SENSYBLE 2020

PRELIMINARY PAPER COLLECTION OF

---

the 3rd Workshop on
Smart Systems for
Better Living Environments

October 1st, 2020 in Karlsruhe, Germany

in conjunction with

GESELLSCHAFT FÜR INFORMATIK

INFORMATIK 2020

September 29th – October 1st, 2020, Karlsruhe, Germany

Editors:
Robert KAISER
Ralf DÖRNER

Contents

Message from the Chairs

Welcome to SENSYBLE 2020, the 3rd workshop on Smart Systems for Better Living Environments. We invite you to join us in participating in a workshop of lively discussions, exchanging ideas about a broad range of topics from smart embedded systems, systems engineering, telecommunication, mobile computing, computational theory, cryptography, to computer graphics, computer vision and artificial intelligence. For the first time, the workshop will be held in a virtual format, inviting participants from anywhere in the world. We do hope that this will enhance participation. A total of 16 papers have been selected for publication. All submissions received three or four reviews from members of the program committee, whom we would like to take this opportunity to thank for their prompt and thorough work. Also, we would like to thank the authors who submitted their work to this workshop. Last but not least, we would like to thank **you** for your interest and your participation! See you at the workshop, stay well!

The Workshop Chairs,

Robert Kaiser and Ralf Dörner
RheinMain University of Applied Sciences
Wiesbaden, Germany

Program Committee

Uwe Brinkschulte *Goethe University of Frankfurt am Main*
Martin Gergeleit *RheinMain University of Applied Sciences*
Bodo Igler *RheinMain University of Applied Sciences*
Reinhold Kroeger *RheinMain University of Applied Sciences*
Detlef Krömker *Goethe University of Frankfurt am Main*
Matthias Pacher *Goethe University of Frankfurt am Main*
Sebastian Pape *Goethe University of Frankfurt am Main*
Kai Rannenberg *Goethe University of Frankfurt am Main*
Steffen Reith *RheinMain University of Applied Sciences*
Ulrich Schwanecke *RheinMain University of Applied Sciences*
Ulrike Spierling *RheinMain University of Applied Sciences*
Marcus Thoss *RheinMain University of Applied Sciences*
Adrian Ulges *RheinMain University of Applied Sciences*

Assessment of Current Intrusion Detection System Concepts for Intra-Vehicle Communication

Oleg Schell[1], Jan Peter Reinhard[2], Marcel Kneib[3], Martin Ring[4]

Abstract: Nowadays, vehicles incorporate a lot of electronics, which offer both advanced functionalities, but also a great attack surface. Once having access to the communication network, an attacker can control critical functions like accelerating or steering. One possibility to detect these malicious intentions consists in the implementation of Intrusion Detection Systems (IDSs), which will even become mandatory via UN regulations in the future. Therefore, it is important for manufacturers and engineers to understand the opportunities and challenges of automotive IDSs. Giving an overview on these detection mechanisms is the primary goal of this elaboration. After the current vehicular communication architectures and protocols are outlined, potential attacks on the communication network are addressed. Afterwards, existing IDS concepts are presented, while the general requirements on these systems from an automotive perspective are stated and highlighted by the example of a state-of-the-art IDS approach. Following the discussion on how to react to a detection, the elaboration is concluded with an outlook on what has still to be achieved to successfully integrate present IDSs into a vehicle.

Keywords: Automotive Security; Intrusion Detection System; Intra-Vehicle Communication

1 Introduction

With increasing functionality of the vehicular ecosystem, the number of electronic components and interfaces that are indispensable for safety realizations and provided services also increases. As a consequence, these advances widen the surface for the execution of cyber-physical attacks that no longer require physical access to the vehicle due to wireless interfaces like Bluetooth, WiFi or the Global System for Mobile Communication (GSM) [Lu14]. By exploiting these interfaces and the security vulnerabilities in the software of Electronic Control Units (ECUs), an adversary can get access to the internal communication network and remotely control crucial functionalities like steering, accelerating or braking [MV15]. It is evident that these possibilities can have severe consequences for both the driver and its environment. At this point, it must be mentioned that this threat does not only affect a single but several vehicle models, including Jeep [MV15], Tesla [NLD17] and BMW [Ca19].

[1] Bosch Engineering GmbH, Robert-Bosch-Allee 1, 74232 Abstatt, Germany, oleg.schell@de.bosch.com
[2] Hochschule RheinMain, Kurt-Schumacher-Ring 18, 65197 Wiesbaden, Germany, janpeterreinhard@gmail.com
[3] Robert Bosch GmbH, Mittlerer Pfad 9, 70499 Stuttgart, Germany, marcel.kneib@de.bosch.com
[4] Bosch Engineering GmbH, Robert-Bosch-Allee 1, 74232 Abstatt, Germany, martin.ring@de.bosch.com

These circumstances made it quickly apparent that malicious activities on the intra-vehicle communication networks had to be detected and prevented. The latest efforts to realize this intent include UN regulations [UN20], which propose to implement countermeasures on a mandatory basis. One of the possibilities that they suggest is the utilization of IDSs to provide a security measure on network basis. Since the demands placed on such systems in the automotive domain are different from those in a classic IT environment, this elaboration will outline different IDS approaches and their requirements for the implementation in vehicles. After a short presentation of one of the state-of-the-art approaches for detecting attacks on the Controller Area Network (CAN), the most frequently used communication protocol in vehicles, open questions regarding the general realization and implementation are addressed. The presented concepts should serve as a guideline for engineers, while the inadequacies should give researchers a direction to advance the topic of automotive IDSs.

Protocol	Rates	Description	Use Case
Linear Interconnected Network (LIN)	11.2 or 19.6 KBit/s	Cheap and simple protocol using a linear bus architecture for small intra-vehicle services.	Battery Monitoring, Window Lifter Control, Temperature Sensors
Media Oriented Systems Transport (MOST)	25, 50 or 150 MBit/s	Relatively expensive protocol, which provides high data rates for infotainment applications.	Audio Module, Navigation System, Infotainment
Controller Area Network (CAN)	125 or 500 KBit/s	Most common protocol for vehicular networks, with new CAN FD and CAN XL standards providing higher data rates.	Engine Control, Electrical Stability Control, Transmission Unit
Ethernet	100 MBit/s	Protocol, which is relatively new in the automotive domain and becomes more popular due to high data rates and cost.	ECU Flash Interface, Cameras, Radar, Network Backbones
FlexRay	5 or 10 MBit/s	Fault tolerant protocol with high bandwidths, which is not often used because of its complexity and high cost.	Steering Angle Sensor, Throttle Control, All-Wheel Drive

Tab. 1: Wired communication protocols for intra-vehicle data exchange based on [Al19; Hu19].

2 Automotive Network Architectures and Protocols

Every vehicle is a distributed system, which is made of ECUs communicating over different protocols. The ECUs represent the computing units of a vehicle and differ in performance, memory capacity and robustness depending on the intended use. In this context, robustness means how well an ECU is protected against temperature, pressure and humidity changes, as well as, its level of failure safety. A single ECU can have multiple network interfaces and thus send data over different media. Wired networks are more common in this regard, for which different protocols exist depending on the area of application as stated in Tab. 1. Besides these, there are also wireless standards like Bluetooth Low Energy or ZigBee, which can also be deployed in the vehicle, but have not been widely used to date [Hu19].

The potential topologies, which can be used for these communication protocols, vary widely. Besides the star topology, where each device is connected to a central gateway, repeater or hub to route the data to its destination, each ECU can also be linked to its neighbor to form a ring topology. While furthermore a point-to-point connection is the easiest of all topologies, the most commonly used interconnection in automotive networks is the bus topology, where every device is connected to a single communication line and transfers data in a broadcasting manner. Apart from the aforementioned interconnection methods for individual ECUs, the entire communication architecture is undergoing a transformation. The trend is shifting from an application-specific communication architecture towards a domain-specific one, where the ECUs in each domain communicate with protocols stated in Tab. 1, while Ethernet is used across domains for fast data exchanges. In the future, it is intended to create a centralized architecture that consists of a few high performance controller, which are connected to most of the ECUs or domains with Ethernet [He19]. Although Ethernet may replace protocols like FlexRay and MOST [Rö17], there is still the need to secure the individual communication sections and the utilized protocols like CAN.

Requirement	Attack	Description
Authenticity	Spoof & Replay	Impersonating network participants without being noticed or replaying prerecorded messages on their behalf.
Confidentiality	Eavesdrop	Unauthorized access to data and information which is transmitted over the vehicular network.
Availability	Flood & Drop	Preventing operation of network participants by either withholding data or flooding the network with irrelevant messages impeding the transmission of relevant data.
Integrity	Manipulate	Manipulating content of transmitted messages in such a way that it remains hidden from the other network participants.

Tab. 2: Security requirements and potential attacks on intra-vehicle networks.

3 Attacks on Intra-Vehicle Communication

The motivation of an intrusion into the communication network of a vehicle is manifold and includes, among others, altering vehicular characteristics like engine performance or mileage, intruding into the driver's privacy or interfering into the control to cause harm. In order to achieve these goals, access to the communication networks is required first. Before vehicles were equipped with wireless interfaces, access could only be gained in a physical way, either by connecting directly to the communication wires or over the On-board diagnostics (OBD)-II port, which is used by workshops for diagnostic purposes. Nowadays, these wireless interfaces like Bluetooth or Wi-Fi of the telematic control unit represent an additional risk through which unauthorized access is possible. By exploiting security breaches and rewriting the software on this ECU [MV13], data can usually be both read and written by the adversary on the network to which this unit is connected to.

Once having access to the communication network, an adversary can perform different malicious actions due to the lack of security mechanisms. As exemplarily stated in Tab. 2, these attacks can be classified according to the security requirement they violate. For this reason, appropriate security mechanisms have to be considered already during vehicle design or integrated afterwards to prevent such actions.

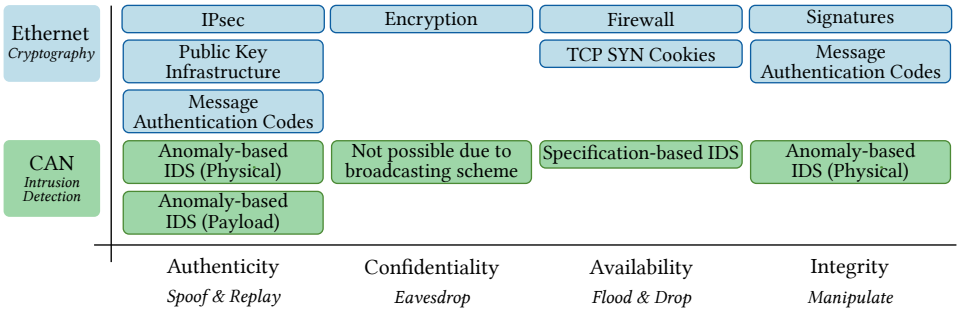


Fig. 1: Possible security mechanisms for vehicular Ethernet and CAN.

4 Intrusion Detection System Concepts

Compared to the remaining communication protocols from Tab. 1, which have been in use for several years, Ethernet is currently finding more and more its way into the vehicle, enhancing communication networks with a high bandwidth and low cost [Rö17]. Here, Ethernet does not provide security mechanisms by itself, they are mainly obtained by using higher level protocols like Transmission Control Protocol/Internet Protocol (TCP/IP). Since these have long been used in classical IT systems such as home computers, server applications or corporate networks, usual security mechanisms, some of which are listed in Fig. 1, can be applied in the automotive sector. Although the bandwidth allows the transmission of additional data for the proper operation of these security mechanisms, the real-time requirements must still be considered for their implementation.

Taking CAN into consideration, the limited hardware resources and communication bandwidths commonly available, exacerbate the implementation of cryptography-based approaches [GM13]. Furthermore, the fact that CAN is already used in almost every vehicle renders a subsequent security provision for existing networks more difficult. This is especially critical, since CAN does not deploy any security mechanisms and can therefore be attacked successfully with ease [Al19]. To remedy this issue, the utilization of IDSs can be considered, which can retrofit basic security aspects into existing and future CAN networks. Since IDSs can be realized in various ways, the most prominent concepts are discussed in more detail, mentioning their merits and shortcomings. At this point it should be noted that the mentioned IDS concepts represent general methodologies and can therefore also be utilized for different communication protocols like Ethernet.

Signature-based IDS Using signature-based detection, the ongoing communication is continuously compared to known attack patterns in the IDS database like the sequence of transmitted data or malicious instruction. Although usually used in anti-virus software for IT systems, the pattern matching procedure is a resource demanding process for vehicular ECUs. Further, the attack database has to be kept updated and distributed to the individual vehicles, while there is no possibility to recognize novel attacks. This is particularly critical as such patterns have not yet been extensively established for attacks on vehicles, which however are constantly increasing in number. At last, unlike classical IT systems, where patterns can be shared, a manufacturer-specific signature cannot be used by another vendor, as other digital platforms are usually implemented. On the other hand, once these patterns are developed, this approach reliably detects known attacks with a low false alarm rate.

Specification-based IDS Communication properties like transmission schedules, communication partner or which ECU is eligible to transmit which messages on the respective network segment, are mostly specified during the design phase of a vehicle and usually do not change after deployment. The same applies to the static communication architectures which are not altered after establishment. An IDS can take advantage of this by considering the specification and establishing rules which are checked during the ongoing communication. For example, in this way it is possible to implement a firewall in units which interconnect several networks and which then are able to block transmissions not complying to the rules. Be it the deviation of data values from a predefined range or the propagation of unauthorized messages in a network, the specification-based approach is able to detect these easily and efficiently. Big disadvantages are that these rules have to be manually created by experts, which is error-prone and thus can lead to a high number of false alarms, while an adversary can circumvent the rules if he acts within acceptable limits.

Anomaly-based IDS Anomaly-based approaches work similarly to specification-based procedures in that they detect deviations from predefined behavior. The difference here is that the predefined behavior is learned by the system itself in the case of anomaly detection. This not only eliminates the need to set up rules but also enables the detection of unknown and novel attacks. Generally, the normal behavior can be learned based on different communication characteristics, which are briefly described in the following.

1. **Payload**: Mainly utilizing machine learning algorithms, IDSs of this category strive to establish a model of the message content and attempt to detect unusual data sequences that can be traced back to attacks. This approach would represent a promising option to detect different types of intrusions, if only the interrelationships were not so complex, the need for data not so high and the computational power not so demanding.
2. **Physical**: ECUs and their electronic components are subject to manufacturing imperfections, leading to small differences in the physical properties like clock timings and voltages. IDSs can utilize these small differences and implement sender identification mechanisms, with which unauthorized transmissions can be detected and the malicious ECU pinpointed. However, since the properties refer to the respective

ECUs, an adversary can send unnoticed authorized messages with malicious content from the compromised ECU. Further, in most cases, high performance analog-to-digital converters (ADCs) or timers are required to record even small differences.

Regardless of which characteristic is selected, anomaly-based approaches necessarily require trustworthy data to learn the normal behavior, whereby the time and data amounts required for this learning should not be neglected. Furthermore, one of the main reasons why these have not yet been widely used is that they have a high false alarm rate, which is especially important when the driver is not to be distracted unnecessarily and when intrusions are not only to be detected but also actively prevented.

As shown in Fig. 1, different IDS concepts provide varying security measures for protocols like CAN. To establish a holistic security system, it is therefore essential to implement a combination of these concepts. For instance, specification-based methods could provide the first line of defense against rudimentary attacks, while anomaly-based procedures detect the presence of more sophisticated attackers. These *hybrid* IDSs allow the incorporation of both digital and physical characteristics making the resulting system more robust and reliable.

5 Requirements

For the design of an intrusion detection approach, different requirements play a major role in the automotive domain. The most significant difference compared to classical IT systems is the high importance of *Safety*. Therefore, an IDS is not allowed to affect data by delaying or removing it which may lead to the loss of safety relevant information. This also includes the fact that a restriction of the information availability by an IDS must not take place. Although an IDS is a security measure by itself, it also has to meet different *Security* requirements. In this context, it is important that the system is not reducing the functionality and effectiveness of other security concepts such as firewalls or encryption, while not creating new exploits and critical security gaps. Other important requirements relate to the *Privacy* of the driver and other passengers. Because of the high connectivity of modern automotive systems, it is important that an IDS does not leak private data without permission to other systems. Especially, if the IDS uses a cloud-based back-end for incident analysis and transmits sensitive data. Finally, the *Update* of an IDS plays a crucial part for the requirements. In contrast to IDSs for classical IT systems, which can be updated almost at any time via the Internet, with vehicular IDSs it must be ensured that, for example, the rule update procedure of a specification-based approach does not open new security breaches and is not corrupted. If such an secure over-the-air update is not possible, the implemented IDS concept can be bypassed and therefore be rendered useless.

These requirements must be fulfilled for the design of both a single and a hybrid IDS. Since the need for a joint intrusion detection method has already been made clear, the most recently presented Edge-based Sender Identification (EASI) [KSH20] approach can be employed,

for example, as a component for anomaly detection which uses physical characteristics. By sampling voltages during transmissions of each ECU and establishing a model based on derived features, this approach can detect unauthorized transmissions by identifying the sender of a message. Because EASI only analyzes voltages on the CAN wires and thus represents a passive network-based approach, no messages are manipulated or delayed allowing an unrestricted exchange of safety critical data. The fact that the voltage is only sampled also plays an important role for the security requirements, as the functionality of other security measures is not restricted. Furthermore, private data is neither processed nor sent during the IDS operation. Finally, if the model needs to be updated, a few authenticated messages can be sent using Message Authentication Codes (MACs) for a trustworthy voltage extraction, resulting in the independence of over-the-air updates. For this reason, EASI represents a possibility to detect intrusions in CAN, which violate the authenticity aspect.

6 Post Detection and Outlook

An important question that has not yet been clarified in this elaboration is, how to deal with detections in the automotive environment. In general there are three possibilities; logging, notifying and preventing, whereby each of the methods can be more or less beneficial in individual aspects. Logging, for example, stores information about the potential intrusion, which can then be read out in case of an incident to patch the security gap in the remaining fleet. Although logging does not distract or hinder the driver, the large amount of information must first be stored and subsequently analyzed in time-consuming manual work. Considering to notify the driver in case of an intrusion, the danger is immediately apparent and actions can be carried out. False alarms play a crucial part here, because even with transmission rates of several tens of milliseconds and a low false alarm rate, the driver is mistakenly warned several times a minute. These false alarms become even more serious if active prevention is taken into account. If safety-relevant data exchange is incorrectly recognized as an attack and on this basis prevented, it becomes apparent that such actions can have far-reaching consequences for both the passengers and the environment.

Only after it is clarified how to deal with these detections, IDSs can be effectively put into utilization, whereby further challenges have to be taken into account. Up to now, the throughout implementation of security in a vehicle is regarded as a matter of course rather than a mandatory component. For this reason the available resources for IDSs are kept as low as possible, which stands in contradiction to increasing data amounts and complexity. Achieving lower latencies and real-time capability, which are particularly relevant for safety-critical tasks, ECUs require more hardware resources and computing power. These requirements are especially true for anomaly-based IDSs, which currently receive the greatest focus, as they are most promising to detect sophisticated attacks [LOA19]. For the evaluation of anomaly-based approaches, individually recorded data from test vehicles are often used. Yet, to ensure a better comparability of existing approaches, a publicly available data set with respective communication characteristics is required. In the end, knowledge of

different disciplines like artificial intelligence, automotive systems and electrical engineering are crucial for the design and consolidation of different security concepts. Only if this knowledge comes together a holistic security system can be developed, which is able to recognize or prevent not only rudimentary but also the presence of advanced attackers.

References

[Al19] Al-Jarrah, O. Y.; Maple, C.; Dianati, M.; Oxtoby, D.; Mouzakitis, A.: Intrusion Detection Systems for Intra-Vehicle Networks: A Review. IEEE Access 7/, pp. 21266–21289, 2019, ISSN: 2169-3536.

[Ca19] Cai, Z.; Wang, A.; Zhang, W.; Gruffke, M.; Schweppe, H.: 0-days & Mitigations: Roadways to Exploit and Secure Connected BMW Cars. Black Hat USA/, 2019.

[GM13] Groza, B.; Murvay, S.: Efficient Protocols for Secure Broadcast in Controller Area Networks. IEEE Transactions on Industrial Informatics 9/4, 2013.

[He19] Helge Zinner Julian Brand, D. H.: Automotive E/E Architecture evolution and the impact on the network. IEEE802 Plenary, March 2019, 802.1 TSN/, 2019.

[Hu19] Huang, J.; Zhao, M.; Zhou, Y.; Xing, C.: In-Vehicle Networking: Protocols, Challenges, and Solutions. IEEE Network 33/1, pp. 92–98, Jan. 2019.

[KSH20] Kneib, M.; Schell, O.; Huth, C.: EASI: Edge-based sender identification on resource-constrained platforms for automotive networks. In: Proc. Netw. Distrib. Syst. Secur. Symp. Pp. 1–16, 2020.

[LOA19] Lokman, S.-F.; Othman, A. T.; Abu-Bakar, M.-H.: Intrusion detection system for automotive Controller Area Network (CAN) bus system: a review. EURASIP Journal on Wireless Communications and Networking 2019/1, p. 184, 2019.

[Lu14] Lu, N.; Cheng, N.; Zhang, N.; Shen, X.; Mark, J. W.: Connected Vehicles: Solutions and Challenges. IEEE Internet of Things Journal 1/4, 2014.

[MV13] Miller, C.; Valasek, C.: Adventures in automotive networks and control units. Def Con 21/, pp. 260–264, 2013.

[MV15] Miller, C.; Valasek, C.: Remote exploitation of an unaltered passenger vehicle. Black Hat USA 2015/, p. 91, 2015.

[NLD17] Nie, S.; Liu, L.; Du, Y.: Free-fall: Hacking tesla from wireless to can bus. Briefing, Black Hat USA/, pp. 1–16, 2017.

[Rö17] Röder, J.: Automotive Ethernet - Die Zukunft der vernetzten Fahrzeugarchitektur - The future of in-vehicle data Management./, July 2017, URL: https://www.vdi-wissensforum.de/news/automotive-ethernet/.

[UN20] UN Task Force on Cyber Security and Over-The-Air issues: Proposal for the 01 series of amendments to the new UN Regulation on uniform provisions concerning the approval of vehicles with regard to cyber security and of cybersecurity management systems. 2020.

(Hrsg.): Sensyble,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2020 1

Development of a Vehicle Simulator for the Evaluation of a Novel Organic Control Unit Concept

Melanie Brinkschulte[1]

Abstract: New challenges in the field of automotive systems (e.g. autonomous driving) require innovative and highly robust vehicle architectures. These are intended to increase the reliability and fault tolerance of the system and therefore realize the transition from Fail-Save to Fail-Operational behavior. Organic computing is a possible approach to achieve these goals. Based on an artificial hormone system and an artificial DNA, a novel organic control unit concept is developed. In this paper we introduce an evaluation tool for this novel concept. Therefore, a simulator physically models the longitudinal and lateral dynamics of a vehicle. For an easy handling, visualization and reproducibility of experiments, an user interface and a scripting language is designed. In extensive evaluation runs the usability of the vehicle simulator is tested. Hereby, real vehicle data is used.

Keywords: Vehicle Simulator; Organic Computing; Fail-Operational

1 Introduction

In this paper we introduce an evaluation tool for a novel organic control unit concept based on an artificial hormone system (AHS) [vBP11] and an artifical DNA (ADNA) [Br15] . Therefore, a simulator physically models the longitudinal and lateral dynamics of a vehicle. The parameters of the vehicle (weight and measures, engine and gear parameters, brake parameters, air and roll resistance, . . .) can be individually chosen. For an easy handling, visualization and reproducibility of experiments, an extensive user interface and a scripting language is designed. Also, this simulator allows the evaluation of various automotive control components (ECUs) like ABS, ASR, power steering and cruise control. All input data like brake, throttle and steering positions can be given by the user interface or the scripting language. In addition, both input options can be used simultaneously. The output values like brake force, wheel speed, vehicle speed, steering angles, etc. are visualized in the user interface, timestamped and written to a log file for detailed examination. Thereby, the user can choose which physical value is logged as well as the time resolution of the logging process. By fault injection, ECU failures at run-time can be induced at arbitrary times during a simulation run.
This paper is structured as follows: Section 2 gives a short overview of the designed

[1] University of Mannheim, Chair of Information Systems II, Schloss, 68131 Mannheim, Germany & Goethe University Frankfurt am Main, Computer Science Department, Robert-Maier-Str. 11-15, 60325 Frankfurt am Main, Germany brinkschulte@uni-mannheim.de

physical models. Following, Section 3 describes the architecture and the interaction of the components of the simulator. Section 4 presents an extract of the extensive evaluation and Section 5 discusses related work. Finally, Section 6 concludes this paper.

2 Models

In this work multiple physical models (vehicle dynamics, steering, brake and engine) are designed and used. For reason of space, we can only shortly enumerate these models here, For more detail, please refer to [Br19].
The **vehicle dynamics model** is based on the linear single-track model. However, this is not sufficient for the desired purpose and is therefore extended (by adding of longitudinal dynamics, frictional conditions and accuracy by removing the small angle approximation, extension to an rudimentary two-track model) to an efficient nonlinear two-track model without small angle approximation. The **steering model** is a speed-dependent steering system with optional steering assistance. The **brake model** includes characteristic curve mappings and brake cylinder delay. The **engine model** includes an optional adjustable four-wheel drive, as well as drive delay and dead times. Furthermore, a simple **gearbox model** was realized.

3 Simulator

In this Section, the architecture as well as the communication and interaction of the components (user interface, physical models, simulator-sensor/actuator interface) of the vehicle simulator is shown. The vehicle simulator is implemented in C++ while Qt 5.11.1 is used to implement the graphical user interfaces.

3.1 Architecture

The simulator consists of three parts: the physical models of the vehicle, the user-interface and the simulator sensor/actuator interface (Figure 1). The physical models have internal state data (e.g. speeds, distances travelled, angles, etc.) and receive vehicle data (e.g. the vehicle mass, vehicle dimensions, etc.), environmental data (e.g. the static/sliding friction value between road and tires) and input data (e.g. a steering angle, an accelerator pedal position, etc.). They then use these to calculate the output data (e.g. forces and accelerations).

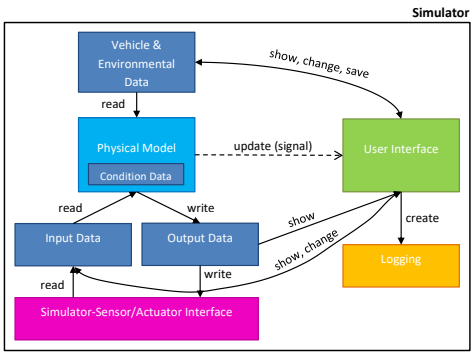


Fig. 1: Architecture of the simulator

Through the input and output data, the physical models are connected via the simulator sensor/actuator interface to the AHS and the ADNA that realizes the vehicle's control units. The user interface can display, change and save vehicle condition and environment data. Furthermore, the settings for the creation of a parameterizable log files can be defined in the interface. The visualization consists of a top view with optional fade-in of different force, track and speed vectors as well as a side view, which shows the wheel speeds and the adhesion conditions.
Furthermore the simulator is real-time capable. The two-layer real-time architecture is shown in Figure 2. The outer layer uses a $10ms$ period to ensure smooth and jitter-free

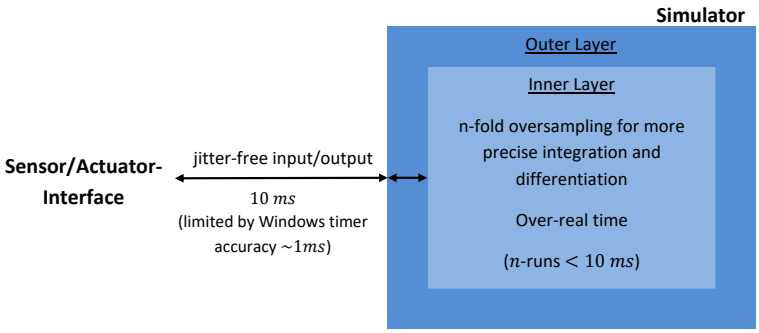


Fig. 2: Real-time architecture of the vehicle simulator

real-time input/output. This cycle time was chosen because the Windows timers used in the implementation have an accuracy of about $1ms$. In the inner layer, a n-fold oversampling (n-fold execution of a simulation run) is performed to achieve a more precise integration and differentiation. The number of passes of the inner layer can be individually adjusted by the user. The only condition is that the time required for this number of runs is less than $10ms$ (run time of outer layer). This means that over-real time is present there.

3.2 Interaction

In Figure 3 the interaction and data transfer between the user interface, the physical models and the simulator-sensor/actuator interface is shown. The physical models are divided into the submodels vehicle dynamics, brake, engine and steering. The sensors of the Simulator Sensor/Actuator Interface receive their inputs from the physical submodels and from the user (e.g. brake pedal sensor, accelerator pedal sensor, etc.). The user has two possibilities to create his inputs. On the one hand, he can make entries via the user interface to directly control the vehicle and influence its environment. On the other hand, in order to enable precisely repeatable experiments, it is possible to specify input in the form of a script file. The input data is then converted by the control units into corresponding actuator values (e.g. brake cylinder control, drive control, steering angle control, etc.) according to the artificial

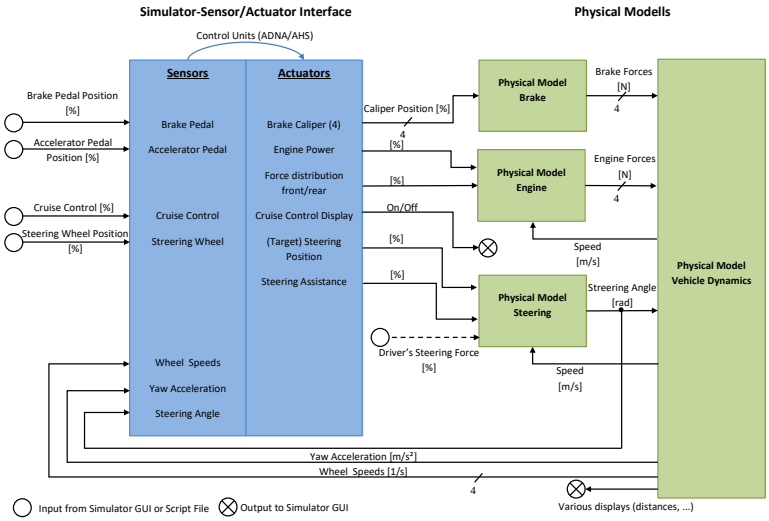
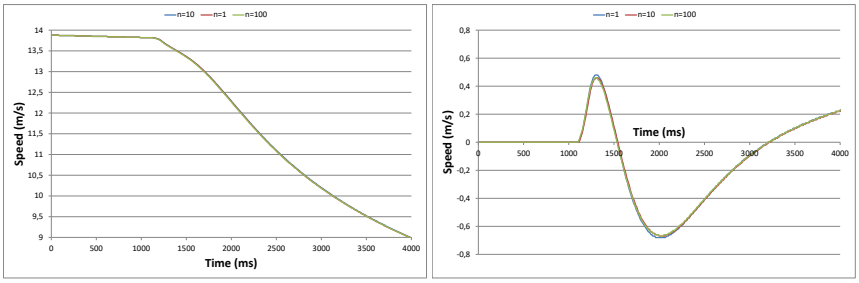


Fig. 3: Interaction and data transfer between the user interface, the physical models and the simulator-sensor/actuator interface

DNA running on them. These values then enter the physical models of the steering, the brake and the engine. The outputs of these models are then passed on to vehicle dynamics model, which in turn generates new sensor signals in a closed control loop.

4 Evaluation

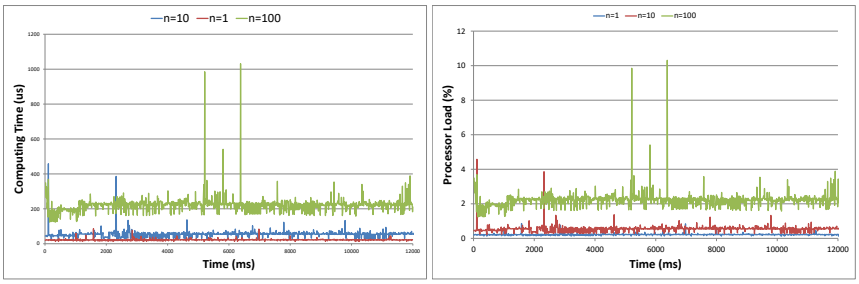
Extensive evaluations of the presented work are made. Unfortunately in the scope of this paper only the most important evaluation results can be presented in detail.
At first, the trade-off between simulator processor load and accuracy is evaluated. Therefore, a simple reproducible experiment (vehicle initiates a curve at a given starting speed of $50\frac{km}{h}$, the static friction is set to a value so that the vehicle doesn't slide) with different numbers of steps ($n = [1, 10, 100]$) per simulation period in the inner layer of the simulator is used. To determine the gain in accuracy, the speed in x- and y-direction are compared exemplary. In the diagrams, minimal deviations of the curves from each other can be seen.



(a) Comparison of the speed in x-direction (b) Comparison of the speed in y-direction

Fig. 4: Comparison of speed in x- and y-direction at $n = 1$, $n = 10$ and $n = 100$ steps per simulation period

The speed in x-direction (Figure 4a) shows the smallest deviations. These never exceed 0.1%. For the speed in y-direction (Figure 4b), slightly larger deviations of a maximum of 4% are visible. Nevertheless, the curves are almost identical at $n = 10$ and $n = 100$ steps per simulation period. Only the curves resulting from only one step per simulation period differ slightly more from the others at two points ($1280ms - 1340ms$ and $1810ms - 2140ms$). The resulting computing time for a simulation period and the resulting processor load are shown in Figure 5. As expected, the required computation time per simulation period and thus the processor load increases with increasing number of simulation steps n. This results in a maximum calculation time of $1.031ms$ with $n = 100$ which corresponds to a maximum processor load of 10%. It turns out that the model works with high accuracy at



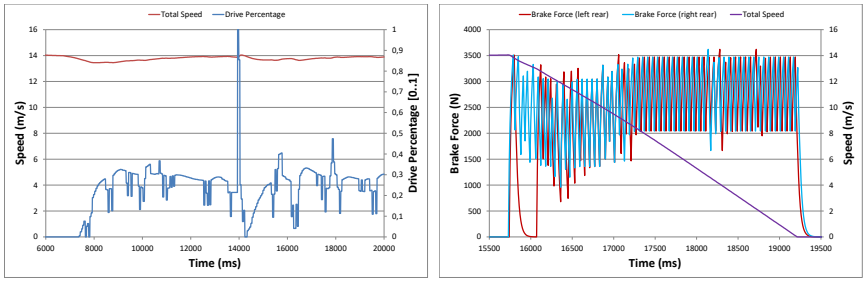
(a) Computing time (b) Processor load

Fig. 5: Required computing time and resulting processor load at $n = 1$, $n = 10$ and $n = 100$ steps per simulation period

low computational effort. Especially $n = 10$ turns out to be a good choice with low overhead and high accuracy.
Next, the longitudinal and lateral dynamics have been evaluated with different experiments .

Longitudinal: reality comparison of acceleration time from 0 to 100 $\frac{km}{h}$, maximum final speed and braking distance from 100 $\frac{km}{h}$ to standstill. Lateral: pure steering and steering in combination with braking experiments

The vehicle is also operated in the non-linear range (skidding). It turns out the simulator behaves as expected and the results are very close to those of a real car. In this paper, we focus on the evaluation of the closed control loop between simulator and control units by means of a cruise control and ABS. The aim of the simulator is to create an evaluation tool for the AHS together with the ADNA as an organic control unit concept. This ECU concept should keep the system operational even in case of component failure and thus show the desired *Fail-Operational* behaviour.
Therefore, in further evaluation the behaviour of the closed control loop in case of failure of individual DNA processors is examined. Failures of processors (ECUs) were added during ABS braking in a corner or speed regulation by cruise control. In Figure 6a the failure of the processor is clearly shown by a peak in the drive power percentage. The processor failure causes the wheel speed sensors of the front wheels to be temporarily lost. The cruise control implemented uses this data to determine the speed of the vehicle. If it now receives no more rotation speed, it assumes that the vehicle has a speed of $0\frac{m}{s}$ and thus accelerates at maximum to reach the set target speed. As soon as the AHS/ADNA has distributed the wheel speed detection to the remaining processors in one of the next hormone cycles by self-healing, the failure is eliminated and the cruise control works correctly again. Here the failure lasts about $100ms$. In Figure 6b the failure of a processor is also clearly visible.



(a) Simulation results of the speed experiment in case of processor (ECU) failure during cruise control

(b) Simulation results of the ABS corner brake experiment in case of processor (ECU) failure

Fig. 6: Simulation results of processor (or control unit) failure experiments

The braking force at the rear left wheel drops to zero shortly after the start of braking at an approximate simulation time of about $15860ms$. After a simulation time of about $16060ms$, the braking force is again controlled by the ABS. The failure was therefore corrected after about $200ms$. In Figure 7 the comparison between the speeds and the distances covered with and without failure is shown. The speed curve during failure shows a slight bend, which is caused by the short-term reduced braking effect during failure. As a result, the vehicle comes to a halt about $250ms$ later than without failure. At the time the vehicle comes to a halt without failure, the vehicle still has a speed of $1.2\frac{m}{s}$ during the failure. This results in an extension of the braking distance of approximately $0.15m$. These evaluations show the suitability of the developed vehicle simulator for the intended application. The simulator

is able to simulate failures of processors or control units, which allows to investigate and analyze such failure scenarios.

5 Related Work

Vehicle modeling has always been of big importance for the automotive industry [WSK11]. A general overview of vehicle models can be found in [SHB13] and [MW14]. Here a suitable compromise between model complexity and the number of model parameters or computing time should be found for the respective application, as is also described in [Un13]. This enables an optimal use of the respective model within the scope of the intended application. Highly complex multi-mass models are used, for example, to evaluate the driver's driving experience [Un13] or in comfort simulation [Am13]. If, on the other hand, the lane control of vehicles [Ar15], [He09] or the evaluation of vehicle measurement data during road tests [Se05] are concerned, simple models such as the linear single lane model are usually sufficient.

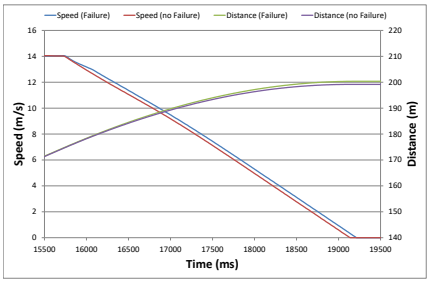


Fig. 7: Comparison between speeds and distances in the ABS-experiments with and without failure

In the presented work, the focus is on vehicle dynamics in connection with a novel, robust control unit concept. In case of an ECU failure, the evaluation of the lateral and longitudinal dynamics of the vehicle is of great importance, whereby the ECUs receive information from all four wheels. In the event of skidding (oversteer and understeer), the vehicle also leaves the linear range. The linear single-track model was therefore too simplified and not suitable for the desired purpose. However, driving comfort was also not in the focus of the work, i.e. there was no need to use a complex multi-mass model.

For this reason, an adapted model was developed, which extends the linear single-track model to a nonlinear model with two-track components. With this model, meaningful simulations for the evaluation of the novel ECU concept can be performed on the basis of an ADNA (especially regarding failure and robustness of ECUs), while the model complexity remains at a reasonably low level.

6 Conclusion

In this paper a vehicle simulator is presented as an evaluation tool of an organic control unit concept (represented by the AHS in combination with the ADNA) in the automotive field. For this purpose, physical models for steering, brake, engine and vehicle dynamics are developed, validated and implemented. With these models a comprehensive evaluation of driving situations for the evaluation of the organic ECU concept or robust ECU is possible.

To enable reproducible experiments, a script language was developed and implemented. This allows flexible experiments under identical conditions and thus enables the comparison of different ECU concepts against each other. In future work, the failure behavior of more sophisticated control units for autonmous driving will be evaluated with this tool.

References

[Am13] Amelunxen, Hendrik: Fahrdynamikmodelle für Echtzeitsimulationen im komfortrelevanten Frequenzbereich. Dissertation, Universität Paderborn, 2013.

[Ar15] Arndt, Albrecht: Querregelung eines spurgeführten Modellfahrzeugs. chapter 5 Modellbildung, 2015.

[Br15] Brinkschulte, Uwe: An artificial DNA for self-descripting and self-building embedded real-time systems. In: Concurrency and Computation: Practice and Experience, volume 28. Wiley Online Library, 2015.

[Br19] Brinkschulte, Melanie: , 2019. Masterthesis at Goethe Universität Frankfurt am Main.

[He09] Hensel, Enrico: Führungskonzept eines autonomen Fahrzeugs, Vorbetrachtung und Bewegungsmodell. Seminarbericht, Hochschule für Angewandte Wissenschaften Hamburg, 2009.

[MW14] Mitschke, Manfred; Wallentowitz, Henning: Dynamik der Kraftfahrzeuge. Springer, 2014.

[Se05] Sentürk, Fikret: Durchführen von Fahrversuchen hinsichtlich einer Optimierung von FHTW-Fahrdynamikfahrzeug. Diplomarbeit, Fachhochschule für Technik und Wirtschaft Berlin, 2005.

[SHB13] Schramm, Dieter; Hiller, Manfred; Bardini, Roberto: Modellbildung und Simulation der Dynamik von Kraftfahrzeugen. Springer, 2013.

[Un13] Unterreiner, Michael: Modellbildung und Simulation von Fahrzeugmodellen unterschiedlicher Komplexität. Dissertation, Universität Duisburg-Essen, 2013.

[vBP11] von Renteln, Alexander; Brinkschulte, Uwe; Pacher, Mathias: The Artificial Hormone System - An Organic Middleware for Self-organising Real-Time Task Allokation. In (Müller-Schloer, Christian; Schmeck, Hartmut; Ungerer, Theo, eds): Organic Computing - A Paradigm Shift for Complex Systems, chapter 4.4. Springer, 2011.

[WSK11] Wiedemann, Jochen; Schröck, David; Krantz, Werner: Fahrzeugdynamik, Themenheft Forschung, volume 7. Universität Stuttgart, 2010-2011.

Effects of the Sampling Technique on Sender Identification Systems for the Controller Area Network

Marcel Kneib[1] Oleg Schell[2]

Abstract: As a result of the ongoing development of vehicle electronics and additional wireless communication interfaces, the possibilities for attacks and their negative consequences are increasing. Once an attacker has obtained access to the internal vehicle communication, in the case of the Controller Area Network (CAN) the attacker is able to forge all messages of the connected Electronic Control Units (ECUs) without a receiving ECU being able to recognize any suspicious behavior. The use of cryptographic methods is only possible to a limited extent due to restricted resources of the ECUs, which is why sender identification systems have been presented which are able to detect these kind of attacks. Presented approaches use different procedures to capture the analog signals on which the detection of attacks respectively the identification of the sender is based. This work shows that the impact on the performance of the sender identification system by the different sampling methods is minimal and therefore the selection of the appropriate technique can be mainly based on the available resources and the communication structure of the corresponding vehicle platform. This is shown on the one hand by the direct analysis of the analog signals captured from a real vehicle as well as by an evaluation of the previously introduced sampling methods using a recently published sender identification system. In addition, an assessment of the procedures based on different parameters shows which method is to be preferred for which application.

Keywords: Automotive Security; Sender Identification; Intrusion Detection

1 Introduction

The connectivity of modern vehicles, as well as the associated amount of interfaces, is constantly increasing. This trend does not only allow additional comfort functionalities and complex driver assistance systems, but also offers additional possibilities to attack a vehicle and its functions [HKD11; LL18]. Evidence that this is not only a theoretical threat was demonstrated by the attack of Miller and Valasek [MV15], as well as the latest research of the Tencent Keen Security Lab [Ca19]. Due to the absence of authenticity in the Controller Area Network (CAN) [Ro91], which is still the most commonly used bus technology in the automotive domain, an Electronic Control Unit (ECU) cannot check whether a received message was sent by a legitimate sender. This enables the forgery of messages, i.e. the execution of impersonation attacks. This problem still exists for its successors, CAN with flexible data rate (CAN-FD) [Ro12] and CAN-XL [CA20]. Unfortunately, the use of

[1] Robert Bosch GmbH, Mittlerer Pfad 9, 70499 Stuttgart, Germany, marcel.kneib@de.bosch.com
[2] Bosch Engineering GmbH, Robert-Bosch-Allee 1, 74232 Abstatt, Germany, oleg.schell@de.bosch.com

cryptographic methods is limited due to the constrained resources of the platforms used in vehicles and the low payload and bandwidth of CAN. As an alternative or in combination with attack detection, methods have been presented in the past which provide sender identification on the basis of analog signals [Kn20]. Due to the static configuration of the internal vehicle communication, such systems allow to verify whether a message was sent by a valid ECU. For identification, however, the signals of CAN messages must first be recorded, for which the considered sender identification approaches suggest different procedures. While some methods capture the entire signal in order to extract the signal characteristics [Ch18; KH18], others concentrate on specific parts [Fo19] or individual points of a frame to determine the sender [KSH20]. The signal recording procedure has a corresponding effect on various properties, such as hardware requirements, cost, complexity and signal quality. In addition, the requirements and architecture of the actual system also have a major influence on the type of recording. This paper presents the different recording approaches and analyzes the associated effects on the relevant properties of sender identification systems for CAN. In addition, the associated performance is analyzed using the example of the recently presented work Edge-based Sender Identification (EASI) [KSH20] utilizing data from a series production vehicle. Furthermore, this work presents the individual application possibilities of the different sampling techniques, so that the reader is able to assess the optimal methodology with corresponding effects and constraints according to the respective requirements.

2 Sampling Approaches

For the CAN communication standard components are used, which can be produced in large quantities and very cost-effectively. These components only provide the connected microcontroller with access to the digital content of the message and not to the analog signals. For this reason, the actual recording of the signals must be independent of the existing hardware and therefore has to be considered and implemented by the respective sender identification approaches.

Since in principle every ECU can send a message at any time, it must be ensured that parallel transmission and thus corruption of the currently sent message does not occur. For this purpose, an ECU first checks whether the bus is free and then begins to send its message. In simplified form, the message consists of an unique message identifier, which also defines the priority of the message, and the associated content. During the transmission of the identifier it can happen that other ECUs start the transmission. The sending ECUs check whether their currently transmitted signal corresponds to the signal currently on the bus, and if not, the transmission of the respective ECU is stopped. The characteristics of the analog signals transmitted in this situation cannot be used for recognition as they contain characteristics of several ECUs. Therefore, all approaches focus on the segment succeeding the identifier.

As introduced in [KSH19], each frame consists of several symbols which represent the transmitted bits on the bus. As there are different kinds of symbols, the most approaches [Ch18;

Fo19; KH18] first group those symbols according to the voltage transitions. There are four different transition groups g, the rising ($g = 1$) and falling ($g = 2$) edges, and the stable high ($g = 3$) and low ($g = 0$) levels. The k-th symbol of a frame m sent by ECU e is defined by

$$S_k^{g,(e,m)} = (x_1, ..., x_l) \tag{1}$$

where, x_i, $i \in \{1, .., l\}$ are the individual voltage values of the symbol. Each group, defined by

$$G^{g,(e,m)} = \bigcup_{k=1}^{K} S_k^{g,(e,m)}, \tag{2}$$

can contain a different amount K of symbols. Since the subsequent calculation of the characteristics from these symbols is computationally expensive, some approaches [Ch18; KH18] initially calculate an *average symbol* per group according to Equation (3), respectively use the averaged symbol directly as characteristic [Fo19].

$$\overline{S}^{g,(e,m)} = (\frac{1}{K} \sum_{k=1}^{K} S_k^{g,(e,m)}[i] \mid \forall i \in \{1, ..., l\}) \tag{3}$$

Another possibility is to utilize only a *single symbol* for the calculation of the characteristics. Without loss of generality, the first symbol of a group is considered for the further calculations, defined by

$$\mathring{S}^{g,(e,m)} = S_1^{g,(e,m)}. \tag{4}$$

The third variant, the *composite symbol* [KSH20], assembles the symbol from individual sample points of several symbols in a group. Based on the available number of samples per symbol L and the number of samples to be used per symbol P, the number of required symbols $K = \frac{L}{P}, P \leq L$ is given. For $L = 20, P = 2, K = 10$ according to

$$\tilde{S}^{g,(e,m)} = \bigcup_{p=1}^{P} \bigcup_{k=1}^{K} S_k^{g,(e,m)}[K * (p-1) + k], \tag{5}$$

the resulting sample points are illustrated in Fig. 1 for $g = 1$.

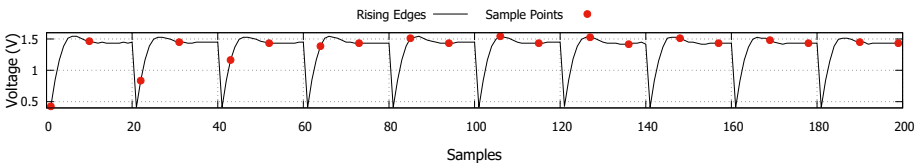


Fig. 1: Considered sampling points of the rising edges for the composite symbol.

3 Signal Analysis

3.1 Data set

For the initial analysis of the effect on the signal quality, a data set is reused which has already been utilized for the evaluation of sender identification approaches [KH18; KSH20]. The signals were recorded from a Fiat 500 which has six internal ECUs, each using up to seven different identifiers. In order to increase the number of ECUs, two additional Raspberry Pis were connected, each equipped with a CAN shield. One Raspberry Pi was connected to the bus in the trunk, while the other Pi was attached to the on-board diagnostics port together with a PicoScope 5204 at a sampling rate of 500 MS/s and a resolution of 8 bit. Since this data set is only slightly affected by changing environmental conditions, it allows the effects of the different sampling approaches to be analyzed as accurately as possible.

3.2 Metrics

In order to allow an approach-independent evaluation of the signal quality, the metrics *intra- and inter-distance* as well as their combination, the *inter-intra-distance* are used. The distance between two symbols $\mathcal{S}$ and $\mathcal{S}'$ regarding the considered sampling approaches, is defined by

$$Symbol - Distance(\mathcal{S}, \mathcal{S}', g, (e, m), (e', m')) = \frac{1}{L} \sum_{l=1}^{L} \left\| 1 - \frac{\mathcal{S}^{g,(e,m)}[l]}{\mathcal{S}'^{g,(e',m')}[l]} \right\|. \tag{6}$$

The intra-distance, calculated by Equation (6) with $\mathcal{S} = \mathcal{S}', e = e'$ and $m \neq m'$, is used to evaluate the deviations of the symbol between all frames of a single ECU. In order to additionally analyze the symbol differences between the ECUs, the inter-distance is utilized, which is calculated by Equation (6) with $\mathcal{S} = \mathcal{S}'$ and $e \neq e'$ for all ECUs and their associated frames. Finally, the inter-intra-distance, i.e. the difference between the inter- and intra-distance, is used as the metric to assess the distance between the ECUs, taking into account the magnitude of the deviations of the ECUs with respect to the considered sampling approach.

Furthermore, the statement must be evaluated that the differences of the variations of the sampling approaches are negligible, since they are in the range of the natural variation of the symbols within a frame [KSH20]. Therefore, first the natural deviation of the symbols in the data set within a frame is calculated with $\mathcal{S} = S_k^{g,(e,m)}$ respectively $\mathcal{S}' = S_{k'}^{g,(e,m)}$ for $k \neq k'$. Following, for the comparison of the deviation and thus the verification of the statement, $\mathcal{S}'$ is replaced by the symbols created by the sampling approaches.

Tab. 1: Effect of the sampling technique on the signal quality.

	Intra-Distance	Inter-Distance	Inter-Intra-Distance
Average Symbol	0.3763 %	6.9193 %	6.5430 %
Single Symbol	0.6886 %	6.9351 %	6.2466 %
Composite Symbol	0.6886 %	6.9351 %	6.2466 %

3.3 Analysis

In Tab. 1 the calculated distances for the different sampling techniques are shown. The symbols $g = 1$, i.e. the rising edges, were used for the calculation, since these symbols contain the most important characteristics for distinguishability [KH18; KSH20]. For the average symbol it can be seen that it has the highest inter-intra-distance, mainly due to the lower intra-distance. This indicates that the use of the average symbol allows the best overall differentiation among all ECUs. However, it can also be seen that the single and composite symbols have no noticeable differences and, with an inter-intra-distance that is less than 0.3 % lower, the distinguishability is only minimally reduced.

Tab. 2: Effect of the sampling technique on the intra-frame deviation.

Data set	Average Symbol	Single Symbol	Composite Symbol
0.6425 %	0.4983 %	0.6230 %	0.6101 %

The deviations of the symbols within a single frame for the data set and the considered sampling techniques are shown in Tab. 2. Basically, it can be noticed that the data set shows the biggest and the average symbols the lowest deviations and the single and composite symbol again are close to each other and also close to the data set. All in all, the results confirm the claim that the differences due to the sampling approaches are negligible.

4 Sender Identification System Evaluation

The previous analysis is based on the signal itself respectively on the calculated distances. However, since the sender identification approaches use much more complex characteristics for classification, this chapter analyzes the effect of the different sampling methods on a real system. For this purpose, the Edge-based Sender Identification (EASI) [KSH20], which also uses only a single rising edge for identification, is considered. For the evaluation, the system uses the same configuration and data set used in the original work for the analysis of the behavior of the characteristics to environmental factors as well as the effect of electrical consumers. The utilized vehicle is the same as mentioned in Sect. 3.1, but without having the additional Raspberry Pis connected. The metrics considered for the evaluation of the approximately 55 000 frames are the *true positive* and *true negative rate*, the *identification rate* and the *confidence* of the system. A high true positive rate indicates the system's ability to detect forged frames, the true negative rate allows an assessment of the amount of wrong

alarms, the identification rate analyzes the general performance of sender identification and the confidence gives an indication on how well the learned model fits to the current situation.

Tab. 3: Effect of the sampling technique on the sender identification performance.

	Average Symbol	**Single Symbol**	**Composite Symbol**
True Positive Rate	99.82 %	99.16 %	99.59 %
True Negative Rate	100 %	100 %	100 %
Identification Rate	99.98 %	99.91 %	99.98 %
Confidence	99.81 %	99.57 %	99.87 %

In Tab. 3, it can be noticed that the use of the average symbol achieves the best results considering the real sender identification system. While no false alarms have occurred in any of the analyses, the usage of the single symbol leads to a slight decrease of the true positive rate. Assuming that an attack requires three messages which are not detected by the system, the probability of a successful attack is increased from 5.8^{-9} to 5.9^{-7} by using the single symbol instead of the average symbol. Accordingly, even by using EASI with the most lightweight configuration, the single symbol, a high probability of detecting potential attacks is achieved and thus still provides a high increase in security. Overall, as already determined during the direct signal analysis in Sect. 3.3, no significant differences can be observed for the different sampling techniques.

5 Assessment

Tab. 4: Assessment of the signal acquisition approaches.

	Performance	**Additional Hardware Requirements**	**Resource Requirements**	**Multi-Channel Capability**	**Complexity**	**Timing Restrictions**
Average Symbol	+	-	∘	-	∘	∘
Single Symbol	-	-	+	+	+	+
Composite Symbol	∘	∘	+	-	-	-

An overview of the assessment is presented in Tab. 4. While the use of the average symbol provides the best results in the previous analyses, it is also expected to have the highest resource consumption. In particular, as with the single symbol, an external analog-to-digital converter (ADC) is required to record the entire signal or symbol at the appropriate sampling rate. For instance, a required sampling rate of 20 megasamples/second is assumed for the classic CAN, but due to the higher requirements respectively the shorter symbol duration of CAN-FD, higher sampling rates will be necessary. The acquisition of a composite symbol offers some advantages, as under certain assumptions regarding the used microcontroller it is possible to utilize the internal ADC for the acquisition. However, this requires a particularly fast comparator [KSH20] to be able to detect the individual level changes fast enough and the observed frames must have a certain amount of corresponding symbol transitions. In principle, both the acquisition of single and composite symbols require the least resources in

terms of calculation and storage, because in the case of the average symbol, it is necessary to store the entire signal in order to process it before the signal characteristics can be extracted. Provided that the computing capability of the implementing ECU is sufficient, however, this disadvantage can be compensated by calculating the running average. With the single and composite symbol this is omitted as the symbols can be used directly. Depending on the communication architecture of the vehicle under consideration, it may be necessary to analyze several CANs in parallel. For example, the networks of many vehicles are nowadays separated by domain or functions, which contributes to an increased security [RFS18]. In the case that multiple channels have to be observed, the usage of the single symbol is especially advantageous, as the sampling unit is only occupied for the time span of the symbol. For a single or a small number of bus segments, the composite symbol approach shows the lowest cost, but the complexity of time-critical sampling should not be underestimated. Reaching a high sender identification performance and a high robustness against fluctuations and signal changes potentially caused by environmental conditions and electrical consumers [KSH19], will allow to prevent attacks by disturbing ongoing transmissions of forged frames [Fo19; KH18]. If the possibility of preventing an attack is intended, the decision whether a forged frame is present must be made in a correspondingly short time. The type of signal acquisition has a considerable influence in this respect, since a certain number of symbols of the same type must have been transmitted for the generation of the composite symbol. A certain number of symbols of the same type are also used for the average symbol, whereby the amount can also be defined variably. For example, a time span can be defined after which all captured symbols are used for the calculation of the average symbol. In case only a single symbol is acquired during this time, this corresponds at least to the single symbol whose direct usage causes the least negative effect on the required processing time.

6 Conclusion

Basically, no significant differences in performance with respect to detection and identification rates could be determined by the different sampling methods. For this insight, on one hand the signals were analyzed directly and on the other hand the performance effects of the sampling method on a sender identification system were investigated. The small difference in performance enables the selection of the method based on the available resources and the underlying communication architecture. The average symbol, for example, not only shows slightly higher performance but also has the highest resource usage, while the recording of a single or composite symbol has advantages depending on the specific application. For single CAN buses the composite symbol is the most cost effective option, while the recording of a single symbol with an external ADC is advantageous for monitoring several CAN segments.

References

[Ca19] Cai, Z.; Wang, A.; Zhang, W.; Gruffke, M.; Schweppe, H.: 0-days & Mitigations: Roadways to Exploit and Secure Connected BMW Cars. Black Hat USA/, 2019.

[CA20] CAN in Automation: CAN XL is knocking at the door./, Jan. 2020, URL: https://www.can-cia.org/news/cia-in-action/view/can-xl-is-knocking-at-the-door/2020/1/3/, visited on: 01/03/2020.

[Ch18] Choi, W.; Joo, K.; Jo, H. J.; Park, M. C.; Lee, D. H.: VoltageIDS: Low-Level Communication Characteristics for Automotive Intrusion Detection System. IEEE Transactions on Information Forensics and Security 13/8, pp. 2114–2129, Aug. 2018, ISSN: 1556-6013.

[Fo19] Foruhandeh, M.; Man, Y.; Gerdes, R.; Li, M.; Chantem, T.: SIMPLE: Single-Frame Based Physical Layer Identification for Intrusion Detection and Prevention on in-Vehicle Networks. In: Proceedings of the 35th Annual Computer Security Applications Conference. ACSAC '19, Association for Computing Machinery, San Juan, Puerto Rico, pp. 229–244, 2019.

[HKD11] Hoppe, T.; Kiltz, S.; Dittmann, J.: Security threats to automotive CAN networks—Practical examples and selected short-term countermeasures. Reliability Engineering & System Safety 96/1, pp. 11–25, 2011, ISSN: 0951-8320.

[KH18] Kneib, M.; Huth, C.: Scission: Signal Characteristic-Based Sender Identification and Intrusion Detection in Automotive Networks. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. CCS '18, ACM, New York, NY, USA, pp. 787–800, 2018, ISBN: 978-1-4503-5693-0.

[Kn20] Kneib, M.: A Survey on Sender Identification Methodologies for the Controller Area Network. In (Reinhardt, D.; Langweg, H.; Witt, B. C.; Fischer, M., eds.): SICHERHEIT 2020. Gesellschaft für Informatik e.V., Bonn, pp. 91–103, 2020.

[KSH19] Kneib, M.; Schell, O.; Huth, C.: On the Robustness of Signal Characteristic-Based Sender Identification. 2019.

[KSH20] Kneib, M.; Schell, O.; Huth, C.: EASI: Edge-Based Sender Identification on Resource-Constrained Platforms for Automotive Networks. In: Proceedings of the 27th Network and Distributed System Security Symposium. 2020.

[LL18] Luo, Q.; Liu, J.: Wireless Telematics Systems in Emerging Intelligent and Connected Vehicles: Threats and Solutions. IEEE Wireless Communications 25/6, pp. 113–119, 2018.

[MG14] Murvay, P.; Groza, B.: Source Identification Using Signal Characteristics in Controller Area Networks. IEEE Signal Processing Letters 21/4, pp. 395–399, Apr. 2014, ISSN: 1070-9908.

[MV15] Miller, C.; Valasek, C.: Remote exploitation of an unaltered passenger vehicle. Black Hat USA 2015/, p. 91, 2015.

[RFS18] Ring, M.; Frkat, D.; Schmiedecker, M.: Cybersecurity Evaluation of Automotive E/E Architectures. 2. ACM Computer Science in Cars Symposium/, 2018.

[Ro12] Robert Bosch GmbH: CAN with Flexible Data-Rate Specification. 2012.

[Ro91] Robert Bosch GmbH: CAN Specification. 1991.

EAVE: Emotional Aerial Vehicle Evaluator

Marc Lieser[1] Ulrich Schwanecke[1] Jörg Berdux[1]

Abstract: Today, semi-autonomous quadrotors are already available at affordable prices and are rapidly becoming part of everyday life. To ensure that people feel safe around quadrotors and optimize flight times, their size should be kept to a minimum which results in their appearances remaining purely functional. This reduces the possibility of adding anthropomorphic or zoomorphic features that are typically used in order to increase acceptability by conveying the robot's inner state or intent. Constrained by mechanical appearance, other non-verbal communication channels can be exploited instead, in particular robot motion. The application *EAVE* presented in this paper was developed with the idea to design and evaluate trajectories that breathe life into inanimate, mechanical quadrotors in order to improve interaction in human-robot companionships. It extends our existing quadrotor testbed *ICARUS*, which is capable of tracking arbitrary trajectories of real and simulated quadrotors that were designed using *EAVE*. We demonstrate that applying some of the established principles of character animation to the design of quadrotor trajectories opens up the possibility of conveying intent and improving interaction, though the appearance of the quadrotor remains purely functional.

Keywords: human-quadrotor interaction; emotional aerial vehicles; non-verbal communication; motion anticipation; quadrotor companion; principles of animation; quadrotor testbed; social robots.

1 Introduction

Quadrotors experienced a huge gain in popularity over the past twenty years. Thanks to their mechanical simplicity and affordability, they have found their way from the hobby sector via research into everyday life. Most applications take advantage of their elevated view, whether for locating victims in disaster scenarios, for inspection of buildings or for filming movies. Private consumers mainly utilize quadrotors as self-flying cameras that accompany them during various leisure activities. Today, semi or even fully autonomous quadrotors are available and follow humans while avoiding obstacles and receiving commands via smartphones or gestures. This also enables them to be helpful companions in home environments, where in future scenarios they could accompany people through daily life as smartphones already do today. Whatever the exact purpose, the number of robot assistants, be it ground-based or of aerial nature, will most certainly increase and require research on possible communication channels for human-quadrotor interaction.

Even with today's state of technology and miniaturization, it is important and challenging to keep the size of the quadrotors to a minimum. Larger rotors of larger platforms, the generated

[1] Department of Computer Science, RheinMain University of Applied Sciences, Wiesbaden, Germany
{marc.lieser,ulrich.schwanecke,joerg.berdux}@hs-rm.de

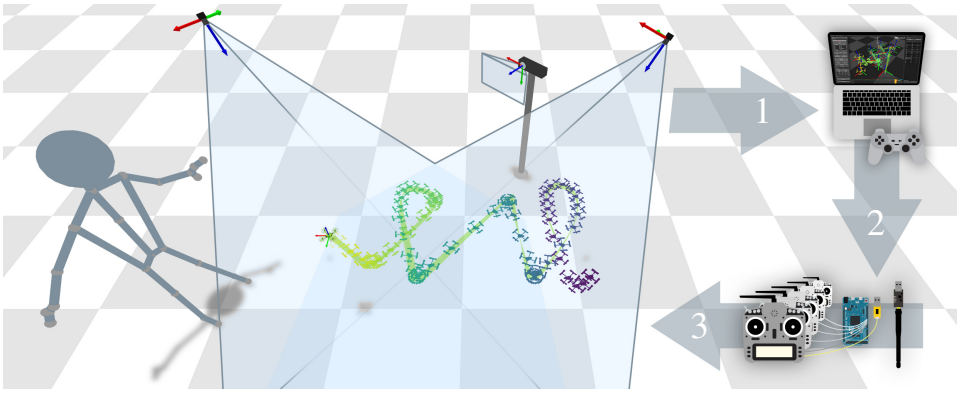


Fig. 1: Overview of our quadrotor testbed *ICARUS*, an affordable, portable indoor testbed for miniature quadrotors: After pose estimation of the individual quadrotors based on optical tracking (1), the control variables are determined (2) and sent via different radio control systems (3) to the quadrotors. The depicted schematic shows a quadrotor tracking an emotional trajectory designed with *EAVE* in a human-quadrotor interaction scenario.

noise and downwash are more likely to be perceived by people as disturbing [Ye17]. This introduces an inhibition threshold which is not desirable in companionships. For this reason, the quadrotors used are lightweight and cannot carry sensors or other electronics beyond ensuring their flight capability. Introducing anthropomorphic features is often used in order to make the robot's intent or inner state more identifiable for humans. This may be hard or even impossible to implement without adding to the already critical payload of quadrotors. Mediating motion intent without social cues such as gestures or gaze can be achieved by Augmented Reality (AR) applications [Wa18], but requires additional hardware. Instead of adding extra hardware, other channels of non-verbal communication can be further explored, in particular robot motion [HJ14], which is capable of transporting information beyond motion intent, namely the inner state of the robot or even its emotions. Hence, for quadrotors that have no moving parts other than rotors, the key to convey intent lies in their trajectories and their augmentation with motions familiar to humans. In order to bring the purely mechanical remaining robots to life and to increase acceptance, we apply some of Disney's well established principles of character animation [TJ81] that have already been adapted for robots other than quadrotors [RP12, GT11]. The desire to animate quadrotors exists [Ka17], but published research remains on a purely conceptual level [De18] or lacks detailed implementation and parameterization [Ca16].

In this paper we present the application *EAVE* (Emotional Aerial Vehicle Evaluator) that allows to design and evaluate quadrotor trajectories. Following animation, trajectories are defined by keyframes, each consisting of position, velocity, acceleration and heading angle. Some principles of animation can be applied and parameterized in order to increase the expressiveness of the quadrotor's motion with the objective to enhance user interaction by enabling trajectories to communicate the quadrotor's motion intent. Based on the keyframes,

smooth polynomial trajectories are generated and tracked by a quadrotor within our testbed *ICARUS* [Li17] that allows for safe test flights of arbitrary quadrotor models in a simulated environment but also controls real quadrotors. A schematic of the testbed is shown in Fig. 1.

In Section 2 we give a short overview of the current state of our quadrotor testbed *ICARUS* and point out where *EAVE* is extending the infrastructure. In Section 3 we describe the trajectory generation and provide some examples of applied principles of animation. Finally, we summarize our work in Section 4 and give an outlook on future research.

2 Overview

This section describes the current state of our general infrastructure *ICARUS*, that is extended by *EAVE* through software. A detailed description of *ICARUS* can be found in [Li17]. This low-cost quadrotor testbed was continuously expanded and improved over the past years. An overview showing its use in the scenario of designing emotional trajectories for quadrotors is shown in Fig. 1. In the following we describe the core components of the system and give an insight of the libraries used. The quadrotors used in our experiments and the overall data flow of *ICARUS* and *EAVE* are shown in Fig. 2.

2.1 Architecture

The *User Code* layer already consists of a classical Graphical User Interface (GUI) that can be seen in Fig. 3, manual control using a gamepad and a Natural User Interface (NUI) implementation using a Microsoft Kinect 2 in order to control the physical quadrotors or the

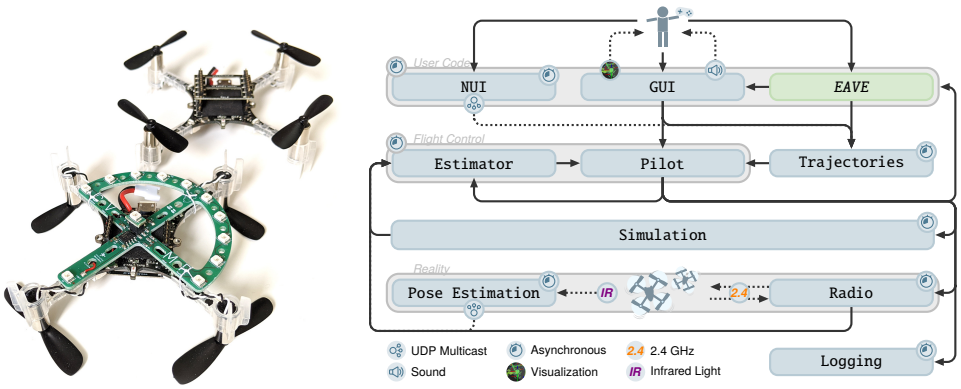


Fig. 2: Bitcraze Crazyflie 2.0 with the rotors mounted upside down so they do not obscure the LEDs of the attached tracking marker and an unmodified platform for comparison in the background (left). Data flow of *ICARUS* with *EAVE* as a component that extends the testbed by custom user code (right).

simulation. At this top layer, the existing infrastructure can be extended by custom user code like *EAVE*. Next to visual feedback, the user interface also generates synthetic rotor sound, a part of the trajectory design stage that should at least be considered. For that, the sound of a real quadrotor was recorded during hover and is played back during simulation. The sound is pitched by the ratio of rotor revolutions to the revolutions required for hovering. Upper and lower limits were chosen by experiment.

The implementation of the *Flight Control* layer uses a predictor–corrector estimator similar to the one described in [Lu14]. It filters noise from pose estimation and latency-compensates the quadrotors' states. Data from the estimator is used by the pilot implementation to hover in position or to track trajectories defined by the user. The control loop runs at a frequency of 100 Hz. For trajectory control a Model-Predictive Controller (MPC) with a discretization time of 0.1 s and a time horizon of 2 s based on the open-source code from [Fa18] is used. The hover controller was presented in [MK11]. The control variables are send to the radio or simulation and back to the estimator for the next control step. A configurable range of logging data is also sent to an asynchronous logging implementation for evaluation purposes. Our *Simulation* is based on the quadrotor dynamics summarized in [Mi10, Lu14].

For off-the-shelf (brushless) quadrotors in the *Reality* layer we use an Arduino-based serial remote control library [Li17] in conjunction with a FrSky Taranis X9D radio system. Less effort but shorter flight times come with the brushed Bitcraze Crazyflie 2.0, which we integrated using the radio library that is part of [HA17]. The in-house developed pose estimation system that we use is described in [Tj19]. The markers utilized in this system were optimized for the use with quadrotors and can be seen attached to a Crazyflie in Fig. 2. Pose estimation runs on a dedicated machine and multicasts UDP messages that consist of a six degrees of freedom pose and a timestamp. The estimator receives position and attitude from pose estimation and determines velocity and angular velocities by numerical differentiation. Telemetry data is also forwarded to the estimator and includes the current battery voltage to compensate the thrust command for varying voltages.

2.2 Libraries

The complete testbed *ICARUS* as well as the trajectory design and evaluation application *EAVE* are developed in `C++`. For multithreading and networking the `Boost` library is used. Serialization of data structures for UPD messages is implemented with `Protobuf`. Optical pose estimation is based on `OpenCV`, skeleton tracking is realized with the `Kinect 2 SDK`. The User Interface is developed with `Qt`, `OpenGL` and `glm`. Gamepad input uses `SDL`, audio output uses `SFML`. Radio control uses our general serial port approach `serialrc` .Communication with the Crazyflie uses the `crazyflie_cpp` library [HA17]. The simulation is implemented using `Eigen`. Optimization for the MPC is set up using `ACADO` and solved using `qpOASES` as described in [Fa18].

`https://github.com/cvmrgroup/serialrc`

3 Trajectories

For most robots it is completely sufficient to determine a trajectory on the shortest path between two points. This results in calculated straight or plain "mechanical" trajectories that render the identification of the robots inner state or intent impossible in a human-robot companionship. In our scenario, where the quadrotors themselves are kept mechanical, the trajectories should be designed to be able to transport happiness or sadness or whatever the quadrotor is up to in order to revalue quadrotors for human-robot companionship.

Trajectory control has already been described as part of the existing testbed in Section 2.1. Parameter tuning of the MPC influences smoothness, aggressiveness and accurateness of how a trajectory is tracked. How a trajectory is constructed as well as our design-approach of applying some of Disney's principles of animation to them is described in the following.

3.1 Trajectory Representation

We describe trajectories as piecewise quintic polynomials, as described for example in [Co13]. Quintic polynomials are a common choice for trajectory representation since their first and second temporal derivatives — velocity and acceleration — are continuous and thus result in smooth trajectories. Furthermore boundary conditions, such as position, velocity, acceleration and time, can easily be provided and solved for. A scalar trajectory quintic polynomial together with its first and second order derivates, given as

$$\begin{aligned} p(t) &= at^5 \quad bt^4 \quad ct^3 \quad dt^2 \quad et \quad f \\ \dot{p}(t) &= 5\,at^4 \quad 4\,bt^3 \quad 3\,ct^2 \quad 2\,dt \quad e \\ \ddot{p}(t) &= 20\,at^3 \quad 12\,bt^2 \quad 6\,ct \quad 2\,d \end{aligned}$$

result in a linear system

$$\begin{pmatrix} p(0) \\ \dot{p}(0) \\ \ddot{p}(0) \\ p(T) \\ \dot{p}(T) \\ \ddot{p}(T) \end{pmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ T^5 & T^4 & T^3 & T^2 & T^1 & 1 \\ 5T^4 & 4T^3 & 3T^2 & 2T^1 & 1 & 0 \\ 20T^3 & 12T^2 & 6T^1 & 2 & 0 & 0 \end{bmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \end{pmatrix} \tag{1}$$

with start boundary time $t = 0$ and end boundary time $t = T$, where T is the duration of the trajectory segment. For $T \quad 0$ the linear system (1) can be uniquely solved for the polynomial coefficient vector $\boldsymbol{c} = (a, b, c, d, e, f)^\top$. A trajectory $\boldsymbol{t}$ is represented by a list of coefficient vectors $\boldsymbol{c}$ along with the segment duration T, i.e. $\boldsymbol{t} = ((\boldsymbol{c}_0, T_0), \ldots, (\boldsymbol{c}_n, T_n))$. Scalar trajectories can be extended to the multi-dimensional vector case in a straightforward

way. Three-dimensional vectors of position, velocity and acceleration are pooled with time and quadrotor heading into keyframes, a term lent from animation. When a trajectory is finally sampled for the controller, the piecewise polynomials are interpolated independently in each dimension and joined to the final trajectory. The heading angle is interpolated linearly. As sampling rate we use 100 Hz according to the update rate of the controller used.

3.2 Design

Lists of keyframes can be loaded, edited and saved using the table view editor of *EAVE*, that can be seen in Fig. 3. Keyframe positions can be moved directly in the visualization, velocity and acceleration can be edited in the table view if necessary. The currently selected keyframe is highlighted in the visualization. Color palettes and lines connecting the keyframes indicate their order. It is also possible to run the simulation to show the actual trajectory, which varies depending on the controller tuning. Sound plays an important but obviously subordinate role at the design stage of trajectories. But it should be kept in mind during the design process, especially when focussing on keeping the interaction threshold as low as possible. For this reason we have introduced synthetic sound as part of the user interface as described in Section 2.1.

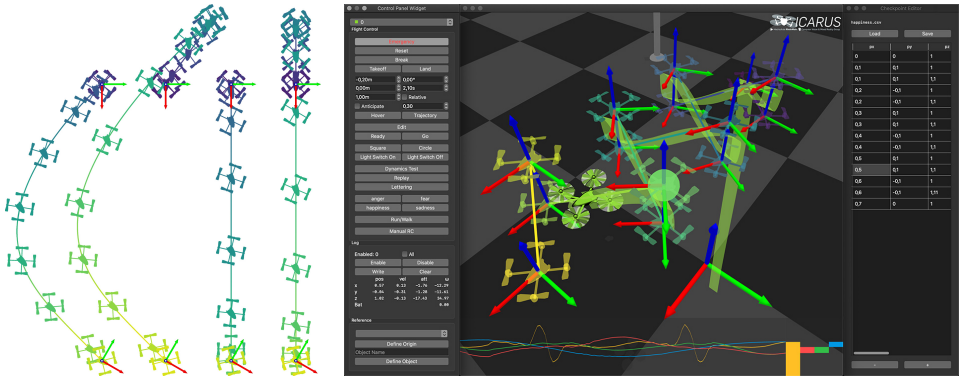


Fig. 3: Two pairs of trajectories, with and without anticipated motion and with different velocity information at the final keyframe (left). *EAVE*'s GUI with the control panel on the left side, trajectory editor on the right side and visualization in the middle (right).

3.2.1 Conveying Intent

It is difficult, if not impossible, to predict the intent of a mechanically designed quadrotor or of robots in general. Principles of character animation [TJ81] have proven beneficial in robot design [HJ14] but have not yet been applied to quadrotors. The use of quintic polynomials already implements some of those principles. Reducing the quadrotor's velocity at keyframes accomplishes the *Slow in and slow out* principle, while the smoothness of

quintic polynomials already describe *Arcs*, as the principle is named. *Timing* can also be adjusted according to visual pleasure. To avoid the user being surprised and thus intimidated by the quadrotor, we implement the design principle of *Anticipation*.

These principles applied can be seen in Fig. 3, where the motion of the quadrotor is anticipated by extrapolating the first piecewise polynomial by a user-defined time coefficient. Thus a counter movement is created right before the start of the actual motion. Extrapolating by large numbers is unpredictable but achieves well-suited results for the intend of anticipation when using low coefficients. Using polynomial extrapolation over linear extrapolation does not only transport information about the next keyframe's position but also adds velocity and acceleration information. Beyond that, a counter rotation is added by extrapolating the heading angle. Evaluating the level of improvement of interaction quality by allowing the user to anticipate the quadrotor's next movement as well as adding further animation principles will be part of our future work.

3.3 Evaluation

While evaluation of trajectory controllers address the robots's spatial and temporal deviation from the planned trajectory, there are several options for the evaluation of trajectories in terms of quality of user experience and quadrotor acceptance. User tests can be carried out directly in the testbed using real hardware or remotely using videos. Testing users on site is more elaborate to conduct, but obviously more realistic than online surveys. Online surveys on the other hand are able to reach a much larger test group. A test video can be filmed from the user's or a third person's perspective or rendered from the visualization. Videos can also be extended by AR to gain immersion for more elaborate tests, where the use of real hardware would be too dangerous, costly or complex. How well the anticipation described in Section 3.2.1 is understood could be evaluated by rendering the entire trajectory information of our 3D visualization on the screen of an AR device, e.g., a head-mounted display.

4 Conclusion & Future Work

In this paper we presented *EAVE*, a software extension to design and evaluate trajectories along with an updated description of the quadrotor testbed *ICARUS* used to control real and simulated quadrotors. We provided an updated description of the core components of said testbed. *EAVE* was implemented with the aim to design trajectories that are able to convey intent and thus improve quality of human-quadrotor interaction. As an example we applied an animation technique called anticipation to a given trajectory. Future work will include user tests on how well interaction can be improved by adding anticipation along with the implementation of further principles of animation. Also, we are currently working on a dynamic, situation-dependent human-quadrotor interaction scenario where the quadrotor's response is triggered by user behavior.

Bibliography

[Ca16] Cauchard, J. R.; Zhai, K. Y.; Spadafora, M.; Landay, J. A.: Emotion encoding in Human-Drone Interaction. In: HRI 2016. pp. 263–270, March 2016.

[Co13] Corke, Peter: Robotics, Vision and Control: Fundamental Algorithms in MATLAB. Springer Publishing Company, Incorporated, 1st edition, 2013.

[De18] Deng, Honghao; Li, Jiabao; Sayegh, Allen; Birolini, Sebastian; Andreani, Stefano: Twinkle: A Flying Lighting Companion for Urban Safety. In: TEI '18. pp. 567–573, 2018.

[Fa18] Falanga, D.; Foehn, P.; Lu, P.; Scaramuzza, D.: PAMPC: Perception-Aware Model Predictive Control for Quadrotors. In: IROS '18. pp. 1–8, Oct 2018.

[GT11] Gielniak, M. J.; Thomaz, A. L.: Generating anticipation in robot motion. In: 2011 RO-MAN. pp. 449–454, 2011.

[HA17] Hönig, Wolfgang; Ayanian, Nora: Flying Multiple UAVs Using ROS. In: Robot Operating System (ROS): The Complete Reference. Springer, pp. 83–118, 2017.

[HJ14] Hoffman, Guy; Ju, Wendy: Designing Robots with Movement in Mind. J. Hum.-Robot Interact., 3(1):91–122, February 2014.

[Ka17] Karjalainen, Kari Daniel; Romell, Anna Elisabeth Sofia; Ratsamee, Photchara; Yantac, Asim Evren; Fjeld, Morten; Obaid, Mohammad: Social Drone Companion for the Home Environment: A User-Centric Exploration. HAI '17, pp. 89–96, 2017.

[Li17] Lieser, M.; Tjaden, H.; Brylka, R.; Löffler, L.; Schwanecke, U.: A low-cost mobile infrastructure for compact aerial robots under supervision. In: 2017 European Conference on Mobile Robots (ECMR). pp. 1–6, Sept 2017.

[Lu14] Lupashin, Sergei; Hehn, Markus; Mueller, Mark W; Schoellig, Angela P; Sherback, Michael; D'Andrea, Raffaello: A platform for aerial robotics research and demonstration: The Flying Machine Arena. Mechatronics, 24:41–54, 2014.

[Mi10] Michael, Nathan; Mellinger, D.; Lindsey, Q.; Kumar, V.: The GRASP Multiple Micro-UAV Testbed. Robotics Automation Magazine, IEEE, 17(3):56–65, Sept 2010.

[MK11] Mellinger, D.; Kumar, V.: Minimum snap trajectory generation and control for quadrotors. In: ICRA 2011. pp. 2520–2525, May 2011.

[RP12] Ribeiro, T.; Paiva, A.: The illusion of robotic life: Principles and practices of animation for robots. In: HRI '12. pp. 383–390, 2012.

[TJ81] Thomas, F.; Johnston, O.: The Illusion of Life: Disney Animation. Disney Editions, 1981.

[Tj19] Tjaden, Henning: Robust Monocular Pose Estimation of Rigid 3D Objects in Real-Time. PhD thesis, Johannes Gutenberg University Mainz, Germany, 2019. https://nbn-resolving.org/urn:nbn:de:hebis:77-diss-1000025478, last accessed 06 May 2020.

[Wa18] Walker, Michael; Hedayati, Hooman; Lee, Jennifer; Szafir, Daniel: Communicating Robot Motion Intent with Augmented Reality. In: HRI '18. Association for Computing Machinery, New York, NY, USA, p. 316–324, 2018.

[Ye17] Yeh, Alexander; Ratsamee, Photchara; Kiyokawa, Kiyoshi; Uranishi, Yuki; Mashita, Tomohiro; Takemura, Haruo; Fjeld, Morten; Obaid, Mohammad: Exploring Proxemics for Human-Drone Interaction. In: HAI '17. ACM, New York, NY, USA, pp. 81–88, 2017.

Citcom – Citation Recommendation

Melina Meyer[1] Jenny Frey[2] Tamino Laub[3] Marco Wrzalik[4] Prof. Dr. Dirk Krechel[4]

Abstract: Citation recommendation aims to predict references based on a given text. In this paper, we focus on predicting references using small passages instead of a whole document. Besides using a search engine for baseline, we introduce two different approaches. All more advanced models are based on neural networks. One of them consists of two sub models, having an embedding model to create combined embeddings from meta data and the reference. Additionally, the model uses feature engineering and a feed forward network. The second model takes advantage of BERT, a language representation model, to tokenize end encode the passages. It predicts references using passages similarity based on BERT embeddings. For training and evaluation of our models, we prepare a large dataset consisting of English papers of different scientific disciplines.

Keywords: citation recommendation; natural language processing; representation learning

1 Introduction

Writing scientific papers or any kind of work that requires a lot of citation can be very exhausting. A lot of research is necessary to find documents that are close to the topic one is working on and are worth citing to proof a point. It would be a great assistance to have a tool which helps to find related work that can be associated with the topic one is writing about. The task focusing on this problem is called citation recommendation. It can be be divided into two categories: global approaches and context-based recommendation. While approaches of the first category pay attention on the whole document, context-aware attempts focus on the context of the current citations, meaning some sentences before and after. After considering various possibilities we decided on pursuing three different context-based approaches to this task. The one is to create a simple prove of concept application in order to get a benchmark for recommendation performance. The two more advanced models are based on neural networks. As a baseline experiment, we decided on using Elasticsearch, a full-text search engine that can find documents due to their structural

[1] RheinMain University of Applied Sciences, Faculty of Design Computer Science Media, Unter den Eichen 5, 65195 Wiesbaden, Germany melina.meyer@student.hs-rm.de
[2] RheinMain University of Applied Sciences, Faculty of Design Computer Science Media, Unter den Eichen 5, 65195 Wiesbaden, Germany jenny.frey@student.hs-rm.de
[3] RheinMain University of Applied Sciences, Faculty of Design Computer Science Media, Unter den Eichen 5, 65195 Wiesbaden, Germany tamino.laub@student.hs-rm.de
[4] RheinMain University of Applied Sciences, Working Group LAVIS – Learning and Visual Systems, Unter den Eichen 5, 65195 Wiesbaden, Germany lavis@hs-rm.de

similarity. In order to achieve better results, the second model uses feature engineering and a feed forward network. Additionally, we wanted to use a tool that learns characteristics of the language used in the documents and takes semantics into consideration, which brought us to BERT and is used in the third model. This model learns passage similarity using BERT embeddings to predict references. For evaluation, we create a data set which is based on a dump of arXiv documents of all scientific disciplines from recent years.

2 Related Work

Citation Recommendation is a complex task that has gained increasing attention in recent years. Besides some methods for proposing references for entire documents [CHY18; Kü12; Re14], many new publications refer to the context-based approach. Local recommendation using contextual information was first introduced by [He10]. Based on this idea, [Hu12] proposed an approach via a machine translation system transfering keywords of the context into cited documents. They continued this work in [Hu15] by adding semantic embeddings of the words of the context as well as cited documents, a recommendation is carried out based on the semantic distance in vector space. [TWZ14] presented the first embedding-based approach for context-aware citation recommendation, which uses TF-IDF vectors to form cross-language embeddings and uses them for the proposals. Approaches without neural networks based on information retrieval techniques and metrics such as TF-IDF or BM25 also have been investigated. [Du16] annotate in their proposals each sentence of the given documents with CoreSC classes, which are indexed in Lucene and used to determine similarity. On the other hand, [EF17] use BM25 as a baseline for their encoder-decoder framework, inspired by neural machine translation, which learns relations between text pairs of variable length. By additionally analyzing the writing style of authors, performance should be further improved. Other procedures also include document metadata. [FS20] is a semi-genetic hybrid recommender system for citation recommendation. The authors combine embedding and information retrieval approaches using a fitness score to receive the top k recommendations. Recent approaches additionally include language models such as BERT [De19]. In [Je19], BERT is used as a context encoder for textual embeddings, supplemented by a Graph Convolutional Network (GCN) for metadata and reference relationship between papers as a citation encoder for the construction of graph embeddings. The concatenation of the output vectors is then used as input of a feed forward network.

3 Dataset

In order to recommend citations for scientific papers and to evaluate our model on a large dataset, we reuse the record provided in [SF20]. The data set is based on an arXiv source dump including papers from 1991 to 2018 and of all scientific disciplines available on arXiv.org. It offers 29.2 million ready-made citation contexts, each consisting of three sentences: the sentence containing the citation and the two surrounding ones. We use these

contexts to find the corresponding sections in the full text files, extract all references of the section and map them to the arXiv- and/or MAG-ID of the cited paper. We only consider references that could be matched to arXiv- or MAG identifier. Furthermore we use a dump of arXiv metadata[5] to obtain basic background information about the cited papers. Since MAG metadata are difficult to obtain for research purposes, we limit our experiment to passages that reference known arXiv papers. Finally, training and test data are divided in the ratio 80:20. Since our models require a lot of computing power, we create another data set to reduce the computing time. We cluster the passages into 176 clusters based on the scientific categories of the papers. From each cluster, five percent of the references were randomly selected and citing passages are part of the reduced data set. This dataset was also randomly split in the ratio of 80:20. Table 1 illustrates the details of our datasets containing the total number of disjoint references, training passages, disjoint references contained by the training passages and the amount of test passages.

Data Set	#Refs	#Train Pass.	#Train Refs	#Test Pass.
full	718,329	14,932,722	700,220	3,733,181
reduced	115,643	1,339,920	115,643	329,130

Tab. 1: An overview of the employed datasets

4 Model Architectures

4.1 Baseline Model

In order to get an initial orientation, a baseline experiment is first performed using Elasticsearch[6]. Elasticsearch is a search engine based on the Lucene program library for full text searches. The engine allows adding documents to indices and searching similar documents to a query using a RESTful API. For both of the datasets, a new index is created. Due to the large amount of training data, the passages are grouped according to their content. This means that passages with the same context but different target references are identified and only indexed once. Each indexed document contains a text field with `english` language analyzer as well as two keyword fields for lists of all associated passage IDs and target references. This reduces the amount of training data to a total of 8,700,191 indexed documents for the full data set and 1,199,747 for the reduced one. Especially for the full data set, this has a significant impact on the time performance of the subsequent evaluation.

4.2 Feed Forward Model with Feature Engeneering

In our first model, we decided to attempt a feature based approach to achieve better results when trying to predict citations. This model consists of two submodels. The first sub model

[5] Downloaded from https://archive.org/details/arXiv-metadata-dump-2019-06-18.tar.xz
[6] See https://www.elastic.co/

focuses on Feature Engineering. The features of each passage that we want to use for our embedding are TF-IDF scores and the passage length. To have equally sized feature vectors for each of the passages, we decided to include the 100,000 words with the most occurences across all passages. The architecture of this model is presented in Figure 1.

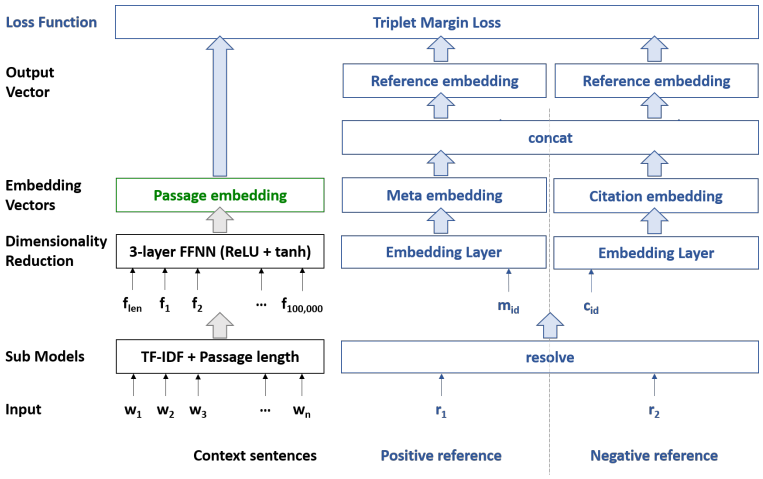


Fig. 1: Architecture of our feature based approach

Each passage is therefore represented as a vector of 100,001 dimensions. For the passage length, we calculated the average length of our corpus and then divided the length of the current passage through this value. The other 100,000 dimensions of the vector are made up of the TF-IDF values each of the most common 100,000 words yields across all passages, sorted by the occurences in the corpus. We chose a three-layer Feed Forward Neural Network (FFNN) to convert the vectors down to 300 dimensions. The network consists of three linear layers, the first two are provided with ReLU activation function and tanh is utilizied for the third layer. Since we use a Triplet Margin loss for training, two references per passage are needed to improve network performance. For the first, we use the correct reference featured in the original text passage and the second one is a randomly chosen reference from the corpus. The required embeddings of the references are obtained by the second sub model. The second sub model is used to encode the references. Additionally, the meta information shall be considered to improve the prediction of suitable references. As meta information, the scientific category of the cited paper is considered. For this reason, each embedding of a reference consists of a meta embedding and a citation embedding. The meta embedding refers to a feature that is shared by several references. So the references are grouped by this feature. The citation embedding refers to the reference itself. The reference data is used as input, which is filtered according to the required features. Based on this, two identifier for metadata and the citation are determined, which are converted into vectors via embedding layers. Both embedding types possess their own lookup table and are concatenated for each reference. The meta embedding has a dimension of 100 and the citation embedding's dimension is 200, leading to an output embedding with a dimension of 300.

4.3 BERT Passage Model

For our third model, we use a pre-trained english BERT model [Wo19] to encode a passage. BERT (Bidirectional Encoder Representations from Transformers) is a language representation model that achieves state of the art results on different natural language processing tasks. Training a BERT model consists of two steps: pre-training and fine-tuning. In addition, BERT uses some special tokens, these are among others the [CLS] token and the [MASK] token. In the first step we use a BERT tokenizer [Wo19] to convert the passage into an id sequence. As it is possible to use a [MASK] token to hide special tokens, we replace the references in the passages by this. Additionally, the encoded sequence has the classification token [CLS] at the beginning. Like all tokens of the sequence, this token is transformed into an embedding and corresponds to sequence representation for different classification tasks. The converted sequence is fed into BERT to receive the word embeddings and the classification embedding of the [CLS] token of the passage. We further only use the classification embedding which is used for fine-tuning to predict references. Our first idea was to reuse the reference sub model described in the previous section. The aim was to train these sub models again by Triplet Margin Loss and fine-tune BERT, so that the encoded [CLS] embedding refers to the positive reference embedding. As this model did not perform well on a first trial data set, we have decided to use other passages for training instead of the references. The aim is to predict references using similar passages as illustrated in Figure 2.

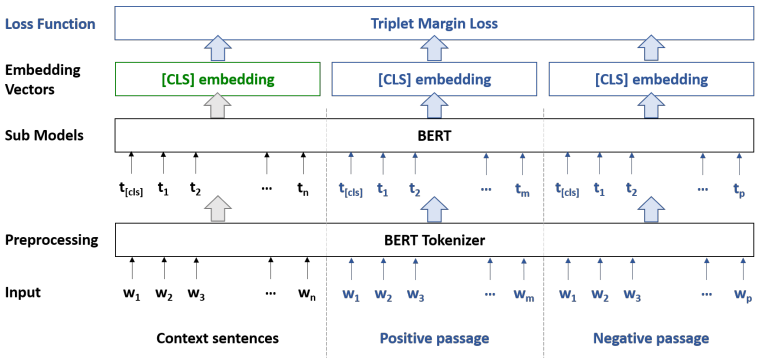


Fig. 2: Architecture of our passage-based BERT approach

For this reason, a data sample consists of a passage for which we want to predict the reference (relevant passage), a passage with the same reference (positive passage), and a passage with another reference (negative passage). As the sample requires two passages with the same reference, we only use passages if at least one other passage with the same reference exists. All passages are fed into BERT to receive the [CLS] embeddings. BERT is fine-tuned so that it learns a similarity between the relevant and the positive passage. To predict the reference of the relevant passage, the cosine similarity between the classification embedding of this

passage and all other passages is calculated. The final reference is determined by the known reference of the passage that is most similar to the relevant passage.

5 Training and Evaluation

Most of citation recommendation tasks use well-known metrics such as Mean Average Precision (MAP), Recall, Mean Reciprocal Rank (MRR), Normalized Discounted Cumulative Gain (NDCG) and hits@k for evaluation. We followed these approaches and decided to use MRR@10, hits@10 and hits@5.

5.1 Baselines

As there are many training passages, the training data was grouped by contexts and inserted one single time for each content yielding corresponding passage IDs and target references. As search query, a *More Like This Query* is formed which promises a better timing performance than usual queries for large indices. The query is mapped to the content field and gets the context of a test passage as *like* statement. In addition, due to the brevity of the context, the minimum term frequency is reduced to 1 and the minimum document frequency to 3. The maximum number of query terms is decreased to 10 to further reduce response times. Using different thread workers, each query is posted separately also returning the top ten results. For each result, the proposed target references are collected and afterwards sorted by their frequency of occurence. Finally, the ten top most references are extracted and browsed for the ground truth reference, which serves as the relevant item for calculation of MRR, hits@10 and hits@5. Table 2 illustrates the evaluation results for the reduced dataset. As the table shows, the baseline already provides respectable results.

Model	MRR@10	Hits@10	Hits@5
Elasticsearch	0.546393	0.793804	0.758596
Feature FFNN	0.000042	0.000125	0.000073
BERT Passage	0.582673	0.763704	0.713639

Tab. 2: Evaluation of the models for the reduced dataset

5.2 Feed Forward Network with TF-IDF

For training the feature embedding network we use a GeForce GTX 1080 with a batch size of 8. We train the model for three epochs using the Adam optimizer for both sub models with a learning rate of 1e−3 on the reduced dataset. We also evaluate this model by MRR for a maximum rank of ten, hits@10 and hits@5. The results of this approach are shown in Table 2. It is easy to see that the scores achieved by this approach are way lower than the

other numbers. A reason for this could be that the model only takes limited context into account when looking for similar passages, mostly focusing on the reference itself. These extremely low results may also be the drastic reduction of the dataset due to computation time which results in having only a few passage examples per unique reference.

5.3 BERT Passage Model

For fine-tuning BERT we use AdamW [LH17] optimizer with a learning rate of 1e−6. AdamW is an optimizer with decoupled weight decay is suitable for fine-tuning BERT. Due to the amount of data, we train the model on four GPUs, three GeForce RTX 2080 Ti and a GeForce GTX 1080 Ti, in parallel with a total batch size of 10 for 3 epochs on the reduced dataset. Table 2 shows the evaluation using Mean Reciprocal Rank, hits@10, and hits@5. While hits@10, and hits@5 is lower than the baseline, MRR@10 provides the best results on the dataset. When training a further epoch, it has been shown that the results of the metrics change only at the third decimal place, which shows that the model converges stably.

6 Conclusion

Recommending citations for a given query is a complex task that will keep researchers engaged for a long time. In our paper we proposed some context-based attempts that presented us with many challenges. These models are based on different approaches, which all have shown their advantages and disadvantages. We presented them in the context of our work and attempted to improve them continuously. The baseline evaluation was particularly striking, since even a pure textual similarity already yields respectable results. Based on this, we tried to use and further improve these features with our models and to try out further attempts with different approaches putting them into contrast. The results of the presented models provided some surprises, but through continuous optimization the results could be increasingly improved. The model based on learning citation embedding and meta embedding leads poor results in comparison to the Elasticsearch baseline. On the contrary, reference prediction by learning passage similarities using BERT embeddings finally gives the best results on MRR but provides poorer results on hits@10 and hits@5 than the baseline calculation. In general, the models using passage similarity for reference prediction provide the best results for our use case and a well-functioning citation recommendation solution.

References

[CHY18] Cai, X.; Han, J.; Yang, L.: Generative Adversarial Network Based Heterogeneous Bibliographic Network Representation for Personalized Citation Recommendation. In: Proc. 32nd AAAI Conf. on Artificial Intelligence. AAAI Press, pp. 5747–5754, 2018.

[De19] Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: NAACL-HLT. 2019.

[Du16] Duma, D.; Liakata, M.; Clare, A.; Ravenscroft, J.; Klein, E.: Rhetorical Classification of Anchor Text for Citation Recommendation. D-Lib Magazine 22/, Sept. 2016.

[EF17] Ebesu, T.; Fang, Y.: Neural Citation Network for Context-aware Citation Recommendation. In: Proc. 40th Int. ACM SIGIR Conf. on Research & Development in Information Retrieval. Pp. 1093–1096, 2017.

[FS20] Färber, M.; Sampath, A.: HybridCite: A Hybrid Model for Context-Aware Citation Recommendation. arXiv preprint arXiv:2002.06406/, 2020.

[He10] He, Q.; Pei, J.; Kifer, D.; Mitra, P.; Giles, L.: Context-aware Citation Recommendation. In: Proc. 19th Int. Conf. WWW. Pp. 421–430, 2010.

[Hu12] Huang, W.; Kataria, S.; Caragea, C.; Mitra, P.; Giles, C. L.; Rokach, L.: Recommending Citations: Translating Papers into References. In: Proc. 21st ACM Int. Conf. on Information & Knowledge Management. Pp. 1910–1914, 2012.

[Hu15] Huang, W.; Wu, Z.; Liang, C.; Mitra, P.; Giles, C. L.: A Neural Probabilistic Model for Context based Citation Recommendation. In: 29th AAAI Conf. on Artificial Intelligence. 2015.

[Je19] Jeong, C.; Jang, S.; Shin, H.; Park, E.; Choi, S.: A Context-Aware Citation Recommendation Model with BERT and Graph Convolutional Networks. arXiv preprint arXiv:1903.06464/, 2019.

[Kü12] Küçüktunç, O.; Kaya, K.; Saule, E.; Çatalyürek, Ü. V.: Fast Recommendation on Bibliographic Networks. In: 2012 IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining. Pp. 480–487, 2012.

[LH17] Loshchilov, I.; Hutter, F.: Decoupled Weight Decay Regularization. arXiv preprint arXiv:1711.05101/, 2017.

[Re14] Ren, X.; Liu, J.; Yu, X.; Khandelwal, U.; Gu, Q.; Wang, L.; Han, J.: ClusCite: Effective Citation Recommendation by Information Network-based Clustering./, Aug. 2014.

[SF20] Saier, T.; Färber, M.: unarXive: A Large Scholarly Data Set with Publications' Full-text, Annotated in-text Citations, and Links to Metadata. Scientometrics/, pp. 1–24, 2020.

[TWZ14] Tang, X.; Wan, X.; Zhang, X.: Cross-language Context-aware Citation Recommendation in Scientific Articles. In: Proc. 37th Int. ACM SIGIR Conf. on Research & Development in Information Retrieval. Pp. 817–826, 2014.

[Wo19] Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Brew, J.: HuggingFace's Transformers: State-of-the-art Natural Language Processing. ArXiv abs/1910.03771/, 2019.

Bidirectional Transformer Language Models for Smart Autocompletion of Source Code

Felix Binder, Johannes Villmow, Adrian Ulges[1]

Abstract: This paper investigates the use of transformer networks – which have been recently become ubiquitous in natural language processing – for smart autocompletion on source code. Our model *JavaBERT* is based on a RoBERTa network, which we pretrain on 250 million lines of code and then adapt for method ranking – i.e. ranking an object's methods based on the code context. We suggest two alternative approaches, namely unsupervised probabilistic reasoning and supervised fine-tuning. The supervised variant proves more accurate, with a top-3 accuracy of up to 98%. We also show that the model – though trained on method calls' full contexts – is quite robust with respect to reducing context.

Keywords: smart autocompletion, deep learning, transformer networks

1 Introduction

AI-based support in software engineering has recently emerged as a research field, and recommenders for software commits [Da16], predicting code changes [Zh19] or semantic code search [Hu19] have been developed. These are usually trained on vast amounts of source code and documentation from open-source platforms such as GitHub. Another challenge – and the subject of this paper – is *smart autocompletion*: As the developer types source code, a *method ranking network* suggests names for methods to use next. We refer to this challenge of ranking an object's method names by their plausibility in a given code context as *method ranking*. Figure 1 illustrates this, where a neural network has learned a suggestion from GitHub projects including similar code passages as the target context.

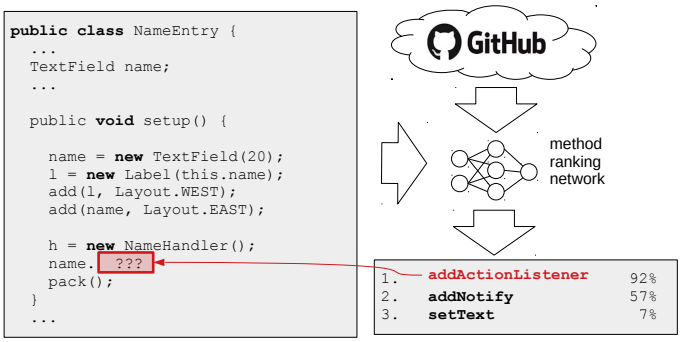


Fig. 1: A method ranker analyzes a position in Java source code (red, left), and infers that – out of the class `TextField`'s methods – `addActionListener()` seems most plausible.

[1] RheinMain University of Applied Sciences, DCSM Department, Wiesbaden/Germany
felix.d.k.binder@gmail.com,[johannes.villmow,adrian.ulges]@hs-rm.de

While previous work on method ranking has used *n*-grams [Hi12] or recurrent networks [Wh15], we evaluate the transformer-based *masked language model* BERT [De18] (more precisely, its RoBERTa variant [Li19]). This approach has been very successful in natural language processing, but there has been – to the best of our knowledge – only one recent publication on smart autocompletion in source code [Ki20]. While this work uses a autogenerative model, we employ *masked* language modeling, i.e. training is done by masking out tokens in pieces of source code and forcing the model to predict those missing tokens. We call the resulting model *JavaBERT* (since our focus lies on the Java programming language).
To utilize JavaBERT for method ranking, we propose two alternatives addressing the fact that method names may consist of multiple tokens (e.g., `add/Action/Listen/er`).:

1. *JavaBERT-unsup*: The pre-trained (unsupervised) JavaBERT model is applied by masking out varying numbers of tokens. JavaBERT's predictions on token level are then combined in a probabilistic reasoning to predictions on method level.
2. *JavaBERT-sup*: JavaBERT is fine-tuned supervisedly as a binary classifier, estimating whether a certain method call is plausible or not in a given code context.

We evaluate both models in quantitative experiments on random samples from the GitHub Java Corpus [AS13]. Our results indicate that masked language modeling is surprisingly accurate, with a top-3 accuracy of up to 98%. We also study the impact of different contexts, e.g. only the code up to the target method call, or shorter vs. larger pieces of code.

2 Related Work

Smart autocompletion The task of code completion has been addressed since 2012 by using n-gram models [Hi12, AS13], cached *n*-gram models for improved localization [Fr15, TSD14, HD17] and graph-based statistical language models [NN15]. More recently, the availability of large code bases has facilitated the creation of neural network language models, including recurrent neural networks [Wh15, Ra16, Li17], gated recurrent neural network models [KS19] and LSTM models [Da16]. Most recent code completion models for Java use a single layer gated RNN [Ka20] model with Byte-Pair Encoding [SHB16].

Transformer networks in NLP Recently, Transformer networks [Va17] have shown great potential in the field of natural language processing. These models use the concept of attention [BCB14] to derive contextualized representations for the single tokens from a sequence (i.e. a sentence or paragraph). Probably the most prominent model is BERT [De18], which applies masked language modeling, i.e. the model is trained to predict random masked tokens in the training text. Other variants use generative transformers (GPTs) trained by left-to-right language modeling [Ra18, Ra19], optimize hyperparameters such as model depth and learning rate (RoBERTa [Li19]), reduce the amount of parameters (ALBERT [La20]) or perform an adversarial training (ELEKTRA [Cl20]). While transformer models have been extremely successful and intensely studied in processing *natural language* recently, we are only aware of one recent publication employing them for code completion [Ki20].

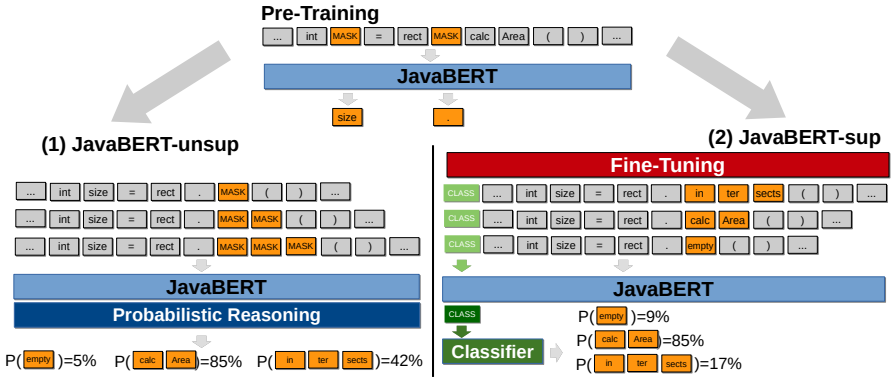


Fig. 2: Our approach JavaBERT is first pretrained using Masked Language Modeling (top). Afterwards, method ranking can either use masking with probabilistic reasoning (JavaBERT-unsup, left) or fine-tuning with a binary classifier (JavaBERT-sup right).

3 Approach

As shown in Figure 2, our approach pretrains an encoder (JavaBERT) by masked language modeling. The resulting model can either be applied in an unsupervised fashion (using a probabilistic reasoning) or by fine-tuning it into a supervised binary classifier. We discuss these three processing steps in depth in the following subsections.

3.1 Pretraining

Our approach starts with training a RoBERTa model on the Github Java Corpus [AS13] using the *fairseq* library [Ot19]. RoBERTa replicates and improves key hyperparameters of the famous BERT model [De18] for a more robust training.
The training set of the Github Java Corpus consists of around 1.1 billion tokens of Java source code, from which we separate the last 5% into a validation set. We tokenize all Java code with a language parser [4] to separate natural language identifiers from syntax symbols. Thereby we also remove multiple whitespace and replace string, character, float, and integer literals with a respective constant (e.g. `<INT>`). As source code may contain unicode identifiers, we use the *unidecode* library to transcribe any non-ASCII letters into ASCII.
Afterwards we train and apply a Byte-Pair Encoding [SHB16] of V=10,000 sub-words on the tokenized source code. We perform neither lower-casing nor camel-case splitting as it is often done in machine learning on source code [ALY18]. A vocabulary of $10K$ tokens is used, which we assume is sufficient for source code (most identifiers are rather short or combined with camel-casing) while improving training speed. For example, our preprocessing tokenizes `public float myFloat = 10.0;` into `public/float/my/Float/=/<FLOAT>/;`.

[4] We use the *javalang* library for tokenization.

Model and Training We use a configuration similar to the RoBERTa$_{BASE}$ model (12 layers, 768-dimensional embeddings, 12 attention heads, 110M params total). Like ToBERTa, we use GELU activation functions [HG16], learned positional embeddings and a dropout of 0.1 throughout all layers and activations. The model is optimized with the Adam optimizer [KB14] (β_1=0.9, β_2=0.98, ϵ=10^{-6}, weight decay 0.01) using a linear warm-up of the learning rate for $6K$ steps up to $6{\times}10^{-4}$ followed by a cosine decay over $30K$ steps.
For efficiency reasons, we first train $15K$ steps on shorter code blocks of 128 tokens each (batch size $8K$) and then increase the block length to 512 (batch size $1,500$). The sampled blocks do not cross document (i.e. source file) boundaries. We use *gradient accumulation* to mimic larger batch sizes on our limited hardware (6 NVIDIA GeForce GTX 1080 Ti).
After a total training time of about two weeks, the JavaBERT model reached a validation perplexity of 1.16 for predicting masked out tokens, which is considered excellent compared to language modeling on *natural* language. This indicates that language modeling on structured language reaches higher accuracy compared to more complex unstructured text.

3.2 Unsupervised Method Ranking (JavaBERT-unsup)

We assume an incomplete piece of code to be given, which is a sequence of n tokens $T{:=}(t_1, \ldots, t_s, \ldots, t_n)$ containing a slot t_s for the missing method call, and a set of candidate method names $\{C^1, \ldots, C^l\}$ which contains the correct method call C^* as well as other method names from the same class as C^*. The task is to maximize the probability $P(C^*|T)$. Note that – after tokenization – each candidate method name may consist of multiple tokens, i.e. $C{=}(c_1, \ldots, c_L)$ with $L{>}1$.
Note that the JavaBERT model's original training is *similar* to method ranking: The original source sequence is transformed into a sequence T', in which – similar to our input sequence – random tokens t_i have been masked out by replacing them with a `<mask>` token. During training the probability $P(T_i{=}t_i|T')$ of predicting the masked token is maximized. The key difference with method ranking is that *multi-token* method names have to be predicted. To do so, we calculate the probability of a candidate method (e.g., $C{=}(c_1, .., c_L)$) by replacing the slot token t_s with L `<mask>` tokens, obtaining a sequence T^L. Then, the overall probability of candidate C is defined as

$$P(C|T) = P(L) \cdot \prod_{j=1}^{L} P(T_{s\ j-1}{=}c_j|T^L) \tag{1}$$

$P(L)$ acts as a prior on method name length, exploiting the fact that shorter names are more likely (an estimate on the training set is given in Table 1).

L	1	2	3	4	5	6	7	8	9	10+
$P(L)$	26.6%	23.9%	21.0%	12.4%	6.9%	3.9%	2.2%	1.2%	0.7%	1.2%

Tab. 1: The prior $P(L)$ indicates the probability of different method name lengths L.

3.3 Supervised Method Ranking (JavaBERT-sup)

Our second approach *fine-tunes* the pretrained JavaBERT for method ranking using a supervised training. The basic idea is to insert candidate method names into code blocks and estimate their plausibility with a binary classifier. To do so, we use an classifier token t_{CLASS} (as is common practice), replace the slot token t_s with the candidate C, and add markers t_{START} and t_{END} before the bound object t_{OBJ} and after the method call, resulting in the input sequence

$$T^C := (t_{CLASS}, \ldots, t_{START}, t_{OBJ}, t_{dot}, \overbrace{c_1, \ldots, c_L}^{=C}, t_{END}, \ldots, t_n).$$

We encode this sequence with the JavaBERT model and feed the resulting contextualized classifier token into a binary classifier, which is trained to predict whether T^C contains the correct method name. The classifier first projects the encoded representation with a linear layer into another embedding space of the same dimensionality as the JavaBERT model, followed by layer normalization and a second projection to our binary output space.
A training set of $3.3M$ labeled code blocks is constructed of $2,649$ repositories from the GitHub Java Corpus' test split. Each positive sample (containing the true method call) is complemented with six negative samples, three of which feature another method name from the same class and the other three containing another random method name from the corpus.

4 Experiments

This section compares the models JavaBERT-sup and JavaBERT-unsup in quantitative experiments on held-out test data from the Github Java Corpus. We also analyze how different amounts of context information affect the model's accuracy.
For this we use another part of original test split, which consists of 969 repositories that are not overlapping with the repositories used for pretraining or fine-tuning from Section 3. From these test projects, we sample $14K$ random code blocks of up to 504 tokens each. In each block, a randomly selected method call is chosen as slot t_s, and a list of candidate method names to be ranked is extracted from the method call's bound object's class. The median length of those candidate lists is 39.

Comparison Unsupervised vs. Supervised We compare the supervised and unsupervised model by measuring the hits@1, hits@3 and hits@5 rates, and the mean reciprocal rank (MRR). For example, a hits@3 of 98% indicates that for 98% of our $14K$ test blocks, the model ranks the correct method name among the top 3. Table 2 illustrates the results. Both approaches surpass a baseline that ranks the target methods randomly and the supervised approach outperforms the unsupervised model by a significant margin, especially for the top 1 predictions (difference $\approx$ 14%) and mean reciprocal rank (difference $\approx$ 10%).
Figure 3 illustrates an example on a random piece of code. Here, both models (supervised and unsupervised) rank the 8 candidate methods from Class `Scanner`, and the method `nextInt()` is ranked highest correctly. Overall, we found the model to prefer methods

	JavaBERT-unsup	JavaBERT-sup	Random guessing
hits@1	77.5	**92.3**	7.4
hits@3	92.5	**98.0**	20.8
hits@5	94.6	**98.9**	29.9
MRR	85.4	**95.2**	18.3

Tab. 2: Results of method ranking: The supervised approach significantly outperforms the unsupervised one and shows remarkable accuracy (hits@3 is 98%). We report all values as a percentage.

```java
import java.util.Scanner;

public class EvenOdd {
  public static void main(String[] args) {
    Scanner reader = new Scanner(System.in);
    System.out.print("Enter a number: ");
    int num = reader . SLOT ();
    String evenOdd = (num % 2 == 0) ? "even":"odd";
    System.out.println(num + "is" + evenOdd);
  }
}
```

JavaBERT-unsup	JavaBERT-sup
1: nextInt	1: nextInt
2: nextLong	2: nextShort
3: nextShort	3: close
4: nextByte	4: hasNext
5: skip	5: nextByte
6: close	6: skip
7: hasNext	7: locale
8: locale	8: nextLong

Fig. 3: Given this example code (left) with a left out target method (`SLOT`), both JavaBERT variants rank the correct method (`nextInt()`) out of 8 candidate methods highest.

with the correct parameters and return types (e.g., boolean methods are ranked high in if-statements). Note that this is not inferred from a static code analysis but only from the method name (e.g., `isEmpty`, `hasConnection`). Also, we found methods that have already been defined or used in the context to be ranked higher.

Context Analysis So far, we have trained and tested our model on *full* code contexts, including the code before and after the target method as well as the passed arguments. In practice, e.g. when typing code from left to right, only the code before the target may be available. Therefore, we evaluate JavaBERT-sup (trained with full contexts) on various forms of reduced context:

- **Original**: uses the complete context.
- **PC** (**P**receding **C**ontext): all tokens after the candidate method are removed, the context only consists of preceding tokens.
- **FC** (**F**ollowing **C**ontext): all tokens before the candidate method call (more precisely, before the bound object) are removed.
- **PC+ParaC** (**P**receding **C**ontext plus **Para**meter **C**ontext): Most tokens after the candidate method token are removed. Only eight complete words or symbols following the candidate method are kept, which can contain up to four parameters.
- **FAM** (**F**ew words Around **M**ethod): Only eight complete word or symbols preceding and following the method call are kept.
- **MAM** (**M**ore words **A**round **M**ethod): Only 40 complete word or symbols preceding and following the method call are kept.

	Original	**PC**	**FC**	**PC+ParaC**	**FAM**	**MAM**
hits@1	**92.3**	70.5	73.2	86.0	61.9	77.7
hits@3	**98.0**	86.1	85.5	94.3	73.8	87.3
hits@5	**98.9**	90.5	88.8	96.1	77.0	89.8
MRR	**95.2**	79.0	80.0	90.0	69.0	83.0

Tab. 3: Comparing JavaBERT's ranking accuracy with different context windows.

These experiments are based on the same test set as before. Accordingly, since the location of the target method call in a code block is chosen randomly, the amount of text for different context forms varies accordingly. Table 3 shows the results of this experiment. As expected, using the full context performs best. The follow-up run is PC+ParaC, indicating that the parameters of a method call form an important source of information for the ranking model. Comparing the results from FAM to PC, FC and MAM and from original to MAM showcases the influence of input size on the method ranking. An observation on the three best runs (Original, PC+ParaC and MAM) is that those are a combination of preceding and following content and are descendingly ordered by the size of their input. The results from MAM show, however, that combining the preceding and following content has a larger influence on the method ranking than the input size when compared to PC and FC. In conclusion, using surrounding content rather than only the preceding content like left-to-right models does have an impact on the ranking of candidate methods.

5 Conclusions

In this paper, we have shown that transformer networks pretrained by masked language modeling are a promising approach towards automated source code understanding, as illustrated for the task of method ranking. Particularly, we have demonstrated the benefits of supervised fine-tuning and studied different context windows, whereas surprisingly small context windows combining a bit of preceding and following code suffices for an accurate inference. Future work will focus on enhancing JavaBERT (which is a token-only model) with syntax trees to obtain richer code representations.

Bibliography

[ALY18] Alon, Uri; Levy, Omer; Yahav, Eran: code2seq: Generating Sequences from Structured Representations of Code. CoRR, abs/1808.01400, 2018.

[AS13] Allamanis, Miltiadis; Sutton, Charles: Mining Source Code Repositories at Massive Scale using Language Modeling. In: Proc. MSR. pp. 207–216, 2013.

[BCB14] Bahdanau, Dzmitry; Cho, Kyunghyun; Bengio, Yoshua: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.

[Cl20] Clark, Kevin; Luong, Minh-Thang; Le, Quoc V.; Manning, Christopher D.: ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In: Proc. ICLR. 2020.

[Da16] Dam, Hoa Khanh; Tran, Truyen; Grundy, John; Ghose, Aditya: , DeepSoft: A Vision for a Deep Model of Software, 2016.

[De18] Devlin, Jacob; Chang, Ming-Wei; Lee, Kenton; Toutanova, Kristina: , BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018.

[Fr15] Franks, C.; Tu, Z.; Devanbu, P.; Hellendoorn, V.: CACHECA: A Cache Language Model Based Code Suggestion Tool. In: Proc. ICSE. pp. 705–708, 2015.

[HD17] Hellendoorn, Vincent J.; Devanbu, Premkumar: Are Deep Neural Networks the Best Choice for Modeling Source Code? In: Proc. ESEC/FSE 2017. ACM, New York, NY, USA, pp. 763–773, 2017.

[HG16] Hendrycks, Dan; Gimpel, Kevin: Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415, 2016.

[Hi12] Hindle, Abram; Barr, Earl T; Su, Zhendong; Gabel, Mark; Devanbu, Premkumar: On the Naturalness of Software. In: 2012 34th ICSE. IEEE, 2012.

[Hu19] Husain, Hamel; Wu, Ho-Hsiang; Gazit, Tiferet; Allamanis, Miltiadis; Brockschmidt, Marc: , CodeSearchNet Challenge: Evaluating the State of Semantic Code Search, 2019.

[Ka20] Karampatsis, Rafael-Michael; Babii, Hlib; Robbes, Romain; Sutton, Charles; Janes, Andrea: , Big Code != Big Vocabulary: Open-Vocabulary Models for Source Code, 2020.

[KB14] Kingma, Diederik P; Ba, Jimmy: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[Ki20] Kim, Seohyun; Zhao, Jinman; Tian, Yuchi; Chandra, Satish: , Code Prediction by Feeding Trees to Transformers, 2020.

[KS19] Karampatsis, Rafael-Michael; Sutton, Charles A.: Maybe Deep Neural Networks are the Best Choice for Modeling Source Code. CoRR, abs/1903.05734, 2019.

[La20] Lan, Zhenzhong; Chen, Mingda; Goodman, Sebastian; Gimpel, Kevin; Sharma, Piyush; Soricut, Radu: ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In: Proc. ICLR. 2020.

[Li17] Li, Jian; Wang, Yue; King, Irwin; Lyu, Michael R.: Code Completion with Neural Attention and Pointer Networks. CoRR, abs/1711.09573, 2017.

[Li19] Liu, Yinhan; Ott, Myle; Goyal, Naman; Du, Jingfei; Joshi, Mandar; Chen, Danqi; Levy, Omer; Lewis, Mike; Zettlemoyer, Luke; Stoyanov, Veselin: RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692, 2019.

[NN15] Nguyen, A. T.; Nguyen, T. N.: Graph-Based Statistical Language Model for Code. In: Proc. ICSE. pp. 858–868, 2015.

[Ot19] Ott, Myle; Edunov, Sergey; Baevski, Alexei; Fan, Angela; Gross, Sam; Ng, Nathan; Grangier, David; Auli, Michael: fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In: Proc. NAACL-HLT. 2019.

[Ra16] Raychev, Veselin: Learning from large codebases. PhD thesis, ETH Zurich, 2016.

[Ra18] Radford, Alec; Narasimhan, Karthik; Salimans, Tim; Sutskever, Ilya: , Improving language understanding by generative pre-training, 2018.

[Ra19] Radford, Alec; Wu, Jeffrey; Child, Rewon; Luan, David; Amodei, Dario; Sutskever, Ilya: Language models are unsupervised multitask learners. volume 1, 2019.

[SHB16] Sennrich, Rico; Haddow, Barry; Birch, Alexandra: Neural Machine Translation of Rare Words with Subword Units. In: Proc. ACL). Berlin, Germany, August 2016.

[TSD14] Tu, Zhaopeng; Su, Zhendong; Devanbu, Premkumar: On the localness of software. In: Proc. 22nd ACM SIGSOFT FSE. ACM, pp. 269–280, 2014.

[Va17] Vaswani, Ashish; Shazeer, Noam; Parmar, Niki; Uszkoreit, Jakob; Jones, Llion; Gomez, Aidan N; Kaiser, Ł ukasz; Polosukhin, Illia: Attention is All you Need. In: NIPS 2017, pp. 5998–6008. 2017.

[Wh15] White, Martin; Vendome, Christopher; Linares-Vásquez, Mario; Poshyvanyk, Denys: Toward Deep Learning Software Repositories. In: Proc. MSR. IEEE Press, Piscataway, NJ, USA, pp. 334–345, 2015.

[Zh19] Zhao, Rui; Bieber, David; Swersky, Kevin; Tarlow, Daniel: , Neural Networks for Modeling Source Code Edits, 2019.

A Decade of Energy Awareness Technology Evolution for Sensor Nodes

Marcus Thoss

RheinMain University of Applied Sciences, Wiesbaden, Germany
marcus.thoss@hs-rm.de

Abstract. Energy awareness is an important aspect of the design of sensor node hard- and software, particularly for battery-powered or energy-harvesting node architectures. Architectural design choices of such systems must regard a multitude of aspects, including size, weight, memory and processing power. Therefore, energy-related design aspects have in recent years become a feature that is being honoured throughout sensor node design. As a result, various technological solutions and strategies have evolved to facilitate energy awareness and energy management aspects.
This paper looks back at the evolution of sensor node technology during the 2010s. Out of ongoing research interests, the author has been monitoring the state of the art of research and industrial solutions aiming at improving the support of energy awareness. Advances were observed for various aspects and levels of relevant technological facets, including electronic measurement and control circuitry, harvesting facilities, power-saving mechanisms at both hard- and software level, energy management strategies and algorithms, networking aspects, and advances and extensions related to operating systems for sensor nodes. A conclusion of these observations, given in this paper, identifies technological increments, leaps and sidesteps that have occurred along the way.
For all aspects described, typical and relevant examples of actual sensor node designs, realised by the author and others, are given. For the overall time span of the decade observed, a short qualitative and quantitative analysis of the technological advances achieved is presented. The paper concludes with an outlook of further evolution of advances in energy awareness technology for sensor nodes to be expected in the near future and to be desired in the long run.

Keywords: Energy Awareness · Sensor Node · Technology Review

1 Motivation

The Internet of Things (IoT) is being rolled out with increasing speed and in a multitude of application areas with competing and complementing communication technologies like 5G networks, massive satellite fleets, LoRA, WPANs and, ultimately, the classical Internet connections. Yet, current reports like Microsoft's "IoT Signals" [13] clearly show that there is not only a strong willingness

to adopt IoT technologies and many platforms to choose from, but also a huge percentage of projects (30% according to the report) failing to reach deployment.

It can be assumed that IoT scenarios relying on widely distributed small-scale self-powered nodes, with or without energy harvesting, are rather common. Engineering of this type of device requires a focusing on the topics of size and weight, node lifetime, connectivity and maintenance. All of those benefit from a better handling of the energy resources available, sometimes in an indirect manner, as when connectivity is being improved by designing a system around the goal to offer the power to transmit just when an application requires it.

When a paper [16] was published by the author 10 years ago, the IoT was still in its infancy. Looking back at this point in time, the decade that has passed since then is referred to in this paper as "the decade". A precursor of the IoT had been the previous 10-year long increase in RFID technology usage during the 2000s. Subsequently, especially in Europe, numerous research programmes for then-topical Ambient Assisted Living were funded, multi-national research cooperations like the European Research Cluster on the Internet of Things [9] began to assemble and the first platform solutions like UniversAAL [3] emerged.

From then on, interest in sensor network technology, both among industry and researchers, boomed, and technological advances, some of which will be described here, helped to increase reliability, applicability to more domains, and efficiency of the solutions. Besides connectivity improvements, both at sensor network and internet-based cloud levels, increasing regard of energy awareness issues were a major factor of the overall development observed.

2 Technological Facets of Energy Awareness

Design measures and system components contributing (or hindering) energy awareness can be found throughout any systems architecture. Therefore, challenges and the potential of different technological facets should first be looked at separately to be considered for an overall, system-wide design strategy for energy awareness.

2.1 Energy Sources

Energy sources can be rated by many factors, the most relevant here being the voltage range and maximum power output. They can be further classified into storage (primary and secondary cells, capacitors and solid state storage) and on-line sources (wired or harvesting sources).

The necessity to rate energy consumed by sensor nodes based on a cost factor the decreases from primary cells to harvesting solutions had been observed at least by the beginning of the decade [16]. With regard to cost as a general concept, rechargeable storage components are seen as a neutral element, energy harvesting supplies can be tagged with a no-cost factor, wired power sources add a moderate, constant cost, and primary cells would be considered the most

expensive form of energy source both in terms of supply logistics and environmental impact.

Consequently, the overall development over the decade moved from primary cells to swappable secondary cells to energy harvesting approaches. Sadly, many industry solutions are still deliberately designed to use primary cells because their lifetime often exceeds that of the application, and cost per cell is, at least monetarily, cheap.

As to the technology of harvesting sources, the most interesting development happened and is to be further expected among solar cell technology. With some cell types now nearing an efficiency of 30%, while 25% had been a good rating by the beginning of the decade (according to [4], the report has been updated regularly since), size of the cells for a given node uptime could be decreased and harvesting nodes can now be utilised in a much broader range of environments.

2.2 Energy Storage

Storage-type energy sources introduce additional parameters like capacity, possibly recharge characteristics, wear-out, size, weight and operating temperature range. Regarding purely economic motivations, secondary cells of the Li-Ion and related types have become favoured over the decade, as small form factors like coin cells were available and the technology had advanced to allow energy harvesting to provide enough recurring energy input for many applications. The other trend that did not gain as much momentum is the use of supercapacitors as rechargeable storage element. From an environmental perspective, the use of carbon-electrode supercapacitors would be highly preferable compared to the more problematic secondary cell technologies often employing scarce or toxic substances. The problems of still minor capacities and more challenging output voltage curves, although partially outweighed by a high number of possible recharge cycles, leaves this technology so far a candidate for future advances.

2.3 Power Controllers

The evolution of microcontroller technology has led to single-supply voltage solutions, as modern controllers are capable of generating diverse voltages required internally by themselves. Although this had been the case by the beginning of the decade already, input voltage ranges now become broader and single-supply architectures are even more common.

On the supply side, power management becomes more complex when storage and harvesting strategies are included. Here, technological evolution has produced an increase in efficiency of conversion and the availability of highly integrated power controllers. These can manage multiple harvesting and wired sources, intermediate energy storage, output voltage and power control, and require few external components. Thus, the complexity of node hardware designs could be reduced dramatically, while efficiency was increased.

2.4 Microcontrollers

In the case of a sensor node, the processor is almost certain to be a microcontroller. A consolidation of instruction set architectures could be observed among semiconductor manufacturers from a multitude of proprietary designs towards classical Intel 8051 Cores for 8-bit processors and ARM architectures covering 32-bit. Notable exceptions are Microchip (including former Atmel), remaining successful with their 8-bit PIC and AVR architectures, and Texas Instruments with their 16-bit MSP430 processor family.

Microcontrollers aid energy aware designs by the fact that power domains of processor core and peripherals are closely coupled and compatible, reducing transfer and conversion losses. A notable improvement has been introduced by controllable power gating of embedded peripherals and an increase of detail in sleep state modelling, both regarding the number of levels and the granularity of control that can be exerted.

2.5 Memory

The choice of RAM type for low-energy designs is of course static RAM offering negligible idle-state power losses. Thus, driven by the need for more complex sensor node firmware, RAM size offers have been increased from a few KB to some 100 KB without greater penalty, as relevant power drains occur only during RAM access, which is a matter of limiting the amount of code execution.

Not yet frequently available in common architectures are the follow-up non-volatile, near-RAM-access-speed technologies FRAM and MRAM, with FRAM being available in some MSP430 variants as the only mainstream platform. A major disadvantage of SRAM as main memory is the loss of information in power-off and deep sleep states. Since zero-power sleeping is a desirable state, though, energy aware designs must currently apply measures to persist or re-initialise RAM contents during these states. Should non-volatile memory technologies become the standard for main memory designs, powerless sleeping would become available with less overhead and complexity. In [14], energy consumption measurements for various sleep states and application aspects were analysed.

2.6 Energy Measurement

Energy measurements can basically be taken with an external meter or, the node can measure its own energy flows. In any case, voltage and current must be determined to calculate a momentary $P = U \cdot I$ and then integrate it over a time span to get $E = \int P dt$. For external measurements, the problem of mapping samples taken to the network node's time domain poses a challenge. So, although external measurements can often afford more precise metering equipment, internal measurements are desirable, i.e. the note would measure itself. Thus, initially, analog input pins of the microcontroller were, and still are, used to measure battery and operating voltage, and a shunt resistor is inserted in the supply path to determine the current by measuring the voltage drop created.

Over the decade, energy aware node designs emerged that were equipped with dedicated on-board power metering solutions, improving accuracy and sometimes allowing the microcontroller to sleep while measurements continue. Yet, such additional circuitry would consume power itself, degrading node uptime and falsifying measurement values.

There is a kind of "holy grail" of measurement solutions in this respect that is starting to be realised more frequently. It is the monitoring of the embedded switched-mode power supply (SMPS) circuits that manage the supply voltages of the node, as noted in section 2.3. An SMPS design regulates the supply according to the demand by transferring small chunks of energy from a supplier to the consumer. Thus, the amount of energy transferred is proportional to the number of chunks converted. The trick applied here is to allow for the switching event to be counted by the microcontroller itself by adapting the pulses through appropriate decoupling and probing circuitry.

Such solutions promise to keep the measurement overhead, both in terms of node circuit complexity and energy investment, to a minimum. As most modern microcontrollers are capable of counting events even in deep sleep modes, even the requirement of continuous monitoring can be fulfilled.

2.7 Operating Systems

There has been a notable evolution of embedded operating systems (OS) meeting sensor network technology requirements. Foremost, strong connectivity support is obviously a necessity. Secondly, to be applicable to a broad range of projects, multi-platform support, also scaling from low-end architectures for small sensor nodes to those found in rather well-equipped routing and gateway nodes, would contribute to the popularity of a sensor network OS.

Energy awareness did not seem to be of primary concern by the beginning of the decade, and has actually begun to be fully integrated into the OS architectures and APIs only recently. To some extent, the open source platforms Contiki OS [2] and Tiny OS [11] pioneered the sensor node research community, and Contiki considered on-line, OS support for energy measurements early on, providing the important feature of attributing measurement data with the execution context of the measurement. During the decade, the open source embedded OS scene saw Mbed OS (2009), Apache MyNewt (2015), Linux Foundation's Zephyr (2016), and only very recently, RIOT (2018) appear. Mbed is being driven by the Arm company, targeting their own architectures, while Texas Instruments has provided their own closed-source solution TI RTOS for their hardware platforms, notably the MSP430 family. All of them show activities towards providing better support for energy awareness, with RIOT having a strong background of research interest driving its development, and its energy awareness aspects [6].

2.8 Testbeds

By the middle of the decade, the survey published in [8] showed, that a large number of approaches had accumulated, and researchers and industry had been

busy investigating sensor networks through testbeds. The authors of the survey themselves had been part of the community from around 2010 on, and their SANDbed setup [7] explicitly focused on energy monitoring. Testbeds continue to evolve regularly within the communities developing platform solutions, a recent, notable example being the RIOT testbed [5].

Basically, there are three methods for assessing energy budgets, that is, model-based prediction (optionally based on previous off-line measurements), and internal and external on-line measurement. With an approach reasonably limiting the overhead of the measurement action, on-line measurements of the actual operation of the sensor network would be preferable. It requires a measurement infrastructure, though, that is able to retrieve and collect the measurement data and offers sufficiently comfortable testbed setup, experiment control and evaluation support. These requirements are likely to be the reason why many solutions still rely on off-line measurements and simulated or modelled energy assessments.

3 Example Designs

3.1 Harvesting Iris Mote

An early, rather simplistic energy aware harvesting sensor node solution was presented and analysed in 2011 in [15], providing all basic ingredients based on the Iris mote platform from MEMSIC. The design and the analysis paper anticipate many problems and approaches that have been discussed in the community in the following years. The researchers added a solar cell, supercapacitor and power management to the base platform, considered energy measurement by counting switched mode power supply events and looked at the pros and cons of simulated off-line versus measurement-based on-line assessment strategies.

3.2 WieDAS

In this research project, the sensor node box and its solar panel measure approximately 8 cm by 12 cm each, which is voluminous by today's standards. The large size was caused by the use of two AAA Batteries and a custom single-layer carrier board for prototype production. Still these nodes were successfully operated in an AAL demonstrator scenario described in [10]. Energy awareness was initially limited to battery voltage measurement using an analog input pin of the main microcontroller and a gap in the supply line to attach a current probe. A major step towards power awareness in this case was the attachment of a TI INA219 current sensor with autonomous measurement cycles and an I^2C control port. This device is capable of determining power consumption measuring both voltage current and even averaging power values over programmable window sizes. Thus, with constant sampling frequency f and $N_{samples}$ per window, energy transfers can can be approximated by $\Delta E = P \cdot \Delta t$ with $\Delta t = N_{samples} \cdot \frac{1}{f}$.

Adding this external measurement component allowed, in principle, leaving the microcontroller in deep sleep modes without stopping the measurements,

if the CPU was woken up in time to collect buffered measurement data. One of the drawbacks of the design was the late addition of solar energy harvesting, which did not harmonise well with the inflexible power domain scheme consisting of jumpers that could be removed manually to disconnect single sensors of the board from the power supply.

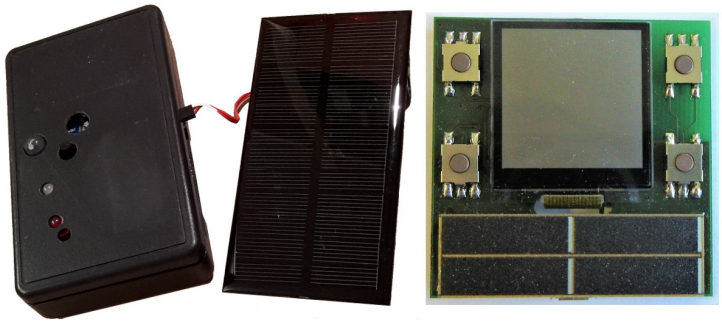


Fig. 1. WieDAS (left) and BASE MOVE (right) nodes compared

3.3 BASE MoVE

By the end of the decade, the BASE MoVE project [1] had created custom sensor nodes, too, which measured less than half the size of a WieDAS node, but included comparatively tiny solar cells and a large display (shown in Figure 1, nodes are not to scale). The project created node hardware with programmable power gating of sensor peripherals, energy harvesting realisations for solar and thermal sources, and an automated testbed integrated with build automation for regression tests. Automated distributed energy measurements and energy management, that had been planned, finally exceeded the scope of the project.

4 Conclusion and Outlook

Within the context of the author's work, the project LONG MOVE [12], a successor to BASE MoVE, focuses on node-local and distributed energy management topics for sensor networks. Its mission statement calls for "energy awareness by design", taking this aspect into account from the beginning on. It thus reflects an insight gained from previous designs that could have benefited from earlier, consistent involvement of energy-related questions. With OS and hardware platforms offering increasingly better support in this area, it can be expected that this will become an overall trend. In fact, energy awareness will be an integral part of designs when self-adaptation features embedded at platform level can obliterate to some degree the need to regard energy within the design of the application proper.

References

1. Akelbein, J.P., Beckmann, K., Hoss, M., Schneider, S., Seyfarth, S., Thoss, M.: BASE MoVE - A Basis for a Future-proof IoT Sensor. In: INFORMATIK2020. Gesellschaft für Informatik, Bonn (2020), to be published
2. Dunkels, A., Gronvall, B., Voigt, T.: Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors. In: 29th Annual IEEE International Conference on Local Computer Networks. pp. 455–462 (2004)
3. Ferro, E., Girolami, M., Salvi, D., Mayer, C., Gorman, J., Grguric, A., Ram, R., Sadat, R., Giannoutakis, K., Stocklw, C.: The UniversAAL Platform for AAL (Ambient Assisted Living). Journal of Intelligent Systems (01 2015)
4. Green, M.A., Emery, K., Hishikawa, Y., Warta, W.: Solar Cell Efficiency Tables (version 35). Progress in Photovoltaics: Research and Applications **18**(2), 144–150 (2010)
5. Günes, M., Engelhardt, F., Nothnagel, K.: Technical report - Designing a Testbed for Wireless Communication Research on Embedded Devices. In: 18. GI/ITG KuVS FachGesprch SensorNetze, FGSN 2019. pp. 41–44 (2019)
6. Hahm, O.: Enabling Energy Efficient Smart Object Networking at Internet-Scale : Experimental Tools, Software Platform, and Information-Centric Networking Protocols. Ph.D. thesis (12 2016)
7. Hergenröder, A., Wilke, J., Meier, D.: Distributed Energy Measurements in WSN Testbeds with a Sensor Node Management Device (SNMD). In: Workshop Proceedings of the 23th International Conference on Architecture of Computing Systems. pp. 341–438. VDE Verlag (Feb 2010)
8. Horneber, J., Hergenröder, A.: A Survey on Testbeds and Experimentation Environments for Wireless Sensor Networks . IEEE Communications Surveys & Tutorials **16**(4), 1820–1838 (Apr 2014)
9. http://www.internet-of-things-research.eu
10. Kröger, R., Lux, W., Schaarschmidt, U., Schäfer, J., Thoss, M., von Fragstein, O.: The WieDAS AAL Platform: Architecture and Evaluation. In: Wohnen - Pflege - Teilhabe. 7. Deutscher AAL-Kongress mit Ausstellung, 21.-22. Januar 2014, Berlin. VDE Verlag GmbH, Berlin/Offenbach (January 2014)
11. Levis, P., Madden, S., Polastre, J., Szewczyk, R., Whitehouse, K., Woo, A., Gay, D., Hill, J., Welsh, M., Brewer, E., Culler, D.: TinyOS: An Operating System for Sensor Networks, pp. 115–148 (01 2005). https://doi.org/10.1007/3-540-27139-2_7
12. https://www.hs-rm.de/de/fachbereiche/design-informatik-medien/forschungsprofil/long-move
13. IoT Signals. Tech. rep., Microsoft Corporation (2019), https://azure.microsoft.com/mediahandler/files/resourcefiles/iot-signals/IoT-Signals-Microsoft-072019.pdf
14. Reinhard, J.: Energiebetrachtungen für Sensorknoten mit FRAM (Energy analysis of FRAM-equipped sensor nodes), https://wwwvs.cs.hs-rm.de/vs-wiki/index.php/EM2018WSP04, lab report, Embedded Systems Course 2018/19, RheinMain Univerity of Applied Sciences
15. Renner, C., Meier, F., Turau, V.: Holistic online energy assessment: Feasibility and practical application. In: Ninth International Conference on Networked Sensing, INSS 2012, Antwerp, Belgium, June 11-14, 2012. pp. 1–8. IEEE (2012)
16. Thoss, M.: Supporting energy awareness in distributed embedded systems. In: Dörner, R., Krömker, D. (eds.) Self Integrating Systems for Better Living Environments: First Workshop, Sensyble 2010. pp. 117–124. No. 1, Shaker Aachen (November 2010)

BASE MoVE - A Basis for a Future-proof IoT Sensor

Jens-Peter Akelbein[1], Kai Beckmann[2], Mario Hoss[1], Samuel Schneider[3], Stefan Seyfarth[3], and Marcus Thoss[2]

[1] Darmstadt University of Applied Sciences, Schoefferstrae 8b, 64295 Darmstadt
{firstname.lastname}@h-da.de
[2] RheinMain University of Applied Sciences, Unter den Eichen 5, 65195 Wiesbaden
{firstname.lastname}@hs-rm.de
[3] Thermokon Sensortechnik GmbH, Platanenweg 1, 35756 Mittenaar-Offenbach
{firstname.lastname}@thermokon.de

Abstract. For a long time, the Internet of Things was considered the vision of interconnecting every device, leading to fundamentally new and pervasive application scenarios. In practice, however, the projected growth and realisation of IoT scenarios is often impeded by practical problems. The BASE MoVE research project, a cooperation between universities and industrial partners, took a holistic look at requirements and inhibitors for investing in IoT solutions, using Ambient Assisted Living as an application domain example. The perspectives of all stakeholders involved were taken into account during the design of a solution architecture, from the user to the manufacturer to the service provider and housing association. This paper presents the resulting modular base platform for IoT applications. Power supply through battery and energy harvesting enables low installation costs, since no expensive installation of cable ducts in existing buildings is required, which reduces the initial investment. The use of open source software and the support of several common smart home protocols also prevents a lock-in effect and dependency on a single manufacturer. This makes it possible to protect investments from market-driven changes from one manufacturer's ecosystem to another. Here the paper takes an in-depth look into the choice of protocols. Over-the-air updates allow for secure operation as well as remote maintenance, no longer requiring expensive in-person maintenance. Finally, the manufacturing of the solution as a hardware module, as realised in BASE MoVE, also allows for easier creation and certification of new sensor devices in a company's product portfolio. To evaluate the developed solution, an apartment was equipped with different sensor devices and a smart home scenario was implemented. The feasibility study could demonstrate that it is indeed possible to create a base platform that meets the requirements of the stakeholders involved. In addition to the scientific results, the project gives an assessment about component maturity and cost, which is valuable for the commercial project partner and its market entry strategy.

Keywords: Internet of Things · Ambient Assisted Living · Smart Home · Home Automation

1 Motivation

The vision of an Internet of Things (IoT) describes a technological revolution to be created by operating interconnected devices that pervade environments of personal life, work, and nature. Although distribution of such devices has been realised to a notable degree already, and solutions for many application areas, commercial and research platforms abound, there are numerous problems hindering further growth and flawless operation. Foremost, sustainability is still questionable, as the market has not yet stabilised, and new protocols and interfaces are still emerging. This, and the existence of so many variants calls for adaptability as a major feature of a design if it is meant to prevail. Without it, the growth of the IoT landscape is bound to decrease or even cease.

It is therefore necessary to identify the relevant impeding factors that could endanger the future of the IoT. Functional correctness or applicability to the solution domain is rarely a problem. Instead, non-functional and platform-level factors must be considered critical for the success of a market solution. As, with increasing distribution of sensor nodes, most of these cannot use wired powering any more, and regular battery changes become impractical and too costly, self-sustained energy harvesting must become a mainstream technology. To further support sustainability, and the success of a platform solution, technological changes must be reacted upon by providing easy integration into future scenarios, should the protocol landscape change. This requires a high degree of adaptability and the possibility to update the firmware, and thus, possibly, support new protocols.

The project idea behind the solution presented here was to not try and reinvent base technologies once again, as many viable solutions are available today. Neither was the intention to focus on offering a consistent application-level protocol and modelling solution, which had been investigated thoroughly in previous research projects. Instead, this project concentrated on integrating existing hardware, (embedded) operating systems (OS) and communications technologies as a flexible IoT platform.

The acronym contained in the project name BASE MoVE states the main aspects regarded in the project. A dedicated sensor node hardware platform should be created as the **b**asis for the solution. **A**daptability is to be achieved by allowing over-the-air (OTA) firmware updates. **S**ecurity should be considered as a first-class design objective and thus be regarded from the very start, and finally, design for **e**nergy awareness at hard- and software level must lead to a viable self-powered sensor node architecture utilising energy harvesting technology. Application scenarios regarded in the project were meant to serve as technology test show cases for the validation of the fulfilment of these goals, application level modelling was not considered a prime objective.

2 Related Work

Enabling real multi-protocol support in IoT-scenarios depends on the capabilities of the underlying hardware. If the transceivers are locked to specific radio

protocols, the freedom to change or replace a protocol is limited. Most flexibility can be achieved if the radio transceivers are generic and the implementation of a specific protocol is a matter of hardware configuration and software.

There are hardware platforms like the EFR32 series from Silicon Labs [4] providing System-on-a-Chip (SoC) solutions embedding generic radio transceivers which allow for the usage of different protocols with the same device. Moreover, a subset of the EFR32 series embeds transceivers for different radio frequencies, like 2.4GHz and the sub-GHz band (915 resp. 868MHz or 433MHz). Silicon Labs provides several, proprietary Smart-Home protocol stacks for their SoC family, like ZigBee, Thread, BLE, Z-Wave etc., and supports the creation of firmware running two stacks in parallel, like BLE and Zigbee [3].

Most hardware vendors provide proprietary protocol stack implementations for the application areas they are targeting. For the broader area of the IoT this can only cover a subset. There are many surveys like [13], [15] or [8] gathering and categorising the protocols proposed and used, and solutions in the IoT or the smart home sector. Regarding the term IoT, there is a consolidating trend towards IP-based protocols noticeable in recent surveys. The emergence of IP for IoT protocols becomes apparent, considering the Zigbee Cluster Library having been made available for IP [1], or the work towards IP over BLE [11]. This is a significant change from the former situation, with manufacturers selling products combining hardware and software. Open vendor-independent protocol stack implementations for the IoT are provided by open source IoT operating systems, like RIOT OS [7], mbed, zephyr, contiki or others [12].

3 Architecture

Figure 1 shows the architecture for the IoT platform developed. It consists of the modular hardware platform and the software layer composed of an IoT OS, exchangeable protocol stacks, management and firmware update functionality and the top-level applications. The aim is to provide a flexible platform for fast and cost-efficient development of smart home and related IoT devices, supporting operation based on energy harvesting. The IoT protocols are kept exchangeable within the base platform to prevent lock-in effects for the OEM and customers. Furthermore, the usage of an open source OS is proposed to protect sold and installed smart home devices in households from becoming abandon-ware.

One important requirement for the *hardware* is it featuring low power modes that allow for energy harvesting powered devices. The transceiver should support arbitrary protocols, like BLE and 802.15.4. It should also be future-proof and be able to support future protocols by using a generic transceiver. The selected Mighty Gecko EFR32MG13P733F512GM48 from Silicon Labs meets these requirements. It is a 40 MHz ARM Cortex-M4 with 512 KB flash and 64 KB RAM. It was manufactured as a PCB module to simplify certification and re-usability.

The devices are developed for indoor use and enable the measurement of the load capacity. It also features an energy harvesting subsystem based on the BQ25570. A monocrystalline solar cell (SLMD121H04) powers a 60 mAh

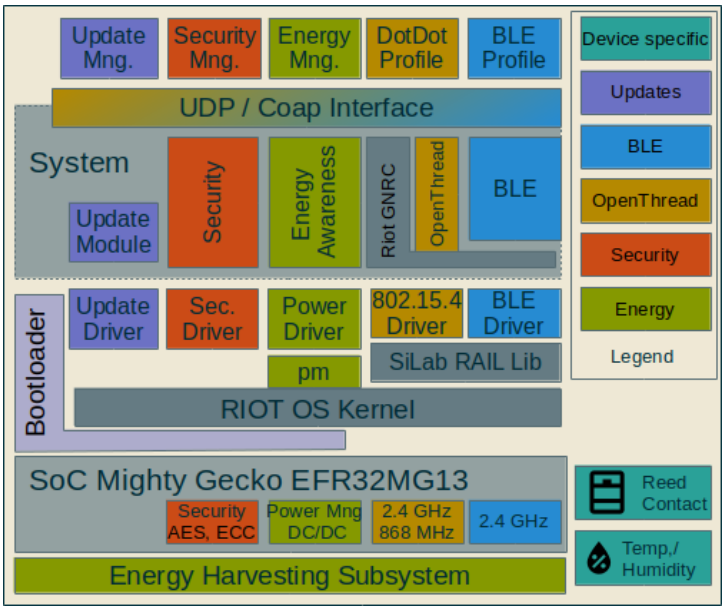


Fig. 1. Architecture

LiPo battery. Several iterative versions of three different devices were created. 1) A relay actuator with a permanent power supply. 2) A window contact with a temperature and humidity sensor (SHT35), a pressure and air quality sensor (BME680), a reed contact and a debug and load interface (FTDI230x). 3) A room control unit with the same sensors as 2) and an additional low power display (LS013B7DH03) and buttons for user input and feedback. 4) An occupancy sensor, which is a modified version of 2), replacing the display with a PIR sensor. The types 2) to 4) were passive (end devices), reacting on external or timer events and capable to be powered by energy harvesting.

The OS as the *software* foundation was chosen based on a requirements analysis. Important aspects were the support of an open compiler like GCC, debugging support and it being an active project. Collaboration with the developer community and the ability to extend the OS with new features had to be possible. Satisfying these requirements, RIOT OS was chosen, even if it did not yet support all of the required hardware and protocol stacks. This was possible as the authors were familiar with RIOT OS, and estimated the development and integration of the missing functionality to be possible within the project.

The modular base platform supports exchangeable IoT protocols by flashing the device with different application images. The efforts to support multiple protocols within the same application, like Silicon Labs allows with their proprietary protocol stacks, were analysed and deferred due to the complexity and necessary modification of the OS. Furthermore, replacing the protocols through

firmware update reduces the resource requirements to keep the devices viable, but requires firmware over the air (FOTA) functionality and the management of devices. To this aim, an additional management protocol was added, enabling the use of unaltered IoT protocol stacks. It is foreseeable that this approach will become obsolete, once a standard for FOTA is adopted by IoT protocols. On a device level an OS bootloader is used for a standard two slot approach.

The IoT platform supports wireless devices without a permanent energy source to reduce installation and maintenance costs. This comes at the cost of additional cross cutting concerns, limiting the possible functionality. In addition, the limited resources of the hardware platform and the required security level for OTA updates have to be considered throughout the system. To be a viable product, security requirements have to be met, while still allowing the operation with limited energy and memory requirements. Therefore, the platform requires both Security- and Energy-Awareness by Design.

For the management protocol, there was no established solution for constrained devices yet. Lightweight Machine to Machine (LWM2M) was evaluated[14], but could not be used due to resource requirements. Instead of a dedicated management protocol with the associated overhead, an update functionality is triggered over standard CoAP. A similar approach was also followed at SUIT[5], the IETF Taskforce which started work shorty before the beginning of the BASE MoVE project. The transmission of the image with parsing information in the form of a SUIT manifest should also enable protocol-independent transmission.

3.1 Multiprotocol

In the smart home sector, no consolidation of protocols is noticeable at this moment. On the contrary, new protocols are proposed, like the relatively new protocol Thread developed by Google [6]. To be able to demonstrate the multiprotocol approach of the BASE MoVE platform, a qualified subset of protocols had to be selected. This selection is differentiated between transport protocols, which distribute data between nodes (summarised OSI layers 1 - 6) and application protocols, defining the structure and semantics of the data exchanged (OSI layer 7). The requirements for this selection are: they have to be usable in smart home scenarios, support low power operations (potential utilisable with energy harvesting) and state-of-the-art security features. Furthermore, there have to be open source protocol stacks available, which can be used on the selected Mighty Gecko SoC hardware platform. For the application protocols, there should be standardised device profiles available enabling interoperability and connectivity between smart home devices.

The first selected group of protocols satisfying these requirements are IP-based. The 6LoWPAN protocol is the IETF standard to run IPv6 over Low - Power Wireless Personal Area Networks [10]. It is set on top of the IEEE 802.15.4 protocol, like ZigBee, and allows transparent mapping to standard IPv6 networks. There are several protocol stacks available, provided by different IoT OS. Using RIOT OS, its native 6LoWPAN stack is used. Additionally, the Thread

protocol is part of this group. Originally developed by Google Nest, it is now supported by different software and hardware vendors [6]. Thread is based on 6LoWPAN, but replaces some parts and adds functionality, especially regarding security, deployment and routing. For this IoT platform, the open source stack OpenThread [2] is used, as there is a port to RIOT OS available. As second, different type of protocol BLE is selected. It is widely used in practice in many different smart home products, and virtually every smartphone offers connectivity and support. Furthermore, BLE Mesh, as a new standard extension, is going to provide the necessary mesh routing for more complex smart home scenarios in the future. For this IoT platform, the open source stack nimBLE from the Apache mynewt project is utilised, which was integrated by the RIOT OS community by the end of the BASE MoVE project. The selected application protocols are applied on top of the IP-based transport protocols. Again, there are several possibilities (see [13]), but in this work, focus was on the Constrained Application Protocol (CoAP), another IETF standard. It was used for the application as well as the management part, because it is relatively lightweight and uses UDP. It can be integrated in edge and regular IT networks and there are several smart home and management protocols utilising CoAP.

For the support of application profiles based on CoAP, two approaches were incorporated. First, probably the most popular approach in practice is to define something new for a particular use case. As a first quick solution, the sensor data provided by the concrete devices were manually mapped to a REST structure and served by the RIOT OS CoAP implementation. The obvious limitation of the approach is that compatibility and interoperability are limited to the CoAP layer. As a second approach, a subset of Dotdot [1] was prototypically implemented, which is a ZigBee Alliance standard mapping the well-established ZigBee Cluster Library (ZCL) and the ZigBee Device Profiles (ZDP) to a CoAP REST interface. Since the official specification of Dotdot was not openly accessible for most of the project time, the proof-of-concept realisation was based on information gathered from presentations, white papers and commercial implementations. It is limited to poll sensor data from end devices over CoAP and 6LoWPAN or the Thread protocol.

Regarding BLE, the application layer is part of the standard itself (Generic Attribute Profile - GATT). The Bluetooth Special Interest Group (SIG) defines the structure and semantics of data as Characteristics, the specific behaviour of a device functionality as Services, which are composed to Profiles defining the functionality of a type of device [9]. Since the BLE stack within RIOT OS was only usable at the very end of the project time, very simple GATT services were implemented for the hardware platform as a proof-of-concept.

4 Application Scenario

To evaluate the base platform, an apartment was retrofitted to test the base-functionality in exemplary scenarios. The apartment was provided by one of the supporting industry partners, Vonovia SE.

The usage of BLE was planned, with IP over BLE for the OTA functionality, but due to delays for BLE support in RIOT, 6LoWPAN was used as a replacement protocol in the apartment instead. The behaviour and logic of the scenarios consist of a set of rules, executed on a smart home service platform. As such, the OpenHAB platform was chosen, necessitating the implementation of protocol bindings for CoAP and the application semantics defined for this project.

For the application scenarios, an exemplary ambient assisted living scenario (S1) as well as a smart home scenario (S2) were realised using the devices presented in chapter 3. For both scenarios, window sensors were attached to windows and doors reporting state changes (open/closed). In S1, a resident is warned when leaving the apartment if the windows are still open. So, if any window and the front door is open, a red warning light is switched on, and the forgotten window is displayed on a map next to the door. In S2, the room temperature is controlled depending on the residents preferences, and the room lights are switched on or off depending on the room being occupied. Residents are using the room control unit to set a desired temperature, and, if the reported temperature from the window contacts surpasses this thresh-hold, an actor turns on a ventilator.

For a practical evaluation, these scenarios were implemented in three different rooms of the apartment. The first two rooms realised the scenarios with Thread and 6LoWPAN, respectively, to demonstrate the functionality. In the third room, the test case for the FOTA functionality was evaluated, which needed two parallel OpenHAB deployments like in the first two rooms, each in a different IoT network and running a placeholder application. The devices were updated with firmware images, supporting the two different protocols, rotating them between the three applications.

5 Conclusion

The evaluation of the application scenarios has shown that the modular platform approach is feasible. The modular platform approach reduced the implementation effort for each new type of smart home device. In the end, the primary effort was on the hardware part. For the software, only drivers for new peripherals parts, some configuration and the additional application functionality were needed, which could largely be generated from ZigBee device profiles. Replacing an IoT protocol with another by firmware update was straightforward. The driver and protocol abstraction of the IoT OS allowed an easy creation of application images with new protocols. The issues encountered often stemmed from immature implementations. The early adaptation of Thread and DotDot, before the specification was available, provided some challenges. The implementation used, OpenThread, still had memory leak and energy consumption issues. While RIOT fulfilled the requirements set for this research project and offered great collaboration with the developer community, usage in a production environment would be premature. However, over the course of the project, clear improvements could be observed. The overall neglect of aspects regarding energy management

throughout the design and the implementation, though, made the application in this scenario difficult. As a lesson learned, an adequate IoT-OS requires a holistic "Energy Awareness by Design".

The evaluation of the application scenario has demonstrated that retrofitting existing living spaces with future-proof sensor devices without costly cable installation is possible. Functional problems in the form of power consumption and stability issues with updates over Thread were caused by the use of immature implementations of OS and protocol stacks. In the apartment setup, an inhibitor for the general use of smart home scenarios was identified in the configuration time and complexity in OpenHAB. It is unlikely that the average end-user, facility manager or electrician installing the devices could implement the apartment-specific rules. Improvements in smart home service platforms are thus still needed for widespread adoption.

References

1. Dotdot, https://zigbeealliance.org/solution/dotdot/
2. OpenThread , https://openthread.io/
3. Silicon Labs - Dynamic Multiprotocol, https://www.silabs.com/wireless/multiprotocol
4. Silicon Labs EFR32 Wireless Gecko Technology Features, https://www.silabs.com/products/wireless/technology
5. Software Updates for Internet of Things (suit), https://datatracker.ietf.org/wg/suit/about/
6. Thread Stack Fundamentals, https://threadgroup.org/ourresources
7. Baccelli, E., Gundogan, C., Hahm, O., Kietzmann, P., Lenders, M.S., Petersen, H., Schleiser, K., Schmidt, T.C., Wahlisch, M.: RIOT: An Open Source Operating System for Low-End Embedded Devices in the IoT. IEEE Internet of Things Journal **5**(6), 4428–4440 (2018)
8. Dizdarević, J., Carpio, F., Jukan, A., Masip-Bruin, X.: A survey of communication protocols for internet of things and related challenges of fog and cloud computing integration. ACM Computing Surveys **51**(6), 1–29 (2019)
9. Gomez, C., Oller, J., Paradells, J.: Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. Sensors (Switzerland) **12**(9), 11734–11753 (2012)
10. Montenegro, G., Kushalnagar, N., Hui, J., Culler, D.: Transmission of ipv6 packets over ieee 802.15.4 networks. RFC 4944 (September 2007)
11. Nieminen, J., Savolainen, T., Isomaki, M., Patil, B., Shelby, Z., Gomez, C.: Ipv6 over bluetooth(r) low energy. RFC 7668 (October 2015)
12. Qutqut, M.H., Al-Sakran, A., Almasalha, F., Hassanein, H.S.: Comprehensive survey of the iot open-source oss. IET Wireless Sensor Systems **8**(6), 323–339 (2018)
13. Salman, T., Jain, R.: A Survey of Protocols and Standards for Internet of Things. Advanced Computing and Communications **1**(1) (mar 2017)
14. Schmelzer, P., Akelbein, J.P.: Evaluation of hardware requirements for device management of constrained nodes based on the lwm2m standard. In: CERC (2019)
15. Zaidan, A.A., Zaidan, B.B., Qahtan, M.Y., Albahri, O.S., Albahri, A.S., Alaa, M., Jumaah, F.M., Talal, M., Tan, K.L., Shir, W.L., Lim, C.K.: A survey on communication components for IoT-based technologies in smart homes. Telecommunication Systems **69**(1) (2018)

Herausgeber et al. (Hrsg.): Name-der-Konferenz,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2017 1

Modelling of Interaction in Interweaving Systems as Ontology Mapping Adaption

Matthias Jurisch[1] Bodo Igler[2]

Abstract: Interweaving Systems (IWS) are systems that have not been designed to cooperate, but can influence each other at runtine through a shared environment. In this paper we present a general approach that enables IWS to gain an explicit view of the shared environment. The approach is based on ontology alignment and model-driven techniques. The core idea is to buid a directory ontology, that shows what systems can know about their environment and local connector ontologies that organize information routing between systems and connect these via ontology alignments. Changes are addressed using ontology mapping adaption. This approach is presented using an application example from the area of traffic control systems. The key benefit of the approach is, that it allows supporting and sustaining data integration in Interweaving Systems using explicit and domain-independent rules.

Keywords: Interweaving Systems; Ontology-Driven Development; Ontology Mapping Adaption

1 Introduction

Interweaving Systems [To16] are systems that have not been designed to cooperate, but can influence each other at runtime by interacting through a shared environment. Interweaving systems usually work under conditions that require some kind of soft or hard real-time constraints. Therefore, they are first and foremost optimized to hold these constraints. Optimization for domain-specific goals are usually regarded as less important. However, deliberate cooperation between interweaving systems can potentially improve these soft aspects without touching the real-time/safety critical part.

Wether and how a shared view on the environments of Interweaving Systems improves cooperation and performance regarding domain-specific goals is still an open question. A first step to address this question consists in designing and evaluating respective approaches for environment sharing. As the typical environments of Interweaving Systems are prone to changes and as parts of such systems can fail, appropriate approaches have to particularly consider these issues, too.

In this work, we present a general approach to support Interweaving Systems in creating a shared view on their environment. It employs knowledge based techniques such as ontologies

[1] Department of Design – Computer Science – Media, RheinMain University of Applied Sciences, Unter den Eichen 5, 65195 Wiesbaden matthias.jurisch@hs-rm.de
[2] Department of Design – Computer Science – Media, RheinMain University of Applied Sciences, Unter den Eichen 5, 65195 Wiesbaden bodo.igler@hs-rm.de

and inference. To allow systems to access this shared view, a modification of these systems is required. What kind of information systems can gather about the environment (e.g., what sensors they can use to observe it) is modeled via a directory ontology. Each system can view this directory ontology and use it to request more information regarding specific aspects of the environment. In this way, systems will only receive data on the environment that is relevant to them. What kind of information is relevant to a system is modeled in a local connector ontology that is connected to the directory ontology using ontology alignments. Changes in the environment and system failures require that this alignment is constantly adapted. This issue is addressed with a technique called Ontology Mapping Adaption [Gr13a]. The key benefit of the approach is, that it allows supporting and sustaining data integration in Interweaving Systems using explicit and mostly domain-independent rules.

In our previous work, we presented a domain-specific technique that addresses this problem for interweaving systems in the domain of autonomous traffic-control [JI17]. The main contribution of this paper is twofold: (1) We present an abstract, i.e. domain-independent approach, that shows how the domain-specific technique can be applied to interweaving systems in general. (2) We show how the case study of [JI17] fits into the application of the abstract approach. This includes an evaluation of how the approach needs to be tailored to the specific use case and which model transformations need to be implemented. We also demonstrate, how autonomous systems can use this data in this domain specific use case and how the shared view on the environment can be useful.

The remainder of this work is structured as follows: Section 2 gives an overview of the background and related work and identifies the research gap. An overview of the approach and the models that are involved and how these models are used to create a shared view is shown in Section 3. Section 4 describes an application and shows how the approach can be tailored to domain-specific problems. A discussion is given in Section 5. The paper ends with a conclusion and an outlook to future work in Section 6

2 Background and Related Work

Interweaving Systems are systems that usually are designed in some kind of technical context that is characterized by several aspects: The systems in this context influence each other, both via *defined interactions* and *not defined interactions*. Also, the systems are heterogeneous in a sense that no central instance controls all of them and they were not designed to take other system's influences into account. The environment of the systems in general is uncertain, but can be partly observed. This has severe consequences for their effectiveness, as determining the outcome of interactions becomes very difficult. Another important aspect is that the application domain usually requires the systems to operate under real-time conditions. [To16]

Interweaving Systems are often supported using so called organic computing approaches [MSSU11] and often focus on making the systems itself more fault-tolerant and more able

to deal with changing environments. Organic computing techniques draw inspiration from biological systems and are often used in the area of self-managing systems. An example from this field of research is the artificial hormone system, where tasks broadcast messages, so-called hormones, to coordinate what kinds of tasks need to be run [PB12]. Hence, the artificial hormone system communicates messages that contain which tasks need to be executed. In this work, we approach the area of interweaving systems from a different perspective: We focus on allowing Interweaving Systems to communicate about their environment and making this environment explicit. The messages exchanged in this model contain information on what data is available. This communication is supported by formal conceptual models (ontologies). This leads to several areas of related work, including (1) Model Driven Software Engineering for communication middleware (2) ontology-driven software engineering and (3) ontology alignment adaption.

In the area of Model Driven Software Engineering for communication middleware, several works have shown that model driven software development can be used to ease the usage of middleware by using models to represent properties of data and middleware aspects [Ed04, BT10]. While these approaches show, that a model-driven approach can significantly reduce the lines of codes needed to be written, the models and the modeling languages are domain-specific and do not allow automated reasoning, which makes them less flexible. This shortcoming leads to the area of ontology-driven software engineering (2) that uses ontologies as a modeling language for model driven software engineering. One of the benefits of using ontologies in this context is, that ontologies can be easily linked to other ontologies and the declaration of these links allows for an automatic generation of model transformations [Pa12]. While approaches from this area can be used to model systems working in a real-time domain [St17], these models are static and do not account for changes in the environment or aligned ontologies.

Reacting to changes in aligned ontologies is an area of research that has been approached using several methods: Both applying rules to changes directly [Gr13b, do15] as well as applying rules to changing inferences [Ju16] have been proposed. In addition to these rule-based approaches, machine learning approaches have been explored [JI19]. While a rule-based approach has been proposed for a use case in the area of Interweaving Systems, this approach is tied to the domain-specific aspects of the use case [JI17]. To our knowledge, no domain-independent approach exists which facilitates environment sharing based on ontologies. This issue is at the core of the research presented in this paper.

3 Approach

The main goal of the approach presented in this work is to allow interweaving systems to cooperatively create a shared view of their environment. This approach needs to fulfill a set of non-functional requirements: (1) it needs to be fault-tolerant in the sense that it can still support a shared view on the environment when some systems fail, (2) it needs to be

fault-tolerant in the sense that it allows dealing with partial communication failures and (3) it may not interfere with real time properties of the interweaving systems.

Such an approach is beneficial for IWS which observe their environment and whose performance (including functional aspects) can be improved by sharing these observations. As environment sharing requires additional computational resources, the systems also need some kind of isolation between real-time critical aspects and the program that implements our approaches, for example implemented by a mixed-criticality system [BD13].

The core aspects of our approach consists of three main ideas: (1) An architecture for supporting data sharing in an interweaving systems context, (2) an implementation strategy for this model based on ontologies and (3) an algorithm that is used to adapt the ontology model to changes. This paper focuses on the first two ideas. Although the approach has to be adapted to the respecteive domain specifics, the core ideas are domain-independent. These ideas are discussed in the following subsections.

3.1 Architecture

A simplified example of an application of the framework with two systems is shown in Figure 1. For the sake of simplicity, ontologies are represented as storages, while they can also be implemented as a view on concepts, that exist in multiple other storages. Pre-existing components of interweaving systems are shown in yellow, our addition to these concepts is depicted in blue. In this paper, we will focus on the connector ontology, the directory ontology and the reaction to changes. The models used in the framework for knowledge sharing mainly need to fulfill two purposes: (1) let systems discover what information on the environment is available (2) bind information from remote locations to local needs. The first purpose is implemented using a model called directory ontology. The Directory Ontology contains information on two aspects: It represents which interweaving systems are able to exchange data. Each system can provide a set of sensors, that can be used to observe the environment. These sensors are also represented in the directory ontology.

The second purpose is fulfilled by a connector ontology for each system. The connector ontology, which is specific to each system, can label a sensor as local as a remote. Remote sensors are deployed on other systems. For remote sensors, connection information is stored. An example for connection information can be a DDS topic ID or a URL. In the directory ontology, sensors can be related to data needs, that are used to represent the requirements of data to improve the self-management capabilities of a system. The connector ontology also contains a similarity ontology, that stores similarity scores. These scores are used to associate remote sensors to other stand-in sensors which can replace them in case of failing remote sensors. (This includes failure of other systems and communication failures, which lead to missing remote sensor data.) Pairs of sensors are assigned a similarity score that represents how well readings from a sensor are suited to replace readings from the other sensor.

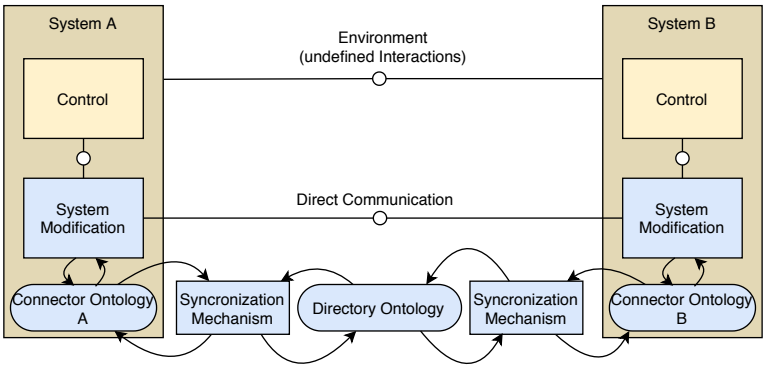


Fig. 1: Approach to Augmentation of IWS

A synchronization mechanism is used to integrate the directory ontology with local connector ontologies. This synchronization mechanism needs to fulfill two purposes: (1) ensure that the directory contains the most up-to-date information on sensors available on a system and (2) allow each system to access what sensors are available on other systems. Hence, this mechanism can be implemented by systems broadcasting their sensor capabilities in a predetermined interval. Based on this information, local systems can interact with each other to exchange data according to the definitions used in local and global ontologies. This information exchange can be implemented using a peer-to-peer framework such as DDS [Pa03] or any topic-oriented middleware that does not rely on a central broker. The same middleware can also be used to exchange directory ontology information.

3.2 Ontology Design

All models and, as far as possible, the change mechanism should be implemented using standardised ontology languages, so that standard tools can be used to process the models. Wherever possible, only subclass reasoning should be used. Subclass-based reasoning requires only low computation complexity. Therefore, in a resource-constrained setting, it is advisable to only use subclass-based reasoning.

These design guidelines lead to implementation strategies for the directory and connector ontology. An example for these ontology types is shown in Figure 2. The class *System* is a class, that contains all sensor readings. Therefore, if a sensor is part of a system, the class representing the sensor is a subclass of the class representing the respective system. A sensor reading would be an instance in this ontology. The connector part of the ontology represents the *Need* concepts and shows, what sensors are *External*. Every sensor that is a subclass of external is a sensor at a remote system that provides data to the local system. All connections between the connector and directory ontology are alignment connections. The similarity ontology part of the connector ontology, which is shown at the bottom of Figure 2 is based

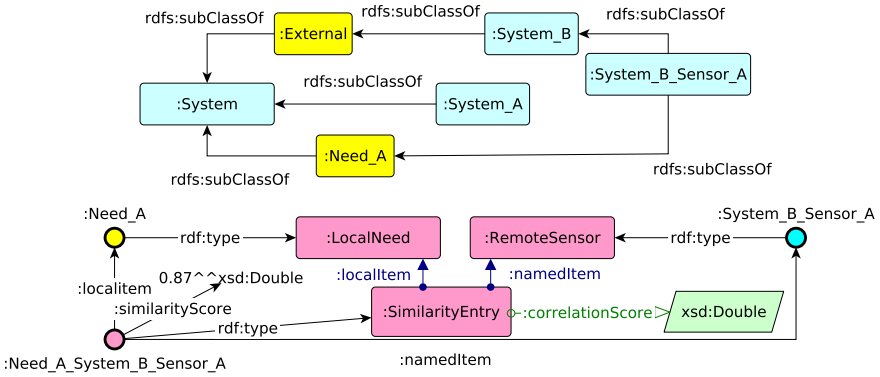


Fig. 2: Example connector ontology (yellow) with mapped directory classes (blue) including similarity ontology (bottom)

on *Similarity Entries*, that relate pairs of local needs and remote sensors to similarity scores. Due to design choices taken in modeling the connector ontology, connector ontology classes need to be represented as instances in the similarity ontology.

3.3 Adapting to changes

To adapt the mapping between directory and connector ontologies when the directory ontology changes, we use a rule-based technique published in [Ju16, JI17]. The basic idea of this approach is to compute inferences before and after a change occurs. A comparison of the inferences leads to a set of deleted and added inferences named Δ_{inf}. Rules are applied on the content of Δ_{inf} to identify which changes of the directory ontology would require the alignment to be adapted. Given this information, another type of rules can determine based on the similarity ontology and the content of Δ_{inf}, how the alignment needs to change.

4 Transformation to a Domain-Specific Example

To demonstrate the framework's application, we use an example from the area of autonomous traffic control systems first published in [JI17]. In this example, traffic control systems controlling the traffic lights at one intersection have a fixed set of signaling plans, that they can change based on traffic flow data. The approach presented in [JI17] is used to exchange traffic flow data between intersections, so that traffic control systems can incorporate traffic flow data measured at remote intersections into their decision making process.

Applying the approach requires some domain-specific artifacts to be created: change rules, the modification of the IWS and the incorporation of environment data into the traffic control

process are always domain specific and need to be created for each use case. Other aspects are transformed in the sense of model-driven software engineering. The specialization transformation is used to add the required domain-specific information to the abstract ontology model. In this case, the class *System* is used to represent traffic control systems. The class *Data* needs to be specialized so that it can hold data for traffic flow. Hence, a datatype is added to the class. *Data Needs* are used to represent data needs specific to intersection entries and exits. The remainder of the model remains unchanged. As described, when reacting to changes, rules are used to search Δ_{inf} for entries with instances of the class *External*. In this use case, a new alignment statement is generated so that the best candidate for a data need can be used. This process is based on a SPARQL query, that explicitly represents the change rule [JI17].

5 Discussion

The core contribution of this work is to present an approach, that shows how to build a data structure that can use mapping adaption to deal with changes in the environment of interweaving systems. The approach supports the separation of concerns on several levels: domain-specific concepts are separated from domain-independent ones, and creating a shared model of the environment is separated from the regular work of the systems. Also, rules for changes, domain-specific rules and rules for connecting systems are separated. Besides, the approach allows for rules concerning the systems to be formulated in an explicit, short and domain-independent way. This also means, that these rules are not hidden in the source-code of the system or as a part of assumptions of a domain-specific language.

6 Conclusion and Outlook

This paper shows, how an ontology-alignment-based approach can support data integration in interweaving systems and sustain this integration when changes in the environment occur. Using these technologies allows a modularization of several aspects that are important to the scenario and an explicit formulation using standardized, domain-independent modeling languages. Future work could explore more application areas for the approach such as the industrial manufacturing domain. Also, the quantitative benefit of this approach could be evaluated and compared to other methods of systems management.

Bibliography

[BD13] Burns, Alan; Davis, Robert: Mixed criticality systems-a review. Department of Computer Science, University of York, Tech. Rep, pp. 1–69, 2013.

[BT10] Beckmann, Kai; Thoss, Marcus: A model-driven software development approach using OMG DDS for wireless sensor networks. In: IFIP International Workshop on Software Technolgies for Embedded and Ubiquitous Systems. Springer, pp. 95–106, 2010.

[do15] dos Reis, Julio Cesar; Pruski, Cédric; Silveira, Marcos Da; Reynaud-Delaître, Chantal: DyKOSMap: A framework for mapping adaptation between biomedical knowledge organization systems. Journal of Biomedical Informatics, 55:153 – 173, 2015.

[Ed04] Edwards, George; Deng, Gan; Schmidt, Douglas C.; Gokhale, Aniruddha; Natarajan, Bala: Model-Driven Configuration and Deployment of Component Middleware Publish/-Subscribe Services. In (Karsai, Gabor; Visser, Eelco, eds): Generative Programming and Component Engineering. Springer, Berlin, Heidelberg, pp. 337–360, 2004.

[Gr13a] Groß, Anika; Dos Reis, Julio Cesar; Hartung, Michael; Pruski, Cédric; Rahm, Erhard: Semi-automatic Adaptation of Mappings between Life Science Ontologies. In (Baker, Christopher J. O.; Butler, Greg; Jurisica, Igor, eds): Data Integration in the Life Sciences. Springer, Berlin, Heidelberg, pp. 90–104, 2013.

[Gr13b] Groß, Anika; Dos Reis, Julio Cesar; Hartung, Michael; Pruski, Cédric; Rahm, Erhard: Semi-automatic Adaptation of Mappings between Life Science Ontologies. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7970 LNBI:90–104, 2013.

[JI17] Jurisch, Matthias; Igler, Bodo: Knowledge-Based Self-Organization of Traffic Control Systems. In: 47. Jahrestagung der Gesellschaft für Informatik, Informatik 2017, Chemnitz, Germany, September 25-29, 2017. pp. 947–954, 2017.

[JI19] Jurisch, Matthias; Igler, Bodo: Graph-Convolution-Based Classification for Ontology Alignment Change Prediction. In (Mehwish Alam, Davide Buscaldi et al., ed.): Proceedings of the Workshop on Deep Learning for Knowledge Graphs (DL4KG2019) Co-located with ESWC 2019, Portoroz, Slovenia, June 2, 2019. CEUR-WS.org, pp. 11–20, 2019.

[Ju16] Jurisch, Matthias: Managing Ontology Mapping Change based on Changing Inference Sets. In: Knowledge Engineering and Knowledge Management - EKAW 2016 Satellite Events. Springer, pp. 255–262, November 2016.

[MSSU11] Müller-Schloer, Christian; Schmeck, Hartmut; Ungerer, Theo: Organic Computing - A Paradigm Shift for Complex Systems. Springer-Verlag, Berlin, Heidelberg, 2011.

[Pa03] Pardo-Castellote, G.: OMG Data-Distribution Service: architectural overview. In: 23rd International Conference on Distributed Computing Systems Workshops, 2003. Proceedings. pp. 200–206, May 2003.

[Pa12] Pan, Jeff Z.; Staab, Steffen; Amann, Uwe; Ebert, Jürgen; Zhao, Yuting: Ontology-Driven Software Development. Springer Publishing Company, Incorporated, 2012.

[PB12] Pacher, Mathias; Brinkschulte, Uwe: Implementation and evaluation of a self-organizing artificial hormone system to assign time-dependent tasks. Concurr. Comput. Pract. Exp., 24(16):1879–1902, 2012.

[St17] Steinmetz, Charles; Schroeder, Greyce; dos Santos Roque, Alexandre; Pereira, Carlos Eduardo; Wagner, Carolin; Saalmann, Philipp; Hellingrath, Bernd: Ontology-driven IoT code generation for FIWARE. In: 2017 IEEE 15th International Conference on Industrial Informatics (INDIN). IEEE, pp. 38–43, 2017.

[To16] Tomforde, S.; Rudolph, S.; Bellman, K.; Würtz, R.: An Organic Computing Perspective on Self-Improving System Interweaving at Runtime. In: 2016 IEEE International Conference on Autonomic Computing (ICAC). pp. 276–284, July 2016.

Herausgeber et al. (Hrsg.): SENSYBLE,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2020 1

A Tangible Object for General Purposes in Mobile Augmented Reality Applications

Linda Rau [1], Robin Horst[1], Yu Liu[1], Ralf Dörner[1], Ulrike Spierling[1]

Abstract: Smartphones and tablets are common technologies within today's private living environments. They are well-suited to serve as a platform for mobile Augmented Reality (AR). Tangible AR is a subclass of AR which includes tangible objects and can make interactions intuitive. With this, new options for human-computer interaction become available at home. Based on literature research and design rationale, we identify requirements that help to develop a tangible object which can intuitively be used as tangible user interface (TUI) for mobile AR applications. Users should be able to handle the tangible object comfortably. Additionally, it needs to be reliably trackable with today's tracking algorithms. The tangible object should also offer affordances to the users. We strive to develop a single, versatile object that is usable in different application scenarios at home. We designed and 3D printed a tangible object that combines different surfaces and shapes. It offers various affordances and interaction possibilities. We argue that this allows users to trigger actions intuitively in an AR environment or to manipulate virtual content.

Keywords: Tangible Augmented Reality; Mixed Reality; Tangible User Interface

1 Introduction

Current smartphones and tablets are suitable devices to serve as mobile Augmented Reality (AR) platform because they are equipped with a camera and have sufficient computing power. Furthermore, AR supporting software can be preinstalled with the operating system, e.g., ARCore on Android or ARKit on iOs. The wide distribution of such devices in private life offers new options for human-computer interactions that are available at home. With this, the number of available mobile AR applications is also increasing. Compared to traditional 2D interfaces, mobile AR applications offer other possibilities for user interactions. One category for such user interactions is physical interaction. Literature such as [BKP08] shows that the intuitiveness of user interactions can be improved when a tangible real-world object serves as user interface. Using this object, users can manipulate objects in their real-world to manipulate virtual content. AR experiences can be improved with the interaction techniques that tangible objects allow because they can offer more affordances than solely virtual interfaces [Bu99]. The notion of affordance is discussed as properties of objects which invite for specific actions by [Gi77], e.g., a button suggests pressing or a door handle suggests levering it. Therefore, tangible objects could help users to have an intuitive tangible AR experience without frustration resulting from a non-intuitive interface.

[1] RheinMain University of Applied Sciences, Faculty of Design Computer Science Media,
Unter den Eichen 5, 65195 Wiesbaden, Germany. E-mail: firstname.lastname@hs-rm.de

In general, a large variety of objects can be utilized as tangible objects for a tangible user interface (TUI) in an AR application. Images or paper cards are commonly used as image targets and are popular for AR applications for private use. One 3D tangible object that is used for AR applications in private life is a cube, e.g., the merge cube [Me20]. While a cube can be a suitable object for viewing virtual 3D content, other objects could suggest more or other affordances. Thus, tangible objects providing affordances can help users to understand interaction possibilities for a mobile AR application without or with a short learning phase beforehand.

Our contributions in this paper are the following. Based on literature research and the elaboration of a design rationale, we identify and explore requirements for a tangible object that can be used as TUI for AR applications. We propose a generic tangible object that can be used in diverse AR application scenarios. It provides several geometrical forms which offer different affordances. With a 3D printer, we create a physical instance from our digital model and discuss which of its properties are conforming to our proposed requirements.

This paper is organized as follows. In the next section, we present related work. Following this, we identify and describe our three requirements. We applied our findings and designed our tangible object, which we introduce in section 4 and evaluate in the fifth section. We draw conclusions in section 6 and share our thoughts on future work.

2 Related Work

Early ideas for TUIs have been discussed in [WMG93]. Following this, Fitzmaurice et al. [FIB95] introduced their concept of graspable interfaces, where they use objects, e.g., handles, to manipulate digital content. These inspired the Tangible Media Group [IU97] for their vision of tangible bits, where the users' physical surrounding becomes an interface itself as objects or surfaces are linked to digital information.

Billinghurst et al. [BKP08] describe Tangible AR interfaces as an intuitive way to interact with an AR interface where users can manipulate a physical tangible object to manipulate a virtual object which is registered to the physical one. They describe design principles for TUIs which can help to develop an intuitive TUI for AR applications. They state that a Tangible AR interface that follows their design principles is intuitive to use and facilitates seamless display and interactions. However, they do not cover design principles for tangible objects but concentrate on functional requirements to the interface, e.g., support for multiple handed-interactions or multiple activities and objects. Furthermore, they propose four prototype AR applications with tangible interfaces: Shared Space, ARgroove, Tiles, and VOMAR. Planar images are used as tangible interfaces in Shared Space and ARgroove. For Tiles, the images have various shapes that are mapped to different semantics. The unique functionality of each image target is similar to the unique functionality of each icon and tool on a computer desktop interface. In VOMAR, they use a trackable cardboard paddle as tangible interaction device.

Different tangible objects have been explored to manipulate AR content. For example, a trackable pen is used for MARS [Hö99] and Studierstube [SFH00]. Other tangible objects for AR applications can be a cup, which allows users to modify virtual object [Ka03], or a cube [Ji15; Me20].

The tangible AR applications and interfaces named above demonstrate that different shapes and objects can be conceivable as tangible interfaces. Some tangible objects offer affordances for specific interactions but are dependent on the use case and cannot be reused in another context, e.g., images or objects that represent one specific real-world object. Generic tangible objects can offer affordances for general purposes and hence be used in several use cases. However, for some use cases, their affordances are not sufficient. For example, a pen offers affordances for tasks like writing and selecting, but no affordances for examining 3D content. In contrast, a cube offers the affordance to examine 3D content but no affordances for writing.

3 Requirements for a Generic Tangible Object in Tangible AR

We identified three requirements for a tangible object that is supposed to be used as generic AR-based TUI. In contrast to tangible objects that are developed for one specific use case, a generic object can be used for several applications. Therefore, it can contribute to a simple TUI where no switching between several objects is necessary. In this section, we go into detail for the three requirements.

This first category is the object's attributes to make it reliably detectable and trackable with current tracking algorithms. This is helpful to provide a frustration-free and immersive user experience. For example, when immersed in an AR experience, the users can feel present in the AR to the point that virtual content is not perceived as virtual but as part of the physical world. Then, whenever the tracking is lost, this illusion can be disturbed or completely disrupted. Additionally, users need to wait and eventually need to put either the tracked object or the AR device in another position or to change the lighting conditions before the tangible object can be detected and tracked again which further disturbs the AR experience. Therefore, the tangible object needs to consist of reliably detectable and trackable textures or shapes and surfaces. This is dependent on the used tracking algorithms because not all tracking algorithms detect or track the same properties. Such properties of an object can be its shape, features in its texture, colors, or a combination of these. Several tracking algorithms make use of more than one property because this adds to tracking stability. For example, the tracking software Vuforia detects objects by shape, but additional information about the material, such as colors, significantly improves the robustness [Pa20]. Therefore, a reliably trackable object should ideally combine multiple well detectable and trackable properties. When users hold the tangible object, they might occlude these properties in part or entirely, which can make the object hard to detect or contribute to a tracking loss. To provide enough detectable and trackable properties in despite of occlusion, each part or each side of the tangible object should be reliable trackable.

The second category we identified is the tangible objects' handling which includes its size, weight, and shape. A study by Sheridan et al. [Sh03] finds that tangible objects, in their case cubes, should naturally fit in the user's hand. We conclude this makes a recommended size for the tangible object dependent on the user's hand size. For example, a child can have smaller hands and require a smaller tangible object than an adult. A size of 8 cm × 8 cm × 8 cm is specified as suitable by Jimenez et al. [Ji15]. They use a webcam with a resolution of 2 megapixels and a focal length of 3.7 mm and find the resolution to be sufficient for target recognition with a distance of up to 1.5 m between object and camera. Furthermore, they state this is a mean value between sizes suggested by AR software developers. In their scenario, the user has both hands available and can choose to grab the object with one or both hands. However, in mobile AR users might be required to hold their AR device and therefore have only one hand left to use a tangible object as TUI. In this case, a smaller size can make the handling with one hand easier. Beside a tangible object's size, its weight influences its handling. Holding a tangible object can become exhausting for the user, especially if it is heavy. Therefore, we determined a tangible object should preferably be as light as possible, but yet durable, to make its usage less exhausting for users. A lightweight object is usually achieved without further effort due to a relatively small size. For example, a 3D printed cube of size 8 cm × 8 cm × 8 cm and with 25% infill (a common density setting) weighs about 150 grams. This value is dependent on the printing material, but common 3D printing materials have similar densities, except for metals. Furthermore, the tangible object's shape has an impact on its handling. The study by Sheridan et al. [Sh03] finds that an object's geometry affects how well users can grasp it. The authors state that curves in an object's geometry as well as a high surface area can enhance its grip. For example, they explain that a rhomboid or star-shaped object is easier to grasp than a cube. They also find that the object's material has an impact on its grip and that a flexible material can aid in grip.

We identified an object's affordances as third category. If specific properties of an object invite for certain actions, they offer affordances. This suggests that we can provide affordances for certain actions, that users can perform in an AR environment, by the tangible object's design. For instance, a hemisphere shape can be perceived as a button and therefore afford to push it. This indicates that various shapes can afford distinctive actions. Therefore, we suggest designing the tangible object consisting of various shapes, that each offer affordances for specific or several actions. General interactions, that a general tangible object could support and that can be useful in several AR applications, could be selecting, viewing 3D content, navigating, or scrolling.

4 Proposal of a Generic Tangible Object

Based on the requirements identified above, we designed an object that can serve as TUI for general purposes in AR applications and describe it in this section. The digital model of our tangible object is shown in Figure 1.

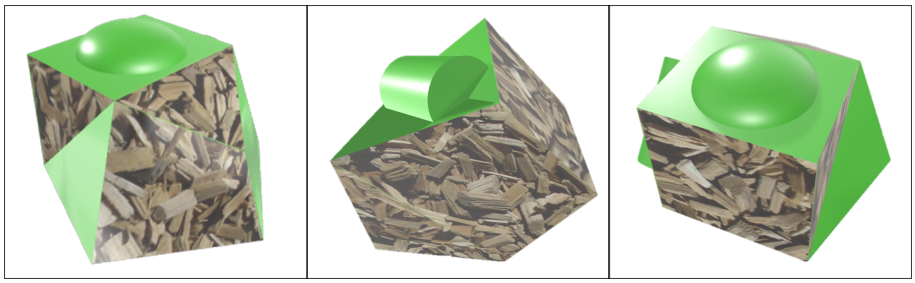


Fig. 1: Digital model of a tangible object for intuitively manipulating virtual content of an AR

We used Vuforia's image target example *chips* to create a texture for the tangible object. It was printed on adhesive film and glued to the object's surfaces. Because the image target is specifically designed to be detected and tracked, it provides a high number of trackable features. Besides this, trackable properties of our tangible object can be its edges and overall shape. The hemisphere and cylindrical shaped parts contribute to an asymmetric shape.

We use PLA material for 3D printing, which has a density of avg. 1.3 g/cm³. The PLA printing material is rigid and allows a firm grip without damaging or compressing the tangible object. Our object was printed at a size of 5.5 cm × 5.5 cm × 4 cm. The chosen infill density of 20% results in a weight of 23 g. The surfaces of our object include one pentagon, four quadrangles, and five triangles. The pentagon has a size of 5.5 cm × 5.5 cm at its widest place. The edges of two quadrangles are 2.8 cm, 2.1 cm, 3.6 cm, and 3.4 cm. A third quadrangle, which is a rectangle, has a size of 3.2 cm × 3.6 cm. From the fourth rectangle surface with size 3.6 cm × 3.0 cm, a hemisphere shaped part with 2.8 cm diameter and height 0.7 cm sticks out. Two of the triangles are isosceles. The first triangle has a base of 3.6 cm and a height of 2.0 cm while the second one has a base of 4.4 cm and a height of 2.2 cm. A third triangle measures 3.6 cm, 3.3 cm and 4.4 cm on its edges. A cylindrical shape (2.5 cm height and 0.7 cm diameter) sticks out from two further triangle surfaces which have the same sizes as the second and third triangle. This results in 28 edges, 16 vertices, and two round shapes. These various surfaces and shapes offer several affordances, depending on the use case. Our surfaces (triangles, rectangles, and one pentagon) can be augmented with different shaped images or virtual content. The cylindrical shape can suggest user interactions like scrolling, zooming, or turning over a page in a virtual book. Pushing a button or buzzer can be suggested with the hemisphere shape.

5 Evaluation and Discussion

We created a physical instance of our model using a 3D printer. We inspected how our proposed tangible object meets the requirements. Our results are described and discussed in this section.

Initially, the tangible object was printed with a 3D printer from a single material in one color. This provided few trackable features and the shapes' edges did not contrast well from the rest of the object. Therefore, the object could not be detected by the tracking software. To improve the object's tracking conditions, we applied a texture to both, the tangible object and its digital model. The digital model with UV mapped texture can be used as a reference for the tracking algorithm. However, gluing the texture on the tangible object is imprecise. With these differences between the physical object and the digital one, the tracking performance is insufficient. Another approach is to scan the physical object to use the scan data as a reference for the tracking algorithm. In this case, suitable lighting conditions must be provided during the scan process. With this approach, our tangible object can be tracked well while the lighting conditions are similar to the ones during the scan. The surface that the object rested on during the scan process is not included in the scan data. Therefore, it cannot be detected or tracked. A second scan from another side is required to make the tangible object reliably trackable from all sides. This process is sufficient but time-consuming.

Our 3D printed object is designed for one-handed interactions in mobile AR and therefore relatively small. Depending on the user's hand size, this can result in occlusion of a substantial number of trackable features or properties and therefore disturb the tracking quality. We find the object can comfortably be grabbed on its edges so that only a few parts are occluded. This is visualized in Figure 2.

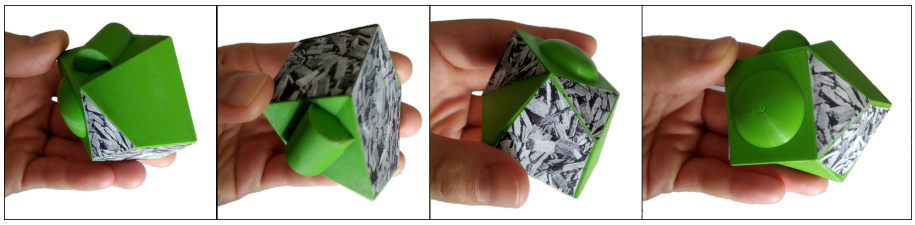


Fig. 2: Views of a tangible object for intuitively manipulating virtual content of an AR

6 Conclusion

With this work, we explored three requirements for intuitive tangible objects serving as TUI for mobile AR applications: tracking conditions, handling (weight, size, shapes), and affordances. We proposed an example for such a tangible object where we considered these. Based on our three requirements, our object is indicated to be suitable for intuitive interactions in an AR environment.

In future work, our proposed tangible object can be improved regarding its size and trackable properties or features. We found our size of 5.5 cm × 5.5 cm × 4.0 cm acceptable, but a greater size would result in less occlusion and therefore can contribute to a stable tracking. We applied a texture to our object to make it reliable trackable, but this made a time-consuming

scanning task to set the object with texture as reference for the tracking software necessary. Future work can include developing 3D-printable textures that can easily be detected and tracked with current tracking algorithms.

Acknowledgments

This work has been funded (in part) by the German Federal Ministry of Education and Research (BMBF), funding program Forschung an Fachhochschulen, contract number 13FH181PX8.

References

[BKP08] Billinghurst, M.; Kato, H.; Poupyrev, I.: Tangible Augmented Reality. In: ACM SIGGRAPH ASIA 7. 2008.

[Bu99] Butz, A.; Hollerer, T.; Feiner, S.; MacIntyre, B.; Beshers, C.: Enveloping Users and Computers in a Collaborative 3D Augmented Reality. In: Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99). IEEE Comput. Soc, pp. 35–44, 20-21 Oct. 1999.

[FIB95] Fitzmaurice, G. W.; Ishii, H.; Buxton, W. A. S.: Bricks: Laying the Foundations for Graspable User Interfaces. In: Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '95. ACM Press, pp. 442–449, 1995.

[Gi77] Gibson, J. J.: The concept of affordances. Perceiving, acting, and knowing 1/, 1977, URL: https://www.macs-eu.org/slides/02_MACS_Affordances%20s-o-a_METU.pdf.

[Hö99] Höllerer, T.; Feiner, S.; Terauchi, T.; Rashid, G.; Hallaway, D.: Exploring MARS: Developing Indoor and Outdoor User Interfaces to a Mobile Augmented Reality System. Computers & Graphics 23/6, pp. 779–785, 1999.

[IU97] Ishii, H.; Ullmer, B.: Tangible Bits. In: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, pp. 234–241, 1997.

[Ji15] Jiménez Fernández-Palacios, B.; Nex, F.; Rizzi, A.; Remondino, F.: ARCube-The Augmented Reality Cube for Archaeology. Archaeometry 57/, pp. 250–262, 2015.

[Ka03] Kato, H.; Tachibana, K.; Tanabe, M.; Nakajima, T.; Fukuda, Y.: MagicCup: A Tangible Interface for Virtual Objects Manipulation in Table-top Augmented Reality. In: ART 2003. IEEE, pp. 75–76, 2003.

[Me20] Merge Labs, Inc.: Merge Cube | AR / VR Lernen & Erstellen, accessed: 07.04.2020, URL: https://mergeedu.com/cube.

[Pa20] Parametric Technology GmbH: Model Targets Supported Objects & CAD Model Best Practices, accessed: 03.04.2020, URL: https://library.vuforia.com/content/vuforia-library/en/articles/Solution/model-targets-supported-objects.html.

[SFH00] Schmalstieg, D.; Fuhrmann, A.; Hesina, G.: Bridging Multiple User Interface Dimensions with Augmented Reality. In: Proceedings IEEE and ACM International Symposium on Augmented Reality (ISAR 2000). IEEE, pp. 20–29, 5-6 Oct. 2000.

[Sh03] Sheridan, J. G.; Short, B. W.; van Laerhoven, K.; Villar, N.; Kortuem, G.: Exploring Cube Affordance: Towards a Classification of Non-Verbal Dynamics of Physical Interfaces for Wearable Computing. In: IEE Eurowearable '03. IEE, pp. 113–118, 4-5 Sept. 2003.

[WMG93] Wellner, P.; Mackay, W.; Gold, R.: Back to the real world. Communications of the ACM 36/7, pp. 24–27, 1993.

Integration of Game Engine Based Mobile Augmented Reality Into a Learning Management System for Online Continuing Medical Education

Robin Horst[1], Dennis Fenchel[2], Reimond Retz[2], Linda Rau[1], Wilhelm Retz[1], Ralf Dörner[1]

Abstract: Physicians must participate in continuing medical education (CME) as part of the medical quality assurance. One possibility is to take online courses in their private living environment. These courses are mostly text- or video-based. Novel technologies such as mobile Augmented (AR) or mobile Virtual Reality (VR) are not yet established although their usage is not out of bounds in private homes anymore. Game engines can facilitate the authoring of applications that utilize VR/AR, as they provide many crucial functionalities out of the box. However, integrating the resulting VR/AR software in online CME courses is not trivial. In this paper, we investigate this integration into an existing learning management system (LMS) for online CME. In the example of a mobile AR application, we propose a system design that extends a course by a mobile AR part. We describe our implementation and how we transition users from their familiar web-interface on the desktop PC to a mobile AR application.

Keywords: Professional Education in Private Living Environments; Online Continuing Medical Education; System Design; Augmented Reality; Virtual Reality; Games Engineering; E-Learning

1 Introduction

Many professionals such as health professionals use their private living environments for continuous education in their field. Continuing medical education (CME) comprises training measures that serve to maintain and permanently update the professional competence of the medical profession. CME also serves as a part of the medical quality assurance. These compulsory training activities for physicians are demanded by governmental-related organizations, such as the Accreditation Council for Continuing Medical Education (USA) or the Bundesärztekammer (Germany). Physicians are required to obtain a specific amount of credits, which can be earned through different activities. One of these activities is online training. Online CME is mostly conducted using technology that is already available in the private living environments of physicians, such as common desktop PCs, tablets or smart phones. Different established media are used here to mediate information, for example, text, images, audio files or videos. In the context of lifelong learning, private living environments are also expected to support modern technologies for learning, such as Augmented (AR) and Virtual Reality (VR) on mobile devices. Game engines such as Unity [Un20] are suitable tools to support authors in creating applications for such technologies. However, integrating

[1] RheinMain University of Applied Sciences, Kurt-Schumacher-Ring 18, 65197 Wiesbaden
[2] health&media GmbH, Dolivostraße 9, 64293 Darmstadt

mobile AR and VR applications developed with game engines into existing systems for online CME and courses involves challenges for authors.

In this paper, we make the following contributions:

- We investigate the integration of mobile AR into an existing CME course that is based on a learning management system (LMS). Our goal is to extend a traditional online course with novel technology rather than replacing it entirely.
- In the example of a mobile AR application (app) that we built with Unity, we highlight crucial aspects of its integration in a CME course. Based on a prototype, we report and discuss lessons learned from the implementation process.
- Physicians usually participate in online courses on their desktop PC. To perceive the AR part, they must switch to our mobile AR app within the course. We propose a transitioning technique that guides users from the web-course to our mobile AR app.

This paper is organized as follows. The next section discusses related work. In Section 3, we describe the LMS that we integrate the mobile game engine technology into. In the fourth section, we present our system design that realizes the integration. Section 5 provides a conclusion and points out directions for future work.

2 Related Work

Web-based e-learning courses are widely used for CME. Various technologies can facilitate the authoring of such courses, including LMSs [CGB09]. There exists a variety of LMSs that serve different purposes. Moodle [DT03] (Modular Object-Oriented Dynamic Learning Environment) is an established multi-purpose and open source LMS. There are also more specific LMSs such as OLE [PDG20] (Open Learning Environment). OLE was developed to serve individual requirements of the local learning context of a university.

Different media can be integrated in LMS courses. Persia et al. [PDG20] show that courses using a variety of media highly support learning activities compared to solely text-based education. The user satisfaction was particularly improved when different multimedia were exploited. They also show how to integrate educational videos and smart text in their LMS courses but did not investigate AR or VR technology. Recent work by de Paive Guimarães et al. [Pa17] propose a tool that utilizes educational AR content in the SCORM learning object standard. It creates AR learning objects and provides packaged AR applications to Moodle courses. It decouples LMSs and AR applications by distributing the applications through online repositories [Pa18]. Therefore, each AR application can be built with separate tools itself, as long as it is packaged with their tool in the SCORM standard. However, each time a course requires AR content, a novel AR application must be downloaded. With a shared runtime environment, users only would have to download content of an application. This facilitates the content delivery of AR applications within LMS courses.

Contrary to this approach by de Paive Guimarães et al. [Pa17, Pa18], Coma-Tatay et al. [Co19] integrate the AR learning content without external applications or plug-ins. They provide a tool (FI-AR) that is based on the open-source software framework FIWARE [Fo20]. It utilizes universal web-technology access to visualize AR content within their online courses. Visualizing AR or VR content within an LMS environment brings the challenge of transitioning from traditional media to these immersive technologies. Dodd and Antonenko [DA12] propose to use signaling methodologies to facilitate the transition for users that enter the virtual worlds. They provide users visual cues to guide them during this transition and use cues during the virtual learning activities until learners return to the traditional LMS content.

Generally, existing work on LMS and virtual technologies focuses on supporting separate custom-made AR and VR applications. Established authoring tools, such as game engines, are not considered. Furthermore, desktop PCs are targeted. Concepts for integrating AR and VR that runs on a mobile device within online courses are lacking.

3 Existing Continuing Medical Education System

Our existing *arztCME* [hG] e-learning framework for CME comprises both LMS and content management system (CMS) functionality aligned with specific requirements to comply with the certification of CME. The foundation of the CME system is a CMS based on WordPress that offers PHP and MySQL interfaces. It was extended with plugins to provide LMS functionality. The LMS features support authors to create CME courses and tests for assessing the learning success of physicians in order to give them credits. One CME course takes approximately 45 minutes and is consists of several 'pages' of learning content.

Physicians usually participate through web-browsers on a desktop PC. Some use mobile devices. In addition to static course representations as PDF documents, courses can be represented as multimedia HTML realization within our e-learning framework (Fig. 1). Established media in this realization are images, videos and texts. Our web-technology can directly integrate them in the HTML environment.

4 Integration of a Mobile Unity Augmented Reality App

There exist ways to include AR and VR technologies directly in a web-browser (e.g., [Li04, La19]). As our online CME courses already take place in a web-environment, a naive integration of these technologies makes them available directly in the web-browser. Unity also offers such a solution. It provides a web-player which can run Unity applications directly in the web-browser. However, this player and specific AR and VR features are designed for web-browsers that run on desktop PCs. Access to these players through web-browsers on mobile devices is possible but not entirely supported. The performance can suffer when

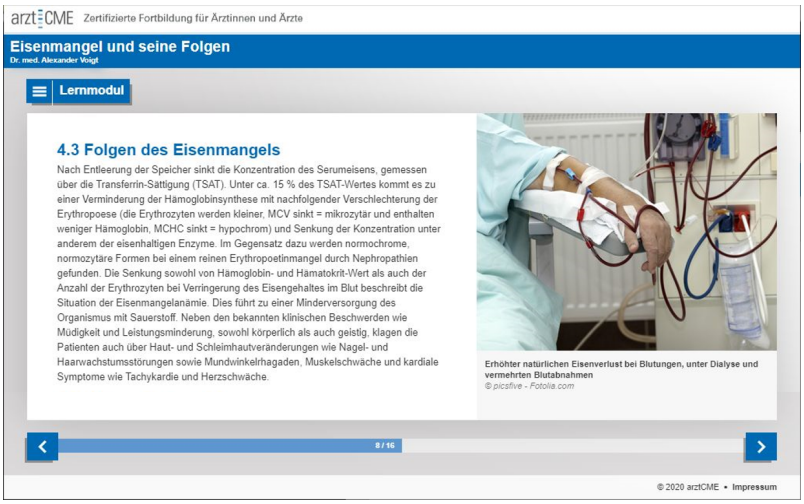


Fig. 1: The view on one page of a traditional online CME course for German-speaking physicians.

using computationally intensive AR or VR functionalities through a mobile web-browser due to the complexity of additional rendering steps that the browser needs to go through [BRR16]. Running these technologies in dedicated apps on a mobile device can increase the performance.

To provide an AR experience at a specific point during the course, users are required to switch from the traditional CME course that runs on their desktop PC to our mobile app. To facilitate the transition between the two environments and to induce the user doing so, we integrated an easily accessibly onboarding approach within our system design (Fig. 2). Using QR codes (Fig. 3), an established linking technique within the mobile domain, we are able to transfer users from a page within the course to the AR app. Furthermore, using dynamic QR codes, created with JavaScript at runtime, we are also able to include context specific information during the transitional phase. This provides an opportunity to attach a dynamic session ID, that can be used in further steps of the onboarding process to retrieve user-specific data, such as the name of the course that the user is in and the current state within the course.

To proceed in the course, users scan the QR code with their mobile device. In case the users do not have the corresponding AR app already installed on their mobile device, they are redirected to the store app of their mobile operating system (OS) (e.g., Appstore for iOS and Google Play Store for Android). Here they can download our AR app. In case the app already exists on the mobile device, it is opened directly instead. We use Firebase Dynamic Links [Go20a] to realize these re-directions. Firebase Dynamic Links (FDL) can store the session IDs from the QR code dynamically and transfer them through the installing process to our mobile app. The app can use the session ID to get further information about

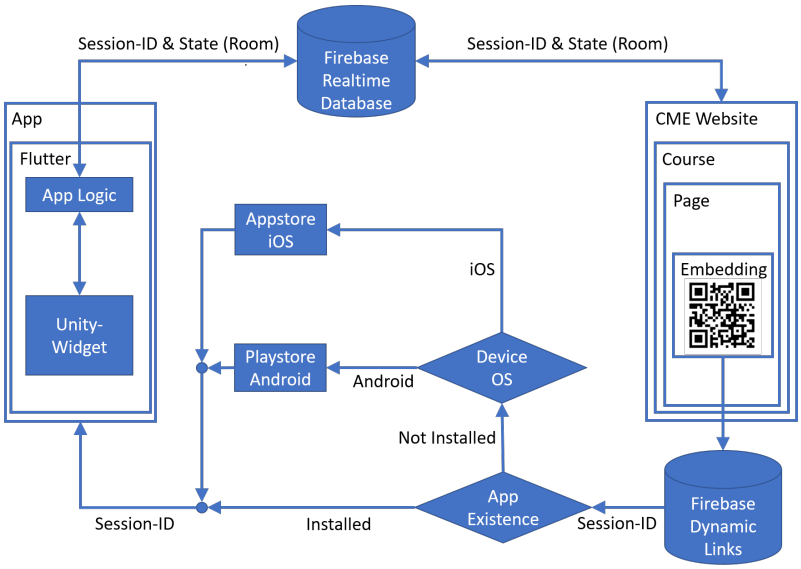


Fig. 2: Our system design extends an existing course in one specific page with mobile AR content. An onboarding process facilitates the transitioning from the website on a desktop PC to the mobile app.

the user from a corresponding Firebase Realtime Database (FRTD) [Go20a] that is also connected to our arztCME LMS (Fig. 2). It uses an open web socket subscription model for the connection. Both the app and the website can access and manipulate entries in the database in realtime. A basic session management system and IDs that track the logged-in users was already included in the original LMS. This has been extended to include current AR content and the information required for it.

Now users can be directed from a course to a separate AR application for each course that extends it. However, it can be cumbersome for users when they have to download a dedicated AR app for each specific course they participate in. To counteract this, we decided to build a single app that serves as AR platform for all courses. When using one application that provides the content for all available courses, it can still be tiresome for users to open the app and search the right content for their current course. We make use of the realtime connection of our app to the web-based LMS to open the app at a specific state directly after scanning the QR code instead of just starting the app. The session ID that we transferred throughout the onboarding process with FDLs can be used to query necessary information such as the specific location within a course. Alternatives without FDL would require scanning the QR code multiple times (e.g., at first for linking to the store or opening the app and then for opening the right content within the app).

The app development itself can be divided into two separate blocks. All AR-related functionality was developed in a Unity project, whereas all mobile-related functionalities

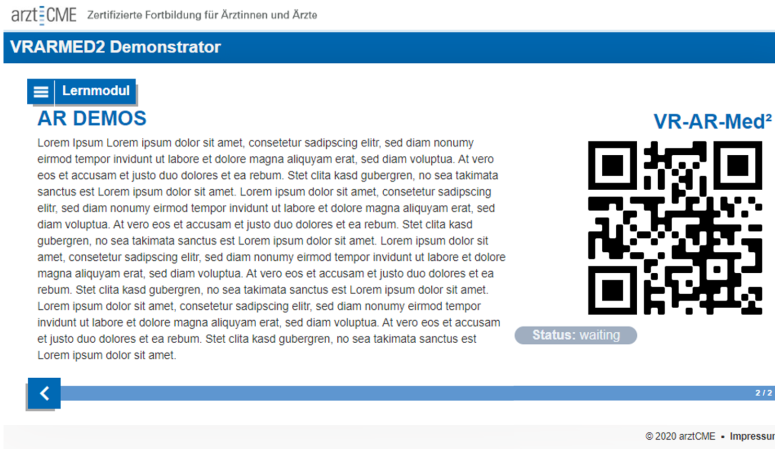


Fig. 3: A screenshot of a page within a course that includes a QR code used for transitioning between the desktop web-browser and our mobile game engine app.

(e.g., mobile app front end) were developed in a separate Flutter [Go20b] project. The advantage of using Flutter for mobile app development is that it supports cross-platform development. It also has the advantage that user interfaces (UIs) developed with Flutter look more conventional to UIs of regular mobile apps in contrast to mobile game UIs build with Unity. Furthermore, Flutter offers interfaces to a variety of common tools that facilitate mobile app development. However, splitting the development requires the integration of the Unity-based AR functionality within the Flutter app environment. This is a challenging task, since Unity development for mobile devices will normally build separate APK or IPA files which cannot easily be integrated in a Flutter app. As of Unity version 2019.3, Unity allows to build a Unity project as a library to be included within other apps. In a Flutter-App, the embedded Unity library can directly be used as a Widget to display its content. This Unity Widget can be developed in the Unity authoring environment without further restrictions except one. Unity Widgets are not directly suitable for an integration in a Flutter multi-platform code basis. For example, it is only supported to display AR Widgets in full-screen- mode. We also had to establish an asynchronous communication interface between the Flutter environment and the Unity Widget that is used as a plugin in Flutter. This communication is used to start the app directly with the content while correctly initializing the application's state. For example, separate Unity scenes can be addressed, or the Flutter app can tell the Unity Widget which image targets for the AR functionality to use. The Unity Widget also sends information through the Flutter environment back to the LMS. For example, this is used to share quiz results for later use in the evaluation of the CME course to award the credits for physicians when the course is successfully finished. The resulting app is illustrated in Fig. 4.

Fig. 4: Left: Start screen of our AR app. Users can select a course. Middle: A course about the functionality of the human heart is selected. It contains AR content in three specific locations within the course. Right: A heart displayed with AR technology. Our onboarding approach is able to direct users from the web-interface directly to this part of the app, even if it was not installed previously.

5 Conclusion and Future Work

In this paper, we investigated the integration of mobile AR into an existing LMS for online CME. In the example of extending one of our courses with an AR functionality, we described how we realized this integration. We propose a system design and a suitable onboarding approach that facilitates the transition from the traditional online CME course on a desktop PC to its AR part on our mobile AR app.

Future work relating to our contribution can be divided into two categories. Now that the mobile AR can technically be integrated within our courses, a user test with physicians can be conducted to draw conclusions on the acceptance of such technologies within the CME domain. As these technologies are novel to CME, certification of such content must also be addressed. Secondly, we will explore how new AR content can be delivered to users efficiently. As of now, our app includes the AR content for all courses but when more and more courses get an AR extension, the size of our app will grow vastly. Content can be delivered on demand. However, delivering novel content to a ready-built Unity application is not a trivial task. Furthermore, virtual assets such as 3D models can have large file-sizes so that downloading on demand may interrupt the learning flow of a course depending on the Internet connection of the mobile device.

Acknowledgements
This project (HA project no. 690/19-10) is financed with funds of LOEWE – Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz, Förderlinie 3: KMU-Verbundvorhaben (State Offensive for the Development of Scientific and Economic Excellence).

Bibliography

[BRR16] Butcher, Peter WS; Roberts, Jonathan C; Ritsos, Panagiotis D: Immersive analytics with webvr and google cardboard. Posters of IEEE VIS, pp. 30–32, 2016.

[CGB09] Cirulis, Arnis; Ginters, E; Brigmanis, K: Virtual reality's technologies use in e-learning. In: Proceedings of the 8th WSEAS International Conference on E-Activities, WSEAS Press, Puerto De La Cruz, ESP. pp. 148–153, 2009.

[Co19] Coma-Tatay, Inmaculada; Casas-Yrurzum, Sergio; Casanova-Salas, Pablo; Fernández-Marín, Marcos: FI-AR learning: a web-based platform for augmented reality educational content. Multimedia Tools and Applications, 78(5):6093–6118, 2019.

[DA12] Dodd, Bucky J; Antonenko, Pavlo D: Use of signaling to integrate desktop virtual reality and online learning management systems. Computers & Education, 59(4):1099–1108, 2012.

[DT03] Dougiamas, Martin; Taylor, Peter: Moodle: Using learning communities to create an open source course management system. In: EdMedia+ Innovate Learning. Association for the Advancement of Computing in Education (AACE), pp. 171–178, 2003.

[Fo20] Foundation, FIWARE: , FIWARE. https://www.fiware.org/, 2020. Accessed: 13.04.2020.

[Go20a] Google: , Firebase Realtime Database and Dynamic Links. https://firebase.google.com/docs, 2020. Accessed: 13.04.2020.

[Go20b] Google: , Flutter. https://flutter.dev/, 2020. Accessed: 13.04.2020.

[hG] healthmedia GmbH: , arztCME. https://www.arztcme.de/year={2020},. Accessed: 13.04.2020.

[La19] Lam, Kit Yung; Lee, Lik Hang; Braud, Tristan; Hui, Pan: M2A: A Framework for Visualizing Information from Mobile Web to Mobile Augmented Reality. In: 2019 IEEE International Conference on Pervasive Computing and Communications (PerCom. IEEE, pp. 1–10, 2019.

[Li04] Liarokapis, Fotis; Mourkoussis, Nikolaos; White, Martin; Darcy, Joe; Sifniotis, Maria; Petridis, Panos; Basu, Anirban; Lister, Paul F et al.: Web3D and augmented reality to support engineering education. World transactions on engineering and technology education, 3(1):11–14, 2004.

[Pa17] de Paiva Guimarães, Marcelo; Alves, Bruno; Martins, Valéria Farinazzo; dos Santos Baglie, Luiz Soares; Brega, José Remo; Dias, Diego Colombo: Embedding augmented reality applications into learning management systems. In: International Conference on Computational Science and Its Applications. Springer, pp. 585–594, 2017.

[Pa18] de Paiva Guimarães, Marcelo; Alves, Bruno Carvalho; Durelli, Rafael Serapilha; de FR Guimarães, Rita; Dias, Diego Colombo: An Approach to Developing Learning Objects with Augmented Reality Content. In: International Conference on Computational Science and Its Applications. Springer, pp. 757–774, 2018.

[PDG20] Persia, Fabio; D'Auria, Daniela; Ge, Mouzhi: Improving Learning System Performance with Multimedia Semantics. In: 2020 IEEE 14th International Conference on Semantic Computing (ICSC). IEEE, pp. 238–241, 2020.

[Un20] Unity Technologies: , Unity game engine. https://unity.com/, 2020. Accessed: 13.04.2020.

Presenters in Virtual Reality in Slideshow Presentations

Robin Horst[1] Linda Rau[1] Lars Dieter[1,2] Manuel Feller[1,2] Jonas Gaida[1,2] Andreas Leipe[1,2] Julian Eversheim[1,2] Julia Wirth[1,2] Jörn Bachmeier[1,2] Julius Müller[1,2] Maik Melcher[1,2] Ralf Dörner[1]

Abstract: Slideshow presentations have become ubiquitous in our everyday life and are used for communicating information of different kind. In this paper, we consider two different concepts that include both slides and VR technology in one presentation, *mixed presentations* and *virtual presentations*, and examine the role of the presenter in these concepts. We conducted three user studies which indicate that it is not necessary that presentations need to be held completely in VR as both virtual and mixed presentations were accepted by our participants and that our participants preferred immersed presenter integrations.

Keywords: Short Virtual Reality Experiences; Slideshow Presentations; Game Engine Integration; E-Learning

1 Introduction

Presentation software, such as PowerPoint, has become a standard tool in different environments of the everyday life, such as work, home or education and supports communicating information. Such software already supports different established resources, such as text, images, sound and video. Virtual Reality (VR) is not among these established means, even though head-mounted displays (HMDs) for VR become affordable and applicable concerning the costs and the ease of use. Therefore, VR is no longer reserved for expert use, but becomes more and more a part of daily life of the public. However, there exist challenges that must be considered before using VR among other resources in slideshow presentations. While audience takes the active part of a VR-mediated presentation and uses HMDs to experience the virtual content, presenters still need basic controls over the presentation procedure (e.g., switch to next/previous slides) to comply with fundamental constraints, such as time limitations. Another challenge relates to the technical integration of VR technology in common presentation software. How can a switch from a common PowerPoint slide to a VR experience be realized?

In this paper, we make the following contributions: We investigate how presenters can be integrated in *mixed presentations*, where a regular slide presentation switches to and

[1] RheinMain University of Applied Sciences, Kurt-Schumacher-Ring 18, 65197 Wiesbaden, Germany, firstName.lastName@hs-rm.de
[2] These authors contributed equally.

from VR applications, and *virtual presentations*, where slides are adopted within a virtual environment. We implemented three prototypes to explore the user-role of the presenter and show how short game-engine-based VR experiences can be integrated within established presentation software. We discuss aspects of our implementations, show VR applications can be triggered from a PowerPoint presentation and use our prototypes to draw conclusions how the audience perceived the attendance and influence of presenters.

2 Related Work

In the literature, there exist examples for VR applications where one user hast to guide another user through the virtual world. A remote instructor guides persons to repair complex machinery [Od15], or a researcher conducts a virtual demo [HD18, HD19].

Fuhrmann et al. [Fu01] contribute technical work about presentation systems that use VR technology. They adopt the slide concept of slideshow presentations in their system and transfer it into a VR environment. In a frontal presentation or a combined setting, they enable presenters to show 3D content to the audience. The presenter takes over the active part of a presentation and the audience can see both the presentation and the presenter in both of their settings. It remains open how the audience could take the active part and make use of the interactive VR technology. Their evaluation is based on technical feasibility.

Steed et al. [St02] suggest the user role of a virtual presenter in their 'ante-room'. It is a virtual representation of the experimenter during a study or a demo and can give participants instructions. The presenter is visualized by a virtual puppet which is controlled from a desktop PC. A study shows that the users' sense of traversal could be reinforced when a transition was provided for the VR users that also included visually transitioning the presenter from the physical to the virtual world.

Price [Pr08] proposes UnrealPowerPoint as a new learning and teaching methodology. The author describes the usage of common PowerPoint slides within a computer game. Both learners and teachers can participate within the presentation by using a desktop PC interface and both user groups are represented by humanoid avatars. Single slides are not presented separately but can also be visualized simultaneously. According to the authors, the unreal slides have also additional functionalities that can go beyond common slides, which makes their concept more specific. Learners are granted the freedom to explore these slides non-linear way, which supported their educational purposes. Since the paper is oriented towards educational sciences, technical details are not considered here.

3 Presenter and Virtual Reality Integration

Both mixed and virtual presentations can integrate presenters in different ways using available devices. While the audience uses HMD-based VR to draw benefit from the

immersive 3D technology, presenters may (1) not interact with the virtual content at all and let the audience explore the VR scene, (2) interact through a desktop interface (*asymmetric integration*) or (3) be fully immersed using HMD-technology (*immersive integration*), as well. The connection of VR and presentation software can serve as basis for these integrations. We implemented three prototypes – one for each aspect to be evaluated:

- PowerPoint integration (*PP*) – This prototype serves as basis technology to connect presentation software and VR technology. As a use-case, it implements a presentation about forestry and forests for both a complete virtual and a mixed presentation.
- Asymmetric presenter (*AP*) – The AP prototype provides an asymmetric interface for presenters to interact with the virtual world and VR users. It uses a presentation about the solar system.
- Immersive presenter (*IP*) – The IP implementation represents an immersive interface for presenters. This prototype presents content about different sights on the world (e.g., Eiffel Tower and Chichén Itzá)

The PP implementation connects PowerPoint with a Unity game engine application. This feature is fundamental for mixed presentations, as game-engine-based VR is to be inserted at specific places during the slideshow. On the basis of using an existing PowerPoint presentation and enriching it with VR experiences, we identified three possibilities for realization. At first, the Microsoft Office Interop interface provides the possibility to react on the advancing of a slide from a running C# application, so that a running Unity application can process these events for example to switch to certain Unity scenes. But an identifier would be needed to support using different VR experiences at different points within the presentation. Secondly, APIs such as OpenXML provide functionalities to search slides with a specific layout for keywords. This mechanism can be used to trigger certain Unity events when a keyword is found on the current active slide. However, this option would restrict authors of the slides to comply to a necessary slide-layout/structure. At last, Interop allows to query the number of the current slide independent of the layout. This information enables a running Unity application to start the VR or turn it off when a specific slide of the presentation is set as active slide. Before the usage of this connection in a presentation, a mapping from VR experiences to PowerPoint slides must be performed during the authoring process. We implemented this third method for our prototype. Since PowerPoint is a software that can be opened multiple times, or even outside of our use case on the same machine, we use a controller script that knows both the path of the .pptx-file and the corresponding Unity application. This script establishes a connection between them and can be used to start the applications, as well. This ensures that VR is only triggered from the correct PowerPoint instance.

The PP prototype also included common PowerPoint slides within the virtual environment itself to implement a virtual presentation. We identified two methods to integrate slides within a Unity application with low additional effort for presenters/authors. At first, a digital

screencast or a webcam that records the physical presentation can be used to stream the slide content on a texture within the virtual environment. This requires additional soft- or hardware and can be difficult to set up, especially for webcam technology. As a result, visual quality may suffer and is directly dependent on the additional components. Secondly, each slide can be integrated within the VR as a separate image. These can be exported automatically with the mentioned Interop API. The images are used as textures on the virtual projection plane. They can be loaded by a running Unity application when they are exported to the Resources-folder within the assets. The drawback of this method is the absence of animations. However, this drawback is relativized as animation in slide lectures can have negative results on the learning outcome [MYJ09]. Therefore, we chose to implement an image-based workflow within our prototype. Necessary animations could be included in this method by using multiple slides to approximate the visual animation. We designed a separate virtual room with a projection plane for all slide adoptions (Fig. 1 top left) and changed the position of the VR users when a switch from a slide adoption to a VR experience (Fig. 1 bottom right and left) was intended. Events to change the positions or swap the slide images can be implemented using the Interop events as described. Presenters only have to interact with the original PowerPoint software in this prototype. All virtual rooms were implemented in one scene in this case (Fig. 1 top right). A simple webcam stream of the presenter was visualized during the slide adoptions to reflect the experience of a common slide presentation, where the presenter can be seen by the audience.

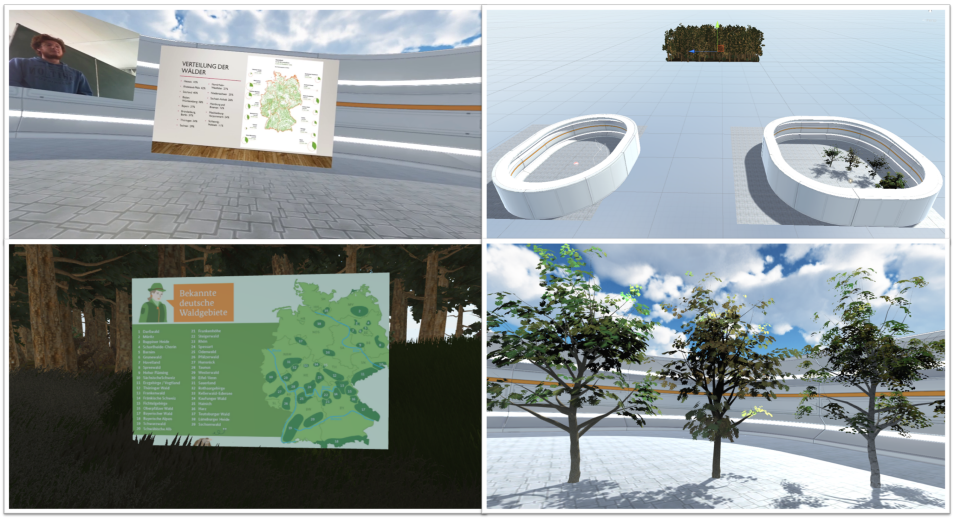


Fig. 1: Screenshots from the PP prototype presentation.

The AP prototype provides presenters a desktop PC interface to interact with the virtual environment during the VR experiences between the slides (Fig. 2). We provide presenters a top-down view on the virtual scene of the VR users. This view includes buttons for adjusting the participants' position, resizing predefined objects and system interactions (e.g.,

switching to next slides/VR experiences). The interface was implemented using Unity UI components and an orthographic camera.

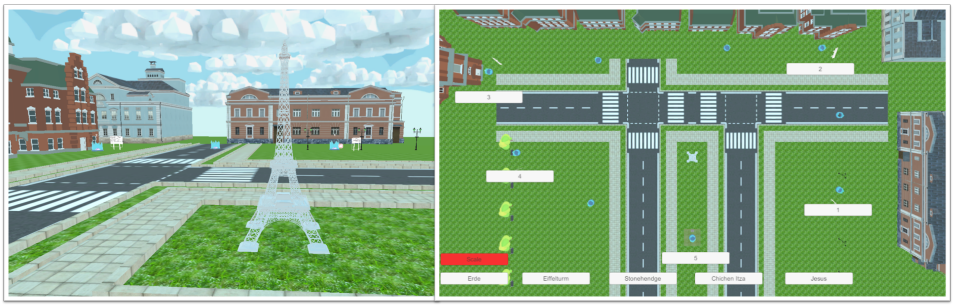


Fig. 2: Screenshots from the AP prototype presentation.

The IP implementation provides presenters an immersive interface to the VR experiences of the VR users (Fig. 3). Presenters are integrated within the virtual environment with a first-person view. It provides them similar interactions as the AP prototype, with the additional functionality to invite VR users to a quiz about the slide content and to rate slides. It also enables them to point with a laser pointer within the scene to guide the VR users through the scene. VR users are represented with a minimalistic humanoid avatar. The same representation is used for presenters, with the difference that they wear a crown to indicate that they have capabilities beyond the ones for VR users (Fig. 3 left). Avatars and interactions are implemented using Steam VR components.

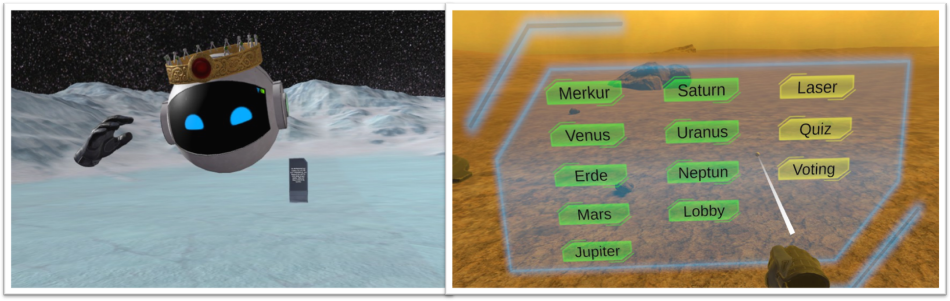


Fig. 3: Screenshots from the IP prototype presentation.

4 Evaluation

We evaluated the proposed presentation techniques and presenter integrations in three distinct user studies – one with each prototype (PP, AP and IP). We call the corresponding studies *PP study*, *AP study* and *IP study* respectively. Overall, the user studies involved 35 unpaid, voluntary and experienced participants. Their VR experience was captured

on a 0-3-point scale, where 0 means they had never used VR technologies and 3 means they regularly use VR. PP study involved 11 participants (3 female, Ø 26,7 years) with Ø 2.0 experience. AP study involved 14 participants (4 female, Ø ~21,5 years) with Ø 2.0 experience. IP study involved 10 participants (2 female, Ø 23,5 years) with Ø 1.9 experience. The procedure of each study took place as follows: At first, participants were welcomed and then informed about the topic of the study. They were briefly introduced to the user interface of the prototype and the VR hardware. Then an experimenter took the role of the presenter and gave a presentation to the participants using the prototypes PP, AP and IP. In the AP study, we divided a longer presentation into two conditions which were executed in a randomized order: 1) Participants experienced the VR parts of the presentations by themselves and 2) the participants were guided by the experimenter using the desktop interface of the AP prototype. Similarly, we divided the PP study into two conditions with randomized order: 1) The presentation was held using a mixed presentation methodology and 2) the presentation was held using a fully virtual presentation.

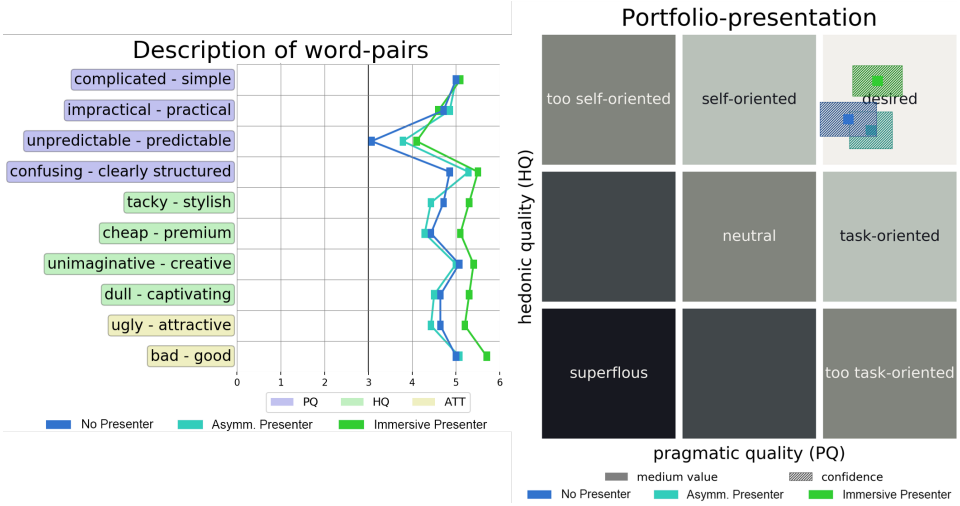


Fig. 4: AttrakDiff analysis [HBK03]. Left: Description of word pairs. Right: Portfolio presentation.

After each condition/presentation, the participants were asked to fill out a questionnaire, which was translated into their native language. The AP and the IP questionnaire consisted of an abbreviated version of the AttrakDiff questionnaire [HBK03]. The questionnaire for the PP study included eight questions that utilized a 7-point semantic differential scale: Q1: Would you like to stay longer in the virtual world? Q2: Did the virtual rooms help in understanding the content of the presentation? Q3: Were the texts, drawings, graphics easily recognizable? Q4: Would you like more interaction with the presentation? Q5: Did you find the HMD unpleasant? Q6: Did you find your way around the VR well? Q7: How did you feel about the different VR rooms? Q8: Would you recommend this type of presentation to others?

The results of the AttrakDiff questionnaires for the AP and IP studies are illustrated in Fig. 4. The charts show that all three presentations (no presenter, asymmetric presenter and immersive presenter) were perceived positively by our participants. Presentations with none or an asymmetric presenter integrations performed similarly well, with the difference that the asymmetric presenter was perceived more task-oriented and whereas the VR experiences without a presenter tends towards self-orientation. The immersive version was perceived best regarding hedonic and pragmatic qualities. In only one item it did not get the highest score, but the lowest, which is 'impractical-practical'.

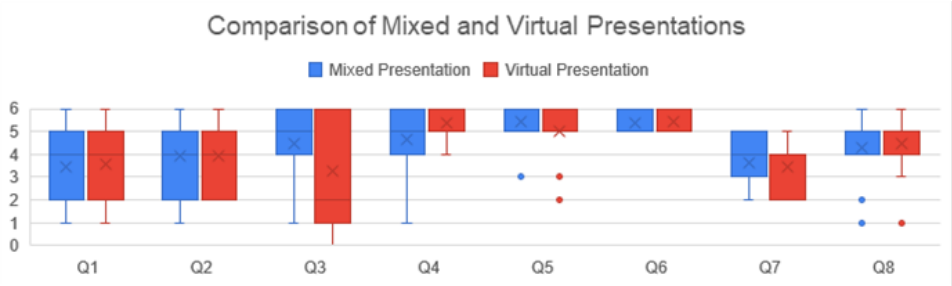


Fig. 5: Box- whisker plots comparing mixed (blue) and virtual presentations (orange) for Q1-Q8.

The chart in Fig. 5 illustrates differences between mixed and virtual presentations drawn from the PP study. The bar chart shows similar scores and distributions for Q1, Q2, Q5, Q6, Q7 and Q8. We performed Wilcoxon Signed-Rank tests [WW64] on the items Q3 and Q4 with a threshold for statistical significance of 5% to analyze further differences between mixed and virtual methodology. The tests could not confirm a statistical significant difference between the two conditions. The absolute differences for Q3 indicate that our participants preferred to view common slides in the physical world as they stated to recognize drawings, graphics and texts more easily there. Even though both presentations were perceived similarly positive, the scores for Q4 indicates that our participants expressed the desire to be able to interact with common slide adoption in VR more than it would be possible in the physical world.

5 Conclusions and Future Work

In this paper, we explored possibilities to integrate presenters when VR is used within the slideshow presentation. As a basis technology, we have shown how game-engine-based VR technology can be used to implement these concepts and how game engine VR can be connected to established slideshow presentation software, such as PowerPoint. Our user studies indicate that both virtual and mixed presentations were accepted by our participants and that an immersed presenter was preferred.

During our work, we noticed a lack of transitioning between physical slideshows and short VR experiences within mixed presentations. Current work targets rather extensive and elaborate transitioning to VR to improve the presence of VR users. As participants of mixed presentations may put on and take off VR HMDs frequently within a single presentation, such elaborate transitions could be disproportionate in relation to our VR experiences in-between slides. This will be addressed in future research directions. Furthermore, we evaluated the presenter integration from a VR user's perspective, but expert presenters must be included in future studies, too, in order to create the best possible experiences for both mentioned stakeholders of VR-enriched presentations.

Acknowledgements
The work is supported by the Federal Ministry of Education and Research of Germany in the project Innovative Hochschule (funding number: 03IHS071).

Literaturverzeichnis

[Fu01] Fuhrmann, Anton L; Prikryl, Jan; Tobler, Robert F; Purgathofer, Werner: Interactive content for presentations in virtual reality. In: Proceedings of the ACM symposium on Virtual reality software and technology. ACM, S. 183–189, 2001.

[HBK03] Hassenzahl, Marc; Burmester, Michael; Koller, Franz: AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität. In: Mensch & computer 2003, S. 187–196. Springer, 2003.

[HD18] Horst, Robin; Dörner, Ralf: Opportunities for Virtual and Mixed Reality Knowledge Demonstration. In: 2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct). IEEE, S. 381–385, 2018.

[HD19] Horst, Robin; Dörner, Ralf: Integration of Bite-Sized Virtual Reality Applications into Pattern-Based Knowledge Demonstration. In: Proceedings of the 16th Workshop Virtual and Augmented Reality of the GI Group VR/AR. Gesellschaft für Informatik, Shaker Verlag, S. 137–148, 2019.

[MYJ09] Mahar, Stephen; Yaylacicegi, Ulku; Janicki, Thomas: The dark side of custom animation. International Journal of Innovation and Learning, 6(6):581–592, 2009.

[Od15] Oda, Ohan; Elvezio, Carmine; Sukan, Mengu; Feiner, Steven; Tversky, Barbara: Virtual Replicas for Remote Assistance in Virtual and Augmented Reality. In: Proceedings of the 28th Annual ACM Symposium on User Interface Software Technology. ACM, 2015.

[Pr08] Price, Colin B: Unreal PowerPoint™: Immersing PowerPoint presentations in a virtual computer game engine world. Computers in human behavior, 24(6):2486–2495, 2008.

[St02] Steed, Anthony; Benford, Steve; Dalton, Nick; Greenhalgh, Chris; MacColl, Ian; Randell, Cliff; Schnädelbach, Holger: Mixed-reality interfaces to immersive projection systems. In: Immersive projection technology workshop. 2002.

[WW64] Wilcoxon, Frank; Wilcox, Roberta A: Some rapid approximate statistical procedures. Lederle Laboratories, 1964.

A Discussion on Current Augmented Reality Concepts Which Help Users to Better Understand and Manipulate Robot Behavior

Kai Groetenhardt[1]

Abstract:

For a safer, more trustful, and more dynamic collaboration, humans should understand and be able to manipulate the behavior of robots they are interacting with. Therefore, a way for a meaningful communication has to be established that takes place in a common perceptual space. One way to accomplish that is to use augmented reality (AR) in which the robot is able to display information for the human in 3D space, and the human can send commands to the robot using interaction methods provided by AR devices. In this work, a brief overview of AR concepts is given and discussed. They are divided into three categories: (1) understanding the movement of robots, (2) understanding the internal states of robots, and (3) manipulating robot behavior. Whereas (1) and (2) already show a number of promising approaches, and (3) is still in need for more innovative ideas.

Keywords: robot behavior; human-robot interaction; augmented reality

1 Introduction

Looking at active research in robotics, it is imaginable that robots will increasingly find their way into private households. Therefore, it seems important that humans without technical background are able to understand the behavior of robots and can control them. Breazeal et al. [Br01] identified an overlapping perceptual space as a key requirement for effective human-robot interaction, and Collett et al. [CM06] further explains that the perceptual space differs in input and output, which is illustrated in Fig. 1. Consequently, one obstacle is that humans and robots have different perceptual spaces, which just partially overlap. Not every action a human can perform is within the perceptual space of a robot, and a robot cannot reach every form of perception that is available to a human. Furthermore, humans and robots have differing conceptual models of the world; robots use several sensors to collect data of their surroundings and use various routines to interpret it. Sometimes they also need knowledge about the real world, which is provided by a knowledge framework. Those different perceptual spaces and different conceptual models of the world are reasons why understanding robot behavior can be difficult for humans.

[1] RheinMain University of Applied Sciences, Faculty of Design - Computer Science - Media, Unter den Eichen 5, 65195 Wiesbaden, Germany, kai.groetenhardt@hs-rm.de

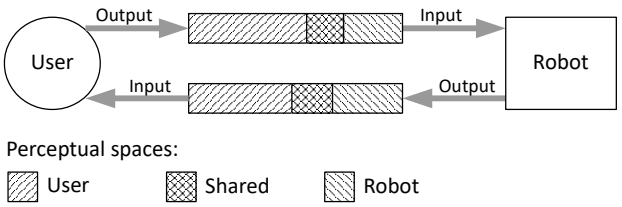


Fig. 1: An illustration showing the concept of the perceptual spaces of a robot and a human overlapping wherein a meaningful communication is possible. This is a rework of a figure from Collett et al. [CM06].

Augmented reality (AR) can be a tool to widen the overlapping parts of the perceptual spaces of humans and robots, and is therefore able to make understanding robots easier for humans. Robots can send information to AR devices, which then can be visualized for the humans to perceive, and humans can use the interaction possibilities of those devices to send commands to the robot.

The aim of this article is to give a brief overview of current work regarding human-robot interaction (HRI) with the focus on understanding and manipulating robot behavior using AR, and discuss possible future work. This article starts with the categorization, presentation and discussion of the state of the art in Section 2. In Section 3 follows a conclusion summarizing what was learned, and suggesting ideas for future work to hopefully enhance the topic of using AR for HRI further.

2 Discussing State of the Art

AR in combination with robotics is not a new topic. In fact, the literature review of Geen et al. [Gr08] about human-robot collaboration AR approaches contains articles dating back to the early 2000s addressing that topic. In the past, efforts using AR were hampered by limitations in the available AR head-mounted displays (HMDs), which often were custom-built. With a new batch of AR HMDs like the *Microsoft HoloLens* and the *Magic Leap 1* some limitations of the past were reduced or eliminated and enabled researchers to conceptualize and implement new AR concepts for HRI.

The following articles, describing the usage of AR for HRI, are divided into the three categories (1) understanding the movement of robots, (2) understanding the internal states of robots, and (3) manipulating robot behavior. Each category is limited to a maximum of two articles to not exceed the scope of this work. If a category contains two articles, they are chosen to be similar for a better comparability, but they are different regarding their goals and approaches. Those articles aren't depicted to the full extent, instead their presentment is

https://www.microsoft.com/en-us/hololens
https://www.magicleap.com/en-us

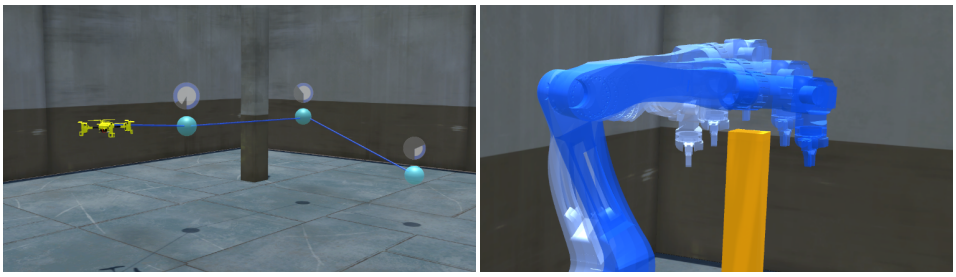
limited to the most relevant parts. To every article the motivation is stated, followed by the description of its AR concept, and concluded with the results of a user study, if available. At the end of each category, the concepts are being discussed.

2.1 Understanding the Movements of Robots

One defining characteristic of robots is their ability to move within the real world, sometimes in collaborative work spaces attended by humans. Unfortunately, robots don't necessarily have the ability to communicate their motion through gestures, gaze, or other social cues like humans. Here, two articles are presented showing possibilities to help the user understand the movement of robots with the help of AR.

Walker et al. [Wa18] argue that there are difficulties identifying when, where, and how a robot will move, which represents a primary challenge towards achieving safe and usable robotic systems. To tackle that problem, they introduce four concepts to indicate future movements of a flying robot using the AR HMD *Microsoft HoloLens*. The first concept, called *NavPoints* (shown in Fig. 2 (A)), adds virtual navigation points displayed as spheres into the 3D space. The spheres are connected through lines, which indicate in what order the robot will pass them. Above the spheres two radial timers are displayed, which show when the robot will arrive and when it will leave that position. The second concept, which is called *Arrow*, is similar, but a more minimal approach. An animated arrow shows the route the robot will take a few seconds in the future. As the arrow moves it leaves a line behind showing the path it was taking. The third concept is called *Gaze*, which augments the robot by a 3D model of an eye that is looking in the direction of travel. The fourth and last concept they presented is named *Utilities*. It is a 2D circular radar displayed at a corner of the user's perceptual space that shows the robots position relative to the user. Eventually, they compared the concepts by conducting a user study to see, among other things, how the displayed virtual imagery affected participant understanding of robot movement intent. The test showed that *NavPoints* ranked best followed by *Arrow*, *Gaze* and *Utilities*.

Rosen et al. [Ro20] indicate that a robot's movement intent can be shown on a 2D screen, but this requires the human to take their attention away from the robot's physical space to observe the display, which could be dangerous. Additionally, a 2D projection of a 3D motion can take time for a human to understand, requiring interaction to inspect different points of view. As a test scenario they chose a robot arm that performs a programmed movement with some objects nearby. The task for the human is to check if the robot will hit the objects before it even starts moving. To make that possible, a virtual 3D model of the robot arm is displayed multiple times along the planned path in 3D space visible through the AR HMD *Microsoft HoloLens*, shown in Fig. 2 (B). To have a reference, they implemented that same concept for a 2D screen with the possibility to move the virtual camera via mouse and keyboard. They compared both concepts and found that their AR system reduced the completion time of the task and increased the average accuracy of collision predictions.



(A) (B)

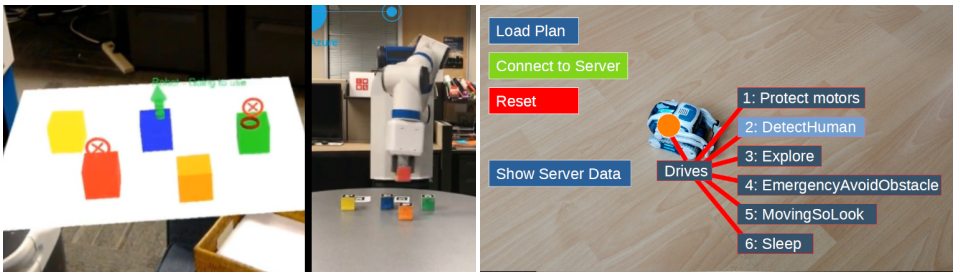
Fig. 2: Two different concepts communicating robot movement intent to humans. (A) shows the concept of view through the *Microsoft HoloLens* displaying the *NavPoints* concept from Walker et al. [Wa18] in which connected waypoints, the arrival, and the departure time of a robot can be seen. In (B) the concept of Rosen et al. [Ro20] is shown in which several steps of the planned movement are displayed in full size.

Comparing the concepts of Walker et al. and Rosen et al., it is apparent that they both show the robots motion intents, but target different scenarios. Walker et al. show where the robot will be located for users to adapt their own behavior towards the robot. Rosen et al. show the robot's future movement for the human to be able to intervene in the robot's behavior. It is imaginable to combine both concepts, but divide them into a planning and execution mode, which can be switched by the user. For the planning mode, the concept of Rosen et al. could be used to see detailed movements and to identify collisions. In execution mode, the concept of Walker et al. would show the path of the robot and when it will reach waypoints.

2.2 Understanding Internal States of Robots

To not only understand the movement of robots, but also the robots' decision-making process that leads to movements or other actions, an interface to the humans' perceptual space needs to be established. In this section, two articles are discussed showing a robot's plan of action via AR.

Chakraborti et al. [Ch18] cite the *Roadmap for U.S. Robotics* [Ch09] by saying "humans must be able to read and recognize robot activities in order to interpret the robot's understanding". They argue that attempts were made to accomplish the idea with natural language, but the state of the art limits the scope of such interactions, especially where precise instructions are required. To show an alternative, they communicate the intentions of a robot using AR to a collaborating human. Their setup consists of a robot that is tasked to stack colored boxes and a human who is equipped with a *Microsoft HoloLens* and has the ability to claim boxes through an AR interface. A virtual 3D model of boxes mirroring the boxes that are positioned in front of the robot is displayed for the human in 3D space, as can be seen in Fig. 3 (A). Those virtual boxes can be annotated by the robot to show what its intentions



(A) (B)

Fig. 3: Two different concepts to communicate a robot's intent to a human. (A) shows the concept of Chakraborti et al. [Ch18]. On the left hand side, the view through the *Microsoft HoloLens* and the mirrored model of the boxes in front of the robot can be seen. The virtual boxes are annotated with symbols that indicate the robot's plan. On the right hand side, the robot is displayed while executing its task of stacking boxes. In (B), the concept of the *Android* AR app of Rotsidis et al. [Ro19] can be seen that shows the robot annotated with its current plan represented as a hierarchical graph in which the active task "DetectHuman" is highlighted.

regarding those boxes are. The robot marks a box with a green upward pointing arrow to communicate that this box is the next one the robot is going to pick up. Also, boxes the robot intends to use in the future are marked with a circled red cross. The human on the other hand, has the ability to claim boxes for themselves even if the robot already has indicated to use them. In that case, the robot removes the mark at the corresponding box and chooses another one to complete its task. Unfortunately, there isn't a user study yet, but Chakraborti et al. announced their intention to conduct one.

Rotsidis et al. [Ro19] state that it's important for end-users to have a mental model of their robot that contains the capabilities and awareness of its limitations in order to trust it. Subsequently, through transparent decision-making of the robot it's possible for the users to adjust their expectations and forecast certain actions of the robot. The authors' attempt to tackle that challenge is based on an AR application running on an *Android* handheld device that shows the plan of the robot in form of an hierarchical graph. If the app detects the robot, it displays the graph next to it. The graph shows tasks the robot is able to perform and highlights the task the robot is currently executing, which can be seen in Fig. 3 (B). The user has the possibility to interact with the graph to see more or less information. They conducted a user study that showed the robot is perceived more alive, livelier, and friendlier with the app than without it.

Interestingly, Chakraborti et al. chose to show a virtual copy of the real objects and annotated them instead of annotating the real objects directly in AR. They didn't disclose why the went this way, but it would be interesting to find out if direct annotations could improve the usability. Comparing the concepts of Chakraborti et al. and Rotsidis et al., it is clear that both show the robots' plan, but, like in the previous Section 2.2, one is more detailed in its approach. Rotsidis et al. only show what task is being executed, whereas Chakraborti et al.

also show how the current task is being executed. Additionally, Chakraborti et al. developed a specific vocabulary to communicate the robots plan in form of annotations. In contrast, Rotsidis et al. communicate the plan via text arranged in a graph. Of course, a more detailed approach is not always the better choice since too much information could lead to problems of its own, for example by overloading the user or showing unwanted information. It needs to be determined in what scenario one concept is more suited than the other, or if a scalable solution combining the two concepts would be the better approach.

2.3 Manipulating Robot Behavior

If a robot needs to be taught how to execute a new task or change its behavior, a typical way to achieve this is to reprogram it by using text-based or even visual programming languages. But there are also other approaches like programming by demonstration (PbD), which is a field of research of its own that also includes AR solutions (e. g. [OK18], [Qu18]), or rather unconventional approaches like knowledge patching used in the following article.

Liu et al. [Li18] point out that machine learning methods have reached a remarkable level of effectiveness in specific tasks, but still have their limitations. For example, they lack interpretability of the knowledge representation, especially about how and why a decision is made, which plays a vital role in the scenarios where robots work alongside humans. In their system, they use interpretable knowledge represented by an And-Or-Graph (AOG) instead. Their setup consists of a robot with two arms and a *Microsoft HoloLens*, which can, among other things, display a 2D interface for the AOG in 3D space in front of the robot. The task to be solved is for the robot to open a medicine bottle with a lid that does not only have to be twisted but also pushed. The user starts with an AOG that describes how to open a normal bottle. The robot needs to be taught a push movement using PbD within the AR environment for it to be patched into the AOG by the human. To do the patching, the human can interact with the graph using hand gestures, comparable to mouse clicks and mouse drags, to remove and add nodes. The interface can be seen in Fig. 4, also a video [In] showing the whole process is available.

One could argue that the described part of Liu's et al. concept is a movement snippet manager that allows the user to combine little movements into a more complex motion, which is an interesting approach to not overburden the user. In their article, they changed the process of opening a bottle globally, which means even non-medical bottles get opened using that push and twist movement. It would be interesting to see this concept combined with some sort of a teachable object recognition system to be able to chose the opening process in a more targeted manner. Consequently, that would need to be implemented into the AR interface, which would present a new challenge. Researching robot behavior modeling via AR, unfortunately, revealed very little approaches outside the PbD field, which resulted in only one article in this section without something comparable, which leaves that topic open for more innovative ideas.

Fig. 4: The concept of Liu et al. [Li18] from two different perspectives extracted from a video [In] linked in their article. On the left hand side, the view trough the *Microsoft HoloLens* is displayed in which the human can see the AOG representing the knowledge to open a medicine bottle. On the right hand side, a third person's view without virtual elements is displayed.

3 Conclusion and Future Work

In this work, a brief overview of AR concepts dealing with robot behavior was given and discussed. The state of the art shows that there are already several concepts proven to be helpful in understanding robot behavior. Others look promising, but their effectiveness needs to be tested. The presented articles differ in their aims and level of detail in a way that makes them prone to be combined. Combinations of the described concepts could lead to improvements that would be interesting to see in future work. All things considered, using AR to understand robots seems to be a viable approach to further pursue.

In contrast, more accessible interfaces to manipulate robot behavior in AR seem to be a difficult endeavor. After a thorough research, only one article could be found that chooses an approach (at least partly) deviant to PbD. More ideas need to be developed and tested to see if AR is the right tool to manipulate robot behavior.

During the discussion, some suggestions for improvements were made, which could be conceptualized in more detail in future work. Especially the concept of Chakraborti et al. [Ch18] is an interesting candidate to pursue further to see if annotating real objects instead of the virtual copies of them feels more natural for the users.

Bibliography

[Br01] Breazeal, Cynthia; Edsinger, Aaron; Fitzpatrick, Paul; Scassellati, Brian: Active vision for sociable robots. IEEE Transactions on systems, man, and cybernetics-part A: Systems and Humans, 31(5):443–453, 2001.

[Ch09] Christensen, Henrik I; Batzinger, T; Bekris, K; Bohringer, K; Bordogna, J; Bradski, G; Brock, O; Burnstein, J; Fuhlbrigge, T; Eastman, R et al.: A roadmap for US robotics: from internet to robotics. Computing Community Consortium, 44, 2009.

[Ch18] Chakraborti, Tathagata; Sreedharan, Sarath; Kulkarni, Anagha; Kambhampati, Subbarao: Projection-aware task planning and execution for human-in-the-loop operation of robots in a mixed-reality workspace. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 4476–4482, 2018.

[CM06] Collett, T. H. J.; MacDonald, B. A.: Developer Oriented Visualisation of a Robot Program. In: Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-robot Interaction. HRI '06, ACM, New York, NY, USA, pp. 49–56, 2006.

[Gr08] Green, Scott A; Billinghurst, Mark; Chen, XiaoQi; Chase, J Geoffrey: Human-robot collaboration: A literature review and augmented reality approach in design. International journal of advanced robotic systems, 5(1):1, 2008.

[In] Interactive Robot Knowledge Patching Using Augmented Reality - YouTube. https://www.youtube.com/watch?v=AqjmIhKGGus. accessed: 07/04/2020.

[Li18] Liu, Hangxin; Zhang, Yaofang; Si, Wenwen; Xie, Xu; Zhu, Yixin; Zhu, Song-Chun: Interactive robot knowledge patching using augmented reality. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 1947–1954, 2018.

[OK18] Ostanin, M; Klimchik, A: Interactive robot programing using mixed reality. IFAC-PapersOnLine, 51(22):50–55, 2018.

[Qu18] Quintero, Camilo Perez; Li, Sarah; Pan, Matthew KXJ; Chan, Wesley P; Van der Loos, HF Machiel; Croft, Elizabeth: Robot programming through augmented trajectories in augmented reality. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 1838–1844, 2018.

[Ro19] Rotsidis, Alexandros; Theodorou, Andreas; Bryson, Joanna J; Wortham, Robert H: Improving robot transparency: An investigation with mobile augmented reality. In: 2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN). IEEE, pp. 1–8, 2019.

[Ro20] Rosen, Eric; Whitney, David; Phillips, Elizabeth; Chien, Gary; Tompkin, James; Konidaris, George; Tellex, Stefanie: Communicating robot arm motion intent through mixed reality head-mounted displays. In: Robotics Research, pp. 301–316. Springer, 2020.

[Wa18] Walker, Michael; Hedayati, Hooman; Lee, Jennifer; Szafir, Daniel: Communicating Robot Motion Intent with Augmented Reality. In: Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction. HRI '18, Association for Computing Machinery, New York, NY, USA, p. 316–324, 2018.

Herausgeber et al. (Hrsg.): Smart Systems for Better Living Environments,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2020 1

Requirements and Mechanisms for Smart Home Updates

Peter Zdankin[1] Oskar Carl[2] Marian Waltereit[3] Viktor Matkovic[4] Torben Weis[5]

Abstract: The interconnection of sensors and actuators of smart home devices creates dependencies that allow for ubiquitous services. These devices can be subject to transformative changes through software updates that might lead to unintended consequences. Users have no tools to predict the negative consequences caused by updating their smart home. In this paper, we address this problem and propose mechanisms that enable organized update planning in a smart home. We compare self-description standard approaches that allow reasoning about resulting functionality before updates are installed. Updating devices to their latest versions is not necessarily the best way to update smart homes, therefore we discuss multi-objective optimization in the update process. Finally, outsourcing functionality to external providers might reduce the complexity of certain tasks, but can also pose threats if the wrong tasks are offloaded.

Keywords: Smart Home; Longevity; Self-Description; Update Configuration; Edge Computing

1 Introduction

A smart home is composed of multiples devices from various vendors. Each vendor is producing software updates on his own, which leaves it to the user to select the best set of updates. Unfortunately, vendors cannot always consider all possible setups, and therefore faults can occur when a device is updated. Furthermore, downgrading software versions is not always possible, which means that a single unfortunate update can permanently impact the functionality of a smart home. Currently, users have no tools to master the update problem without risking damage to the system. Thus, it is important to investigate the update process and to propose solutions to compute the optimal update configuration automatically. In Section 3 we discuss how self-description of devices and services can principally solve the update problem. Then we discuss possible definitions of optimality of such a solution in Section 4. Finally, in Section 5 we compare solutions to the update problem relying on centralized services in the cloud with solutions that work locally in the smart home and discuss their impacts on ease of use and autonomy.

[1] University of Duisburg-Essen, Distributed Systems Group, Duisburg, Germany peter.zdankin@uni-due.de
[2] University of Duisburg-Essen, Distributed Systems Group, Duisburg, Germany oskar.carl@uni-due.de
[3] University of Duisburg-Essen, Distributed Systems Group, Duisburg, Germany marian.waltereit@uni-due.de
[4] University of Duisburg-Essen, Distributed Systems Group, Duisburg, Germany viktor.matkovic@uni-due.de
[5] University of Duisburg-Essen, Distributed Systems Group, Duisburg, Germany torben.weis@uni-due.de

2 System Model

The architecture of smart home systems may vary considerably between different implementations. Some systems are designed and deployed during the construction of a house and permanently deployed e.g. via bus systems in walls. While this is an interesting option, most homes require expensive renovation to become smart this way. An alternative to this is the usage of modular subsystems, for example, standalone lighting or heating systems, which can be bought individually and may offer their own separate platform, or integrate into bigger smart home platforms such as Amazon Alexa, Google Home, Samsung SmartThings or HomeKit. The benefit of modular smart home systems is a lower entry price for consumers, as they might start off with light bulbs or thermometers that can be controlled through already available smartphones or affordable hubs. In this paper, we focus on the latter, as the distributed and heterogeneous nature of these systems is more likely to break at some point during its lifetime, in comparison to a permanent solution installed during house construction. A smart home configuration can be described through the following system model:
*A **smart home** has a set of **devices** that are connected through a **platform**. Each device has a certain **software version** and a set of available **updates**. Each software version has a set of available **predefined services**. Devices can use services of other devices which creates **dependencies**. An **update configuration** is one of the finite states of the nondeterministic finite automaton NFA that can be constructed by using the configurations as states and connecting them using individual updates as transitions.*

During the lifetime of devices in a smart home, security issues or new features might require new software, which might alter the functionality of a device through for example a modified public API. These alterations might be intended, as part of a necessary change, or accidental because certain side effects were not considered. In any way, if such an update is installed, it will likely break an existing dependency. Currently, users cannot predict the changes that will be imposed on a smart home if a subset of its devices is updated simultaneously. Naive approaches that only consider individual updates for each update step, miss the broad picture because dependency problems can manifest after a certain set of devices have been updated already and rolling them back to previous versions may not always be possible.

3 Self-Description of Smart Home Devices

To enable interoperability, devices can describe their services to other devices, as was proposed in an architecture by Barbas et al. [Ve12]. Devices must be able to list all their currently available services, either through self-awareness or information included in the updates. By comparing the currently available services with the available services of an update, the differences can be computed. That way a system can predict possible faults in dependencies before they occur. However, comparing service definitions cannot capture unintended incompatibilities due to programming errors. Furthermore, most service

definitions are only syntactic, e.g. via an API definition, but they do not specify behaviour. Thus, an update might cause a change of behaviour without a change of its service definition. Two widely used approaches for service definition are *descriptive* or *prescriptive* standards.

3.1 Descriptive vs Prescriptive

A standard for service definitions that only regulates how a device can describe itself is a *descriptive standard*, because it allows vendors to describe their devices in their own terms. A recently finalized descriptive standard is the Web Of Things [Ka20; Ko20]. If devices need to communicate across different descriptive standards, a translation must be considered. However, McCool et al. have stated security concerns about purely descriptive approaches, such as scanning for door locks with known physical weaknesses [MR18].

A *prescriptive* standard does not only regulate the *how* but also the *what* of self-description. These standards prescribe how devices and services should be defined and vendors can use these terms for unambiguity. By using the same terms to define services, translation is not required anymore. A prescriptive standard defines all possible devices and all their services they could implement. However, a device can implement a subset of these services only. This restriction has benefits as well because it ensures compatibility across device vendors. Widely used examples are the smart home standards of Amazon, Apple, and Samsung[6].

3.2 Descriptive Standards Only

In a descriptive standard, each vendor can create its own namespace and definition for a certain type of device. This open approach is flexible and allows vendors to act quickly and independently. Unfortunately, it also allows vendors to reinvent service definitions, as they are incentivized to invent proprietary definitions in order to support new features. This can lead to multiple definitions of the same device type, thus creating redundancy as no incentive is given to find an agreement between vendors. As multiple definitions can coexist, communicating devices might need to translate between the definitions. Alternatively, a middleware could be introduced to translate between incompatible but equivalent descriptions. Over time, device vendors could update their own definitions in ways that impair interoperability, even to the point of defective device functionality. Therefore, we assume that descriptive standards alone are not suited for the evolving smart home ecosystem.

[6] Amazon Alexa Skill API Documentation: `https://developer.amazon.com/de/docs/smarthome/understand-the-smart-home-skill-api.html` (accessed May 6, 2020),
Samsung SmartThings: `https://smartthings.developer.samsung.com` (accessed May 6, 2020),
Apple HomeKit Accessory Protocol Specification: `https://developer.apple.com/support/homekit-accessory-protocol/` (accessed May 6, 2020)

3.3 One Prescriptive Standard

In a single prescriptive standard, each device type and service would be defined free of redundancy. While this allows for optimal interoperability, many practical problems require consideration. First of all, a regulatory body needs to be decided on to define this standard. This might cause long delays in the process of reaching consensus between the stakeholders, given the current interest in smart home applications.

If only devices that are defined in this standard can function in such a smart home, innovative new products that are not yet defined must also first pass the committee before being able to operate in a smart home. This does not just create obstacles for new products, it also enables competitors in the market to start production of these devices while they are being standardized. Small vendors that invent new smart home products might have problems to compete against bigger ones, due to the time loss introduced by the certification process. Due to this, we assume that a single prescriptive standard is too restrictive since the domain is evolving rapidly.

3.4 Hybrid Standards

Based on the above conclusions, we propose a hybrid solution between descriptive and prescriptive standards. Smart home devices are usually connected to a single platform and not part of multiple platforms simultaneously. Hence, a prescriptive standard per platform could have the desired flexibility, because platform vendors can innovate independently. To improve interoperability, this can be combined with a descriptive standard between all platforms. All devices – regardless of their respective vendors and platforms – would self-describe using that descriptive standard but adhere to the rules of the prescriptive standard given by the respective platform vendor. This way, each platform maintains optimal interoperability via prescriptive definitions. Since all prescriptive definitions are instances of a single descriptive language, it is still possible to translate between multiple platforms as discussed above. Furthermore, it is more likely that multiple prescriptive standards converge (at least partially), since they use the same descriptive language. Thus, they only have to agree on terms, but not on the syntax used to define services and devices.

4 Optimality of Update Configurations

When smart home systems are updated, the objective of current systems is to install the latest software on all devices, regardless of dependencies or other objectives. While this is one possible approach, there exist others such as dependency robustness or flexibility of the smart home system. These different goals are much harder to achieve because additional constraints must be upheld. Single-objective optimization might be able to yield acceptable

results to some degree. Users might want to hold onto their existing automations, which in some cases could be damaged by the latest software policy.

If the goal is only to ensure that no functionality is broken, the solution is to simply pick the latest versions that do not remove any functionality currently in use. However, as soon as an update removes functionality, the problem is expanded to multiple objectives: It requires weighting between the benefits provided by the latest software version and the convenience of keeping all functionality unchanged. Determining an optimal solution becomes even more complicated when new functionality is introduced at the cost of another one. Zitzler et al. have compared evolutionary algorithms as a means to search for possible solutions while considering conflicting optimization goals [ZT99]. Solving multiple-objective optimization is a subject that has been explored for a long time, and a large number of possible solutions have been found [MA04; ZT99]. In this problem space the aim is not perfect optimization, which is commonly impossible, but for *Pareto optimality*. It is an approximation in which multiple configurations might be considered equal. This requires a choice between these solutions to be made, which can be implemented in the form of user choice between policies like *feature stability* and *security*. Equivalent results according to optimality can be presented to the user, who is then required to choose *a posteriori* [Br08].

However, the devices available in a smart home are usually also constrained in terms of performance or power. This can render such approaches unviable in the smart home context. To amend this situation, the user choice should be made *a priori* [MA04].

Self-description of services allows considering dependencies in a smart home analytically to find optimal update configurations for new objectives before the updates are installed. As it is possible to know which changes will happen once a certain update configuration is chosen, configurations that disrupt dependencies can be discarded. Static analysis over the available services and possible dependencies can be enhanced through a dynamic approach that tracks which services are actually used in a smart home. This way, services that are actually used can be considered during the update process, while services that are not used can be completely ignored in the search for an optimal configuration.

5 Autonomy of Smart Homes

User management, device communication, update planning, and automated tasks of a smart home might not occur locally but on remote servers. If the autonomy of a smart home is constrained by outsourcing functionality to external providers, the smart home depends on the availability of these providers. This availability cannot be guaranteed for the lifetime of smart homes. Despite this, it represents a common mode of operation for smart home systems currently in use[2].

External services (located in the cloud) can offer resources to solve computationally intensive tasks, manage authentication and other security-critical necessities, and offer a

gateway for remote access. These benefits provide a large incentive to waive autonomy in a smart home and assume that external services and internet connection are always available. An external service can even use approaches like testing updates of devices against their specification to reason about the correct functionality. It might also perform updates on specific configurations under laboratory settings. Nevertheless, if smart homes target lifetimes of at least 10 years and multiple vendors are involved, it becomes likely that some external services are shut down. Possible threats against the longevity of smart homes exist and have happened before [Zd20a; Zd20b].

5.1 Remote Update Planning

Giving away autonomy can be dangerous if update planning is performed remotely. To find an optimal update configuration, a smart home system must transmit information about all devices in the smart home, their installed software, and used dependencies to the remote service. Transferring information about usage habits in the form of dependencies and usage patterns is privacy-invasive. Specifics on the software installed on devices can disclose vulnerabilities currently open for exploitation at a location. By abusing the knowledge of vulnerable software on smart devices, access to various parts of the smart home could be gained and used to invade the privacy of users or even risk the security of the entire local network. Remote update planning can also pose additional dangers, as devices can be advised to update to a vulnerable software version, which might open up an attack vector. Waiving autonomy in a smart home must be considered carefully, as the impact depends on the task performed remotely.

5.2 Local Update Planning

Autonomy in a smart home requires local resources to solve problems that would otherwise be resolved with the help of a centralized external service. Smart home platforms like OpenHAB[7] strive to be autonomous, at the cost of much higher complexity. The higher complexity is a burden for non-technical users who feel overwhelmed by the amount of work necessary to configure and maintain a completely autonomous system.

To find the optimal update configuration autonomously, a suitable device must be available in the smart home. We will refer to this as the *central device*. This device needs to find out which other devices are part of the smart home, how they are connected, and which dependencies exist. Furthermore, the central device should monitor which functionality is actually being used in the smart home installation. Thus, the central device must query devices in the network or it must query local hubs, at least one for each platform in use. The practical problem of this approach is that some platforms provide no API to query this

[7] OpenHAB Documentation: `https://www.openhab.org/docs/` (accessed May 6, 2020)

information. While it is usually possible to enumerate all devices connected to a hub, it is often not possible to query which functionality is being used, or how these devices are connected among each other.

As smart home devices are created by numerous vendors using various platforms, a single database for available updates does not exist in general. The central device must therefore either query all device vendors for updates, or it must rely on the user to make updates available locally. Automatically querying device vendors implies that an external service is being used again. This implies that the smart home is leaking information about the devices deployed and the software versions installed to a wide range of device vendors. From a privacy perspective, this might be even more questionable than transmitting this information to a platform vendor like Apple, Amazon or Google, because these are at least known to the user, while users are usually not able to judge the trustworthiness of an oversea device manufacturer.

Once the central device has information about all possible updates including the self-description for all updates, it can compute which services are added or removed if a specific update is installed. Through optimality criteria, many of these update configurations can be discarded and the most advantageous configurations can be obtained. As stated before, multiple optimal configurations might exist. In the worst case, the number of possible options is $O(2^n)$ where n is the number of updates because this is the count of all possible subsets of updates. Obviously, it is not reasonable to present these options to the user.

Therefore, we propose a policy selection like *latest version*, *conservative*, or *feature set* to capture the intent of the user. This policy can be used to further filter the set of Pareto optimal configurations. Finally, the central device performs an update path that is Pareto optimal and this complies best to the chosen policy. Further research is required to actually develop and evaluate such a system to gain insight into the feasibility of this approach.

6 Conclusion

We analyzed the update problem for smart homes. The currently dominant approach is potentially dangerous, as not enough measures are taken to prevent harmful updates from being installed in a live system. Furthermore, the update process can potentially leak critical information, which violates privacy and can pose security risks for the entire network, because it might disclose attack vectors. We have shown that service definitions are required for the update planning and discussed descriptive and prescreptive approaches and their practicality. While autonomy is a desireable property for smart homes, local update planning is more complex than update planning that relies on cloud-based services of platform vendors. Finally, we discussed what optimality means for update planning and concluded that optimality alone is not sufficient to select an update path, since multiple optimal configurations can exist. We proposed policies to capture the user intent and to finally select one optimal update path.

Optimality criteria are not exclusive to smart homes. Dependency management of software projects can encounter similar issues. Hence, research in one of these domains might benefit both. As devices become more powerful, it might be possible to pull services running in the cloud into the home network and to deploy them in containers. Thus, smart homes could use external services for convenience, but lack no features if the cloud becomes unavailable or the user does not want to use external services.

References

[Br08] Branke, J.; Miettinen, K.; Deb, K.; Sowiski, R.: Multiobjective Optimization, Vol. 5252 of Lecture Notes in Computer Science. Multiobjective Optimization 5252/, pp. 1–8, 2008.

[Ka20] Kaebisch, S.; Kamiya, T.; McCool, M.; Charpenay, V.; Kovatsch, M.: Web of Things (WoT) Thing Description, first Edition of a Recommendation, https://www.w3.org/TR/wot-thing-description/, W3C, Apr. 2020.

[Ko20] Kovatsch, M.; Matsukura, R.; Lagally, M.; Kawaguchi, T.; Toumura, K.; Kajimoto, K.: Web of Things (WoT) Architecture, first Edition of a Recommendation, https://www.w3.org/TR/wot-architecture/, W3C, Apr. 2020.

[MA04] Marler, R. T.; Arora, J. S.: Survey of multi-objective optimization methods for engineering. Structural and multidisciplinary optimization 26/6, pp. 369–395, 2004.

[MR18] Mccool, M.; Reshetova, E.: Distributed Security Risks and Opportunities in the W3C Web of Things. In. Jan. 2018.

[Ve12] Vega-Barbas, M.; Casado-Mansilla, D.; Valero, M. A.; López-de-Ipiña, D.; Bravo, J.; Flórez, F.: Smart Spaces and Smart Objects Interoperability Architecture (S3OiA). In: 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing. Pp. 725–730, 2012.

[Zd20a] Zdankin, P.: Longevity of Smart Homes. In: PerCom PhD Forum 2020: 18th Annual IEEE International Conference on Pervasive Computing and Communications PhD Forum (PerCom PhD Forum 2020). Austin, USA, Mar. 2020.

[Zd20b] Zdankin, P.; Waltereit, M.; Matkovic, V.; Weis, T.: Towards Longevity of Smart Home Systems. In: PerIoT 2020: 4th International Workshop on Mobile and Pervasive Internet of Things (PerIoT 2020). Austin, USA, Mar. 2020.

[ZT99] Zitzler, E.; Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation 3/4, pp. 257–271, 1999.

Complexity Analysis of Task Dependencies in an Artificial Hormone System

Eric Hutter[1], Mathias Pacher[1], Uwe Brinkschulte[1]

Abstract: The Artificial Hormone System (AHS) is a self-organizing tool able to allocate tasks in a distributed system. We extend the AHS in this paper by negator hormones to enable conditional task structures and provide a thorough complexity analysis of the resulting system. The analysis shows that the problem to decide if a given task A is instantiated at all respecting the negators is NP-complete.

Keywords: Aritifical Hormone System; negators; conditional task execution; complexity analysis

1 Introduction

We describe and analyze the decision problem NEGATOR-SAT occurring when using an *Artificial Hormone System* (AHS) [BP12] in this paper. The AHS is able to allocate tasks on a set of distributed processors without using a central instance and offers a high dependability of the task allocation. While the original AHS assumes a task model of independent tasks, we extend it here by assuming conditional dependencies between the tasks: E.g. a task T_1 can only be executed when another task T_2 is *not* executed. This allows to enable alternative task structures within the AHS.

Our contribution in this paper is twofold: (1) We shortly describe our extension of the AHS including the *negator hormones*. Their purpose is to enable conditional task structures. (2) Conditional task dependencies induced by negators make it hard to determine if a given task A can be instantiated at all. We call this decision problem NEGATOR-SAT and prove its NP-completeness. We end the paper by providing a transformation example of a satisfiable propositional formula to a task set using negators allowing to instantiate task A.

The paper is structured as follows: Section 2 presents the State of the Art in self-organizing systems. Section 3 gives an introduction to the original AHS while section 4 briefly explains our negator implementation. The complexity analysis of NEGATOR-SAT as well as an example are provided in section 5. Section 6 concludes the paper and describes future work.

2 State of the Art

IBM's *Autonomic Computing* initiative [LMD13] introduced so-called *self-x* properties such as self-configuration, self-optimization and self-healing. The MAPE-K loop was

[1] Goethe University, Frankfurt am Main, Germany, {hutter, pacher, brinks}@es.cs.uni-frankfurt.de

established to realize monitoring and analyzing of a system's behavior and to plan and execute actions controlling its behavior according to a knowledge base and user-defined goals. This loop has recently been adopted to establish self-explainable systems by using a MAB-EX loop (monitor, analyze, build, explain), see [Bl19]. The above mentioned self-x properties are also central to systems realized using *Organic Computing* concepts [TSM17]: Here, computer systems and embedded systems are constructed by incorporating concepts inspired by biological systems and their organization principles. This approach allows systems to dynamically adapt to changing operational conditions, realizing self-x properties like self-configuration or self-healing at run-time.

3 The Artificial Hormone System

The AHS' main purpose is to allocate tasks in a distributed system of processors, called *processing elements* or *PEs*. It is completely decentralized and has no single point of failure. In addition, it provides self-x properties such as self-configuration, self-optimization and self-healing and guarantees real-time bounds [BP12].

The AHS uses different kinds of hormones (which are short messages) to allocate the tasks. The main hormone types are eager values, suppressors and accelerators. Eager values indicate the suitability of a PE to take a task. As soon as a PE takes a task it sends suppressors for it. In this way, it tells the other PEs that it has taken the task: This is a life-sign on the one hand and it saturates the hormone balance on the other hand, thus limiting the number of allocated instances of this task. Accelerators are used to locate related tasks (i.e. tasks with communication relations or access to the same sensors or actors) nearby each other.

The core of the AHS is the hormone loop. Each PE iterates the loop, computing the hormone balance for each task. The duration needed by one hormone loop iteration is called a *hormone cycle*. In the *receive stage*, the hormones for each task are received. In the *compute and decision stage*, the suppressors received for a task are subtracted from its local eager value and the accelerators for this task are added. The result is the modified eager value which indicates the PE's current suitability to take this task. This computation is performed for each task. A PE's AHS instance then decides for a *single* task allocation per hormone loop iteration in order to allow the suppressors and accelerators to unfold their effect. In the following *send stage*, the PEs send eager values for all tasks (with the exception of an eager value that is 0) as well as suppressors and accelerators for all tasks they are currently executing. In this paper, we want to express conditional task relationships using the self-organizing AHS. A conditional task relationship means that a task T_1 can only be executed when another task T_2 is *not* executed. The concept is realized by special hormones called *negators* and allows to use alternative task structures in the AHS. This may be useful in a heterogeneous system of PEs. Details on the negators are described in section 4. Figure 2 gives a sketch of the hormone loop (already including the negators).

4 Conception

As mentioned before, our goal was to enhance the AHS by introducing the possibility to model conditional dependencies between tasks. To be precise, we introduced so-called *negator* relationships between tasks as visualized in Figure 1: Here, task T_j negates task T_i, meaning that task T_i cannot be assigned to any PE if T_j is assigned to a PE. Thinking in terms of a directed graph, this relationship can be expressed as the tuple (T_j, T_i).

Fig. 1: Negator relationship between tasks T_j and T_i: If T_j is assigned, T_i must not be assigned

This allows to express task dependencies: Suppose one PE_α can execute a task T realizing some functionality. Multiple other PEs cannot execute T but rather a set of tasks that realize the same functionality as T but with some kind of degradation, e.g. loss of precision. If PE_α is running, T's negator relationships to the other tasks prevent them from being instantiated. If PE_α fails, T is no longer available but the other tasks can be instantiated, keeping the functionality available.

4.1 Implementation

We implemented our concept of negator relationships that realize conditional inter-task dependencies by modifying the AHS' hormone loop as shown in Figure 2 (cf. [Br13] for

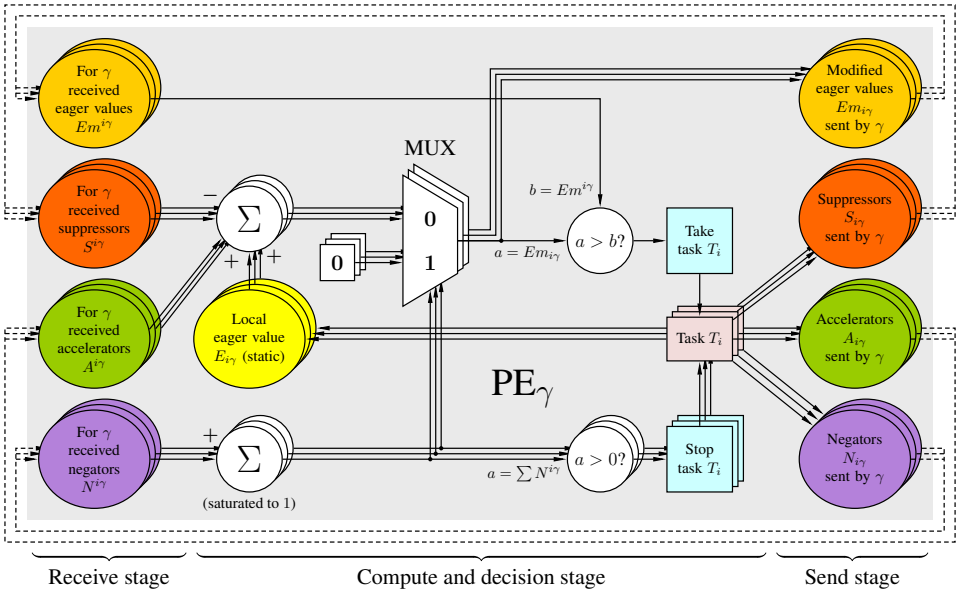


Fig. 2: Hormone loop with negators, running on PE_γ

information on the AHS' original hormone loop): We added an additional type of hormone, the so-called *negators*. If some task T_j is running and a negator relationship (T_j, T_i) exists, T_j will send a negator hormone to T_i during the hormone loop's send stage.

The received negator hormones are counted for each task. If at least one negator was received for some task T_i, two things happen: (1) T_i is stopped if it is running. (2) T_i gets blocked by forcing its modified eager value to 0, regardless of any suppressors or accelerators received for T_i. This prevents T_i from being assigned. If the negating task T_j is no longer assigned to any PE (e.g. because the PE it was running on failed), it won't send a negator for T_i any longer. This allows T_i's modified eager value to rise above 0 again, allowing T_i to be assigned again.

5 Theoretical Analysis

As has been seen in the previous section, the introduction of negators allows to model task dependencies, e.g. alternate sets of tasks to realize some functionality. However, negators also introduce new kinds of possible mistakes a designer can make during a system's design. Consider the following problem:

Definition 1 (NEGATOR-SAT)**.** Let $\mathcal{T}$ be a finite set of tasks and $\mathcal{N} \subseteq \mathcal{T} \times \mathcal{T}$ a set of negator relationships among those tasks.

The decision problem NEGATOR-SAT is now stated as follows: Given a task $A \in \mathcal{T}$, does a set of assigned tasks $\mathcal{V} \subseteq \mathcal{T}$ exist (with $T \in \mathcal{V}$ iff T is assigned to a PE) so that the following conditions are all satisfied:

(1) There is no task $T \in (\mathcal{T} \setminus \mathcal{V})$ that could be assigned to a PE even if all PEs had infinite computational resources,
(2) $\mathcal{V}$ is a stable task assignment, i.e. all negator relationships among tasks from $\mathcal{V}$ are respected,
(3) $A \in \mathcal{V}$, i.e. task A is assigned to some PE.

In simple terms, NEGATOR-SAT asks whether a stable task assignment exists so that A can be assigned to a PE. Condition (1) prevents the system's computational capacities from imposing any limits on such task assignment. Clearly, it can be regarded a design mistake if some task cannot be assigned to a PE at all. Thus, it should be checked if each task is assignable. However, it turns out that this seemingly simple problem is difficult to solve algorithmically:

Theorem 2. NEGATOR-SAT is NP-hard.

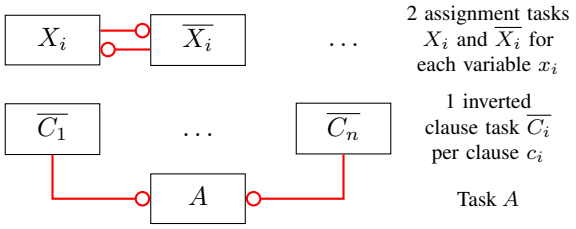
Proof. By reduction of 3-SAT to NEGATOR-SAT: We will show 3-SAT $\leq_p$ NEGATOR-SAT where $\leq_p$ denotes a polynomial-time reduction. If 3-SAT is reducible to NEGATOR-SAT in polynomial time, we can deduce that NEGATOR-SAT is at least as hard as 3-SAT. With 3-SAT being NP-complete, the NP-hardness of NEGATOR-SAT follows.

We will thus describe a transformation τ so that

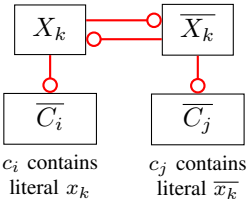
(i) τ can be computed in polynomial time w.r.t. the input length and
(ii) $f \in$ 3-SAT $\iff \tau(f) \in$ NEGATOR-SAT.

Let $f \coloneqq \bigwedge_{i=1}^{n} c_i$ with $c_i \coloneqq (l_{i,1} \vee l_{i,2} \vee l_{i,3})$ be a 3-SAT formula with $l_{i,j} \in \{x_k, \overline{x_k}\}$ for some k. We will transform f into a task set $\mathcal{T}$ with negator relationships $\mathcal{N}$ so that the transformed input $\tau(f) \coloneqq (\mathcal{T}, \mathcal{N}, A)$ is an instance of NEGATOR-SAT for the task $A \in \mathcal{T}$.

Construction of τ. The basic construction principle of this transformation is shown in Figure 3a. For each variable x_k occurring in f, we create two *assignment tasks* X_k and $\overline{X_k}$ that negate each other. This ensures only one of them can be assigned in a stable system, representing x_k's interpretation. Condition (1) ensures at least one of those assignment tasks is assigned per variable while condition (2) ensures that both cannot be assigned simultaneously.



(a) Basic construction principle



(b) Negator relationships between assignment and inverted clause tasks

Fig. 3: Construction principle of transformation from 3-SAT to NEGATOR-SAT

Additionally, we introduce one *inverted clause task* $\overline{C_i}$ per clause c_i. Thus, the resulting task set is

$$\mathcal{T} = \bigcup_{x_k \in \text{Variables}(f)} \{X_k, \overline{X_k}\} \cup \bigcup_{i=1}^{n} \{\overline{C_i}\} \cup \{A\}$$

where n is the number of clauses in f.

We will ensure that each inverted clause task $\overline{C_i}$ can (and per condition (1) will) be assigned iff f's interpretation does *not* satisfy the corresponding clause c_i by introducing negator relationships as follows (cf. Figure 3b):

- $(X_k, \overline{C_i}) \in \mathcal{N} \iff c_i$ contains the literal x_k and
- $(\overline{X_k}, \overline{C_i}) \in \mathcal{N} \iff c_i$ contains the literal $\overline{x_k}$.

Finally, the following negator relationships ensure that task A can (and, again, per condition (1) will) be assigned iff no inverted clause task $\overline{C_i}$ is assigned (and thus, all corresponding clauses c_i are satisfied): $(\overline{C_i}, A) \in \mathcal{N}$ for all $1 \leq i \leq n$.

For simplicity, we require that each clause in f consists of exactly three literals. Note that this restriction does not change the problem's complexity as a clause can be padded to exactly three literals by repeating one of its literals.

See section 5.1 for an example of this construction.

τ is a polynomial-time reduction. We now need to show that above claims (i) and (ii) hold for τ, i.e. that τ is indeed a reduction of 3-SAT to NEGATOR-SAT.

(i): Polynomial time: It is easy to see that a formula f with n clauses and v different variables (with $v \leq 3n$) can be transformed in polynomial time w.r.t. f's length: We only need to construct $2v$ assignment tasks, n inverted clause tasks and the task A. This sums to $2v \quad n \quad 1 \leq 7n \quad 1$ tasks which is polynomial in the input formula's length.

Additionally, we construct $2v$ negator relationships for mutual exclusion of X_k and $\overline{X_k}$, $3n$ negator relationships between X_k resp. $\overline{X_k}$ and $\overline{C_i}$ and n negator relationships between $\overline{C_i}$ and A. This totals at $2v \quad 4n \leq 10n$ relationships which is also polynomial in the input formula's length.

(ii), part 1: $f \in$ 3-SAT $\Rightarrow \tau(f) \in$ NEGATOR-SAT:

Proof. Since $f \in$ 3-SAT, there must exist a satisfying interpratation $\mathcal{I}$: Variables$(f) \to \{0, 1\}$. Thus, consider the set $\mathcal{V} \subseteq \mathcal{T}$ of assigned tasks given as follows:

- $A \in \mathcal{V}$,
- for each inverted clause task $\overline{C_i}$: $\overline{C_i} \quad \mathcal{V}$,
- $X_k \in \mathcal{V} \iff \mathcal{I}(x_k) = 1$ and $\overline{X_k} \in \mathcal{V} \iff \mathcal{I}(\overline{x_k}) = 1$.

Due to τ's construction, it is easy to see that $\mathcal{V}$ satisfies conditions (1) to (3) as given by Definition 1:

(1) All tasks in $(\mathcal{T} \setminus \mathcal{V})$ have an inbound negator link coming from an assigned task, thus no additional task can be instantiated.
(2) All negator relationships are respected: No two tasks from $\mathcal{V}$ share a negator relationship.
(3) A is assigned. ◇

(ii), part 2: $\tau(f) \in$ NEGATOR-SAT $\Rightarrow f \in$ 3-SAT:

Proof. Let $\mathcal{V} \subseteq \mathcal{T}$ be a set of assigned tasks so that conditions (1) to (3) as given by Definition 1 are satisfied. Thus, task A must be assigned. Therefore, per condition (2), no inverted clause task $\overline{C_i}$ can be assigned. Thus, at least one assignment task per inverted clause task $\overline{C_i}$ must be assigned (else, $\overline{C_i}$ would have to be assigned per condition (1)). Additionally, per condition (2), for each assignment task X_k resp. $\overline{X_k}$, the inverse assignment task $\overline{X_k}$ resp. X_k cannot be assigned.

This allows to construct an interpretation $\mathcal{I}$ for f so that $\mathcal{I}(x_k) = 1 \iff X_k \in \mathcal{V}$ and

Note that—since $\mathcal{I}$ satisfies f—at least one literal is satisfied for each clause c_i, thus the corresponding inverted clause task $\overline{C_i}$ is not assigned. Since all inverted clause tasks are *not* assigned, A can (and per condition (1) will) be assigned to some PE.

$\mathcal{I}(\overline{x_k}) = 0 \iff \overline{X_k} \in \mathcal{V}$. In addition, $\mathcal{I}$ must satisfy f: Suppose $\mathcal{I}$ would not satisfy f. Then, there must be a clause c_i in f so that $\mathcal{I}$ does not satisfy any of its literals $l_{i,j}$. Due to τ's construction, this would mean that the inverse clause task $\overline{C_i}$ must be assigned per condition (1) which forbids A's assignment per condition (2). ⋄

Final remarks. Since τ can be constructed in polynomial time w.r.t. the input length, it follows that τ is indeed a polynomial-time reduction from 3-SAT to NEGATOR-SAT. Thus, NEGATOR-SAT is at least as hard as 3-SAT. Since 3-SAT is NP-complete, the NP-hardness of NEGATOR-SAT follows. □

However, NEGATOR-SAT is not only NP-hard, but also complete for NP:

Theorem 3. NEGATOR-SAT is NP-complete.

Proof. Let $(\mathcal{T}, \mathcal{N}, A)$ be an input consisting of a set of task $\mathcal{T}$, a set of negator relationships $\mathcal{N}$ and a task $A \in \mathcal{T}$. A nondeterministic Turing machine can now nondeterministically select a subset $\mathcal{V} \subseteq \mathcal{T}$ of assigned tasks and then deterministically check that conditions (1) to (3) from Definition 1 are all satisfied:

(1) This condition is satisfied if, for each $T \in (\mathcal{T} \setminus \mathcal{V})$, there is a negator relationship $(T', T) \in \mathcal{N}$ so that $T' \in \mathcal{V}$. However, this can be checked in $\mathcal{O}(\text{poly}(|\mathcal{T}|, |\mathcal{N}|))$ time.
(2) This condition is satisfied if, for each $T \in \mathcal{V}$, there is *no* negator relationship $(T', T) \in \mathcal{N}$ so that $T' \in \mathcal{V}$. This can also be checked in $\mathcal{O}(\text{poly}(|\mathcal{T}|, |\mathcal{N}|))$ time.
(3) This condition is satisfied if $A \in \mathcal{V}$ which can be checked in $\mathcal{O}(\text{poly}(|\mathcal{V}|))$ time.

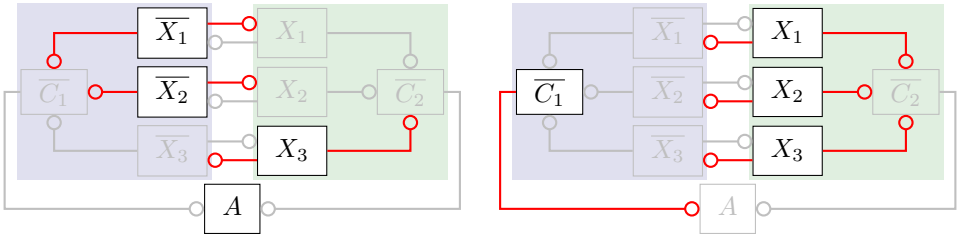
The Turing machine shall accept the input iff all three conditions are satisfied.

Since, after nondeterministically guessing $\mathcal{V}$, the verification can be performed in polynomial time w.r.t. the input length, it follows that NEGATOR-SAT ∈ NP. Together with Theorem 2, it follows that NEGATOR-SAT is NP-complete. □

This shows the power introduced by negators: Unless P = NP holds, it is not possible to decide in deterministic polynomial time whether a given task A can be assigned to a PE at all (when requiring a stable task assignment in which no additional tasks can be assigned).

5.1 Example of Construction

Figure 4 shows the construction result $\tau(f)$ for the formula $f = (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3}) \wedge (x_1 \vee x_2 \vee x_3)$. Note that f is satisfiable and hence $f \in$ 3-SAT. It is easy to see that assigning either X_i or $\overline{X_i}$ for $i \in \{1, 2, 3\}$ allows assignment of A iff the assignment corresponds to a satisfying interpretation of f.



(a) With task assignment corresponding to satisfying interpretation of f

(b) With task assignment corresponding to unsatisfying interpretation of f

Fig. 4: NEGATOR-SAT instance constructed from 3-SAT instance $f = (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3}) \wedge (x_1 \vee x_2 \vee x_3)$

6 Conclusion

We presented an negator extension for the AHS middleware in this paper. Negators enable conditional task execution in the AHS which is useful in a heterogeneous processor system. The use of negators impose the problem to determine if a given task A can be instantiated in the context of a stable task allocation in the overall system. We called the problem NEGATOR-SAT and proved its NP-completeness. Future work will consider the negators' impact on the AHS' real-time bounds and the stability of task allocations. This is important as it is simple to see that negators can generate oscillating task allocations.

Bibliography

[Bl19] Blumreiter, Mathias; Greenyer, Joel; Garcia, Francisco Javier Chiyah; Klös, Verena; Schwammberger, Maike; Sommer, Christoph; Vogelsang, Andreas; Wortmann, Andreas: Towards Self-Explainable Cyber-Physical Systems. In: 22nd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion, MODELS Companion 2019, Munich, Germany, September 15-20, 2019. pp. 543–548, 2019.

[BP12] Brinkschulte, Uwe; Pacher, Mathias: An Agressive Strategy for an Artificial Hormone System to Minimize the Task Allocation Time. In: 15th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops, ISORC Workshops 2012, Shenzhen, China, April 11, 2012. pp. 188–195, 2012.

[Br13] Brinkschulte, Uwe; Pacher, Mathias; von Renteln, Alexander; Betting, Benjamin: Organic Real-Time Middleware. In (Higuera-Toledano, M. Teresa; Brinkschulte, Uwe; Rettberg, Achim, eds): Self-Organization in Embedded Real-Time Systems, pp. 179–208. Springer New York, New York, NY, 2013.

[LMD13] Lalanda, Philippe; McCann, Julie A.; Diaconescu, Ada: Autonomic Computing - Principles, Design and Implementation. Undergraduate Topics in Computer Science. Springer, 2013.

[TSM17] Tomforde, Sven; Sick, Bernhard; Müller-Schloer, Christian: Organic Computing in the Spotlight. arXiv:1701.08125 [cs], January 2017.

Unified Approach to Static & Runtime Verification

Olga Dedi, Andreas Werner, Robert Kaiser, Reinhold Kroeger [1]

Abstract: Formal verification of a system's functional and non-functional properties is often regarded as the ultimate way to achieve the highest levels of trust as demanded for today's dependable systems. However, static verification, though sound in theory, is often impractical given the ever-increasing complexity of software and the non-deterministic nature of some mechanisms of the underlying hardware architecture. We argue that by supplementing static verification with runtime verification, a high level of trust can be achieved. In this paper, we report on an ongoing effort for tool-supported verification of functional and non-functional properties by combining static and runtime verification techniques.

Keywords: static verification; runtime verification; OS microkernel; SPARK; WCET; AQUAS

1 Introduction

Today's systems become more and more complex, and even domain experts are sometimes in doubt regarding their correct behaviour in rare / non-standard situations. Especially embedded systems incorporate increasing functionality and have to deal with a wide spectrum of sensors and actors interacting with the environment. Real-time properties requiring a guaranteed reaction of the system within a given limited time window are often associated as well, and safety is taken for granted by customers all-the-time when using them.

Furthermore, these critical systems often have to face uncertainty which may originate from unknown device configurations at design time or unforeseen changes of the environment during operation. Uncertainty may also exist in control algorithms. For example, to guarantee a safe behaviour of trained AI algorithms in previously unseen situations is inherently difficult, if not impossible.

Under these conditions it is a complex and highly responsible task for developers to deliver a high level of trust in the developed software. This is commonly achieved through certification. To certify software for a given Safety Integrity Level (SIL), or ASIL level in the automotive systems context, it has to be thoroughly tested, specific models and analysis methods have to be used up to a formal, mathematical verification of required system properties. Today, all this has to happen before the system is actually used.

Due to the described complexity of current and future systems we do not believe that a full static verification and validation at design time is possible any longer to deliver the necessary

[1] RheinMain University of Applied Sciences; firstname.lastname@hs-rm.de

trust. Instead, we have started to work on a methodology which distinguishes between verification activities carried out at design time and those at runtime. In summary, at design time static verification takes place, i.e. specified functional as well as non-functional/timing system properties are formally proven to the highest possible degree for a reasonable maximal effort. For properties which cannot be proven statically, sufficiently strong monitors are generated which are executed at runtime to monitor correct system behaviour. In the undesired case of detecting a property violation at runtime, the underlying system architecture is prepared to reconfigure the application to ensure acceptable behaviour. This adaptivity has to be supported by the application design. In total, a trusted self-adapting application seems to be reachable.

In the following chapter 2, the functional properties and their verification are considered. In chapter 3, timing is taken into account as this is the most important non-functional property regarding real-time behaviour. In both chapters, the current status of work is described. The paper closes with a summary and conclusion. The considered use case and examples are taken from the AQUAS EU project [20b] to which we contribute.

2 Functional Verification

Functional verification activities are discussed in detail in this section.

2.1 Static Verification

Typically, in the design phase a system model is constructed which specifies the overall system and its functional and non-functional properties and constraints. To simplify the designer's work, rather than using the industry standards SysML and OCL directly, we propose to use a DSL (Domain Specific Language) with an expressiveness to target the class of applications in mind. Such standard models shall then be generated from the DSL.

This generated system model will then be semi-automatically transformed into a set of views by exporting and transforming the model. Each view provides information concerning a specific aspect of the system. For the functional view and its verification, the SysML model is transformed into a SPARK interface definition, and functional OCL constraints into SPARK contracts with pre- and post-conditions (see Fig. 1). In addition to specified application constraints, other conformance rules, e.g. standards, company rules etc., can be taken into account as well, resulting in additional contracts or contract restrictions.

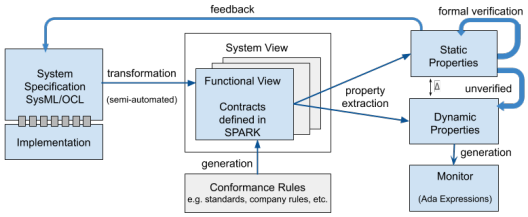


Fig. 1: Functional Verification During the Design Phase

During the design phase, the verification of selected functional properties can be carried

out solely at the contract/API level by taking advantage of a hierarchical system structure, deducing higher-level contracts from lower layers and assuming elementary contracts as facts [Le80]. Thus, it is possible to provide early feedback regarding correctness of the system specification. Later, the assumed facts have to be proven by verification of the corresponding implementation.

2.2 Runtime Verification

Runtime verification is regarded as the discipline of computer science dealing with techniques to monitor systems during runtime in order to detect violations of given correctness properties [LS09]. More recent research, however, also considers controlling the system via feedback as belonging to runtime verification as well [LS09; Ru16]. Augmenting a functional system with a corresponding management/control system allows for autonomous behaviour of the system during operation.

As previously explained, not all contracts can be statically verified at design time. Our methodology extends the verification activities to the system runtime. The amount of possible static analysis and needed dynamic verification depends on the specific system and the complexity of its constraints. During the design phase, our methodology aims to separate as many properties or partial predicates as possible that can be verified statically and to automatically derive the complementary properties or predicates that have to be ensured/verified at runtime. Concerning the application level, necessary monitors will be generated from the unverified SPARK contracts during the design phase and executed by the runtime architecture.

In principle, monitors may be associated with all critical parts of the system which may be a source of uncertainty, like the application itself, the operating system and the hardware, but also the environment, especially when considering embedded systems. If a monitor assertion fails, an event is signalled to the runtime system (see Fig. 2).

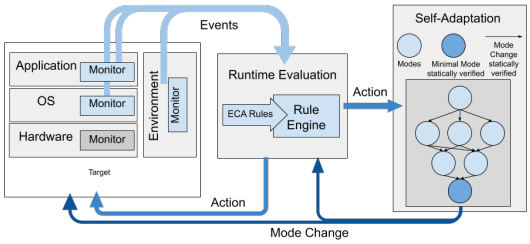


Fig. 2: Architecture for Runtime Verification

Thus, by extending the verification process to runtime, the methodology supports finding a manageable approach to verification of complex systems also covering uncertainty. This is especially important when static verification is based on pre-conditions, whose outcomes can only be evaluated during runtime, e.g. if the pre-conditions are depending on user input or environmental factors.

As a disadvantage, incomplete static verification of system properties may result in property

violations at runtime. Thus, provisions for events originating from failing runtime verification may be required. This is considered in the following section.

2.3 Adaptation Architecture

This section proposes a mechanism in the system architecture, in cases where runtime verification fails and halting the system is not acceptable.

In the runtime system, signalled events will be received by an Event Condition Action (ECA) rule engine. Based on an application-dependent statically defined rule set, appropriate actions can be taken for detected runtime violations. Such actions can lead to a simple reconfiguration, like changing control parameters, or may require a more complex adaptation of the system.

As a basis for adaptation, the well-known concept of operational modes will be used. At each point in time, the system runs in a certain mode determining the provided functionality. At design time, a mode change graph will be developed, defining the set of operational modes and allowed changes. Also, the mode change algorithm must be statically verified for correctness. At runtime, mode changes may take place in response to signalled events. Thus, the system is guaranteed to be able to enter a well-defined state in case of a runtime verification failure, thus supporting graceful degradation.

In case of safety-critical systems, not every mode must ensure safety. A so-called "Minimal Mode"providing a safe state for the system is assumed to exist, whose functionality is statically verified. Due to the verified mode change algorithm, the minimal element can be reached in any case. Thus, a minimal level of service is ensured under all conditions.

2.4 Current Status

For initial evaluation, the functional part of the methodology was applied to the design and implementation of a queueing system and its use inside the scheduler of our microkernel Marron. Marron was developed by our group to serve as a template for a future verified microkernel. Besides scheduling, it features strict separation between user and kernel space, interrupt handling and inter-task communication. The queues where designed directly in SysML and constraints were defined in OCL. An appropriate DSL will be specified later, when more experience has been made. The use of SysML/OCL features was manually restricted to fit SPARK 2014 capabilities, and the transformation was done manually for now. The specification was verified based on the SPARK API. Necessary facts that have to be verified by the implementation were indicated by the `assume` pragma. Finally, the implementation was verified separately.

Based on the verified queueing system, a graduate student developed and verified the Marron scheduler in SPARK. The student had no prior experience with Ada, SPARK or formal verification in general. The goal of this experiment was also to evaluate the efficiency of our approach by measuring the effort needed to develop a verified operating system component. The student spent a total effort of 450h over a course of six months, including literature work,

project management, documentation, etc. He was able to verify 216 out of 224 verification conditions (VCs). The measured efforts spent on implementation and verification, as well as the relative effort in minutes per line of code are shown in Table 1.

	hours	**loc**	**ratio (min/loc)**
implementation	62.25	296	4.7
verification	95.25	330	19.31

Tab. 1: Effort analysis for implementation and verification of an OS scheduler in SPARK.

Regarding runtime verification of the remaining eight statically unverified VCs, the pragma `Assertion_Policy` was simply used to execute all contracts as assertions during runtime, but no exhaustive tests were yet carried out, nor have the predicates for runtime verification been optimized. The adaptation architecture has not been implemented yet.

3 Non-Functional Verification

Non-functional properties in context of this work refer to the timing behaviour of a system. In order to reason about the timing of a system, a notion of time, a specification of timing properties and constraints and also a verification environment are needed. To formally verify timing behaviours means to find a mathematical proof showing that the system will behave, temporally, as specified under all conditions.

3.1 Modelling and Verification of Timing Behaviour

There is a long history of formal languages that can be applied to specify diverse aspects of timing behaviours [Wa04]. The classical event-oriented temporal logics such as *Linear-Time Propositional Temporal Logic (LPTL)* or *Computation Tree Logic (CTL)* only model the temporal order of events (e.g. *before*, *after*, *always*, *never*, *eventually*, ...), but do not provide a notion of real, physical time, as is needed to model real-time systems. One possible language to start with is called *Timed CTL* (TCTL*)*. It is the foundation of the UPPAAL verification framework [20d], which is based on model checking techniques. However, such an approach often leads to state explosion or undecidability problems, making it impractical for complex real systems. Our method wants to avoid these limitations by keeping the human in charge as director for the proof, aided by semi-automated theorem provers like Coq [MT18].

3.2 WCET Estimation

In order to check whether a real-time program temporally behaves as specified in a model, it is necessary to know the actual execution times of relevant program sections, and to associate states in the model with program states.

The actual execution time of any piece of code can vary each time the code is executed. In real-time systems, the *Worst Case Execution Time* (WCET) is a commonly used concept to abstract from these variations. A good overview of the classification and techniques for WCET calculation can be found in [Ca19]. The paper differentiates between static, measurement-based or a combined approach to determine the WCET, each in a deterministic or probabilistic variant. The static deterministic approach, called *Static Deterministic Timing*

Analysis (SDTA), uses symbolic execution on an accurate model of the hardware. This approach is only practical for systems which are amenable to modelling, but it is well trusted and well established in industry. However, as today's multicore hardware achitectures frequently do not fulfil the assumptions made for their modelling, newer methods combine these approaches with probabilistic ones such as Extreme Value Theory (EVT). These are subject of ongoing research [Ca19].

3.3 Current Status

We evaluated the AbsInt tool for SDTA named aiT [20a] with a small application from the AQUAS space usecase and compared the WCET bounds with real measurements. The application cyclically receives a message from a serial communication interface, encrypts the message and sends it out over another serial communication interface. This application was executed on top of two different operating systems: (1) the library-based RTEMS kernel [20c] designed for microcontrollers and mostly used in avionic and space systems, and (2) our own microkernel Marron equipped with a small RTEMS adaptation layer which currently only implements the interface subset needed by the application. As target hardware, we use the TI TMS570 microcontroller with two ARM Cortex-R4 in lock-step mode. Marron was originally designed to run on Cortex-A multicore processors, but these more complex processors are not supported by aiT. The ARM Cortex-R4 was the smallest processor on which our system runs without modification.

The results of the WCET analysis are presented in Table 2. Annotations of the source code using the AbsInt AIS2 language are instructions to the AbsInt tools directing the static analysis. Declaring routines as infeasible means that the developer is sure they are never executed by the analysed code (e.g. POSIX and kernel error handling) and thus, they do not contribute to the estimated WCET.

	AIS2 code	**Infeasible routines**	**Analysis times**	**aiT WCET**	**Max Exec. Time**
RTEMS	659 loc	36	40 s	0.659 ms	0.321 ms
Marron	476 loc	23	4 s	0.580 ms	0.247 ms

Tab. 2: WCET and Measurement results

We also compared the WCET estimations with real measurements of the execution times of the application. The execution time measurement starts upon reception of the first byte and ends when the last byte is sent out, thus being the same code sequence as for the WCET analysis. Both versions were compiled for ARM Thumb Code and with optimisation set to -Og (i.e. weak, "debug-friendly" optimisation). For the measurements we use the ARM Performance Monitoring Unit which measures the number of elapsed CPU cycles. All measured data was buffered in SRAM, as SRAM accesses are deterministic on the used platform. The buffering overhead was determined in a separate measurement and subtracted for compensation. For the static analysis, the buffering overhead was excluded by appropriate annotations. The measurement overhead itself was measured to be 60 CPU cycles based on 10 x 1000 single measurements. The aiT tool estimated the WCET for one

measurement to be 77 CPU cycles. The density function of the measured execution times for

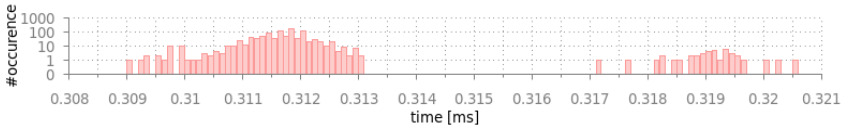


Fig. 3: Measurement Results for RTEMS

the RTEMS version is shown in Fig. 3, for the Marron version in Fig. 4. RTEMS execution times below 0.3131ms are caused by the communication through a software queue between the application and the receiver interrupt. The execution times above 0.317ms are the result of the same communication delayed by the system timer interrupt. This interference does not appear in the WCET analysis, as the tools do not model task communication or processor interrupts. The Marron RTEMS adaptation layer does not provide software queues for

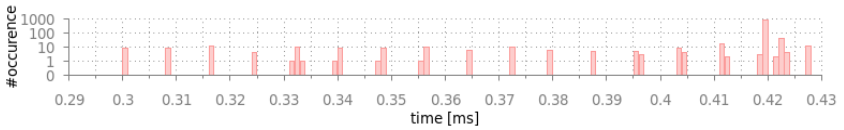


Fig. 4: Measurement results for Marron

signalling, the only buffering mechanism used are the hardware FIFOs built into the serial interfaces. If no new data is available, the layer waits for the interrupt through a system call in user space. Execution times below 0.45ms result from rare cases of beneficial interference between the application and the system timer interrupt: If the application is interrupted before the hardware FIFO is checked, it is possible that the execution time will be shorter because new data arrived while the system timer interrupt was being processed. In this case, the data is ready upon exit from the timer interrupt and the receiving task does not need to wait for a receive interrupt.

Comparing the measurement results for RTEMS and Marron with the correspnding WCETs in Table 2 we can detect an overestimation of the WCET. This can be reduced up to a certain degree with many more annotations. Up to this point there were 120 hours spent in WCET analysis including training period and meetings. The author of the analysis started without any previous knowledge in WCET analysis with aiT but with strong prior knowledge in hardware and software development.

4 Conclusion / Outlook

In this paper, we presented the early stage of a project aiming at the practical application of tool-supported verification to complex embedded systems, considering both functional properties and timing as a non-functional property. For functional properties, simple examples were used to evaluate parts of the methodology with promising results. However, more complex, realistic problems need to be considered and the methodology needs to be developed further. For timing properties, static WCET estimations were compared against measured execution times. It turns out that static methods are difficult to apply to today's increasingly complex multicore hardware architectures, especially when these were

not designed for determinism [Ca19]. In order to model worst-case behaviour for such architectures, very pessimistic assumptions need to be made, correspondingly leading to pessimistic WCET estimations.

To deal with these problems, monitors observing execution times could be generated from the timing specification and serve as a basis for supplementing static WCET estimation with runtime verification. This would lead to a similar approach as presented above for functional verification. Such a method would belong to the class of measurement-based probabilistic timing analysis methods in the sense of [Ca19]. A unified approach for the verification of functional as well as timing properties, supplementing static verification with runtime verification such that even complex systems remain controllable seems to be feasible.

Acknowledgements: This project has received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 737475. This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation programme and Spain, France, United Kingdom, Austria, Italy, Czech Republic, Germany. This project has also received funding from the Federal Ministry of Education and Research (BMBF) under agreement No 16ESE0157. We would like to give special thanks to the people from *AbsInt Angewandte Informatik GmbH* for their support and *Thales Alenia Space* for the usecase application.

References

[20a] AbsInt aiT, Apr. 2020, URL: https://www.absint.com/ait/.

[20b] AQUAS EU Project, 2020, URL: https://aquas-project.eu.

[20c] RTEMS Real Time Operating System, Feb. 2020, URL: https://www.rtems.org/.

[20d] UPPAAL, 2020, URL: http://www.uppaal.org/.

[Ca19] Cazorla, F. J.; Kosmidis, L.; Mezzetti, E.; Hernandez, C.; Abella, J.; Vardanega, T.: Probabilistic Worst-Case Timing Analysis: Taxonomy and Comprehensive Survey. ACM Comput. Surv. 52/1, Feb. 2019, ISSN: 0360-0300, URL: https://doi.org/10.1145/3301283.

[Le80] Levitt, K.; Neumann, P.; Robinson, L.; of Standards, U. S. N. B.; for Computer Sciences, I.; Technology: The SRI Hierarchical Development Methodology (HDM) and Its Application to the Development of Secure Software. U.S. Department of Commerce, National Bureau of Standards, 1980.

[LS09] Leucker, M.; Schallhart, C.: A Brief Account of Runtime Verification. Journal of Logic and Algebraic Programming 78/5, pp. 293–303, May 2009, URL: http://dx.doi.org/10.1016/j.jlap.2008.08.004.

[MT18] Mahboubi, A.; Tassi, E.: Mathematical Components. 2018.

[Ru16] Rufino, J.: Towards integration of adaptability and non-intrusive runtime verification in avionic systems. ACM SIGBED Review 13/, pp. 60–65, Mar. 2016.

[Wa04] Wang, F.: Formal verification of timed systems: a survey and perspective. Proceedings of the IEEE 92/8, pp. 1283–1305, Aug. 2004, ISSN: 0018-9219.

SENSYBLE 2020 Program

	Thursday, October 1st 2020
8:30 – 8:45	Welcome and Introduction
8:45 – 9:30	Session 1: Smart and Connected Vehicles
	Assessment of Current Intrusion Detection System Concepts for Intra-Vehicle Communication *Oleg Schell, Jan Peter Reinhard, Marcel Kneib and Martin Ring*
	Development of a Vehicle Simulator for the Evaluation of a Novel Organic Control Unit Concept *Melanie Brinkschulte*
	Effects of the Sampling Technique on Sender Identification Systems for the Controller Area Network *Marcel Kneib and Oleg Schell*
9:30 – 9:45	Discussion Break
9:45 – 10:30	Session 2: Smart Interaction with Real and Abstract Objects
	EAVE: Emotional Aerial Vehicle Evaluator *Marc Lieser, Ulrich Schwanecke and Jörg Berdux*
	Citcom - Citation Recommendation *Melina Meyer, Jenny Frey, Tamino Laub, Marco Wrzalik and Dirk Krechel*
	Bidirectional Transformer Language Models for Smart Autocompletion of Source Code *Felix Binder, Johannes Villmow and Adrian Ulges*
10:30 – 10:45	Discussion Break
10:45 – 11:30	Session 3: Smart Sensors and Shared Environments
	A Decade of Energy Awareness Technology Evolution for Sensor Nodes *Marcus Thoss*
	BASE MoVE - A Basis for a Future-proof IoT Sensor *Jens-Peter Akelbein, Kai Beckmann, Mario Hoss, Samuel Schneider, Stefan Seyfarth and Marcus Thoss*
	Modeling of Change Response in Interweaving Systems as Ontology Alignment Adaption *Matthias Jurisch and Bodo Igler*
11:30 – 11:45	Discussion Break
11:45 – 12:45	Lunch Break
12:45 – 13:45	Session 4: Smart Applications Using Augmented Reality
	A Tangible Object for General Purposes in Mobile Augmented Reality Applications *Linda Rau, Robin Horst, Yu Liu, Ralf Dörner and Ulrike Spierling*
	Integration of Game Engine Based Mobile Augmented Reality Into a Learning Management System for Online Continuing Medical Education *Robin Horst, Dennis Fenchel, Reimond Retz, Linda Rau, Wilhelm Retz and Ralf Doerner*
	Presenters in Virtual Reality in Slideshow Presentations *Robin Horst, Linda Rau, Lars Dieter, Manuel Feller, Jonas Gaida, Andreas Leipe, Julian Eversheim, Julia Wirth, Jörn Bachmeier, Julius Müller, Maik Melcher and Ralf Doerner*
	A Discussion on Current Augmented Reality Concepts Which Help Users to Better Understand and Manipulate Robot Behavior *Kai Groetenhardt*
13:45 – 14:00	Discussion Break
14:00 – 14:45	Session 5: Smart Foundations
	Requirements and Mechanisms for Smart Home Updates *Peter Zdankin, Oskar Carl, Marian Waltereit, Viktor Matkovic and Torben Weis*
	Complexity Analysis of Task Dependencies in an Artificial Hormone System *Eric Hutter, Mathias Pacher and Uwe Brinkschulte*
	Unified Approach to Static and Runtime Verification *Olga Thoss, Andreas Werner, Robert Kaiser and Reinhold Kroeger*
14:45 – 15:00	Discussion Break
15:00 – 15:30	Conclusion