Towards Interactions with Augmented Reality Systems in Hyper-Connected Cars

Ovidiu-Andrei Schipor and Radu-Daniel Vatavu

MintViz Lab, MANSiD Research Center
University Ştefan cel Mare of Suceava
`schipor@eed.usv.ro, radu.vatavu@usm.ro`

Abstract. Hyper-connected cars can store, process, and share a large amount and variety of digital content, which creates opportunities for using high-definition Augmented Reality (AR) and live video streaming to enhance current in-vehicle driving assistance and navigation systems. However, several challenges must be overcome to make such systems viable and efficient, such as dealing effectively with a variety of smart devices, platforms, and in-vehicle standards and technologies or delivering dynamic digital content to users in interactive time. In this paper, we propose a solution to these challenges by *modeling the smart car as a distinct type of a smart environment.* This model enables us to introduce a five-layer software architecture proposal based on Euphoria, a recent high-performing event-driven software architecture design for supporting effective communications between heterogeneous I/O devices in generic smart environments. We discuss the ways in which Euphoria can provide effective solutions to our identified challenges and hope that our contributions will stimulate interesting discussions towards defining a practical roadmap of engineering interactions with AR systems and high-definition video for hyper-connected cars.

Keywords: Hyper-connected cars · Augmented reality · Smart devices · Software architecture · High-definition video · Challenges.

1 Introduction and Context

The automotive industry has witnessed a fast transition to the concept of the *car as a software-driven electronics device* to support consumers' needs for more services [3, 10, 18]. Examples of applications that demand complex in-vehicle software infrastructure are autonomous driving, connectivity to wireless networks and to the Internet, to road infrastructure and to other vehicles, but also sharing data with smart devices, such as smartphones, smartwatches, smartglasses, etc. [1, 3, 4, 19, 28]. No more than ten years ago, the premium cars of the year 2010 embedded an impressive number of about 100 microprocessor-based Electronic Control Units (ECUs) that were running 100 million lines of software code [4]. Since then, these figures and the complexity of the corresponding in-vehicle software and systems have increased considerably [8] towards the 1.7 Gbps hyper-connected car. This trend has created an urgent need for software architectures to deal effectively with the complex systems that smart cars turned into as well as with the amount of data that hyper-connected cars can create, process, and share with other smart devices.

Before we move on, we briefly overview the concept of a hyper-connected car. This term describes a vehicle that is (i) part of the Internet-of-Things [9] and (ii) capable to initialize and maintain communications with other entities that deliver relevant and useful information [11]. To this end, hyper-connected cars require dedicated hardware and software architecture designs [3, 8, 10, 26]. For example, a hyper-connected car should be able to act both as a consumer and producer within its informational ecosystem; it must be able to deliver relevant information to other entities and, at the same time, to be aware of any changes occurring in its environment [11, 13]; moreover, hyper-connected cars adhere to the vision of "vehicle-to-everything" (V2X) communications, which emerges by putting together specific types of communications designed for smart vehicles, such as vehicle-to-infrastructure (V2I), vehicle-to-vehicle (V2V), vehicle-to-network (V2N), vehicle-to-pedestrian (V2P), and vehicle-to-device (V2D). To enable such advanced features, designers of hyper-connected cars must firstly redefine the way in which the in-vehicle components interact with each other by assuring high decoupling between hardware and software components via the high-speed in-vehicle Ethernet network. On this foundation, new trends and innovations in the automotive industry, such as the application of Augmented Reality (AR) concepts and technology to enhance the driving experience [6], can take off effectively into mass production and consumer cars.

2 Challenges for AR in Hyper-Connected Cars

The features envisioned for AR and the hyper-connected car basically reformulate the meaning of *what a car is* and *how the interactions between the car and its users are modeled* towards effective design and development of in-vehicle software applications. In this position paper, *we model the hyper-connected car as a specific kind of a smart environment*, where the various in-vehicle modules, users of the car, and other devices (*e.g.*, smartphones, tablets, smartwatches, etc.) that co-inhabit the smart environment of the hyper-connected car create, process, and render AR content. In this section, we present a series of challenges in relation to practical use case scenarios represented by see-through displays and cars [2, 17, 20, 21, 27]. In these scenarios, the driver's visual field is extended to cover other objects, including pedestrians and other cars, that normally would not be visible directly. The virtual 360-degree model around the vehicle is generated based on data from video cameras, proximity sensors, road infrastructure, or other cars and delivered by an AR system and display. The goal of such systems is to reduce the risk of road accidents by extending the range of visual information available to the driver during difficult or complex maneuvers, *e.g.*, overtaking other vehicles, parking, etc. Thus, AR can play a key role in assisting drivers with enhanced navigation information, while providing enriched entertainment experiences to passengers [12]. We identify the following challenges for delivering AR content inside the hyper-connected car:

1. *Dynamically generated AR content.* In contrast to locally stored multimedia files or video streams from Internet repositories, AR content is generated on the fly by using data from various hardware and software components, devices, and services. This feature relies on continuous synchronization between virtual models and the physical world [2, 20] and requires the existence of a chain of specialized modules that interact with each other to deliver an augmented version of the visual reality. The virtual models are continuously adapted, which creates the challenge of how to enable flexible associations between the components of the virtual model and the modules of the system.
2. *Managing a wide variety of in-vehicle technologies.* AR capabilities are tightly coupled with a wide variety of technologies [3, 25], such as object, car, and pedestrian detection and recognition under complex conditions, which necessitate robust computer vision algorithms; mobility services and inter-vehicle communications demanding high-speed networks and efficient data transmission protocols; and distance estimation, emergency breaking, or engine control requiring real-time processing capabilities. An effective software architecture for creating and delivering AR content in a hyper-connected car should enable unobtrusive running of such complex modules and technologies, skilfully interweaving engine operation, in-vehicle entertainment services, connections to smart devices and services from the cloud, and safety-related modules [8, 25]. A recent trend in the community advocates for a complete rethinking of the architectural approach in the automotive industry towards highly scalable and inter-operable approaches [3, 15, 24, 25].
3. *A network of heterogeneous components.* Hyper-connected cars rest on the principle of high abstraction of their hardware and software components so that heterogeneous entities can be easily integrated. Instead of myriad one-to-one communications between the various components, the new approach proposes a decoupled architecture where modules become hosts within a high-speed Ethernet network [5, 10, 26, 28]. In fact, the automotive industry has gradually increased the level of abstraction, from microcontrollers to systems-on-chip and to generic scalable computing platforms [8, 26]. The decoupling between production and the utilization of processing power is one key step towards new architectures. By moving from hardware-centric to function-centric approaches, software applications need to be rethinked as well [3, 8, 26]. However, this trend also comes with several challenges, since it heavily relies on asynchronous communications between heterogeneous devices and software modules.

3 Towards a Solution for Effectively Delivering AR Content in Hyper-Connected Cars

In this section, we relate to recent advances in software architecture design for generic smart environments, such as the event-driven Euphoria architecture [22], specifically designed to deal effectively with application scenarios in which many

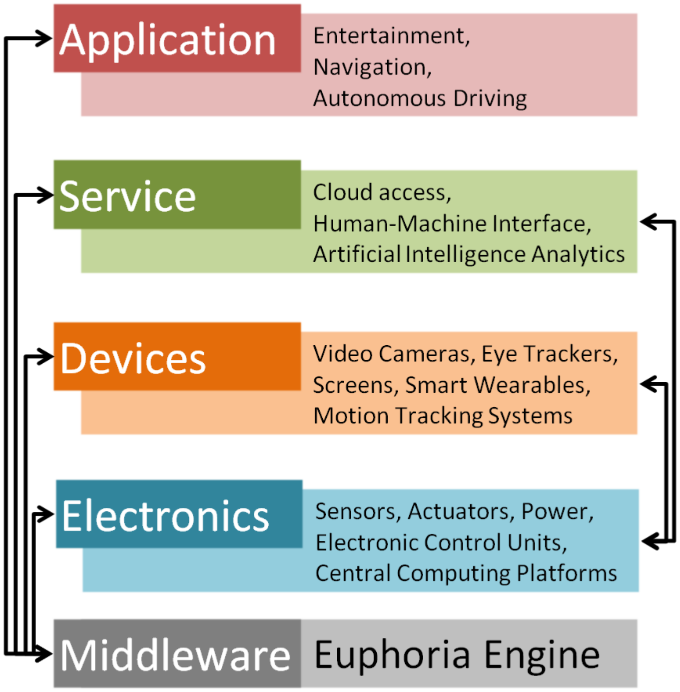

Fig. 1. Integration of the Euphoria engine [22], depicted in this figure as the Middleware layer, with the standard four-layer infrastructure of the connected car.

heterogeneous I/O devices communicate and inter-operate towards a common goal, such as understanding users' intentions and reacting in consequence [23]. To this end, we adopt the model of a smart car as a smart environment to introduce a software architecture for hyper-connected cars that (1) connects to the four main layers designed by previous work for connected cars [3, 14, 15, 26, 28] and (2) is inspired from and reuses recent software architecture designs for smart environments [22]. Specifically, we conduct our discussion around the Euphoria architecture, a generic event-driven middleware for engineering interactions in smart environments between heterogeneous entities; see Schipor *et al.* [22] for technical details, evaluations of performance, and application examples. Euphoria rests on ten design criteria: two handling techniques (*i.e.*, event-driven and asynchronous processing), four quality features that characterize how entities interact with each other (*i.e.*, adaptability, modularity, flexibility, and interoperability), and four contextual properties linked to a specific choice of technology (*i.e.*, web-based, open-source, smart space orientation, and JavaScript). We propose to employ Euphoria to mediate non-critical interactions between in-vehicle systems, as it is Ethernet-based and implements the abstraction of hardware and software modules via JSON configuration files.

Figure 1 illustrates the integration of Euphoria with the four main layers of an in-vehicle architecture: Application, Service, Devices, and Electronics. The

Application layer consists of all the software modules that provide interfaces for the in-vehicle occupants, forming the IVIS (In-Vehicle Infotainment) and ADAS (Advanced Driving Assistance) systems [7, 16]; entertainment, navigation, and autonomous driving applications rely on the Service layer; and the in-vehicle hardware is addressed by the Devices and Electronics layers, respectively. While Devices relate to high-level components with which users can interact directly, Electronics comprise the processing and power circuits of the vehicle. Each layer contains a dedicated adapter module to interface the Euphoria middleware. However, critical, real-time, and safety-related functions benefit of direct informational shortcuts illustrated by arrows in Figure 1. We suggest Euphoria as an effective solution to address the challenges mentioned previously, as follows:

1. *Euphoria can handle dynamically generated content.* The entities, *i.e.*, devices and software modules, that are connected through Euphoria exchange standardized messages that contain identification parameters. Each entity can therefore know what component of the AR model needs to be processed or updated. Moreover, the associations between the various components of the virtual model and the modules of the system can be dynamically at the level of the parameters included in the headers of those messages. This aspect is particularly important when content is not static, but dynamically generated and modified by the various components of the system.
2. *Euphoria can integrate a wide variety of devices, platforms, and technologies.* By design, Euphoria adheres to common Internet standards and protocols for data processing and exchange, *e.g.*, HTTP, WebSockets, and JSON, that are implemented by virtually all operating systems and programming languages. Existing embedded systems, devices, and software modules can be integrated within Euphoria simply by exposing their public interface.
3. *Euphoria can manage networks of heterogeneous hardware and software components.* By design, Euphoria is flexible to integrate heterogeneous hardware and software modules and components. The registration and notification mechanisms implemented in Euphoria treat both hardware and software modules similarly so that the actual implementation of a component is transparent for developers and end-users.

Future work will consider technical evaluation of the Euphoria software architecture to implement Augmented Reality and live high-definition video application scenarios in hyper-connected cars with specific focus on performance measurements, such as the request-response time [22] for transmitting high-definition digital content using the in-vehicle Wi-Fi network.

4 Acknowledgments

This work was supported by a grant of the Romanian Ministry of Research and Innovation, CCCDI-UEFISCDI, Complex Project no. PN-III-P1-1.2-PCCDI-2017-0917, contract no. 21PCCDI/2018 (project P2), within PNCDI III.

References

[1] Aiordachioae, A., Vatavu, R.D., Popovici, D.M.: A design space for vehicular lifelogging to support creation of digital content in connected cars. In: Proceedings of EICS '19, the 11th the ACM SIGCHI Symposium on Engineering Interactive Computing Systems. ACM, New York, NY, USA (2019)

[2] Alhaija, H.A., Mustikovela, S.K., Mescheder, L., Geiger, A., Rother, C.: Augmented reality meets computer vision: Efficient data generation for urban driving scenes. International Journal of Computer Vision **126**(9), 961–972 (2018)

[3] Burkacky, O., Deichmann, J., Doll, G., Knochenhauer, C.: Rethinking car software and electronics architecture. McKinsey & Co., February (2018), `https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/rethinking-car-software-and-electronics-architecture`

[4] Charette, R.N.: This car runs on code. IEEE Spectrum **46**(3), 3 (2009), `https://spectrum.ieee.org/transportation/systems/this-car-runs-on-code`

[5] Corbett, C., Schoch, E., Kargl, F., Preussner, F.: Automotive Ethernet: Security opportunity or challenge? Lecture Notes in Informatics (LNI) (2016)

[6] Davies, C.: Your car windshield could be an AR hologram in 2020 (2018), `https://www.slashgear.com/porsche-hyundai-wayray-augmented-reality-holographic-windshield-investment-18546322`

[7] Engen, T., Lervag, L.E., Moen, T.: Evaluation of IVIS/ADAS using driving simulators. comparing performance measures in different environments. In: Proceedings of the 16th ITS World Congress and Exhibition on Intelligent Transport Systems and Services (2009)

[8] Gai, P., Violante, M.: Automotive embedded software architecture in the multi-core age. In: Proceedings of the 21th IEEE European Test Symposium (ETS). pp. 1–8. IEEE (2016)

[9] Gerla, M., Lee, E.K., Pau, G., Lee, U.: Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds. In: Proceedings of the IEEE World Forum on Internet of Things (WF-IoT). pp. 241–246. IEEE (2014)

[10] GreenCarCongres: Cisco and Hyundai to bring "hyper-connected" car to production in 2019; software defined vehicle architecture (2018), `https://www.greencarcongress.com/2018/01/20180113-cisco.html`

[11] Karnouskos, S., Kerschbaum, F.: Privacy and integrity considerations in hyper-connected autonomous vehicles. Proceedings of the IEEE **106**(1), 160–170 (2018)

[12] Lim, C., Choi, J., Park, J.I., Park, H.: Interactive augmented reality system using projector-camera system and smart phone. In: Proceedings of the International Symposium on Consumer Electronics (ISCE). pp. 1–2. IEEE (2015)

[13] Lu, N., Cheng, N., Zhang, N., Shen, X., Mark, J.W.: Connected vehicles: Solutions and challenges. IEEE Internet of Things Journal **1**(4), 289–299 (2014)

[14] Marisetty, S., Srivastava, D., Hoffmann, J., Starks, B.: Low power Intel architecture platform for in-vehicle infotainment. Intel Technology Journal **13**(1) (2009)

[15] Milosevic, M., Bjelica, M.Z., Maruna, T., Teslic, N.: Software platform for heterogeneous in-vehicle environments. IEEE Transactions on Consumer Electronics **64**(2), 213–221 (2018)

[16] Okuda, R., Kajiwara, Y., Terashima, K.: A survey of technical trend of ADAS and autonomous driving. In: Technical Papers of 2014 International Symposium on VLSI Design, Automation and Test. pp. 1–4. IEEE (2014)

[17] Olaverri-Monreal, C., Gomes, P., Fernandes, R., Vieira, F., Ferreira, M.: The see-through system: A VANET-enabled assistant for overtaking maneuvers. In: Proceedings of the IEEE Intelligent Vehicles Symposium. pp. 123–128. IEEE (2010)

[18] Pelliccione, P., Knauss, E., Heldal, R., Ågren, S.M., Mallozzi, P., Alminger, A., Borgentun, D.: Automotive architecture framework: The experience of Volvo cars. Journal of Systems Architecture **77**, 83–100 (2017)

[19] Petit, J., Shladover, S.E.: Potential cyberattacks on automated vehicles. IEEE Transactions on Intelligent Transportation Systems **16**(2), 546–556 (2015)

[20] Qiu, H., Ahmad, F., Govindan, R., Gruteser, M., Bai, F., Kar, G.: Augmented vehicular reality: Enabling extended vision for future vehicles. In: Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications. pp. 67–72. ACM (2017)

[21] Rameau, F., Ha, H., Joo, K., Choi, J., Park, K., Kweon, I.S.: A real-time augmented reality system to see-through cars. IEEE Transactions on Visualization and Computer Graphics **22**(11), 2395–2404 (2016)

[22] Schipor, O.A., Vatavu, R.D., Vanderdonckt, J.: Euphoria: A scalable, event-driven architecture for designing interactions across heterogeneous devices in smart environments. Information and Software Technology **109**, 43–59 (May 2019)

[23] Schipor, O.A., Vatavu, R.D., Wu, W.: Sapiens: Towards software architecture to support peripheral interaction in smart environments. Proceedings of the ACM on Human-Computer Interaction (3), Article no. 11 (June 2019)

[24] Sharma, K.K., Sharma, H., Ramani, A.: mCAR: Software framework architecture for in-vehicle pervasive multimedia services. In: Proceedings of the International MultiConference of Engineers and Computer Scientists. vol. 1 (2009)

[25] Staron, M.: Automotive software architectures: An introduction. Springer (2017)

[26] Traub, M., Maier, A., Barbehön, K.L.: Future automotive architecture and the impact of it trends. IEEE Software **34**(3), 27–32 (2017)

[27] Yuan, B., Chen, Y.A., Ye, S.: A lightweight augmented reality system to see-through cars. In: Proceedings of the 7th International Congress on Advanced Applied Informatics (IIAI-AAI). pp. 855–860. IEEE (2019)

[28] Zheng, B., Liang, H., Zhu, Q., Yu, H., Lin, C.W.: Next generation automotive architecture modeling and exploration for autonomous driving. In: Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI). pp. 53–58. IEEE (2016)