ROBOPS - APPROACHING A HOLISTIC AND UNIFIED INTERFACE SERVICE DEFINITION FOR FUTURE ROBOTIC SPACECRAFT

Steffen Jaekel, Bernhard Brunner [(1)]
Christian Laroque, Zoran Pjevic [(2)]
Felix Flentge [(3)]
Thomas Krueger, and André Schiele [(4)]

[(1)] German Aerospace Center (DLR), Muenchner Str. 20, 82234 Wessling, Germany
[(2)] Telespazio VEGA Deutschland GmbH, Europaplatz 5, 64293 Darmstadt, Germany
[(3)] European Space Operations Centre, Robert-Bosch-Straße 5, 64293 Darmstadt, Germany
[(4)] European Space Research and Technology Centre, Keplerlaan 1, 2201 AZ Noordwijk, Netherlands
Email: firstname.lastname@{dlr, telespazio-vega, esa}.{de, de, int}

ABSTRACT

This paper describes a unified and holistic approach to the identification and definition of interface services and protocols for future robotic spacecraft. Hardware-in-the-loop (HiL) demonstration results are outlined based on the implementation of selected end-to-end services. The developed communication interface is intended to facilitate the command, control and monitoring of classic satellites as well as attached robotic devices and robotic mobile platforms. Both system autonomy and distributed mission architectures are promoted. Based on an initial state-of-the-art review of current and past robotic space missions, required capabilities with respect to communication, levels of robotic control and autonomy were investigated. Thereof derived, a general categorization of possible robotic missions was developed, including the definition of roles, responsibilities and major use cases. By applying a hierarchical mission composition, a definition of functional classes and a classification of autonomy levels, a systematic and holistic categorization could be found to the definition of the required services. The concept can be applied to arbitrary hardware and deal as a standard for on-board, spacecraft-to-spacecraft and ground-spacecraft communication. Subsequently, suitable technologies for the definition and implementation of these services were analyzed and a conceptual architecture was developed. As underlying communication protocols and architectures, various options have been evaluated. A disruption-tolerant network (DTN)-based solution was chosen, however, the defined services can work over a variety of different communication protocols. For demonstration purposes, a subset was implemented within the METERON operations environment (MOE) [17] and demonstrated with two different robotic devices, a 7-degrees of freedom (DoF) dexterous manipulator and a sample-collecting rover mockup. The experiments showed that the developed architecture can successfully be used to control robotic manipulators and rovers over DTN in a standardized way.

1. INTRODUCTION

Mainly relying on the packet utilization standard (PUS) [1] and partly new trends towards file-based operations [2], current monitoring and control (M&C) architectures in the European context have been purposely designed for controlling classic spacecraft such as Earth-observing or telecommunication satellites. However, for future space missions such as robotic on-orbit servicing (OOS) [3] or the exploration of planetary surfaces, where the satellite or rover turns into a highly integrated *space robot*, the established standards reach the edge of their capabilities with respect to successfully being able to control and monitor the remote system in an efficient way [4]. For complex use cases such as the capture of a tumbling and uncooperative target [5], the operator on ground is not able to perceive all required information in a timely manner. In general these limitations become particularly clear when complex autonomous systems and communication architectures with large roundtrip times (RTT) and potential disruptions are involved. A more suitable complement with respect to interfaces, protocols and service definitions is required, including a clear distinction between methods of data transport and communication data semantics. The operator should be able to efficiently command and observe the spacecraft on all applicable levels of autonomy.

2. STATE OF THE ART

The effective operation of robotic spacecraft becomes an increasingly important topic, as there is a multitude of applications in this domain. Using the Shuttle and Space Station Robotic Manipulator System (SRMS, SSRMS) [6, 7], the International Space Station (ISS) was assembled out of several modules by applying the principle of *in-space robotic assembly (ISRA)* [8]. The major advantage of ISRA is that it allows to overcome launcher limitations with respect to size and mass. Small robotic satellites are planned to serve for inspection purposes [9] and NASA's Robonaut [10] or comparable systems such as DLR's humanoid robot

Justin [11] are candidates for future *EVA support operations*. Similar to ISRA and EVA support, dexterous robotic manipulators are planned to be utilized to capture, maintain and/or de-orbit operational and defective satellites within *on-orbit servicing* (OOS) missions [12]. The Robotic Refuelling Mission [13] recently demonstrated technologies required for OOS using the Special Purpose Dexterous Manipulator (SPDM) [14] aboard ISS. Finally, *robotic exploration* of other celestial bodies, such as the Moon, Asteroids [15], Near Earth Objects (NEOs), or Mars is envisaged or has already been accomplished, e.g. with the Mars Exploration Rovers (MERs) [16].

Current spacecraft are mostly controlled from a single ground station transporting telemetry (TM) and telecommand (TC) by packets as defined in the packet utilisation standard (PUS). Future complex and potentially distributed robotic missions require a suitable complement in terms of communication (interfaces, protocols and service definitions) and control approaches. First attempts in this direction are currently made with the METERON project [17]. METERON features a distributed mission control architecture with shared human and robot control strategies within the Supvis-Justin [18] experiment and real-time haptic control within the Haptics-1 and 2 experiments [19]. Integrated autonomous spacecraft control architectures for future robotic spacecraft are currently developed [20] and would benefit from a more efficient and function-oriented external communications approach. Standardization in this regard has evolved to some extend in the past yielding a new generation of Consultative Committee for Space Data Systems (CCSDS) space protocols hat include the CCSDS File Delivery Protocol (CFDP) [21], the Asynchronous Message Service (AMS) [22] and the Disruption-Tolerant Network (DTN) [23] standards as well as newly defined mission operations services with the Message Abstraction Layer (MAL) [24, 25], the Monitor and Control-Common Services (MO) [26], and the Operations Common Object Model (COM) [27] in order to adapt to new challenges such as disrupted networked communication, distributed monitoring and control (M&C) mission architectures, and the integration of real-time control.

3. SCENARIO ANALYSIS

As shown in the previous chapter, there exist some ongoing standardization efforts. However, the current standards and control approaches for spacecraft suffer from a combination of the typical space community conservativeness and old heritage that allow to somehow fit new technology but will never be optimal and effective in this regard. For new challenges such as the control of complex robotic spacecraft that are embedded in distributed mission architectures, they are not sufficient. Unfortunately, even the new generation of standards that are still being developed do not focus on robotic spacecraft and miss the ability to effectively control the spacecraft and its components on arbitrary levels of autonomy in a similar fashion.

In order to find relevant operations use cases for future robotic missions, analyses regarding communication, different levels of autonomy and control were conducted. Subsequently, multiple detailed scenarios were created based on typical use cases and dealt as basis for the definition of required interface services.

3.1. Communication & Control Modes

Figure 1 shows an exemplary communication setup for robotic spacecraft. In case of a telepresent control authority, e.g. for OOS spacecraft, the operator on ground steers the robot in space (teleoperator) by issuing force and position commands and receiving multimodal feedback. For other missions such as celestial exploration, different control approaches with a higher level of system autonomy and decreased interaction possibilities apply due to decreased communication link quality (jitter, packet loss) and time delay. While telepresent control allows very fine-grained manipulation as well as increased robustness and flexibility with respect to unknown dynamic environments, increased roundtrip time (RTT) in the communication channel restricts the operator's ability to effectively interact with the remote environment. The properties of the communication channel highly correspond to the existing distance as dominant barrier to be breached. Additionally, space and ground-based network instances can potentially decrease the link quality and have to be optimized for a given type of mission. Pure distance ranges from approx. 250 km in low Earth orbit (LEO), over 36,000 km in geostationary orbit (GEO), 350,000-400,000 km for the Moon and finally up to 50-400 million km for Mars. The time required for communicating simply due to the speed of light and additional network instances as stated above varies from approx. 10ms up to more than one hour. This circumstance highly influences feasible control modes. In addition, achievable communication windows are of high importance. When communicating from earth to a spacecraft in LEO or vice versa, direct communication is limited to approx. 10 minutes per fly-

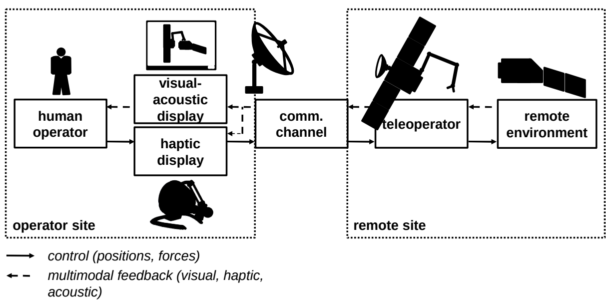

Figure 1. Communication infrastructure

Autonomy Level	Possible Control mode	Max. RTT
0	Telepresent control	< 800ms
1	Teleoperation/Shared control with supervision of fine-grained operations	< 2.5s
2	Teleoperation/Shared control with supervision of coarse-grained operations	< 25s
3	Fully self-contained autonomous control	> 25s

Table 1. Autonomy levels with connected possible control modes and required roundtrip time

by. It has been shown, that by using a geostationary relay satellite, stable communication with a delay of up to 800ms and a window of approx. 40min can be accomplished [28]. This represents the maximum limit for manual, non-model based telepresent operations with respect to the current state of the art. Higher communication delays therefore result in control approaches that involve a higher level of autonomy. For interplanetary communication and by using the approach of a deep space delay-tolerant network (DTN) the RTT will increase respectively. However, by utilizing the principle of (multiple) nodes, i.e. relay satellites or stationary landers on the surface, in connection with DTN technology, stable deep-space communication can be achieved over a very long time window, while the decreased properties of RTT and jitter will only allow robotic operations with a highly autonomous control mode when operated from Earth.

3.2. Operator Roles & Responsibilities

In order to define potential roles and responsibilities for robotic missions, an analysis and comparison of the organizational structure of typical missions operated by the German Space Operations Center (GSOC) and the European Space Operations Center (ESOC) has been performed. Apart from other roles, three major shareholders have been identified for the subsequent scenario analyses: the Spacecraft Operations Manager (SOM) / Flight Director, the Spacecraft Operations Engineer (SOE) / Subsystem Engineer, and the Spacecraft Controller (SPACON) / Command Operator (CMD). By moving from classical spacecraft to space robots, this structure can be transferred by introducing the Robotics Operations Manager (ROM), Robotics Operations Engineer (ROE), and Robot Operator (RO), respectively. For classic missions the required time scale for decision making usually spans from minutes to hours. For robotic missions, however, the reaction time is reduced to seconds in the case of supervised and shared autonomy or up to fractions of a second in the case of telepresent operations. Consequently, not all commanding can be initiated or authorized via the ROM or the ROE, especially if the voice-loop is included. A direct control authority between the spacecraft and the RO has to be introduced. Consequently, the ROM/ROE rather set high-level goals and supervize the ongoing actions performed by the RO.

3.3. Detailed Scenario Analysis

Based on the state of the art with respect to robotic applications in space, and above described assessment and classification of communication architectures, control modes and required actors, a detailed use case analysis has been conducted that defined and structured typical actions for OOS spacecraft and exploration rovers. Subsequently, several detailed scenarios were designed. They include an OOS mission in LEO, autonomous and free-flying nano-satellites outside the International Space Station (ISS) dealing as extra-vehicular activity (EVA) support, as well as a distributed Mars rover scenario based on the METERON project, where the robot can either be controlled from Earth or from a planetary orbiter by astronauts. The scenarios listed a timely order of actions issues by the various actors and autonomous agents in order to fulfil the mission. Both autonomous operations, e.g. sample collection and retrieval by a Mars rover, and teleoperation, e.g. capturing a tumbling target spacecraft in LEO, are included.

As given with the METERON scenario, several M&C systems, located at different centres on Earth together with an orbiting station around Mars, need to be coordinated together in order to monitor and control the robotic systems concurrently. While the operators on Earth need to be informed about the ongoing operations and issue coarse-grained actions, the Astronaut located aboard the orbiter may act as a RO with the ability to control the robot on the Mars surface by the means of telepresence. Potentially, a team of robots with different individual capabilities needs to be controlled by multiple actors in a similar and coordinated fashion, which raises the question of command authority. In contrast to the classical M&C approach, services do not only need to exchange messages for atomic and immediate information, e.g. parameter values, commands, acknowledgments, but also files for larger, self-contained operations and more permanent data, e.g. scientific observations, software patches, on-board procedures, or command procedures.

Based on the evaluation of the required actions within the use cases and thereof created scenarios combined with the capabilities of existing robot M&C systems and the requirements for future distributed mission architectures, an initial plain list of required services and combined functionalities could be derived. The next chapter presents a grouping of these services aiming to find a holistic and functional approach to the control of robotic spacecraft.

4. DEFINITION OF SERVICES

The challenge with defining a common list of services lies within the novel approach of combining the operation of completely different fields and systems, i.e. platform control, robotics, arbitrary sensors etc., in one standardized approach. However, in their core, they can all be traced back to the same three groups of *functionality*, namely

1. Control
2. Monitoring, and
3. Configuration.

In the context of mission operations, every autonomous system has parameters that need to be configured. Parameters, modes and derived states represent a way of abstraction, the autonomous system can operate on. Additionally, monitoring is required for the system itself to perceive its state and for the operator to receive information about the ongoing mission, the spacecraft and all of its subsystems or devices. Opposed to monitoring, control is necessary in order to master the system.

In addition, the robotic spacecraft mission *architecture* as described in the scenario analysis needs to be considered in order to find a common structure of what exactly is to be commanded, configured and monitored. Figure 2 (left) depicts the parts that comprise a space-robotic mission. There may be several systems, i.e. autonomous or partly autonomous spacecraft such as a robotic satellite, rover or manned orbiter. Each system may contain several subsystems, e.g. a robotic manipulator, a sample return device or a GPS receiver. Additionally, all systems of a mission assemble a specific hierarchy, cp. Figure 2 (right). The definition of hierarchy is important for knowing which system can be controlled on a high level of autonomy by the operator to achieve an appropriate chain of commands to all other systems. The design of hierarchy may also be reflected in how the actual communication infrastructure of the mission is implemented and vice versa.

Finally, the service specification should also follow the implemented system *autonomy*. Figure 3 depicts a model for autonomous systems commonly known as three-tier (3T) architecture. The deliberative layer maintains the world model and is responsible for planning. It provides tasks to the second layer, the so-called sequencing or executive layer. The sequencer divides the tasks from the deliberative layer into single commands for the reactive layer and monitors the execution of these commands. The reactive layer contains all the system functions which are necessary for executing the commands received from the sequencer to finally achieve the goal of the deliberative layer. The three layers are each directly connected to the sensory input and action output. The sequencing layer thereby holds and observes the execution of the plan. The plan contains encapsulated units called actions. Each action comprises preconditions and expected postconditions as result of the action execution. Postconditions may represent preconditions for subsequent actions to be executed. One typical type of condition is time, with the connected actions together forming a mission timeline (MTL) or time schedule. However, arbitrary types of conditions are considerable, such as orbit position, termination of a previous action, manual trigger etc.

Figure 4 presents a service composition matrix that takes the aforementioned categorization into account. The ordinate (y-axis) represents the three layers of the 3T autonomy model: the reactive, sequencing and deliberative layers. The abscissa (x-axis) represents the architecture as depicted in Figure 2, which is split into mission, system and subsystem. Each defined service has to cover the three groups of functionality: control, monitoring and configuration. The leftmost column represents the operator that can interact with the systems on all layers of the 3T model in every level of system architecture. There can be common services, such as monitoring, e.g. realized by a flat (non-hierarchical) data management, that implements similar functionality on all three levels of autonomy and is therefore depicted in a separate row. Apart from the centralized monitoring service that represents a rather explicit way of monitoring, i.e. the software components share dedicated monitoring data with this service (that is for example implemented within a central database solution), services also include an implicit way of monitoring by responses to invocation of service operations. In general, the y-axis represents the capacity of the system to solve some task autonomously, whereas the x-axis represents what is done (control, monitor, configure) at which point in the architecture (mission, system, subsystem). The area with white background in the scheme shows the composition of classical satellite

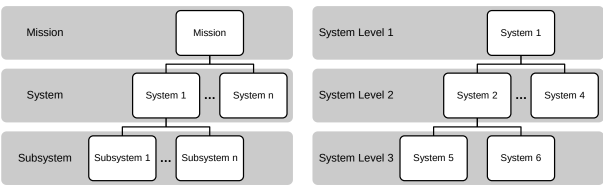

Figure 2. Hierarchy in a distributed mission architecture

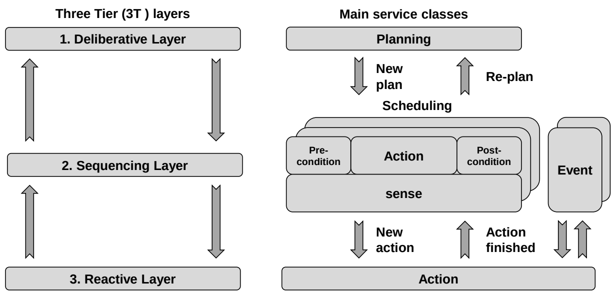

Figure 3. Three-Tier (3T) approach to system autonomy

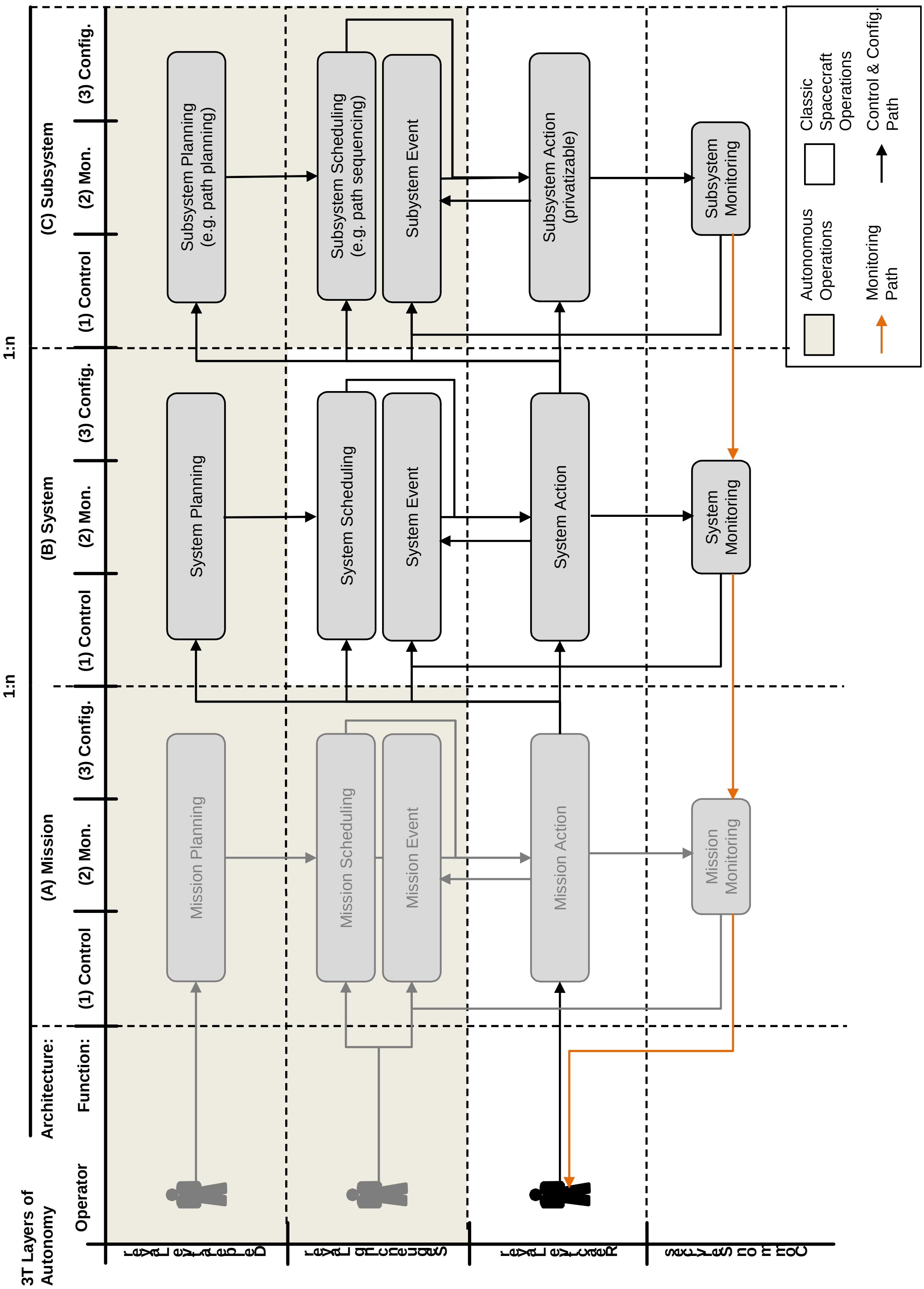

Figure 4. Services composition matrix – classification of services based on the 3T autonomy architecture, functionality groups and mission architecture

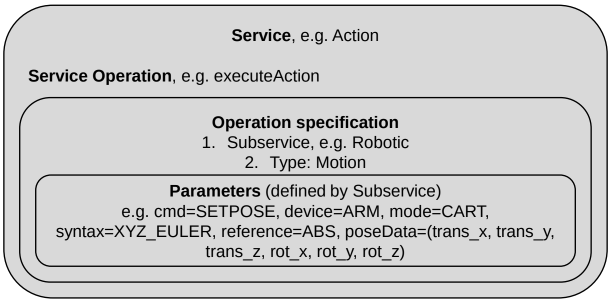

Figure 5. Example of RobOps service operation

operations, where mostly reactive functions exist on-board of the spacecraft and commands are issued manually by the operator or are partly triggered by the scheduling service holding the mission timeline. The area with grey background depicts services that need to be added on top of classic satellite operations in order to achieve a higher level of autonomy. In general, the given services can be located at different places within a distributed mission concept, e.g. on ground, in an additional orbiter, or at the target spacecraft itself. Black arrows in the scheme indicate control and configuration, whereas orange arrows indicate monitoring. The complete mission column is greyed out. The respective services are named in order to show a complete picture of what is needed to achieve mission autonomy, but are not within the scope of analysis conducted within this study.

For each service, required *service operations* that can be issued by the operator, were defined for fulfilling the service's full functionality. In contrast to existing standards, the found categorization concentrates on the autonomy aspect rather than on the specific device or platform function that needs to be controlled. The aim was to find one interface specification for autonomous systems that can be applied to arbitrary hardware and deal as a standard for on-board, spacecraft-to-spacecraft and ground-to-spacecraft communication. Specific functionality, e.g. a robotic arm or thermal control subsystem interface, would be a so-called *service privatization* specified for each operation. For typical devices that are commonly used, defined privatizations may eventually become a standardized *subservice* specification that is used across missions. In addition to service privatization/subservices, a *type* specifier is used for detailed function specification within the chosen subservice interface. Figure 5 depicts the describe service breakdown. In this context, various subservice interface specifications have already been defined, e.g. "Robotic", "Collaboration", "System" amongst others.

5. IMPLEMENTATION

5.1. Technology Analysis

Subsequently, suitable technologies for the definition and implementation of the defined services were analyzed and a conceptual architecture was developed. As underlying communication protocols and architectures, various options have been evaluated including the message-based solutions DDS [29], ActiveMQ, ZeroMQ, DTN, and AMS [30], as well as the file-based solutions CFDP and multiple well-established ground protocols. A more detailed evaluation is given within [33]. Finally, a solution with MAL over DTN (using the Bundle protocol and Licklider transmission protocol) was chosen for prototype implementation.

5.2. Architectural Design

Figure 6 presents an overview of the architectural design for the demonstration prototype. The Message Abstraction Layer (MAL) for message encoding and decoding was adapted to use the ION DTN stack as transport layer. On top of this, the Service Layer interfaces with MAL and provides a Java API to the RobOps service adapter that understands and implements the functionality as defined in the

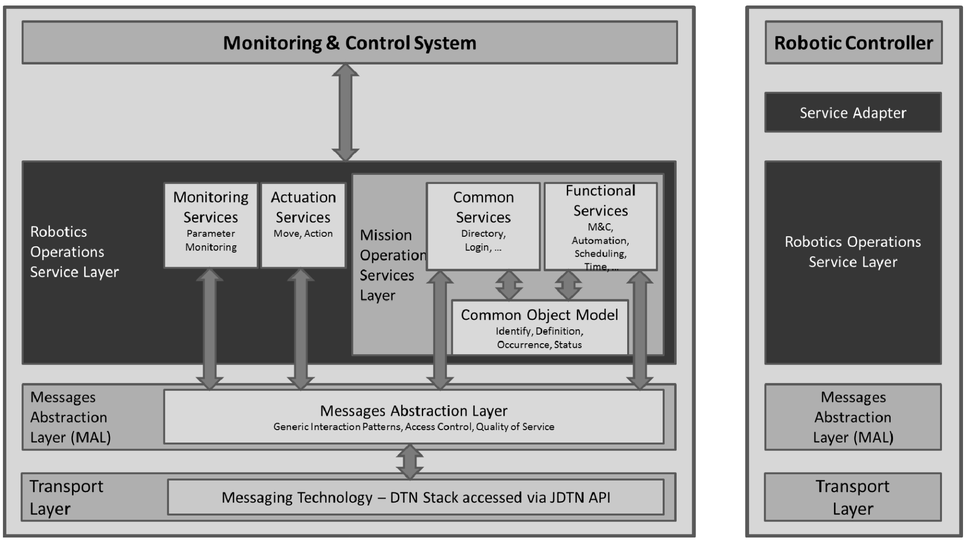

Figure 6. Prototype architectural design

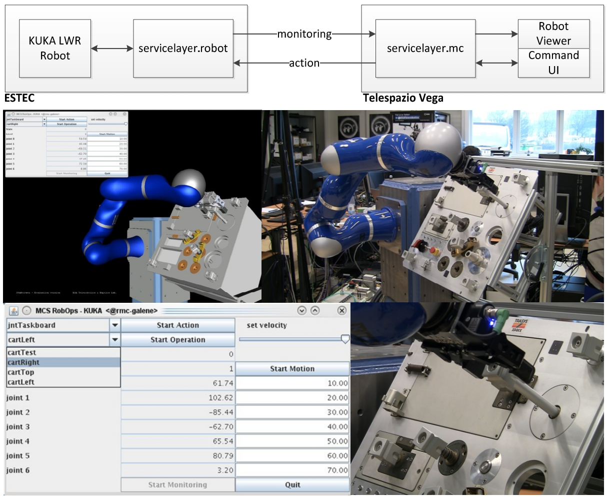

Figure 7. Final demonstration setup (top) with KUKA lightweight robot (LWR) at ESTEC (right) and the M&C GUI and 3D viewer at Telespazio (left)

subsystem specification. On the operator side, a Java-based monitoring and control system with a graphical user interface (GUI) was developed to directly interface with the service layer. For demonstration purposes, the *Monitoring* and *Action* services were chosen as subset to be implemented.

6. DEMONSTRATION

The described setup was demonstrated with two different robotic devices: the KUKA lightweight robot (LWR), a 7-degrees of freedom (DoF) dexterous manipulator [32] and the MOCUP rover from Telespazio. Figure 7 (top) shows the final demonstration with the manipulator setup being located at the Telerobotics & Haptics Laboratory at ESTEC and the M&C setup at Telespazio in Germany. The LWR was connected to the service adapter via the KUKA Fast Research Interface (FRI) over local DDS using UDP as transport protocol. Figure 7 (bottom, right) shows the robot together with a task-board it is attached to. Several predefined poses could be chosen and executed in the M&C GUI, the current joint status was displayed together with a 3D graphical representation using ESTEC's SPANviewer [33], cp. Figure 7 (bottom, left). In addition, user-defined joint motions could be defined in the M&C GUI using a joint value template. That way parameterized actions could be set and sent to the robot servicelayer for immediate execution. The service adapter subsequently decomposes the incoming actions into several steps and activates the respective controller and interpolator to generate the joint set-point values for the real-time robot control at system clock frequency.

7. SUMMARY AND FUTURE WORK

In contrast to classical service definitions such as the packet utilization standard (PUS) which was introduced to standardize ground-to-space communication between the operator and the satellite platform, the RobOps architecture tries to go one step further and present a holistic and functional approach to robotic spacecraft control that can be extended in the future by adding new service privatizations and adapting existing ones. The experiments showed that the developed architecture can successfully be used to control robotic manipulators over DTN in a standardized way. Future work will include the implementation of further services such as the planning and scheduling service, in order to broaden the capabilities of the HiL-demonstrator. In addition, by adding artificial delays and disruptions to the communication channel, the robustness and flexibility of the chosen DTN transport method can be verified. The implemented services may also be used in future METERON experiments, where astronauts aboard ISS act as robotic operators, controlling robotic devices on Earth.

REFERENCES

1. Ground systems and operations — Telemetry and telecommand packet utilization (PUS), ECSS-E-70-41A, Jan. 2003, URL: http://www.ecss.nl [cited 27 March 2014].

2. Haddow, C. R., Pecchioli, M., Montagnon, E. and Flentge, F., "File Based Operations – The Way Ahead?", SpaceOps 2012

3. Hirzinger, G., et al., DLR's robotics technologies for on-orbit servicing. Advanced Robotics, 2004. Special Issue Service Robots in Space: p. 139–174.

4. S. Jaekel, B. Scholz, Utilizing Artificial Intelligence for Achieving a Robust Architecture for Future Robotic Spacecraft, Aerospace Conference, IEEE, Big Sky, MT, USA (March 2015).

5. S. Jaekel et al., Robotic Capture and De-Orbit of a Heavy, Uncooperative and Tumbling Target in Low Earth Orbit, ASTRA 2015 - 13th Symposium on Advanced Space Technologies in Robotics and Automation (May 2015)

6. Aikenhead, B.A. et al. (1983), Canadarm and the space shuttle. Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films

7. Gregor, R., & Oshinowo, L. (2001). Flight 6A: Deployment and Checkout of the Space Station Remote Manipulator System (SSRMS). In Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS).

8. Mohan, S. and D.W. Miller, SPHERES Reconfigurable Framework and Control System Design for Autonomous Assembly, in AIAA Guidance, Navigation, and Control Conference. 2009: Chicago, USA.

9. Stoll, E., Jaekel, S., Katz, J., Saenz-Otero, A. and

Varatharajoo, R. (2012), SPHERES interact—Human–machine interaction aboard the International Space Station. J. Field Robotics, 29: 554–575. doi: 10.1002/rob.21419

10. Diftler M.A. et al. (2012) Robonaut 2 - Initial Activities On-Board the ISS. 2012 IEEE Aerospace Conference, Big Sky, MT.

11. Zacharias, F. et al. (2010) Exploiting Structure in Two-armed Manipulation Tasks for Humanoid Robots. the IEEE International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan.

12. Sellmaier, F. et al. (2010), On-orbit Servicing Missions ar DLR/GSOC, in 61st International Astronautical Congress. Prague, CZ.

13. Jill McGuire (2013), NASA's Robotic Refueling Mission, 2nd Annual ISS Research and Development Conference, Greenbelt, USA.

14. Mukherji, R. et al. (2001) Special Purpose Dexterous manipulator (SPDM) Andvanced Control Features and Development Test Results. in Proceedings of the 6th International Symposium on Artificial Intelligence and Robotics and Automation in Space (i-SAIRAS). Montreal, CA

15. Kuninaka, H., & Kawaguchi, J. I. (2011, March). Lessons learned from round trip of Hayabusa asteroid explorer in deep space. In Aerospace Conference, 2011 IEEE (pp. 1-8). IEEE

16. Biesiadecki, J.J.; Maimone, M.W., "The Mars Exploration Rover surface mobility flight software driving ambition," IEEE Aerospace Conference (2006), doi: 10.1109/AERO.2006.1655723

17. Bosquillon de Frescheville, F., Martin, S., Policella, N., Patterson, D., Aiple, M., Steele,P.: Set-up and Validation of METERON End-To-End Network for Robotic Experiments, Proceedings of 11th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA), April 2011.

18. Neal Y. Lii et al. (2015), Simulating an Extraterrestrial Environment for Robotic Space Exploration: the METERON Supvis-Justin Telerobotic Experiment and the Solex Proving Ground, ASTRA 2015 - 13th Symposium on Advanced Space Technologies in Robotics and Automation

19. A. Schiele et al. (2015), Preliminary Findings from the Haptics-1 Space Experiment, ASTRA 2015 - 13th Symposium on Advanced Space Technologies in Robotics and Automation

20. S. Jaekel, M. Stelzer, H.-J. Herpel (2014), Robust and Modular On-Board Architecture for Future Robotic Spacecraft, Aerospace Conference, IEEE, Big Sky, MT, USA.

21. File Delivery Protocol (CFDP), Blue Book, CCSDS 727.0-B-4, 2007-01, URL: http://public.ccsds.org/publications/archive/727x0b4.pdf(cited May/03/2015)

22. Asynchronous Message Service, Blue Book, CCSDS 735.1-B-1, 2011-09, URL: http://public.ccsds.org/publications/archive/735x1b1.pdf (cited May/03/2015)

23. DTN, URL: http://sourceforge.net/projects/ion-dtn/ (cited May/03/2015)

24. Operations Message Abstraction Layer, Blue Book CCSDS 521.0-B-1, 2010-10, URL: http://public.ccsds.org/publications/archive/ 521x0b1.pdf

25. ESA MAL Implementation, URL: https://github.com/esa (cited May/03/2015)

26. Monitor and Control-Common Services, Draft Recommended Standard, CCSDS 521.1-R-1, 2007-09, URL: http://public.ccsds.org/sites/cwe/rids/Lists/CCSDS%205211R1/Attachments/521x1r1.pdf (cited May/03/2015)

27. Operations Common Object Model, Draft Recommended Standard,CCSDS 522.0-R-2, 2011-09, URL: http://public.ccsds.org/sites/cwe/rids/Lists/CSDS%205220R2/Attachments/522x0r2.pdf (cited May/03/2015)

28. E. Stoll et al. (2009), Ground Verification of the Feasibility of Telepresent On-Orbit Servicing, Journal of Field Robotics, 26(3), 287 – 307.

29. Data Distribution Services, OMG Standard, URL: http://www.omg.org/spec/DDS/1.2/ 1.2 (cited May/03/2015)

30. Asynchronous Message Service, Blue Book, CCSDS 735.1-B-1, 2011-09, URL: http://public.ccsds.org/publications/archive/735x1b1.pdf (cited May/03/2015)

31. F. Flentge, S. Jaekel, B. Brunner, C. Mateo, P. Steele (2014), RobOps - Services for Telerobotic System Operations, 13th International Conference on Space Operations, Pasadena, CA, USA

32. Hirzinger, G. et al. (2002), DLR's torque-controlled light weight robot III-are we reaching the technological limits now?, Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on

33. Kimmer, S., Rebelo, J., Schiele, A. (2013) SPANviewer - AVisualization Tool for Advanced Robotics Applications, 12th symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)