END-TO-END COMMUNICATION MODELLING FOR LARGE SCALE VEHICULAR NETWORKS USING AADL

Ansar-Ul-Haque Yasar, Naeem Muhammad, Davy Preuveneers and Yolande Berbers

Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3001 Heverlee, Belgium

Abstract. Nowadays novel embedded computing devices enable nodes to form large scale mobile peer-to-peer networks with high communication and computational demands. Inter-vehicular communication is an example of such a large scale flow intensive system where vehicles assist each other to improve the driving experience. Currently, network simulators are being used as a cost effective means to experiment with different inter-vehicle flows of traffic related information, but lack a more analytical approach to identify trade-offs. Architecture Analysis and Design Language *(*AADL) is a modelling formalism that supports architectural based modelling and analysis of functional and non-functional requirements of embedded systems. In our paper, we perform an analysis using AADL for large scale flow intensive vehicular networks. In order to better analyze the trade-off points and optimize the communication in vehicular networks we model end-to-end (EtE) flows with different levels of various in-flows using AADL. Our results show that with the help of AADL flow models, we analyzed the EtE flows in a flow intensive system like large scale vehicular networks at multiple levels of abstraction which is a significant advantage unlike in simulations and also optimize the information dissemination by fine-tuning for the vehicular network quality attributes of interest.

Keywords: Communication modelling, vehicular networks, large scale, AADL.

1 Introduction

The embedding computing technology has already been developed to such an extent that we can collect information from several objects around us. These embedded systems range from consumer electronics such as mobile phones, navigation systems and RFID tags to sophisticated devices such as automobiles and sensor networks. These embedded systems are designed to perform specific computational tasks and have high communication needs. Some of the major constraints for such systems are high reliability, performance, safety and dependability, small memory size, low power and low processing capabilities [1]. Currently, such flow intensive systems are evaluated in a simulated environment which is often more cost efficient than a real life setup. However, due to the black-box nature of a simulation – by setting the simulation initialization parameters and obtaining the output – it is very hard to evaluate the different trade-off points of the system as would be possible with a more analytical approach. Architecture based analysis has gained importance as it helps in analyzing a flow intensive system for these constraints before its implementation and also helps in refactoring a legacy system at a reasonable cost. A number of definitions of software architecture are available in literature, following we list a few of them stated by some well known researchers and organizations working in this domain. According to ISO/IEC 42010 standard [11] software architecture is defined as "*Fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution*".

Vehicular networks are currently an important target of research [2,3] on flow intensive systems. Some of the major application areas include emergency message transmission, collision avoidance, congestion monitoring and intelligent parking space allocation. Optimizing information sharing and retrieval between nodes in large scale dynamic networks is not straightforward, as processing the information *to the right destination at the right time* remains a significant challenge.

In this paper, we use *Architecture Analysis and Design Language* (AADL) for architecture based modelling and analysis of a large scale flow intensive system and apply it to vehicular networks. In order to better analyze the trade-off points and further optimize the communication in vehicular networks (by reducing bottlenecks and communication of irrelevant information) we model end-to-end (EtE) flows using AADL. Using our architecture based approach we can analyze the EtE flows in a flow intensive system like large scale vehicular networks at multiple levels unlike in simple simulated experimentation. We can also optimize the information dissemination by fine-tuning the current and future components in a flow intensive system like large scale vehicular networks.

The rest of the paper is organized as follows: In section 2, we discuss the communication modelling aspects in large scale vehicular networks with a set of requirements and a motivating scenario. A detailed explanation about the AADL flow modelling and analysis for large scale vehicular network with the results obtained is presented in section 3. In section 4, we describe our related work. Last but not the least we present our conclusion with opportunities for future work in section 5.

2 Communication Modelling in Large Scale Vehicular Networks

In this section we discuss the stakeholders involved in large scale vehicular networks and their concerns. We also present our motivating scenario for the architecture based analysis related to our vehicular networks case.

2.1 Large Scale Vehicular Networks: At Large

The vehicle manufacturers worldwide are producing vehicles in a rapidly growing amount. They are equipped with highly efficient, reliable and cheap embedded computing devices due to the increased demand for smart vehicles to ensure a safe and pleasurable driving experience. With these technological advancements in vehicles, more and more mobile vehicular networks will appear, exchanging bulks of information. Some of the major application areas include *emergency message transmission*, *collision avoidance*, *congestion monitoring* and *intelligent parking space allocation.* Currently, the major constraints like reliability, throughput, safety and dependability and overall network performance are analyzed in a simulated environment. However, issues like *what to send, when to send, to whom and how often* are not very clear. There is a need to analyze such flow intensive systems from the architecture's point of view involving various key players.

The major stakeholders involved in the development of large scale vehicular networks from *automobile drivers* to the *infrastructure* as shown in Table 1. Each of these stakeholders has different concerns from *on-time delivery of information* to *cost effectiveness* based on their areas of interest as show in Table 1.

Table 1. Concerns of the stakeholders in a large scale vehicular network.

Concerns of the Stakeholders	Description of the Concerns	Stakeholders
On-time delivery of information	The information from the source should be delivered at the destination under a certain time	Automobile Drivers, Developers, Emergency Response Teams, Infrastructure
Completeness of information	The information is complete if it has all the desired attributes	Developers, System Architects, System Engineers
High throughput of the system	The average rate of successful message delivery in a network	Emergency Response Teams, Government, Infrastructure
Less computational overhead	The system should use least resources for processing a certain task	Systems Architects, Decvelopers, Automobile Industry
Less communication overhead	The redundant and irrelevant communication should be reduced in a network	System Engineers, Automobile Industry
Less power consumption	The system should consume least power possible to process a task	Government, Infrastructure, System Architects, Automobile Industry
High degree of reliability	The information received by a node should be reliable	Automobile Industry, Emergency Response Teams, System Architects, System Engineers
Safety and dependability of the system	The system should behave in the desired manner while processing a task	Infrastructure, Government, System Arhictets, Sytem Engineers
Ease of use and interoperability	The system should be easy to navigate by the users	Automobile Drivers, System Architects, Developers
Cost effectiveness	The system should consume reasonable costs for development and maintainence	Government, Automobile Industry, Automobile Drivers

All the concerns of the stakeholders are important to be considered for an architecture-based analysis and an efficient system development. Later in this paper we will look into the timing concern of the stakeholders using the AADL's flow latency models. These models will elaborate the current EtE flows from the source to the sink (destination) existing in the large scale vehicular networks. A typical scenario of communication between nodes in a flow intensive large scale vehicular network is provided in the next section.

2.2 Motivating Scenario: Deployment of emergency response teams and emergency message dissemination to vehicles in a traffic incident

In this section we describe a motivating scenario in the domain of mobile and scalable context-aware inter-vehicle communication.

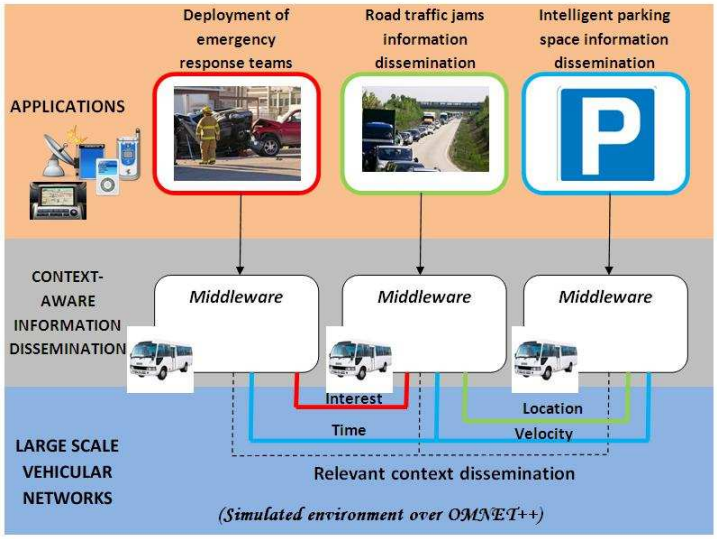


Figure 1. Information dissemination in large scale vehicular networks for context-sensitive tasks.

Deployment of Emergency Response Teams to a traffic incident has always been a crucial point with authorities. Traditionally in case of an incident information is sent to the concerned authorities about the type, location and time of incident over the cellular or wired telephone networks and there might be a road traffic jam at the same place. The problem with the current system is that vehicles are often informed too late and the message itself is usually broadcasted to all the vehicles, also to those that are not in the neighborhood of the accident.

The types of interaction in this scenario can either be in the form or a query or a message, for example;

1. *Notify the location of the accident to the nearby nodes.*
2. *Query for a doctor or emergency response team near to the accident.*
3. *Send emergency signal only to the vehicles travelling towards the accident.*

Let us consider a scenario where an accident occurs between two vehicles travelling from Leuven to Brussels on the highway E40 causing a traffic jam as shown in Figure 1. A vehicle owner on the same side of the road approaching the accident site is being informed about the situation. It sends an acknowledgement to the sending node that it has received the emergency information. The same receiving node forwards this emergency information to the nearby road side unit (RSU), which is connected to the emergency services, and also to its nearest oncoming neighbors. The sender receives feedback from the RSU and the oncoming neighbors that they have received the information and also took the necessary actions required.

In our scenario, we can clearly see that there is one composite flow from the source to the destination and return which completes the communication. This EtE flow has a timing constraint i.e. the information should be reach the destination and back within a certain time interval to remain relevant. The detailed analysis of our case is explained later in the paper.

3 Flow Modelling and Analysis for Large Scale Vehicular Networks

In this section we present a brief background on AADL modelling and analysis. Later in this section we also discuss our flow models and EtE flow analysis for large scale vehicular networks in detail.

3.1 Architecture Analysis and Design Language (AADL)

Architecture Description Languages are modelling languages that provide support for describing system architectures through their formal notations. In this section we briefly describe the AADL architectural elements with tool support with some details about modelling the end-to-end flows using AADL.

3.1.1 Architecture of AADL

AADL consists of a rich set of architectural elements for modelling components, their interactions and their configurations. Components in AADL are used in terms of component types and component implementations. A component type defines the externally visible characteristics of a component usually by using features, flows and properties, whereas a component implementation models the internal structure of the component. An internal structure may consist of subcomponents, connection among them, flows across them and their operational behaviour. AADL distinguishes between three types of components: software components, hardware components and composite (system) components. Further details can be found in [1].

AADL also provides support for various kinds of analyses along with conventional modelling. A few of the supported analyses are; (i) flow latency analysis, (ii) resource consumption analysis, (iii) real-time schedule analysis, (iv) security analysis and (v) safety analysis.

3.1.2 End-to-End Flows with AADL

Flows in AADL describe the different sequences of an information flow through a set of contributing components. The description of this flow is subsequently used in certain analyses such as a flow latency analysis. In AADL, flows are defined with a flow specification and a flow implementation. A flow specification represents the externally visible flow of information in a component; it is specified within the component type declarations using flow sources, flow paths and flow sinks [5]. A flow source represents the originator of the flow, the flow sink represents the end consumer of the flow information, and the flow path embodies the link between incoming and outgoing ports involved in the flow. A flow implementation on the other hand represents the actual realization of a flow within a component; it is specified within the component implementation declarations. Our detailed analysis is presented later in this section.

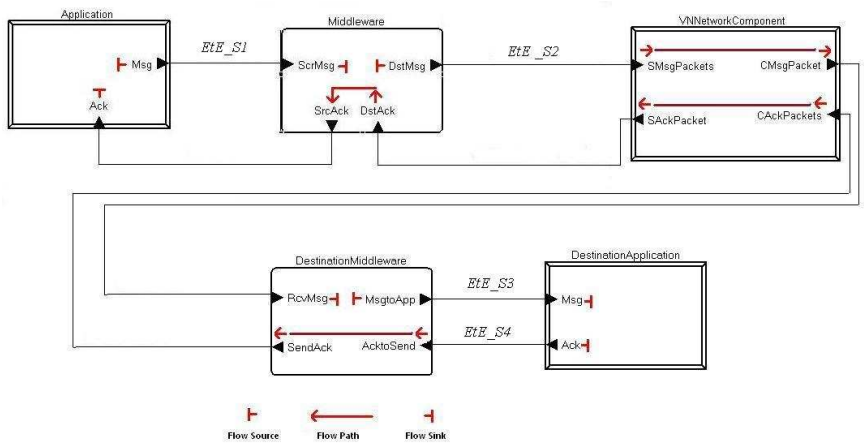


Figure 2. Emergency message dissemination in a large scale vehicular network.

3.2 Flow Models for Large Scale Vehicular Networks

Flows in AADL present the different sequences of an information flow through a set of contributing components. This description is used by various kinds of analyses such as flow latency analysis. *Flow specification* and *flow implementation* define an AADL flow.

```
system VSource
  features
    Msg: out data port;
    Ack: in data port;
  flows
    s2_flow: flow source Msg {
      Latency => 124 Ms;
      };
    s4_flow: flow sink Ack {
      Latency => 0 Ms;
      };
end VSource;
```

(a)

```
system VDestination
  features
    Ack: out data port;
    Msg: in data port;
  flows
    s2_flow: flow sink Msg {
      Latency => 86 Ms;
      };
    s4_flow: flow source Ack {
      Latency => 96 Ms;
      };
end VDestination;
```

(b)

```
device VNetwork
  features
    SMsgPackets: in data port;
    CAckPackets: in data port;
    CMsgPacket: out data port;
    SAckPacket: out data port;
  flows
    s2_flow: flow path SMsgPackets->CMsgPacket {
      Latency => 90 Ms;
      };
    s4_flow: flow path CAckPackets->SAckPacket {
      Latency =>90 Ms;
      };
end VNetwork;
```

(c)

Figure 3. AADL Flow Specification

```
system implementation VSource.VNSource

s2_flow: flow source Middleware.s2_flow->DataConnection2->Msg;

s4_flow: flow sink Ack->DataConnection3->Middleware.s4_flow
        ->DataConnection4->Application.s4_flow;

end VSource.VNSource;
```

(a)

```
system implementation Middleware.CAMiddleware
  s4_flow: flow path DstAck->DataConnection1->MsgFiltering.s4_flow
          ->DataConnection5->SrcAck;
end Middleware.CAMiddleware;
```

(b)

Figure 4. AADL Flow Implementation

A *flow specification* represents the externally visible flow of information in a component; it is specified within the component type declarations using *flow sources, flow paths and flow sinks* [10]. A *flow source* represents the originator of the flow, a *flow sink* represents the end consumer of the flow information, and a *flow path* defines the link between the incoming and the outgoing ports involved in the flow.

A *flow implementation* on the other hand represents the actual realization of a flow within a component; it is specified within the component implementation declarations.

In Figure 3 and 4, for example, both a flow specification and a flow implementation are provided which are based on our scenario in section 2.2. The system consists of three major components working in a peer-to-peer fashion: (i) the application running at a node, (ii) the middleware for the intelligent routing and (iii) the network component responsible for physical communication. An emergency signal (SoS) is generated by the application of a node for other interested nodes in the vicinity. This SoS message after passing through several processing steps inside our middleware reaches the physical network for dissemination. The flow of the SoS message (packet) is shown in Figure 2 along with the textual AADL representation in Figure 3. EtE flow latency analysis requires the specifications of EtE flows. An EtE flow represents a logical flow of information from a source to destination passing through various system components which we will analyze.

In our example of the vehicular networks a SoS message (`EtE_SoS`) dissemination we have four distinct discrete flows (`EtE_S1, EtE_S2, EtE_S3 and EtE_S4`) in our systems as shown in Figure 2. `EtE_SoS` is a composite EtE flow where the flow starts when the application at a node generates a message for the other interested node and later on receives the acknowledgement that the SoS message was delivered.

The ***discrete flows*** of `EtE_SoS` are shown in Figure 2 and explained below;

1. Discrete Flow - EtE_S1:

This flow is started by the application running in an embedded system inside a vehicle in case of an incident. The emergency message is generated by the source application and sent to the source middleware for further processing. This flow starts at the source application and ends at the source middleware. A part of the AADL textual representation is shown in Figure 3 and 4 which represents the timing behavior of the flow.

2. Discrete Flow - EtE_S2:

This flow is started by the source middleware and ends at the destination middleware passing through the network. The details of the network layer are not are discussed in this paper. The source middleware after receiving the message begins the processing step.

The middleware is responsible for filtering irrelevant and redundant information, labeling it for further dissemination based on the context of the information inside the message and finally forward the message to the network layer. The AADL textual representation is shown in Figure 3 and 4 which represents the timing behavior of the flow.

3. Discrete Flow - EtE_S3:

This flow is started by the destination middleware and ends at the destination application. The middleware at the receiving node makes sure that the forwarded information is still relevant for the purpose of the node. The destination middleware extracts the information from the message and forwards to the destination application for further processing. The AADL textual representation is shown in Figure 3 and 4 which represents the timing behavior of the flow.

4. Discrete Flow - EtE_S4:

This flow is started by the destination application and ends at the source application. This flow actually represents an acknowledgement which is sent from the destination node to the source node. The destination application forwards the message to the destination middleware for processing which sends the same information to the application through the network layer and the middleware at the source node. The AADL textual representation is shown in Figure 3 and 4 which represents the timing behavior of the flow.

The flow latency models will enable the system architects to analyze the system flows at a higher abstraction and improve it according to the requirements / concerns of the stakeholders.

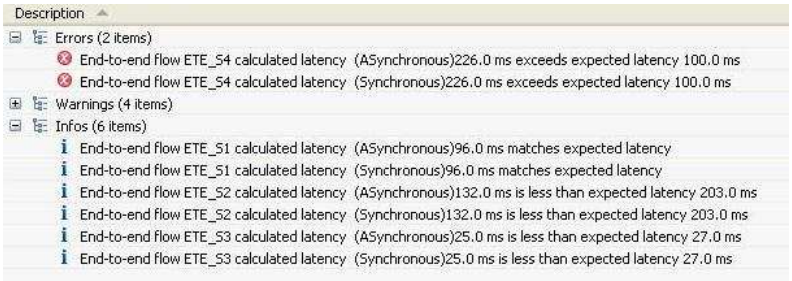
3.3 End-to-End Flow Analysis for Large Scale Vehicular Networks

In this section we will present our composite EtE flow analysis of the emergency signal (SoS) dissemination flow in the large scale vehicular networks scenario. We will identify the timing contributions of system components to the total time consumed by the SoS flow. The analysis is performed using our tool [1] built on-top of the AADL design and analysis tool OSATE.

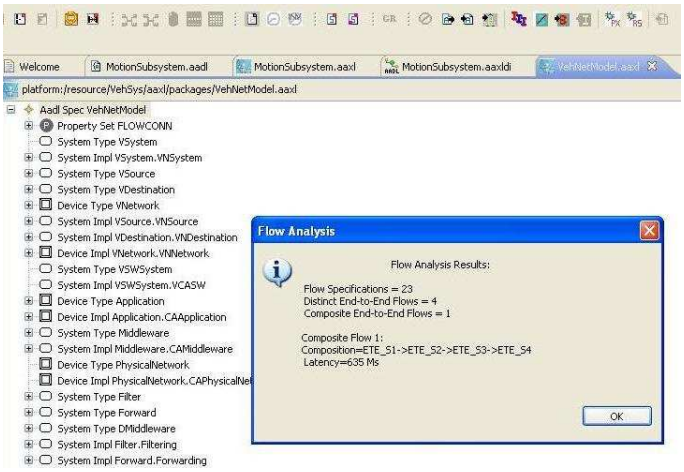
Our tool extends the existing OSATE AADL tool to provide support for modeling and analyzing composite EtE flows. Here we present results of analysis we performed with both OSATE and our tool, Figure 5 contains the output. Communication in a system can analyzed by either using the *Reverse Engineering* or the *Forward Engineering* approach. In the *Reverse Engineering* approach, the latency values are based on the run-time behavior of the system in question. Whereas, our work relies on the foundations of the *Forward Engineering* research domain where the latency values are assumed based on the experience of the systems architects.

The output of the OSATE tool provides insight into the actual and expected EtE flow latencies of the system flows. It compares both latencies and distinguishes flows with same values for both latencies from those having different values. For our example system, we can observe from Figure 5(a) that the actual latency of the *EtE_S1* flow is matching the expected latency of 96 milliseconds (ms). Whereas, it is not true for *EtE_S2* and *EtE_S3* flows, their actual latencies are less than the expected. Moreover, we found that *EtE_S4* is consuming more time than what it should.

The actual latency represents the existing timing behavior of the system flows whereas an expected latency serves as a benchmark value. Expected latencies are identified by the concerned stakeholders. We identified expected latencies of flows in our example system with extensive discussions with the previously mentioned stakeholders of the system. These values are 96, 203, 27 and 100 ms.



(a)



(b)

Figure 5. EtE Flow analysis of an emergency signal dissemination scenario in large scale vehicular networks.

Figure 5(b) describes the composition of the composite EtE flows in our example system. The system contains one such flow, which is composed of 5 distinct EtE flows. The composite flow involves all EtE flows involved in sending the SoS message and receiving the acknowledgment back. In total 23 flow specifications are contributing to this composite EtE flow. The architecture description provides support for modeling and analyzing composite EtE flows whereas it is hard to analyze these flows at such higher abstraction with simulations. The architecture-based analysis also shows that the emergency message dissemination scenario has 4 discrete and 1 composite EtE flow.

The analysis identifies the timing related design decision made by the system engineers while developing it. An insight into these decisions can help in identifying areas of possible improvements in the system. For instance, as observable in Fig, 5(a), *EtE_S4* which is a flow carrying acknowledgment of the actual message is consuming 226 ms, which is more than the time consumed in delivering the actual message that is 132 ms. Acknowledgment time can be reduced by changing the behavior of the involved components, which can be identified by looking into the implementation of this flow. On the other hand, simulation based analyses lack support for recovering such design decisions.

4 Related Work

Oleg and Alexander [6] analyzed EtE timing properties of an AADL model with the help of the Real-Time Calculus (RTC) technique, because of its suitability for analyzing hard real-time timing properties. They proposed an algorithm for transformation of the AADL models to RTC based analytical model. They applied the proposed approach on a wireless sensor network architecture system to analyze the EtE delays. In our work we use AADL and analyze EtE flow latency to optimize the communication between the nodes in large scale vehicular networks.

Lee, Mallet and Simone [7] performed a study to understand how an AADL flow latency model can be represented with the MARTE profile. In this regard, authors discussed mapping of the AADL elements to MARTE elements, and compared models developed with both modelling formalisms. The focus of their work is on modelling and analysis of discrete EtE flows with both AADL and MARTE. Whereas, we distinguished discrete EtE flows from composite EtE flows, and modelled composite flows by linking discrete EtE flows.

Eichler et al. [8] address the issue of optimal information dissemination in vehicular networks. The authors proposed a framework which integrates many of the existing broadcast based strategies that deal with the reduction of the superfluous transmissions. They test their schemes for optimal information dissemination using a network simulator. Our approach uses the idea of architectural level flow modelling and analysis using AADL for optimizing information dissemination in a large scale network.

Nadeem et al. [9] present a formal model of data dissemination in Vehicle Ad-Hoc networks (VANETs). They measure how the performance of data dissemination is affected by bi-directional lane mobility. Three models of data dissemination are explained and simple broadcasting technique is found to be sufficiently enough in their simulated experiments. In our research, we use AADL flow modelling and analysis to identify the bottlenecks in a large scale network. Later we fine tune those components with bottlenecks to improve the information flow in a network.

5 Conclusion and Future Work

Nowadays novel embedded computing devices enable nodes to form large scale mobile peer-to-peer networks with high communication and computational demands. Inter-vehicular communication is an example of such a large scale flow intensive system where vehicles assist each other to improve the driving experience. Currently, network simulators are being used as a cost effective means to experiment with different inter-vehicle flows of traffic related information, but lack a more analytical approach to identify trade-offs. AADL is a modelling formalism that supports architectural based modelling and analysis of functional and non-functional requirements of embedded systems. In our paper, we perform an analysis using AADL for large scale flow intensive vehicular networks. In order to better analyze the trade-off points and optimize the communication in vehicular networks we model end-to-end (EtE) flows with different levels of various in-flows using AADL.

Our architecture-based analysis has several benefits as in comparison to the plain simulation-based experimentation. In our architectural level analysis we have modelled the communication artifacts of a flow intensive system. In our analysis we have analyzed various flows in the large scale vehicular networks at multiple levels which is not possible with simple simulated experimentation. Using our architecture-based modeling and analysis we can easily add an additional component in a flow intensive system like large scale vehicular networks and analyze the behavior in terms of the concerns like throughput, timeliness, cost, etc.

We found the architectural level analysis a more structured way of analyzing the runtime behavior of a system. Furthermore, an architecture description served as a better way of communication among the system stakeholders. They found the description more understandable than the simulation results, which are commonly used for systems like our example case. The analysis also recovered timing related design decisions, which gave a clear insight into the areas of possible improvements in the system on a higher level of abstraction.

We plan to enhance our current research work on modelling composite EtE flows by extending capabilities of the existing AADL connector element. Currently, our technique supports modelling of a single link for each discrete EtE flow. We will, as a part of our future work, look into the multiple EtE flows in a large scale dynamic network. We plan to evaluate other types of models like behavior model, resource model, state model, etc. to improve the performance of a large scale vehicular network. In the last, but not the least, we also plan to optimize communication in other large scale and dynamic networks using AADL by modelling EtE flows.

References

[1] **Naeem, M., Yves, V., Yolande, B. and Sjir, v. L.** *Modelling Embedded Systems with AADL: A Practical Study, New Advanced Technologies, Aleksandar Lazinica (Ed.)*, ISBN: 978-953-307-067-4, INTECH 2010.

[2] **Behera, P. K. and Meher, P. K.** *Prospects of Group-Based Communication in Mobile Ad hoc Networks. In Proceedings of the 4th international Workshop on Distributed Computing, Mobile and Wireless Computing (December 28 - 31, 2002). S. K. Das and S. Bhattacharya, Eds. Lecture Notes In Computer Science*, vol. 2571. Springer-Verlag, London, 174-183 2002.

[3] **Mahajan, R., Zahorjan, J., and Zill, B.** *Understanding wifi-based connectivity from moving vehicles. In Proceedings of the 7th ACM SIGCOMM Conference on internet Measurement, IMC '07*. ACM, New York, NY, 321-326 2007.

[4] **Medvidovic, N. and Taylor, R.N.** *A classification and comparison framework for software architecture description languages, Software Engineering. IEEE Transactions on* , vol.26, no.1, pp.70-93, (2000).

[5] **Feiler, P.H. and Hansson, J.** *Flow Latency Analysis with the Architecture Analysis and Design Language (AADL). Technical Note CMU/SEI-2007-TN-010*, Software Engineering Institute, (2007).

[6] **Oleg, S. and Alexander, C.** *Analysis of AADL Models Using Real-Time Calculus with Applications to Wireless Architectures. Technical Report MS-CIS-08-25, University of Pennsylvania Department of Computer and Information Science*, 2008.

[7] **Su-Young, L., Mallet, F. and de Simone, R.** *Dealing with AADL End-to-End Flow Latency with UML MARTE. Engineering of Complex Computer Systems, ICECCS*. 13th IEEE International Conference on, pp.228-233, 2008.

[8] **Eichler, S., Schroth, C., Kosch, T. and Strassberger, M.** *Strategies for Context-Adaptive Message Dissemination in Vehicular Ad Hoc Networks, In Proceedings of Mobile and Ubiquitous Systems, Annual International Conference on, pp. 1-9, 2006 Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services*, 2006.

[9] **Nadeem, T., Shankar, P. and Iftode, L.** *A comparative study of data dissemination models for vanets. In Proceedings of 3rd International conference on Mobile and Ubiquitous systems*, IEEE, San Jose, 2006.

[10] **Feiler, P.H. and Hansson, J.** *Flow Latency Analysis with the Architecture Analysis and Design Language (AADL), Technical Note CMU/SEI-2007-TN-010*, Software Engineering Institute, 2007.

[11] **ISO.** *ISO/ICE 42010 systems and software engineering – recommended practice for architectural description of software-intensive systems*, 2007.