Demonstrating interoperability between unmanned ground systems and command and control systems

Kim Mathiassen*

Norwegian Defence Research Establishment (FFI),
Postboks 25, 2027 Kjeller, Norway
Email: Kim.Mathiassen@ffi.no
*Corresponding author

Frank E. Schneider

Fraunhofer FKIE,
Fraunhoferstraße 20, 53343 Wachtberg, Germany
Email: Frank.Schneider@fkie.fraunhofer.de

Paul Bounker

US Army CCDC-GVSC,
FCDD-GVR-TI/MS 264, 6501 E. 11 Mile Rd. Warren,
MI 48397-5000, USA
Email: Paul.J.Bounker.civ@mail.mil

Alexander Tiderko

Fraunhofer FKIE,
Fraunhoferstraße 20, 53343 Wachtberg, Germany
Email: Alexander.Tiderko@fkie.fraunhofer.de

Geert De Cubber

Royal Military Academy (RMA),
Av. De La Renaissance 30, 1000 Brussels, Belgium
Email: Geert.De.Cubber@rma.ac.be

Magnus Baksaas

Norwegian Defence Research Establishment (FFI),
Postboks 25, 2027 Kjeller, Norway
Email: Magnus.Baksaas@ffi.no

Jakub Główka and Rafał Kozik

Łukasiewicz Research Network – Industrial Research Institute for Automation and Measurements PIAP,
Al. Jerozolimskie 202, 02-486 Warsaw, Poland
Email: Jakub.Glowka@piap.lukasiewicz.gov.pl
Email: rafal@rafalkozik.com

Thomas Nussbaumer

RUAG AG,
Thunstrasse 29, 3770 Zweisimmen, Switzerland
Email: Thomas.Nussbaumer@ruag.ch

Juha Röning

Biomimetics and Intelligent Systems Group (BISG),
University of Oulu,
P.O. Box 4500, 90014 University of Oulu, Finland
Email: Juha.Roning@oulu.fi

Johannes Pellenz and André Volk

Federal Office of Bundeswehr Equipment,
Information Technology and In-Service Support (BAAINBw),
Ferdinand-Sauerbruch-Straße 1, 56073 Koblenz, Germany
Email: JohannesPellenz@bundeswehr.org
Email: AndreVolk@bundeswehr.org

Abstract: This paper describes the research and experiment efforts of the NATO STO group IST-149-RTG capability concept demonstrator for interoperability within unmanned ground systems and C2 and the NAAG team of experts on UGV. The main purpose of the group was to investigate possible standards for controlling UGVs and tests them in a real world scenario. The efforts have been two folded, where the first effort was two NATO groups having an experiment demonstrating interoperability between the UGVs and OCUs available within the group. The Belgium contribution is done in the EU project ICARUS. Both efforts used the Joint Architecture for Unmanned Systems (JAUS) with the interoperability profile (IOP) to successfully enable interoperability between the systems. The trials showed that it is possible to extend the systems quite easily and achieve compliance with parts of the standard in a relatively short time.

Keywords: interoperability; interoperability profile; IOP; unmanned ground system; UGS; unmanned ground vehicle; UGV; command and control systems; Joint Architecture for Unmanned Systems; JAUS; robotic operating system; ROS.

Reference to this paper should be made as follows: Mathiassen, K., Schneider, F.E., Bounker, P., Tiderko, A., De Cubber, G., Baksaas, M., Główka, J., Kozik, R., Nussbaumer, T., Röning, J., Pellenz, J. and Volk, A. (2021) 'Demonstrating interoperability between unmanned ground systems and command and control systems', *Int. J. Intelligent Defence Support Systems*, Vol. 6, No. 2, pp.100–129.

Biographical notes: Kim Mathiassen received his Master's in Engineering Cybernetics from the Norwegian University of Science and Technology in 2005. In 2017 he received his PhD at the Department of Infomatics, University of Oslo. His topic of his thesis was robotic ultrasound systems, and was conducted in collaboration with the Intervention Center, Oslo University Hospital. In 2012 he was a visiting researcher at the Altair Robotics Laboratory at the University of Verona, Italy. Since 2015 he is a researcher at the Norwegian Defence Research Establishment, and his main research interests are unmanned ground vehicles and autonomous systems.

Frank E. Schneider is a scientific employee at the Fraunhofer Institute for Communication, Information Processing and Ergonomics. He is the Deputy Department Head and program manager for the German Federal Armed Forces R&D program 'Unmanned Systems'. His current program topics include CBRNE, RSTA and EOD/IED. He has also been involved with robotics competitions since 1994 when he made the third place at the AAAI Robot Competition with the 'Rhino'-Team from the University of Bonn. In 2004 he founded the EuropeanRobotics Group and later in 2005 the European Land Robot Trials (ELROB). Under his governance the event was developed into the lead European outdoor/off-road robotic competition. In 2017 he founded the new European Robotics Hackathon (EnRicH) event. ENRICH is the world's first and only robotics trial that gives you pure and unspoiled real world scenarios in a full scale nuclear facility.

Paul Bounker is a senior robotics project engineer at the US ARMY's Ground Vehicle Systems Center, where he has been employed since 1994. Currently, he is working on Program Manager Force Projection's Robotic Autonomous Systems – Ground Interoperability Profile Effort. In, addition he is leading the integration for the autonomous systems test capability which is looking at using simulation to enhance the testing of autonomous systems and eventually artificial intelligence/machine learning. He was lead for Systems Engineering and Technical Development for the Cross Command Collaboration Effort (3CE) which developed the M&S portion of the Network Integration Evaluation (NIE) testing that was done at Ft. Bliss. He worked as a project engineer at the US Navy's Naval Aviation Depot in Norfolk, Virginia from 1989–1994. He holds a BSE in Electrical Engineering from the Wayne State University in Detroit, Michigan.

Alexander Tiderko obtained his Dipl. Inform. in Computer Science from the University of Bonn, Germany in 2006. At present he is a Scientist at the Fraunhofer Institute for Communication, Information Processing and Ergonomics FKIE in Wachtberg Germany. His primary research interests include networked multi-robot systems and robot interoperability.

Geert De Cubber is a team leader of the Robotics and Autonomous Systems Unit of the Department of Mechanics of the Belgian Royal Military Academy. He is also a senior researcher at this institute with a research focus on developing robotic solutions for solving security challenges like crisis management, the fight against crime and terrorism and border security. He received his Diploma in Mechanical Engineering in 2001 from the Vrije Universiteit Brussel (VUB) and his Doctoral in Engineering in 2010 from the Vrije Universiteit Brussel and the Belgian Royal Military Academy (RMA). His research interests include methodologies for perceiving and control of tough multi-agent robotic systems, across the air, land and maritime domain. His major goal is to find new ways to make multiple robotic systems capable of understanding their environment and decide on optimal collaborative strategies.

Magnus Baksaas received his Master's in Electronics and Computer Technology where main field of study was Cybernetics from the University of Oslo in 2015. Since 2015 he work as a researcher at the Norwegian Defence Research Establishment. His main research interests are unmanned ground vehicles and autonomous systems.

Jakub Główka is a robotic researcher leading an interdisciplinary team working on innovative robotic solutions. He graduated from the Warsaw University of Technology, Faculty of Electronics and Information Technology, specialisation Electronics and Computer Engineering. He is working with robotics in Łukasiewicz – PIAP since 2008 in different roles starting from engineering and design to leading projects and a team. He is an author or co-author of more than 20 papers, published in conference proceedings and journals.

Rafał Kozik is a software engineer with over ten years of experience. He is an author of solutions for robotics, embedded systems and safety-critical components in telecommunication. While working for Samsung managed a team of ten software developers and participated in the Software Architecture Board operations. At ROSCon 2016 he gave a presentation on robustness of ROS2 communication layer. He is graduated in the Faculty of Physics and Applied Computer Science at AGH University of Science and Technology in 2008.

Thomas Nussbaumer studied Physics and received his Master's and PhD from the Swiss Federal Institute of Technology, ETH, Zurich, then he was a Postdoc at the University of Basel. Since 2008 he is a Senior Scientist and research manager at the RUAG, Switzerland, working mostly in projects related to unmanned systems (UMSs) by appointment of the Swiss MOD (Armasuisse S&T).

Juha Röning is a Professor of Embedded System at the University of Oulu. He serves also as the Visiting Professor of Tianjin University of Technology, China. He is a principal investigator of the Biomimetics and Intelligent Systems Group (BISG). In 1985 he received Asla/Fullbright Scholarship. From 1985 to 1986 he was a Visiting Research Scientist in the Center for Robotic Research at the University of Cincinnati. From 1986 to 1989 he held a Young Researcher Position in the Finnish Academy. In 2000 he was nominated as Fellow of SPIE. He has three patents and has published more

than 300 papers in the areas of computer vision, robotics, intelligent signal analysis, and software security. He is currently serving as a Board of Director for euRobotics AISBL. He is also a steering board member of ARTMIS-IA.

Johannes Pellenz studied Computer Science at the University of Koblenz-Landau and the University of Georgia in Athens (GA, USA). He received his Diploma and his PhD from the University of Koblenz. His topic of his PhD thesis was 'Active sensing for autonomous mobile systems'. Since 2009, he has been working as a civil servant at the Procurement Office (BAAINBw) of the German Armed Forces (Bundeswehr). He is a project manager of severals R&D and procurement projects and a member of the NATO team of experts UGV working group.

André Volk studied Computer Science/Computational Visualistics at the University of Koblenz-Landau. He wrote his Master's thesis on the enhancement of the self-localisation of outdoor mobile robots with 3D LiDAR data. For his thesis, he received the AFCEA award for studies. Since 2009, he has been working as a civil servant at the Procurement Office (BAAINBw) of the German Armed Forces (Bundeswehr). He is responsible for the research and technology area of unmanned land systems. He is a member of the NATO team of experts UGV working group.

1 Introduction

Unmanned ground vehicles (UGVs) have been used for bomb disposal for a long time, but are becoming increasingly relevant for other applications in the modern battle-space. These systems can carry massive sensor suits, delivering unprecedented streams of data right from the front line. On the other hand these systems still require tele-operation in most cases. It is very important to realise that the ISR data will remain largely useless if there is no proper way in exchanging the information between the coalition partners and/or piping it into C2 systems.

There has been several standardisation attempts and trials experimenting using the standards. Here we will give a short overview of relevant research, while a more thorough description of relevant standards will be given in Section 2. In the academic robotics community several attempts has been made, mostly by designing a middleware and default message definitions. Many of these focus on meeting real-time requirements, such as OROCOS (Bruyninckx et al., 2003) and RT-middleware (Ando et al., 2005). The robot operating system (ROS) (Quigley et al., 2009) has in resent years become one of the most use robotic middlewares, and implement its own communication solution with one central component that manages connections between components. This had some limitations, and in the development of ROS2 it was decided to use the Data Distribution Service (DDS), which make ROS2 handle real-time requirements (Maruyama et al., 2016) and has security measures (Kim et al., 2018).

In the military land domain other attempts has been made for standardisation. Many of these has been open system architectures. The US Modular Open Systems Approach (MOSA) (Open Systems Joint Task Force, 2004) is one such strategy, and this lead to several efforts on open systems. The US army the Vehicle Integration for C4ISR/EW Interoperability (VICTORY) (Elliott et al., 2012) was one of the efforts, and is a open system architecture for military land platforms. UK has a similar approach with their Land Open Systems Architecture (LOSA), which is an architecture that aim to bring together land vehicles, dismounted soldiers, static bases and more (Henshaw, 2011). The Generic Vehicle Architecture (GVA) (UK Ministry of Defence, 2013) is a one part of this, which is an open architecture for land vehicles. In Bergamaschi et al. (2013), a practical experiment using LOSA and in Pearson and Kolodny (2013) an effort between UK and US is described to use the LOSA to enable disparate assets to autonomously operate collaboratively and coherently. Inspired by the national efforts NATO has also developed their own standard for land-based platforms, the NATO Generic Systems Architecture (NGVA) (NATO, 2018a).

Another effort for standardisation is the Joint Architecture for Unmanned Systems (JAUS). It was initiated in 1998 by the US as the Joint Architecture for Ground Unmanned Systems, and was in 2005 adapted by Society of Automotive Engineers (SAE) (Rowe and Wagner, 2008). In Serrano et al. (2015), the EU funded projects ICARUS and DARIUS explain their different approaches towards interoperability. ICARUS uses the JAUS middleware to facilitate this, while DARIUS used a central ground control station (GCS) that interfaces with all the unmanned system. Both projects had successful field tests using air, land and maritime unmanned system in different scenarios. NATO efforts to the UGV interoperability profile (IOP) are given in Bounker and Volk (2015), where an experiment with IOP complaint components from the US, Germany and Turkey is described.

Battle management language (BML) (Heffner et al., 2010) is an human-readable language to exchange orders, reports and requests among command and control systems (C2 systems), simulation systems and real units. In Seehuus et al. (2019), a modified version of the coalition battle management language (C-BML) is used to control unmanned system in the air, land and maritime domain. The usage is limited to one or more vehicles in the same domain, and experiments in each domain are described. In Langerwisch et al. (2013) a field experiment is conducted using BML with six heterogeneous UGVs and unmanned aerial vehicles (UAVs) doing reconnaissance and surveillance.

As there exist many standards, with many of the focusing on different aspects of interoperability, i.e., inter-vehicle, intra-vehicle or command and control, there has been some attempts to bridge some of the standards together. In Pradhan et al. (2017a), an approach to bridge the UGV IOP standard to NGVA is proposed, and in Pradhan et al. (2017b) this is expanded to also include military internet of things standards. There are also examples of efforts to join the land domain with other domains, for instance in Elliott et al. (2019) where the possible interoperability between VICTORY and future airborne capability environment (FACE) Standards were explored.

The PhD thesis (Marques, 2018) provides a more in depth overview of relevant standards for autonomous systems.

ELROB and EURATHLON are two outdoor robotic competitions that encourage standardisation in several aspects (Schneider et al., 2015). Before the competition a common data set is shared with all participants using the ROS bag format, while the exchange of real-time data during the competition is done using WebDAV. The competitions also encourage the usage of ROS, as it makes sharing code between participants easier.

This paper describes the research and experiments of the NATO STO group IST-149-RTG ‘capability concept demonstrator for interoperability within unmanned ground systems and C2' and the NATO Army Armaments Group 'team of experts on UGV'. The main purpose of these groups was to investigate possible interoperability standards for controlling heterogeneous UGVs. This included experimental trials with various operator control units (OCUs) in real world scenarios. More detailed information can be found in the IST-149 group's final report (Schneider et al., 2020).

The main focus was on a joined experiment of the two groups demonstrating interoperability between the UGVs and OCUs that were available within the groups. In addition the group member from Belgium was able to establish a cross-fertilising link to work that is done in the EU project ICARUS. Both efforts used the JAUS-based IOP to successfully enable interoperability between the robotic operating system (ROS) driven systems. Both Fraunhofer FKIE and US Army CCDC-GVSC have independently developed software to exchange information from the IOP domain to ROS and visa-versa. The synergetic trials showed successfully that it is possible to extend the systems quite easily and achieve compliance with major parts of the standard in a relatively short time.

This paper begins with a brief overview of relevant interoperability standards in Section 2. In Section 3 the overall software architecture used by the groups is described. Section 4 gives an overview of the robotic systems and OCUs and how they were used in a interoperability experiment. In Section 5 the experimentation in the ICARUS project is outlined. Finally, in Section 6 the experiences from the work is discussed.

2 Standards in robotics

The use of standardised message formats and processes in the development of a robot control software increases the interoperability between systems and enables new functions to be integrated more quickly. The using of standard formats and communication can be achieved by standardised middleware systems. There exist many of middleware systems for robotic, both in the civilian sector and the military sector. In this section we provide a brief overview of relevant systems for military related UGVs in our experiments.

2.1 JAUS IOPs for UGV

The US Department of Defence initiated an effort to standardise the interfaces between ground robotics platforms, payloads, radios and OCUs. The IOP, and particularly the logical interface definitions, makes extensive use of the JAUS services publish by the SAE AS-4 standards body. Where necessary, the IOP extends published JAUS services by providing additional profiling rules and custom services to support a wider variety of projects and programs. In some cases, these custom services are being provided back to the AS-4 committee for possible adoption. The IOP is expected to maintain this close relationship with the AS-4 JAUS body going forward.

The primary goal of the IOP is to define the physical, electrical, and logical interfaces between platforms, payloads, radios and OCUs for unmanned ground systems (UGSs). This promotes interoperability between such devices, without prescribing any internal implementation. This allows vendors and suppliers to use whatever tools, intellectual property, or proprietary implementations they wish. The IOP does not encourage one solution over another, so long as the interfaces are compliant.

A primary goal of this initiative is to leverage existing and emerging standards within the unmanned vehicle (UxV) community such as the SAE AS-4 JAUS standard and the US Army Unmanned Aircraft Systems (UAS) Project Office IOPs. The end goals of this effort are:

- facilitating interoperability among new UGV initiatives and legacy systems
- facilitating interoperability between controllers and UxV robotic system(s)
- facilitating collaboration between UGV and UAS systems
- providing a path forward to standardised interoperable technology solutions
- promoting payload and on-board subsystem modularity and commonality across the portfolio of UGV systems.

Furthermore, the unmanned control systems (UCS) segment architecture has been brought under the AS-4 group. UCS uses a model driven architecture (MDA) approach to define a platform independent model (PIM) of the data being exchanged. While some examples and projects have been created to define a platform-specific model (PSM) to demonstrate plug-and-play interoperability, this is not currently the main emphasis of the effort. Ultimately, AS-4 looks to align JAUS and UCS such that JAUS could be considered a specific PSM for an extended UCS architecture. This is likely a long term process, and it is not expected that this alignment will have significant impact on the IOP.

2.2 STANAG 4586

The NATO Standardization Agreement (STANAG) 4586 is a NATO Standard Interface of the UCS/UAV interoperability (NATO, 2012). It includes data link, command and control, and human/computer interfaces, and defines five levels of UAV interoperability:

1 indirect receipt/transmission of payload data

2 direct reception of the payload data by the UCS, when it has direct communication with the UAV

3 control and monitoring of the UAV payload in addition to direct receipt of ISR and other data

4 control and monitoring of the UAV, excluding launch and recovery

5 control and monitoring of the UAV, including launch and recovery.

JAUS and STANAG 4586 are both message-based standards. Consequently, duplication of effort exists in terms of capabilities and the domains targeted by both standards. For example, both JAUS and STANAG 4586 continue to introduce functionality in order to support unmanned surface vehicles (USVs). However, STANAG 4586 is still primarily focused on UAVs and JAUS is primarily focused on UGVs.

Consequently, JAUS currently supports some different capabilities than STANAG 4586 in terms of representing the environment surrounding unmanned systems and providing functionality for avoiding obstacles. Another technical difference between the standards is that JAUS treats unmanned systems as generic assets, whereas STANAG 4586 employs vehicle specific modules (VSMs) within the standard.

2.3 Battlefield management language

BML (Heffner et al., 2010) is an artificial, unambiguous, human-readable language and open standard used to express and to exchange orders, reports and requests among C2 systems, simulation systems and real units. In addition, BML also may be used to interact with robotic forces. In short, BML allows C2 systems and their users to interact with robot systems in the same way as with real units or with units simulated in simulation systems.

Since BML is unambiguous it allows automatic processing to command single as well as a multitude of robots, provided each robot is programmed how to interpret and execute commands within its capabilities. This allows human forces to be assisted by the robot units using the same commands.

Two examples of BML commands (which could be either given to a single robot or a group of robots) are given below:

- OB advance tasker taskee route-where start-when (end-when) mod why label
- OB reconnaissance tasker taskee at-where start-when end-when mod why label.

The Fraunhofer FKIE is one of the main drivers, not only of the robotics related BML development. More details can be found in Langerwisch et al. (2013) and Remmersmann et al. (2013).

2.4 NATO Generic Vehicle Architecture

The NATO defined STANAG 4754 'NATO Generic Vehicle Architecture (NGVA) for land systems' (NATO, 2018a). It covers aspects of a vehicle's infrastructure. It mandates the usage of open standards for interoperability between components in the vehicle, and specifies this for physical, electrical and data interfaces. The data exchange between components is based on data distribution service (DDS) (OMG, 2015), a standardised middleware defined by the Object Management Group (OMG). The NGVA data model (NATO, 2018b) defines a semantic reference for the data types and topics to be used in all messaging across the infrastructure. Together with DDS it ensures an interoperable communication with all components.

In addition, STANAG 4754 also cover other aspects of the vehicle infrastructure. These include safety (NATO, 2018c), verification and validation (Ota et al., 2017) and Crew Terminal Software Architecture (CTSA) (NATO, 2018d). The CTSA aims at defining a standard Inter-Process Communication interface and the nature of its deployment in alignment with Crew Terminal hardware, so that CT software modules can be made portable and re-usable.

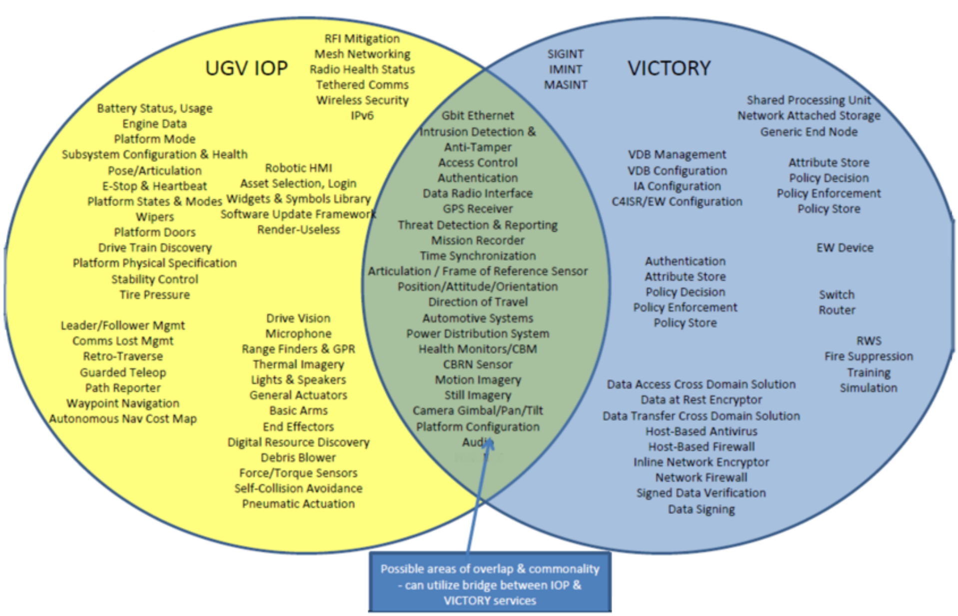
Currently no NGVA models published to include UxV into NGVA infrastructure. There are only approaches for using gateways to include other standards (Pradhan et al., 2017a).

2.5 Vehicle integration for C4ISR/EW interoperability

The VICTORY initiative began in an effort to resolve US Army vehicle troubles created by 'bolt on' field equipment. Through use of VICTORY, tactical wheeled vehicles and ground combat systems may recover lost vehicle space, weight and power. The implementation enables platform systems to share information. VICTORY architecture implementations provide an integrated picture to the warfighters. Through use of open architecture, platforms can accept future technologies without significantly re-designing platform systems. The initiative is developing a framework for integration electronic mission equipment including C4ISR and wlectronic warfare system on ground platforms. The framework includes: an architecture, a standard specification, and reference designs. The architecture includes definitions of common terminology, systems, components, and interfaces. The specification provides technical specifications for the systems, components, and interfaces identified in the architecture. The VICTORY technical approach includes: a 'data bus-centric' design; sharable hardware components; open standard physical and logical interfaces; a shared set of data bus services; and shared hardware and software information assurance (IA) components. The shared hardware components enable the deployment of software additions without adding additional hardware. The standard physical and logical interfaces enable system and C4ISR/EW components to communicate with each other.

Figure 1 An overview over the main components of VICTORY and IOP, and how the two standards overlap (see online version for colours)

s



It is envisioned that portions of VICTORY will overlap with Robotic Autonomous Systems – Ground (RAS-G) IOP, see Figure 1 as an example of such an intersection. Such overlap will be handled by sharing information across standard data buses between VICTORY supported subsystems and RAS-G IOP supported subsystems.

2.6 ROS

The ROS is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behaviour across a wide variety of robotic platforms. ROS allows an easy and fast prototyping. UGVs, both in the civilian sector and military sector, often utilise ROS, as it is a convenient middleware with many predefined messages. Thanks to open source and a very large community it has many components for perception, navigation and control. The spread of ROS makes it to quasi standard in robotics.

ROS uses message-based publish-subscribe communication, which can be extended by service calls. ROS is used in many different areas, so many requirements must be met. These concern discovering, communication, memory consumption, real-time requirements or approval in commercial products. To meet these requirements, the Open Source Robotics Foundation (OSRF) is currently working on ROS2, which is based on a standardised communication middleware.

2.7 ROS-M

ROS-military (ROS-M) is an on-going effort by the US Defence to promote ROS, maintained independently by the OSRF, by developing and fostering an ecosystem of stabilised, validated, and approved ROS 2.0 packages. The goal is to promote rapid development through shared implementation of common robotic behaviours freely accessible as open source software.

Since the IOP emphasises interfaces between subsystems of ground robotics while ROS-M focuses on defining the middleware of autonomous implementations that would primarily reside on subsystems of ground robotics, the two can be used together or entirely independently of one another. If both are used within an implementation, it will be necessary to convert or bridge between the IOP/JAUS messages used by the interface and the ROS/DDS topics used by the internals.

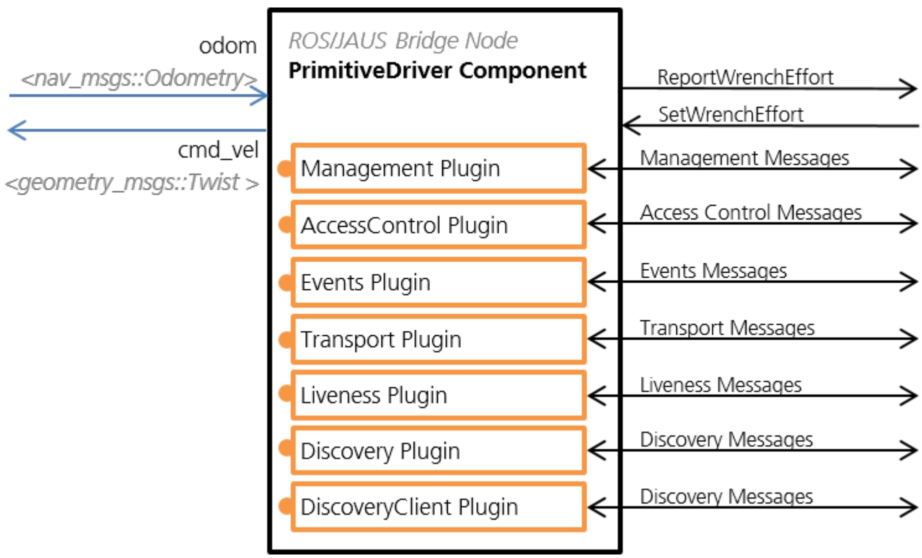
3 Software framework for interoperability

Most of the systems available in the groups IST-149 and the team of expert on UGV use ROS. Unfortunately ROS is ill suited to use as an interoperability standard, because each ROS deployment need a centralised master component to coordinate publishing and subscription of information. With multiple UGVs and OCUs involved which might lose network connection during the mission a centralised system like ROS has several potential problems. In order to adapt the system to use an interoperability standard a bridge between ROS and the standard was proposed. The groups independently decided to use IOP as interoperability standard. Several partners had experience using that standard, and Fraunhofer FKIE had already developed a bridge for communicating between ROS and IOP (open source at https://github.com/fkie/iop_core). GVSC developed their own bridge during the project. The concept of the two bridges are described in the next two sections.

3.1 Fraunhofer FKIE ROS-IOP bridge

The ROS-IOP bridge establishes the communication between ROS software and IOP services. The bridge itself uses a plugin-based design. Each plugin translates an IOP service to corresponding ROS functionality or offers required IOP mechanics for depending services. Some IOP mechanisms, like discovery, events or access control are important in IOP but has no relevance in ROS. However, they are implemented into the bridge to comply the IOP requirements and build up fully-fledged IOP-complaint robot. An IOP component is build up in ROS configuration by including all needed plugins. An example of the PrimitiveDriver component is shown in Figure 2. Depending on system design, all services can be combined to one or split into multiple components.

Figure 2 An example of the PrimitiveDriver component with all required plugins (see online version for colours)



The ROS-IOP bridge has two logical units. One unit contains all plugins to setup an IOP-compliant robot. The second unit offers plugins to connect to IOP-services and enables the use of the bridge for an ROS-based OCU.

3.2 GVSC ROS-IOP bridge

The US Army Combat Capabilities Development Command Ground Vehicle Systems Center (GVSC) has hand-coded a number of RAS-G interoperability (IOP) to X bridges including a RAS-G IOP to ROS 1.0 bridge. It has been challenging to maintain and extend multiple bridges, with new requirements and features often forced into an architecture not inherently designed for new capabilities. This along with schedule constraints tends to result in business logic being placed inside of the bridge which further makes the maintenance of the bridge more challenging. As GVSC works to promote adoption of common interfaces like the RAS-G IOP, which in turn extends published standards from the SAE AS-4 JAUS committee, the capabilities and requirements for protocol bridges continue to expand. In parallel, a growing number of programs and projects are relying on the ROS, and its variants like ROS 2.0 and ROS-M, to provide a common baseline for RAS-G autonomy software implementations across platform types. As a result, there is a growing need and reliance on software bridges capable of converting messages and data between IOP and X endpoints.

GVSC is looking into the feasibility of developing a tool capable of modeling and ultimately creating dedicated bridges which would replace the hand-coded efforts of today. By relying on modern software engineering techniques like model-driven architecture and automated code generation, the goal is to model, generate, and deploy bridges that can be rapidly modified and extended as new requirements are identified.

4 Experiments of IST-149

An experiment was conducted in order to verify and test the implementation of IOP used by all the partners of the two groups. The overall goal was to have each OCU control and receive information from all UGVs involved.

4.1 IOP components used

As the IOP standard is very large, and was made to cover all aspects of communication between UGVs and OCUs, the group agreed on a subset of IOP messages to be used in the experiment before conducting the experiment. Each participant had to support at least the following command message on their system:

- urn:jaus:jss:mobility:PrimitiveDriver.

While the following messages were defined as the minimum sensor feedback of each UGV:

- urn:jaus:jss:mobility:GlobalPoseSensor
- urn:jaus:jss:iop:PathReporter
- urn:jaus:jss:environmentSensing:StillImage
- urn:jaus:jss:environmentSensing:DigitalVideo.

It is worth noting that the video message in IOP only contains the URI to the video. Videos are provided using the Real-Time Streaming Protocol (RTSP), and the URI should point to a server providing the video. The H-264 encoding is defined in IOP to be used as video encoding. Many of the UGVs had cameras or encoder boxes which performed the encoding, but some also used GStreamer, an open source software framework for audio and video processing, to encode and decode video. The Chromium web-kit engine was also used for encoding by one of the participants.

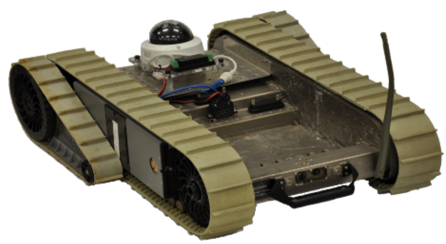
Besides the minimal set of messages, many of the participating systems provided additional capabilities. However, the focus of the experiment was to verify that the above mentioned messages could successfully be sent and received, and that the UGVs could be controlled by the different OCUs.

4.2 UGVs

A total of four UGVs and one UAV was planned to be included in the experiment. Unfortunately, the UAV had an accident shortly before the experiment, and could not be repaired in time. The UGVs used in the experiment are presented in the following sections, and shown in Figure 3.

Figure 3 UGVs used in the experiment, (a) Garm (b) GVR bot (c) Telemax (d) Olav (see online version for colours)

(a)



(b)

(c)

(d)

4.2.1 Garm

Germany and Switzerland provided the UGV platform ‘Garm’, shown in Figure 3(a), and its OCU for the experiment. Garm supports all the interoperability standards which have been proposed in the RTG-149 (and its predecessors); it also took part in different scenarios at the ELROB competitions since 2012. Due to a modular payload concept, Garm can be equipped with different sensors and tools depending on the desired scenario and application. During the IOP experiment in Rena a reconnaissance payload was used on Garm.

The payload consists of an OCULUS SCOUT pan-tilt-zoom camera with thermal imaging function, a pan-tilt searchlight, a GPS receiver, three ACTI Ethernet cameras (one 360° camera to observe the surrounding area of the robot and two cameras for the lateral periphery to allow a better overview in bottlenecks), a ZELOS-C mesh radio, a PC with robot control software and connectors for external laser scanners, which are attached to the robot chassis.

The payload is connected to the robot platform via defined power and communication connectors (usually RJ45). The robot platform is controlled by the PC inside the payload. The control software is based on ROS. It provides assistance and autonomy functions for different applications. In preparation of the experiment in Rena, the FKIE ROS-IOP bridge was added to the control software on the PC. The ROS-IOP bridge enabled any IOP-compliant OCU to access the payload sensors and to send drive commands to the robot. Additionally, autonomous waypoint navigation was integrated, which was successfully tested with both the FKIE's own IOP-compliant OCU and the PIAP OCU.

4.2.2 GVR bot

iRobot (endeavour robotics/FLIR) PackBot is a fielded small robot used primarily for IED interrogation. The base platform is expensive to purchase and expensive to maintain. The design is 'semi-modular' – payloads can be installed to the base chassis, but they can only be installed in specific locations, and the system must have the proper software installed and configured with any new payload. This has led a dependence on the manufacturer for integration of new payloads. US Army Robotic Systems – Joint Program Office funded a standardisation effort to create a kit for retrofitting the PackBot to make them IOP compliant and reduce some of the other sustainment costs

GVSC and ground vehicle robotics (GVR) has continued the development and therefore, GVR bot is a continuation of the standardisation effort to create a good government research platform with an open architecture that is IOP compliant. The system is shown in Figure 3(b).

Using a scanning LiDAR, an accurate digital map can be produced for digital mapping of rooms and buildings by driving the GVR bot around. Having a robust mapping and localisation system allows for waypoint navigation, automated exploration, sensor fusion (chemical, biological, radiation, nuclear and temperature as fused overlays) and object classification. This will assist soldiers in search and cordon of potentially hazardous areas.

4.2.3 Telemax

Telemax is an EOD robot built by Telerob, shown in Figure 3(c). The approximately 80 kg robot (without batteries) is designed for use in cramped confines. The 7-axis manipulator has a long reach and can lift objects with a maximum weights between 3 and 20 kg depending on the manipulator position. With its four independent chain flippers it can easily overcome an obstacle and reach speeds of up to 10 km/h (with wheels attached). By default, Telemax comes with its own proprietary OCU. However, this OCU has no interface for accessing the sensor data (e.g., the images of the cameras) or for injecting control commands by alternative control instances. To make Telemax IOP-compliant we had to access the Ethernet and CAN interfaces on the robot. A control computer was attached to the robot's chassis and connected to these interfaces. The computer is used to run software for assistance and autonomy functions.

The autonomy box is an extension, which is available as an accessory for the Telemax. This is designed similar to the form factor of a Telemax battery and replaces one of the two batteries on the robot. Internally, the autonomy box is equipped with an Intel i7 quad-core processor, an 6-axis inertial measurement unit (IMU) and a Wi-Fi hotspot. In addition, a Hokuyo 2D laserscanner and an Xbox Kinect are mounted on the box, which can be used for local navigation. The autonomy box communicates with Telemax via CAN interface.

The internal cameras of the Telemax provide only an analogue signal. To get digital video for IOP an analogue-to-digital converter with Power-other-Ethernet (PoE)-injector was attached to the autonomy box.

The control software in the autonomy box is based on ROS. However not all functions of the robot are properly supported in the current version of the ROS Telemax driver. Therefore not all skills of the robot could be fully utilised during the IOP tests in Rena. To get this system compliant with IOP the Fraunhofer FKIE ROS-IOP bridge was used.

4.2.4 Olav

The Norwegian Defence Research Establishment (FFI) started in 2015 a project to develop an autonomous vehicle for off-road driving. Parts of that development are documented in Mathiassen et al. (2016). The on-going project Olav has two primary missions. The first is to navigate autonomously to a specific location in order to observe a sector. It should be able to observe the scene of interest for a prolonged period of time. The second mission is aiding a base protection team, where Olav can be sent out to check the perimeter of the base (Mathiassen et al., 2019). Here, Olav will primarily drive autonomously, but sometimes remote operation is needed to do fine navigation of the vehicle.

Olav is based on a Polaris Ranger XP 900 EPS vehicle, and has been modified to be controlled autonomously. It is depicted in Figure 3(d). It has been equipped with cameras and a LiDAR as perception sensors and an inertial navigation system (INS) and Global Navigation Satellite System (GNSS) for localisation. The vehicle uses the ROS as middleware, and thus run mainly on Ubuntu Linux.

A control station developed at the Norwegian Defence Research Establishment is used to define missions for the vehicle, and the missions are defined in a modified C-BML (Seehuus et al., 2019). A higher level decision making module has been implementing by adapting an existing framework used on autonomous underwater vehicles (Wiig et al., 2016). This module composes a complex mission into a tree of task. The decision making module use a route planning service. The route planning service provides routes planned used geographical map data. A perception module (Haavardsholm et al., 2016) has been developed that produces a map over the surrounding area of the vehicle. LiDAR is the primary sensor when producing the map. The map estimates the driveability of the sounding area and is used by a motion planning component. This component plans a drivable trajectory that avoids obstacles and at the same time follows the desired route. In addition there is a localisation component which use a GNSS and an inertial measurements unit to estimate the global position and orientation of the vehicle.

The system described above does not fit well in to the direct tele-operation architecture used in this experiment. Therefore it was necessary to make adoptions in the system in order for the system to be compliant with IOP. The main adoption was to use the Fraunhofer FKIE IOP bridge for ROS, and to use GStreamer for providing RTSP servers for publishing video in an IOP compliant format.

4.2.5 UAV

The University of Oulu provided low-cost quadcopter was built using a commercially available flight control unit (FCU) with GPS localisation, a custom video link solution utilising a Raspberry Pi mini-computer, a 300 Mbps 2.4 GHz Wi-Fi dual antenna router, a gimbal for a Raspberry Pi camera and a 433 MHz telemetry radio. Hardware wise, the overall cost of the quadcopter was about €400–500. The FCU and the rest of the system was integrated with ROS Kinetic running on a remote laptop computer, referred here also as GCS. The ROS-IOP bridge implementation used was the one provided by Fraunhofer FKIE, which was reconfigured so that it could be used with the UAS. In ROS, the MAVROS node was used for monitoring and controlling the UAS via FCU with micro air vehicle link (MAVLink) protocol.

Figure 4 UCUs used in the experiment, (a) PIAP (b) GVSC (c) Fraunhofer FKIE (see online version for colours)

(a) (b) (c)

Unfortunately, just before the IOP experiment, a major crash occurred during testing due to a failed motor bearing, causing the copter to shut down abruptly in more than 10 metres altitude. The analysis of the logs indicated that the FCU probably interpreted the failed bearing as a sign of crashing in to something and triggered a fail-safe, shutting down the drone in mid-flight. This led to UAV frame being broken and the destruction of most of the motors, which could not be replace in time for the IOP experiment in Rena, Norway.

4.3 Operator control units

For controlling the UGVs three different OCUs were provided by different partners. The OCUs are shown in Figure 4 and described in the following sections.

4.3.1 PIAP OCU

PIAP conceptual OCU consisted a docking station with joysticks and buttons and a tablet computer inserted into it. It is intended to be operating as a mobile (remote) control station for a UGV of small and medium sizes. Tablet and docking station are powered from the internal battery placed inside the tablet with the possibility of recharging it via an external power source. The OCU contains connectors enabling connection to the tablet of external peripherals such as wired communication (RJ45), wireless communication system, additional battery, gamepad and an additional screen. The operator can use a tablet touch screen and two joysticks located on both sides of the docking station.

The software of the PIAP conceptual OCU was realised to be interoperable within IOP version 2 framework. The main available resources are platform state and control, mobile base movement, waypoint driving, visualisation of geo-positioning on built-in mapping service, simple manipulator operation, and video configuration and displaying. PIAP conceptual OCU implementation has been based on OpenJAUS SDK (http://openjaus.com/) that has simplified the design and testing phases and allowed for immediate application of any corrections or modifications in the way the OCU cooperated with platforms. No custom ROS-IOP bridges were used on the OCU side. The OCU is shown in Figure 4(a).

4.3.2 GVSC OCU

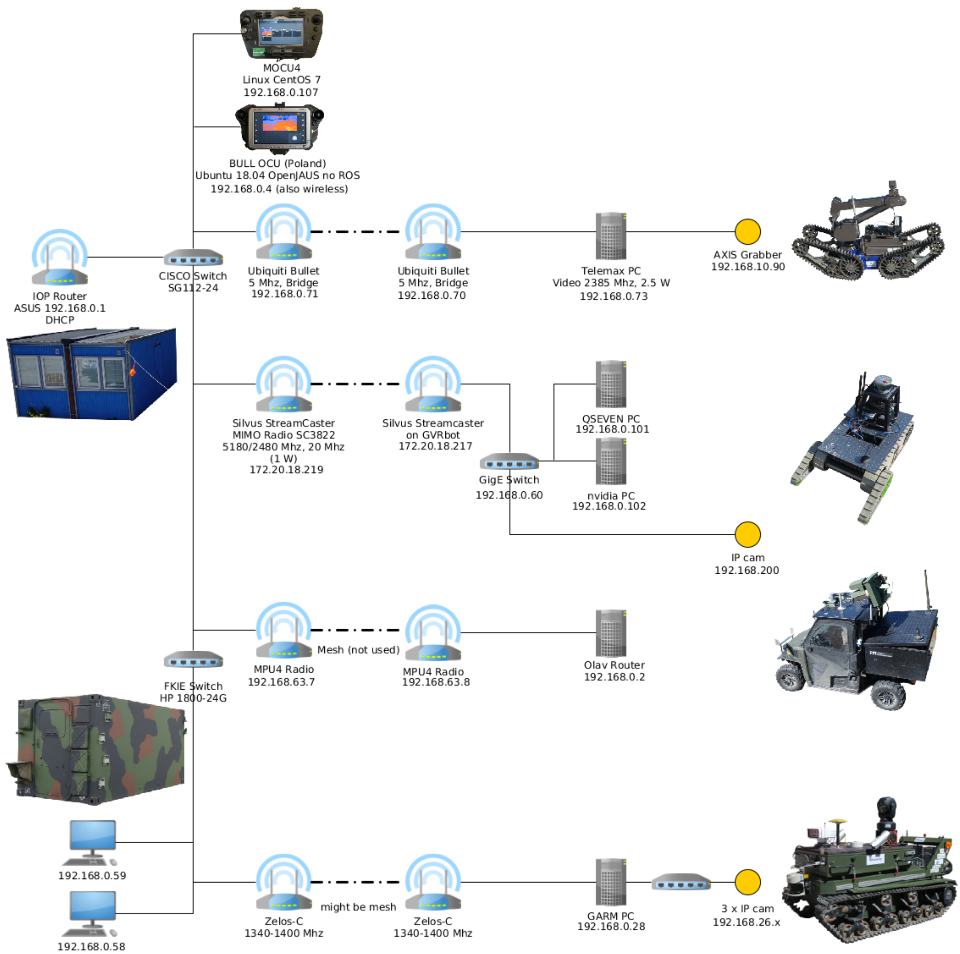
The OCU is based on a consolidated piece of hardware for control of unmanned systems. Based on a commercial ruggedised 7” tablet computer with an Intel M5 processor. The tablet is wrapped in Hapflex (rugged plastic) enclosure to provide for realistic human machine interface with rugged I/O. The total system weight is approximately 1.8 kg.

The controller is shown in Figure 4(b) and includes two joysticks, two covered switch guards, deadman grip switch, touchscreen and optional second screen (not shown in the current picture). Upon startup the controller shows available IOP compliant robots on its screen.

4.3.3 Fraunhofer FKIE OCU

The control software of Fraunhofer FKIE is a ROS-based software, and is shown in Figure 4(c). This consists of the plug-ins for RQT and RVIZ. This allows the GUI to be adapted to any robot and application scenario. RQT includes the plugins for video display of cameras on the payload, control elements for pen-tilt units, visualisation of power supply and IOP specific control elements. Map, position and laser data are visualised using RVIZ. RVIZ was also used to build a multi-robot overview. The ROS-based OCU was connected to IOP network using Fraunhofer FKIE ROS-IOP bridge.

Figure 5 The network architecture during the experiment (see online version for colours)



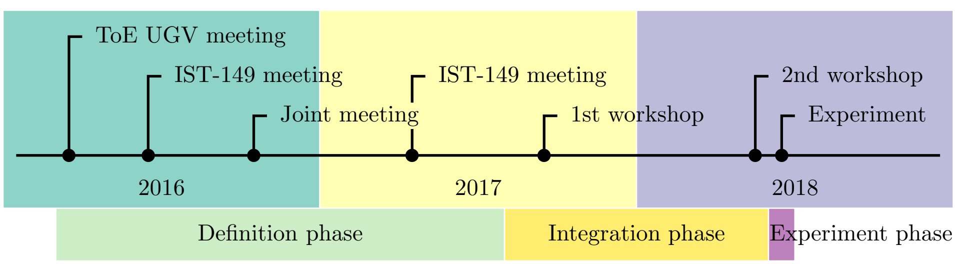
4.4 Communications

Communication between the systems during the IOP tests was IP-based. Which devices are used for this, the IOP does not prescribe. So each robot system was equipped with its own radio system. The Garm communicates via ZELOS-C radios, the OLAV via persistent systems MPU4 radios, Telemax via Ubiquiti Bullet 5 radios and the GVR-BOT via Silvus Streamcaster. To enable each OCU to access the robots, all receivers were connected to a common switch, mostly by detours. In order to avoid any address conflicts, the addresses were assigned in advance and a DHCP server was set up. The overview of the most important network participants is shown in Figure 5. The infrastructure for controlling the robots was distributed to the two different containers.

4.5 Planning and workflow

The planning of the experiment was initiated by the NAAG team of experts on UGV group in March 2016. There had been some discussion regarding an experiment before this, but in the meeting in March the experiment scope and participants were more firmly settled. The NATO STO group IST-149-RTG capability concept demonstrator for interoperability within UGSs and C2 was asked to join the experiment, and a joint meeting between the groups was held in October. the same year. At this meeting the workflow was established. First a definition phase, to define the scope, aim and participants of the experiment. After this phase an integration phase began. This phase consisted of two integration workshops to prepare for the experiment. In the integration workshops all components were tested. For the larger vehicles simulators were used instead of the physical hardware. The first integration workshop was in autumn 2017 and the second in the spring 2018, approximately one month before the actual experiment. The timeline and workflow is shown in Figure 6.

Figure 6 The figure shows the timeline for meetings of the two groups, integration workshops and the experiment (see online version for colours)



Notes: In addition the timespan of the workflow phases are shown.

4.6 Scenario and execution

The experiment was held at a close quarter's combat training ground at the training site Rødsmoen, which is adjacent to Rena Army Camp. Rena Army Camp is one of the largest camps in the Norwegian Army, and is located in Åmot municipality.

A satellite image of the closed quarter's combat training ground is shown in Figure 7. Marked 1 in the map is where the operation centre was setup, as shown in Figure 8(a). It was a stationary container, and it was used for development and running the operation. Fraunhofer FKIE brought their own operation centre container, and this was placed at the point marked 2 in the map, and the interior is shown in Figure 9(b).

In order to test the interoperability between the system in a field setting, a basic scenario was developed. The four UGVs was set up in the staging area (marked 3 in Figure 7). The robots mission was to scout the area and look for possible IEDs, and disarm them if possible. The Telemax robot was the bomb disposal robot and the GVR-bot was used at an observer asset for the Telemax. The robots Olav and Garm was used as perimeter observers, in case any hostiles entered the area. Olav and Garm was first deployed, and during their deployment a person is observed running from the area marked 4 in Figure 7. The Telemax and GVR-bot was sent in to investigate, and a suspected IED was found. The Telemax robot attempted to disarm the IED.

Figure 7 Map over the experiment area at Rena, Norway (see online version for colours)

Figure 8 Control room setups used in the experiment, (a) control room setup in the stationary container (b) control room setup in the mobile container (see online version for colours)

(a) (b)

In the operation centre in the stationary container two operators used the OCUs from GVSC and PIAP. On the wall a projector displays a video feed from each robot. The video feed was set up using GStreamer, in order to have a low latency viewer for the streams. A cr een showing the location of all the robots in a map was also set up. Mapviz in ROS was used to show the map and the positions. Four ROS OCUs was started using the Fraunhofer FKIE ROS-IOP bridge, and the ROS RQT GUI provided with the bridge was used to be an observer for the four robots. For space and demonstration reasons, the robots were controlled during the final demonstration from the stationary container. The control system in the mobile container was used exclusively to monitor the systems.

Table 1 Overview over which interoperability messages that was tested for each robot in experiment

Robot	*PIAP OCU*	*GVSC UCU*	FKIE OCU
Garm			
Access control	✓	✓†	✓
mobility:PrimitiveDriver	✓	✓	✓
mobility:GlobalPoseSensor	✓	✓	✓
environmentSensing:DigitalVideo	✓	✓	✓
RTSP video server	✓	✓	✓
GVR bot			
Access control	✓	✓†	✓
mobility:PrimitiveDriver	✓	✓	✓
mobility:GlobalPoseSensor	✓	✓	✓
environmentSensing:DigitalVideo	✓	✓	✓
RTSP video server	✓	✓	✓
Telemax			
Access control	✓	✓†	✓
mobility:PrimitiveDriver	✓	✓	✓
mobility:GlobalPoseSensor	✓	✓	✓
environmentSensing:DigitalVideo	✓	✓	✓
RTSP video server	✓	✓	✓
Olav			
Access control	✓	✓†	✓
mobility:PrimitiveDriver	✓	✓	✓
mobility:GlobalPoseSensor	✓	✓	✓
environmentSensing:DigitalVideo	✓	✓	✓
RTSP video server	✓	✓	✓

Notes: The *access control* indicated if the OCU manage to take control of the robot or not. Also note that the *environmentSensing:DigitalVideo* message only contains an URI for the video. Therefore a *RTSP video server* is added to the table to indicate if actual video was successfully played in the UCU or not. The † sign indicated that this OCU required that the robot name was Packbot.

4.7 Interoperability results

During the experiment different messages was tested to be tranfered correctly from between the robots and OCUs. Table 1 give an overview of all the tested messages for each OCU group by robot, where the ✓ sign indicates that the message was successfully deliverd with the correct contents. Note that the urn:jaus:jss: prefix for the message types are omitted in the table. The table has a seperate *access control* entry. This indicates if the OCU managed to take control of the robot.

The DigitalVideo message in the IOP standard only contains a URI for where the video can be obtained. In the experiment an RTSP server URI was provided by all the robots, therefore a seperate *RTSP video server* entry in the table was added. A success for this entry means that the video was successfully played in the OCU.

5 Experiments in the ICARUS project

A series of validation trials conducted within the scope of the EU-FP7-ICARUS project (De Cubber et al., 2013) validated the interoperability concept in the context of a search and rescue application scenario. Within the ICARUS project, not less than ten different robotic systems (three aerial robots, three ground robots and four marine robots) had to communicate and collaborate for assisting human search and rescue workers (Doroftei et al., 2014). This description focuses on the contributions with respect to interoperability for the UGVs. The 'land' component of the ICARUS robotic search and rescue system was validated using a simulated earthquake response scenario, involving three aerial robots and three ground robots, as depicted in Figure 9.

5.1 Interoperability mechanisms used

In order to maximise the interoperability between the heterogeneous ICARUS fleet and to decouple between the custom implementations on the different robot platforms, the ICARUS project adopted an approach where the different robotic systems were adapted to a single standardised external interface. This standard interface is heavily based upon JAUS and UDP, but defines some extra features that are specifically required for heterogeneous search and rescue robotic systems, such as fleet management, real-time robot and sensor discovery and dynamic self-reconfiguring software-defined radio communications (López et al., 2017). As an example: all the aerial platforms use MAVLink as a messaging protocol, the large UGV uses a proprietary protocol, the medium uses ROS, while the small UGV uses the FinROC software architecture. All these robot-specific middlewares are interconnected through adapters to the ICARUS JAUS-based interoperability layer.

5.2 Results and validation

The ICARUS interoperability mechanism was validated using a simulated earthquake deployment exercise, where search and rescue workers from the Belgian B-FAST team engaged in real search and rescue operations, but helped by the robotic team agents. In order to fully integrate into standard operating procedures of the end users, the robotic tools were involved into each phase of the operation: from the deployment to the base of operations, to the mission planning, assessment, area reduction, victim search and victim rescue operations both in urban and semi-urban and CBRN scenarios. The following scenarios involving heterogeneous multi-agent robotic systems were validated:

- cooperative multi-stage aerial surveillance (UAV/UAV)
- aerial scouting for traversability analysis (UAV/UGV)

- victim search (UAV/UGV)
- carrier (UGV/UGV)
- communications relay (UAV/UXV).

Figure 9 Robotic tools used in the ICARUS earthquake response scenario, (a) solar aircraft (ETHZ)* (b) outdoor rotorcraft (Eurecat) (c) indoor rotorcraft (SkyBotix) (d) heavy UGV (Metalliance) (e) medium UGV (RMA)** (f) small UGV (University of Kaiserslautern)*** (see online version for colours)

Source: *Oettershagen et al. (2017), **De Cubber et al. (2014) and ***Berns et al. (2017)

A qualitative and quantitative validation methodology was employed in order to objectively measure the performance of the system. This showed that for more than 80% of the identified performance metrics, the ICARUS concept achieved a performance above the goal levels identified by the end users.

6 Discussion and future work

The IOP field trials in Rena can be considered successful for several aspects. The main goal of the trial was to verify that the UGV IOP standard could be used to both control different UGVs and receive information from these UGVs using different OCUs in a real environment. All the UGVs in the trial was successfully controlled using all the different OCUs at the trial.

The IOP field trial was also very informative for testing the ROS-IOP bridges. With the ROS-IOP bridge a ROS-based robot can provide IOP-compliant functions with very small effort. The ROS-IOP bridges were used on multiple new systems which proves the easy adaptation of the bridge to alternative ROS-based robots. The adaptation was quite simple and flexible: The software to provide a common position overview of the IOP systems [using a ROS-based GUI called Mapviz (http://wiki.ros.org/mapviz)] was implemented within a very short timeframe.

A possible challenge using one of the ROS-IOP bridges is that the control architecture may assume some specific signals or messages that are not provided by the bridge. For instance when using the bridge on Olav a signal from the bridge indicating if someone was actively controlling the vehicle or not was required. This functionality was added to the Fraunhofer FKIE ROS-IOP bridge for the UGV Olav. This shows that it is important to have some knowledge of the IOP standard and the applied bridge, even though the bridge hides most of the IOP standard specific details from a ROS user perspective.

Another aspect of the interoperability test was that different teams used different JAUS software toolkits. For instance the Fraunhofer FKIE ROS-IOP bridge was implemented using JAUS Toolset, while the PIAP OCU used OpenJAUS SDK. The usage of different JAUS software toolskits did not cause any technical problems.

Video stream compatibility between different platforms and OCUs was discovered as most important interoperability challenges. During the trials there were some difficulties streaming video from Olav to the PIAP OCU. This was resolved after selecting another H264 profile. Another issue with video streaming is that an RTSP server must be provided for streaming. If multiple users want to stream the same video feed, each of them will have a unicast connection to the server. This means that the bandwidth requirement increases linearly with the number of viewers. A mechanism for multicasting the video could in some cases be beneficial.

Another issue is that IOP is a large standard, so before the experiment all involved parties agreed on a subset of IOP messages that should be supported. This made it easier to get the robots to be IOP compliant, but came at a cost. The PrimitiveDriver was used to control the robots, but this may not be the best way to control an Ackermann vehicle, like for instance Olav.

The last issue that appeared was that some of the OCUs made assumptions that were not part of the standard. For instance the OCU provided by GVSC only accepted robots named ‘Packbot’. This caused us to name all robots ‘Packbot’ in the experiment, and made it harder to know where each robot was. Another issue was that one of the controllers only accepted a single video stream, even though two streams were available. This was easily solved by reducing the number of available streams.

For the experiment we defined a scenario that we played out a number of times. We had two operators controlling the four robots. In our opinion IOP provided a convenient way of controlling multiple robots. Additionally, to have all the video sources and all positions of all the robots available is very valuable for the operators when they are cooperating in performing a task. Test with military personnel is needed to verify these results.

6.1 Communication

During the setup phase of the trials communication problems were frequently observed, which in particular severely impaired the control of the robots. Identification of the problems was very difficult for several reasons.

Each robot system was controlled by its own radio system. In addition, some systems also had separate radio emergency stops. Although not all radio systems shared the same frequency range, the interference between the systems could not be ruled out. For precise measurements the appropriate equipment was not available. Ideally, all the robots should use the same radios and form a mesh network back to the operation centre. This might not be possible in practise. In our case we did not have a sufficient amount of radios, and each team was responsible for their own radio communication. In other cases the robotic system could be delivered with a radio. Also, routing could potentially be a problem when connecting the robots together. Many of the robots have their own internal network, and it might be available for the entire network. This could pose a problem when connecting robots together, as the internal networks might be on the same subnet and have conflicting IP addresses. This was solved in this experiment by placing a NAT router on the robot to mask the internal network away from the common network.

The video transmission is done in IOP via common standards, e.g., RTSP. While all other sensor and control data are passed through the JAUS node manager, the transmission of the audio-visual data (streams) is ‘parallel’ to it, causing the prioritisation mechanism of the IOP has little impact on the streams. As a result, the controlling OCU may lose control of the robot.

The current implementations of the JAUS node manager, which runs on every host and administrates the IOP communication, offer no possibilities to monitor the current network traffic and thus to provide helpful hints in case of problems. Providing information about current participants and their activity can provide evidence of problems, especially during development and integration, but also in operation. In IOP all subscribers are subscribed to one multicast group. Appropriate mechanisms to detect such and other disruptive activities at runtime would be very helpful in the JAUS node manager.

6.2 Future work

In the future, the focus will be on using higher level of autonomy functions such as waypoint navigation. This is addressed by the newly created IST-179 group in collaboration with the NAAG team of experts on UGV group.

7 Conclusions

In this paper we have shown how the IST-149 and team of expert UGV groups have tested the IOP standard using some of the messages defined in the standard. We found that a bridge component between the IOP and ROS domain made it faster to make the robots using ROS as middelware IOP compliant. It was also identified that lack monitoring of network traffic makes is hard to identify possible problems with the network.

Disclamer

The authors are responsible for the use of figures and the military equipment in this paper, and the publisher is not liable for any consequences arising out of this.

References

Ando, N., Suehiro, T., Kitagaki, K., Kotoku, T. and Yoon, W-K. (2005) 'RT-middleware: distributed component middleware for RT (robot technology)', in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, DOI: 10.1109/iros.2005.1545521.

Bergamaschi, F., Conway-Jones, D. and Pearson, G. (2013) 'Practical experience in deploying and controlling the data sharing interoperability layer at the U.K. Land Open Systems Architecture (LOSA) field trials in October 2012', in Pham, T., Kolodny, M.A. and Priddy, K.L. (Eds.): *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR IV*, SPIE, May, DOI: 10.1117/12.2016518.

Berns, K., Nezhadfard, A., Tosa, M., Balta, H. and Cubber, G.D. (2017) 'Unmanned ground robots for rescue tasks', in *Search and Resuce Robotics – From Theory to Practice*, 23 August, pp.53–76, InTech, ISBN: 978-953-51-3376-6, DOI: 10.5772/intechopen.69491.

Bounker, P. and Volk, A. (2015) 'NATO LCG UGV ToE interoperability experiences', in *Proceedings of AVT-241 Specialists' Meeting on Technological and Operational Problems Connected with UGV Application for Future Military Operations*, NATO Science and Technology Organization, April, DOI: 10.14339/STO-MP-AVT-241, NATO UNCLASSIFIED.

Bruyninckx, H., Soetens, P. and Koninckx, B. (2003) 'The real-time motion control core of the Orocos Project', in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, IEEE, DOI: 10.1109/robot.2003.1242011.

De Cubber, G., Balta, H. and Lietart, C. (2014) 'Teodor: a semi-autonomous search and rescue and demining robot', in *Advanced Concepts on Mechanical Engineering (ACME)*, DOI: 10.4028/www.scientific.net/amm.658.599.

De Cubber, G., Doroftei, D., Serrano, D., Chintamani, K., Sabino, R. and Ourevitch, S. (2013) 'The EU-ICARUS project: developing assistive robotic tools for search and rescue operations', in *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, IEEE, pp.1–4, DOI: 10.1109/ssrr.2013.6719323.

Doroftei, D., Matos, A. and De Cubber, G. (2014) 'Designing search and rescue robots towards realistic user requirements', in *Advanced Concepts on Mechanical Engineering (ACME)*, DOI: 10.4028/www.scientific.net/amm.658.612.

Elliott, L., Williams, N. and Siddapureddy, V. (2012) 'Open Management Group Data Distribution Service (OMG-DDS) as a data transport for vehicular integration for C4ISR/EW interoperability (VICTORY) services', in *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium*, National Defense Industrial Association, August.

Elliott, L., Jenkins, S.P., Moore, M.S. and Yee, H.S. (2019) 'Potential for VICTORY and FACE alignment – initial exploration of data interoperability and standards compliance/conformance', in *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium*, National Defense Industrial Association, August, pp.13–15.

Haavardsholm, T.V., Smestad, R., Larsen, M.V., Thoresen, M. and Dyrdal, I. (2016) 'Scene understanding for autonomous steering', in *Proceedings of IST-127/RSM-003 Specialists' Meeting in Intelligence & Autonomy in Robotics*, NATO Science and Technology Organization, Bonn, Germany, October, DOI: 10.14339/STO-MP-IST-127, NATO UNCLASSIFIED.

Heffner, K., Pullen, J.M., Simonsen, K.J., Schade, U., Reus, N.D., Khimeche, L., Zevassvik, O-M., Brook, A. and Veiga, R.G. (2010) 'NATO MSG-048 C-BML final report summary', *Fall Simulation Interoperability Workshop*.

Henshaw, M. (2011) *Assessment of Open Architectures within Defence Procurement Issue 1: Systems of Systems Approach Community Forum Working Group 1 – Open Systems and Architectures*, Report, Loughborough University, January [online] https://hdl.handle.net/2134/8828.

Kim, J., Smereka, J.M., Cheung, C., Nepal, S. and Grobler, M. (2018) *Security and Performance Considerations in ROS 2: A Balancing Act* [online] https://arxiv.org/abs/1809.09566 (accessed 10 December 2020).

López, D.S., Moreno, G., Cordero, J., Sanchez, J., Govindaraj, S., Marques, M.M., Lobo, V., Fioravanti, S., Grati, A., Rudin, K. et al. (2017) 'Interoperability in a heterogeneous team of search and rescue robots', in *Search and Rescue Robotics-From Theory to Practice*, InTech, Rijeka, DOI: 10.5772/intechopen.69493.

Langerwisch, M., Wittmann, T., Thamke, S., Remmersmann, T., Tiderko, A. and Wagner, B. (2013) 'Heterogeneous teams of unmanned ground and aerial robots for reconnaissance and surveillance – a field experiment', in *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, IEEE, October, DOI: 10.1109/ssrr.2013.6719320.

Marques, M.R.M. (2018) *Reference Model for Interoperability of Autonomous Systems*, PhD thesis [online] http://hdl.handle.net/10362/57144.

Maruyama, Y., Kato, S. and Azumi, T. (2016) 'Exploring the performance of ROS2', in *Proceedings of the 13th International Conference on Embedded Software – EMSOFT'16*, ACM Press, DOI: 10.1145/2968478.2968502.

Mathiassen, K., Baksaas, M., Olsen, L.E., Thoresen, M. and Tveit, B. (2016) 'Development of an autonomous off-road vehicle for surveillance missions', in *Proceedings of IST-127/RSM-003 Specialists' Meeting on Intelligence & Autonomy in Robotics*, NATO Science and Technology Organization, Bonn, Germany, October, DOI: 10.14339/STO-MP-IST-127, NATO UNCLASSIFIED.

Mathiassen, K., Hyndøy, J.I., Østevold, E., Valaker, S., Danielsen, T., Baksaas, M., Olsen, L.E., Thoresen, M., Ruud, E-L., Selvåg, J. and Sandrib, J. (2019) *Base Defence Demonstration at Trident Juncture 2018 – TACT Unmanned Systems for Base and Force Protection*, FFI-Report 19/00807, Norwegian Defence Research Establishment [online] http://hdl.handle.net/20.500.12242/2618.

NATO (2012) *NATO, STANAG 4586, Standard Interfaces of UAV Control System (UCS) for NATO UAV Interoperability*, 3rd ed.

NATO (2018a) *STANAG 4754, NATO Generic Systems Architecture (NGVA) for Land Systems*, 1st ed., February.

NATO (2018b) *AEP-4754 NATO Generic Systems Architecture (NGVA) for Land Systems, Data Model*, 1st ed., February, Vol. 5.

NATO (2018c) *AEP-4754 NATO Generic Systems Architecture (NGVA) for Land Systems, Safety*, 1st ed., February, Vol. 6.

NATO (2018d) *AEP-4754 NATO Generic Systems Architecture (NGVA) for Land Systems, Crew Terminal Software Architecture*, 1st ed., February, Vol. 6.

Oettershagen, P., Melzer, A., Mantel, T., Rudin, K., Stastny, T., Wawrzacz, B., Hinzmann, T., Leutenegger, S., Alexis, K. and Siegwart, R. (2017) 'Design of small hand-launched solar-powered uavs: From concept study to a multi-day world endurance record flight', *Journal of Field Robotics*, Vol. 34, No. 7, pp.1352–1377, DOI: 10.1002/rob.21717.

Object Management Group (OMG) (2015) *Data Distribution Service (DDS), Version 1.4*, Object Management Group [online] http://www.omg.org/spec/DDS/1.4 (accessed 10 December 2020).

Open Systems Joint Task Force (2004) *Modular Open Systems Approach to Acquisition, Version 2*, September.

Ota, D., Charchalakis, P. and Stipidis, E. (2017) 'Towards a verification and validation test framework for open system architectures', in *2017 International Conference on Military Technologies (ICMT)*, pp.115–122, DOI: 10.1109/MILTECHS.2017.7988742.

Pearson, G. and Kolodny, M. (2013) 'U.K. MoD Land Open Systems Architecture and coalition interoperability with the U.S.', in Pham, T., Kolodny, M.A. and Priddy, K.L. (Eds.): *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR IV*, International Society for Optics and Photonics, SPIE, Vol. 8742, pp.71 – 79, DOI: 10.1117/12.2016131.

Pradhan, M., Tiderko, A. and Ota, D. (2017a) 'Approach towards achieving interoperability between military land vehicle and robotic systems', in *2017 International Conference on Military Communications and Information Systems (ICMCIS)*, May, pp.1–7, DOI: 10.1109/MILTECHS.2017.7988788.

Pradhan, M., Tiderko, A. and Ota, D. (2017b) 'Approach towards achieving an interoperable C4ISR infrastructure', in *2017 International Conference on Military Technologies (ICMT)*, May, pp.375–382, DOI: 10.1109/ICMCIS.2017.7956477.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R. and Ng, A.Y. (2009) 'ROS: an open-source robot operating system', in *ICRA Workshop on Open Source Software*, Kobe, Japan, Vol. 3, p.5.

Remmersmann, T., Tiderko, A., Schade, U., Langerwisch, M. and Thamke, S. (2013) 'Smart control and detection feedback for a multi-robot border control system', in *Proceedings of 8th Future Security 2013 Security Research Conference*, Fraunhofer, Berlin, ISBN: 978-3-8396-0604-9.

Rowe, S. and Wagner, C.R. (2008) *An Introduction to the Joint Architecture for Unmanned Systems (JAUS)*, Technical Report, Cybernet Systems Corporation [online] http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.472.489 (accessed 10 December 2020).

Schneider, F.E., Wildermuth, D. and Wolf, H-L. (2015) 'ELROB and EURATHLON: improving search & rescue robotics through real-world robot competitions', in *2015 10th International Workshop on Robot Motion and Control (RoMoCo)*, July, IEEE, DOI: 10.1109/romoco.2015.7219722.

Schneider, F.E., Baksaas, M., Bounker, P., De Cubber, G., Główka, J., Kozik, R., Mathiassen, K., Nussbaumer, T., Röning, J. and Tiderko, A. (2020) *Capability Concept Demonstrator for Interoperability within Unmanned Ground Systems and C2*, Technical Report STO-TR-IST-149, NATO Science and Technology Organization, DOI: 10.14339/STO-TR-IST-149.

Seehuus, R.A., Mathiassen, K., Ruud, E-L.M., Simonsen, A.S. and Hermansen, F. (2019) 'Battle management language for robotic systems', in *Modelling and Simulation for Autonomous Systems, MESAS 2018, Lecture Notes in Computer Science*, Vol. 11472, pp.302–320, Springer International Publishing, ISBN: 978-3-030-14984-0, DOI: 10.1007/978-3-030-14984-0_23.

Serrano, D., De Cubber, G., Leventakis, G., Chrobocinski, P., Moore, D. and Govindaraj, S. (2015) 'ICARUS and DARIUS approaches towards interoperability', in *IARP RISE Workshop, Proceedings of the NATO STO Lecture Series SCI-271*, Citeseer, Lisbon, Portugal [online] http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.734.9768 (accessed 10 March 2020).

UK Ministry of Defence (2013) *Defence Standard 23-09 – Generic Vehicle Architecture*, No. 3, Defence Equipment and Support, UK Defence Standardization.

Wiig, M.S., Løvlid, R.A., Mathiassen, K. and Krogstad, T.R. (2016) 'Decision autonomy for unmanned vehicles', in *Proceedings of IST-127/RSM-003 Specialists' Meeting on Intelligence & Autonomy in Robotics*, NATO Science and Technology Organization, Bonn, Germany, October, DOI: 10.14339/STO-MP-IST-127, NATO UNCLASSIFIED.