73rd International Astronautical Congress (IAC), Paris, France, 18-22 September 2022.

IAC-22-D.1.1.x69259

Toolbox Design to Demonstrate Application-Specific Configurable Space Robots Using Modular Components

Roland U. Sonsalla [a], Henning Wiedemann [b], Wiebke Brinkmann [a], Priyanka Chowdhury [b], Malte Langosz [a], Marko Jankovic [a], Christopher Schulz [a], Tobias Stark [a], Moritz Schilling [a], Niklas A. Mulsow [a], Daniel Pizzutilo [a], Leonard Maisch [a], Patrick Schoeberl [a]

[a] *Robotics Innovation Center, German Research Center for Artificial Intelligence (DFKI), Robert-Hooke-Str. 1, 28359 Bremen, firstname.lastname@dfki.de*
[b] *Robotics Research Group, University of Bremen, Robert-Hooke-Str. 1, 28359 Bremen, firstname.lastname@uni-bremen.de*

Abstract

In many space activities, robotic systems are indispensable. Existing systems are mission-specific in order to meet the individual needs. Rather than creating separate highly specialized systems, the goal is to create specialized and standardized modules that can be (re-)combined to configure a robot for certain tasks, hence greatly decreasing development work. This is where the MODKOM project (Modular Components as Building Blocks for Application-specific Configurable Space Robots) is heading, with the goal of creating a modular building block system that incorporates both specially produced components based on a standardized building block system and industrial third-party commercial off-the-shelf components. A software toolkit is developed in addition to the development of mechatronic and software modules for operation, helping non-expert users put those modular functional parts together to form robotic systems. Modules for system configuration (offline) and modules for system reconfiguration (online) are the two tiers of modules that the building block systematic distinguishes. To connect the toolkit modules, two types of interfaces are used: an active standard interconnect, which allows mechanical and electrical connection as well as data transfer, and a passive inter-module-interface, which allows the interconnection of atomic modules for configuring and assembling the system or greater modules. The efficiency of the system developed in MODKOM will be demonstrated by building a complex mobile manipulation system entirely out of the robotic assembly kit. As a result, a manipulator based on the toolkit's basic modules will be combined alongside a stationary modular platform and a mobile roving platform. The development of new systems based on modular payload components from the toolkit will then be accomplished using the manipulator, which will also be used to modify the built-in systems with additional function blocks. The purpose of the performance demonstration is to use the manipulator's end effector to interchange modules for reconfiguration, demonstrating the modular system's capability. Finally, the outcomes of MODKOM will assist in the future by creating modules and standards that can be quickly and easily modified to meet new or changing requirements.

Keywords: Modular Robotics, Building Blocks, Reconfiguration, Standard Interconnect, Software Toolbox.

1. Introduction

Robotic systems will be crucial in the future, whether they are used for planetary exploration, maintenance, orbital assembly, or satellite operations. Currently the vast majority of space system solutions are highly mission specific. This, of course, has the advantage of tremendous mass reduction and precise adaptation to that specific mission. As space debris increases and planetary missions with a huge variety of specific sub-tasks become more relevant, the need shifts further to modular systems; Systems that can be adapted to the next part-mission, while still keeping the mass overhead minimal. That way not 10 robots have to be developed and sent to their target for 10 specific operations, but only one with exchangeable modules that can be adapted to the next task.

Over time, 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. Common spacecraft are amogst other goals designed towards mass and cost reductions and so are their individual components as well as integration and interfaces. Due to that, servicing them on-orbit is arduous or impossible. The minimally modular spacecraft is already one step closer towards advanced modularity, 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].

When it comes to serviceable modularity, then modularity at the component level is a great advantage. Examples of spacecrafts with this level of modularity are the *Hubble Space Telescope (HST)* and *International Space Station (ISS)*, equipped with serviceable components and standard interfaces. However, because these components are not grouped into serviceable modules, any *On-Orbit Servicing (OOS)* activity would have to be conducted at the component level, with tools and procedures built particularly for each component independently. 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, this can be avoided. *The Multi-mission Modular Spacecraft (MMS)*, the *SolarMax* spacecraft, and the *Reconfigurable Operational spacecraft for Science and Exploration (ROSE)* take advantage of exactly that. They contain components grouped into serviceable modules, integrated on the main bus via a standardized interface, thus allowing for immense flexibility both on-ground, during I&T activities, and on-orbit, while keeping the complexity of those tasks at the minimum [1].

Regarding on-orbit operations, 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)* designed with an even greater spacecraft modularity in mind. These examples are composed of small interconnected modules, each providing only a fraction of functionality of a traditional spacecraft. Those modules are connected 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 [2, 3, 4].

In planetary applications up to now highly specialized, single-mission systems were used, which had there development focus on ruggedness and redundancy over serviceability and repairability. As currently deployed Mars rovers Spirit, Opportunity and Curiosity, are highly specialized, and conceived to be mobile laboratories to single-handedly carry out all the required exploration tasks, they are on the other hand inappropriate for future large scale exploration missions. For those missions, 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) [5, 6, 7]. Over the last years various standard interconnects (SI) 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 [8, 9]. 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 [10].

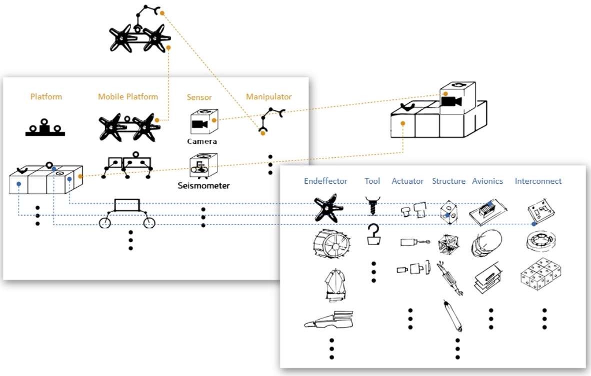


Fig. 1. Flexible use of the modular building blocks for the realization of mission-specific systems and requirements

This progress is being continued in the MODKOM project (Modular Components as Building Blocks for Application-specific Configurable Space Robots) by developing a modular building block system, as outlined in Fig. 1. MODKOM is developing not only a set of hardware and software modules that will be demonstrated in a composed system, but also software tools to assist non-experts in assembling systems from those modules. MODKOM continues the work started 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, off-shore and human-shared environments.

Based on the progress previously made in this project, which are explained in the following, a closer look at the evaluation and demonstration will be conducted.

2. Modular Toolbox Systematics

To deliver an approach for such a modular toolbox, a systematic has been defined based on a set of requirements. After the granularity has been determined, the software architecture was engineered to meet those requirements. We'll conclude this section by outlining the variety of applications.

2.1 Requirements

For the layout and definition of the modular construction kit, a set of top-level requirements was identified in

---

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

[11] and hence the specifications derived. These requirements include:

Application area The modular toolbox is 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.

Modularization and interconnectivity The chosen system modularization has to allow the configuration and integration of a complete system using modules of the toolkit. Furthermore, a reconfiguration of specific system parts shall be possible during the runtime of the system. Moreover it shall be possible to extend the toolkit with additional modules (see Sec. 2.2).

Hardware design The hardware realization of modules allows for clearly distinguished module functions. The modules shall allow an easy integratability and be compatible to each other. For reconfiguration purposes two different SI are applied, supporting a compatibility to the module definition (see Sec. 3.2).

Electromechanics design Standardized power connections and communication interfaces, including the integration of SI, shall be applied for all modules. For reconfiguration purposes, the de- and recoupling of modules shall be supported by the active SIs, as well as the identification and handling of these modules (see Sec. 3 and [11]).

Software design It shall be possible to perform a system check for consistency on software level, by comparing the system identification via NDLCom (Node-level Data Link Communication) and the planned set-up. The software framework enables automatic start and stop actions of tasks and recognize changing system functions due to reconfiguration. (see Sec. 3.8) System configuration and generation shall be user-friendly supported by providing graphic user interfaces (see Sec. 3.7).

2.2 Granularity

A central aspect of a modular toolkit is the granularity of the modules, as outlined in [11]. 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 Fig. 2, where one end of the spectrum are swarm robots and the other are monolithic robots.

Due to the complexity and limited functionality of swarm robots and the high development effort of monolithic systems, a more balanced approach is followed: To reduce cost and time in the development of new systems, a toolbox with standardized modules was chosen. 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 toolkit with new modules. To make the system more flexible, at least one active electromechanical interconnect is included in the toolkit. This interface allows the online reconfiguration of the system to fulfill 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. An overview of distinct modules, that are realized in the course of the activity is given in Sec. 3. It has to be noted, that some of the modules, such as the manipulator, employ modular components from a higher granularity level to be configured and integrated. As outlined in Fig. 1, a system module, such as e.g. a manipulator, can be designed and integrated by using modules from a higher granularity layer, such as joints, electronic devices or structural components.

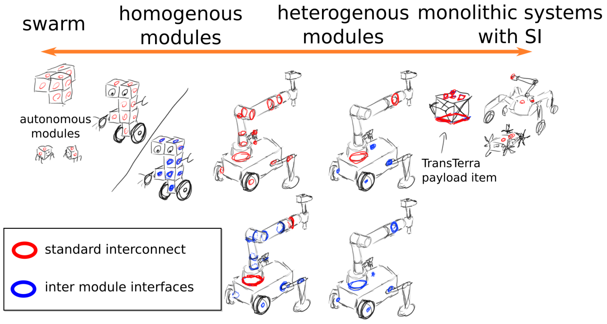
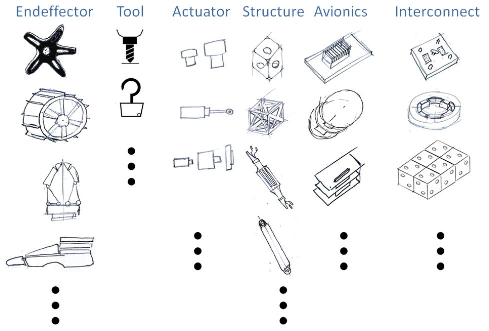


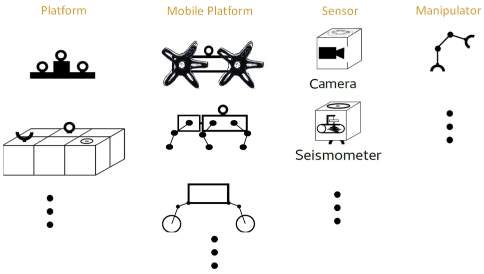
Fig. 2. Overview of granularity spectrum of modular systems

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 toolkit, 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.



(a) Examples of basic functional units for system configuration



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

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

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, as described in Sec. 3.2. This multifunctional interface can be controlled by the system during operation. The system can be reconfigured online or reconfigure itself via the SI. This allows the system not only to act flexibly according to the situation, but also to be used for different tasks within the mission.

2.3 *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 (see Fig.4). The following paragraphs provide an explanation of this type, as well as how components are defined and used.

XTypes *XTypes* are a generic type of objects. The speciality of this type is that it has a standardized type definition that allows to add properties and relations to other *XTypes*. Due to this handling, *XTypes* can be easily stored to graph databases while maintaining the relationships. Based on the initial advancements by [12] the *XTypes* have been reworked to now build a stand-alone software package. This package contains the basic and generic definition of *XTypes* as well as a generator tool which helps user to generate custom *XTypes*. Using this generator tool, the user only needs to perform these two actions: Create a template file to define their *XTypes* specializations, then populate the C++ source files. The generator will generate C++ header files and C++ sample files from the YAML template to assist the user, as well as Python bindings that do not require further editing. This generation step is simplified by providing a CMake macro that handles it.

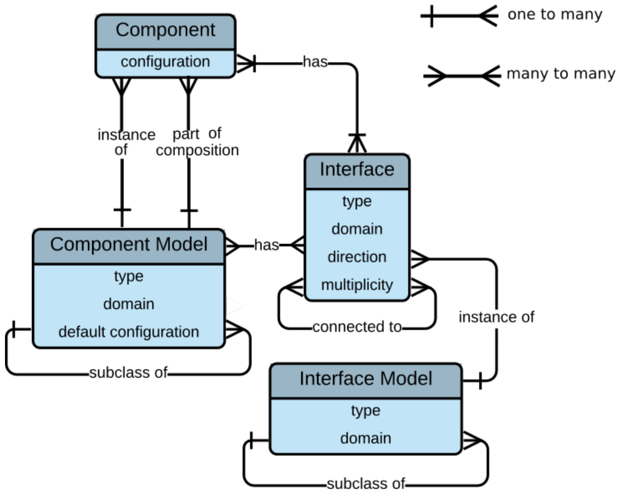


Fig. 4. Overview of module representation. The type boxes list their most important properties, while the arrows show possible relations to other type instances and cardinality constraints.

XDBI - Database Interface In order to maintain a database containing XTypes, the XDBI package, which was initiated by [12], has been reworked accordingly, providing generic handling for all *XTypes* specializations that might be created. XDBI is basically the *XTypes* database interface which takes schema of *XTypes* and realizes it on DB backend. Correspondingly, it adds or updates instances of *XType* in the database and has the ability to remove *XType*.

Besides local database access, this package can establish a database connection via HTTP/REST, as well as cutting the server bridge to use the backend directly serverless. Moreover, like the XTypes it comes with Python bindings, allowing to combine the benefits of Python and C++. Additionally, multi-database access is offered with one read/write database for the particular project and numerous read-only databases to import existing entities from in order to support import from other databases.

MODKOM-Types Based on the *XTypes* package a set of MODKOM-Types are specialized, that will take the core role in supporting hardware and software modularity from the computation side.

From the MODKOM-Types *Component* and *ComponentModel* are the most important ones for the subjects in scope of this project. The presented concepts are based and extended upon [12].

Components represent the hard- and software building blocks of robotic systems, which can be combined to create more complex components. Hence, a hierarchy of components of varying complexity is formed. 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, the following XTypes have been defined and will be used:

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 identical, have a unique ID, allowing each physically existing module to 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. By having abstract component models, it's possible to place abstract components in a network. These abstract components can later be replaced by an inheriting component that implements the functionality of the abstract component. Thereby exchange of components (hard- & software) is possible. 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.

Using the previously described XDBI-tool, all of these representations, or instances of MODKOM-Types, are saved to a graph-based 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 existing 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 sections 3.7 and 3.8 give an overview about the usage of the software tooling.

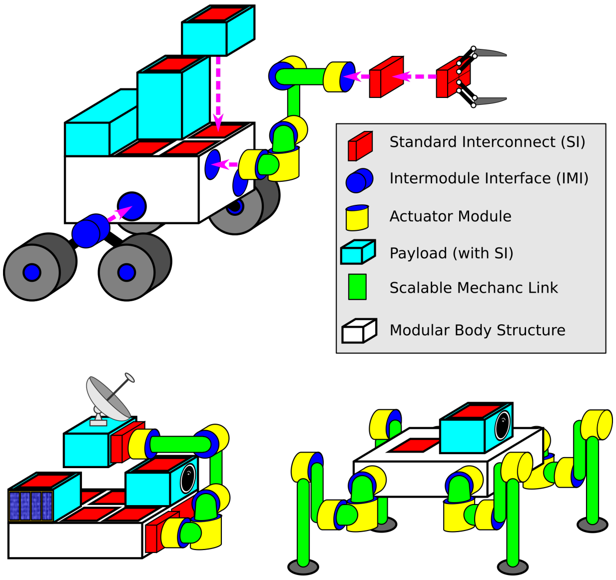


Fig. 5. Using the explained modularization, lots of compositions are possible and thereby systems can be adapted to many applications.

2.4 *Applicability*

By the chosen modularization of the toolbox design and the software architecture, the here developed approach offers a large area of application. The modules can be adapted and recreated in the workshop using the IMI in order to configure the target system. Those design decisions and/or changes can then be updated in the module database. During operation as the modules can be reassembled using the active SIs. This way the module can be reused and reconfigured to realize plenty different types of systems (see Fig. 5) with the same atomic components. The application area spans from orbital applications, e.g. for manipulation tasks and/or in-orbit assembly towards surface robots for extraterrestrial bodies, such as driving or walking systems. Here a core benefit is to establish a common set of robotic function units by means of modules to allow for a streamlined design and qualification process. The different modules, however, have to meet the specific environmental requirements of the targeted area of operation.

3. Distinct Module Realization

Selected modules of the toolbox are currently developed and implemented for a proof-of-concept and validation of the toolbox system. Modules are selected to cover the full range of granularity levels, as described in Sec. 2.2. The deployment of the modules will be realized in the context of a system demonstration as described in Sec. 4. Besides the evaluation of the entire workflow and the application of the toolbox during the demonstration, an additional important aspect is the integration of already

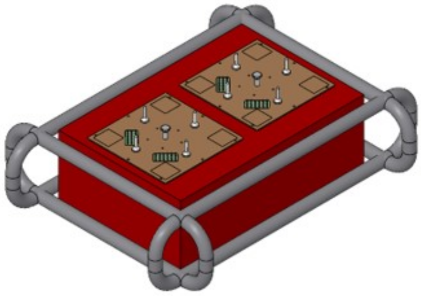


Fig. 6. Modular component of common rover body

existing industrial robotic systems. This is to evaluate and demonstrate the flexibility of the toolbox, both on the hardware and software side. An overview of the modules under development and proposed third party robotic elements is given in the following sub-sections.

3.1 Base Station

In the MODKOM project, the base is an object equipped with IMIs and SIs. It therefore serves as a demonstrator for the capabilities of the toolbox concept with regard to (offline) configuration and (online) reconfiguration, using electronic modules and a structural frame design based on a parametrised model. It also fulfills the function of an object that the mobile platform can interact and exchange payload items with, using the manipulator (cf. Sec. 3.6).

Shown in Fig. 6 is an early sketch of a minimal base station design. The red box represents a frame that contains the electronic modules, including the wiring harness. Two SIs are mounted on its top side to accept payload items. The compartment housing the electronic modules has an aspect ratio of 1:2:3. this allows stacking and combining multiple compartments in different orientations. The structural frame is currently envisioned to be a pipe frame which is based on a parametrised CAD design. This approach allows designing new frames for differently sized systems quickly and efficiently at the cost of limited possible topologies within reasonable boundaries.

For the proof of concept demonstration a base-station will be integrated based on these core modules. It will contain a set of relevant robotic sub-sub systems, such as an on-board computer, electrical power supply by means of a primary battery, a communication module and is equipped with several SIs. With this approach, a highly modular system configuration can be realized. As described in Sec. 4, the base-station serves as a core element to demonstrate the system reconfiguration by handling and changing its functions using the SIs.

Fig. 7. The Electro Mechanical Interface (EMI) attached to standardized 1U payload item container

3.2 Standard Interconnects

For reconfiguration on system module level, the application of one or more multifunctional interconnects also called as standard interconnects (SI) is a key element. The electro-mechanical interface (EMI) [5], developed by DFKI, is the baseline SI for the toolbox systematic. The EMI, as shown in Fig. 7 consists of an active and a passive part. While the active (and female) part contains the locking mechanism along with interface/module management electronics, the passive side provides pins for mechanical guidance and marker for visual servoing purposes. The EMI is developed for planetary exploration purposes, enabling its use in environments with a high dust load. During the course of the MODKOM activity the locking mechanism and mechanical guidance will be improved, based on lessons learned during extensive use within a Mars analogue environment in the desert of Utah [6].

Furthermore, a commercial SI will be implemented into the toolbox, as a proof-of-concept. Here, three SIs, specialized for orbital applications, are in close consideration: iSSI® by iBOSS GmbH [13], HOTDOCK by Space Application Services [14] and SIROM by SENER Aeroespacial [15]. For the toolbox demonstration, it is envisioned to implement an SI adapter between EMI and one of the previous candidates, to demonstrate the full integration and extension of the toolbox with already existing modular components.

3.3 Payload Items

The modular payload items are based on previous developments, as described in [5, 7, 16]. They combine two EMIs (active and passive) with a standardized payload container, c.f. Fig. 7. This allows to easily add new function blocks for system extension and reconfiguration into the toolbox. For the proof-of-concept demonstration it is envisioned to implement three distinct payload items, each representing an individual application on the reconfiguration layer.

Processing and Communication Module The processing and communication module will be equipped with an on-board computer and mesh capable communication module. The module can therefore be used to gain additional processing power on a given system, e.g. for pre-processing of high demanding sensor data streams. The additional mesh capable communication module enables to integrate the payload item into a given communication network and even extend it, as soon as a new system entity is formed.

Environmental Sensor and Power Module This module consists of various environmental sensors for different gases, temperature and humidity and holds its own battery pack to act as a range extender or to provide power for a new system entity. The module was previously developed and tested and is described in more detail in [5]. The module will be used to demonstrate the creation and integration of new systems entities by mating with the processing and communication module.

Stereo Camera Module A stereo camera module with integrated gimbal for image stabilization is proposed for system reconfiguration purposes, by combining the module with the modular manipulator (c.f. Sec. 3.6) to create a camera mast and mobile platform (c.f. Sec. 3.4), to extend the functionality of the system. The gimbal is proposed to be integrated by using a derived direct drive robotic joint from the DFKI-X2D development (c.f. Sec. 3.5). With the additional degrees of freedom added to the stereo camera, the PLI represents a robotic extension to its parent systems which needs to be considered and activated in the module management and operating software.

3.4 Mobile Platform

In addition to the consistent description of a modular system, an important aspect is to keep the modular system open for already existing third-party robotic components. To cover this aspect for the performance demonstration (c.f. Sec. 4), a commercial mobile robot platform is introduced as a system module. This is integrated into the module systematics on the hardware side by means of SIs and must also be integrated into the module framework on the software side accordingly in order to serve the operational and reconfiguration tasks. This is especially true for the bridging between the ROCK and ROS frameworks, since the module framework is natively based on the ROCK framework (c.f. Sec. 3.7), but the commercial platforms rely on ROS out of the box. Currently, several robot platforms, for example from Cleerpath Robotics, Robotnik and Agile X, are being evaluated with regard to their suitability for the intended performance demonstration.

Table 1. Requirements for DFKI-X2D Joint

Requirement	Value
Mass	800g (target)
Accuracy	0.02°
Power	150W
Torque	20Nm (continuous) 50Nm (peak)
Speed	350rpm
Voltage	48V
Sensors	2 x Encoders (min. 14 Bit) (Commutation / Output)
Control	Position, Speed & Torque via current
Electronic	Micro-Controller & COTS based
Communication	CAN-Bus
Additional functions	Recuperation Compliance

3.5 DFKI-X2D Joint

For several kinds of applications, like walking robots, or a gimbal for sensor stabilization, a direct or quasi-direct drive electric motor unit is beneficial. Such a joint is built, with the requirements as summarized in Tab. 1.

More than in the conventional rover design, the electrical energy supply [17] and the efficiency under thermal aspects poses a challenge in the design of dynamically walking robots for space. For its actuator designs, that means to find a good trade-off between:

1. high mass-specific torque (torque density),
2. impact mitigation capability,
3. and efficiency [18].

While the focus of terrestrial running machines is mainly on point one and two, for realizing the dynamic control and raise the physical systems performance, the point for efficiency does not play the decisive role normally. Enabled, due to the high availability of electrical energy and a usually active cooling of the actuators. Thus, optimization for high, mass-specific torque and impact mitigation capability inevitably leads to a design of actuators that operate at the physical limit and have high dissipation. From a systemic point of view, that approach is still beneficial, but arises locally high linear power losses and is a threat to thermal stability when using such actuators in a space environment.

Towards this challenge, there are two ways to achieve a more efficient operation: The reduction of linear losses by choosing a larger motor with a higher motor constant, as described in [19], or to increase the gear ratio. The latter has a strong effect on the impact mitigation capability

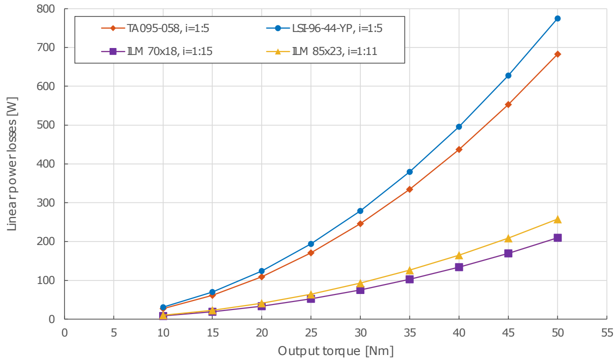


Fig. 8. Power losses of different actuators and gearbox combinations for 10 Nm to 50 Nm output torque. Selected to build and test were the combinations of a TA095-058 out-runner motor (similar to Halodi REVO1) with a 1:5 gear ratio and an ILM 85x23 in-runner motor with a 1:11 gear ratio

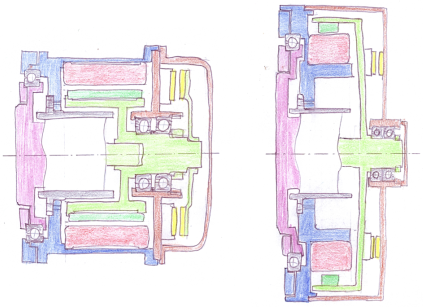


Fig. 9. Visualization preliminary design of the two concepts for the DFKI X2D Joint. Output bearing type not finally selected. *Left:* In runner motor with high ratio planetary gear (i=1:11), *Right:* Outrunner motor with low ratio planetary gear (i=1:5), *Purple:* Output shaft, *Blue:* Housing, *Brown:* Lid, *Red:* Stator, *Green:* Rotor with carrier, *Grey:* Gearbox and bearings, *Yellow:* Encoder.

and requires a good friction model of the drive. However, it offers also advantages for the mass of the drive. Based on that fundamentals, two strategies were selected to design a usable actuator as a module in this project.

1. Using a new developed torque and efficiency optimized out-runner motor in combination with a one stage planetary gear box (i=5).
2. Using a Robodrive in runner motor with two stage planetary gearbox (i=11) as competitor.

Fig. 8 shows the current consumption of several concepts and their selection. The reflected inertia can be regarded as equivalent for all concepts since the rotor inertia could be significantly reduced for higher reduction ratios. To create comparable values, the envisaged actuator mass of all the combinations considered are in a range of 600 g to 900 g. To realize higher reduction motors compatible with control algorithms, special attention will be paid to the inner friction of the bearings and the gear stage used. A first design approach of the two chosen motor-gear concepts integrated into a full joint assembly is given in FIG. 9.

To have precise knowledge of the position of the joint and thus enable accurate control, rotary encoders are implemented. For investigation two commercially available digital encoders are chosen. Analog solutions are not further pursued, due to higher cost and development effort related to building costum analog/digital electronics. The first investigated unit is an inductive encoder by Zettlex. Several sources, e.g. [20], state its suitablity for space applications, though no precise specifications are given. As comparison a capacitive encoder by Netzer is evaluated. The manufacturer states a version with space-qualified electronics being developed.

Like its successor, the DFKI-X joint [21], the electronics will be designed using COTS components. In order to make the power electronics small, efficient, powerful and as radiation-tolerant as possible, it will be based on GaN-FETs. Also, by using GaN-FETs, the PWM (Pulse Width Modulation) frequency can be raised from the original 30kHz to 50-100kHz without compromising performance and efficiency compared to MOS-FETs. Increasing the frequency allows for a reduction in bus capacitance and a physical downsizing of the filtering effort, both of which further aid compactness.

However, a major change is intended for logic part.In previous developments, FPGAs were used, which required a large number of external peripherals (e.g. ADC (analog/digital converter), memory, CAN-Phy) and a wide variety of voltage regulators, thus taking up a lot of components and physical space on the PCB (Printed Circuit Board). At the same time, the manufacturing yield and FIT rate declined with each additional component. In the new development, the FPGA and its external peripherals are largely replaced by the highly integrated automotive microcontroller SAMV71Q21 by Mircochip Technology (formerly Atmel), which already includes most of the required peripherals and thus offers significant space advantages compared to an FPGA solution. If required, this is also available as a "radiation tolerant" version (SAMV71Q21RT) or as a slightly modified "radiation hard" version (SAMRH707), which simplifies portability.

In addition, the single-design PCB known from the previous design study (DFKI-X joint), which contains logic and power section, is split into two separate PCBs for manufacturing advantages (c.f. [21]).

Since the previous design violated the ECSS-E-ST-20-07C specifications for electromagnetic compatibility

(EMC) in some points, more attention is paid to their EMC in the new design. This includes the consistent use of shielded cables for external sensors, motor phases and communication lines as well as the filtering of the power supply.

3.6 Manipulator

As one of the core elements, a modular manipulator is built. The design here directly follows the toolbox systematics, so that the manipulator is constructed from various functional units. The core modules are modular robot joints, each of which is combined into a 2 DOF pan-tilt unit. These are connected to each other by means of standardized connectors and the IMI to form a 6 DOF arm. The link lengths can be adjusted according to the application, so that the necessary working space can be configured during the design. For both purposes, end effector and manipulator base, the arm is equipped with an active EMI on both sides. This allows the location of the manipulator to be changed during operation by bridging over from EMI to EMI and even changing the deployment system. In order not to limit the payload capacity after a change of position, the base and wrist joints are designed identically. This also avoids preferential orientation of the manipulator and provides maximum deployment flexibility. In addition, equipping the arm with two EMIs allows the system to be fully integrated into the modular tool kit, creating a stand-alone system module for reconfiguration.

The manipulator is designed to handle a stack of at least two payload items. The modularity concept will be further developed here based on an existing underwater manipulator [22] (see Fig. 10) according to the toolbox systematics. For the joints, a combination of Robodrive ILM70 paired with a Harmonic Drive gearbox will be used. The motor control electronics will be integrated directly into the joints and contain a communication and an FPGA board [23] in addition to the motor driver. A comparable robotic joint has already been developed and prequalified for use in low Earth orbit [21].

3.7 User Interface

A modular system is a collection of modules, interfaces, and configurable options. Each has its unique configuration and performs a specialized purpose and these modules are linked together to build a network. Formerly, domain experts had to spend a lot of time composing such complicated component networks. Due to the complexity and need for in-depth understanding of each sub module for such composition of a modular system, such component networks were traditionally developed manually. This process took a long time and was prone to human errors.

The tools explained in the following assist the user when graphically composing complicated modular robotic systems. Some of them are already in use e.g. in the project KIMMI-SF[2] to setup the navigation stack for a mobile manipulator that supports a human employee in a workspace environment.

Fig. 10. Example of a modular manipulator for underwater applications.

3.7.1 Building Block Creation

Building block creation is the initial step in assembling a modular kit. For the hardware side, the Blender add-on PHOBOS [24] will be used to produce new atomic component models from CAD data (see Fig. 11). It helps the user to pre-process the model, e.g. adapt meshes and annotate the hardware with information that is not yet there from the CAD export. This includes customization possibilities, SI locations and orientations, motor and sensor data, loop constraints and further annotations. Finally, the model can be exported, and the newly represented atomic component model will be saved to the database. Larger artifacts like (e.g. kinematic data, meshes, annotations) that cannot be stored in a database will be moved to an external repository and referenced in the component model (e.g. kinematic representation; see also Sec. 2.3).

---

[2] KIMMI-SF: `https://robotik.dfki-bremen.de/en/research/projects/kimmi-sf/`

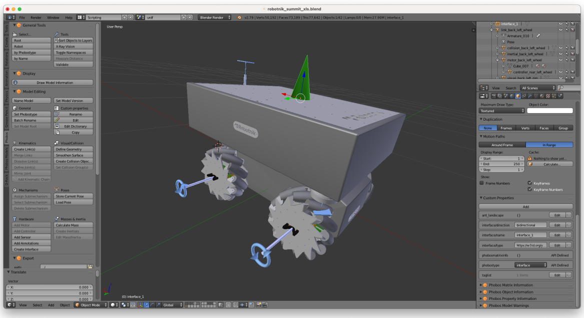


Fig. 11. Phobos: Graphical interface to visualize, assemble, and configure component models.

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. For custom ROCK-tasks an `orogen2xrock` script is provided which generates the database entry. Those atomic component models can in the next step be composed to greater component models that can be used in the same manner. These hierarchies make it simple to debug even the most complex networks. GUIs developed in the D/Q-ROCK-projects make this more user-friendly as described in the following paragraphs.

3.7.2 *Composing an Assembly*

The following tools are used to generate new component models, both hard- and software, out of existing (atomic) component models from the database. As a first step when realizing a project, the necessary components have to be identified and then picked from the existing ones. A component browser, as shown in Fig. 12, is offered to make this simple by displaying all component models accessible in the database. From this variety the user can then pick what is needed. This includes also components delivering AI based software components that enhance the robot operation. Those robot behaviors can be optimized using optimization tools like BoLeRo [25].

The X-ROCK-GUI (see Fig. 13) displays components as nodes and allows the user to establish connections between them by drawing edges in a graph view. During this project the user is supported by ensuring that only valid connections between the interfaces of components are established. Using the X-ROCK-GUI they can not only be composed but also configured. Even for components that have dynamic interfaces - interfaces that only exist for specific configurations - the X-ROCK-GUI provides support, by displaying available ports for the current configuration. It is a comprehensive tool which also enables configuration of components and thereby offers to edit all parameters of a component network, for both hardware and software. The user can provide exchangeable components by providing numerous components that implement an abstract component in the graph.

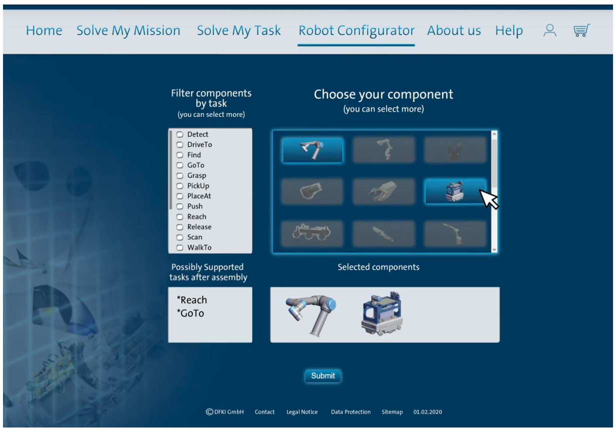


Fig. 12. Using the component browser depicted, desired components can be selected from all existing.

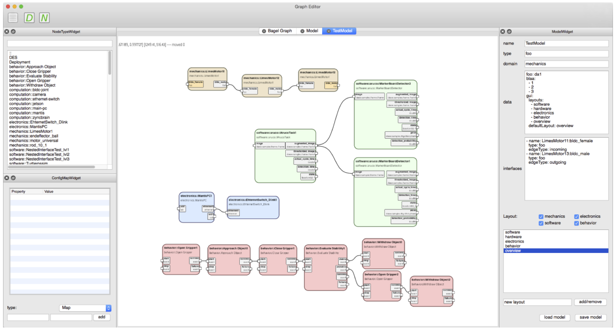


Fig. 13. Graphical interface to visualize, assemble, and configure component models.

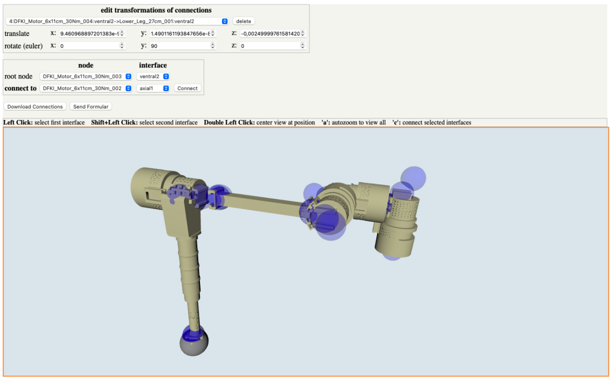
DEIMOS is a 3D visualization of hardware components and their interfaces (see Fig. 14). It is simple to designate which SI connects to which other SI in this GUI, as well as their respective orientation. By selecting two interfaces and performing necessary transformations, hardware components can be assembled and the resulting component model can be stored again to the database.

3.8 *Software, Data & Hardware Integration*

The world's most advanced technical advancement is increasingly occurring on a platform that integrates hardware and software. Software is being added to all types of hardware to make them more intelligent and capable of a far wider range of activities. Integration of software, data, and hardware is covered in the following subsections.

3.8.1 *Software*

To set up the software on the system, buildconf and bundles (as a package to be installed in the system) must be created. The buildconf for the system is versioned, just like Components and ComponentModels. All versions



edit transformations of connections
4:DFKI_Motor_6x11cm_30Nm_004:ventral2->Lower_Leg_27cm_001:ventral2
translate x: 9.460968897201383e-9 y: 1.4901161193847656e-8 z: -0,00249999761581420
rotate (euler) x: 0 y: 90 z: 0

	node	interface
root node	DFKI_Motor_6x11cm_30Nm_003	ventral2
connect to	DFKI_Motor_6x11cm_30Nm_002	axial1

Left Click: select first interface **Shift+Left Click:** select second interface **Double Left Click:** center view at position **'a':** autozoom to view all **'c':** connect selected interfaces

Fig. 14. The DEIMOS-GUI is used to easily connect hardware components with the desired orientations.

(including OS requirements and source packages) must be fixed in this buildconf. Versions can be easily and safely switched this way.

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

A mapper tool generates the CND by mapping individual software components to available execution hardware (component network description). The resulting mapping is used by system deployment tools to bundle and compile 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).

Everything will eventually be gathered to offer a quick manner of programming the robot's software installation: The system's software will be made available as a directory that may be downloaded to a bootable storage device. It will include both the appropriate Ubuntu distribution and a script to download and install the program.

3.8.2 Data & Hardware integration

The hardware is assembled and stored to the database using the DEIMOS or X-ROCK-GUI. The assembled model is then used to generate a blueprint for its construction. A built assembly is an instance of the model, and thus a component with a unique identifier and properties. Once built, the assembly instance is saved to the database (including updated properties and other relevant information). The created assembly will then have the bootable storage medium inserted into it, and it will finally start up. The software either detects every active subcomponent of the system on boot up or periodically receives a status report from each one. The software, if necessary, updates the subcomponents' firmware to the version specified in the assembly model. The required software modules then begin to run and the system is then ready for use.

A robotic manipulator with up to six robotic joints and FPGA-based motor controllers is the first assembly made in this manner. To obtain reference updates and send the current status to a central control instance, they communicate via NDLCom.

4. Demonstration of Integration and Application

For proof-of-concept purposes two application scenarios have been designed, allowing to evaluate all levels of the defined modular toolkit 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.

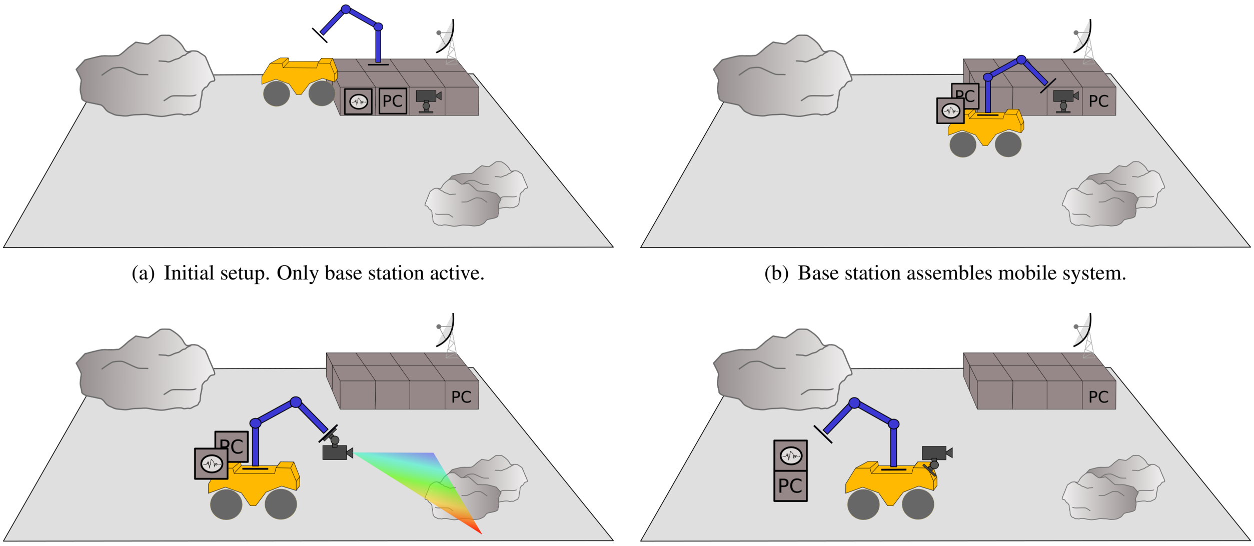
4.1 Scenario Overview

Both demonstrations are working on the same application scenario, one on the system creation and one on the operation itself. The general outline of this evaluation scenario is depicted in Fig. 15: A modular base station will be used to assemble a previously defined mobile, modular system.

The scenario will begin with both systems operational and their respective software stacks installed (see Fig. 15(a)). Then (see Fig. 15(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. 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. 15(b)). As soon as it is connected (in hardware), the base station will switch off 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. 15(c)). Once the map has been generated the mobile system can find a reasonable place for setting up a immobile sensor station from the payload items (see Fig. 15(d)).

4.2 System Composition & Configuration

The first demonstration will show the functionality of the MODKOM tooling workflow by composing and integrating the systems for the described scenario based on modules from the toolkit. 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 con-



(a) Initial setup. Only base station active.

(b) Base station assembles mobile system.

(c) Mobile system explores and generates map.

(d) At a reasonable location a sensor station is setup.

Fig. 15. Envisioned demo scenario for modular system operation at project end.

nected using the GUIs described in Sec. 3.7.2. 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 by using abstract components. Later on it can be specified which components implements those abstracts. 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.

4.3 System Operation

The second demonstration deals with operating the composed systems on a modular systems level. Here the online module management, system reconfiguration and expansion of the existing multi-agent system will be demonstrated by creating new entities via combining individual modules. While the online module management has to be active throughout the whole demonstration scenario, different aspects of system reconfiguration and system creation will be performed. For the proof-of-concept demonstration, the different modules described in Sec. 3 will be used.

The demonstration starts with the base-station equipped with the modular manipulator and all modular payload items (processing and communication module, environmental and power module and stereo camera module), leaving the mobile rover platform in its initial condition. After approaching the base-station, the manipulator will transfer the processing and environmental sensor module to the rover. To enable this transfer, the rover has to communicate its position and orientation with respect to the base-station, as it is the system carrying the mapping sensors while the manipulator has only close range visual servoing capabilities for EMI connection. After successful docking of the payload items, the module management of the rover and base-station have to acknowledge the transfer and system reconfiguration, e.g. with additional sensor set-up, respectively.

After the payload item transfer, the manipulator is supposed to switch its the system it belongs to from the base station to the rover. This new robotic device has now to be loaded and activated in the rover's operational layer to support the full reconfigured system. From the rover, the manipulator will dock to the stereo camera payload item, demonstrating a successful system switch and recognition of the manipulators orientation and adaption of its kinematic model accordingly, plus potentially integrating the additional DOF of the attached payload item.

From here the fully equipped mobile rover will head to a pre-defined location, while demonstrating the stereo-camera mast during traversal. At the goal position, the camera will be decoupled from the manipulator and stored on the rover's deck. Now, the stack of the processor and communications module connected to the environmental sensor and power module will be deployed from the rover and activated as a new system. In this sense, a new system entity has to be created within the module management, being able to act as additional communication node and providing environmental sensor data.

5. Summary and Outlook

The previous sections give insight into the formulation and creation of a modular building block system, that incorporates specially designed modules based on the general modular toolbox systematics as well as industrial third-party components. The overall toolbox systematic is outlined, explaining the underlying top-level requirements, level of module granularity and system decomposition as well as the software architecture, enabling the actual operation of all modules in the end. Here a distinction is made between functional units for offline system configuration and system modules, which can allow online system reconfiguration. The interaction between these layers and their modules is established by introducing two kinds of module interfaces, namely the inter-module interface (IMI) (for configuration purposes) and the standard interconnect (SI) (for reconfiguration purposes).

Furthermore, a set of distinct modules is described, which are developed in the course of the activity and will be integrated into a performance demonstration scenario. The applicability of the toolbox to space hardware is followed by the development and qualification of a quasi direct robotic drive as a key element of future space robotic applications. The building block application relies on a user-friendly GUI environment for module and system management, which allows the integration of AI-supported system optimizations. The creation of the operation software and the corresponding hardware integration is also connected to this environment. Both the user interface as well as the software architecture is presented in this paper.

This project does not only provide 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.

During the course of the MODKOM project the described software architecture as well as system modules will be realized and implemented. The different granularity and functional layers of the toolkit will be demonstrated by means of a terrestrial proof-of-concept demonstration. This encompasses system composition and configuration as well as system operation and reconfiguration by using modules from the toolkit as well as already existing commercial robotic elements, which will be integrated into the modular approach as well.

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 Aerospace Center (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.

[1] D. Rossetti et al. “Spacecraft Modularity for Serviceable Satellites”. In: *AIAA SPACE 2015 Conference and Exposition*. Reston, Virginia: American Institute of Aeronautics and Astronautics, Aug. 2015. ISBN: 978-1-62410-334-6. DOI: `10.2514/6.2015-4579`.

[2] A.A. Kerzhner et al. “Architecting Cellularized Space Systems using Model-Based Design Exploration”. In: *AIAA SPACE 2013 Conference and Exposition*. Reston, Virginia: American Institute of Aeronautics and Astronautics, Sept. 2013, pp. 1–24. ISBN: 978-1-62410-239-4. DOI: `10.2514/6.2013-5371`.

[3] D. Sternberg et al. “A Bottom-up Modeling Approach for the Profit Analysis of Cellularized Spacecraft Architectures”. In: *64th International Astronautical Congress (IAC)*. Beijing, China: International Astronautical Federation, 2013, pp. 1–11.

[4] B. Karlow et al. “Tradespace investigation of strategic design factors for large space telescopes”. In: *Journal of Astronomical Telescopes, Instruments, and Systems* 1.2 (Apr. 2015), p. 027003. ISSN: 2329-4124. DOI: `10.1117/1.JATIS.1.2.027003`.

[5] W. Brinkmann et al. “Modular Payload-Items for Payload-assembly and System Enhancement for Future Planetary Missions”. In: *2018 IEEE Aerospace Conference*. Big Sky, Montana, USA: IEEE Comput. Soc. Press, 2018, pp. 1–10.

[6] R.U. Sonsalla et al. “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)*. ESA/ ESTEC. Leiden, the Netherlands, June 2017.

[7] R.U. Sonsalla et al. “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.

[8] M. Jankovic et al. “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*. IEEE. Big Sky, Montana, Mar. 2018.

[9] S. Estable et al. “PERIOD – PERASPERA In-Orbit Demonstration toward the transition into the in-space services, assembly and manufacturing paradigm”. In: *IOP Conference Series: Materials Science and Engineering, IOP Publishing* 1226 (Feb. 2022), pp. 1–8.

[10] Astrolab. *FLEX Payload Interface Guide*. v 1.0. Astrolab Vebturi. Jan. 2022.

[11] Roland U. Sonsalla et al. “Towards Modular Components as Building Blocks for Application-specific Configurable Space Robots”. In: *Proceedings of the 16th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA 2022)*. ESA/ESTEC. Leiden, Netherlands, June 2022.

[12] T.M. Roehr et al. “A development cycle for automated self-exploration of robot behaviors”. In: *AI Perspectives* 3.1 (2021), pp. 1–29.

[13] TA Schervan et al. “New Horizons for Exploration via Flexible Concepts Based on Building Blocks Using the Standardized iSSI (Intelligent Space System Interface) Modular Coupling KIT by iBOSS”. In: *Proceedings of Global Space Exploration Conference (GLEX 2021), St. Petersburg, Russian Federation*. 2021.

[14] Pierre Letier et al. “HOTDOCK: Design and Validation of a New Generation of Standard Robotic Interface for On-Orbit Servicing”. In: *International Astronautical Congress, IAC 2020*. IAF. 2020.

[15] Javier Vinals, J Gala, and G Guerra. “Standard Interface for Robotic Manipulation (SIROM): SRC H2020 OG5 Final Results-Future Upgrades and Applications”. In: *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*. 2020.

[16] Thomas M. Roehr, Florian Cordes, and Frank Kirchner. “Reconfigurable Integrated Multirobot Exploration System (RIMRES): Heterogeneous Modular Reconfigurable Robots for Space Exploration”. In: *Journal of Field Robotics* Special Issue on Space Robotics (2014), pp. 3–34.

[17] Niklas A. Mulsow, Benjamin Hülsen, and Patrick Schöberl. “Towards Influences of the EPS on Lunar Rover’s Systems Design”. In: *Proceedings of the 16th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA 2022)*. ESA/ESTEC. Leiden, Netherlands, June 2022.

[18] Patrick M Wensing et al. “Proprioceptive actuator design in the mit cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots”. In: *Ieee transactions on robotics* 33.3 (2017), pp. 509–522.

[19] Giorgio Valsecchi et al. “Preliminary design of actuators for walking robot on the Moon”. In: *Proceedings of the 16th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA 2022)*. ESA/ESTEC. Leiden, Netherlands, June 2022.

[20] Jiwen Han et al. “A compact motor drive system design for integrated joint of space manipulator”. In: *CSAA/IET International Conference on Aircraft Utility Systems (AUS 2018)*. 2018, pp. 383–387. DOI: `10.1049/cp.2018.0130`.

[21] Niklas A. Mulsow et al. “Design and Environmental Testing of the DFKI-X Robotics Joint for Space Applications”. In: *Proceedings of the 19th European Space Mechanisms and Tribology Symposium (ESMATS)*. ESA. online, Sept. 2021.

[22] Leif Christensen et al. “The Hydrobatic Dual-Arm Intervention AUV Cuttlefish”. In: *OCEANS 22 Hampton Roads*. to be published. MTS / IEEE OES. IEEE, 2022, pp. 1–8.

[23] Roland U. Sonsalla et al. “DFKI-X: A Novel, Compact and Highly Integrated Robotics Joint for Space Applications”. In: *Proceedings of the 17th European Space Mechanisms and Tribology Symposium (ESMATS 2017)*. ESMATS, Sept. 2017. ISBN: 978-90-827435-0-0.

[24] K. von Szadkowski and S. Reichel. “Phobos: A tool for creating complex robot models”. In: *Journal of Open Source Software* 5.45 (2020), p. 1326.

[25] Alexander Fabisch, Malte Langosz, and Frank Kirchner. “BOLeRo: Behavior optimization and learning for robots”. In: *International Journal of Advanced Robotic Systems* 17.3 (2020). DOI: `10.1177/1729881420913741`.