

Heterogeneity for Increasing Performance and Reliability of Self-Reconfigurable Multi-Robot Organisms

S. Kernbach¹, F. Schlachter¹, R. Humza², J. Liedke⁴, S. Popescu¹, S. Russo³, T. Ranzani³, L. Manfredi³, C. Stefanini³, R. Matthias⁴, Ch. Schwarzer⁶, B. Girault⁵, P. Alschbach¹, E. Meister¹, O. Scholz²

¹Institute of Parallel and Distributed Systems, University of Stuttgart, Germany, ²Fraunhofer Institute for Biomedical Engineering (IBMT), St. Ingbert, Germany, ³The BioRobotics Institute, Scuola Superiore Sant'Anna, Italy, ⁴Institute for Process Control and Robotics, Karlsruhe Institute of Technology (KIT), Germany, ⁵École Normale Supérieure de Cachan, Cachan, France, ⁶Institute for Evolution and Ecology, University of Tuebingen, Germany

Abstract—Homogeneity and heterogeneity represent a well-known trade-off in the design of modular robot systems. This work addresses the heterogeneity concept, its rationales, design choices and performance evaluation. We introduce challenges for self-reconfigurable systems, show advances of mechatronic and software design of heterogeneous platforms and discuss experiments, which intend to demonstrate usability and performance of this system.

I. INTRODUCTION

Modular and reconfigurable robotics represents the area of mechatronic systems with dynamically changing structures and functionalities [1]. The principle of modularity is useful in obtaining several advanced properties such as reliability, adaptation [2], encapsulation of complexity [3], or in exploiting evolutionary and self-developmental capabilities of artificial systems [4]. Not only technological but also natural complex biological systems utilize the concept of modularity, which is known as multicellularity. Currently, artificial multicellularity attracted attention of researchers in exploring new biological inspirations, homeostatic mechanisms, macroscopic regulation and other issues [5].

State of the art solutions in both lattice-based and chain-based reconfigurable systems count primarily homogeneous platforms, e.g. [6], [7], [8], [9]. There are multiple reasons for such a development: complexity of mechanic, electronic and software design, exploration of reconfiguration approaches, finding a balance between "rigid" and "soft" design principles. In this work we primarily focus on challenges and driving forces for introducing heterogeneity into modular and reconfigurable systems.

Homogeneity and heterogeneity provide different advantages and represent two opposite points on the scale of universality and specialization [10]. For instance, homogeneous elements can be easily replaced, such systems are more scalable. Most artificial and natural swarms consist of structurally homogeneous elements, capable of behavioral specialization, e.g. [11]. However, increasing the complexity of the system, e.g. by aggregation into multi-robots organisms, we encountered bottlenecks in complexity of homogeneous elements, which are expressed in terms of weight, power limitation, or locomotion capabilities.

Contact author: serge.kernbach@ipvs.uni-stuttgart.de. IROS11, workshop on "Reconfigurable Modular Robotics", San Francisco, 2011.

This paper addresses the research and technological challenge of finding the compromise between homogeneous and heterogeneous design principles in general, and in particular within the scope of the European funded projects *Replicator* and *Symbion*. First of all, the degree of heterogeneity is an open-ended trade-off between complexity, scalability, cost factors, and required functionalities [12]. Finding a compromise, especially in the initial stage of development, is a tough problem [13]. To establish a rationale for the heterogeneous design, so-called challenges for modularity and reconfiguration are introduced. These challenges define not only the design choices but also set up experiments for performance measurement. We discuss these issues in Sec. II.

Heterogeneity requires specific technological solutions for docking elements, energy and communication buses, mechatronics, power management and others. The developed solutions are shortly overviewed in Sec. III. Special attention is paid to compatibility and common elements of this system as well as to a specialization of corresponding platforms. Secs. IV, III-H and Sec. V introduce a software architecture, manufacturing issues and two experiments, which aim at a qualitative demonstration of increased performance and reliability of this system. Finally, in Sec. VI we draw conclusions about achieving an optimal degree of heterogeneity.

II. SCIENTIFIC AND TECHNOLOGICAL CHALLENGES

The development started from the swarm concept and was originally intended for a small ($< 50 \text{ cm}^3$) reconfigurable homogeneous swarm robot. From the original idea to the current stage, the platform development can be split into 6 generations, see Fig. 1. Original design (generation 0) started in 2007 from the Jasmine III robot, where the 1st generation extended this design for docking and 3D actuation capabilities. This is a homogeneous platform, following a chain-based reconfigurable design.

Since new partners with additional resources joined the project, the envisaged demonstration scenario increased its complexity. The first step into heterogeneity was made with the 2nd generation design, which is primarily motivated by increased functional requirements, see Table I. After initial topological studies [3], it became clear that 1DoF (vertical and/or 180° rotational DoF) are enough for 3D actuation. However this system is not optimal for different required

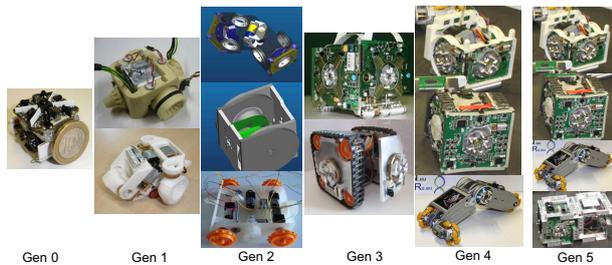


Fig. 1: From homogeneous to heterogeneous: five generations of platform development.

tasks. The proposal of making a complex universal homogeneous platform essentially increased the risk of particular or full failure due to complexity. Thus, the following factors: (a) different requirements of strong/weak DoFs in 2D and 3D mode, (b) geographic distribution of partners where a close collaboration became problematic, (c) increased risks of failure of a particular platform influenced the initial decision to split a design from one into two platforms.

Type	1st Gen	2nd Gen	3rd Gen	4th Gen	5th Gen
2D Locomotion	+	+	+	+	+
Autonomous	+	+	+	+	+
Docking					
3D DoF	1	2	3-4	3-4	3-4
3D Movement	-	-	+	+	+
Power Bus	+	++	++	++	++
Power trophallaxis	-	-	+	+	+
Commun. Bus	+	+	++	++	++
N of sensors	10	22	22	22	22
Passive Modules	-	-	-	-	+
Risks	-	+	++	+++	++++
Demo scenarios	3D Aggregation	Artificial Organisms	NIST Benchmarks	Grand Challenges	Key Highlights

TABLE I: General functional requirements, first published in [14].

These factors remained valid on all later stages, however we also encountered new bottlenecks. For example, during the third generation design, the conflict between the weight and available on-board power led to the decision to introduce the third platform (the active wheel), during the fourth generation – the conflict between the number of modules and developmental plasticity led to the decision to introduce additional modules (passive modules). Thus, each new platform either resolves a constraint or optimizes a performance of previous solutions.

Each year, the funding agency and project reviewers increased the level of challenges for the final demonstration even more, from originally intended aggregation in 3D up to the “Key Highlights”, see Table I. This essentially increased the impact of the projects, but also the complexity of platforms and risks of failures, see Table II. To some extent, the complexity between each generation increased by the factor of 3–4 in such criteria like on-board energy, computation capabilities or software architecture. Scientific and technological challenges, addressed by each design generation (expressed in terms of demonstration scenarios), vary

Type	1st Gen	2nd Gen	3rd Gen	4th Gen
MPU	AVR	ARM7	ARM11+2 Cortex M3	2x-core Blackfin + 4 MSP3100
MIPS	12	36-80	1100	3100
Memory	16kB	128kB	1mB	64mB
Power	1,4W/h	4W/h	13W/h	33W/h
OS	–	–	freeRTOS	μLinux
Bus	–	CAN	CAN/FlexRay	Ethernet

TABLE II: Several characteristics of the platforms.

from the aggregation in 3D up to complex ecologies with different robot species.

1. Aggregation: Docking of modules into 3D structures and a collective actuation within such structures.

2. Artificial Organisms: 3D structures possess homeostatic regulation, evolutionary framework, distributed control system, aggregate/disaggregate into different multi-robot organisms.

3. NIST Benchmarks: This scenario is motivated by benchmarks, developed in NIST, USA. Experiments in different environments should test and evaluate such parameters of robot organisms like flexibility, collective actuation, degree of adaptability, cognitive capabilities [15].

Feature	Description
1. Structural and Functional Diversity	Combination of different heterogeneous modules improves performance of common system and allows such a functionality, which was not possible for any of homogeneous elements.
2. Advanced Reliability	A high number of heterogeneous modules increase robustness and reliability of the system.
3. Adaptive (evolutionary) Structural and Functional Properties	High developmental plasticity enables short- and long-term adaptive and evolutionary processes. This results in a substantial increase of performance and efficiency in a predictably and unpredictably changing environment.

TABLE III: Key challenges addressed by heterogeneous platforms.

4. Grand Challenges: These are two complex high-impact scenarios. In the first scenario 100 robots should survive for 100 days without any human assistance in a changing environment. The second scenario was intended to explore evolutionary conditions similar to those that led to the appearance of multi-cellular species during the Cambrian explosion.

5. Key Highlights: These demonstration scenarios represent further refinement of the Grand Challenges with three or more dedicated experiments highlighting key scientific and technological challenges of artificial organisms.

Thus, each time we observe an increasing pressure on the number and complexity of different experiments. Generalizing the main challenges from all these scenarios, the advantages of heterogeneous platforms are located around three following aspects: *diversity*, *reliability* and *adaptivity*, see Table III. The following sections introduce these heterogeneous solutions and demonstrate their specialization and common elements.

III. TECHNOLOGICAL SOLUTIONS FOR LARGE-SCALE HETEROGENEOUS SELF-RECONFIGURABLE ROBOTS

Four platforms have been developed within the scope of this discussion, see Table IV, which are specialized in different DoF in 2D and 3D mode as well as optimized for different tasks. To make a joint operation of all these

Common Elements				
Docking	genderless, 90° symmetric, Ethernet and power buses			
Energy Bus	8 A at 25 V, current limiter at each module, controllable power distribution			
Battery	6 LiPo cells, 1,400 mAh, total nominal voltage: 22,2 V			
Communication Electronics	Internal Ethernet bus, ZigBee, local IR, sound			
Software	Universal electronic architecture, the same set of sensors			
	Universal software framework			
Heterogeneous Elements				
	Scout Platform	Backbone Platform	Active Wheel	Passive Modules
Specialization	exploration tasks	strong 3D actuation	transportation tasks	e.g. add. power, electronics
Locomotion	tracked locomotion	nearly omnidirectional	Omni-directional	none
<i>N</i> of Docking Elements	4	4	2	1 - ...
Speed, loc.	12.5 cm/s	6 cm/s	31 cm/s	none
DOFs of actuation	bending: ±90° rot.: ±180°	<i>bend./rot.</i> : ±90°	bend./rot.: ±180°	none
Torque	up to 4 Nm	up to 7 Nm	up to 5 Nm	none
Speed, act.	37.2°/s	90°/s	50°/s	none
Volume	1356 cm ³	1157 cm ³	1470 cm ³	
Weight	1000 g	1000 g	1550 g	

TABLE IV: Overview of common (unified) and heterogeneous elements in the design of heterogeneous reconfigurable robots.

platforms possible, a number of common elements have been developed. The following sections provide a detailed sight into these elements.

A. Common Elements: Unified Docking Mechanism

A key element of heterogeneous robot platforms is the ability to dock mechanically and electrically to merge into a larger artificial multi-cellular organism. The docking design used in our modular robots is called CoBoLD (for Cone Bolt Locking Device, see Fig. 2) and among other features it is genderless, 90° symmetric and handles misalignment.

The design itself is based on 4 cone shaped, spring loaded bolts. During docking the bolts will be guided by chamber-bevels of the facing docking units inserted holes and finally locked inside by a hook. The locking mechanism uses a DC motor driven worm gear providing enough force to securely connect two robots. Due to the self-locking feature of a worm gear, no energy is required to hold the connection.

It is possible to push the bolts inside the docking unit resulting in a single boundary box regardless of the orientation or state of the locking mechanism. This can be important, if e.g. one robot is turned lying on its docking unit and

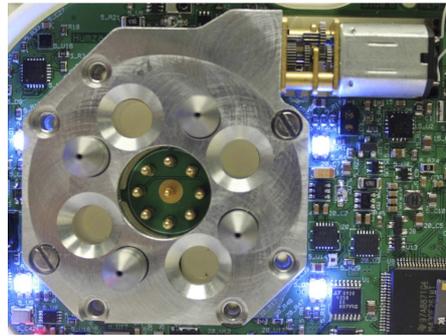


Fig. 2: Unified homogeneous docking unit.

another robot needs to dock. In the center of the docking unit, spring loaded electrical contacts are placed to assure an automatic connection of power and wired communication when two robots dock. The contacts are placed on the same PCB holding all the other electrical components on this robot side. This eases the electronic design since no cables or additional contact PCBs are required. The arrangement of these contacts plus a special switching circuit ensure that robots can dock no matter how they are oriented with respect to each other.

Several versions of CoBoLD are available, depending on the required functionality. For example, if active docking (the ability to lock bolts) is not required, a passive docking unit can be used, reducing the size of the assembly while still providing for all other mechanical and electrical features.

B. Common Elements: Unified Energy Sharing System

A “unified energy sharing” system of a reconfigurable multi-robotic organism takes its inspiration from the ant’s social stomach. It not only allows them to exchange power but also distributively and collectively monitors and regulates the dynamic power flow between the multiple power sources that are now connected in parallel through an organism’s power sharing bus. A “unified solution” is in-fact required to reduce the complexity of such a dynamically evolving system [16]. In principle, it provides a base platform to the application layer control routines to develop and evolve complex behaviors with the changing environment [11], e.g., establishing energy homeostasis, power aware fault tolerance, dynamic power sharing, emergent phenomena based on energy distribution, etc [17].

Fig. 3 shows the block diagram of the proposed power management system of a single robotic module. The hardware design of the power management system includes a battery management module, a battery re-charging unit, ideal diodes for uni-directional low loss current flow, a system power manager, an energy sharing module, a high power control switch for powering on-board peripherals, and four docking interfaces one on each lateral side of the robot.

The unified energy system mainly consists of 4 docking interfaces and an energy sharing module. The on-board energy sharing module in its ON state allows an individual to share its battery energy with other modules. While in its OFF

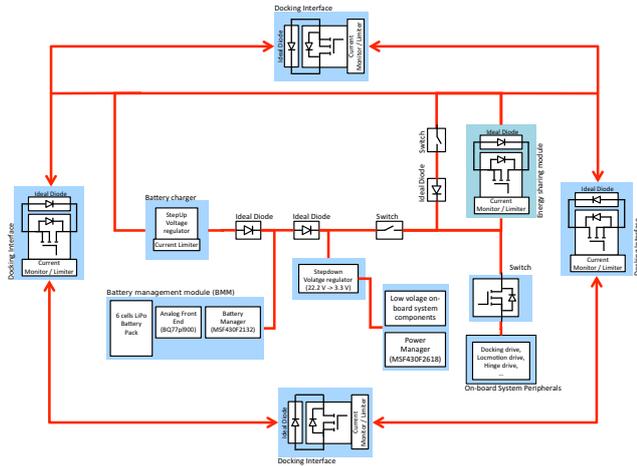


Fig. 3: Block diagram of the power management system.

state, allows an inward current flow, provided that the on-board source voltage is lower than the voltage on the power bus. The docking interface on each side of the robot provides the on-board power manager with a means to control the outward flow of energy. Additionally, the current monitor at each docking interface and in the energy sharing module allows the system power manager to measure the current flow in either direction. This way, each individual distributively but collectively controls the current flow through the organism's power bus. Besides the common power management architecture, all platforms are equipped with the same custom designed battery packs.

C. Common Elements: Electronic Framework

The general electronic architecture is shown in Fig. 4. The

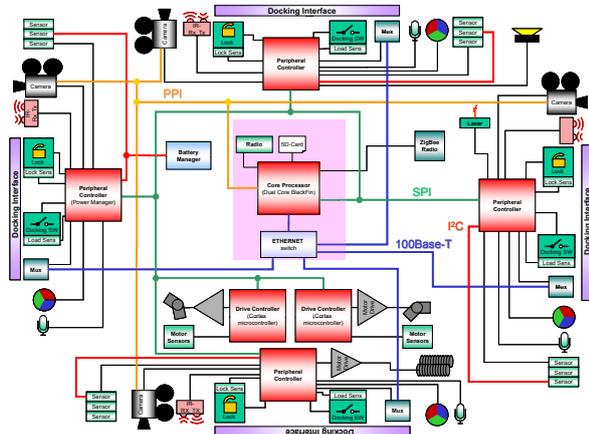


Fig. 4: Block diagram of the electronic architecture.

main processing unit of each of the robot platforms is a *CM-BF561* drop-in module from BLUETECHNIX equipped with a dual core *Blackfin BF561* μ controller from ANALOG DEVICES. Via a Serial Peripheral Interface (SPI), this unit is attached to 4 peripheral μ controllers (*MSP430F2618* from TEXAS INSTRUMENTS), which are responsible for sensor data acquisition, low level actuator control and processing in

order to take off burden from the main processing unit. These peripheral controllers mainly serve as an interface between the application software routines running on the *Blackfin* processor and the system components. All μ controllers and corresponding peripheral elements are placed on separate PCBs, installed on each side of a robot module. Each of these PCBs have in common certain sensors and actuators (e.g. docking sensors and actuators, microphones, RGB-LEDs, etc.), but may in addition take over specialised tasks (e.g. 2D or 3D locomotion, ZigBee radio interface, etc.). For 3D actuation, up to two additional *LM3S8962 Cortex* μ controllers (LUMINARY MICRO) are integrated to permit high-performance brushless DC-motor control. On each of the PCBs, a local I²C bus is implemented for interfacing the local controller with the respective sensors. In addition, a global I²C bus has been implemented to facilitate multi-master communications between the peripheral controllers.

D. Heterogeneous Elements: Backbone Platform

The backbone platform, see Fig. 5, is specialized in strong 3D locomotion and actuation capabilities in the organism mode. The robot possesses a strong brushless drive capable of lifting several robots. A single DOF approach reduces the cost and since the main motor is a rotation-symmetric subassembly, the available space inside is better suited to support one strong motor instead of several smaller ones. To compensate the reduced number of DOFs, the frame of the robot uses two L-shaped halves, which can be rotated against each other. This allows for lifting or rotating connected robots depending on their docked position. A symmetric, genderless, active docking was chosen to eliminate the need to search for compatible docking interfaces. This increases the diversity of possible organism structures that can be built. In order to use the feature of the four mounted docking units, which not only provide for a stable connection but also for power and communication transfer, a special 2D drive was included allowing the robot to additionally move sideways. The robot can move freely as a single unit and it is possible to use the 2D drives of several connected robots to drive the organism. A cubical shape was chosen because in our opinion it combines best the requirements for a small sized swarm robot and the requirements for modular self-reconfigurable robots, where a symmetric shape would simplify the reorganisation of an organism.

So far, we demonstrated that the 2D drive can move a single robot in all directions, all docking units can be used to connect the robots. The cubical shape allows for docking no matter how the module is oriented. In addition, docking between the different platforms is possible. General functionality of the 3D drive was demonstrated in the 3rd generation module, the implementation into the current design of the robot is ongoing work. Due to the internal design, bending and rotating is limited to 180°.

E. Heterogeneous Elements: Active-Wheel Platform

The Active Wheel, see Fig. 6, is a fully functional platform, specialized in transportation tasks. It was designed to



Fig. 5: A rapid prototype version of the Backbone platform.

carry and transport an organism consisting of several Scout or Backbone robots in a most energy-efficient way. The robot is able to approach the requested module/organism and dock to it. The current design consists of two symmetrically arranged arms. On each end two 90° shifted Omni-Wheels are attached, which together form the locomotion mechanism of the robot. These arms are connected via a 180° turning hinge. Docking to other robots is provided by 2 docking elements placed on the same axis as the hinge. To allow docking in any position or height, the docking elements can rotate, actuated by two separate motors. For the production of the prototypes a vacuum-casting process was chosen. This method allows to produce multiple chassis in one working step simultaneously. Thereby, the manufacturing cost of the mechanics was reduced by 40 % compared to standard rapid prototyping.

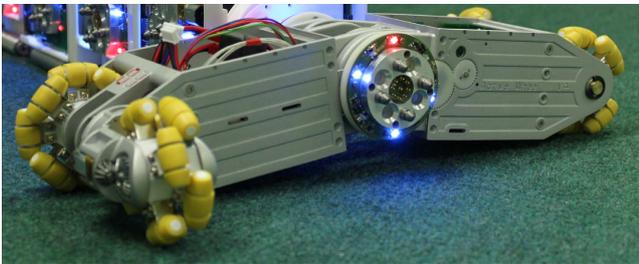


Fig. 6: Active wheel.

The chassis of the robot has two cavities, which are ideal to place the main electronics. On the one hand, this configuration protects the sensitive components from damage; on the other hand, it allows placing all sensors outside in order to perceive the environment. However, this separates the electronics into different areas and increases the number of necessary printed circuit boards. In total, 15 boards were installed on each robot, whereby 6 of them are used as a sensor for detecting the motor speed for driving and docking.

F. Heterogeneous Elements: Scout Platform

The Scout robot, see Fig. 7, is specialized in fast locomotion on challenging terrains for exploration tasks. Therefore, the platform is equipped with additional sensors compared to the other platforms in order to scan the surroundings

and the floor. In particular, the Scout robot has two laser-camera systems, on its front and rear to provide far and short range obstacle detection. The Scout robot's locomotion is based on tracks. Fast locomotion for the exploration on rough terrains is more important for the Scout robot than slow and precise locomotion for aggregation and docking alignment, compared to the Backbone Robot. Tracks enable the Scout robot to move forward, backward, turn left, turn right and turn on its axis. Moreover, the Scout robot can move on rough terrains, climb slopes and overcome small obstacles. The speed of locomotion is more than one body length per second. Thanks to continuous elastic tracks, the robot is able to perform locomotion even after overturning accidentally.

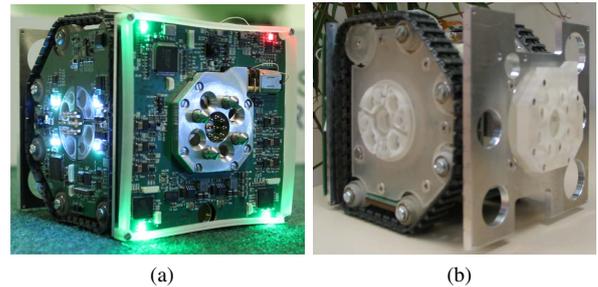


Fig. 7: The Scout robot platform: (a) front and right sides of the Scout robot, (b) rear and left sides of the Scout robot with stripped off PCBs.

Similar to the Backbone robot, the Scout robot has four docking units, one on each lateral side wall. The docking units are centered on the walls of the robot so that modules can dock regardless of their orientation. This platform has 2 DOF, bending and rotation, with a maximum torque of 4,7 Nm. The bending DOF ($\pm 90^\circ$) allows the robot's rear wall to lift up and down while the rotational DOF ($\pm 180^\circ$) allows the docking unit on the robot's left side to rotate along its axis. The 3D DOF has lower torque than the Backbone robot since the main role of the platform is exploration and not macro-locomotion. In any case, the Scout robot is able to lift two other modules. The Scout robot, with higher DOF and lower torque compared to the Backbone robot, performs best if docked to the tail or to the head of an organism to scan the environment.

G. Heterogeneous Elements: Passive Modules

Passive modules will increase the heterogeneity degree of an organism. Depending on the design, a passive module can have the same shape like a robot to match inside an organism. It is equipped with at least one docking unit robots can connect to and e.g. is equipped with additional power packs or electronics. Such a passive module could increase for example the operation time of an organism or add additional features like specialized sensors and high-resolution cameras.

Since passive modules can be used in place of a robot, each version of must be able to interact with a docked robot. Passive modules are therefore equipped with the

same communication interfaces (wired, wireless) and power circuits like a robot to assure communication is forwarded and power can be distributed. Furthermore, since passive modules interact with the organism (e.g. processing camera pictures in a module with a specialized FPGA mounted) a connection to the internal communication bus is available.

H. Software Platform

Although we have heterogenous platforms, unified interfaces are necessary for the ease of use of the resulting software. It needs to be stated here that the choice to go for a Blackfin DSP as main CPU and to have several minor processing units already implies some constraints for the general software architecture. It is intended to have a layered architecture consisting of three layers. Those three layers as well as their composition with respect to each other can be seen in Fig. 8.

Due to the way the μ Clinux OS running on the Blackfin DSP deals with scheduling threads on both cores it is necessary to have several heavy weight processes within the architecture. Therefore the need for inter-process communication arises, which will be handled within the core layer (CORE). This layer will mainly be comprised of a Daemon which monitors the communication among all the processes. The application layer (APP) is to hold controllers and high-level behaviors. It is able to utilize the functionality of the layers residing under it. The hardware-abstraction layer (HAL) underneath the core layer will deal with all the physical interfaces the Blackfin provides. It is meant to provide further hardware abstraction and to hide the complexity of the underlying hardware up to a certain extent. Each one of the white boxes in Fig. 8 denotes a heavy weight process and encapsulates all the functionality of the according hardware interface. The two most complex parts here will be the Ethernet interface and the SPI-bus interface.

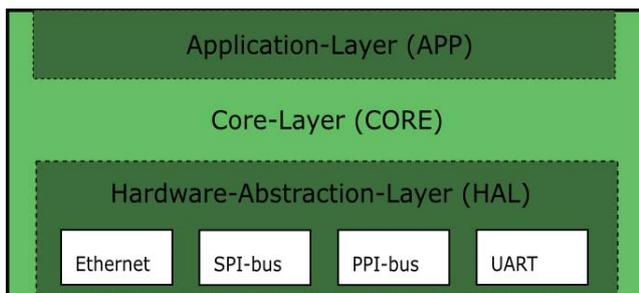


Fig. 8: Layered architecture of the software framework.

The SPI-bus deals with most of the local functionality of the robots, as it connects the MSP controllers to the Blackfin. It therefore needs to be explained here in detail. Several tasks have to be performed within the SPI-process in Fig. 8: Initiating communication to the MSP controllers, scheduling the use of the bus to ensure proper bandwidth allocation and error handling as well as correction. Furthermore the differences within the electronics among the three different platforms have to be taken into account here. The software

framework therefore supports all three types of robots and allows a reusable design of controllers for the robots. For all robot types a common interface exists, which allows to write controllers for all types of robots. The interface can distinguish between the types and is extendable to new functions or future robot design. One part of the SPI software is run on the MSP controllers. The other part resides on the Blackfin within μ Clinux. Fig. 9 shows the principle of the SPI-software and the intended data flow on the robots.

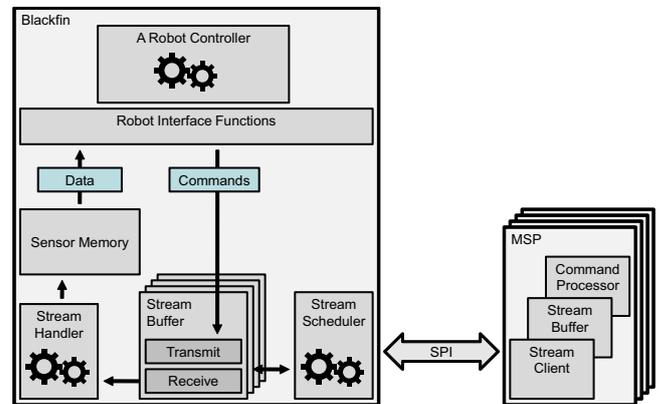


Fig. 9: Architecture of the SPI-software.

The SPI communication is a bidirectional stream and is driven by the Blackfin. Additionally, the MSP can trigger a read, if new data is available. The received sensor data on the Blackfin is pushed into a sensor memory, where controllers can access the data via the interface later on. Vice versa, interface commands sent by a robot controller are transmitted via SPI to the designated sub processor.

Due to the different robot type based characteristics an adjusted hardware abstraction layer is implemented for each robot. This layer is specialized for each robot type and is abstracted to the user for better use. A common interface unifies the functionality of all three robot platforms into a collection of higher level functions. Most of the functions can be used on all three robots (e.g. sensor access, communication). Some of the functions, like locomotion, are specialized to the robots, but can be abstracted to simple common locomotion directives, like MOVE or TURN, which are valid for all types again. Via the interface, a controller can distinguish on which robot platform it runs and call the according function (e.g. four-wheeled locomotion for the Active Wheel).

IV. MANUFACTURING ISSUES

Manufacturing of a large number of modules is a critical process. Currently we produced 15 prototypes of fully functional robots (5 pieces of each platform) by using RP technology, see Fig. 10. This intermediate run is used for improvements, resolving hardware problems as well as for developing the software framework. In the final run, it is planned to produce a large scale swarm (100+ robots). Manufacturing and maintaining such a number of complex modules involves new aspects into design of platforms:



Fig. 10: Manufactured heterogeneous robot platforms.

simplicity of assembly, protection of electronics, allocation of funding for maintaining these robots and for performing experiments. Generally, PCBs, SMD assembly and mechanical elements can be ordered. However, a final assembling and testing of modules represents a highly challenging problem for academic partners.

V. PERFORMED EXPERIMENTS

The following two scenarios aim at a demonstration of simplest qualitative cases leading to improved performance and increased reliability of a heterogeneous system.

A. Improving Performance of a Heterogeneous Organism

In this scenario the robots build up a heterogeneous organism of several modules. The objective is to increase the common computational and energetic performance and to create a computationally powerful and efficiently moving organism. By combining two types of robots, the computational power can be raised. Each of the modules adds additional computational power of around 3000 MIPS to the robot organism. Since the Active Wheels can't dock to each other due to their shape, the Backbone robot is used as a connector or skeleton module. The final configuration represents a distributed 4x-2core Blackfin computational systems with an Ethernet bus between CPU nodes. In addition, the Backbone robot adds additional power sources of 33 Wh and sensors to the organism and improves the sensory awareness on the sides, where the Active Wheel has only limited sensors. Thus, two additional Backbone robots, which are not used for locomotion, provide additional 66 Wh energy for the organism. Furthermore, the locomotion of these four robots is faster (31 cm/s for all robots instead of minimal 6 cm/s) and more efficient. Using the wheels of the Active Wheel instead of the screws and lifting up some robots from the ground minimizes the power consumption for locomotion. Thus both robot types can complement each other. The video sequence in Fig. 11 shows how multiple robots connect to a heterogeneous organism. In (a)-(b), an Active Wheel connects to two already connected robots. In (c)-(e) the fourth robot docks to the organism and (f) shows the final organism.

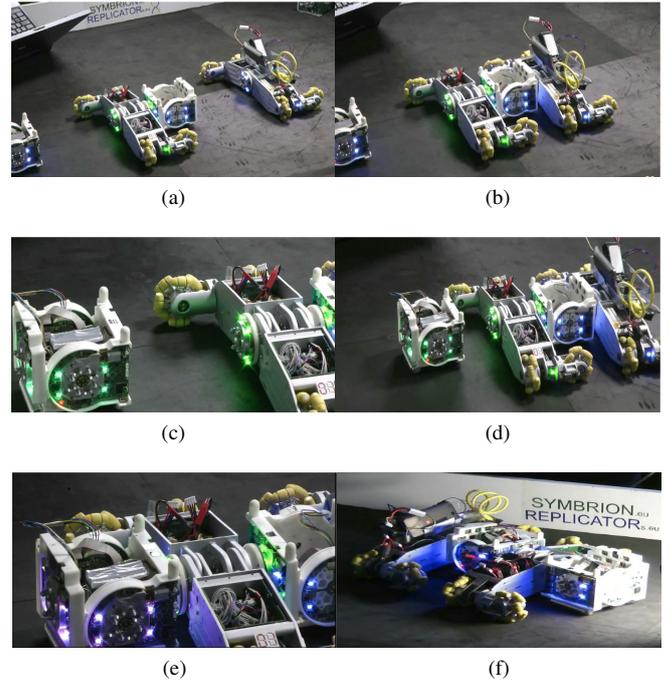


Fig. 11: Sequence of the assembling of a heterogeneous organism.

B. Improving Reliability: Rescue Scenario

In this scenario a Backbone robot has fallen on its side for some reason and cannot move any more. This case represents a failure of a platform and excludes it from all further activities. To escape this situation, the robot can call other robots for help. An Active Wheel may catch the cry for help and come to assist the Backbone robot. Because of the rotational docking element on the Active Wheel, it has the capability of rotating another robot. Since the robot cannot be rotated on the ground, the second capability of lifting a robot comes into play. Combining these two features of the Active Wheel allows to lift up the Backbone Robot, turn it around and put it back on its screw drive. A series of images of one of these trials can be seen in Fig. 12. In (a)-(c) the Active Wheel is approaching the Backbone robot. In (d) it lifts the fallen robot and rotates it in (e)-(g). (h) shows the Active Wheel putting the Backbone robot back onto the ground and finally in (i) the Backbone robot can continue its activities.

VI. CONCLUSION

In this work we presented the heterogeneous approach of a self-reconfigurable system, its design rationales, main argumentations and challenges. Four heterogeneous platforms, specialized in different tasks, as well as their common elements such as docking mechanism or electronics were shortly demonstrated.

Originally, the self-reconfigurable platform was planned as a homogeneous swarm-like robot of the size $\sim 50 \text{ cm}^3$, weighing around 200 g. Multi-robot organisms should consist of a large number of such modules. The main factors which introduced and continuously increased the level of

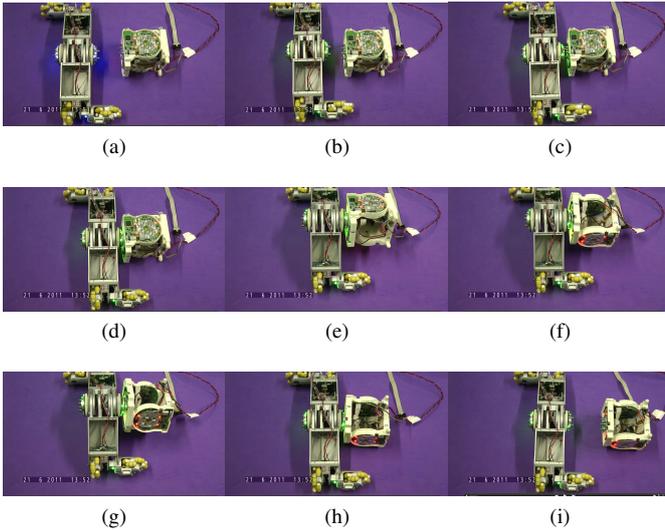


Fig. 12: Sequence of the rescue scenario.

heterogeneity are physical constraints imposed on the system in the organism mode: the number of modules to be lifted, mechanical strength of docking elements or requirements for energy/communication buses. This finally led to the size of $\sim 1000 \text{ cm}^3$ and the weight of around 1 kg per module and a lower number of modules in the artificial organism.

Generalizing this experience, we argue that the degree of heterogeneity is defined by a diversity of required functionalities. As demonstrated by the ATRON and MTRAN systems, morphing capabilities can be achieved with only one type of modules. As soon as any additional requirement is introduced, the optimization of the system's performance leads to an appearance of specialized modules. The optimal degree of heterogeneity is a trade-off between optimization of performance and such factors like "replaceability of modules", "costs of manufacturing", "maintenance", "available developmental power", i.e. with organizational/managerial factors.

Heterogeneity strongly requires several universal elements, this can be denoted as "homogeneity inside of heterogeneity". In our example, these are docking mechanism, communication and energy sharing system. To some extent, compatible electronic and software frameworks in all modules are also necessary to minimize the effort of development and for performing experiments.

We demonstrated the benefit of a heterogeneous system by two qualitative experiments: aggregation into an artificial organism, where the organism became much more computationally and energetically powerful than a sum of non-aggregated modules, and by a rescue scenario, where heterogeneous robots assist each other in increasing the reliability of the whole system.

ACKNOWLEDGEMENT

The authors are supported by the following grants: EU-ICT FET project "SYMBRION", grant agreement no.

216342; EU-ICT project "REPLICATOR", grant agreement no. 216240. Additionally, we want to thank all members of the projects for their cooperation and fruitful discussions.

REFERENCES

- [1] Esben Hallundbæk Østergaard, Kristian Kassow, Richard Beck, and Henrik Hautop Lund. Design of the atron lattice-based self-reconfigurable robot. *Auton. Robots*, 21:165–183, September 2006.
- [2] S. Kernbach, E. Meister, F. Schlachter, K. Jebens, M. Szymanski, J. Liedke, D. Laneri, L. Winkler, T. Schmickl, R. Thenius, P. Corradi, and L. Ricotti. Symbiotic robot organisms: REPLICATOR and SYMBRION projects. In *Proc. of Performance Metrics for Intelligent Systems Workshop (PerMIS-08)*, pages 62–69, Gaithersburg, MD, USA, 2008.
- [3] S. Kernbach, L. Ricotti, J. Liedke, P. Corradi, and M. Rothermel. Study of macroscopic morphological features of symbiotic robotic organisms. In *Proc. of the International Conference on Intelligent Robots and Systems (IROS-08), Workshop on Self-reconfigurable robots*, pages 18–25, Nice, France, 2008.
- [4] S. Kernbach, E. Meister, O. Scholz, R. Humza, J. Liedke, L. Ricotti, J. Jemai, J. Havlik, and W. Liu. Evolutionary robotics: The next-generation-platform for on-line and on-board artificial evolution. In A. Tyrrell, editor, *Proc. of the IEEE Congress on Evolutionary Computation (IEEE CEC-2009)*, Trondheim, Norway, 2009. IEEE Press.
- [5] P. Levi and S. Kernbach, editors. *Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution*. Springer-Verlag, 2010.
- [6] B. Salemi, M. Moll, and W.M. Shen. Superbot: A deployable, multi-functional, and modular self-reconfigurable robotic system. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-06)*, pages 3636–3641. IEEE, 2006.
- [7] Kasper Stoy. How to construct dense objects with self-reconfigurable robots. In H. Christensen, editor, *European Robotics Symposium 2006*, springer tracts in advanced robotics 22, pages 27–37. Springer-Verlag, Berlin, Heidelberg, New York, 2006.
- [8] David Johan Christensen, Esben Hallundbæk Østergaard, and Henrik Hautop Lund. Metamodule control for the atron self-reconfigurable robotic system. In *Proc. of the IAS-8*, pages 685–692, Amsterdam, 2004.
- [9] S. Murata, E. Yoshida, H. Kurokawa, K. Tomita, and S. Kokaji. Concept of self-reconfigurable modular robotic system. *Artificial Intelligence in Engineering*, 15(4):383–387, 2001.
- [10] S. Kernbach, V. Nepomnyashchikh, T. Kancheva, and O. Kernbach. Specialization and generalization of robotic behavior in swarm energy foraging. *Mathematical and Computer Modelling of Dynamical Systems*, DOI 10.1080/13873954.2011.601421, 2011.
- [11] S. Kernbach and O. Kernbach. Collective energy homeostasis in a large-scale micro-robotic swarm. *Robotics and Autonomous Systems*, DOI 10.1016/j.robot.2011.08.001, 2011.
- [12] Serge Kernbach, Oliver Scholz, Kanako Harada, Sergej Popesku, Jens Liedke, Humza Raja, Wenguo Liu, Fabio Caparrelli, Jaouhar Jemai, Jiri Havlik, Eugen Meister, and Paul Levi. Multi-robot organisms: State of the art. In *ICRA10, workshop on Modular Robots: State of the Art, Anchorage, 2010.*, 2010.
- [13] S. Kernbach. *Structural Self-organization in Multi-Agents and Multi-Robotic Systems*. Logos Verlag, Berlin, 2008.
- [14] D1.1: Specification of hardware architecture. Technical report, FP7 GA No.216240 REPLICATOR and FP7 GA No.216342 SYMBRION, 9 2008.
- [15] S. Kornienko, O. Kornienko, C. Constantinescu, M. Pradier, and P. Levi. Cognitive micro-agents: individual and collective perception in microrobotic swarm. In *Proc. of the IJCAI-05 Workshop on Agents in real-time and dynamic environments, Edinburgh, UK*, pages 33–42, 2005.
- [16] L. König, K. Jebens, Serge Kernbach, and Paul Levi. Stability of on-line and on-board evolving of adaptive collective behavior. In Herman Bruyninckx, Libor Preucil, and Miroslav Kulich, editors, *European Robotics Symposium 2008*, pages 293–302. 2008.
- [17] S. Kornienko, O. Kornienko, and P. Levi. Generation of desired emergent behavior in swarm of micro-robots. In R. Lopez de Mantaras and L. Saitta, editors, *Proc. of the 16th European conference on artificial intelligence (ECAI 2004), Valencia, Spain*, pages 239–243. Amsterdam: IOS Press, 2004.