iTETRIS Platform Architecture for the Integration of Cooperative Traffic and Wireless Simulations

J. Maneros[1], M. Rondinone[2], A. Gonzalez[3], R. Bauza[2], D. Krajzewicz[4]

[1]CBT Communication Engineering, Bilbao, Spain

[2]Uwicore Laboratory, University Miguel Hernandez, Elche, Spain

[3]Innovalia Association, Bilbao, Spain

[4]German Aerospace Centre Institute of Transportation Systems (DLR), Berlin, Germany

jmaneros@cbt.es, mrondinone@umh.es, aigonzalez@innovalia.org, rbauza@umh.es, Daniel.Krajzewicz@dlr.de

***Abstract*— The use of cooperative wireless communications can support driving through dynamic exchange of Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) messages. Traffic applications based on such systems will be able to generate a safer, faster, cheaper and cleaner way for people and goods to move. In this context, the iTERIS project aims at providing the framework to combine traffic mobility and wireless communication simulations for large scale testing of traffic management solutions based on cooperative systems. This paper addresses the description and explanation of the implementation choices taken to build a modular and interoperable architecture integrating heterogeneous traffic and wireless simulators, and application algorithms supporting traffic management strategies. The functions of an "in-between" control system for managing correct simulation executions over the platform are presented. The inter-block interaction procedures identified to ensure optimum data transfer for simulation efficiency are also introduced.**

Keywords: simulation platform, architecture, vehicular communications, traffic, modularity

I. INTRODUCTION

Exploiting the cooperative exchange of wireless messages between vehicles (V2V) and between vehicles and communication infrastructure (V2I) allows the realization of new road safety and traffic management application solutions. The final result of a proper execution of such applications may imply a significant set of beneficial effects for the society. The iTETRIS project (http://ict-itetris.eu/) aims at implementing an open-source integrated wireless and traffic simulation platform that will allow testing the efficiency of cooperative traffic management strategies using V2X technologies over large-scale scenarios. This paper will focus on practical issues concerning the implementation of the platform: the description of the different parts it consists of along with the relative interactions; the rationales behind the implementation choices; the overview of the whole system operation.

II. ETSI ITS COMMUNICATION ARCHITECTURE

iTETRIS is aligned with the communication architecture defined by ETSI for Intelligent Transport Systems (ITS). This architecture is depicted in Figure 1. The standard specifications concern a communication system designed for various types of traffic applications which can use several coexistent communication technologies. The architecture assumes three different actors communicating in an ITS scenario, each representing a given subsystem: vehicle, roadside and central subsystems. In the following, an overview is provided of the different layers represented in Figure 1.

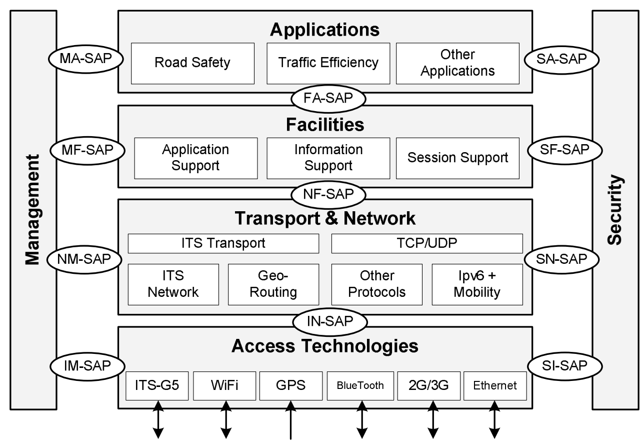

Figure 1 – the ETSI ITS Communication Architecture

ITS Communication Architecture's Access Technologies layer reflects CALM's objective to allow seamless communication over several coexisting radio access technologies [3]. The architecture has therefore to include mechanisms to dynamically select the most appropriate communication technology to be used. The Networking & Transport layer contains the different networking and transport protocols needed for a fully functional communication in an ITS scenario. Each networking protocol may be connected to a specific dedicated ITS transport protocol or to pre-existing transport layer protocols, e.g. UDP, TCP. The ITS Facilities layer collects a set of common functionalities which are shared by several applications for various tasks. The facilities provide data structures to store, process and maintain data of different type and source. As specified in [2], they can be classified into "*Application Support*", "*Information Support*" and "*Communication Support*" facilities. The Applications layer contains the user applications exploiting the communication functionalities provided by the remaining part of the communication protocol stack. "*Road Safety*", "*Traffic Efficiency*", and "*Other Applications*" are defined. The

Management layer is a transversal layer handling cross-layer information exchange among the horizontal layers. The main functionalities implemented in this block include the dynamic selection of the access technology for a given application, the monitoring of communication interfaces' parameters, the management of transmission permissions and priorities, the management of services, and the implementation of congestion control mechanisms. Finally, Security is the block implementing security services for the communication protocol stack and the management layer.

III. iTETRIS Simulation Platform Architecture

The iTETRIS simulation platform architecture consists of a set of functional blocks that collaboratively interact to simulate the application of innovative traffic strategies over ITS scenarios. In the following, a discussion is presented on the reasons motivating the iTETRIS simulation platform Architecture, which is described at the end of this Section.

A. *iTETRIS implementation requirements*

As already mentioned in Section I, iTETRIS aims at evaluating through simulations the performance of traffic management strategies supported by novel cooperative V2X communication systems. This focus on traffic management imposes in turn to consider large scale scenarios. These scenarios mean the involvement of a high number of vehicles distributed over wide areas, and whose behavior is analyzed over extended time periods. These conditions, whose adoption is synonym of reliability of the generated simulation results, may however request the employment of a considerable amount of resources in terms of both computational power and simulation time. This aspect is a very crucial point to be considered in the development of the iTETRIS architecture. It affects not only the choice of the architecture sub-components (traffic and wireless simulators), but also the way how they shall interact. The iTETRIS architecture sub-components have to provide acceptable degrees of accuracy but employing at the same time simplified models able to reduce the computational cost. All the interactions occurring between the different blocks of the resulting architecture shall be engineered to optimize the exchange of information by avoiding uncontrolled or inefficient data transfers. In this context, solutions are preferable which consist of centrally controlled, time-regulated, synchronized, sporadic and data-aggregating inter-block communications rather than autonomous, unsynchronized and too frequent exchanges of small amount of data.

B. *Traffic and Wireless Simulation Platforms*

The first and essential objective that iTETRIS wants to achieve is the accuracy in simulating the behavior of every ITS station making use of cooperative ITS communication systems. This objective is fulfilled in iTETRIS through the simultaneous and combined use of two simulation platforms, each of them addressing the emulation of vehicles' movement and wireless communication, respectively. Combining two pre-existing traffic and wireless simulator is a newly adopted method used to simulate vehicular communications [4]. This combined approach allows assessing the effects of actions triggered by exchanging messages over V2X communications on vehicles' mobility. The dynamic vehicular exchange of traffic information through the radio interface will cause, in some cases, route changes. At the same time, vehicles' movement will decisively affect wireless communications and will have a certain impact on communication protocols.

iTETRIS has opted for integrating two well-known and widely used open source simulation platforms. SUMO (http://sumo.sourceforge.net) is an open-source microscopic platform developed by the German DLR laboratories. It emulates vehicles' traffic mobility by a representation which is space-continuous and time-discrete. For what concerns the wireless communications simulator, iTETRIS decided to adopt the ns-3 platform (http://www.nsnam.org/) due to its capability to perform large-scale simulations and support multi-radio/technology [14]. The way of combining traffic and communication simulators was another aspect studied for the definition of the iTETRIS architecture. This combination can be achieved using different coupling approaches, like for example direct coupling [5][11], the use of middleware architectures for distributed simulations [11], or the adoption of an "in-between" control module [13] used for synchronization and coordination. Interoperability of the resulting platform represents a key aspect: the opportunity has to be taken into account to replace one of the simulators by another one. This would ensure, as well as a wider applicability of the platform, also its sustainability in case one of the two simulators became obsolete. Also, it is very important that the platform is autonomously developable, extendable and maintainable in each of its sub-components. Finally, if the reusability of the platform beyond the duration of the project is wanted, the possibility has to be given to an external user to implement and simulate his traffic applications in a fully flexible way. All the desired features listed above can be resumed in one single world: *Modularity*. By using a modular approach, the platform can be developed independently in each of its parts. The developers of each block of the platform are allowed to use the programming language they are familiar with and do not need to have a knowledge of the internal implementation of the other parts.

C. *The 3-Blocks approach*

Figure 2 shows the final choice made by iTETRIS for the definition of the platform architecture.

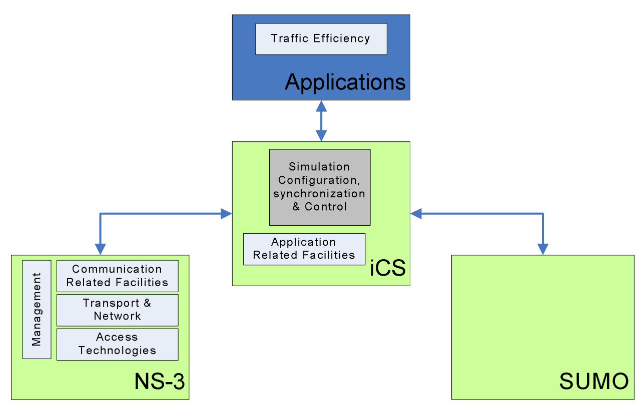

Figure 2 – the iTETRIS 3-Blocks Architecture

As it can be seen, in order to fulfill the modularity requirements explained above, a 3-Blocks approach has been adopted. The two open source simulation platforms, which are developed autonomously by dedicated experts, do not communicate directly, but are connected with a central “in-between” module named iTETRIS Control System (iCS). In order to consider simulation efficiency for large scale simulation, the iCS and the simulation platforms components run under the same machine. The traffic applications that can be simulated in iTETRIS are defined as an iCS-assisted external block running on the top of the platform. In this way, platform users can create their application definitions in a “language-agnostic” fashion. As the central block in the simulation architecture, the iCS is the entity that triggers the start of the different platforms, sets up the simulation environment, coordinates and controls the execution of the simulation. Each ITS station simulated on the platform has a specific representation in each of the iTETRIS blocks. SUMO is controlled by the iCS for emulating objects’ movements. It receives through the iCS a set of Application driven commands indicating the actions that have to be performed on the road network of the simulated traffic scenario (e.g. modifications of vehicles’ routes or actuation of particular traffic light policies). At the same time SUMO keeps tracks of the position of its objects (i.e. vehicles) and provides the other blocks with up-to-date values, in case these positions change over the time. ns-3 simulates the wireless communications that have to be run in the iTS scenarios. As it happens with SUMO, the interaction between ns-3 and iCS depicted in figure 2 is bidirectional. The iCS, if requested by the simulated Application, schedules control messages to be sent down to ns-3. This is needed to force the radio transmission of packets through the implemented ns-3 radio access technologies, according to the specifications of the running traffic Application. At the same time, ns-3 sends up to the iCS information messages for every packet received over the radio channel interfaces whose content has to be processed by the Application. Moreover, ns-3 is continuously fed by the iCS with the position updates it retrieves from SUMO. ns-3 uses these values to update the positions of its internal vehicle node representations. Based on these positions, the communication capabilities of ns-3 nodes vary a lot. Besides affecting the quality of vehicular radio links, changing nodes’ position may led vehicles to zones where a given radio technology cannot be used (e.g. absence of UMTS signal coverage). The Application, besides determining the actions that have to be executed in terms of both vehicles’ movement and wireless communication, reacts to the events generated by the simulators. A notification about a given packet reception simulated in ns-3, may imply various different reactions at Application side. A given packet transmission may need to be immediately executed in ns-3, or the change of a vehicle’s route may need to be triggered in SUMO. As it will be better explained in the next sections, the iCS is not only capable to administrate this switching of messages between the blocks it interfaces. The iCS also contains some internal specific functionalities (the Application-related facilities), which are essential for the platform. These functionalities implement methods and databases supporting the simulated object representations and the relative Applications. Through these methods and databases, the messages coming from the simulation platforms are filtered, fused, stored, updated and maintained for every simulated object. After having extracted the data it needs from these databases, the Application is executed for every addressed object. If the results of the Application have to be stored in the iCS, the same databases will be accessed again and the corresponding data will be updated.

IV. COUPLING

To achieve efficient interaction between the different iTETRIS sub-components, when defining the iCS, various candidate technologies and architectures have been considered. Common Object Request Broker Architecture (CORBA) presents considerable drawbacks such as location transparency, and thus results an expensive implementation, especially if it is considered that the simulation platforms reside in the same machine [6][7]. Simple Object Access Protocol (SOAP) [8], despite the advantage of being platform and language independent, relies on XML format, which is very verbose. This causes network overloading and slows down simulation executions, especially in case of large scenarios (25.000 nodes). For this reason, SOAP was discarded [9].

High Level Architecture (HLA) specifications provide the framework to integrate different simulators and specify Run Time Interface (RTI) components [10]. However, large scale and highly populated simulation scenarios could not be met by existing open source implementations of the RTI. Internal tests were carried out by the project partners even with commercial versions of the RTI. The approach seemed not to be performing since messages were exchanged at an object level over the HLA bus. Hence, HLA was also discarded. The fourth option, and the one finally selected for iTETRIS iCS implementation, has been Internet Sockets based on the Internet protocol. Through a very simple mechanism and a straightforward technology, the iCS can communicate with the other blocks by just specifying on which IP address and port SUMO, ns-3 and the Traffic Engineering Applications will be listening to. The Internet Sockets option puts more focus on effective iCS implementation compared to HLA counterpart. On the other hand, it allows for a key feature in terms of iTETRIS performance, namely the possibility of designing a highly effective communication protocol across simulation platforms.

A. Approach to iCS-ns-3 Coupling

The iTETRIS 3-Blocks Architecture is fit for a clear separation of functionalities and therefore suggests a split of the ETSI ITS communication stack (Figure 1) over the combination of ns-3 and iCS (Figure 2). In this way, both ns-3 and iCS, for their internal operation (i.e. when the Application layer and the lower layers do not need to communicate with each other) can use the functionalities implemented by themselves, without the need to query the other block. This approach avoids unnecessary exchange of information between the different parts of the platform, therefore addressing the requirements explained in Section III.A to reduce the simulation time. As it can be appreciated in Figure 2, ns-3 implements all the layers which are essential to support the radio communication, while the iCS provides some supporting functionalities for the Applications running on the top of it. A similar consideration can be made for the iTETRIS implementation of the ETSI ITS Facilities layer. Some facilities are mostly accessed by the Application (Application-related Facilities) and are placed in iCS, while some others (Communication-related Facilities) are needed to support communication sessions and are in ns-3. Clearly, iCS and ns-3 rely on the facilities they strictly need for their autonomous and dedicated functioning, so they will not consult the other block, if not needed. Figure 2 shows the ETSI ITS stack's layers implemented in ns-3. The iTETRIS Access Technology block models the radio access interfaces considered in the project: IEEE 802.11p (also known as WAVE and named ITS-G5 by ETSI), UMTS, IEEE 802.16 (WiMAX) and DVB-H. The Transport & Networking layer implements the relative needed protocols. The iTETRIS Management mostly implements functionalities for the dynamic selection of the suitable radio access technology.

The interactive communication process between iCS and ns-3, theoretically described in Section III.C, is accomplished by a client/server relationship, where the iCS always is the controlling actor. A client entity is created in the iCS facilities layer. This client entity schedules all the commands to trigger ns-3 transmissions, and actively request ns-3 for returning all the relevant data resulting from packet receptions. At the same time, ns-3 performs as a server entity. Through the use of adequate primitives, it first executes the iCS commands for transmissions, and then returns to iCS all the data derived from simulated receptions. Two different primitives will be used. The iCS Communication Control Primitive (iCSCCP) is used when the Application requires triggering the simulation of a packet transmission over the radio. Through this primitive, the iCS communicates to ns-3 instructions about the content of the message to be transmitted, and the transmission modalities as required by the Application. ns-3 responsibility is to correctly interpret this information and perform the transmission in the most convenient and effective way. In other words, the iCS forwards instructions in a general Application-related format, and ns-3 can convert them into practical solutions in terms of radio access technology and communication networking protocol. The ns-3 Communication Control Primitive (ns3CCP) is instead the primitive adopted by ns-3 to communicate to the iCS the content of a packet after simulating its reception over the radio. The information received through this primitive will be then processed by both the iCS and the Application. iCS may eventually update its own data repositories in the Application-related facilities, while the Application may react according to the content of the packet and its own internal rules. iCSCCP and ns3CCP have a similar structure and are defined in the following way:

iCSCCP(EntityID, MsgType, ServiceID, AppCommParams, AppPrefs, Priority, AppDataPayload)

ns3CCP(EntityID, MsgType, ServiceID, AppDataPayload, TrInfo)

where:

- EntityID indicates the simulated communication or Application entity which has to process this primitive in ns-3 (in the case of iCSCCP) or iCS (in the case of ns3CCP).
- MsgType indicates one message type among the various message used in iTETRIS (e.g.CAM, DNM, Service REQ, Service REP, Dissemination message, etc…).
- ServiceID along with MsgType, is an identifier that defines how the primitive has to be processed, according to Application specifications.
- AppCommParams (only for iCSCCP) is a group of parameters indicating the Application requirements that define how this message should be transmitted (e.g. communication technologies specified by the Applications, timing information, addressing specifications, etc…).
- AppPrefs (only for iCSCCP) is an optional field specifying the Application transmission preferences, in case that more than one communication solution can be adopted (e.g. a criterion may be the costs implied by using a given transmission technology…).
- Priority (only for iCSCCP) indicates the priority of this message. The priority is taken into account by lower layers to schedule transmission of the corresponding packet. Packets with higher priority will be transmitted first on the medium.
- AppPayload contains all the Application data to be transmitted (in the case of iCSCCP) or processed after a reception (in the case of ns3CCP).
- TrInfo (only for ns3CCP) is an identifier of the ITS entity that transmitted the received message, in case it influences the Application reaction.

B. Approach to iCS-SUMO Coupling

A previous attempt of integration between traffic and wireless simulators concerned the direct coupling of SUMO with the network communications simulator ns-2 through the use of socket connections [5]. Since iTETRIS clearly refuses this sort of direct integration in favor of a modular approach, the need exists to find interaction modalities with the iCS. SUMO is equipped with an interface called Traffic Control Interface (TraCI) allowing users to start, pause and stop traffic simulations. Data concerning simulated objects (e.g. vehicles' positions, speed, direction) can also be retrieved through TraCI. Moreover, through this interface, it is possible to impose actions on the simulated traffic scenario from the outside (e.g. assignation of new route to vehicles, etc.). The iTETRIS iCS exploits TraCI to implement the exchange of information with SUMO as described in Section III.C. As for the interaction with ns-3, a client/server communication model is adopted. The iCS always acts as client entity generating commands to be executed in SUMO and retrieving data from it. An iCS module will be defined to implement the methods to build inter-block messages having a format which is appropriate to be processed by SUMO. On the other hand, TraCI will be enabled as a server entity during the start-up process. iCS will retrieve information from SUMO based on the concept of "Subscription". Through this method, the possibility is given to the iCS to use only one call (subscription) per object (e.g. a vehicle) to be continuously informed by SUMO on the monitored value (e.g. vehicle's position), every time this information is needed. Getting information using subscriptions rather than actively and continuously questioning SUMO for updates makes this process very efficient. In addition, subscribed values will be submitted from SUMO to iCS in single blocks. For what concerns the triggering of actions to be implemented in SUMO, the iCS client entity will use suitable messages to be interpreted and executed by SUMO. These messages will be used by the iCS to forward to SUMO the actions that the running Application wants to impose over the traffic network. In order to reduce the inter-block data transfer frequency, the possibility will be studied to join all these commands in aggregated messages. This last point, along with the subscription mechanism, considerably reduces the amount of messages exchanged between SUMO and iCS. In this way, the requirements imposed by using collaborating platforms to simulate large scale scenarios (Section III.A), which impose a careful use of inter-block communications, are successfully addressed.

C. *Approach to iCS-Traffic Engineering Applications Coupling*

In the context of iTETRIS, Applications are the algorithms that traffic engineers develop to give an appropriate solution to common daily traffic problems. Cooperative ITS applications relay on information collected from many sources (e.g. vehicles, road side units, historical data, etc.). Hence, the iCS implementation needs to know what the information required by the applications is, which mechanisms are used to collect such data and how to inform the relevant simulation objects about behaviour modifications based on application algorithms (e.g. adaptations of green phase duration, route changes, etc). In the following discussion, it is assumed that applications reside at the Roadside Station, Vehicular Station and ITS Service Centre (ETSI central subsystem). As already discussed in Section III, when designing the interface between Applications and iCS, a clear requirement is to avoid adding programming language constraints to Application developers. To achieve the objective, the architecture of Figure 3 has been proposed:

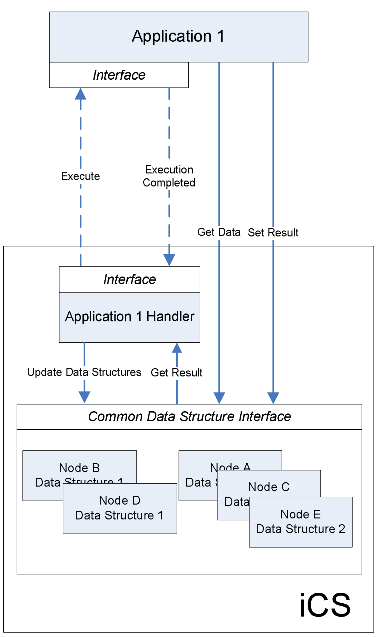

Figure 3 - Application-iCS coupling strategy

Within the iCS an Application Handler is defined. This Application Handler is an object that controls when Applications should be executed and which data is relevant for the execution, as agreed during interface configuration phase. There are as many Application Handlers as different Applications are running simultaneously during the simulation. The iCS also contains a series of data structures. These are tables accessible from SQL sentences that constitute the common ground where the iCS (through the Application Handler) and the Application algorithms exchange information. The Application Handler updates values in these structures and gets the results left by the Applications there. The Application Handler, by using a socket, commands Applications when they must execute their algorithms. Once finished, the Application replies to the Applications Handler indicating that the execution is completed. Then, the new data stored in the Common Data Structure is retrieved from the iCS. For Application developers this approach results in only one requirement: opening an internet-socket on a known IP address and port (defined by the user along with the data needed to properly set-up the Application Handler).

V. SIMULATION FLOW OVERVIEW

This Section explains the procedure defined at iCS level to run the simulation. Such procedure relies on three main iCS capabilities namely, simulation set-up, simulation synchronisation and position update. The use of all these capabilities permits that a complete simulation flow can be defined.

A. *Simulation Set-up*

Before the simulation process itself starts, the iCS is in charge of setting-up the environment by initialising the appropriate objects and starting the rest of the platforms. From a common simulation scenario configuration space, the iCS commands SUMO to initiate with a configuration defined in its domain configuration file. SUMO will be aware of the location of the file in the system through the information sent from the iCS. In a similar way, ns-3 starts running by configuring itself with another file notified by the iCS. To accomplish this task, it is not only necessary to agree on a suitable configuration data structure but also to prepare ns-3 to handle iTETRIS configuration data. This ns-3 enhancement is currently being performed by the iTETRIS consortium. Last but not least, the Applications Handlers and the Applications that will operate decisions about traffic configuration parameters need to be configured and bound to the adequate nodes where they will reside (e.g. vehicle side service, roadside service, etc). At this point the simulators, SUMO and ns-3, are up and Applications are ready to be started.

B. *Event Synchronization*

Simulation platforms, although located in the same computer are autonomous simulation programmes. In order to obtained accurate simulation results, it is important that the simulation events are executed in the correct order and that events taking place at each simulation platform are synchronised and correlated. Such synchronisation is achieved through iCS, which implements specific functions to perform such global event scheduling among simulation platforms and Applications. The number of events happening in the same time frame in the wireless communications domain is usually larger than that of the traffic domain. Currently, SUMO works with a time-step of 1 second whereas ns-3 simulates events in different precision that goes from the second to the femtosecond (10^{-15} seconds). To properly synchronize the iTETRIS simulation, its global time-step must match the time-step of the simulator that constrains most this time leap, in this case SUMO. The iCS, through its Synchronization Manager module, is in charge of controlling the current time-step of the simulation and increasing it correctly. The aim of this module is therefore the delivery of messages to the simulators to trigger on each of them the execution of the events labelled in their respective event lists with the same time-step of the global simulation.

C. Position update.

The iCS, aside of assuring the simulators are correctly synchronized in time, has to guarantee they are also "synchronized in space", that is the simulated objects hold positions which are consistent in the two simulation platforms. This is vital to properly simulate the communication effects that will take place at ns-3 such as fading or interference level experienced. The iCS keeps track of all the vehicles involved in the simulation introducing the notion of the "*iTetrisNode*". This object correctly matches the identifier of a vehicle in SUMO and a node in ns-3, and it also contains the last location of the node. The strategy selected to maintain node location consistency can be described as follows. Whenever SUMO performs its execution, the iCS checks if a vehicle has changed its location and eventually updates this value in its internal representation. Afterwards, this data is forwarded to the correct ns-3 node before the iCS lets it execute the simulation on the wireless communications domain.

VI. CONCLUSIONS

In this paper, an implementation description of the iTETRIS simulation platform has been presented. This tool is engineered for large scale evaluation of cooperative traffic management strategies based on V2X communication systems. To address the essential modularity requirements needed for a fully flexible development of its subsystems, a 3-Block approach has been considered. The iTETRIS Control System (iCS) is the central entity of the platform. It allows an efficient integration of SUMO and ns-3, two widely used open-source simulation platforms for dedicated traffic and communications simulations. iCS ensures the synchronisation of events over the time and the consistency of simulation objects' positions in the two simulation platforms. The paper has demonstrated the steps followed by the iTETRIS architecture to guarantee alignment with the ITS communication architecture defined by ETSI. The modular design of iTETRIS permits a programming language-independent development of each of the platform subsystems, where traffic scientists can autonomously create external traffic applications to be tested over the platform. Efficiency aspects have been discussed in terms of optimum data exchange and intelligent transmission protocols. Sockets technology and ad-hoc data exchange protocols making use of native primitives have been adopted. This will significantly speed up the simulation time and keep inter-platform message exchange data volume and rate to a minimum. Further work will explore the performance obtained over large scale simulation scenarios to characterise the intensity of the data exchange procedures using the primitives presented in this paper.

ACKNOWLEDGMENT

This work has been partly funded by the European Commission through FP7 ICT Project iTETRIS: An Integrated Wireless and Traffic Platform for Real-Time Road Traffic Management Solutions (No. FP7 224644). The authors wish to acknowledge the Commission for their support.

REFERENCES

[1] ETSI TS 102 665 v0.0.9, "Intelligent Transport Systems (ITS); Communications; Architecture", (2009-03).

[2] ETSI TR 102 638 v1.1.1, "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Definitions", (2009-06).

[3] ISO/DIS 21217, "Intelligent transport systems, Communications Access for land mobiles(CALM), Architecture", (2007-12).

[4] Sommer, C., Dressler, F., "Progressing toward realistic mobility models in VANET simulations", Communications Magazine, IEEE 46 (11), 2008, 132-137.

[5] Krajzewicz, D., Teta Boyom, D., Wagner, P.,"Untersuchungen der Performanz einer auf C2C-Kommunikation basierenden, autonomen Routenwahl bei Stauszenarien", In: Heureka '08, 2008-03-05 - 2008-03-06, Stuttgart (Germany).

[6] http://www.davidchappell.com/articles/article_Trouble_CORBA.html. Last visit August 2009..

[7] http://queue.acm.org/detail.cfm?id=1142044. Last visit August 2009.

[8] http://en.wikipedia.org/wiki/SOAP#Advantages-. Last visiti August 2009.

[9] http://www.ibm.com/developerworks/library/ws-pyth9. Last visit August 2009.

[10] http://www.pitch.se/technology/about-hla

[11] Sommer C., Yao Z., German R., and Dressler, "On the need for bidirectional coupling of road traffic microsimulation and network simulation. In *Proceeding of the 1st ACM SIGMOBILE Workshop on Mobility Models*" (Hong Kong, Hong Kong, China, May 26 - 26, 2008). MobilityModels '08. ACM, New York, NY, 41-48.

[12] Queck T., Schuenemann B., Radusch I., "Runtime Infrastructure for Simulating Vehicle-2-X Communication Scenarios", In Proceedings of the Fifth ACM international Workshop on Vehicular inter-Networking, San Francisco, California, USA, September 15, 2008. VANET '08. ACM, pp. 78, New York, NY, USA, ISBN: 978-1-60558-191-0

[13] Piórkowski M, Raya M., et al, "TraNS: Realistic Joint Traffic and Network Simulator for VANETs", SIGMOBILE Mob. Comput. Commun. Rev., Vol 12, No 1, Year 2008. ISSN 1559-1662. Pages 31—33.

[14] Gozálvez J., Turksma S., Lan L., Lazaro O., Cartolano F., Robert E., Krajzewicz D., Bauza R., Filali F., Röckl M., Leguay J., Michelacci C., Vreeswijk J., Maneros J., Gonzalez A. and Lenardi M., "iTETRIS: the Framework for Large-Scale Research on the Impact of Cooperative Wireless Vehicular Communications Systems in Traffic Efficiency", *Information and Communications Technologies (ICT-MobileSubmit 2009)*, June 2009, Santander (Spain).