Towards Modelling 5G Communication in Software architectures of Vehicular CPS

Zenepe Satka*, Saad Mubeen*, John Lundbäck†, Mohammad Ashjaei*,
Hossein Fotouhi*, Masoud Daneshtalab*, Mikael Sjödin*
*Mälardalen University, Västerås, Sweden; firstname.lastname@mdu.se
†Arcticus Systems, Järfälla, Sweden; john.lundback@arcticus-systems.com

***Abstract*—Advanced vehicular Cyber Physical System (CPS) applications like autonomous quarries require high-bandwidth and low-latency communication among the vehicles and their control centers. 5G offers a promising solution to meet these communication requirements. One of the core challenges is how to model and timing analyze distributed software architectures of these large and complex systems that utilize 5G communication. To the best of our knowledge, there is no existing modelling language or a component model that models distributed software architectures of CPS that use 5G communication. In this paper, we take the first step in addressing this gap by proposing extensions to an existing industrial component model for vehicular systems, namely RCM, to support modelling of 5G communication in the distributed software architectures.**

I. Introduction

Vehicular software systems have been drastically increasing in size and complexity for the past few years [1], [2]. This can be attributed to the increasing demand for new software-based features in modern vehicles. Furthermore, these systems are often constrained by real-time requirements that are specified on their software architectures. Hence, the developers of these systems have to deal with the challenges of managing the size and software complexity as well as verify the specified timing requirements at the design time. This makes the development of these systems a daunting task. Model-based Engineering (MBE) and Component-based Software Engineering (CBSE) [3], [4], complemented by timing analysis techniques [5], [6], are proving effective in dealing with these challenges during the development of these systems.

Many industrial CPS use cases in the vehicular domain require the vehicles to communicate with each other to perform a joint functionality, e.g., collaborating vehicles in an autonomous quarry. These vehicles are often equipped with high data-rate sensors (e.g. lidars can transmit data at 100 Mbit/s and more). The large amount of acquired data needs to be processed and then communicated wirelessly to other vehicles with low latencies. 5G offers a promising solution to meet the high-bandwidth and low-latency wireless communication requirement. With the intensive development of 5G [7], it is foreseen that CPS will eventually depend on the availability offered by 5G and beyond to interact with the physical world as well as process and communicate information in an end-to-end manner [8]. Modelling of 5G at the software architecture abstraction is challenging because of its higher communication and protocol complexity compared to wired networks like Switched Ethernet. **One of the core challenges is how to model and timing analyze distributed software architectures of these large and complex systems that utilize 5G communication.**

The existing modelling languages and component models in the vehicular domain, such as EAST-ADL [9], [10], AUTOSAR standard [11], [12], Fraunhofer ESK [13], Rubus Component Model (RCM) [14], to name a few, support modelling and timing analysis of distributed software systems that use only wired (onboard) real-time communication. This communication is often based on low-bandwidth and low latency networks like CAN standard [15] and low-latency and high-bandwidth networks like Ethernet TSN standards [16]. **To the best of our knowledge, there is no existing modelling language or a component model that supports software architecture modelling of distributed software systems that use 5G communication.**

This paper takes the first step in bridging the gap by extending an existing industrial component model for vehicular systems, namely RCM [14], to support modelling of 5G communication. The proposed extensions are comprehensive enough to model the detailed timing information about 5G on the software architectures. In the future, the modelled timing information will be utilized to perform end-to-end timing analysis [6], [17] of the software architectures of vehicular software systems that use 5G communication. The concrete contributions in this paper include:

- We present the system model of 5G communication that is used to introduce an abstract model of 5G that can be incorporated at the abstraction of software architectures.
- We propose extensions to an existing industrial component model, RCM, to support modelling of 5G. The extensions are expressive enough to model all the timing information that is required to perform end-to-end timing analysis of distributed software architectures using 5G.

II. Background

A. 5G Communication

5G is the fifth generation of mobile networks designed to provide higher network speed, higher bandwidth, ultra-low latency, and high reliability. It introduces a new feature called Ultra-Reliable Low-Latency Communication (URLLC) to support real-time applications with latencies as low as 1 ms and reliability levels up to 99.999% [18]. URLLC together with the QoS mechanisms [19] is a promising candidate for wireless communication in time-critical CPS. Furthermore, 5G provides support for enhanced mobile broadband (eMBB), and massive machine-type communications (mMTC) [20]. The eMBB provides high data rates, higher user mobility, and high density to support a wide range of high-bandwidth applications. Fig. 1 depicts the features and services supported by 5G [21].

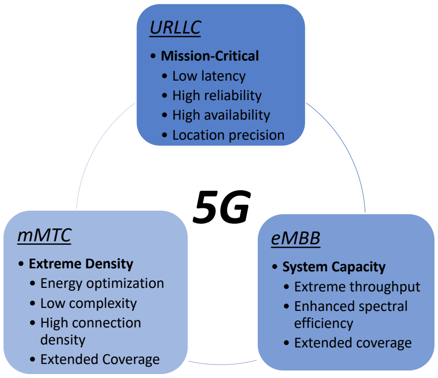

Fig. 1: Features and services provided by 5G.

B. Rubus Component Model (RCM)

Rubus, developed by Arcticus Systems[1], is a collection of models and tools for model-driven development of time-critical software systems [14]. The modelling language and tool suite supported by Rubus are called RCM and Rubus-ICE respectively. Rubus also provides a real-time operating system that is certified according to ISO 26262 safety standard. Rubus has been used in the vehicle industry for over 30 years by several OEMs and Tier-1 companies, e.g., Volvo Construction Equipment, BAE Systems Hägglunds, Hoerbiger, Knorr Bremse, to mention a few.

In RCM, the highest-level hierarchical element is called the *system*, which contains one or more models of nodes or Electronic Control Units (ECUs) and zero or more models of networks as shown in Fig. 2. The *network* object consists of two parts: one is independent of the underlying communication protocol, while the other is protocol dependent. The protocol-independent part defines messages, their properties, and data/signals that are mapped to them. The protocol-dependent part of the network is defined for each individual communication protocol separately.

The lowest-level hierarchical element in RCM is called the *Software Circuit (SWC)*, which encapsulate a software function. The interaction between the SWCs is clearly separated in terms of data (via data ports) and control flows (via control ports) as shown in Fig. 2. One important principle in RCM is to separate functional code from the infrastructure that implements the execution model. This, in turn, facilitates visualisation of explicit synchronisation and data access at the abstraction of software architectures.

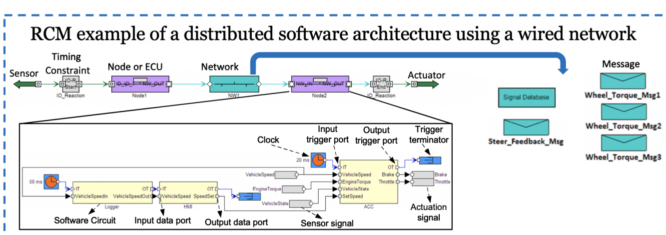

Fig. 2: RCM model of a distributed software architecture.

Today, Rubus supports modelling and end-to-end timing analysis of distributed software architectures that employ onboard (wired) networks like CAN [22] and TSN [23]. Similar to other modelling languages in the vehicular domain like AUTOSAR, EAST-ADL, and AMALTHEA, RCM does not support modelling and end-to-end timing analysis of 5G communication in distributed software architectures.

[1] https://www.arcticus-systems.com

III. Model of End-to-end 5G Communication

This section presents an abstract model of end-to-end communication in 5G. This model will provide input for extensions to RCM. We consider the 5G system to consist of a single-antenna or general Node B (gNB) using mid-band frequencies and a channel bandwidth of 20MHz. The end-to-end connection in the 5G network is shown in Fig. 3, where the user equipment 1 (UE 1) is the transmitter, while UE 2 is the receiver. UE 1 uses the uplink (UL) transmission to send data to the 5G network via the gNB, while UE 2 uses the downlink (DL) to receive the data via the gNB.

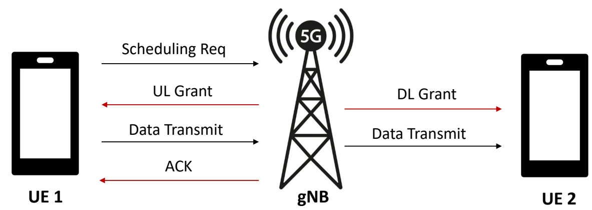

Fig. 3: End-to-end connection in 5G network.

To establish an uplink transmission, the 5G network follows some procedures which are simplified below:

1) *Scheduling Request*: The user sends a scheduling request via the control channel to the gNB asking for a specific amount of data to be sent with some level of priority. In this request, the user also sends the Channel Quality Indicator (CQI) value which is an indicator of the channel quality considering the signal strength and different interference components existing on the radio side.
2) *Uplink (UL) Grant*: The gNB evaluates the request and sends an uplink grant to the user considering its CQI value. This grant holds information related to the specific time and frequency slot for the uplink transmission, the appropriate Modulation Coding Scheme (MCS), and the coding rate used by the user when transmitting the data.
3) *5G Data Transmission*: The user takes the grant from the gNB and starts the uplink transmission using the allocated time and frequency slot.
4) *Acknowledgement (ACK)*: The gNB sends a positive ACK to the user in cases where the transmission is done correctly and no errors were encountered, otherwise it will send a negative ACK.

Once the data is transmitted successfully to the network, the gNB identifies the receiver and starts establishing a downlink transmission as follows:

1) *Downlink (DL) Grant*: The gNB allocates the required radio resources and sends a downlink grant to the receiver with information about the allocated time and frequency as well as the MCS and coding rate to be used for the downlink transmission.
2) *5G Data Transmission*: After the needed resources are allocated, the gNB starts the downlink data transmission.

Higher CQI value results in the assignment of higher modulation schemes in the uplink or downlink grant, which means a higher amount of data can be sent/received within the allocated time slot as also shown in Fig. 4.

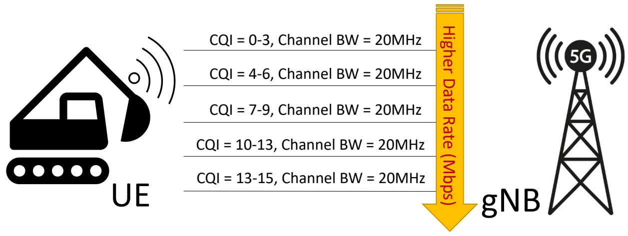

Fig. 4: Data rates values depending on the CQI value.

IV. Proposed Extensions to RCM

5G communication in RCM is modelled by extending the protocol-dependent part of the existing network object (see Fig. 2). The extensions include new properties of the protocol-dependent part corresponding to 5G communication as well as interfaces for the device side for uplink and downlink transmission. Fig. 5 shows the abstract model of end-to-end 5G connection in RCM.

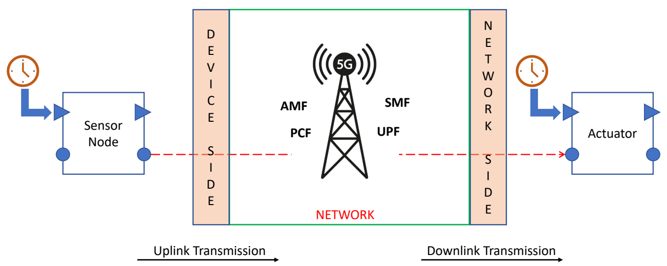

Fig. 5: Abstract model of end-to-end 5G connection in RCM.

User Equipment (UE) Model in RCM: The UE can be viewed as the sensor node in Fig. 5. In RCM, the UE is modelled with the existing node element (see Fig. 2) that can send or receive messages over the 5G network object.

gNB Model in RCM: The gNB is modelled as a special type of switch element in RCM. This element contains a set of communication channels. Depending on the user-requested QoS-es and the received CQI value, the gNB assigns one of the channel IDs which fulfills the user's requirements. The properties of the gNB model in RCM are shown in Fig. 6.

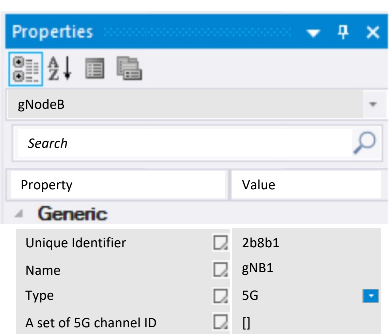

Fig. 6: gNB properties in RCM.

Scheduling Request Model in RCM: The scheduling request in RCM is modelled as a special type of message (see Message element in Fig. 2). This special message has additional properties that are depicted in Fig. 7. This message is sent by the user to notify the gNB about the requested Quality of Services (QoS) and the size of the data payload in bytes. The properties associated to this message include "Transmission Type" (Periodic or Sporadic), "Size of Data Payload" in bytes, a "CQI" value, "Priority" and "Deadline".

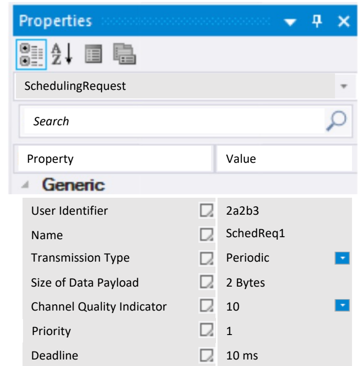

Fig. 7: Scheduling Request properties in RCM.

5G Channel Model in RCM: The properties of the 5G channel model in RCM are presented in Fig. 8. There are two types of channels in 5G: control channels and data channels. The control channels are used to transmit control information such as the Scheduling Request or the Configured Grant. On the other hand, the data channels are used to transmit the data payload on the uplink side or downlink side. A channel is assigned to each connected user with a specific Modulation Coding Scheme (MCS) which is related to the number of bytes that a device can transmit using this channel. Depending on the channel quality information, modulation scheme and coding rate, there is a calculated data rate in Mbit/s for each channel.

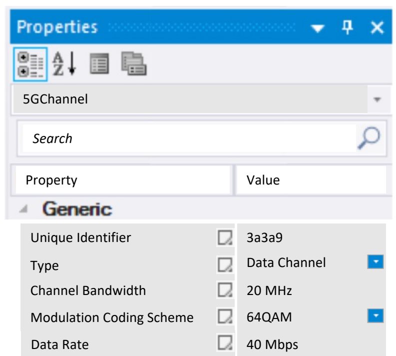

Fig. 8: 5G channel properties in RCM.

Configured UL/DL Grant Model in RCM: The configured grant contains control information notifying a specific user about the channel ID which can be used for data transmission. The channel ID itself contains a set of characteristics to ensure the requested QoS for the uplink/downlink transmission. The properties of configured grant in RCM are presented in Fig. 9.

5G Data Transmission Model in RCM: The data transmission contains all the 5G message properties: (i) the transmission type which can be periodic or sporadic; (ii) the data rate in Mbit/s also known as the network speed for the channel being used; (iii) a priority level of the message; (iv) a deadline constraint which needs to be met by the 5G network, and (v) the size of the payload in bytes. These properties are modelled in RCM as depicted in Fig. 10.

Acknowledgement (ACK) Model in RCM: In Rubus, the acknowledgment is a simple control message, sending a

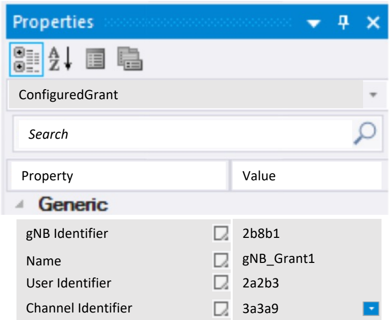

Fig. 9: Configured Grant properties in RCM.

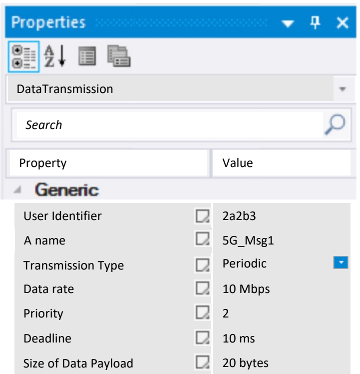

Fig. 10: Data Transmission properties in RCM.

positive integer in cases where the transmission is done correctly and no errors were encountered, otherwise, it sends a zero to notify the user that an error was encountered.

V. Summary and Ongoing Work

The high-bandwidth and ultra low-latency communication supported by 5G has the potential to support real-time communication in many time-critical CPS. The work in this paper presents the first step to address the core challenge of modelling 5G communication in distributed software architectures of complex CPS in the vehicular domain. In this regard, we proposed extensions to an industrial component model for vehicular systems, RCM, to support modelling of 5G communication. The extensions are designed to be comprehensive enough to model the detailed timing information about end-to-end 5G connection on the software architectures. The modelled timing information will be used in the future to perform end-to-end timing analysis of the distributed software archtiectures.

Currently, we are in the process of validating the proposed extensions to RCM by modelling distributed software architecture of a large industrial use case of an autonomous quarry. The outcomes of modelling the use case, corresponding discussions and usability feedback from the vehicle manufacturer will be used to refine the proposed extensions to RCM. In the future, we plan to develop an end-to-end timing analysis of 5G based distributed software systems and support the application of the analysis at the software architecture abstraction.

Acknowledgement

This work is supported by the Swedish Governmental Agency for Innovation Systems (VINNOVA) via the DESTINE, PROVIDENT & INTERCONNECT projects.

References

[1] J. Schroeder, C. Berger, A. Knauss, H. Preenja, M. Ali, M. Staron, and T. Herpel, "Predicting and evaluating software model growth in the automotive industry," in *IEEE International Conference on Software Maintenance and Evolution*, Sep. 2017, pp. 584–593.

[2] L. Lo Bello, R. Mariani, S. Mubeen, S. Saponara, "Recent advances and trends in on-board embedded and networked automotive systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, 2019.

[3] T. Vale, I. Crnkovic, E. S. de Almeida, P. A. da Mota Silveira Neto, Y. C. Cavalcanti, and S. R. de Lemos Meira, "Twenty-eight years of component-based software engineering," *Journal of Systems and Software*, vol. 111, pp. 128 – 148, 2016.

[4] T. A. Henzinger, J. Sifakis, "The Embedded Systems Design Challenge," in *14th International Symposium on Formal Methods*, 2006.

[5] B. Akesson, M. Nasri, G. Nelissen, S. Altmeyer, and R. Davis, "A comprehensive survey of industry practice in real-time systems," *Real-Time Systems*, vol. 58, 09 2022.

[6] N. Feiertag, K. Richter, J. Nordlander, J. Jonsson, "A Compositional Framework for End-to-end Path Delay Calculation of Automotive Systems under Different path Semantics," in *Workshop on Compositional Theory and Technology for Real-time Embedded Systems*, 2008.

[7] Z. Satka, M. Ashjaei, H. Fotouhi, M. Daneshtalab, M. Sjödin, and S. Mubeen, "A comprehensive systematic review of integration of time sensitive networking and 5g communication," *Journal of Systems Architecture*, vol. 138, 2023.

[8] C. Tranoris, S. Denazis, L. Guardalben, J. Pereira, and S. Sargento, "Enabling Cyber-Physical Systems for 5G Networking: A Case Study on the Automotive Vertical Domain," in *4th International Workshop on Software Engineering for Smart Cyber-Physical Systems*, 2018.

[9] "EAST-ADL Domain Model Specification, V2.1.12,," http://www.east-adl.info/Specification/V2.1.12/EAST-ADL-Specification_V2.1.12.pdf.

[10] R. T. Kolagari, D. Chen, A. Lanusse, R. Librino, H. Lönn, N. Mahmud, C. Mraidha, M.-O. Reiser, S. Torchiaro, S. Tucci-Piergiovanni, T. Wägemann, and N. Yakymets, "Model-based analysis and engineering of automotive architectures with east-adl: Revisited," *Int. J. Concept. Struct. Smart Appl.*, vol. 3, no. 2, pp. 25–70, 2015.

[11] The AUTOSAR Consortium, "Autosar technical overview," in *Version 4.3.*, May 2016, http://autosar.org.

[12] S. Fürst and M. Bechter, "Autosar for connected and autonomous vehicles: The autosar adaptive platform," in *46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W)*, Jun. 2016, pp. 215–217.

[13] Fraunhofer ESK, Safe Adaptive Software for Fully Electric Vehicles, https://www.iks.fraunhofer.de/en/projects/safeadapt.html.

[14] S. Mubeen, H. Lawson, J. Lundbäck, M. Gålnander, and K.-L. Lundbäck, "Provisioning of predictable embedded software in the vehicle industry: The rubus approach," in *4th IEEE/ACM International Workshop on Software Engineering Research and Industrial Practice*, 2017.

[15] ISO 11898-1, "Road Vehicles–interchange of digital information–controller area network (CAN) for high-speed communication, ISO Standard-11898, Nov 1993."

[16] Time-Sensitive Networking (TSN) Task Group, TSN standards, https://1.ieee802.org/tsn.

[17] M. Becker, D. Dasari, S. Mubeen, M. Behnam, and T. Nolte, "End-to-end timing analysis of cause-effect chains in automotive embedded systems," *Journal of Systems Architecture*, vol. 80, pp. 104 – 113, 2017.

[18] M. Bennis, M. Debbah, and H. V. Poor, "Ultra-Reliable and Low-Latency Wireless Communication: Tail, Risk, and Scale," *Proceedings of the IEEE*, vol. 106, no. 10, pp. 1834–1853, 2018.

[19] Z. Satka, M. Ashjaei, H. Fotouhi, M. Daneshtalab, M. Sjödin, and S. Mubeen, "QoS-MAN: A Novel QoS Mapping Algorithm for TSN-5G Flows," in *28th IEEE RTCSA Conference*, 2022.

[20] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A Survey on 5G Networks for the Internet of Things: Communication Technologies and Challenges," *IEEE Access*, vol. 6, 2018.

[21] Mallinson, K. 3GPP-The Path to 5G: As much Evolution as Revolution. The Mobile Broadband Standard. [Online]. Available http://www.3gpp.org/news-events/3gpp-news/1774-5g_wiseharbour.

[22] S. Mubeen, J. Mäki-Turja, and M. Sjödin, "Communications-oriented development of component-based vehicular distributed real-time embedded systems," *Journal of Systems Architecture, Vol. 60 (2)*, 2014.

[23] S. Mubeen, M. Ashjaei, and M. Sjödin, "Holistic modeling of time sensitive networking in component-based vehicular embedded systems," in *45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2019, pp. 131–139.