Reconfigurable Integrated Multirobot Exploration System (RIMRES): Heterogeneous Modular Reconfigurable Robots for Space Exploration

Thomas M. Roehr
DFKI Robotics Innovation Center Bremen, 28359 Bremen, Germany
e-mail: thomas.roehr@dfki.de
Florian Cordes
DFKI Robotics Innovation Center Bremen, 28359 Bremen, Germany
e-mail: florian.cordes@dfki.de
Frank Kirchner
DFKI Robotics Innovation Center Bremen and University of Bremen, 28359 Bremen, Germany
e-mail: frank.kirchner@dfki.de

Received 7 December 2012; accepted 12 July 2013

This paper presents the multirobot team RIMRES (Reconfigurable Integrated Multirobot Exploration System), which is comprised of a wheeled rover, a legged scout, and several immobile payload items. The heterogeneous systems are employed to demonstrate the feasibility of reconfigurable and modular systems for lunar polar crater exploration missions. All systems have been designed with a common electromechanical interface, allowing to tightly interconnect all these systems to a single system and also to form new electromechanical units. With the different strengths of the respective subsystems, a robust and flexible overall multirobot system is built up to tackle the, to some extent, contradictory requirements for an exploration mission in a crater environment. In RIMRES, the capability for reconfiguration is explicitly taken into account in the design phase of the system, leading to a high degree of flexibility for restructuring the overall multirobot system. To enable the systems' capabilities, the same distributed control software architecture is applied to rover, scout, and payload items, allowing for semiautonomous cooperative actions as well as full manual control by a mission operator. For validation purposes, the authors present the results of two critical parts of the aspired mission, the deployment of a payload and the autonomous docking procedure between the legged scout robot and the wheeled rover. This allows us to illustrate the feasibility of complex, cooperative, and autonomous reconfiguration maneuvers with the developed reconfigurable team of robots.

1. INTRODUCTION

In space exploration scenarios of different agencies, the moon is seen as a stepping stone in human space exploration (ISECG – International Space Exploration Coordination Group, 2011). For an extended stay of humans on the lunar surface, *in situ* resource utilization is crucial for a successful mission and for the preparation of human exploration of more remote destinations such as, for example, Mars. Robotic precursor missions are part of the roadmaps for human space exploration. Apart from *in situ* production of building materials for shelter, water ice is an important resource that is needed for generating fuel or oxygen for human habitats on the moon.

Direct correspondence to: Thomas M. Roehr, e-mail: thomas.roehr@dfki.de

Building up deposits of water ice on the lunar surface is possible by different mechanisms, such as water ice contained in meteorites or comets that impact the lunar surface, hydrogen from the solar winds that react with lunar oxides, or outgassing of water from the inner parts of the moon (Arnold, 1979). The deposits of water ice are diminished by the dissociation of photons coming from the sun, which leads to the assumption that water ice is most likely to be found in permanently shaded regions—the so-called cold traps—of the lunar surface, as can be found in the polar crater regions of the moon (Zuber et al., 2012).

The LCROSS mission (Colaprete et al., 2010) showed indications of the presence of water ice and other volatiles in the Cabeus crater at the lunar south pole. However, the proof of water ice has up to now been only indirect measurements via spectral analysis. Thus, *in situ* confirmation

View this article online at wileyonlinelibrary.com • DOI: 10.1002/rob.21477

and a better understanding of the distribution of resources are still open questions that need to be addressed.

As indicated, not only might water ice be present as a volatile in the cold traps on the moon, but other volatiles are likely to be found there as well (Mosher and Lucey, 2006). The main scientific goals for a mission to explore the lunar cold traps for volatiles include the following:

- determination of the volatile composition (isotopic, elementary, mineralogically),
- mapping of the local distribution and the identification of the volatile's sources,
- mineralogical diversity at the landing site, including age, distribution, origin, and composition, and
- the lunar environment including dynamic processes, such as weathering and meteoroid impacts.

Technologically more challenging than orbiting missions and with higher risk are landing missions that make use of surface-deployable probes such as landing units and/or mobile systems. However, this approach can provide deeper insight into the above-mentioned scientific goals (Mosher and Lucey, 2006).

In this paper, a heterogeneous modular multirobot approach is presented that is intended to bring a robotic system down into the permanently shaded regions of a lunar polar crater in search of volatiles bound to the lunar regolith. In the presented approach, a wheeled rover (Sherpa[1]) and a legged scout robot (CREX[2]) are used together with surface-deployable modular payloads. The idea is to combine the energy-efficient locomotion principle of wheels with the high mobility of a legged system. Both systems can be combined via an electromechanical interface (EMI); in the connected state, the robots act as a monolithic system, whereas in detached mode, both systems act independently of each other. In Figure 1, the systems are depicted. In general, the EMI serves several use-cases: (1) docking of Sherpa and CREX, (2) manipulating modular payload-items, (3) stacking of payload-items to form payloads, and (4) attaching payload-items to the mobile systems, which is made possible by using the EMI (Wenzel et al., 2011).

This paper is structured as follows: In the following section, a short overview on some fields of related work for RIMRES is provided. A more elaborate discussion on reconfiguration in terms of system design and levels of reconfiguration is provided in Section 3, while Section 4 presents details about the hardware components constituting the overall system. Section 5 presents the software framework that is developed for representing the hardware reconfiguration possibilities and modularity in software and mission control. Experiments with the systems and a comparison to a former multirobot approach are discussed in Section 6. Section 7 explicitly describes the lessons learned, and the paper is concluded in Section 8.

[1]Sherpa: Expandable Rover for Planetary Applications.
[2]Crater Explorer.

2. RELATED WORK

To the best of our knowledge, there is no system that is directly comparable to the approach presented here in terms of reconfiguration capabilities, technological complexity of the single subsystems involved in modular reconfiguration, and seamless integration of self-contained systems into a new system. However, there are several systems that are related to the RIMRES system in one way or another. This section gives a brief overview of some relevant systems and research activities. Because RIMRES tackles a broad range of topics, the following descriptions are not intended to be complete, but are meant to give an overview of some of the relevant systems.

2.1. Modular and Reconfigurable Systems

Modular reconfigurable systems in the literature are mostly systems that provide a kind of atomic modularity, meaning that the systems are built up from identical modules or modules at least on the same scale with slightly different functions. The approach followed in RIMRES is opposite to that, since the systems that interconnect range from cubic modular payload-items (150 mm $\times$ 150 mm $\times$ 150 mm, mass $<$ 5 kg) over the legged scout system (around 27 kg) to the wheeled rover with a (variable) footprint of up to 2.5 m $\times$ 2.5 m and a weight of around 160 kg. Thus, sizes and weights of subsystems in RIMRES range over two and three orders of magnitude, respectively. A similarity between the common modular (self-)reconfigurable systems in the literature and the approach presented here is the need for a common EMI that is shared between all systems.

In the literature, there is a broad range of designs realizing a connector mechanism for the connection of single modules in a multi module system (MMS). Approaches using permanent or electromagnets, such as the Telecubes (Suh et al., 2002), are elegant because no moving parts are necessary. However, these approaches might require high power when loads of several kilograms have to be securely fastened in environments with mechanical shocks and there is a high probability of dirt accumulation in the systems.

A pin/hole mechanism combined with a shape memory alloy is an alternative to build latch mechanisms only requiring actuation for detaching the single systems from each other. Conro (Castano et al., 2002) or PolyBot (Yim et al., 2002) are examples of this class of latching mechanism.

The systems M-TRAN III (Kurokawa et al., 2007) and Atron (Ostergaard et al., 2006) make use of active hooks and appropriate bails for connecting mechanically to other

(a) RIMRES systems as CAD models. The six-legged scout CREX is beneath the wheeled rover Sherpa. Attached to Sherpa's manipulator are two payload-items that form a surface-deployable payload.

(b) Photograph of final integration status of RIMRES mobile systems. Wheeled rover Sherpa with manipulator arm and docked six-legged scout robot CREX.

Figure 1. Systems in the heterogeneous modular multirobot system RIMRES.

modules of the system. Even though the two systems make use of a similar principle for mechanical connection, data connections are different between both systems: M-TRAN makes use of electrodes to transfer data electrically, while Atron modules communicate with each other via infrared signals.

The minimal requirement for the physical connection is that the mechanism can withstand forces that might occur during operation. Typically, these forces depend on the weight of the attached modules. In Sproewitz et al. (2008), a connection mechanism is presented that can withstand tensile and shear forces of approximately 180 N and shear torques of 7 Nm. A heavy-duty connector for self-reconfigurable robots that withstands forces of more than 700 N is presented in Nilsson (2002).

2.2. Examples for Rover Systems with Reconfiguration Capabilities

The ATHLETE (All-Terrain, Hex-Limbed, Extra-Terrestrial Explorer) uses six wheels on actuated legs to walk and to drive (Wilcox et al., 2007). This concept combines the advantages of both locomotion possibilities: energy efficiency and high mobility. Each leg possesses six degrees of freedom (DOF) [seven DOF in the case of the Tri-ATHLETE version (Wheeler et al., 2010)].

A quick-disconnect tool adapter is employed so that each leg can be used as a general-purpose manipulator in a second use-case. Different tools can be applied. This incorporates drilling devices and grippers. Even scoops for shifting greater amounts of soil can be attached when using two legs in combination. Thus, the legs of the ATHLETE family of robots are reconfigurable devices that can be used for both locomotion (driving motions and undulating behaviors) and manipulation tasks.

The design with the six limbs arranged in a hexagon enables it to be operated in an inverted position, and even more importantly it helps to prevent tipping over. Each face of the hex frame possesses a stereo camera with appropriate lighting to navigate and avoid hazardous objects. The cameras are also used for visual odometry and visual docking in the case of a desired cooperative maneuver. The current version is powered by a gasoline-motor generator and lead-acid batteries, while a future flight model is intended to be powered by solar panels and H_2O_2 fuel cells.

Scarab (Wettergreen et al., 2009) is a four-wheeled rover that combines a rather classical bogie suspension with an active DOF to enhance the ability to climb and drive along slopes. Furthermore, control of the body height as well as the roll angle of the robot is possible. The aspired mission for the system is to take drilling cores within perpetual darkness of lunar polar craters. Therefore, an upright drill is employed in the center of the robot. The suspension system is used to lower the body of the rover in preparation for the drilling process. Since the drill is in an upright position, the structure has a dual use: Apart from the drill itself, it supports navigation sensors and thus works as a navigation mast (Bartlett et al., 2008).

Tri-Star IV (Aoki et al., 2011) is a three-wheeled rover that can reconfigure and adapt to changing terrain types by rotating its wheeled arms and using flexible wheels. It represents the latest advancement in the development of a series of three-wheeled rovers. The capability to recover from an upside-down position is an essential feature of this rover along with an optimized storage posture. Furthermore, it is embedded into a multirobot architecture consisting of so-called parent- and child-type rovers, which shall be deployed for lunar crater exploration using tether-based connections to allow drilling and collecting samples.

2.3. Walking and Climbing Robots

In general, walking systems provide a high mobility, since they have the ability to position the ground contact points (i.e., the feet) nearly arbitrarily within the work space of the respective leg. This enables them to step over obstacles or to cling to foot holds in steep slopes. Furthermore, the lifting of a leg off the ground avoids the so-called bulldozing effect that wheeled systems have to cope with in loose soils.

The climbing robot Dante II demonstrated in extensive field experiments its capability of climbing into a volcanic crater (Bares and Wettergreen, 1999). The movements in the crater were partially remotely operated by human supervisors and partly autonomous, relying on onboard vision systems (laser-range finder and video cameras). The robot is a framewalker with eight legs. Additionally, a winch/tether mechanism is used to support the robot in steep slopes.

The LEMUR family of robots includes six-legged (LEMUR I and LEMUR IIa) and four-legged (LEMUR IIb) robots that are designed for use in orbital tasks as well as for exploring planetary surfaces (Bretl, 2006; Kennedy et al., 2001). The robots provide tool-exchange interfaces for reconfiguring the legs and equipping them with different tools for the task at hand. By making use of a stereo vision system, appropriate foot holds for freely climbing nearly vertical surfaces is made possible.

The Scorpion robot (Spenneberg and Kirchner, 2007) is an eight-legged system for traversing various terrain types, and it can still operate when suffering leg loss (Spenneberg et al., 2004). The robot makes use of a decentralized locomotion control approach. It is able to climb in steep slopes and can use the front pair of legs as manipulation devices. Although it was not specifically designed for a multirobot team, it was still able to act as a scouting robot in a scenario similar to the one presented in this approach (Cordes et al., 2010). Based on the experiences with Scorpion, different types of legged walking robots have been developed, e.g., the four-legged Aramies (Spenneberg et al., 2005) and the six-legged SpaceClimber (Bartsch et al., 2012). SpaceClimber is a walking and climbing robot that successfully demonstrated the locomotive abilities of multilegged robots in steep terrains. In contrast to previous robot designs, SpaceClimber's kinematics have been optimized using evolutionary computation (Roemmerman et al., 2009)—an approach that has been reused to optimize the rover's manipulator in RIMRES. The walking robot CREX used as a scout in RIMRES is based on the robot SpaceClimber and thus benefits from this series of developments toward walking robots specialized for steep terrains.

2.4. Standards and (Software) Technologies

A crucial part of reconfiguration and modularization and its reflection in higher levels of software is the underlying framework. Within RIMRES, the framework Foundation for Autonomous, Modular Systems (FAMOS) was developed, which builds upon and extends common standards in networking and control. In the following, some of the parts that constitute FAMOS are presented.

Service-oriented architectures (SOAs) are a common approach to support a modular software design and are inherently more robust, due to the component's single responsibility. A similar approach is also propagated by the Foundation of Intelligent Physical Agents (FIPA) (2002) and Consultative Committee for Space Data Systems (CCSDS) (CCSDS/AIAA Inc., 2012). While FIPA developed a standard for a complete abstract architecture, the CCSDS published a "Reference Architecture for Space Data Systems" (CCSDS/AIAA Inc., 2008). Multiple implementations of the FIPA standard exist with JACK (Winikoff, 2005), FIPA-OS (Poslad et al., 2000), Mobile-C (Chen et al., 2006), and JADE (Bellifemine et al., 1999), to name only a few. These frameworks implement the (experimental) FIPA standards to wide parts, but mostly use JAVA; Mobile-C as a rare exception. These implementations originate from the software-agent domain, and due to the choice of programming language they are not directly applicable to RIMRES, which relies in large part on C/C++ software. Another JAVA-based framework can be found with Cougaar (Snyder et al., 2004). Cougaar implements a blackboard-based communication for a multiagent system and allows for highly scalable systems. However, it does not rely on additional standards such as FIPA for building up its infrastructure.

Developing software for the robotics domain comes with a number of common and repetitive tasks, and it has triggered the development of different frameworks such as Microsoft Robotics Studio Developer (Microsoft, 2012), MIRO (Utz et al., 2002), Orocos (Bruyninckx et al., 2003), Rock (DFKI Bremen Robotics Innovation Center, 2011), ROS (Willow Garage, Inc., 2012), and Yarp (Metta et al., 2006). These frameworks usually wrap functionality in a single kind of component though using different terminology, such as task, node, or service. While all of these frameworks deal with communication, dedicated peer-to-peer connection management and publish-subscribe mechanisms exist. MIRO, Orocos, and frameworks that build upon those (such as Rock) include an application of CORBA (Common Object Request Brokering Architecture), and Yarp allows an easy integration of it. CORBA has been around since 1991 and has reached a level of maturity and broad acceptance.

Since CORBA requires a central name-service, it does not directly fit into a fully distributed setup. A tool that can act as a replacement for the central name-service in a distributed context—and has been used in RIMRES—is Avahi (Poettering et al., 2012). As a so-called zeroconf solution (IETF Zeroconf Working Group, 2011) and operating on the two complementary technologies mDNS (Multicast Dynamic Name Service) (Cheshire, 2011) and DNS-SD (DNS-Service Discovery) (Cheshire and Krochmal, 2011), Avahi allows name-based resolution for service records, which can be detailed using multiple, customizable text records. In addition to a simple name-service, it also allows to detect when a service is started or stopped.

Merely establishing communication is not sufficient, and thus FIPA allows for reasoning on communication using a message specification in combination with so-called performatives. Communication between two agents is looked at as a speech-act, and interaction protocols, e.g., for a contract net implementation, allow validation of the message flow. Lyell et al. (2009) already applied these standards in the domain of space robotics.

3. DESIGNING FOR RECONFIGURATION

Assuming the usage of mobile robots for further exploration of volatiles in the lunar polar cold traps, some basic requirements for the system design can be established. The systems have to be robust and reliable in order to survive the harsh conditions on the lunar surface. The mobile units have to provide and sustain a general framework for the planned scientific experiments. This includes the power system, environmental protection, the capability of placing the right instruments in the right place, and safely transporting measurement equipment to designated places. For a maximized impact, the amount of local surface coverage is also a determining factor. The systems should be able to cover distances on the order of several tens of kilometers. The locomotion should be efficient, since power in the permanently shaded regions is a critical resource. In general, the system has to be able to react to unforeseen circumstances in an appropriate way.

Some of these requirements are partly contradicting, e.g., an energy-efficient surface coverage in the order of tens of kilometers might not coincide with a system that is capable of climbing steep slopes that often are to be found in areas of scientific interest, i.e., target areas for exploration and in-depth measurements. However, by combining the benefits of heterogeneous systems into one overall robotic team, the respective strengths of each system can be exploited to maximize the outcome of a mission, and a larger safety margin is maintained—upcoming problems can also be handled by collaboration, e.g., wheeled locomotion is more efficient in general in terms of energy consumption than legged locomotion, while a legged system is more appropriate for challenging terrain types. Combining both locomotion types appears to be logical, and thus in RIMRES one exploration system consisting of a main rover (a wheeled system) and a legged scout robot is designed for the task of exploring the inner part of the lunar polar cold traps.

To meet these challenges, the usage of heterogeneous reconfigurable systems is proposed, an approach the authors already successfully presented within the project LUNARES and extended substantially with the RIMRES system (Cordes and Kirchner, 2010). The proposed systems are modular and allow for physical reconfiguration, i.e., they are able to react with the physical adaptation to different challenges that might occur during mission time. The modular design also allows for extension in successive missions, so that exploration missions and hardware can be gradually implemented to build a lunar exploration infrastructure and prepare for human presence on the moon.

Since reconfiguration is a main characteristic of the system design, the following section gives a theoretical discussion on that topic in the context of strategic adaption.

3.1. Strategic Flexibility

One of the main elements in the project RIMRES is the integration of reconfiguration capabilities into a system of heterogeneous robots to allow for "strategic flexibility" (Evans, 1991). This is a major distinction from similar projects, since reconfiguration becomes an integral part of the system design.

Reconfiguration comprises three essential states: (1) an actual (stable) configuration, (2) a (stable) target configuration as a result of the reconfiguration process, and (3) an (unstable) transition phase between the two configuration states. This process description does not give any idea as to how significant the changes of a single reconfiguration might be, yet a transition has to involve changes, and in that context reconfiguration can be viewed as a "controlled type of evolution" (Dunin-Keplicz and Verbrugge, 2010). The desired outcome of the reconfiguration—the target configuration—is known, while the transition might be initially unknown but needs to be performed in a controlled manner to produce the desired outcome. Thus, knowing the start and target configuration requires the application of a planner to outline the transition phase and minimize the side effects of such a configuration. While a planner is required for an autonomous exploitation of this kind of flexibility, RIMRES uses predefined semiautonomous action sequences or lets a human operator perform a transition.

Reconfiguration can be found in a variety of domains—robotics being just one—and organization theory has studied the issue of reconfiguration as part of improving upon strategic flexibility. The goal of improving strategic flexibility lies in providing a better response to external changes, e.g., a market, suppliers, etc., but in general it involves improving the characteristics of an organization in terms of adaptability, flexibility, robustness, or resilience. For these terms, the authors use the semantic description

Table I. Terms of strategic flexibility as defined in Evans (1991).

adaptability:	"a singular and permanent adjustment to a newly transformed environment"
flexibility:	"the ability of successive, but temporary approximations to this state of affairs"
robustness:	"a system's ability to absorb, deflect, or endure the impacts of unanticipated changes"
resilience:	"the tendency to rebound or recoil, [...] and the capability to withstand shocks without permanent damage or rupture"

as listed in Table I and collected by Evans (1991), which serves as the authors' primary source for the discussion on strategic flexibility:

Primarily and following Evans (1991), an improvement of these characteristics produces a higher DOF for an organization and allows for a better operational range. The additional DOF increases the solution space, and an organization has better chances to find an appropriate solution to upcoming and potentially unforeseen problems. Yet, for more advanced reconfiguration and comparison of reconfiguration strategies, the response time or the speed of tackling and solving a problem will become a decisive factor for either success or failure (Dignum, 2009). The complexity of the discussion increases when looking at additional dimensions of strategic flexibility in the temporal domain and considering reactiveness, proactiveness, and a differentiation between offensive and defensive actions. For now, the authors will leave out a detailed discussion of these additional dimensions and focus an applying the basic capability of reconfiguration.

Transferring these findings to a physical system, RIMRES targets the increase of the DOF of the overall system by embedding reconfiguration capabilities at hardware and software levels—initially aiming at higher adaptability and resilience.

3.2. Dimensions of Change

Adaptation of the systems in RIMRES involves change and affects both hardware and software. On a low level, reconfiguration of hardware can involve the exchange of mechanical parts or a rearrangement of physical links. Similarly for software, a change of the setup of running components and relinking communication channels and data processing chains is a low-level reconfiguration. On a higher level, reconfiguration can take advantage of the parametrization of components using hardware switches or configuration properties of software modules, and it allows for more sophisticated reconfiguration approaches. This can be compared to an online system optimization—again supporting the view of a "controlled type of evolution." Generally, the authors are looking at adaptation in the dimensions listed in Table II.

In most of the cases, a change in structure is followed by a change of functionality. On the one hand, this can result in an extension of available functionalities, but on the other hand particular functionalities can also be disabled since they cannot be performed with the new system structure.

3.3. Reconfiguration Examples

Reconfiguration of (robotic) systems can be considered in almost all phases of system design and at all levels of system architecture. Considering the basic reconfiguration capability, i.e., reconfiguration by exchange of (structural) parts of the system, one has to account for (1) mechanical interfaces allowing reconfiguration and (2) a mechanism to perform reconfiguration. Industrial robots provide a practical example by exchanging tools as preparation for different tasks. This kind of reconfiguration is tightly connected to the modularity of systems and the one targeted in this project.

A core element for mechanical reconfiguration in this and other projects is the design of an EMI (see Section 4 for details). This interface allows to connect to previously independent systems. However, reconfiguration in general also contains less extreme examples. Accounting for different locomotion modes and a morphology change of a system is an example with a lower impact on the overall system structure. The change of morphology can be of special benefit for improving locomotion capabilities for specific terrain types or tasks. For example, the Scarab rover (Bartlett et al., 2008), while driving, uses the suspension system for leveling the robot's body in changing slopes. In the so-called inch-worming locomotion mode, the suspension system is actively used to increase the locomotive abilities of the system in steep slopes. Furthermore, the suspension can be used to lower the body of Scarab to the ground in order to prepare for drilling the lunar surface.

The system designer accounts for predefined reconfiguration options using a modular architecture, and reconfiguration can almost always be achieved by (re)using parts of the system in other ways than originally intended. The Hayabusa mission is one prominent example in which reconfiguration and reuse of structural parts of the system were successfully applied to lead the overall mission to success. In this case, anomalies were detected in one of the thruster engines, but by reconfiguring the two engines the return cruise of the space craft to Earth was made possible (JAXA, 2009).

Thus, embedding reconfiguration options into a system is not a novel idea and is already present in various applications, e.g., to perform error recovery or situation adaptation. However, existing applications operate either at a very low level or use systems with lower complexity, e.g., activities of swarm-based research take a very general approach to reconfiguration, at the price of practicality and decreased

Table II. Dimensions of change.

	physical/hardware	virtual/software
structural	change of morphology, tool exchange	change of distribution of modules across physical devices, reorganizing and relinking data flow, changing dependencies for running components
functional	tool exchange, modes of operation: wheel also being used as foot or sensing device, manipulator also being used as supporting leg	modalities, application of various solution strategies, parametrization of components, e.g., adaption of thresholds, configuration parameters in a signal processing chain
mixed	change of morphology changes the set of active capabilities, and for exploitation requires adaption of the high-level software stack	

system performance, while this project tries to maintain the specialized capabilities of robots like Sherpa and CREX and provide the capability for structural reorganization of these systems at the same time.

4. RIMRES – A SYSTEM OF SYSTEMS

The goal of RIMRES is to try to simulate typical elements of a situation in an exploration mission and/or infrastructure buildup. The mission is operated from an earth-bound (mission) control center, which communicates with a system control station at the lunar surface. This system control station is the focal point for communication of all robotic systems that are part of the mission: in RIMRES this encompasses two mobile subsystems as well as immobile payload-items and assembled payloads.

The two mobile subsystems are a wheeled rover and a legged scout. Both systems can act completely independently from each other, but at the same time a close electromechanical connection between both systems can be established combining both separated systems into one functional unit. Further reconfiguration abilities are added by the introduction of modular payload-items that (1) can extend the capabilities of the mobile systems or (2) can be used to create payload stacks[3] during the mission. These payloads can either be part of a science mission or represent basic infrastructure elements, e.g., for communication. For the RIMRES scenario, four types of so-called payload-items are pursued:[4] (1) a battery module for extending the range of the mobile units and for powering the assembled science packages, (2) a camera module, simulating a data-generating science payload, (3) the mole subsurface sampling system that already flew on the Beagle-2 mission, which was planned to be implemented in the RIMRES framework, and (4) a communication/navigation item (REIPOS[5]) (Bindel and Bruns, 2010). The wheeled rover serves as a transporter and provides a manipulation arm. This manipulator allows to combine payload-items and to deploy individual payload-items or payload stacks in the lunar environment. Alternatively, payload-items can be attached to the legged scout.

The overall system in RIMRES serves as a technology demonstration and is used under earth conditions. For demonstration and validation, an artificial lunar crater (surface area 105 m^2) with realistic slopes and lighting conditions has been set up in the DFKI laboratories. Thermal management, radiation, and other environmental issues are not explicitly taken into account at this stage of development.

While a mission with the RIMRES system can be arbitrarily complex, the following outline of actions illustrates a feasible mission. The rover starts transporting the scout and five payload-items—three battery modules and two science modules—to the rim of a lunar polar crater. As mentioned, the rover's manipulator can be used to assemble scientific payloads from payload-items, and on the way to the crater rim it deploys two payloads consisting of one science and one battery module each. Furthermore, during transport with the rover, the scout is fully functional, thus its scientific instruments can also be used during this phase to probe the terrain. With one battery module and the scout attached, the rover reaches the crater rim, where it detaches the scout and deploys an additional battery module on the back of the scout. Subsequently, the scout climbs into the permanently shaded regions of the crater to conduct *in situ* measurements in search for water ice or other volatiles. During the travel to the crater rim, the scout is supplied with energy from Sherpa via the EMI, so that it requires its own internal energy supply only for the descent into the shaded crater and the return. While the scout is exploring the crater, the rover can drive back to the lander to pick up additional

[3] In the context of RIMRES, payload-items are single cubic modules that can be stacked to form scientific or infrastructural elements, which are called payload stacks or payloads.

[4] While the battery module and the camera module are integrated to the status of fully functional payload-items, the REIPOS-system is in the state of a laboratory example, and the setup of the mole module has only been investigated theoretically.

[5] Relative Interferometric Position Sensor, development by the project partner ZARM.

payload-items available at the lander to extend or maintain the infrastructure and experiments. It furthermore uses the lander's infrastructure to replenish its own energy supplies. Eventually, the rover will reunite with the scout at a designated meeting point at the crater rim to continue the exploration from a different starting point.

In RIMRES, reconfiguration aspects are part of various layers. First, the overall system and team of robots can be reconfigured by either stacking of payload-items (onto each other or onto the mobile systems) or by docking the legged scout and the wheeled rover. Secondly, on a subsystem level the individual systems are capable of different operating modes, which the authors also describe as reconfiguration property: (1) the wheeled rover can be reconfigured in the sense that the active suspension system can be used in various ways to propel the robot; (2) in addition, its manipulator can be used for handling the payload-items as well as for locomotion support, system inspection, and system supervision; and (3) the legged scout is reconfigurable in the sense that the legs used for locomotion are equipped with gripper elements, and using a gripper mode they are able to pick up geological samples at a site of interest.

The following paragraphs describe the single systems of RIMRES in more detail. The EMI as the central part of the system is described in Section 4.1, the closely related payload-items are described in Section 4.2, while the rover as a main mobile unit is presented in Section 4.3. A description of the scout is provided in Section 4.4.

4.1. Electromechanical Interface for Modular Reconfiguration

The electromechanical interface developed in RIMRES is the central device for interconnecting subsystems with each other. Thus, it is of special importance for realizing the reconfiguration capability in RIMRES. The design of the EMI was driven by the requirements of establishing a reliable and robust physical connection between two systems (Wenzel et al., 2011), i.e., allowing Sherpa to carry CREX, building subsystems of combinations of payload-items, and at the same time providing data and energy transfer between systems via this interface. Additionally, to fulfill these requirements in a lunar environment, dust-resistance of the interface was a primary objective, and experiments as illustrated in Figure 3 proved the working of the interface even under these extreme conditions.

The current design of the EMI—achieved after multiple iterations—provides a secure mechanical connection and routes data signals as well as energy in a combination of subsystems. The interface consists of a male (passive) and a female (active) part as shown in Figure 2. Apart from the mechanical parts that are displayed in the figure, a dedicated electronics board is part of the EMI. A microcontroller controls the latch mechanism and the illumination LEDs of each bottom interface. When two systems get connected, these microcontrollers establish local communication (LOC) in order to gather topology information and route high-level commands. Furthermore, the module electronic provides power management within a system connected via an EMI. Details on the general concept, mechanical design, and power management can be found in Dettmann et al. (2011), Wenzel et al. (2011) and Wang et al. (2011).

4.2. Modular Payload-Items

Specifically designed to allow construction of additional payloads, or to extend the functionality of some of the main systems, payload-items are cubic modules with an active EMI in the bottom face and a passive EMI in the top face (Wenzel et al., 2011). By stacking the payload-items, different scientific payloads and infrastructure elements can be assembled. All payload-items come with a processing unit (Gumstix) to run the high-level software framework and a microcontroller to support low-level intelligence, e.g., to communicate with an EMI. As part of the low-level intelligence, an internal communication protocol has been designed allowing to infer the current topology of a stack of payload-items from the EMI connections, and control basic operations such as opening and closing the mechanic latch to attach an active EMI to a passive one. These capabilities are exposed to higher levels of control to allow for more complex reconfiguration activities leading to various system combinations, as illustrated in Figure 4.

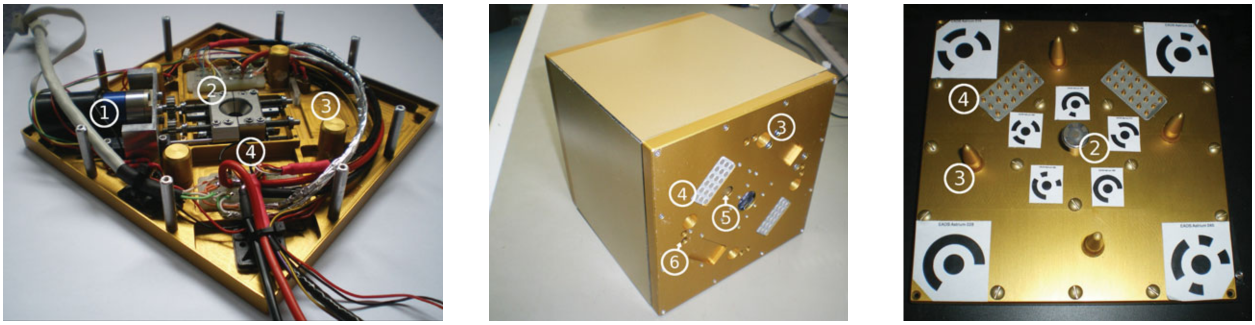
The following sections briefly describe the specialized module types that have been developed in RIMRES.

4.2.1. Battery Module

Within the earth demonstration scenario of RIMRES, battery modules are used as replacement for energy-harvesting payload-items. In later stages, additional solar modules for actually harvesting energy are conceivable. A battery module always constitutes the basis of a payload stack, since functional modules and energy modules are separated within the modular framework of this project. The battery module comprises power switching intelligence and therefore it is possible to connect multiple systems and multiple battery modules at the same time. Each payload-item can be a power sink while the battery modules can be a power source as well. To protect the systems from uncontrolled charging and connecting two power sources with different power levels at the same time, a power management system (Wang et al., 2011)—as previously mentioned—has been set up.

4.2.2. Science Modules

To simulate science payload-items, a camera payload-item serves as a primary example for a science module. The camera payload-item is a placeholder for more sophisticated scientific equipment, but it demonstrates the core feature:

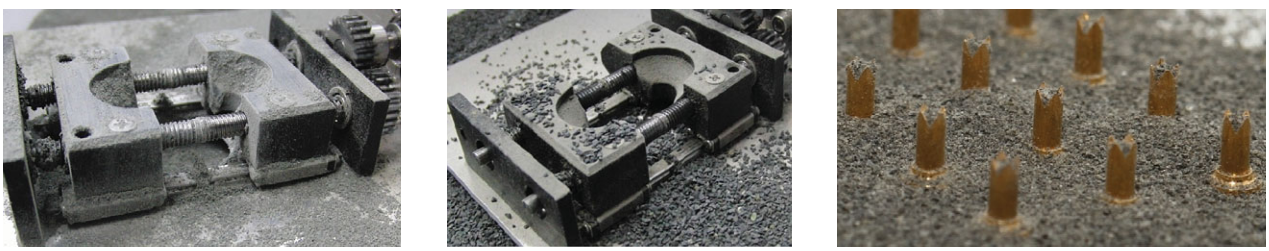


(a) Inside of the bottom interface

(b) Bottom interface from the outside

(c) Top interface with unique visual servoing markers

Figure 2. Description of the RIMRES electromechanical interface in real implementation of a payload-item (cube with a side length of 15 cm): (1) actuator for latch mechanism, (2) bottom latch mechanism and top central connection pin, (3) bottom receptor cylinder and top guidance pin as counterpart, (4) contact blocks for electrical connections via 18 pins, (5) camera opening [camera is not illustrated in (a)], and (6) LED openings.



(a) Latch mechanism after closing and opening with basalt of grain size $0.02 - 0.2$ mm

(b) Latch mechanism after closing and opening with basalt of grain size $0.7 - 1.3$ mm

(c) Close-up of crown-shaped electrical pins when covered with basalt dust

Figure 3. Situation after covering movable parts of the EMI with amounts of dust exceeding by far the expected amount in normal operation.

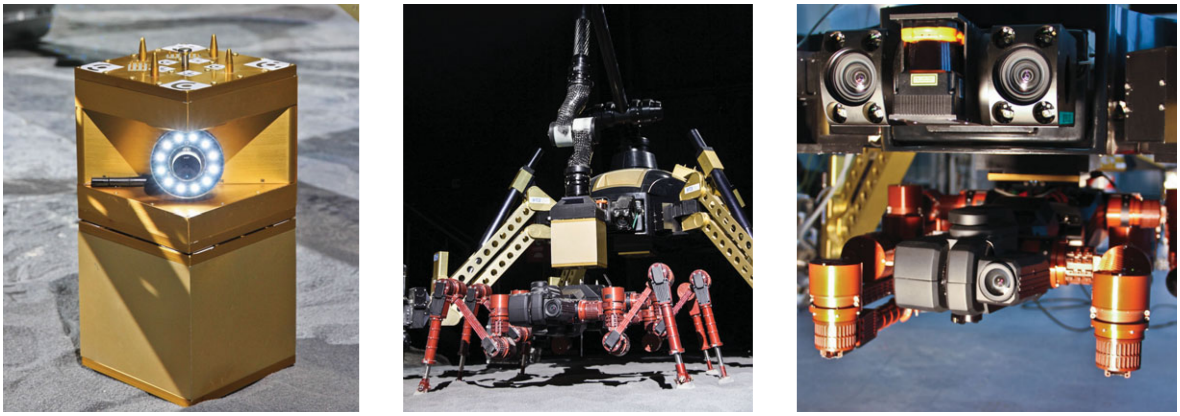
attached to a battery payload-item, it can form an active payload. It can receive control commands from the mission control, e.g., to set the orientation of the camera that is mounted on a rotational table, and it provides image data. Thus, this example payload allows to verify the process for data acquisition and distribution, and communication of high volume data from the payload-items to the system control station via WLAN communication. Using the general communication framework that is used by all subsystems, this payload is seamlessly integrated into the system control and communicates within the software framework using the same means as CREX and Sherpa.

4.3. Four-wheeled Rover Sherpa

The wheeled rover Sherpa is the key team member in the presented multirobot system. Only Sherpa is capable of assembling payloads (on demand) using the manipulator arm attached to the central body. It is also capable of transporting the legged scout to the crater rim and transporting payload-items, thus increasing the reach of less efficient or even immobile systems.

Sherpa makes use of an active suspension system that allows to select from a set of locomotion modes depending on the current terrain situation. These modes range from various postures to enhance the relationship between the center of gravity and the center of the support polygon to substantially different drive modes, for example planar omnidirectional movements or inchworming modes (Cordes et al., 2011). Figure 5 displays the final state of the integration of Sherpa. The key properties of Sherpa are summarized in Table III.

Sherpa shows great flexibility to adapt to various terrain conditions. The active suspension allows to adapt the footprint of the rover according to the challenges the current terrain imposes on the rover. This can also be interpreted as a posture reconfiguration. Furthermore, the active suspension can be used to propel the robot: instead of just using the wheel actuators, the suspension actuators can be incorporated into locomotion, as, for example, in an inchworming fashion or for (short traverses of) undulating behaviors.



(a) Active payload composed of battery payload-item and camera payload-item. This stack simulates a data generating scientific payload in the RIMRES context.

(b) CREX is being equipped with a battery payload-item for extending its range of operation. The payload-item is handled with Sherpa's manipulator.

(c) CREX is docked to Sherpa via Sherpa's Bottom-EMI.

Figure 4. Electromechanical interface (EMI) and payload-items in RIMRES: Used to form scientific payloads, to extend the mobile systems capabilities, and to interconnect the mobile systems.

(a) Sherpa using its active suspension to step onto an obstacle. The arm was used to support the robot while lifting each of the front legs.

(b) Sherpa using its manipulator as a fifth limb. The manipulator is strong enough to bear the weight of the robot, when two legs are lifted off the ground.

Figure 5. Photographs of Sherpa, hybrid wheeled-leg mobile rover in RIMRES.

Another main property of Sherpa is the manipulator arm attached to the rover's main body. Its primary use is to handle payload-items that are attached to the four EMIs located around the central tower. By manipulation of payload-items, various scientific and infrastructural payloads can be assembled. Furthermore, the arm can be used as a fifth limb, thus reconfiguring an arm into a leg; cf. Figure 5(b). The manipulator's palm camera is normally used for grasping the payload-items in a visual servoing process, but it can be used to allow a human operator to supervise the rover system. Additionally, payload-items attached to the arm can further extend the functionality of the manipulator, e.g., for scooping or sophisticated gripping (these types of payload-items are not part of the development in the RIMRES project). Details of the manipulator design are provided in Manz et al. (2012).

In the final stage of expansion, the wheels[6] are planned to be adaptable subsystems of the rover. In the current stage of development, however, the wheels are flexible metallic wheels with passive adaptation to the ground (Kroemer et al., 2011). Figure 5 displays the wheels mounted on Sherpa. Similarly to the manipulator, the rover's functionality can be extended using payload-items. For

[6]The wheels are a development of the project partner DLR-RY.

Table III. Key dimensions of Sherpa.

Description	Value
Max. ground clearance	711 mm
Min. ground clearance (wheels above body)	−189 mm
Square-shaped footprint in cross stance	2100 mm (high stance) to 2500 mm (body low)
Mass (w/o scout or payload-items, incl. manipulator)	approx. 160 kg
Mass of manipulator	25 kg
Length of fully stretched arm	1955 mm
Max. static load on stretched arm (stretched wrist)	183 N
Max. static load on stretched arm (hanging wrist)	537 N

Table IV. Key dimensions of CREX.

Description	Value
Min./max. body height	150 mm/400 mm
Min./max. longitudinal body shift	−150 mm/150 mm
Min./max. lateral body shift	−50 mm/50 mm
Dimensions in standard posture [L×W×H]	850 mm × 1000 mm × 220 mm
Stretched leg length (front and rear)	640 mm
Stretched leg length (middle)	650 mm
Body dimension (incl. head, central joint in neutral pos.) [L×W×H]	895 mm × 208 mm × 165 mm
Mass (with battery)	27 kg

example, additional sensors can be attached via one of the fixed EMIs of Sherpa that are attached to the main body. Currently, the authors assume an additional battery pack to extend the operational time of the system, or scientific payloads attached to the docking interface beneath Sherpa.

4.4. Six-legged Scout CREX

The six-legged scout CREX is the second mobile system in RIMRES. It is based on the SpaceClimber robot (Bartsch et al., 2012) and adapted to the requirements of the multirobot system RIMRES, e.g., to carry payloads or dock to Sherpa, an EMI has been placed at the back of CREX. Further improvements compared with SpaceClimber have been made concerning the mechanical design of the single joints and the lower legs as well as a new sensor head with two degrees of freedom for the camera and the laser range finder.

Apart from the reconfiguration of the overall system by (un)docking CREX and Sherpa, CREX also provides several reconfiguration capabilities by itself. First, gripping elements are employed in the front legs in order to be able to pick up geological samples. Thus, the legs used to propel the robot can be reconfigured to be used as manipulation/sampling devices.

Via the EMI on its back, CREX can be connected to the wheeled rover, Figure 1(b). However, the EMI can also be used in the same manner as on Sherpa: arbitrary payload-items can be stacked onto CREX to extend its capabilities. This ranges from additional batteries to specialized sensors for a task at hand. In later stages, a second EMI on the belly of the scout system is conceivable, allowing to dock specific and bigger sampling devices. By using the high degree of mobility of the system, these devices can be positioned precisely over a spot of interest. Figure 6 shows the integrated scout robot CREX in DFKI's Space Exploration Hall.

4.5. Combinations of Subsystems

Table V displays the currently possible physical combinations of subsystems in RIMRES. Note that the combination rover-manipulator is static in the current setup[7] and here illustrates a theoretic modularization. Otherwise, the table shows the range of reconfiguration the system is currently capable of. The connection rover-rover refers to the possibility of connecting a rover's manipulator to the payload-bay of another rover, which—due to the lack of a second rover—is not part of RIMRES. The example of transporting CREX with the manipulator is illustrated in Figure 7(a). Specific capabilities of the systems can be improved, but also disabled

[7]However, the manipulator is detachable (by loosening the bolt flange) and was used for development purposes as a singular unit without the rover.

(a) CREX robot in artificial crater environment. CREX is equipped with an electromechanical interface for attaching to Sherpa and for carrying payload-items.

(b) CREX beneath Sherpa after release from Sherpa's electromechanical interface (yellow square in the center of Sherpa's belly)

Figure 6. Photographs of CREX: Scouting robot in RIMRES.

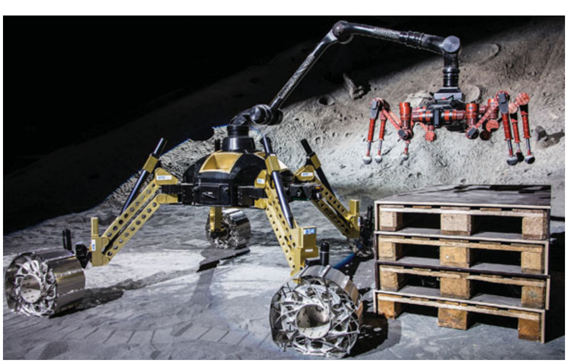
Table V. Possible combinations of subsystems in RIMRES. In principle, it is possible to connect the manipulator of one rover to a payload-bay of another rover, resulting in the check mark for the rover-rover connection.

	Rover	Scout	Manipulator	Payload-Item
Rover	✓	✓	✓	✓
Scout	✓	×	✓	✓
Manipulator	✓	✓	×	✓
Payload-Item	✓	✓	✓	✓

in specific configurations, e.g., a combination of Sherpa and CREX is more limited with respect to terrain difficulty it can traverse—the maximum ground clearance is lower. However, the main benefit of this cooperation is energy-efficient transport on rather planar surfaces and over long distances. Though possible, the authors do not expect this monolithic configuration to be applied in very rough or steep terrain.

5. SOFTWARE FOUNDATION FOR A RECONFIGURABLE SYSTEM

The project RIMRES serves as a terrestrial demonstrator and assumes a traditional setup of a ground/mission control station. As already mentioned, a system control station at the lunar surface represents the focal point for the communication of the robotic team, and represents the main link to the Earth-bound mission control. The use of this control station introduces a centralized control approach in the first place, since all robots need to communicate with the system control station. However, to achieve robustness, a



(a) Photograph of Sherpa using its manipulator to lift CREX. A possible use-case is the deployment of CREX off a landing unit. Furthermore, a reconfiguration-scenario where CREX is used as a six-fingered hand is also conceivable.

(b) CREX docked to Sherpa. CREX has four orientation possibilities to dock to Sherpa (in 90° steps). In this image CREX is oriented with its head toward the primary movement direction of Sherpa.

Figure 7. Feasible combinations of Sherpa and CREX.

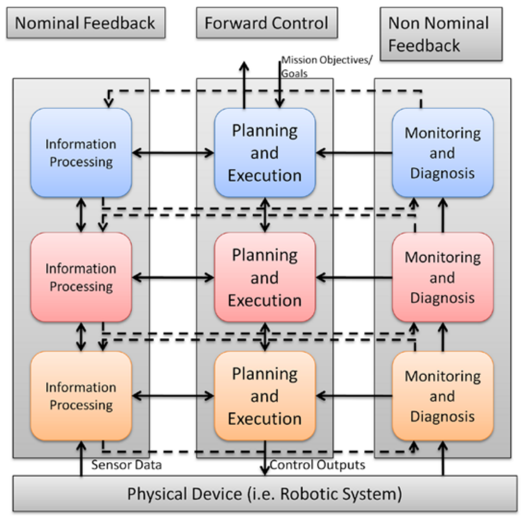


Figure 8. FRM model with three control layers, following Visentin (2007).

distributed setup has been selected for the robotic team using peer-to-peer communication. This communication setup minimizes the effects of a single-point of failure and accounts for flexible robot-to-robot interaction schemata when a central communication hub is not available.

As a project targeting a space application, the project RIMRES embeds ESA's Functional Reference Model (Ferrarini and Carpanzano, 1999; Visentin, 2007) as illustrated in Figure 8 as a general architecture model. This architecture comprises three layers: subsystem control (Level A), task control (Level B), and mission control (Level C).[8] For the scenario the authors assume a predefined mission sequence, which can be split into several main tasks. Each task again can be split into sequences of trivial to complex actions, e.g., a trivial probe action allows to verify the communication between two distributed systems and validates the full communication stack, while a complex and even cooperative docking action commands a main system to start and control a docking maneuver.

The mission control is responsible for scheduling actions, while the actual management is done via system control, which applies a forward control to the subsystem level. Subsystems in the context of RIMRES are represented by the robots Sherpa and CREX, as well as payload-items or payloads.

The team of robots will be operated from a mission control center, and three different operation modes are considered: (1) manual operation: the team of robots executes given actions or action sequences that are forwarded by the mission control center to achieve a certain objective, (2) semiautonomous operation: the mission control relies on (complex) task sequences to achieve a mission objective, or to reconfigure systems to compensate for errors, and (3) autonomous operation: mission control or system control fails to operate or cannot communicate with the robotic team—the architecture allows for self-organization of the robotic team, either to continue with the still known objectives or to reestablish communication with the mission control. All autonomous operations can be interrupted by intervention from an operator, representing the so-called human-initiated switch between autonomy modes. Meanwhile, the architecture also accounts for a system-initiated switch, i.e., allowing subsystems to request an interaction by their operator.

In the following, the authors describe the approach taken in RIMRES to design the subsystem control level (Level A) from three different perspectives: the intrarobot, interrobot, and mission control perspectives.

[8]The mission control infrastructure for levels C and B has been developed by the project partner EADS Astrium.

5.1. Intrarobot Architecture

The intrarobot perspective has its focus on the individual robot and management of robot resources. The software stack applied on a single robot in RIMRES is mainly based on Rock (DFKI Bremen Robotics Innovation Center, 2011), which itself uses the Orocos Realtime-Toolkit (RTT) (Bruyninckx et al., 2003; Soetens, 2012) for its components. Rock uses a model-based approach to create an infrastructure of software components, and designing a system with Rock has proven to be useful not only for the RIMRES project, but for multiple others at the authors' institute that try to solve complex tasks.

5.1.1. Components and Compositions

The first step toward designing the subsystem control level and enabling the reconfiguration capability in the software architecture is the development of appropriate, specialized drivers for the hardware. These driver libraries are then used or wrapped in Orocos components. Each such component (or software module), which has dedicated input and output ports, allows for remote procedure calls and can perform a specific task. Components are specified using an (oroGen) model description. For example, Figure 9 illustrates the specification of the system monitor component. The system monitor component depends on a number of libraries and inherits functionality from a component called system_state, which is a key value store that is accessed to apply a given rule-base. The main driver that is wrapped by this component implements the rule-engine. The component outputs monitoring events as soon as a rule fires, and the event can be used and interpreted by other components, allowing to isolate the pure monitoring functionality to this specific component.

```
name "system_monitor"
version "0.1"

using_library 'rule_engine'
using_library 'base-lib'
using_library 'utilmm'

using_task_library 'system_state'
import_types_from "SystemMonitorTypes.hpp"

task_context "Task" do
   subclasses 'system_state::Core'
   needs_configuration

   property("rulebase", "/std/string").
       doc("File which contains the rule descriptions")

   output_port("monitoring_events", "system_monitor/MonitoringEvent").
       doc("Output port for the monitoring events")
end
```

Figure 9. The oroGen specification of the system monitor component.

Components can be viewed as the lowest level of modularity in the software stack and are specialized to fulfilling one task. Aggregating multiple such components allows to solve more complex tasks, but since this aggregation can get very complex, a plan manager (Joyeux et al., 2010) is applied to provide supervision. This supervision can be easily applied to manage component networks since the interfaces of the underlying Orocos tasks are specified by the models. Figure 13(a) outlines the basic set of components available in RIMRES.

So-called compositions specify component networks in detail, i.e., they define required components and the data flow between these components. Figure 10 illustrates the specification of a composition to connect producers and consumers of monitoring events. Compositions can be nested, so that a single composition can be designed either in a very general way to allow reuse in other compositions or rather to fulfill the special requirements of one application. The definition of a composition can be extended to handle component events and by introducing custom events. This way compositions can wrap complex capabilities of the underlying system.

A single component can be part of multiple compositions as long as the same component configuration can be used, and the supervision evaluates if there is a need for a specific component to run and stops it otherwise. Hence, the supervision also represents a resource-saving means. However, having a single component that is part of multiple compositions can lead to conflicts that may arise at runtime; a parallel usage might be impossible due to different required configurations. The supervision is responsible for detecting such conflicts and thus prevents the startup of invalid network configurations.

The active management of the component network eventually allows to control granularly (1) which instances are up and running, and (2) the data flow, i.e., while in one configuration a camera might forward its images only to a visual servoing component, in another setup it forwards them to the mission control. The development and architecture levels involved to create a robot action are illustrated in Figure 11.

5.1.2. Reconfiguration at Various System Levels

In RIMRES, aspects of reconfiguration can be found at various levels of the hardware and software design. Clearly, the EMI represents the essential piece of hardware toward designing a heterogeneous reconfigurable multirobot system. The EMI allows a reconfiguration and combination of robots such as Sherpa and CREX, and allows to extend the robots' functionality using existing payload-items. In addition, subsystems can appear at runtime of a mission by creating payloads from existing payload-items. Depending of the number of existing payload-items, it will be undesirable to account for all possible permutations of such system. Nevertheless, the standardized mechanical interface allows to easily extend a system with new add-ons, which can even be designed after the mission has already been started and the robots have been deployed.

An EMI allows to query for attached neighboring devices and provides an interface to transparently communicate with all payload-items attached to it. To handle EMI information programmatically and embed it into high-level processing, each EMI can be uniquely identified in the overall robot team; a low-level device id is used for this purpose and mapped to individual systems as part of a static

```
composition 'Monitoring' do
    add SystemCore::Task, :as => 'system_core'
    add Monster::InterfaceModule, :as => 'monster'
    add SystemMonitor::Task, :as => 'monitor'

    connect monster.telemetry_out => monitor.monster_telemetry_in
    connect monitor.monitoring_events => system_core.monitoring_events
end
```

Figure 10. Component model of the monitoring composition.

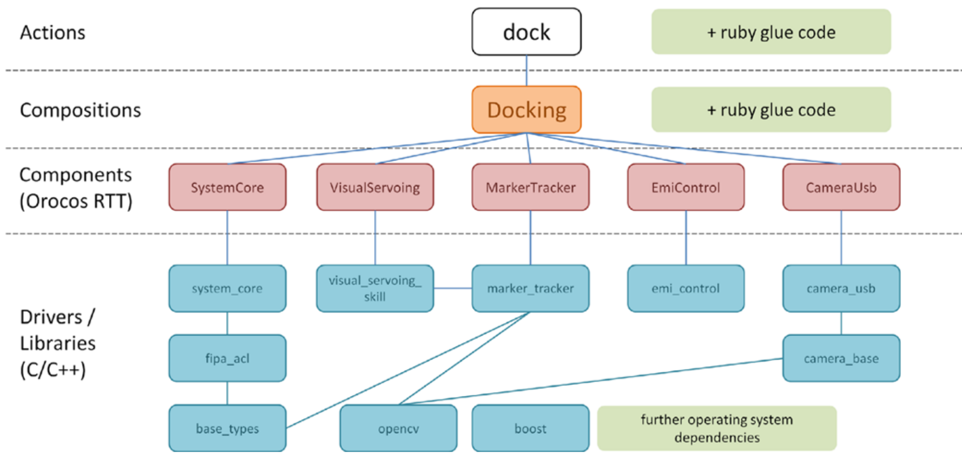


Figure 11. Development levels on Sherpa to allow for the dock action - which starts and controls the reconfiguration procedure to dock CREX.

configuration. This allows to infer the current physical configuration state by gathering the stati from all available EMIs attached to a robot.

For inference of the physical configuration state, an organization model has been implemented using a Ruby-based domain specific language (DSL). At the current stage, the organization model represents the physical connection status of the team of robots, and is based on the agent-group-role model (Ferber and Gutknecht, 1998). Currently, all systems that serve as a basis for a larger system are represented as a group that comes with a persistent actor, e.g., the group *sherpa* requires a single robot Sherpa as a persistent actor, and similarly for CREX and payloads. Figure 12 outlines the current description of the organization model for RIMRES and includes an instance description, i.e., defining dedicated actors and groups. By querying the EMIs for available neighbors with a regular frequency, a monitoring of the configuration state can be achieved. Thereby, a system can validate a successful reconfiguration and can also recognize dynamically attached payload-items or docked systems (in the case of Sherpa and CREX). While a dynamic detection might not be necessary in a space application context, it still allows to validate a reached configuration state in an event-based fashion.

While generic tools such as the supervision contribute to reconfiguration capabilities, further individual component design does as well. The telemetry provider serves as another example and represents the adoption of recommendations given by the CCSDS (CCSDS/AIAA Inc., 2008). The telemetry provider acts as a generic packaging component to support multiplexing of the sensor data streams toward the mission control station. Acquiring sensor data and forwarding to the mission control center might be costly in terms of processing power and energy, and a continuous delivery of all camera images will not be feasible regarding communication bandwidth. Thus, mission control has to carefully select active devices such as cameras. The actual transferred data can be of any format, i.e., images are provided in an internal (standardized) binary format, which adds meta information to a jpeg encoded image. On an operator's request, devices will be activated and the output data stream of the sensor is dynamically attached to the telemetry provider component. This component converts the images to the target format as expected by the system control station and adds them to a generic telemetry container package. This container package is then forwarded to the system control station and can be transparently routed through the network without a need for inspecting the payload data. At its final destination, the package is unwrapped and split into the sensor-specific packets. While this is a rather common procedure, this dynamic multiplexing of sensor data allows to manage the existing sensor data in a dynamic fashion, leading to an efficient use of a robot's limited resources.

```ruby
require 'famos'
include Famos::Organization

domain 'rimres' do
    actor_model :payloaditem

    group_model :payload do
        requires Role => :payloaditem, :min => 2 do
            requires ActorModel => :payloaditem
    end

    group_model :sherpa do
        persistent_actor

        # Is associated with the bottom interface
        requires Role => :client, :min => 0 do
            requires ActorModel => :crex
        end

        requires Role => :extension, :min => 0 do
            requires ActorModel => :payloaditem
        end
    end

    group_model :crex do
        persistent_actor

        requires Role => :extension, :min => 0 do
            requires ActorModel => :payloaditem
        end
    end

    # Initial configuration/setup of the rimres scenario
    group_instance :sherpa => "sherpa_0"
    group_instance :crex => "crex_0"

    group_instance :payload => "payload"

    actor_instance :payloaditem => "payloaditem_0"
    actor_instance :payloaditem => "payloaditem_1"
    actor_instance :payloaditem => "payloaditem_2"
end
```

Figure 12. Organization domain model combined with the organization instance description.

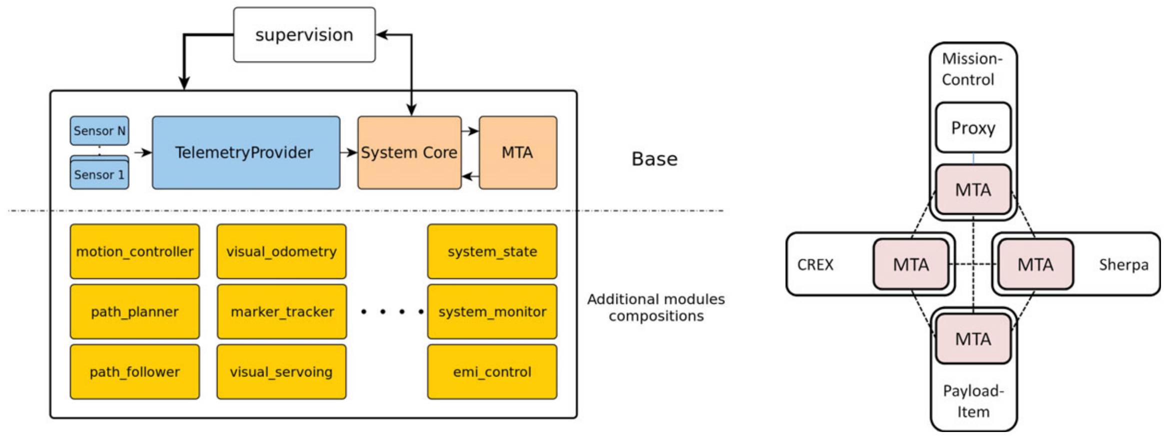
5.2. Interrobot Architecture

For the interrobot challenges, the authors build upon achievements of the multiagent community. FIPA[9] has defined a number of standards that have been applied mainly in the domain of software agents. The team or robots uses elements of the abstract architecture described by FIPA (Foundation of Intelligent Physical Agents, 2002) to build up the interrobot infrastructure.

The communication infrastructure is built with message transport services (labeled MTA in Fig. 13) to create a distributed communication network. Each robotic system, i.e., Sherpa, CREX, and payload-item, runs a local message transport service. All these local message transport services create a peer-to-peer communication network.

This communication network is created dynamically, since each message transport service announces its existence on the network using the zeroconf solution Avahi (Poettering et al., 2012). This mechanism enables the message transport services to dynamically find each other in a network—this mechanism can be also used for other components (or services) to announce their presence, thus providing a dynamic view on appearing or disappearing components. Thereby, payloads that are stacked during a mission are automatically included in the peer-to-peer communication and will be visible to other modules after powering up. Service discovery comes with an additional benefit: appearing and disappearing systems due to communication losses,

[9]Foundation of Intelligent Physical Agents.



(a) Intrarobot perspective showing the structure of base elements such as the system core which performs communication protocol validation and mediates between the system control center and supervision.

(b) Interrobot perspective illustrating the peer-to-peer network comprising subsystems and mission control.

Figure 13. Schematics of the architecture from an intra- and interrobot perspective.

power down, or similar can be detected in the network, i.e., adding an additional means to monitoring.

All participating robots in RIMRES communicate via FIPA messages (which can transport any type of content), i.e., communication between MTAs uses bit-efficient FIPA messages. Furthermore, each robot uses so-called interaction protocols to validate conversations within the MTA-based communication network. FIPA messages are described by performatives, such as *request* or *inform*, which allow validation of the flow of conversation. For all communication between robots and for communication between system-control and subsystems, a simple request-based protocol as illustrated in Figure 14 is applied—this interaction protocol is a single-request, single-response protocol.

Without an EMI-based connection, systems in RIMRES communicate via WLAN. To extend the coverage of the communication, message relaying can be applied. The presented communication stack works on top of existing meshing protocols such as BATMAN (Open-mesh, 2012), and can thus easily benefit from message relaying, only at the cost of additional bandwidth consumption.

5.3. Embedding Mission Control

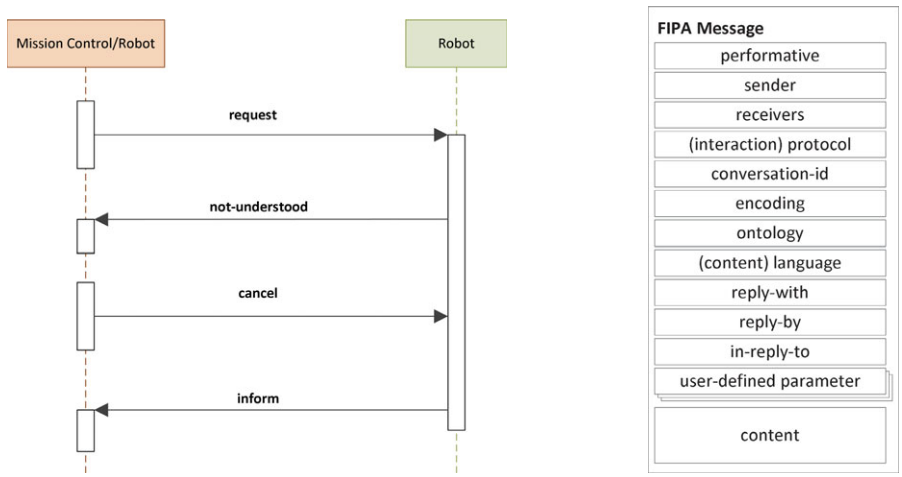
The team of robots in RIMRES does not operate in a completely autonomous fashion. The dominating operation mode is semiautonomous. Mission control outlines the major mission timeline and serves as a planning interface (cf. Figure 15)—it is possible to pause or stop a running mission and upload a new mission timeline to the system control station. Thus, while no automated planner has been embedded, the mission control uses the operator as the main planner for the initial mission and also for handling errors.

The mission control center has already been used in the project LUNARES (Cordes et al., 2010) and was adapted to cope with the requirements for reconfiguration in RIMRES. Mission control communicates via a moon-based system control with the team of robots. To establish the communication between the system control and the team of robots, a proxy has been put into place. The proxy connects the socket-based communication with the FIPA-message based communication. Similarly to LUNARES, a human-readable content-language has been defined to communicate commands, stati, and the like, e.g., to request a robot to perform an action to probe the communication: ACTION probe EXEC. The text-based content-language facilitates both debugging and implementation, and allows to easily enable communication between previously incompatible systems.

Apart from specific functionality such as querying the set of available actions from a reach robot, the definition of this content-language allows a high-level abstraction to commanding all available robots. In RIMRES, all systems, i.e., Sherpa, CREX, and payload-items, support this content-language, showing the capability of scaling the system and facilitating to embed dynamically created payloads.

5.4. Modularization and Heterogeneity

The generally adopted development approach in RIMRES (and that of Rock) outputs highly modularized software, which allows to reuse components to a large extent within the overall system. Modularization starts by designing low-level drivers that are embedded into the Orocos modules, and continues regarding robot capabilities. While there is



(a) Interaction protocol in use for robot-to-robot and mission-control-to-robot communication.

(b) Elements of a FIPA message.

Figure 14. Communication in the RIMRES project is using FIPA messages and interaction protocols as a basis for autonomous cooperative robot activities.

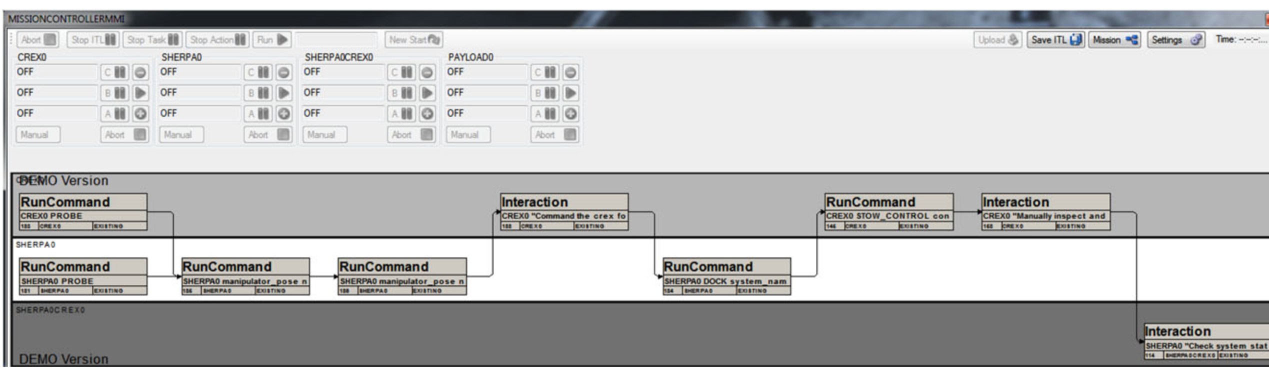


Figure 15. Interface for the mission timeline management. The illustrated timeline applies to Sherpa and CREX and their monolithic configuration.

common functionality for all systems, e.g., such as the communication infrastructure, each system has specific skills, which need to be accounted for in the software design.

To balance between generalization and specialization—and in order to keep the code base maintainable—at the level of supervision all robots use a hierarchical structuring that bundles generic functionalities and extensions to enable specific capabilities. This approach minimizes the effect of quantity and heterogeneity of the systems in RIMRES and allows to run the same basic software stack on Sherpa, CREX, and all payload-items.

Clearly, an identical configuration cannot be applied to all these systems. Configuration and fine-tuning of parameters has to be performed especially to account for lower system resources on the payload-items or for CREX. However, a common high-level software framework has been deployed on the robotic team in RIMRES, which offers a standardized interface for robot-to-robot and human-to-robot communication.

6. EXPERIMENTS

The multirobot system presented in this paper has a range of capabilities comprising locomotion, navigation, manipulation, distributed communication, and visual servoing. The latter is a central capability in the context of reconfiguration of Sherpa and CREX, and it can also be used for attaching Sherpa's manipulator arm to a payload-item. In the following description, the reconfiguration process to assemble payloads using Sherpa's manipulator is called stacking, while reconfiguring Sherpa and CREX to form a combined, mechanically connected robotic system as illustrated in Figure 7(b) is called docking. The authors selected stacking and docking as the two essential reconfiguration maneuvers involving all subsystems designed in RIMRES to verify the current system setup.

Due to the complexity of the experimental setup, a number of soft and not yet quantified parameters exist, such as effects of longer-term[10] operation of the robots. These experiments, however, investigate the following aspects: (1) feasibility and practicality, but also weaknesses of the hardware and software architecture to control a reconfigurable, multirobot system; (2) behavior of the multirobot system under typical communication load; and (3) achievable precision and accuracy—compared with the former approach—hardware platform CREX to allow for an improved visual docking procedure.

The stacking procedure allows an evaluation of the reconfiguration capabilities using the manipulator and payload-items. The docking procedure allows this evaluation for Sherpa and CREX, but it also adds a good test coverage of the software architecture of the multirobot system, since it involves elements such as (1) communication and synchronization between two robotic systems, and (2) communication between an operator and a robot. Both experiments cover (1) dynamic activation and deactivation of (software) components and devices (on a distributed subsystem), and (2) controlling the EMI. The following paragraphs present the experimental setup and results for the stacking and docking procedures.

6.1. Stacking procedure

The stacking procedure is applied to assemble payloads from payload-items. The stacking procedure is a semi-autonomous experiment in which a human operator activates each individual step in the experiment. The main purpose of this experiment is a qualitative analysis, i.e., an evaluation of reliability, speed, or energy consumption is not part of the experiment. Figure 16 illustrates the sequence of actions [also refer to the video material available (DFKI Bremen Robotics Innovation Center, 2013)].

6.1.1. Experimental Setup

The stacking experiment involves the robot Sherpa and two payload-items: one battery payload-item and one scientific (camera) payload-item. Both payload-items are already attached to the Sherpa system. The positions of the manipulator are previously taught so that Sherpa can extract payload-items from the storage bays without a requirement for visual servoing. A force torque sensor is part of the manipulator's wrist and embedded into the procedure; it allows safe pickup of payload-items and deployment of the payload in the absence of visual information.

[10]This denotes a continuous operation of the robots for a factor $n > 1$ battery cycles (CREX: approx. 1.5 h, Sherpa: approx. 3 h).

6.1.2. Experiment Procedure

Sherpa uses the manipulator to pick a scientific payload-item and detach the payload-item from Sherpa's main body; cf. Figures 16(a)–16(c). Subsequently the payload-item is stacked on top of the battery payload-item [cf. Figures 16(d) and 16(e)]. Note that the scientific payload-item is activated before attaching to the battery module to validate the capabilities of the power management system. After locking the bottom interface of the scientific payload-item and unlocking the bottom interface of the battery payload-item, the assembled payload is extracted from the storage bay. To prepare deployment of the payload, the force torque sensor needs to be recalibrated for the new weight attached to the manipulator's EMI. Recalibration is performed in a previously taught position of the manipulator, as illustrated in Figure 16(f). After switching from the payload's external power supply (the rover's battery) to its internal one (the battery payload-item), the assembled and now power-independent payload is finally detached from the manipulator.

6.1.3. Experiment Results

The stacking experiment successfully verified the capability to assemble a payload at mission time. Subsystem features that contributed to the success are (1) functionality and precision of the manipulator, in combination with (2) effective guidance of the EMI, (3) locking and unlocking capabilities of the EMI, and (4) power switching of the EMI. The active LED ring of the scientific payload-item indicates that the power switching has been successfully performed and the software stack has been started properly. The full procedure required approximately 10 min in real time.

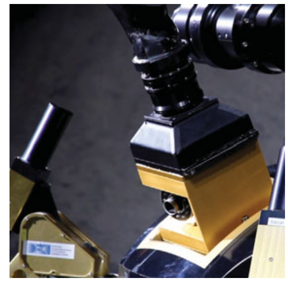
6.1.4. Experiment Discussion

The stacking experiment illustrates Sherpa's capability of assembling and deploying a subsystem. It verifies the hardware design of the manipulator and the payload-items and at the same time shows the practicality of the approach. Additionally, it demonstrates the benefits of power management, since stacking and thus any upcoming maintenance procedures for payloads can be performed in a hot-swap manner. However, for safety reasons and the direct interaction of the operator, the procedure has been executed only with low manipulator joint speeds.

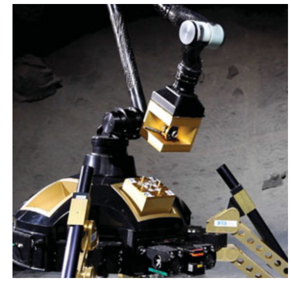
The deployment of a payload can be performed without visual means if necessary, which facilitates further automation. But the stacking procedure is currently limited to the deployment of payloads and does not allow for pick up. Embedding visual data is part of future developments, and it is necessary for picking up payload-items from the environment. It will also help to select suitable places for deployment of a payload.

(a) Initial setup with two payload-items attached to Sherpa

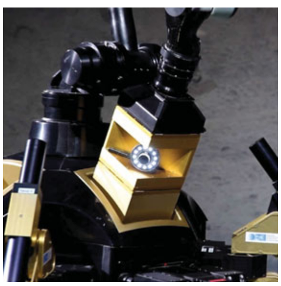
(b) Attaching the manipulator to the scientific payload-item



(c) Detaching the scientific payload-item from Sherpa's main body

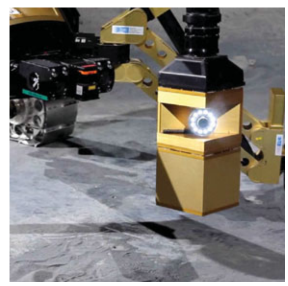


(d) Preparing to stack



(e) Detaching the fully assembled, active payload

(f) Recalibration of the force-torque sensor



(g) Approaching ground position

(h) Finally deployed payload

Figure 16. Stacking sequence for assembly and deployment of a payload consisting of one scientific payload-item and a battery payload-item.

6.2. Docking procedure

The docking procedure transforms the separate robots Sherpa and CREX into one monolithic system. Before docking between Sherpa and CREX is possible, a reference pose of CREX has to be taught. This reference position is the position that shall be reached during the visual servoing process. Within the teaching process, the Jacobian for the relative target pose is computed. Previous experiments to find optimal conditions for marker detection identified that for optimal results, a minimum distance of 15 cm to the camera needs to be maintained. Under varying lighting conditions (from completely dark over only using ambient light to ambient light with additional lighting from the EMI LEDs), most robust results are achieved using the available LED lighting in the EMI. Further, preliminary docking and teaching procedures eventually resulted in an optimal teaching position with the markers being approx. 21 cm below the actual camera. Teaching has been performed at a body height of 194 mm of CREX and by shifting CREX's pose around the reference position with offsets of 1 mm and 1°. The positions of the visual markers are extracted and the corresponding Jacobian is computed. The core algorithms for marker extraction and visual servoing have been developed by the project partner Astrium and have already been used successfully in the project LUNARES (Roehr et al., 2010).

6.2.1. Experimental Setup

The conducted experiment evaluates the performance of the main parts of the docking procedure. The steps involved in the docking procedure and part of this performance evaluation are the following:

1. Initiating the docking procedure by an operator.
2. Startup of software components by the supervision, which are needed to perform the docking.
   - Autocalibration of the electromechanical interface as preparation for the mechanical docking,
   - Switching on LEDs of Sherpa's bottom EMI,
   - Starting camera in Sherpa's bottom EMI,
   - Starting marker extraction component and visual servoing component, and
   - Creating data connections between components.
3. Executing control loop until target position (termination condition) is reached.
   - Marker detection and extraction,
   - Calculation of current offset and generation of new posture command, and
   - Direct robot-to-robot commanding and feedback.

The authors assume that the main approach of the docking procedure (CREX walking beneath Sherpa) has already

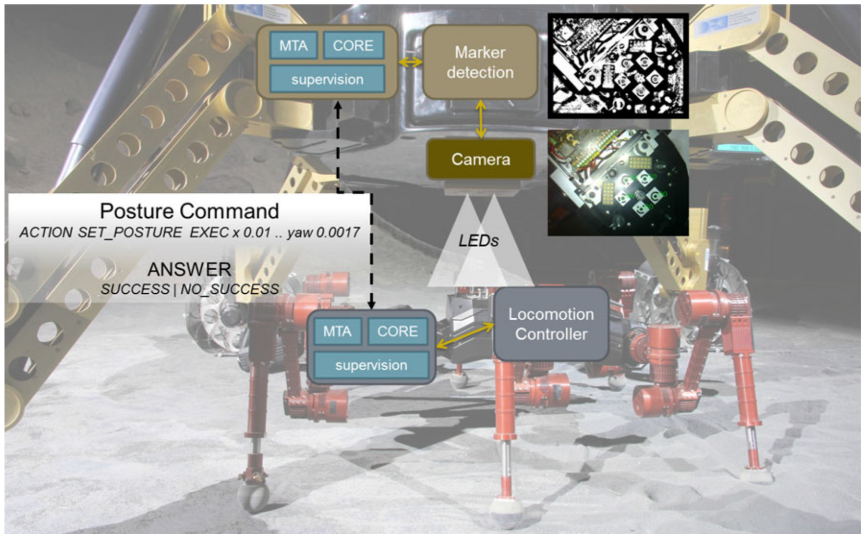


Figure 17. Process of visual servoing: The camera in Sherpa's bottom EMI identifies the visual markers placed on CREX' EMI. From the extracted positions of the markers in the video image, appropriate posture shifts of CREX are commanded, until the reference pose is reached. Subsequently, a blind docking is executed to mate the two EMIs (MTA denotes Message Transport Agent).

been performed and the final step to align CREX with the EMI at Sherpa's bottom has to be conducted. To evaluate the accuracy of this approach, the authors perform the experiments with an initial and isolated offset regarding translation along the x, y, and z axies and rotation around the z axis (yaw). For this setup, the authors further assume a planar surface where the docking is performed, and thus this experiment does not need to evaluate a compensation for roll and pitch offsets. While this assumption might not hold perfectly for a real mission, it anticipates an advanced docking procedure, where Sherpa and CREX are independently responsible for setting up or maintaining a level posture using their inertial measurement sensors. Alternatively, the visual servoing can be used to compensate for the additional degree of freedom—this approach depends on an improvement of the visual marker setup[11] and will demand finetuning of the visual servoing controller, otherwise easily leading to prolonged time required for convergence. Eventually, a perfect docking procedure is not required, but is one the EMI can compensate for.

The initial offset regarding translation and rotation is 50 mm and 5°, respectively.

All pose measurements represent CREX' telemetry data based on the inverse kinematic of the locomotion controller, i.e., no external tracking system has been used.

[11] Ideally, the set of markers is distributed in three dimensions.

6.2.2. Experiment Procedure

The docking procedure in RIMRES relies on a set of six markers located at the EMI on the back of CREX. The convergence (stopping) criterion for the visual servoing is set to a translation error of 1.2 mm and a rotation error of 0.5°. In addition, the pixel error for all visible markers needs to be below a threshold of 30 px. The target pose is defined to be reached when these criteria are met for four subsequent controller cycles.

Once the control loop is started, Sherpa computes posture commands for CREX in order to correct the pose. Sherpa does not change its pose throughout the experiment and docking procedure, while CREX is adapting its posture in order to reach the previously taught position—this position is relative to the EMI at Sherpa's bottom. The control loop works across system boundaries (cf. Figure 17) so that synchronization is needed. Using the FIPA-based message infrastructure, each communication between Sherpa and CREX involves a short interaction consisting of a request to set the posture and a response about success or failure once the action has been executed. When Sherpa receives the latest response on a posture setting request, it waits for an up-to-date marker extraction result. Marker results are timestamped using the corresponding marker image. The pose of CREX is corrected in small increments and controlled by Sherpa, which serves as the controller of the overall docking procedure. The visual servoing controller allows for a maximum translation step of 1 mm and a rotation of 1° to be performed by CREX – this corresponds to the maximum resolution for setting the posture of CREX.

The experiment has been performed with ambient lighting conditions. However, the main light sources to allow marker detection are the LEDs, which are part of the EMI at Sherpa's bottom, and as already mentioned, marker detection works best with this additional illumination. Overall, a set of 100 (+20, due to wear, see below) runs was performed, spread over multiple days and involving restarts (power-off) of both systems. Five different test setups were used for the experimental series. The setups have to be distinguished based on the pose offsets that were given for the reference position:

1. Offset in the x-direction (forward/backward of CREX) only,
2. Offset in the y-direction (sideways movements of CREX) only,
3. Offset in the z-direction (up/down movements of CREX) only,
4. Offset in yaw (rotation around the z-axis) only, and
5. Offset simultaneously and randomly applied to all four DOFs.

For each daily test setup, ten runs were conducted beforehand in order to calculate a mean pose as a reference pose. For the actual evaluation, offsets are used to shift CREX's pose away from this target pose. In the case of the single-DOF-runs, ten runs with the minimum and maximum offsets from Table VI are conducted, respectively. For the simultaneous shift in all DOFs, random offsets are generated. The random offsets are limited in order to keep CREX's EMI within the field of view of Sherpa's bottom EMI's camera. The range of the applied offsets is given in Table VI. In the runs with offsets given only in one DOF, still all four used DOFs of CREX are commendable by the visual servoing process running on Sherpa.

6.2.3. Experimental Results

This section presents the results of the experiments conducted to perform visual servoing as part of an overall docking maneuver. After more than 6 h of continuous operation of the systems in the course of the experiments presented here, a wear in CREX's position accuracy was observable. The wear expressed itself as an increase of convergence time,

Table VI. Range of random numbers for the posture offset for each experimental run.

DOF	min	max	unit
x	−50	+50	mm
y	−50	+50	mm
z	−50	+50	mm
yaw	−5	+5	deg

eventually leading to timeouts. The corresponding experiment for the set -x was therefore aborted after two subsequent timeouts and the results excluded from this presentation. The experiment was repeated the next day. However, including these failed runs, i.e., two additional runs with timeouts for -x, an overall termination rate of approx. 94% (6 out of 102 runs failed to terminate) is still reached, and the effect can still be observed looking at the duration statistics of the preceding experiment (+x offset) (cf. Table VII). Limiting the evaluation to the runs with a random initial offset, a termination rate of 95% (1 out of 20 runs failed to terminate) is reached.

The following presentation illustrates selected experiment samples as well as the accuracy and precision of the approach in Tables VII–X. Finally, this approach is compared to a previous application of the same visual servoing control algorithm in combination with different hardware systems.

6.2.4. Pose Accuracy and Precision with Offsets in Single DOF Runs

The following section presents the results for the visual servoing procedure for experiments that started with an offset on a single DOF. In the graphs in the following subsections, the error is stepwise corrected. Each of the steps corresponds to one correction step and thus one cycle, as shown in Figure 17. Thus, the plots allow to identify any delays in the startup of the process and in the actual control loop. Startup and execution times are increased due to a maximum logging level during the course of these experiments. Note further that visual servoing graphs represent single runs, since an averaging over multiple runs into one plot is not reasonable due to varying convergence times.

The statistic describes the final pixel error per marker—with an overall set of six visible markers. Duration describes the time from sending the operator command to start the visual servoing until the time of final convergence to the taught position. Precision describes which position with what kind of deviation has eventually been reached. Compared to that, accuracy describes the error of the reached position compared to the reference position, which is the taught position being expected to be reached. Even though some of the given values in the following table contain redundant information, the presentation is chosen to allow an easier understanding of both statistics. Note that while all measurements are taken in mm, the statistical values have been rounded to a tenth of a mm.

The first approximately 30–40 s with no changing errors in each plot is the startup time between starting the logging and the start of the actual visual servoing loop. Within this time, the start-up procedures described in Section 6.2.1 are executed. In the following, the authors highlight the experiments starting with an x offset, since they have been performed after multiple hours of operation time of CREX.

Table VII. Precision and accuracy results of runs with offset in the x-direction.

initial offset		+x		initial offset		−x	
number of timeouts (out of 10 runs)		2		number of timeouts (out of 10 runs)		0	
		mean	**stdev**			**mean**	**stdev**
final pixel error per marker		2.24	1.32	final pixel error per marker		3.67	1.26
duration in s		322.43	373.82	duration in s		168.6	35.49
precision: reached pose				precision: reached pose			
	ref pose				**ref pose**		
x (mm)	−5	−6.1	0.9	x (mm)	0	−0.6	1.4
y (mm)	−4	−6.9	0.7	y (mm)	5	3.7	0.6
z (mm)	194	194.6	0.6	z (mm)	195	196.2	0.4
yaw (deg)	1	0.7	0.5	yaw (deg)	0	0.0	0.0
accuracy: pose error				accuracy: pose error			
x (mm)		−1.1	0.9	x (mm)		−0.6	1.4
y (mm)		−2.9	0.7	y (mm)		−1.3	0.6
z (mm)		0.6	0.6	z (mm)		1.2	0.4
yaw (deg)		−0.3	0.5	yaw (deg)		0.0	0.0

Table VIII. Precision and accuracy results of runs with offset in the y-direction.

initial offset		+y		initial offset		−y	
number of timeouts (out of 10 runs)		0		number of timeouts (out of 10 runs)		0	
		mean	**stdev**			**mean**	**stdev**
final pixel error per marker		2.96	1.39	final pixel error per marker		2.15	1.58
duration in s		127.63	125.43	duration in s		149.4	82.53
precision: reached pose				precision: reached pose			
	ref pose				**ref pose**		
x (mm)	−5	−6.6	0.8	x (mm)	−5	−6.4	0.7
y (mm)	−4	−2.8	0.7	y (mm)	−4	−4.7	1.6
z (mm)	194	194.9	0.3	z (mm)	194	194.7	0.8
yaw (deg)	1	0.9	0.3	yaw (deg)	1	0.5	0.5
accuracy: pose error				accuracy: pose error			
x (mm)		−1.6	0.8	x (mm)		−1.4	0.7
y (mm)		1.2	0.7	y (mm)		−0.7	1.6
z (mm)		0.9	0.3	z (mm)		0.7	0.8
yaw (deg)		−0.1	0.3	yaw (deg)		−0.5	0.5

The visual servoing procedure timed out for two times during the set of ten experiments for the positive offset—also shown by the high mean and standard deviation of the duration time in Table VII compared to other experiments. The experiment performed directly afterwards for the negative offset suffered from increasingly higher convergence times and—as already mentioned in the introduction—has been aborted with the assumption of increased system wear.

The results show that the duration of experiments with a positive x offset is significantly higher, though this is due to significant outliers. Only terminated runs are part of the precision and accuracy computation. The results show that the visual servoing procedure is still able to reach good precision; e.g., see the experiment sample in Figure 18. The authors assume the outliers to be an effect of system wear after multiple hours of operation, since the continuation of

Table IX. Precision and accuracy results of runs with offset in the z-direction.

initial offset		+z	
number of timeouts (out of 10 runs)		0	
		mean	**stdev**
final pixel error per marker		2.44	0.9
duration in s		128.67	46.76
precision: reached pose			
	ref pose		
x (mm)	−5	−5.9	1.0
y (mm)	−4	−1.9	0.3
z (mm)	194	194.4	0.5
yaw (deg)	1	1.0	0.0
accuracy: pose error			
x (mm)		−0.9	1.0
y (mm)		2.1	0.3
z (mm)		0.4	0.5
yaw (deg)		0.0	0.0

initial offset		−z	
number of timeouts (out of 10 runs)		1	
		mean	**stdev**
final pixel error per marker		2.43	1.31
duration in s		192.1	231.13
precision: reached pose			
	ref pose		
x (mm)	−5	−6.3	0.6
y (mm)	−4	−2.3	0.5
z (mm)	194	195.0	0.0
yaw (deg)	1	0.9	0.3
accuracy: pose error			
x (mm)		−1.3	0.6
y (mm)		1.7	0.5
z (mm)		1.0	0.0
yaw (deg)		−0.1	0.3

Table X. Precision and accuracy results of runs with offset in the yaw-direction.

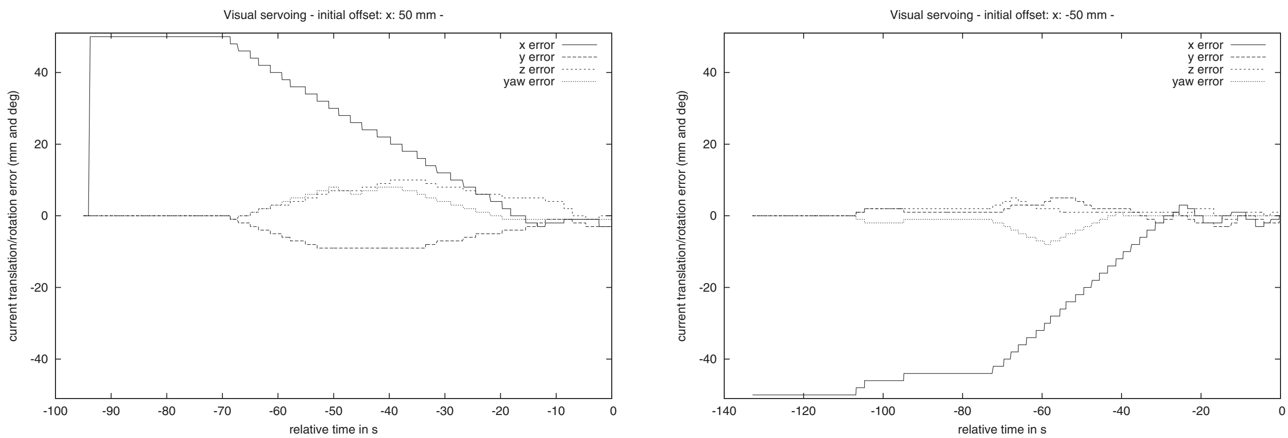
initial offset		+yaw	
number of timeouts (out of 10 runs)		0	
		mean	**stdev**
final pixel error per marker		2.99	1.49
duration in s		51.9	22.34
precision: reached pose			
	ref pose		
x (mm)	0	−0.9	0.8
y (mm)	5	2.9	0.7
z (mm)	195	195.8	0.4
yaw (deg)	0	0.0	0.0
accuracy: pose error			
x (mm)		−0.9	0.8
y (mm)		−2.1	0.7
z (mm)		0.8	0.4
yaw (deg)		0.0	0.0

initial offset		−yaw	
number of timeouts (out of 10 runs)		0	
		mean	**stdev**
final pixel error per marker		2.47	1.01
duration in s		54.6	20.91
precision: reached pose			
	ref pose		
x (mm)	0	−0.7	1.1
y (mm)	5	2.5	0.8
z (mm)	195	195.7	0.6
yaw (deg)	0	0.0	0.0
accuracy: pose error			
x (mm)		−0.7	1.1
y (mm)		−2.5	0.8
z (mm)		0.7	0.6
yaw (deg)		0.0	0.0

the experiments with a fresh system (after cold start the next day) did not show such a timeout, as also observed in earlier runs.

Table XI shows the final results concerning pose accuracy and pose precision for moving in all four DOFs with random pose offsets. This can be considered as the standard case during system operation.

In Figure 19, two experiment samples of the temporal course of CREX's pose are presented. With the beginning of the actual visual servoing control loop, it is observable that the position errors in x and y are basically continuously decreasing. In both displayed examples, the yaw error is increasing in the first place, but finally gets corrected. Noticeable is the fact that the z-DOF seems to be only correctable



(a) Positive x-offset (b) Negative x-offset

Figure 18. Example trends for runs with offset in the x-direction.

Table XI. Final results of docking experiment.

initial offset		random	
number of timeouts (out of 20 runs)		1	
		mean	**stdev**
final pixel error per marker		1.78	1.11
duration in s		161.72	80.69
precision: reached pose			
	ref pose		
x (mm)	0	0.0	0.9
y (mm)	5	2.0	1.1
z (mm)	195	195.6	0.6
yaw (deg)	0	0.0	0.2
accuracy: pose error			
x (mm)		0.0	0.9
y (mm)		−3.0	1.1
z (mm)		0.6	0.6
yaw (deg)		0.0	0.2

when x- and y-errors are already close to being corrected, e.g., in Figure 19(a) the z-error decreases only after x- and y-errors are significantly reduced, while in Figure 19(b) the error increases until the x-error and the y-error are significantly reduced. However, for concluding a direct relation in the described way, a broader database than that collected in this experimental series is needed.

The experiment starting with random offsets on all four DOFs confirms previously received results from the experiments of the single DOF. The typical (mean) convergence time is just lower than 3 min while the final accuracy achieved shows a very low rotation error, but a translation error of multiple millimeters. The to be expected translation offset of about 4 mm can be compensated by the mechanical tolerances in the EMI design using the guidance pins. The guidance pins allow either compensation of an offset of 6 mm or a rotation of 7° [for details, see Wenzel et al. (2011)]. The error in z is not of great significance and can also be easily compensated for by the final docking process using a force torque measurement by CREX.[12] A slight bias regarding a higher pose error along y compared with x can be seen and leads to the assumption of a systematic error introduced by the hardware. Further experiments on the hardware accuracy and precision will be performed with CREX to investigate this matter.

Overall this evaluation shows that the accuracy and precision of the visual servoing are high enough for the outlined docking procedure of CREX and Sherpa.

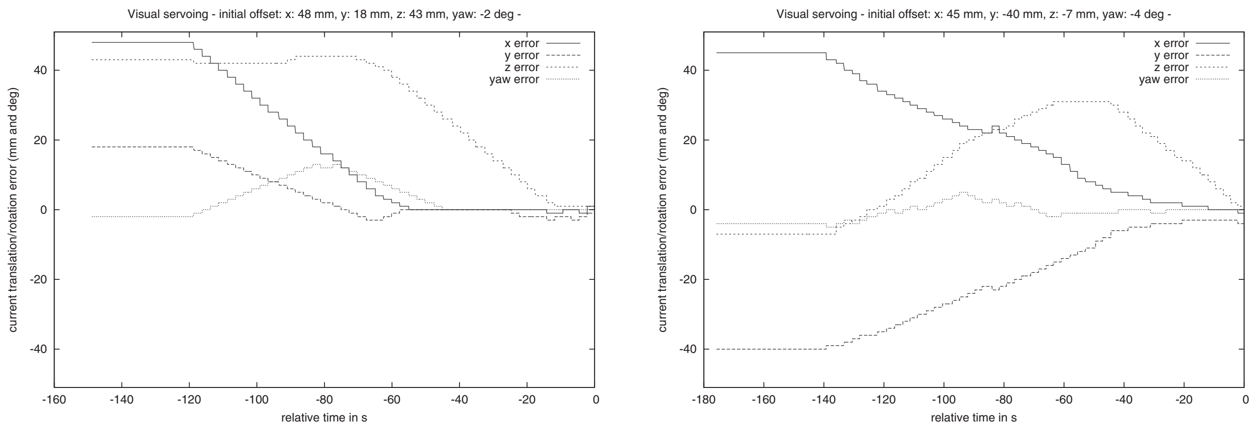
6.2.5. Comparison with Former Approach

In a former approach, it was already possible to demonstrate the feasibility of visual servoing as a means for docking preparation. In the project LUNARES, a mechanical connection between a wheeled rover and an eight-legged scout robot was successfully established after autonomous positioning via visual servoing. Figure 20 illustrates a typical scene from the docking process.

In the approach, the scout had four rear-facing visual markers that were identified by the camera system of the rover. The algorithm calculated movement commands for the scout robot in order to bring the scout into a previously defined reference pose. For the docking, three DOFs were commanded: forward/backward (x direction) and sideways movements (y direction) and the heading (yaw angle) of the robot.

As already presented in Cordes et al. (2010), the standard deviations achieved in the docking procedure were

[12]The flange of each of CREX's legs is a 6 DOF force-torque sensor.



(a) Trend for pose corrections (x, y, z, yaw) for initial offset from taught pose of (48 mm, 18 mm, 43 mm, −2°)

(b) Trend for pose corrections (x, y, z, yaw) for initial offset from taught pose of (45 mm, −40 mm, −7 mm, −4°)

Figure 19. Examples for the trend of pose corrections in experiments with a (bounded) random offset for dimension x, y, z, and yaw.

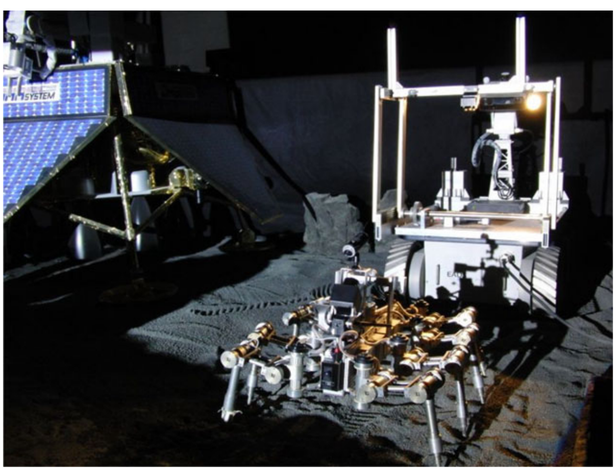


Figure 20. Visual servoing as preparation for docking in LUNARES. The rover identifies the rear-facing visual markers and generates movement commands for the scout. Once the goal position is reached, the docking lever is lowered, and by shifting the body backward and down, the scout places its bail in the docking adapter's hook. The connection is purely mechanical.

within the mechanical tolerances of the docking mechanism and the play in the commanded robot's joints. Table XII displays the results achieved in the former approach.

The approach in LUNARES did not require a high-precision approach, since the connection was only mechanical and no electrical contact needed to be established. In RIMRES, however, the EMI and its design (including contact pins for the electrical-connection) require a much more precise approach, which is mainly feasible only due to the improved design of CREX.

Table XII. Results of docking in the LUNARES project.

initial offset	random	
	mean	**stdev**
duration (s)	184	35.5
pose accuracy/final pose error		
x (mm)		9
y (mm)		4
z (mm)		12
yaw (deg)		0.7

Nevertheless, it has to be noted again that the measurement in LUNARES were taken by a tracking system and using the walking mode of the robot, while the experiment performed for RIMRES relies on internal measurements of CREX and needs to be extended with an additional evaluation of the pose precision of CREX posture control. Still, the results are significant, since the final docking procedure relies on the combination of internal measurement and the marker detection.

6.2.6. Experiment Discussion

Along with the stacking experiment, the docking experiment has been setup to investigate (1) feasibility and practicality, but also weaknesses of the hardware and software architecture to control a reconfigurable, multirobot system; (2) the behavior of the multirobot system under typical communication load; and (3) achievable precision and accuracy—compared with the former approach—of the hardware platform CREX to allow for an improved visual docking procedure. The docking experiment allows a direct extraction of the results regarding precision and accuracy of the approach. Furthermore, it validates the infrastructure regarding tight cooperation of two mobile systems Sherpa and CREX using a master-slave approach in a closed-loop. This approach requires only a set of visible markers for teaching and visual servoing of the slave system, and it has to support setting the posture from the master system. Hence, due to the simple and standard command interface for robot-to-robot communication, this approach could be easily applied using a different slave system.

A major weakness of the current architecture can be seen in the complex setup compared to a specialized solution. Multiple layers of abstraction and the distribution of the multirobot systems make debugging harder. Especially, since a variety of tuning parameters exist and details of the underlying software stack such as Orocos can become significant, e.g., during the set of experiments and related to the second aspect under investigation, the authors observed that components on a relatively slow system did not stop properly as expected and commanded by the supervision. This was due to a fixed timeout setting within Orocos RTT, which was only relevant in this context, e.g., the Gumstix that was responsible in this context to manage the EMI camera.

In addition, embedding standards such as FIPA seem to add to this complexity without any initial benefit. However, the authors can confirm the practicality of this approach, especially regarding system cooperation, conversation monitoring, and a dynamic setup of robot communication. With respect to the long-term goal of autonomous cooperative systems, the authors expect the application of standards such as FIPA to be of even higher significance. Meanwhile, the authors have already shown the feasibility of the overall approach by applying it to a multirobot reconfiguration.

The complexity of the current approach is a drawback, but it is needed to allow for a consistent, generalized approach that scales to a multirobot system such as presented here. The authors believe that the approach is effective and facilitates transfer to other robotic systems beyond the scope of this project, and thus serves a long-term development plan.

7. LESSONS LEARNED

The project RIMRES allowed to expand the authors' knowledge not only regarding technology, but also regarding the overall development approach. The following sections highlight the lessons learned for each of the systems developed as part of a reconfigurable multirobot system.

7.1. EMI Design

The validation of selected requirements for the EMI was successfully performed in isolated test setups, e.g., such as the mechanical stability. Still, applying the EMI in the real systems showed that the guidance pins of the EMI were initially too short to provide actual guidance, i.e., they had the same height as the central pin. For visual servoing, a marker pattern was imprinted using a laser to the central pin's head. The final result was a small and too light black imprint on a highly reflective material. The central pin was the only marker outside of the payload-items' top face plane, and for good visual servoing performance, markers should be distributed in three dimensions. Thus marker detection required special tuning of the parameters to include the top marker into the visual servoing process. Meanwhile, limiting the DOF for docking to 90° orientation steps was sufficient for all tested applications. Mechanical and electric connections with EMIs in RIMRES are sufficiently stable and reliable, but they cannot replace a dedicated monolithic design. The connection between Sherpa and CREX, however, showed it to be sensitive to play between the connecting interfaces, i.e., while the interface is capable of creating a persistent mechanical and electrical connection, any available play can lead to an undesired shaking effect during traveling when CREX is attached to Sherpa.

Establishing serial LOC between all EMIs was a useful backup in situations when a wireless connection did not work fully reliably or had not been established yet.[13] In addition to power management, LOC allows to increase the system's resilience in particular use cases, e.g., communication and controlling EMIs.

Since the EMIs build a topology of the overall system, inferring the structure and verifying the identity of a system can be easily performed, e.g., when connecting the manipulator to a payload-item; due to LOC, the high-level software stack does not need to be active on the payload-item in order to identify it.

The EMI is a seemingly simple device, and the complexity of the EMI can be easily underestimated compared to systems like Sherpa or CREX. In RIMRES, the central importance of the interface for a reconfigurable system and the set of detailed requirements for the EMI design led to an outstretched development involving multiple iterations.

7.2. Payload-items

All payload-items come with the same set of infrastructure components in order to provide processing power and robust interfacing capabilities. This serves the purpose of redundancy, but in future designs the ratio between infrastructure and actual available payload volume has to be improved. Furthermore, scientific instruments such as REIPOS or the aspired integration of the mole module demand a special design, so that the constraint of a 15 cm^2 interface can be too limiting for some real missions. Especially for developing multiple systems of the same type—such as the payload-items—it has been beneficial to maintain a database to keep

[13] Payload-items require a few minutes until they are fully functional and embedded into the communication infrastructure.

track of (sub)components, software builds, changes, and issues.

7.3. Rover Design

Sherpa is the most complex system developed in RIMRES considering the integration density of EMIs, the manipulator, and the active suspension system.

Regarding the development of the manipulator, an exploration of multiple sensors and setups had to be performed to achieve a sufficiently good precision of the joint control and allow for blind stacking procedures. Though the required precision had eventually been reached, the current setup still required an initial calibration procedure after startup of the system to guarantee precise operation. Meanwhile, the manipulator proved its practicality regarding manipulation and inspection—since the EMI contains a camera, the manipulator can be used for visual inspection. It also showed its suitability for locomotion support, i.e., as a fifth leg, and it confirms the benefits of evolutionary optimization; most likely designers would not have considered a bent first link of the manipulator otherwise.

The active suspension system of Sherpa has been designed with six DOF per leg. In most cases, the actual application was limited to four DOF, so that the initial design can be considered overly complex. The actual benefit of the additional and currently locked DOF can be questioned and needs further evaluation, also with respect to redundant functionality since flexible wheels are used. Thus, further experiments have to be performed to evaluate this matter.

Sherpa is the key player in the multirobot system and therefore remains a bottleneck and single point of failure for a mission. In the current setup, it is the only mobile system that can manipulate payload-items. Hence, Sherpa has to be highly reliable. Alternatively, the deployment of additional robots with the capability for manipulation or at least detaching and attaching payload-items should be considered.

7.4. CREX

The main features of CREX are inherited from SpaceClimber and thus the lessons learned are focused on new additions to the system, e.g., up to now a depleting battery posed a significant risk for a mission since CREX and SpaceClimber require energy to hold a position other than lying on the ground. In the RIMRES, however, Sherpa's manipulator can be used to attach to CREX and recharge using the facilities of the power management, thus minimizing this operational risk. CREX can be manually operated and an operator initiates the docking process after guiding CREX underneath Sherpa. Nevertheless, the existing and to be expected additional delay between control commands and visual feedback makes the remote operation unsafe, and a higher degree of automation is desirable. Future procedures should therefore rely on visual information and onboard processing on either CREX or Sherpa.

7.5. Software Architecture

Developing the software architecture for this multirobot system has been a challenge, especially due to the quantity and heterogeneity of the systems involved. Establishing dedicated development procedures, multiple layers of abstraction, and a model-driven design have been essential contributors to a finally successful application. Having a large quantity of subsystems means that even small (systematic) errors—regarding hardware or software—can lead to a severe setback. Main efforts went into enabling the system's basic functionality and making them work reliably, including reconfiguration. Any system development in that context—even if it is not aiming at a high TRL (technology readiness level)—has to go thoroughly through the steps of (hardware) component selection, verification, and integration testing. In addition, detailed debugging and message tracing capabilities across systems need to be available. To limit the impact of heterogeneity, it should be actively controlled and reduced in the system design phase by fostering the reuse of hardware components, e.g., using a limited set of camera types.

Additional layers of abstraction can add constraints on accessing low-level components, or they might reduce functionality exposed to higher layers. Still, access to low-level components should be generally maintained for remote operation. Furthermore, the architecture applied to the heterogeneous multirobot system had to cope with limited computing resources. The payload-items required special performance optimization, e.g., allowing image processing by activation of the Digital Signal Processor (DSP) and minimizing the overhead of infrastructure components. This shows that performance optimization is well needed to achieve a broadly applicable software stack.

7.6. Mission Control

The configuration of mission control in RIMRES is static and requires the previous configuration of all known individual systems and possible combinations. In addition, the multilayered control design requires information replication. While the multilayered approach allows better generalization, it also introduces a source of error if workflows for configuration generation are not fully automated. Furthermore, the current level of autonomy of the multirobot system is limited and restricted to (semi)autonomous operation, such as docking and stacking. For full exploitation of a reconfigurable multirobot system, a higher degree of autonomy is required. Similarly, the handling of errors and deviations from original plan sequences should be an integral part of manual and autonomous mission control and thus the center of further improvements.

8. CONCLUSION AND FUTURE WORK

This paper presents results of the project RIMRES. Within this project, a novel approach of tightly cooperating heterogeneous robotic systems is pursued. Reconfiguration of the subsystems themselves and the overall system by combining the individual subsystems is considered in the design phase and enables a wide range of actions to be taken in cases of failure.

A central role in the design of the reconfigurable system takes the electromechanical interface (EMI) that is common to all subsystems. The EMI is a standardized interface allowing for a modular hardware design of the team of robots. The wheeled rover Sherpa not only shows high adaptability to terrains that should be covered in planetary exploration, but it also plays a central role in the reconfiguration of the robotic team: in order to reconfigure other robots, it uses its manipulator and stacks payload-items, which provide additional functionalities for the mobile subsystems or can be combined with other items to form independent (immobile) surface-deployable subsystems.

Throughout this paper, the authors have shown that reconfigurability and modularity can be accounted for at different levels of the overall system. While reconfiguration capabilities already exist in systems that have not even been designed for it, RIMRES intentionally makes it part of the design and shows a great range of flexibility toward new applications. Regarding an increase of adaptability by physical reconfiguration, RIMRES opens new directions for (field) research by providing a platform for studying complex physical reconfigurations. Most importantly, though, this project shows that despite the modularity of the developed systems, they are capable of performing complex maneuvers such as stacking and docking. Both maneuvers have been presented as part of an experimental evaluation. While stacking has been used for a qualitative evaluation of the hardware and verification of low-level software functionality, the high-level software stack was evaluated by the docking maneuver. The authors were able to show the adaptability of the multirobot system using these basic functionalities. Furthermore, the authors assume that increasing the autonomy of the multirobot system, e.g., by using an improved organization model, will allow a more sophisticated exploitation of the hardware capabilities.

Clearly, the modularization and flexibility of the reconfigurable multirobot system in RIMRES come at a cost. Using EMIs introduces a management overhead, and the complexity of this new piece of hardware introduces a point of failure, especially having to maintain two interface types: male and female. This demands an equally strong focus on the design, verification, and integration testing of this device compared to the more complex systems. Generalizing from this observation, the modularized design facilitates the exploitation of existing redundancies in the multirobot system. However, the design comes with increased complexity and probability of error. Therefore, the capability for reconfiguration cannot replace existing safety measures, but it has to be seen as an add-on.

Evaluation of the general EMI concept and resulting reconfiguration capabilities provided good evidence regarding the feasibility and practicality of the approach toward supporting reconfigurable systems. The detailed design elements have to be improved further and a thorough evaluation of the efficiency gain has to be performed. A future improvement is the reduction to a homogeneous EMI design, and the authors expect a benefit regarding design and maintenance efforts when using a single-gender interface. Generally, improving the individual systems is the next step to making the overall multirobot system more capable and reliable, e.g., the wheels of the rover are planned to be extended to actively adapt their stiffness in order to react to changes in the environment, and in a further step to be able to act as a sensor for the characterization of soil properties. A final integration of a relative positioning system, which already started in RIMRES, will be another research direction to pursue.

Future applications similar to RIMRES can benefit from flexibility in the system design for proactive actions as well as for reactive actions. Improving the overall outcome of a multirobot mission remains the main intention and will be achieved by a reduction of the operational risk as well as increasing the efficiency of the overall team of robots. Further development will target system modeling for the intelligent and automated use of the additional degrees of operational freedom and investigate more thoroughly how to exploit the systems' capabilities in order to balance efficiency and operational risk for multirobot activities.

ACKNOWLEDGMENTS

The authors would like to thank the RIMRES team members, partners, and supporting staff at the DFKI Bremen Robotics Innovation Center. The project RIMRES was funded by the German Space Agency (DLR, Grant No. 50RA0904) with federal funds of the Federal Ministry of Economics and Technology (BMWi) in accordance with the parliamentary resolution of the German Parliament. RIMRES was a cooperation between the DFKI Bremen–Robotics Innovation Center and ZARM–Center of Applied Space Technology and Microgravity. Further partners were DLR-RY, EADS Astrium, and OHB System.

REFERENCES

Aoki, T., Murayama, Y., & Hirose, S. (2011). Mechanical design of three-wheeled lunar rover "Tri-Star IV". In Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA '11) (pp. 2198–2203), Shanghai, China.

Arnold, J. R. (1979). Ice in the lunar polar regions. Journal of Geophysical Research, 84(B10), 5659–5668.

Bares, J., & Wettergreen, D. (1999). Dante II: Technical description, results and lessons learned. International Journal of Robotics Research, 18(7), 621–649.

Bartlett, P., Wettergreen, D., & Whittaker, W. R. L. (2008). Design of the Scarab rover for mobility and drilling in the lunar cold traps. In Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS '08), Hollywood, LA.

Bartsch, S., Birnschein, T., Roemmermann, M., Hilljegerdes, J., Kuehn, D., & Kirchner, F. (2012). Development of the six-legged walking and climbing robot SpaceClimber. Journal of Field Robotics, 29(3), 506–532.

Bellifemine, F., Poggi, A., & Rimassa, G. (1999). JADE—A FIPA-compliant agent framework. In Proceedings of the 4th International Conference on the Practical Application of Agents and Multi-Agent Systems (PAAM '99) (pp. 97–108). The Practical Application Company Ltd.

Bindel, D., & Bruns, R. (2010). REIPOS—Relative Interferometric Position Sensor. In Proceedings of the 61st International Astronautical Congress (IAC '10), number IAC-10-A3.2C.5, Prague, Czech Republic.

Bretl, T. (2006). Motion planning of multi-limbed robots subject to equilibrium constraints: The free-climbing robot problem. International Journal of Robotics Research, 25(4), 317–342.

Bruyninckx, H., Soetens, P., & Koninckx, B. (2003). The real-time motion control core of the Orocos project. In Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA '03) (pp. 2766–2771), Taipei, Taiwan.

Castano, A., Behar, A., & Will, P. (2002). The Conro modules for reconfigurable robots. IEEE/ASME Transactions on Mechatronics, 7(4), 403–409.

CCSDS/AIAA Inc. (2008). Reference architecture for space data systems. Retrieved June 20, 2013, from http://public.ccsds.org/publications/archive/311x0m1.pdf.

CCSDS/AIAA Inc. (2012). The Consultative Committee for Space Data Systems—The official web site. Retrieved December 6, 2012, from http://www.ccsds.org.

Chen, B., Cheng, H. H., & Palen, J. (2006). Mobile-C: A mobile agent platform for mobile C-C++ agents. Software: Practice and Experience (Software Pract Ex), 36(15), 1711–1733.

Cheshire, S. (2011). Multicast DNS. Retrieved October 4, 2011, from http://www.multicastdns.org.

Cheshire, S., & Krochmal, M. (2011). IETF internet draft—DNS-based service discovery. Retrieved October 4, 2011, from http://tools.ietf.org/pdf/draft-cheshire-dnsext-dns-sd-10.txt.

Colaprete, A., Schultz, P., Heldmann, J., Wooden, D., Shirley, M., Ennico, K. et al. (2010). Detection of water in the LCROSS ejecta plume. Science, 330(6003), 463–468.

Cordes, F., Ahrns, I., Bartsch, S., Birnschein, T., Dettmann, A., Estable, S. et al. (2010). LUNARES: Lunar crater exploration with heterogeneous multi robot systems. Intelligent Service Robotics, 4(1), 61–89.

Cordes, F., Dettmann, A., & Kirchner, F. (2011). Locomotion mode control for a hybrid wheeled-leg planetary rover. In Proceedings of the 2011 IEEE International Conference on Robotics and Biomimetics (Robio '11), Phuket, Thailand.

Cordes, F., & Kirchner, F. (2010). Heterogeneous robotic teams for exploration of steep crater environments. In Workshop Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA '10), Anchorage, AK.

Dettmann, A., Wang, Z., Wenzel, W., Cordes, F., & Kirchner, F. (2011). Heterogeneous modules with a homogeneous electromechanical interface in multi-module systems for space exploration. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA '11) (pp. 1964–1969), Shanghai, China.

DFKI Bremen Robotics Innovation Center (2011). The Robot Construction Kit. Retrieved October 4, 2011, from http://www.rock-robotics.org.

DFKI Bremen Robotics Innovation Center (2013). Sherpa: Stacking camera payload. Retrieved February 3 2013, from http://youtu.be/ikUM8nUBMjI.

Dignum, V., editor (2009). Handbook of research on multi-agent systems: Semantics and dynamics of organizational models. IGI Global.

Dunin-Keplicz, B. M., & Verbrugge, R. (2010). Teamwork in multi-agent systems: A formal approach, 1st edition. Wiley.

Evans, J. S. (1991). Strategic flexibility for high technology manoeuvers: A conceptual framework. Journal of Management Studies, 28, 69–89.

Ferber, J., & Gutknecht, O. (1998). Aalaadin: A meta-model for the analysis and design of organizations in multi-agent systems. In Proceedings of the 3rd International Conference on Multi-Agent Systems (ICMAS '98) (pp. 128–135), Washington, DC.

Ferrarini, L., & Carpanzano, E. (1999). Reference models for the supervision and control of advanced industrial manipulators. In Proceedings of the 1999 American Control Conference (ACC '99) (vol. 4, pp. 2440–2444), San Diego, CA.

Foundation of Intelligent Physical Agents (2002). FIPA abstract architecture specification. Retrieved October 4, 2011, from http://www.fipa.org/specs/fipa00001/SC00001L.pdf.

IETF Zeroconf Working Group (2011). Zero configuration networking (Zeroconf). Retrieved October 4, 2011, from http://zeroconf.org.

ISECG—International Space Exploration Coordination Group (2011). The global exploration roadmap. Retrieved July 20, 2013, from http://www.nasa.gov/pdf/591066main_GER_2011_for_release.pdf.

JAXA (2009). HAYABUSA: Resumption of return cruise by combining two ion engines. Retrieved November 26, 2012, from http://www.isas.jaxa.jp/e/topics/2009/1119.shtml.

Joyeux, S., Kirchner, F., & Lacroix, S. (2010). Managing plans: Integrating deliberation and reactive execution schemes. Robotics and Autonomous Systems, 58(9), 1057–1066.

Kennedy, B., Agazarian, H., Cheng, Y., Garrett, M., Hickey, G., Huntsberger, T. et al. (2001). LEMUR: Legged excursion mechanical utility rover. Autonomous Robots, 11, 201–205.

Kroemer, O., Beermann, D., Cordes, F., Lange, C., Littau, B., Rosta, R. et al. (2011). Adaptive flexible wheels for planetary exploration. In Proceedings ot the 62nd International Astronautical Congress (IAC '11), Cape Town, South Africa.

Kurokawa, H., Tomita, K., Kamimura, A., Kokaji, S., Hasuo, T., & Murata, S. (2007). Self-reconfigurable modular robot M-TRAN: distributed control and communication. In Proceedings of the 1st International Conference on Robot Communication and Coordination (RoboComm '07) (pp. 211–217), Athens, Greece.

Lyell, M., Webb, A., Nanda, J., & Chen, W. (2009). Human-autonomous system interaction framework to support astronaut-multi-agent system interactions. In Proceedings of the 47 AIAA Aerospace Sciences Meeting, number AIAA-2009-428, Orlando, FL.

Manz, M., Hilljegerdes, J., Dettmann, A., & Kirchner, F. (2012). Development of a lightweight manipulator-arm with selective laser sintering parts combined with fibre-reinforced plastic structures. In Proceedings of the 11th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS '12), Turin, Italy.

Metta, G., Fitzpatrick, P., & Natale, L. (2006). YARP: Yet Another Robot Platform. International Journal of Advanced Robotics Systems, Special Issue on Software Development and Integration in Robotics, 3(1).

Microsoft (2012). Microsoft Robotics Studio Developer. Retrieved October 5, 2012, from http://www.microsoft.com/robotics.

Mosher, T. J., & Lucey, P. (2006). Polar night: A lunar volatiles expedition. Acta Astronautica, 56, 585–592.

Nilsson, M. (2002). Heavy-duty connectors for self-reconfiguring robots. In Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA '02) (vol. 4), Washington, DC.

Open-mesh (2012). B.A.T.M.A.N. (better approach to mobile ad-hoc networking). Retrieved December 4, 2012, from http://www.open-mesh.org/projects/open-mesh/wiki.

Ostergaard, E., Kassow, K., Beck, R., & Lund, H. (2006). Design of the ATRON lattice-based self-reconfigurable robot. Autonomous Robots, 21(2), 165–183.

Poettering, L., Lloyd, T., & Estienne, S. (2012). Avahi. Retrieved October 4, 2012, from http://www.avahi.org.

Poslad, S., Buckle, P., & Hadingham, R. (2000). The FIPA-OS agent platform: Open source for open standards. In Proceedings of the 5th International Conference on the Practical Application of Intelligent Agent and Multi Agent Technology (PAAM '00) (pp. 355–368), Blackpool, England. The Practical Application Company Ltd.

Roehr, T. M., Cordes, F., Kirchner, F., & Ahrns, I. (2010). Cooperative docking procedures for a lunar mission. In Proceedings of the Joint Conference of the 41st International Symposium on Robotics and the 6th German Conference on Robotics) (ISR/ROBOTIK '10) (pp. 564–571), Munich, Germany.

Roemmerman, M., Kuehn, D., & Kirchner, F. (2009). Robot design for space missions using evolutionary computation. In Proceedings of the 2009 IEEE Congress on Evolutionary Computation (CEC '09) (pp. 2098–2105), Trondheim, Norway.

Snyder, R. D., Douglas, D., & Mackenzie, C. (2004). Robustness infrastructure for multi-agent systems. In Proceedings of the 1st Open Cougaar Conference on Applications, Survivability Infrastructure and Architecture of the Cougaar Agent Software (OCC '04), New York.

Soetens, P. (2012). RTT: Real-Time Toolkit. Retrieved December 5, 2012, from http://www.orocos.org/rtt.

Spenneberg, D., & Kirchner, F. (2007). The bio-inspired SCORPION robot: Design, control, & lessons learned, Climbing, & walking robots, Towards new applications (pp. 197–218). I-Tech Education and Publishing, Wien, Austria.

Spenneberg, D., Mccullough, K., & Kirchner, F. (2004). Stability of walking in a multilegged robot suffering leg loss. In Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA '04) (p. 2004), New Orleans, LA.

Spenneberg, D., Strack, A., Hilljegerdes, J., Zschenker, H., Albrecht, M., Backhaus, T. et al. (2005). Aramies: A four-legged climbing and walking robot. In Proceedings of the 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS '05), Munich, Germany.

Sproewitz, A., Asadpour, M., Bourquin, Y., & Ijspeert, A. (2008). An active connection mechanism for modular self-reconfigurable robotic systems based on physical latching. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA '08) (pp. 3508–3513), Pasadena, CA.

Suh, J., Homans, S., & Yim, M. (2002). Telecubes: Mechanical design of a module for self-reconfigurable robotics. In Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA '02) (vol. 4, pp. 4095–4101), Washington, DC.

Utz, H., Sablatnog, S., Enderle, S., & Kraetzschmar, G. (2002). Miro—Middleware for mobile robot applications. IEEE Transactions on Robotics and Automation, 18(4), 493–497.

Visentin, G. (2007). Autonomy in ESA planetary robotics missions. PPARC KITE Club/Sensors KTN. Workshop on Autonomous Sensors. Retrieved March 8, 2013, from http://www.scitech.ac.uk/Resources/PDF/Gianfranco Visentin.pdf.

Wang, Z., Cordes, F., Dettmann, A., & Szczuka, R. (2011). Evaluation of a power management system for heterogenous modules in a self-reconfigurable multi-module system. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '11), San Francisco, CA.

Wenzel, W., Cordes, F., Dettmann, A., & Wang, Z. (2011). Evaluation of a dust-resistant docking mechanism for surface exploration robots. In Proceedings of the 15th International

Conference on Advanced Robotics (ICAR ’11), Tallinn, Estonia.

Wettergreen, D., Jonak, D., Kohanbash, D., Moreland, S., Spiker, S., & Teza, J. (2009). Field experiments in mobility and navigation with a lunar rover prototype. In Proceedings of the 7th International Conference on Field and Service Robotics (FSR ’09), Cambridge, MA.

Wheeler, D., Chavez-Clemente, D., & SunSpiral, V. (2010). FootSpring: A compliance model for the ATHLETE family of robots. In Proceedings of the 10th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS ’10) (pp. 644–651), Sapporo, Japan.

Wilcox, B., Litwin, T., Biesiadecki, J., Matthews, J., Heverly, M., Morrison, J. et al. (2007). ATHLETE: A cargo handling and manipulation robot for the moon. Journal of Field Robotics, 24(5), 421.

Willow Garage, Inc. (2012). Robot Operating System. Retrieved October 12, 2011, from http://www.ros.org.

Winikoff, M. (2005). Jack Intelligent Agents: An industrial strength platform. In Bordini, R., Dastani, M., Dix, J., Fallah Seghrouchni, A., & Weiss, G. (eds.), Multi-agent programming, volume 15 of Multiagent systems, artificial societies, and simulated organizations (pp. 175–193). Springer US.

Yim, M., Zhang, Y., Roufas, K., Duff, D., & Eldershaw, C. (2002). Connecting and disconnecting for chain self-reconfiguration with PolyBot. IEEE/ASME Transactions on Mechatronics, 7(4), 2002.

Zuber, M. T., Head, J. W., Smith, D. E., Neumann, G. A., Mazarico, E., Torrence, M. H. et al. (2012). Constraints on the volatile distribution within shackleton crater at the lunar south pole. Nature, 486, 378–381.