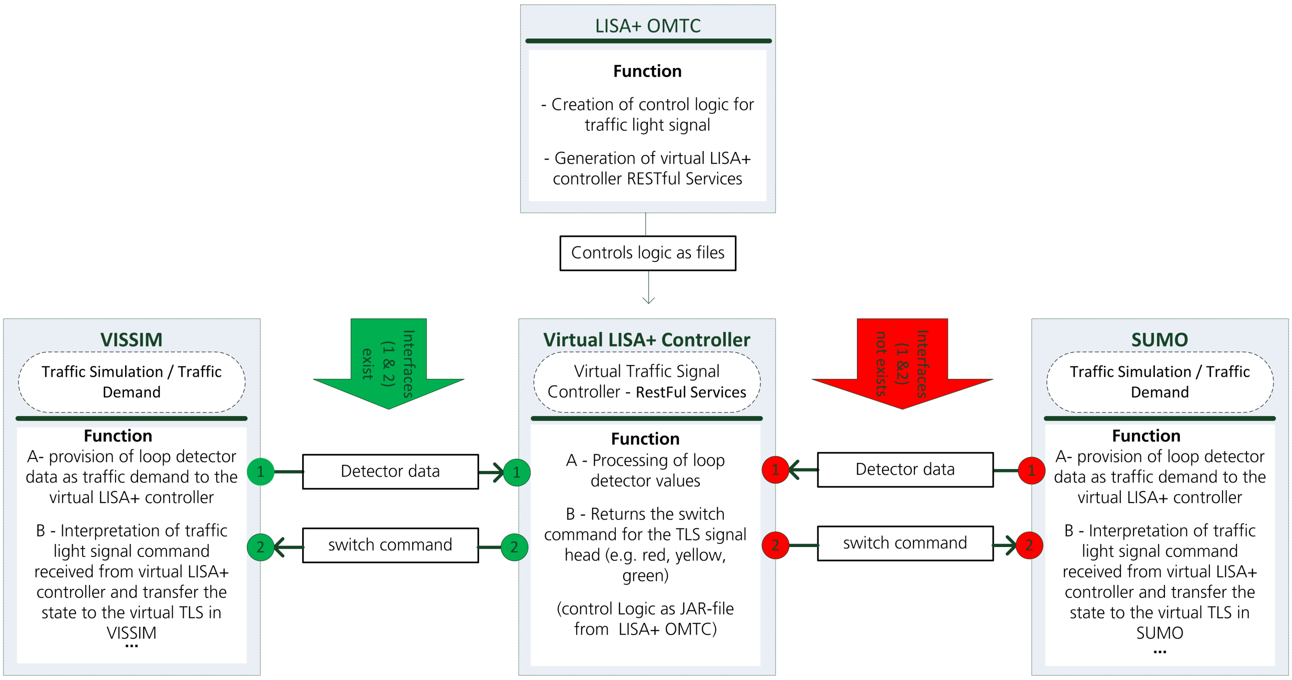


Figure 1.1: Overview about the communication between Virtual LISA+ TLS Controller, VISSIM and SUMO simulation [Designed by DLR]

the virtual LISA+ TLS controller and SUMO is called LiSuM (LISA SUMO Middleware) middleware (see Figure 1.2). It provides the following functionalities:

- LiSuM is a standalone middleware that provides the SUMO interface to LISA+ controller helping to execute more complex traffic controls on the intersections than SUMO originally permits.
- SUMO communicates with the LISA+ virtual controller through LiSuM.
- LiSuM supports the RESTful service interface as well as TraCI (Traffic Control Interface) [6] interface therefore acts as interface converter between LISA+ RESTful interface and SUMO TraCI4J interface.
- LiSuM provides also friendly graphical user interface and allows the user to manage the whole interaction process (E.g. configuration, switching signal programs, start and stop) between SUMO and the virtual LISA+ controller.
- LiSuM interacts as mapping tool for example for detector, signal groups between SUMO and the virtual LISA+ controller as well as green phase, wherever naming convention in the two tools are different.

1.3 Introduction

In this section a short introduction about the following three tools LISA+, VISSIM and SUMO is given.

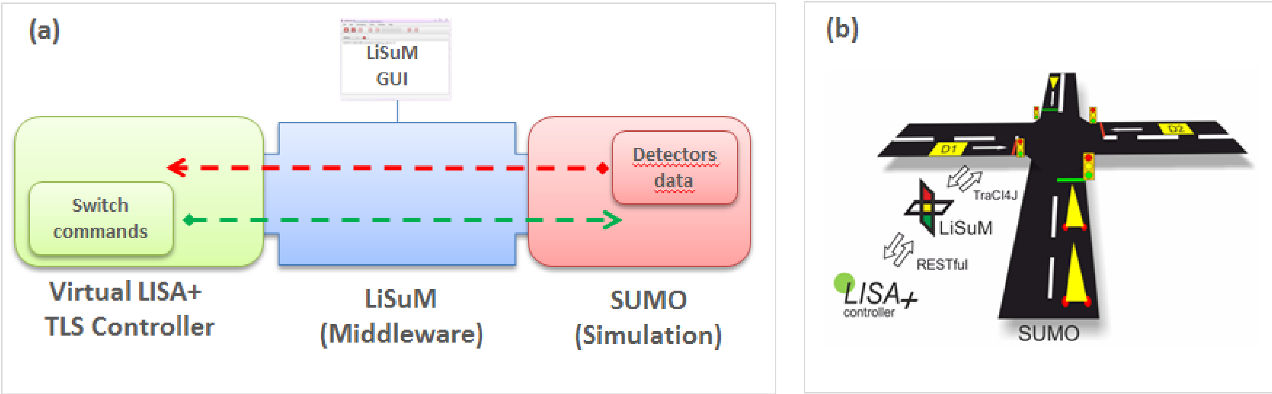


Figure 1.2: (a) SUMO communicates with the virtual LISA+ TLS controller through LiSuM middleware interface, (b) Schematic representation with RESTful interface between LISA+ and LiSuM and TraCI4J between SUMO and LiSuM

1.3.1 LISA+

LISA+ is a proven traffic engineering software for planning and evaluation of traffic signal lights developed, updated and commercialized by the German company Schlothauer & Wauer [4]. The Scope of the Software is a traffic engineering workstation to create traffic signal controls for individual intersections and progressive systems, evaluate controls based on current guidelines, plan, test and simulate traffic-actuated controls and upload data to controllers directly and remotely. Target groups for LISA+ are traffic engineering companies, municipal administrations, signal manufacturers and universities. The LISA+ tool is mostly used in Europe (e.g. Germany, Austria, and Belgium) and in South America (e.g. Mexico, Brazil, and Columbia) [4]. For the traffic simulation software VISSIM, Schlothauer & Wauer already provides an interface to LISA+. With this interface and LISA+ it is possible to run a traffic simulation with traffic-actuated traffic light controls built in LISA+. While the LISuM middleware is free software and is customized to support the LISA+ controller, it is recommended to have the LISA+ tools installed to run LiSuM middleware. Due to the fact that LISA+ is a commercial tool additional cost is needed to acquire the LISA+ planning tools as well as the virtual LISA+ TLC controller. The software license to used LISA+ products can be directly purchase by Schlothauer & Wauer company [4]

1.3.2 VISSIM

VISSIM is a microscopic multi-modal traffic flow simulation software package developed by PTV Planung Transport Verkehr AG in Karlsruhe, Germany [1]. VISSIM was first developed in 1992 and is today a global market leader for commercial traffic flow simulation software. "Multi-modal simulation" describes the ability to simulate more than one type of traffic. All these types can interact mutually. In VISSIM the following types of traffic can be simulated: Vehicles, (cars, buses, and trucks), Public transport (trams, buses), Cycles (bicycles, motorcycles), Pedestrians and Rickshaws. The scope of application ranges from various issues of traffic engineering (transport engineering, transportation planning, signal timing), public transport, urban planning over fire protection (evacuation simulation) to 3D visualization (computer animation, architectural animation) for illustrative purpose and communication to the general public. To simulate traffic lights VISSIM can connect with an interface to the LISA+ software developed by Schlothauer & Wauer. The simulation software sends information about the current detector state and the signal state to the LISA+ interface and the simulated LISA+ traffic light controller. The simulated controller processed all received information and sends traffic signal switch commands back to VISSIM. Multiple intersections can be controlled simulataneously. How the simulated controller processes the data and reacts depends on the traffic-actuated control planed with LISA+ before.

1.3.3 SUMO

SUMO (Simulation of Urban Mobility) is an open source, highly portable, microscopic road traffic simulation package designed to handle large road networks. It is mainly developed by employees of the Institute of Transportation Systems at the German Aerospace Center [5] [2]. SUMO is licensed under the GPL. It allows to simulate how a given traffic demand which consists of single vehicles moves through a given road network. The simulation allows addressing a large set of traffic management topics. It is purely microscopic: each vehicle is modelled explicitly, has an own route, and moves individually through the network. Simulations are deterministic by default but there are various options for introducing randomness. The software works on command line or can be used with the additional SUMO-GUI application which provides a graphical user interface to the simulation. Furthermore, SUMO has a list of additional tools like TraCI (Traffic Control Interface). TraCI giving the access to a running road traffic simulation, it allows to retrieve values of simulated objects and to manipulate their behavior "on-line". It can be used for adaptive traffic light control by a python script, for example.

1.4 Software Conception of the LISUM Middleware

This section describes the design and the development of the new software component LiSuM to couple virtual LISA+ controller and SUMO. Figure 1.3 shows how these components interact. It consists of communication interfaces, a graphical user interface and the configuration file. All these components are described in details in the following section.

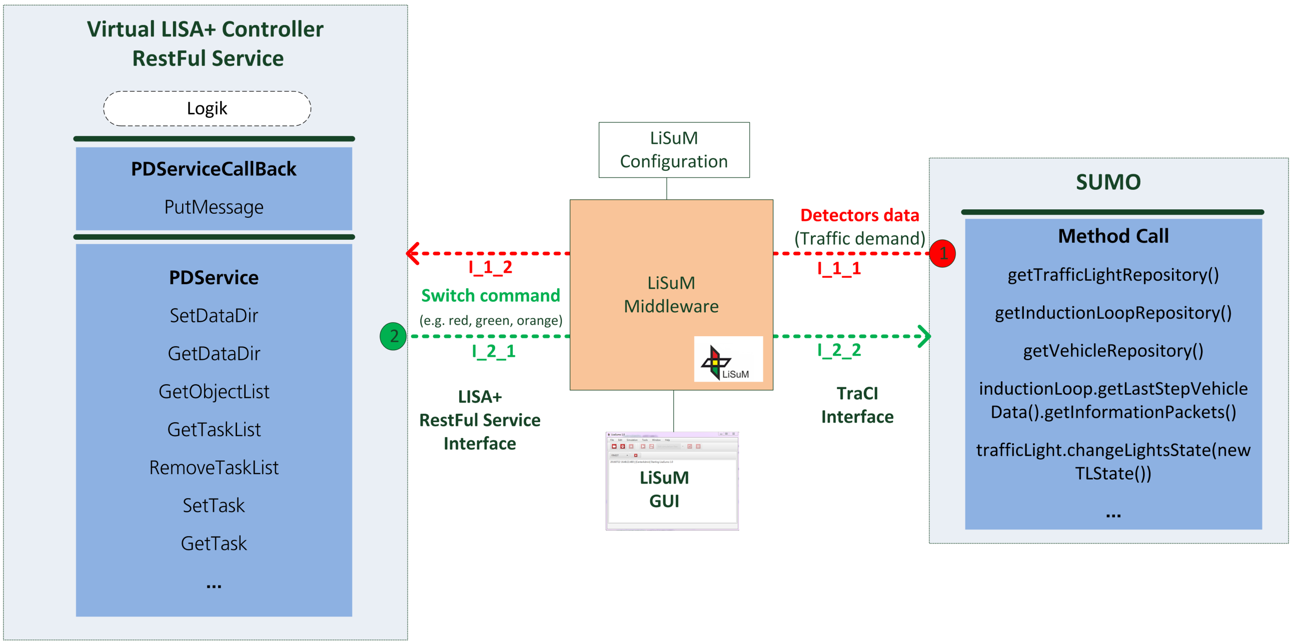


Figure 1.3: Overview about the LiSuM System

1.4.1 LiSuM Communication Interfaces

LiSuM middleware provides communication interfaces to couple virtual LISA+ controllers and SUMO. On one side, it communicates with the LISA+ controller over its RESTful interface[3], the communication protocol is REST/HTTP and the data format is described using WSDL/XSD files provides by Schlohauer & Wauer [4] and used to describe SOAP services. On the other side, LiSuM communicates with SUMO over TraCI4J (Traffic Control Interface for Java) interface [6]. Table 1.1 gives an overview about the implemented interfaces by LiSuM. To retrieve or send the detector data and switch command for header signal, LISA+ controller server provides some RESTful services and SUMO some appropriate methods for this purpose (see Figure 1.3). Using the provides RESTful web services (LISA+) and TraCI4J (SUMO) interface the LiSuM middleware calls the corresponding method or services to transfer detector data from to the LISA+ Controller and send switch commands for all controlled traffic lights to the SUMO simulation. The LISA+ side of the interface is

Table 1.1: Supported communication interfaces by LiSuM

No	Interface Identifier	Interface Type	Interface Description	Exchange Data	Source Component	Destination Component
1	I_1_1	TraCI	Interface to retrieve the detector data from SUMO to LiSuM	Detector data	SUMO	LiSuM
2	I_1_2	RESTful	Interface to send the detector data from LiSuM to LISA+	Detector data	LiSuM	LISA+
3	I_2_1	RESTful	Interface to retrieve the switch command from LISA+ to LiSuM	Switch Command	LISA+	LiSuM
4	I_2_2	TraCI	Interface to send the switch command from LiSuM to SUMO	Switch Command	LiSuM	SUMO

a simulated traffic light controller in an executable stand-alone Java JAR called LISA+ OMTC that simulates control logic programs. The virtual LISA+ controller acts as virtual RESTful Server and need to be installed in order to run LiSuM.

1.4.2 LiSuM Graphical User Interface (GUI)

In addition to the communication interface features to couple the virtual LISA+ TLS controller and SUMO, the LiSuM middleware provides also a graphical user interface. It main window is depicted in the Figure 1.4. The LiSuM GUI consists of menu items, toolbar buttons, and management console to the activities of the running processes during each simulation step, a configuration file and a Control Unit Management panel. The LiSuM GUI provides a series of functions linked to the toolbar buttons [see Table 1.2] and the menu items (and shortcut keys combinations) which allows to manage all aspects related to the interaction between LISA+ and SUMO such as opening a simulation project, starting a SUMO GUI instance, editing simulation project files without switching application and managing the LISA+ control units.

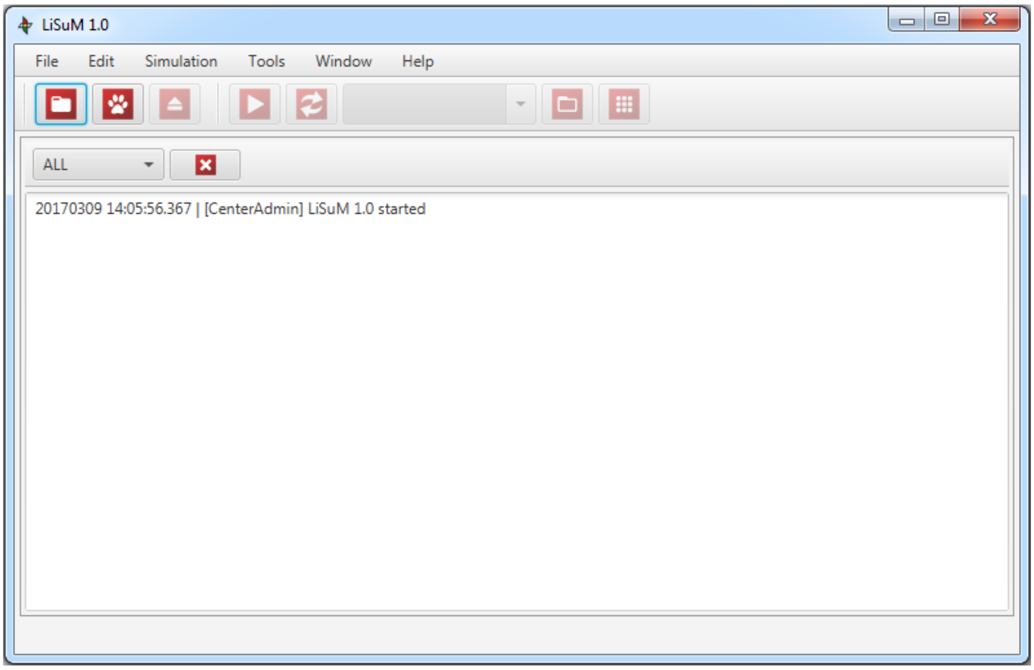


Figure 1.4: LiSuM GUI main window

1.4.3 LiSuM Control Unit Management

The Control Units Management dialog (Figure 1.5) allows selecting different signal programs for the each of the controlled traffic lights. It allows starting the programs and switching them to traffic actuated operation. By default, control units that do not have any Sumo intersection assigned are disabled. The communication with LISA+ can be toggled via the Ónćheckbox.

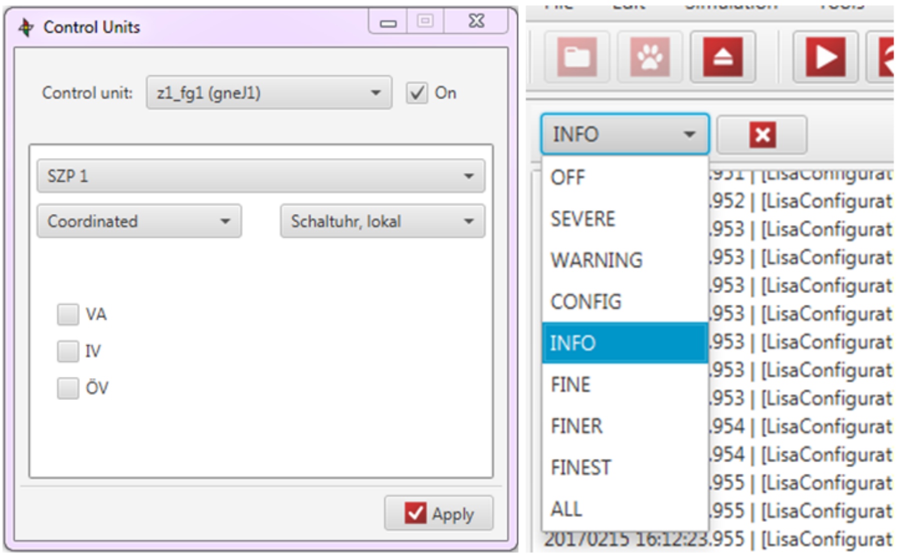


Figure 1.5: LiSuM Control Unit Management window

1.4.4 Detector and Signal Group Mapping between LISA+ and SUMO

LiSuM interacts as mapping tool for detectors and signal groups between SUMO and the virtual LISA+ controller, because naming conventions in the two tools are different. This mapping is defined in the LiSuM configuration file (see Section 1.6 for more details). The differences in representation and naming between LISA+ and SUMO are described below:

Table 1.2: LiSuM toolbar functions

No	Toolbar Icon	Function, shortcut and menu item
1		Opens the file selector to open a simulation project(Ctrl+O)(File > Open)
2		Opens the last simulation project (Ctrl+1)
3		Closes the current simulation project (Ctrl+W) (File > Close)
4		Opens the SUMO-GUI (Ctrl+P) (Simulation > Open SUMO)
5		Reloads the files of the current opened simulation project (Simulation > Reload simulation project files)
6		Explores the folder of the current opened simulation project
7		Opens the control units management window(Ctrl+M) (Tools > Control Units Management)

- Induction loops
  - LISA+ assigns every detector a name as combination of character and number (e.g. D1 etc.), while SUMO allows arbitrary strings (e.g. myLoop1, etc.). `<detector lisa="D1" sumo="myLoop1" />`
- Traffic lights
  - Different names, in LISA+ format central number and field device number (z1_fg1) in SUMO any name (gneJ1). `<controlUnit lisa="z1_fg1" sumo="gneJ1" >`
  - LISA+ assigns every signal group a name as combination of character and number (e.g. K1, K2, etc.), while SUMO uses only a number (e.g. 0, 1, 2, etc.). `<signalGroup lisa="K1" sumo="0" />`

1.5 The Green Phase Mapping between LISA+ and SUMO

While LISA+ only defines and uses only one green phase, SUMO uses two different types of them: g (no priority) and G (priority). This distinction corresponds to additional signals which may signal full priority to left-turning vehicles. To do this modeling choice in SUMO, when LISA+ switches a signal group to green, LiSuM has to decide which of the two types of green it must send to the corresponding SUMO signal group. The ideal solution would be to represent to model the correspondence with the additional signals but the current version of LiSuM uses a simplified solution: LiSuM decides beforehand which green phase each SUMO signal group will ALWAYS receive based on the Gs and gs that each signal group has in their configuration strings in the net.xml file of the simulations project (tag tlLogic), having g priority over G. In other words, if a signal group has at least one g, that group will always switch to g, otherwise to G. This solution (see Figure 1.6) achieves save traffic behavior but slightly reduces travel speed in some situations.

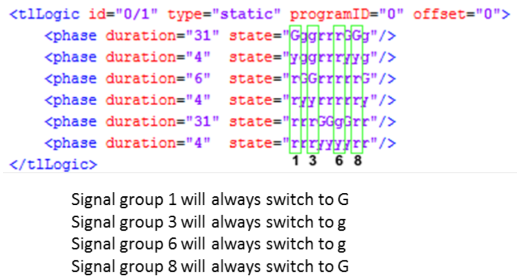


Figure 1.6: Some examples

1.6 LiSuM Configuration File

The LiSuM configuration file is a XML file (always named lisum.xml) which contains all necessary information used to control the way that LISA+ communicates with SUMO during the execution of a simulation. It is basically composed of the mandatory elements input and controlUnits that is composed of the two non-mandatory tags controlUnits and detectors. Since LISA+ and SUMO use different naming conventions for their elements, the fields controlUnits and detectors tell LiSuM how they are called in each system. The configuration file shown in Figure 1.7 above declares the

```
<?xml version="1.0" encoding="UTF-8"?>
<simulation>
   <input>
      <lisa>lisaDirectory</lisa>
   </input>
   <controlUnits>
      <controlUnit  lisa="z1_fg1" sumo="gneJ1" >
         <signalGroups>
            <signalGroup  lisa="K1" sumo="0" />
            <signalGroup  lisa="K2" sumo="1" />
            <signalGroup  lisa="K3" sumo="2" />
            <signalGroup  lisa="K4" sumo="3" />
         </signalGroups>
         <detectors>
            <detector  lisa="D1" sumo="myLoop1" />
         </detectors>
      </controlUnit>
      <controlUnit  lisa="z1_fg2" sumo="gneJ2"
>
         <signalGroups>
            <signalGroup lisa="K1" sumo="0,1,2" />
            <signalGroup lisa="K2" sumo="3,4" />
            <signalGroup lisa="K3" sumo="5,6,7"/>
            <signalGroup lisa="K4" sumo="8" main="K3" />

</signalGroups>
      </controlUnit>
   </controlUnits>
</simulation>
```

Figure 1.7: LiSuM sample configuration file

following:

- The path of the directory where the LISA+ control logic files generate from the LISA+ OMTC Editor tool are locate is specified in the configuration file in the mandatory tags input and the element lisa as shown here `<lisa>lisaDirectory</lisa>`. (e.g. lisaDirectory = C:/lisa/).

The generated LISA+ control logic files consist for example of followings files: z1_fg1.jar, z1_fg1.kfg, z1_fg1.szp, z1_fg1.xml

According to the configuration file, two control units are configured. The explanation is as follows:

- the first control unit:
  - is called z1_fg1 in LISA+ and gneJ1 in SUMO..
  - mapping of LISA+'s control unit z1_fg1 with gneJ1 in SUMO is provided.
  - contains one detector D1 in LISA+ and myLoop1 in SUMO and the mapping table.
  - is composed of four signal groups (K1, .., K4) for LISA+ and four (0, .., 4) for SUMO and contains the mapping information.
- the second control unit:
  - is called z1_fg2 in LISA+ and gneJ2 in SUMO.
  - contains also four signal groups (K1, .., K4) for LISA+ and nine (0, 1, .., 8, 9) for SUMO. It contains also the mapping information for example; the LISA+ signal group K2 controls the SUMO signal groups 3 and 4.
  - the fourth signal group contains the attribute main set to K3. This means that if the LISA+ signal group K4 is disabled (OFF) at any time of the cycle the SUMO signal group 8 will take the state of K3.

1.6.1 Getting started with LiSuM

LiSuM is licensed under the GPL and its current version is the 1.0.0 and was built on Java and thus can be run on any operating system. LiSuM is scheduled to be made available as part of the SUMO distribution. It is run by executing its jar file. It requires the Java SE Runtime Environment (version 8 or later), SUMO (version 0.29.0 or later) and LISA+ TLS planning tool and virtual controller installed in advance before beginning with the installation. The LISA+ tool and its virtual controller component is a commercial product and can be obtained directly from the Schothauer & Wauer company [4]. When LiSuM is started for the first time, the user is prompted to select a the workspace directory. The workspace is the directory where LiSuM looks for existing simulation projects, where new ones should be stored and where the system preferences are saved. If needed use the system preferences window to change the workspace path. For better understanding of how LiSuM works, two sample projects (sampleSimulation and simpleSampleSimulation) are provided as part of the distribution.
First step: The first step consists of creating a new LiSuM simulation project, which is a directory containing:

- A LiSuM configuration file (lisum.xml)
- SUMO files (*.add.xml, *.net.xml, *.rou.xml, *.sumocfg, etc)
- A directory containing LISA+ control units files (exported from LISA+ OMTC)

Second step: The second step is to open LiSuM, set the SUMO path and select a workspace in the system configuration dialog and open a simulation project. In the tool menu, select "Start Lisa+ Virtual Controller" to start an instance of the virtual LISA+ Controller. Pressing Ctrl+p or clicking on the "Play" button on the toolbar will open an instance of the SUMO GUI, which will take control over the system. Almost all menus, toolbars and dialog windows of LiSuM get blocked and from here on the simulation may be started, paused, resumed and stopped from SUMO. Only the Control Units Management dialog window stays enabled so it is possible to change the control units settings during the execution of the simulation.

1.7 LiSuM Example: Single Intersection Control

In this section the exemplary simulation of a single intersection with a LISA+ TLS controller and SUMO is shown.

1.7.1 Using LISA+ OMTC Editor to plan and generate the control logic for an exemplary intersection with TLS and detector

The topology of the sample intersection selected to demonstrate the single-intersection control is depicted in Figure 1.8. It consists of three traffic lights and one loop detector. The simple example intersection has a major road from the north to the south and one minor road from the east. It has also single-lane approaches and added lanes for turning (see Figure 1.8). With LISA+ is a

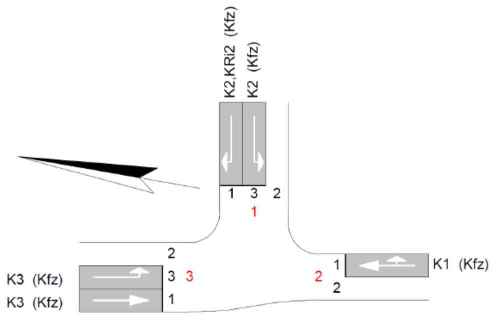


Figure 1.8: Topology and signal groups of the example intersection

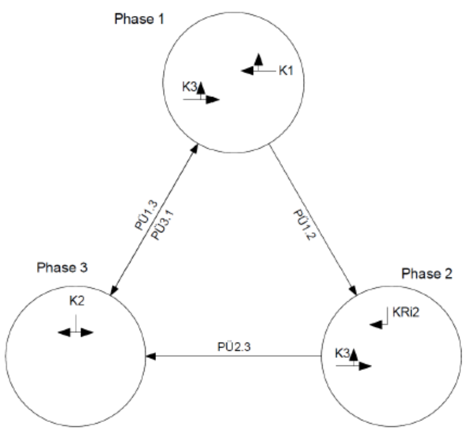
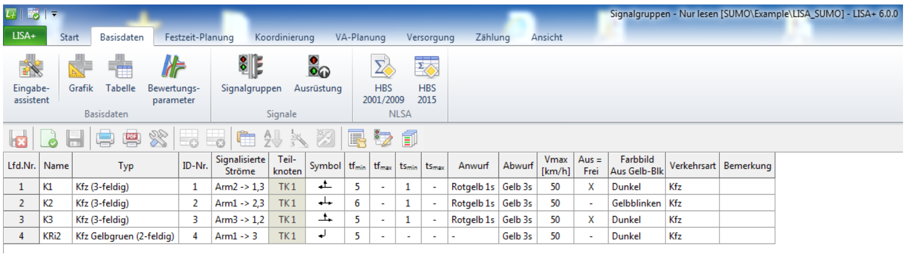


Figure 1.9: Sketch and phase plan of the example intersection

conventional vehicle-actuated control based on three phases build for this example intersection (see Figure 1.9).

- Phase one gives green to the approaches from north and south.
- Phase two gives green only for the north approach for the better traffic flow from north to east if it is necessary.
- Phase three gives green to the minor road. Furthermore, the minor road only gets green on demand and all phases have a green time adjustment by vehicle time gaps. A time gap from 1.5 seconds or more stops the current phase. The information about vehicle presence and time gaps are detected by induction loops in SUMO.

The signal groups (K1, K2 and K3) corresponding to the single traffic light signals located in the sample intersection is planned with the LISA+ Editor as shown in the Figure 1.10. The next step is to use LISA+OMTC Tool (see Figure 1.11) to plane the traffic-actuated control logic for the single signal group (K1, K2, K3) and the loop detector (D1). After that the files containing the traffic actuated control logic can be generated and save into the LISA directory into the LiSuM middleware as described in Section 1.6.



Lfd.Nr.	Name	Typ	ID-Nr.	Signalisierte Ströme	Teil-knoten	Symbol	tf_{min}	tf_{max}	ts_{min}	ts_{max}	Anwurf	Abwurf	Vmax [km/h]	Aus = Frei	Farbbild Aus Gelb-Blk	Verkehrsart	Bemerkung
1	K1	Kfz (3-feldig)	1	Arm2 -> 1,3	TK 1		5	-	1	-	Rotgelb 1s	Gelb 3s	50	X	Dunkel	Kfz	
2	K2	Kfz (3-feldig)	2	Arm1 -> 2,3	TK 1		6	-	1	-	Rotgelb 1s	Gelb 3s	50	-	Gelbblinken	Kfz	
3	K3	Kfz (3-feldig)	3	Arm3 -> 1,2	TK 1		5	-	1	-	Rotgelb 1s	Gelb 3s	50	X	Dunkel	Kfz	
4	KRi2	Kfz Gelbgruen (2-feldig)	4	Arm1 -> 3	TK 1		5	-	-	-	-	Gelb 3s	50	-	Dunkel	Kfz	

Figure 1.10: Planed signal groups in LISA+ OMTC

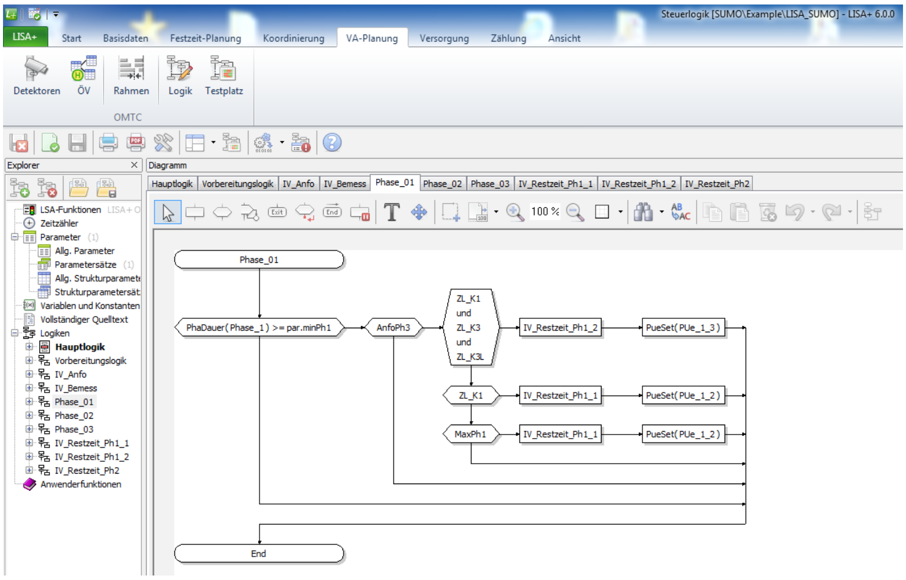


Figure 1.11: Planed traffic-actuated control logic in LISA+ OMTC [4]

1.7.2 Virtual LISA+ TLS Controller (LISA+ RESTful Server)

The virtual LISA+ TLS controller is provided as an executable stand-alone Java JAR file containing the LISA+ RESTful Server and is used to simulate control units devices and run control logics on it. Therefore it is necessary to have LISA+ installed to be able to run LiSuM. When starting a simulation using LiSuM GUI an instance of a LISA+ RESTful server must be running (see Figure 1.12). Per default the LISA+ RESTful server files can be found in the directory OmlFgServer. Per default, LiSuM searches for simulations in the simulations directory, which can be found in the installation directory (../DLR/LiSuM). This can be changed in the System Preferences Window (Menu: Tools –> Preferences). To start an instance goes to Menu: Tools –> Start LISA+ RESTful Server. The control logic (generated files in Section 1.7) previously created using LISA+ OMTC including all its parameter for the sample intersection is uploaded to the virtual LISA+ controller. The control logic files can be found in the OmlFgServer directory located in the LiSuM installation directory. The communication port and the RESTful web service address can be found and edited in the OmlFgServer.ini file (see Figure 1.13), if necessary.

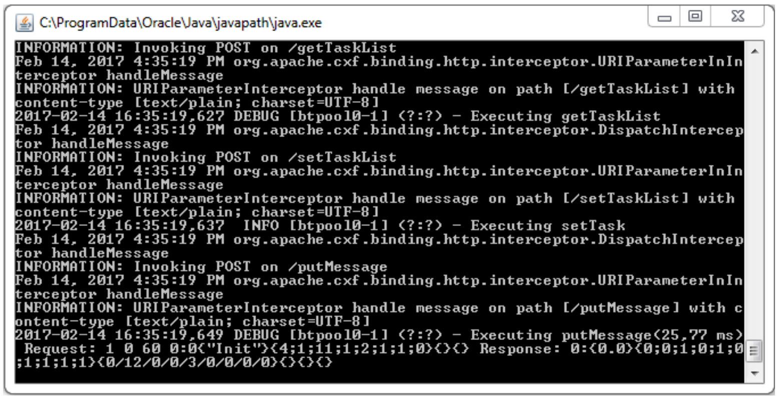


Figure 1.12: Running Terminal window of LISA+ RESTful Server (as virtual LISA+ TLS Controller)

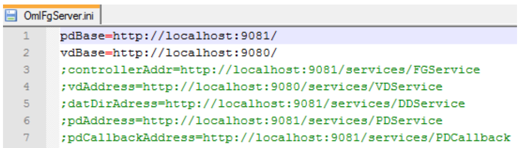


Figure 1.13: Overview about the OmlFgServer.ini file of LISA+ RESTful Server

1.7.3 SUMO Simulation

After starting the SUMO Simulation using LiSuM GUI (start button in toolbar or simulation item in menu bar). The controls created previously with LISA+ can be directly simulated with the simulation tool SUMO. The exchange of traffic demand information from configured loop detector in SUMO and the corresponding switch commands of the header signal (e.g. red, yellow and green) from the virtual LISA+ TLS controller happen via the implemented interfaces of LiSuM. The result of the simulation with the LiSuM interface can be shown the green light allocation in SUMO (see Figure 1.14 and 1.15) depending on the given traffic demand and the control method described previously in Section 1.7.

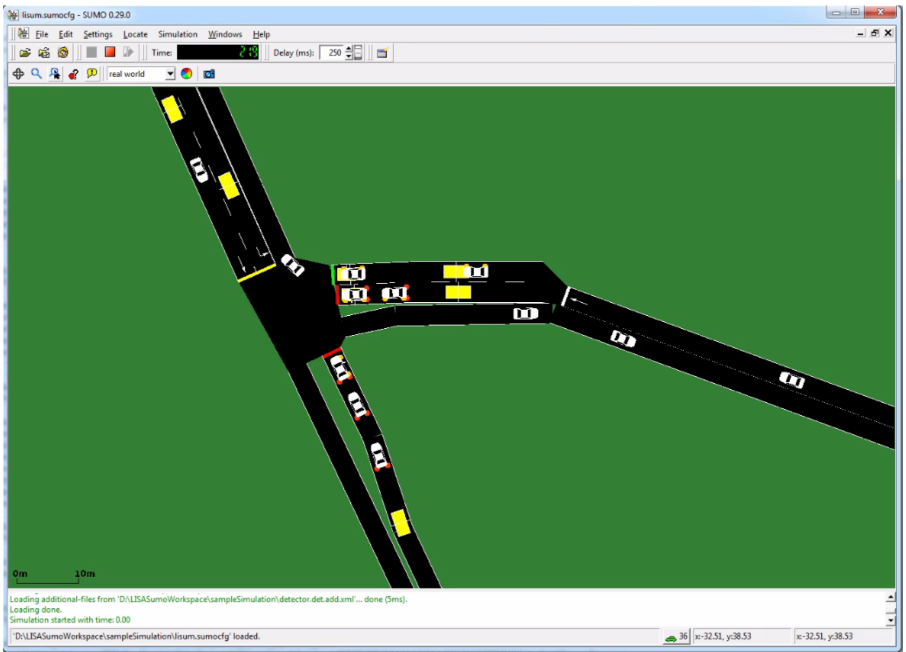


Figure 1.14: Sample intersection in SUMO

Figure 1.15: Signal timeline for the Sample intersection in SUMO

1.7.4 Performance Measurement of the Single Example Intersection

To compare the effect of LiSuM on simulation speed, the example scenario was executed with LiSuM and without out (using a fixed traffic light control program) on the same computer. Without LiSuM, the example scenario runs with a real time factor of 290 (simulating 290 minutes of traffic can be accomplished in a single minute of computing time). With LiSuM the overall simulation speed is reduced to a real time factor of 85 due to communication and synchronization delays. The LiSuM middleware can simulate multiple traffic light controllers at the same time and due to these preliminary results we expect scenarios with at least 10 controllers to be feasible.

1.8 Conclusion and Outlook

Only a small number of traffic-actuated controls schemes is available in SUMO and therefore most traffic lights can only be simulated with custom control scripts that must be developed on a case-by-case basis. To allow SUMO to simulate complex scenarios for traffic light signal (TLS) control like single intersections, coordination and adaptive controls, the realization of the communication interface between SUMO and existing virtual TLS controller is required. To close this gap DLR designed and developed a new software tool called LiSuM. LiSuM allows the communication between SUMO and the virtual LISA+ controller. LISA+ is a commercial product from the German company Schlothauer & Wauer that provides a series of tools for traffic signal and control logic planning. Control logic created with LISA+ OMTC is uploaded to the virtual LISA+ controller and can be directly simulated with the SUMO. LiSuM middleware provides communication interfaces to exchange detector data as traffic demand provided by SUMO and the switch commands of header signals (red, yellow and green) obtained from the virtual controller. LiSuM also provides a graphical user interface, which allows managing all aspects related to the interaction between the virtual LISA+ controller and SUMO. To facilitate this interaction, name-mappings between the LISA+ and SUMO domains must be configured beforehand. This paper describes the design and development of the LiSuM tool and demonstrates its usage with an example scenario.

Bibliography

[1] AG, PTV: *PLANUNG TRANSPORT VERKEHR, VISSUM*, 2017.

[2] Krajzewicz Daniel, 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, December 2012.

[3] RESTfull: *RESTful web service*, 2017.

[4] SCHLOTHAUER and WAUER: *LISA+*, 2017.

[5] SUMO: *Simulation of Urban Mobility*, 2017.

[6] SUMO: *TraCI Traffic ControlInterfaces description*, 2017.