TOWARDS MODULAR COMPONENTS AS BUILDING BLOCKS FOR APPLICATION-SPECIFIC CONFIGURABLE SPACE ROBOTS

Roland U. Sonsalla[1], **Wiebke Brinkmann**[1], **Henning Wiedemann**[2], **Malte Langosz**[1], **Priyanka Chowdhury**[2], **Christopher Schulz**[1], **Marko Jankovic**[1], **Tobias Stark**[1], **Moritz Schilling**[1], **Niklas Mulsow**[1], **Daniel Pizzutilo**[1]

[1] *German Research Center for Artificial Intelligenz (DFKI) - Robotics Innovation Center, Bremen, Germany*

[2] *University of Bremen - Robotics Research Group, Bremen, Germany*

ABSTRACT

Robots are becoming increasingly important in future space missions. Instead of building separate monolithic systems, the idea is to develop specialized and standardized modules that can then be (re-) combined to set up a robot for specific tasks. If necessary, this robot can also be reconfigured during runtime to adapt to different mission objectives and widen its area of application. The emphasis here is on the overall formulation of such a modular building block system, which encompasses technological, mechatronic and software design. The developments will enable the realization of a modular robot configuration and implementation, as well as the ability to reconfigure the system online. To give an outlook of how the concept can handle a real application a proof-of-concept evaluation mission is defined.

Key words: Modular Robotics, Building Blocks, EMI, Reconfiguration.

1. INTRODUCTION

For service operations on satellites or the exploration of foreign planets, robotic systems will be of particular interest in the future. Until now, the space system solutions developed and deployed have been highly mission-specific. Modular systems that can be combined to perform specific mission tasks should enable a rapid response to future exploration and service missions. The modularity of spacecraft or planetary robots defines the level of subdivision of a system in standardized and easily replaceable units, interconnected with each other or with the main bus via a relatively simple standard interface [1]. These units can contain any number of replaceable system components such as inertial reference units, payload, electronics, power distribution units, batteries, etc. [2].

Over the years, different levels of spacecraft modularity have been implemented, ranging from highly integrated, specialized systems, to highly modular ones, comprised entirely of a large number of small modules. Typical spacecraft generally consist of many individual components, whose integration and interfaces are highly optimized towards mass and cost reduction. Therefore, they are not easily serviceable on-orbit, if at all. One step closer towards advanced modularity is represented by the minimally modular spacecraft, such as those being part of families of commercial communication spacecraft. They are generally composed of two to three large modules that allow parallel integration and testing (I&T) and provide significant cost savings but not necessarily servicing [1].

The serviceable modularity or modularity at the component level represents an even higher level of modularity then the previous one. Examples of spacecrafts with this level of modularity are the Hubble Space Telecope (HST) and International Space Station (ISS), equipped with serviceable components and standard interfaces. However, these components are not grouped into serviceable modules, meaning that any On-Orbit Servicing (OOS) task would need to be performed at a component level with tools and procedures specifically developed for each component separately. This complication can be avoided by developing systems with a subsystem level of modularity, consisting mainly of components integrated into modules which can be easily removed/replaced on-ground as well as on-orbit. Examples of such type of spacecraft are the Multimission Modular Spacecraft (MMS), the SolarMax spacecraft, and the Reconfigurable Operational spacecraft for Science and Exploration (ROSE). They contain components grouped into serviceable modules, integrated on the main bus via a standardized interface, thus allowing a great deal of flexibility both on-ground, during I&T activities, and on-orbit, while keeping the complexity of those tasks at the minimum [1].

The intelligent Building blocks for On-orbit Servicing (iBOSS), Autonomous Assembly of a Reconfigurable Space Telescope (AAReST), DARPA's Satlets and Self Assembling Wireless Autonomous and Reconfigurable Modules (SWARM), are designed with an even greater spacecraft modularity in mind. In these concepts the overall spacecraft system is composed of small interconnected modules, each providing only a fraction of functionality of a traditional spacecraft, comparable to cells in a living organism. Modules are envisioned to be interconnected via intelligent plug-and-play interfaces, allowing almost total on-orbit reconfiguration and assembly, with the highest level of flexibility in mind [1]. The type and number of individual modules will be based on an optimization process that will depend not only on engineering metrics, such as the cost and mass, but also on other less quantifiable metrics, such as future market uncertainties/projections and influence of stakeholders [3, 4, 5].

In line with the typical spacecraft design, planetary rovers are generally composed of many individual, highly integrated components, not meant for serviceability nor repairability, but with ruggedness and redundancy in mind. In fact, currently deployed Mars rovers Spirit, Opportunity and Curiosity, are highly specialized, single mission systems, conceived to be mobile laboratories to single-handedly carry out all the required exploration tasks. However, these systems are proving to be inappropriate for future large-scale exploration missions of planetary surfaces, where coordinated, modular, multi-robot systems will play a pivotal role.
The payload-items (PLIs) developed at DFKI-RIC Bremen represent one existing solution for such systems able to support robot-to-robot interactions in multi-robot scenarios through the usage of an electro-mechanical interface (EMI) [6, 7, 8]. Over the last years various standard interconnects for orbital and potentially planetary applications have been studied and developed including the design of modular robotic components that can be (re)configured via a standard interconnect to be used for space specific applications [9, 10]. Along with the further advancements of the plans to return to the moon, concepts of modular robotic system designs come into realization, such as Astrolab's Flex rover [11].

Instead of building separate monolithic systems, the idea is to elaborate specialized and standardized modules that can be (re-)combined to setup a robot for specific tasks; if necessary, this robot can also be reconfigured during runtime to adapt to different objectives. Such a modular building block system, as outlined in Fig. 1, is being developed in the project MODKOM (Modular Components as Building Blocks for Application-specific Configurable Space Robots), which unifies both, specially developed components according to a standardized building block systematics as well as industrial third-party commercial off-the-shelf (COTS) components.

The building block systematic is described in Sec. 2 and is used to define the modular building blocks. Besides developing mechatronic and software modules, designed for various space and terrestrial missions, a software tool kit is implemented to support non-expert users composing those modular functional units to robotic systems, as discussed in Sec. 3. MODKOM continues the work begun in preceding projects like the X-ROCK [1] projects which also aimed at long-term autonomy through model-based, holistic robot development for use in space, offshore and human-shared environments.

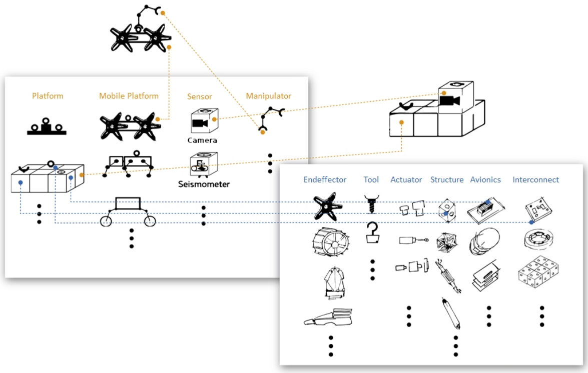


Figure 1. Flexible use of the modular building blocks for the realization of mission-specific systems and requirements

The performance of this technology, the methods, and the modules developed in this project will subsequently be evaluated by the realization of a complex composite system for mobile manipulation. An outline of the functional realization of the modular toolbox is given in Sec. 4. The paper concludes with a summary and outlook of future work in Sec. 5.

2. MODULAR BUILDING BLOCK SYSTEM

To formulate standardized module descriptions and achieving a coherent building block systematic, various robotic systems and application scenarios have been analyzed and used to derive a reasonable level of module granularity. The scope of application was evaluated based on a functional decomposition of various robotic reference systems, including rover systems and in orbit servicing space crafts. Different concepts for analyzing and defining modules within a modular toolbox have been studied [12, 13]. Here the Metus principle as described in [12] was used to identify and group reoccurring functions. With the help of Metus, it is easy to find the functional relationships of existing components and visually represent them so that they can be grouped into functional modules. The method can be applied very early during the design phase, which is why it is best suited for the MODKOM project. This analysis builds the baseline for the definition of general top-level requirements in order to define the scope of the modular toolbox as well as the level of system modularization and granularity - for both hardware and software - , as described in the following sections.

---

[1] D-Rock: `https://robotik.dfki-bremen.de/en/research/projects/d-rock.html` and Q-Rock: `https://robotik.dfki-bremen.de/en/research/projects/q-rock.html`

2.1. Top-Level Requirements

For the layout and definition of the modular construction kit, a set of top-level requirements was identified. The formulation of these requirements is based on the functional analysis of existing systems. Furthermore, the state-of-the art of e.g. modular spacecraft designs is taken into account as well as the needs of future use case scenarios and mission concepts, as described by agencies and industry.

In the scope of this paper, only the main functional objectives of the modular toolbox are outlined:

Application area The modular toolbox shall be designed to support future robotic activities in various space applications, i.e. ranging from an orbital manipulator use case towards mobile planetary exploration and implementation of infrastructure. Therefore, different relevant use cases shall be evaluated and analyzed with respect to common function blocks. A proof of concept of the modular tool kit shall be demonstrated by means of a representative application scenario in terrestrial environment.

Modularization and interconnectivity The modularization of the different toolbox modules shall be generally formalized in a way to formulate rather a users and/or interface guide than a complete set of modules. The system modularization shall allow the configuration and integration of a complete system using modules of the tool kit. Furthermore, a reconfiguration of specific system parts shall be possible during the run time of the system. It shall be possible to extend the tool kit with additional modules. These can either be newly developed, following the tool box guide line or already existing third party components, which may be integrated either by adapting to the toolbox interfaces or by introduction of adapter elements. The proof of concept shall include the realization and application of various modules from the tool kit on all levels of granularity to form a robotic system, as well as the integration of third party robotic components and/or modules.

Hardware design The hardware realization of modules shall allow for clearly distinguished module functions. Where necessary cable routing and load oathes shall be kept as short as possible. The modules shall allow an easy integratability and be compatible to each other. For reconfiguration purposes an SI (Standard Interconnect) shall be applied, supporting a compatibility to the module definition.

Electromechanics design Standardized power requirements and standard communication protocol, including the integration of SI shall be applied for all modules. For reconfiguration purposes, the de- and re-coupling of modules shall be supported, as well as the identification and handling of modules in the communication bus.

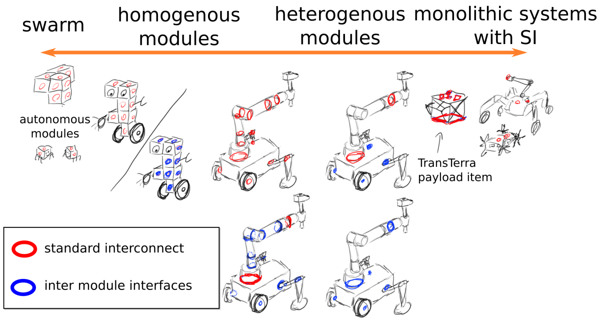


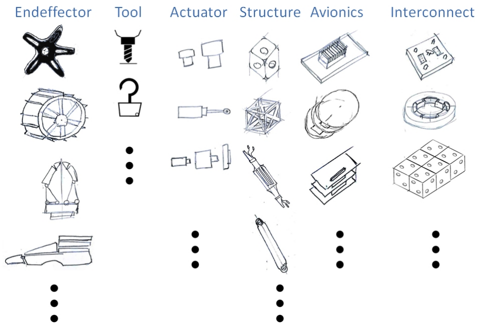
Figure 2. Overview of granularity spectrum of modular systems

Software design It shall be possible to perform a system check for consistency on software level. The software framework has to enable automatic start and stop actions of tasks and recognize changing system functions due to reconfiguration. Where applicable error massages shall be generated to allow system monitoring and FDIR (fault detection, isolation and recovery). Furthermore, software modules shall be set-up to allow compatible module representation, composition and cataloging. System configuration and generation shall be supported by software on a user-friendly level.

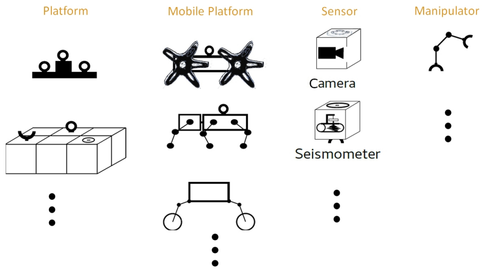
2.2. Modularization and Granularity

A central aspect of a modular tool kit is the granularity of the modules. This encompasses not only the character and topology of the modules, but also the connectivity between them. The goal is to elaborate which functions the individual modules should cover, as well as which modules should be compatible with each other. The spectrum of the granularity of modular systems is shown in figure 2, where one extreme are swarm robots and the other end of the spectrum is bounded by monolithic rovers. Swarm robots usually consist of a large number of identical systems that are all compatible (i.e. can be connected) to each other. They all have the same functionality and can only perform higher level tasks by working together and/or by forming more complex topologies by arranging themselves in new patterns. They are, in theory, extremely flexible but are complex to control. Compared to monolith systems they create a certain mass overhead, because all robotic systems of the swarm are also part of the load-bearing structure and every subsystem is redundant.

Monolithic systems on the other hand are specifically build to fulfil a defined task and therefore, are (ideally) the optimal solution to a given problem or task. This limits their flexibility and the development of a monolithic system is usually time consuming and expensive. Due to



(a) Basic functional units for system configuration



(b) Alternative system modules, e.g. for reconfiguration

Figure 3. Schematic representation of two different layers of granularity, here for offline system configuration and online system reconfiguration

the complexity and limited functionality of swarm robots and the high development effort of monolithic systems, a more balanced approach is proposed: To reduce cost and time in the development of new systems, a toolbox with standardized modules will be developed. The system can be configured and built using these modules that are equipped with and connected by standardized electrical and mechanical interfaces. This also allows to extend the tool kit with new modules. To make the system more flexible, at least one active electromechanical interface is included in the tool kit. This interface allows the online reconfiguration of the system to fulfil a wide range of tasks and to even extend its functionality by providing new modules or components, once they are equipped with at least one interface.

For the design of a modular system kit, a clear definition of the interfaces between the individual modules is essential. Based on the targeted granularity the building block systematic distinguishes between two different layers of modules: a) modules for system configuration (offline) (cf. Fig. 3(a)) and b) modules for system reconfiguration (online) (cf. Fig. 3(b)). For interconnecting the modules of the tool kit, a distinction is made between two primary types of interfaces: inter-modular interfaces (IMIs) and standard interconnects (SIs).

The IMIs are intended to enable mechanical and electronic connections between modules. To facilitate maintenance and testing, the connections should be detachable. They are used to connect modules during configuration and integration of the overall system. Due to the expected differences in size and performance, especially of structural modules and actuators, at least three different sizes of mechanical IMIs should be defined. Mechanical and electronic connections use separate interfaces to prevent damage to electronic connectors during mechanical assembly, to facilitate maintenance, and to provide more flexibility in the design of the system. For flexibily in communication speed and the use of industrial third-party COTS components within the robotic system two different communication interfaces are envisioned to be supported: Ethernet and CAN.

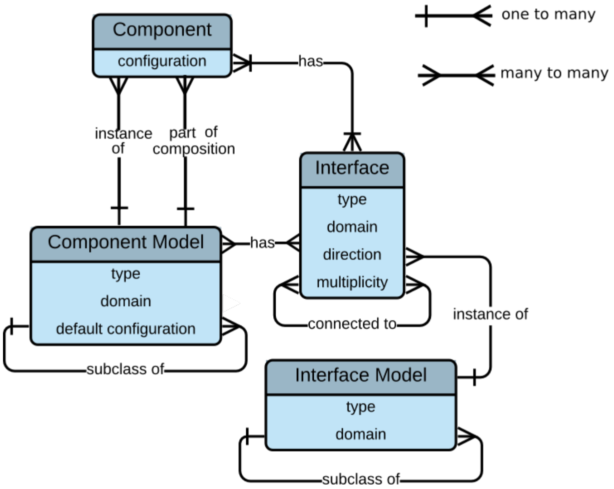


Figure 4. Overview of module representation. The type boxes list their most important properties, while the arrows show possible relations to other type instances and cardinality constraints.

In contrast to the IMI, the SI is a multifunctional interface consisting of a mechanical locking mechanism and interfaces for data and electrical power transmission. This multifunctional interface can be controlled by the system during operation. The system can be reconfigured online or reconfigure itself via the SI. The SI can be used as both an end-effector on the robot arm for handling different tool attachments or payloads, and as an interface for tethering payloads to the robot body. This allows the system not only to act flexibly according to the situation, but also to be used for different tasks within the mission. The electro-mechanical interface (EMI) developed at DFKI [14] is such an SI and will be further developed in MODKOM.

3. SOFTWARE ARCHITECTURE

To represent and manage the various types of building blocks required for the construction of modular robots, a generic type capable of meeting these requirements has been introduced. The following paragraphs will provide an explanation of this type, as well as how components are defined and used.

3.1. Component Models, Components and Modules

The representing type set called *XTypes* specializes into several subtypes, from which *Component* and *ComponentModel* are the most important ones for the subjects in scope of this project. The presented concepts are based on and extend [15].

Components represent the hard- and software building blocks of robotic systems, which can be combined to generate more complex components. Hence, a hierarchy of components of different complexity is created. At the lowest level of this hierarchy are *atomic components*, which can not be further divided into other components in our model.

To properly construct a component-based system, it is necessary to distinguish between the following:

Component models (type name *ComponentModel*) which form the type system of components,

Components (type name *Component*) which always are instances of some component model and can be used to construct new component models, and

Modules are components that exist in the physical world (e.g. have been built by someone). Modules - even if they are identical - have a unique id; so each physically existing module can be identified.

Interfaces (type name *Interface*) define how components can be interconnected (for software this means porting of data, for hardware this means the way they are assembled)

Interface models (type name *InterfaceModel*) are - like component models - the type system of interfaces

A component cannot exist without a component model but a component model can exist without a component. The handling of modules contributes to the expansion of the existing database as well as the development of a software solution by modeling and storing the hardware module life cycle. This allows the tracking of real hardware instances of the modeled components.

3.2. Database

A database has been implemented in the project Q-ROCK (see [15]) that is a combination of hand-curated ontologies and a graph database. The database contains information about known hard- and software components and component models, as well as their relationships, such as component-interface compatibility and the structure of available robotic systems. The database serves as a central repository for each stage's results, allowing data to be used immediately across all workflow steps, resulting in a fully integrated development workflow. The next paragraphs shall give an overview about the usage of the software tooling.

3.3. Building Block Creation

This is the initial step in assembling a modular kit. For the hardware side, the Blender add-on phobos [16] will be used to generate new atomic component models from CAD data and add them to the database of component models. Larger artifacts that cannot be stored in a database will be moved to an external repository and referenced in the component model (e.g. kinematic representation). For the software side, default ROCK-tasks are deployed automatically into the database via a continuous integration job that extracts the task interfaces and creates its component model.

3.4. Building Block Representation & Description

The hardware representations will be stored using the SMURF file structure, which includes the following: The kinematic structure as URDF including the interface frames, the geometries as meshes, geometry annotations (such as the closest primitive description and the smallest number of points required to adequately describe the geometry using a convex hull of those) definitions of loop closures and the definitions of motor and sensor.

Depending on whether the component is public, either the blueprints or information on how to order this component will be included in the database.

Customizable components are specified via the configuration when instantiating a component model. When assembling the module to the system, this adaptation will be done in accordance to the transformation information provided during the system configuration. GUIs developed in the D/Q-ROCK-projects make this more user-friendly as described in the following paragraphs.

3.5. Composing an Assembly

Out of existing (atomic) components in the database, new component models are created with the hereinafter described tools. This is the case for both hard- and software component models.

DEIMOS is a 3D visualization of the hardware components and their interfaces. By selecting two interfaces and performing necessary transformations, hardware components can be assembled and the resulting component model can be stored in the database.

X-ROCK-GUI (see Fig. 5) accomplishes the same task, but it does so by displaying components as nodes and allowing you to establish connections between them by drawing edges in a graph view. It is a more in-depth tool which also enables configuration of components.

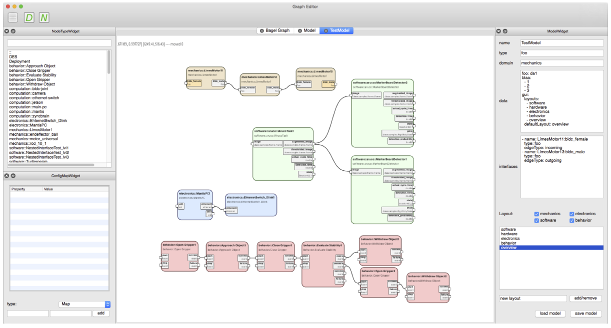


Figure 5. Graphical interface to visualize, assemble, and configure component models.

3.6. Software, Data & Hardware Integration

Finally, creation of buildconf and bundles (as a package to be installed in the system) is required to setup the software on the system. Like Components and ComponentModels, the buildconf for the system is versioned too. In this buildconf all versions (both, OS-dependencies and source packages) shall be fixed. This way, versions can be switched easily and safely.

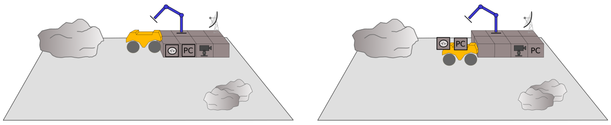
An installable buildconf consists of 3 components: All package_sets, software layout (manifest) and the bundle with ROCK-config files. The command line interface of Phobos can be used to construct and derive all necessary representations for all hardware compositions.

A mapper tool maps the individual software components to available execution hardware and generates the CND (component network description). System deployment tools use the resulting mapping to bundle and compile the ROCK deployments. Those include solutions for: The mapping process from software components to executional hardware, CND generation out of this map, automatic generation of all needed deployments, calls to ROCK-runtime as well as starting software from CND (start, configure, wire).

Eventually, everything will be collected in order to provide a simple method of installing the software to the robot: The software for the system will be provided as directory, which can be saved to a bootable storage medium. It will contain both the corresponding Ubuntu distribution as well as a script to checkout and install the software.

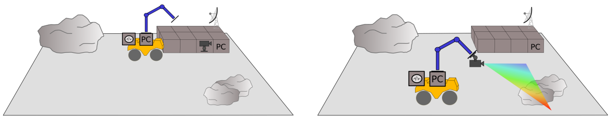
4. FUNCTIONAL REALIZATION

For proof-of-concept purposes an application scenario is designed, allowing to evaluate all levels of the defined modular tool kit in a real robotic application. Two scenarios are therefore envisioned; one to evaluate the system configuration layer and one for the operation and reconfiguration of a composite system.



(a) Initial setup. Only base station active.

(b) Base station assembles mobile system.



(c) The arm mounts it-self and then the RGB-camera.

(d) Mobile system is put to operation.

Figure 6. Envisioned demo scenario for modular system operation at project end.

The first scenario will show the functionality of the MODKOM tooling workflow by composing and integrating a (mobile) manipulator based on modules from the tool kit. This includes both utilizing existing hardware and software components from the database as well as adding new components of both domains to the database. Then the components are connected using the GUIs described in section 3.5. For the software side, this is accomplished by connecting of data interfaces and for the hardware side, by specifying which component will be assembled where in relation to the other components. In both domains, it is possible to specify which modules may be exchanged during operation. Following that, both domains' components will be configured. The task structure can then be created using the CND tools by referencing the component structure created in the preceding steps. Finally, it will be demonstrated how the created setup can be saved to the database and then installed and integrated into the system via a bootable USB drive.

The general outline of the second scenario, the evaluation mission is depicted in Fig. 6. A modular base station will be used to assemble a previously defined mobile, modular system. The demonstration will begin with both systems operational and their respective software stacks installed (see Fig. 6(a)). Then (see Fig. 6(b)) the base station uses its arm module to assemble the second mobile system by installing a computational unit (that has its prepared software stack also set-up). When the system is assembled onto the mobile platform, it is detected and the computational unit initiates the necessary tasks to control the platform, eventually restarting it. (The remaining modules are installed in the subsequent steps, while the computational unit accepts and restarts necessary tasks to incorporate the added hardware's software into the running task structure.) When those modules are installed, the arm module of the base station must also connect to the mobile platform via the interface at its end-effector (see Fig. 6(c)). As soon as it is connected (in hardware), the base station will switch of the arm, and the mobile platform will connect to it in the software domain. The arm can now be used to pick up and install a camera module onto the now free interface at the end of the arm, completing the mobile system's preparation for exploration and map generation (see Fig. 6(d)).

5. SUMMARY AND OUTLOOK

With the goal to facilitate a paradigm shift in the development and creation of robotic systems for robotic space applications, the design and set-up of a modular construction kit was outlined. Based on the current state of the art and future application scenarios, a definition approach has been described for the modular building block system. In this study, an analysis of existing robotic systems was carried out in order to identify common functional groups and modules. Along with a set of general top-level requirements for the modular tool box, the definition of system modularization and level of granularity is presented. A multi-dimensional modular building block approach is proposed in this paper, which relies on the definition of inter-module interfaces (IMI) and standard interconnects (SI). This allows on the one hand, to define modules, applying the IMI, for configuring various robotic systems based on a set of predefined modules. On the other hand, a system reconfiguration can be introduced by adding SI to the system, allowing reconfiguration on payload level as well as on system level. The presented modular toolbox allows to setup a robot for specific tasks by using specialized and standardized modules. These modules can be (re-)combined and if necessary, reconfigured during run-time to adapt to a variety of objectives.

On software side, a generic solution has been found to adequately represent hard- and software components that form the foundation of all software tools. Moreover, based on this digital twin of the real hardware, the MODKOM pipeline can automatically build and deploy the robot control software to the systems completing the work flow for non expert users. Regarding the software toolbox, this project's ongoing development will advance previous projects' software tools to provide user-friendly tools for adding, editing, creating, and maintaining components. It could be COTS or custom-developed hardware and software components. Thus, in addition to introducing the first modules for a plug-and-play building-block kit, MODKOM introduces a completely new method for developing such systems.

As development progresses, the database for the newly created building-block-system will be populated with additional modules that expand the range of possible application fields. Along with the module definition and design, further steps are being taken to consider the special requirements for space applications at an early stage. Using the modular building block system as an example, a modular robotics joint module (DFKI-X2D) will be further developed and evaluated in terms of space constraints.

This project not only provides operating hardware and software modules, but also a toolbox that enables non-expert users to assemble application-oriented, reconfigurable robots, thereby simplifying and speeding up their development. By defining modules and standards, the results from MODKOM will help in the future to provide flexibly configurable solutions that can be adapted to new or changing requirements with minimal effort. Rather than having to carry out a completely new development every time, the provided technology benefits not only the development cycle but also the sustainability in space robotics and thus, can counter space debris from two sides.

ACKNOWLEDGMENT

The authors would like to thank the MODKOM team and all supporting staff at DFKI Robotics Innovation Center as well as University of Bremen Robotics Research Group. The work presented is part of the project MODKOM, which is funded by the german space agency (DLR) with federal funds of the Federal Ministry for Economic Affairs and Climate Action in accordance with the parliamentary resolution of the German Parliament under grant no. 50RA2107 and 50RA2108.

REFERENCES

[1] D. Rossetti, B. Keer, J. Panek, B. Ritter, B.B. Reed, and F. Cepollina. Spacecraft Modularity for Serviceable Satellites. In *AIAA SPACE 2015 Conference and Exposition*, Reston, Virginia, 8 2015. American Institute of Aeronautics and Astronautics.

[2] C.M. Reynerson. Spacecraft modular architecture design for on-orbit servicing. In *2000 IEEE Aerospace Conference. Proceedings*, volume 4, pages 227–238. IEEE.

[3] A.A. Kerzhner, M.D. Ingham, M.O. Khan, J. Ramirez, J. De Luis, J. Hollman, S. Arestie, and D. Sternberg. Architecting Cellularized Space Systems using Model-Based Design Exploration. In *AIAA SPACE 2013 Conference and Exposition*, pages 1–24, Reston, Virginia, 9 2013. American Institute of Aeronautics and Astronautics.

[4] D. Sternberg, S. Arestie, R. Harris, N. Ricci, M. A. Schaffner, C. Whitlock, and D. Miller. A Bottom-up Modeling Approach for the Profit Analysis of Cellularized Spacecraft Architectures. In *64th International Astronautical Congress (IAC)*, pages 1–11, Beijing, China, 2013. International Astronautical Federation.

[5] B. Karlow, C. Jewison, D. Sternberg, S. Hall, and A. Golkar. Tradespace investigation of strategic design factors for large space telescopes. *Journal of Astronomical Telescopes, Instruments, and Systems*, 1(2):027003, 4 2015.

[6] W. Brinkmann, F. Cordes, T. M. Roehr, L. Christensen, T. Stark, R. U. Sonsalla, R. Szczuka, N. A. Mulsow, and D. Kuehn. Modular Payload-Items

for Payload-assembly and System Enhancement for Future Planetary Missions. In *2018 IEEE Aerospace Conference*, pages 1–10, Big Sky, Montana, USA, 2018. IEEE Comput. Soc. Press.

[7] R.U. Sonsalla, F. Cordes, L. Christensen, T.M. Roehr, T. Stark, S. Planthaber, M. Maurus, M. Mallwitz, and E.A. Kirchner. Field testing of a cooperative multi-robot sample return mission in Mars analogue environment. In *Proceedings of the 14th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA 2017)*, Leiden, the Netherlands, June 2017. ESA/ ESTEC.

[8] R.U. Sonsalla, F. Cordes, L. Christensen, S. Planthaber, J. Albiez, I. Scholz, and F. Kirchner. Towards a heterogeneous modular robotic team in a logistic chain for extraterrestrial exploration. In *Proceedings of the 12th International Symposium on Artificial Intelligence, Robotics and Automation in Space – i-SAIRAS 2014*, Montréal, Canada, June 2014.

[9] M. Jankovic, W. Brinkmann, S. Bartsch, R. Palazzetti, and X. Yan. Concepts of active payload modules and end-effectors suitable for standard interface for robotic manipulation of payloads in future soace missions (SIROM) interface. In *Proceedings of the 2018 IEEE Aerospace Conference*, Big Sky, Montana, March 2018. IEEE.

[10] S. Estable, A. Ampe, A. Chamos, G. Aridon, D. Silveira, F. J. Colmenero Lechuga, I. Soto, J. Gancet, M. Shilton, M. Jankovic, and T. Vogel. PERIOD – PERASPERA in-orbit demonstration toward the transition into the in-space services, assembly and manufacturing paradigm. *IOP Conference Series: Materials Science and Engineering, IOP Publishing*, 1226:1–8, February 2022.

[11] Astrolab. *FLEX Payload Interface Guide*. Astrolab Vebturi, v 1.0 edition, January 2022.

[12] I. Renner. *Methodische Unterstützung funktionsorientierter Baukastenentwicklung am Beispiel Automobil.* Dissertation, Technische Universität München, München, 2007.

[13] MB Collaborations - part of Modular Management. Developing modular systems in collaboration - successfully reducing complexity. Technical report, https://mb-collaborations.com/en/modular-system, 2022.

[14] W. Wenzel, F. Cordes, and F. Kirchner. A robust electro-mechanical interface for cooperating heterogeneous multi-robot teams. In *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2015)*, pages 1732–1737, Hamburg, Germany, October 2015. IEEE/RSJ. ISBN: 978-1-4799-9994-1.

[15] T.M. Roehr, D. Harnack, H. Wöhrle, F. Wiebe, M. Schilling, O. Lima, M. Langosz, S. Kumar, S. Straube, and F. Kirchner. A development cycle for automated self-exploration of robot behaviors. *AI Perspectives*, 3(1):1–29, 2021.

[16] K. von Szadkowski and S. Reichel. Phobos: A tool for creating complex robot models. *Journal of Open Source Software*, 5(45):1326, 2020.