Evolutionary Robotics: The Next-Generation-Platform for On-line and On-board Artificial Evolution

S. Kernbach*, E. Meister*, O. Scholz†, R. Humza†, J. Liedke‡, L. Ricotti§, J. Jemai¶, J. Havlik‖, W. Liu**

*Institute of Parallel and Distributed Systems, University of Stuttgart, Universitätstr. 38, 70569 Stuttgart, Germany
† Fraunhofer Institute for Biomedical Engineering, Ensheimer Strasse 48, 66386 Sankt Ingbert, Germany
‡ Institute for Process Control and Robotics, University of Karlsruhe, Engler-Bunte-Ring 8, 76131 Karlsruhe, Germany
§ Scuola Superiore Sant'Anna Viale Rinaldo Piaggio 34, 56025 Pontedera, Italy
¶ Ubisense AG, Landshuter Allee 12, 80637 Munich, Germany
‖ IMA, s.r.o. Valentince 1003/1, 150000 Prague, Czech Republic
** University of the West England, Bristol, Coldharbour Lane BS16 1QY Bristol, UK

Abstract— **In this paper we present the development of a new self-reconfigurable robotic platform for performing on-line and on-board evolutionary experiments. The designed platform can work as an autonomous swarm robot and can undergo collective morphogenesis to actuate in different morphogenetic structures. The platform includes a dedicated power management, rich sensor mechanisms for on-board fitness measurement as well as very powerful distributed computational system to run learning and evolutionary algorithms. The whole development is performed within several large European projects and is open-hardware and open-software.**

I. INTRODUCTION

Currently, the evolutionary paradigm represents a new research trend for adaptive, reconfigurable and collective robotic systems [1]. These systems are primarily characterized by their ability to reconfigure autonomously, where the robots can work either individually or aggregate autonomously into collective robotic organisms [2]. The application of an evolutionary approach for these systems enables adaptation to new, even previously unknown, environments by evolving an individual or cooperative behavior (e.g. [3], [4]). In particular, a collective actuation of robotic organisms, being a complex function of many interacting individuals, can be obtained in the evolutionary way [5], [6]. Here, we think about several relevant long-term applications, where a continuous communication with a human operator is impossible, like deep space or ocean explorations, robotic Mars missions or other human-less environments. However, working on such systems, we face a few serious fundamental issues needed to be considered.

Firstly, such robotic systems operate either in fully autonomous mode or with intermittent human corrections of strategic goals. Consequently, the learning and evolutionary processes should be executed only by those means which are available on board. This is related to an acquisition of sensor data, fitness estimations, computational and communication resources. Secondly, the robotic evolution should succeed within operative time-scale of a concrete environment. Usually, it means a time interval between several minutes and several hours. We consider these properties necessary for on-board and on-line evolving. Finally, several essential issues appear due to reconfigurability of the system. Since a collective functionality depends on the morphology of robotic organism, the evolution process of a collective functionality implies a collective morphogenetic process. This, in turn, poses many open questions regarding the relationship between functions and structures [7], between individual and collective fitness, between learnt and evolved behavior.

Hence, for the development of real evolutionary adaptive systems, we need a robotic platform which can provide several specific capabilities. First of all, it should be self-sufficient in order to support self-reconfigurable capabilities with individual/collective actuation and with autonomous docking mechanisms. Moreover, it should possess enough computational power and sensing capabilities. Finally, it should be reproducible by interested research groups with a reasonable amount of effort. The last point is closely related to the following issue.

Developing approaches and mechanisms for such evolutionary reconfigurable systems, we faced the problem, known as a "gap of reality". Since on the early stage of projects no real hardware was readily available, simulations were used allowing a preliminary software development. However, these simulations do not provide the complexity of a real environment, resulting in simplified sensor data, which do not reflect many real non-linear phenomena (like reflection of IR-light around proximity sensors). Moreover, deriving a fitness functions in simulative environments, we often encounter a "linearity" of fitness. The "linear fitness" does not consider some specific "ecological fitness niches" existing in a real fitness landscape. As a result, such evolutionary solutions, which are obtained in reality, are not accessible in simulation [8]. As a reaction to the "gap of reality", we developed the "hardware simulator", a composition between real hardware and computer simulation, which is easily available for the software community and provide a reality-

Contact author: serge.kernbach@ipvs.uni-stuttgart.de

978-1-4244-2959-2/09/$25.00

close evolving of robot controllers.

In this paper we represent the development of such a self-reconfigurable platform for evolutionary community. This development is conducted by a consortium of twenty research organizations within the European projects “SYMBRION” [9] and “REPLICATOR” [10]. The development as well as the platform itself are intended to be open-hardware and open-source. Moreover, on the early developmental stage, the consortium is collecting ideas and requirements from the areas of evolutionary computation, swarm and reconfigurable robotics in order to integrate as many trends of modern robotic research as possible.

This paper is organized as follows: In section II we collect the main evolutionary requirements imposed on the platform. In the following three sections III, IV and V we consider the capabilities of autonomous morphogenesis, on-line and on-board evolution and on-board fitness measurement more in detail. The section VI is intended to show the “hardware simulator”. Finally, we conclude this work in Section VII.

II. GENERAL CONCEPT

The general concept of the hardware development is aimed at satisfying three important requirements:

1) Capabilities of autonomous morphogenesis;
2) Performing on-line and on-board evolving approaches;
3) On-board fitness measurement.

We consider pragmatical short-term applications in indoor environments, therefore all requirements are applied to in-house scenarios, as e.g. reconfigurable mobile sensor networks.

The first and the most important requirement is related to the **capability of autonomous morphogenesis**. This means that the robot organism consists of small elementary modules, see Fig. 1. These modules can operate and move independently as robot individuals. For that they have on-board energy, computation, and communication capabilities. These modules can autonomously dock to each other and share energy and computational resources through internal buses. In this way, modules can aggregate and disaggregate any time in different forms or even dynamically change the collective shape. This requirement is the most demanding one because it imposes essential constraints on mechanical structures, docking elements, electrical connectors and energetic homeostasis.

The **on-line evolution** means a capability to perform the cycle *sensing→evolving→reaction* in such a time when the environment still remains unchanged (or almost unchanged). In typical in-house environments we distinguish three time-scales: short-term (seconds-to-minutes), middle-term (20-60 minutes) and long-term (several hours). The platform reacts on short-term events based on rule-based or plan-based mechanisms. However, the reaction on middle-term and long-term events may be based on evolutionary approaches. Therefore evolving in middle and long-term time scales we denote as on-line evolving.

The “**on-board**” generally means that a platform works completely autonomous, i.e. all computational processes run by using on-board resources. However, at the beginning of the development, it was clear that the evolutionary techniques should be combined with other approaches, like learning, artificial neuronal networks or artificial immune systems. Therefore on-board in this context means not only a demand on computational resources, but also specific energy management, sensing and communication.

Capability for a **local fitness measurement** is a key issue for on-line and on-board evolution. The idea is that a robotic organism can use its local sensors as well as communication and interaction with other organisms in order to estimate its own fitness related to:

- the local situation around the organism;
- neighbors surrounding the organism;
- the approximation of global situation.

Typical local fitness measurements are e.g. estimation of real and virtual energy for specific activities, number of collisions with other organisms, number of “sexual” contacts and descendants, area of explored territory and so on. Local fitness measurement means not only different sensors but also sensor-fusion approaches as well as world modelling to extract the context of sensor data. This, in turn, requires computational resources and memory. In the following sections, we consider these three issues more in detail.

III. CAPABILITIES OF AUTONOMOUS MORPHOGENESIS

The design philosophy behind the self-reconfigurable robotic platform is to provide abilities to an individual module that can undergo autonomous morphogenesis within a changing environment. To achieve self-sufficiency [11] and autonomy in the platform design following three design elements are critical:

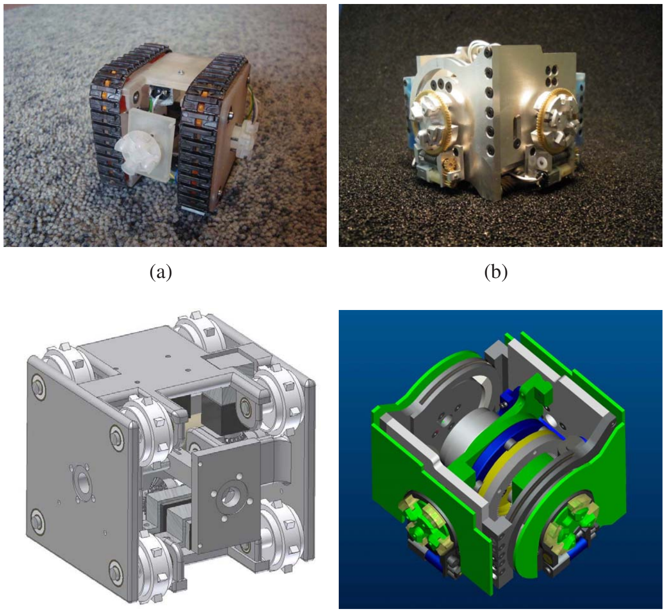
1) Autonomous movement of each module;
2) Docking mechanism;
3) Power sharing and energy management.

A. Capabilities for autonomous morphogenesis: individual modules and their docking

At the moment, two different approaches are developed as seen in Fig. 1. For autonomous locomotion of each module, the first design, see Fig. 1(a,c), uses a differential chain drive powered by a standard DC-gearbox motor. The revolving chain enable the robot to pass small obstacles (e.g power cables) as well as move on top. This is an advantage, if the robot is used inside an organism and can’t be placed on floor for disaggregation.

The second design, shown in Fig. 1(b,d) is based on the assumption, that a single robot should be able to move freely in its environment. Therefore an omnidirectional drive also using DC-gearbox motors is used. Due to space restrictions, the usual approach of omnidirectional wheels (e.g. Mecanum wheel) couldn’t be used. Instead, a roll barrel with a thread combined with proper control algorithms allows the robot to perform all necessary movements.

To be able to build up an organism or transport tools (see Fig. 2), each robot is equipped with four docking elements, one for each direction. The docking elements allow a stable,

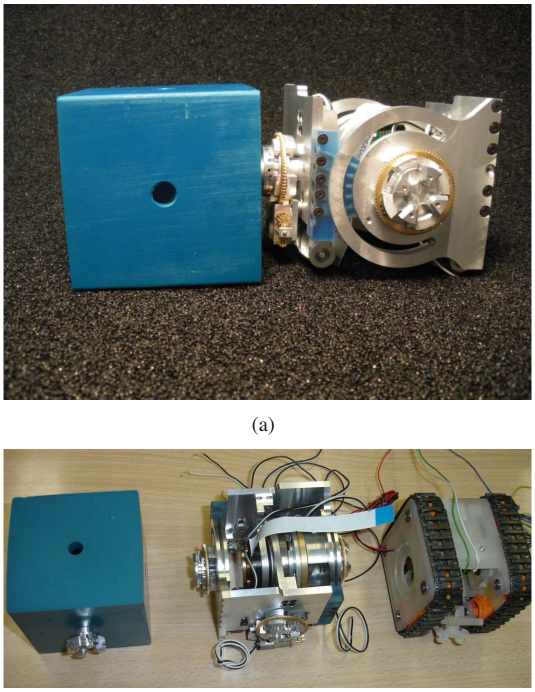

(a) (b)

(c) (d)

Fig. 1. *Hardware platform design for Replicator-Symbrion robots. **(a,c)** Design from Scuola Superiore Sant'Anna of Pisa; **(b,d)** Design of University of Karlsruhe (TH).*

physical connection between two robots by form closure in such a way that electrical energy is needed only during docking. The docking ports are designed to handle misalignment in horizontal and vertical direction as well as rotation. Also, the docking element is genderless (all ports look alike) and can dock to other robots which are turned ±90° and ±180°. For power sharing and inter-robot communication electrical contacts are included into the docking element, which will be coupled automatically. For compatibility, both robot designs and diverse tools (see [12]) use the same docking element. Therefore, multiple organisms can be built by combining different structures/functions of active platforms and passive tools.

There are several aspects related to the constructional differences between two platforms, like weight, availability of internal space, power resources or motor torque. The first platform has less weight, however provides more internal space than the second platform, mainly because of less-torque and less-power motors inside. This feature allows the placement of more sensors, electronic boards and batteries as well as a re-charging and re-powering (when docked) all other platforms. Moreover, the chains provide better grip on ground. The second platform uses a high power brushless motor in combination with a two-stage planetary gearbox to produce enough torque to lift up several docked robots. Furthermore, due to a special design, the robot can work as a hinge or a pivot joint, depending on the required function inside the organism.


(a)

(b)

Fig. 2. ***(a)** Robot docked to a model of a tool. **(b)** Comparative geometry and size of robots and tools.*

B. B. Self-sufficiency of the platform: Power sharing and energy management

The self-sufficiency in self-reconfigurable robots demands dedicated hardware to perform on-board computation, dynamic power management, energy sharing and sensoring tasks. Power management and energy sharing are an essential building block in different scenarios. For instance, such issues as distribution of available on-board energy to peripherals or energy transfer through docking elements among many aggregated robots are very relevant. Fig. 3 shows the hardware architecture which includes power sharing/charging bus, CAN bus, a Li-Po battery charging unit, a power sharing unit, Li-Po battery pack, battery protection unit, a shadow and core processors, different sensors, a localization unit, a radio communication module (ZigBee transceiver), motor drive for locomotion in 3D, motor drives for locomotion in 2D, and docking motor drives for each side.

An important design consideration made in the design of power management and energy sharing mechanism is to get enough control from the hardware so that the software applications can easily emulate different practical scenarios. As shown in Fig. 3, the battery charging unit gives the control to monitor the on-board charging activity, the battery management module provides control to monitor individual cell voltages, current, temperature, relative state of charge, absolute state of charge, remaining capacity, charging voltage, charging current, run time to empty and performs cell balancing, battery over charge/discharge protection and protection against short circuiting. The control from power

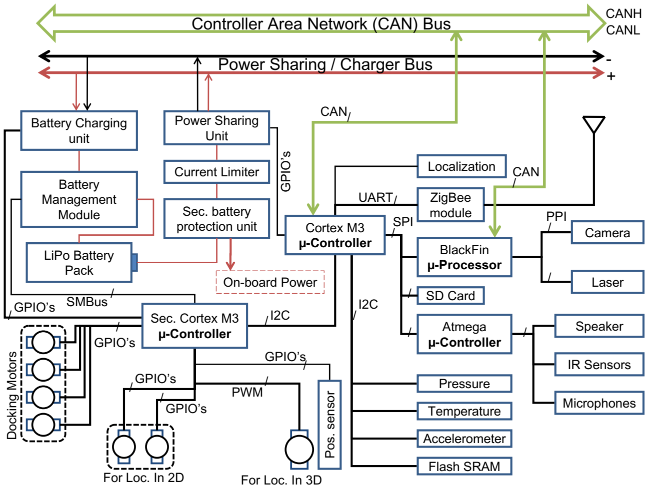


Fig. 3. *Hardware architecture of self-reconfigurable platform.*

sharing unit allows an individual robot to decide whether to share its on-board energy or not with the robots in an organism.

From the software design perspective, an autonomous robot in a stand-alone mode can keep track of its resources, for instance:

- how long it can continue its normal operations with the amount of energy it has on-board;

- how to make an efficient utilization of available energy by maintaining an energy budget of all the on-board peripherals;

- how to prolong the battery life by turning on and off certain devices, when is the time to search for power station (charging unit) to recharge its battery pack.

When docked to other robots in an organism, it can maintain a log of all the energy resources available in an organism, it can decide on its own when and how much energy it can donate to weaker robots in the chain, how much power it should drain from the power/charger bus to recharge its on-board batteries when the organism is docked to a power station.

IV. CAPABILITIES OF ON-LINE AND ON-BOARD EVOLVING

In order to enable on-line and on-board evolution, the platform needs adequately powerful processors, sufficient memory, as well as a communication system. A powerful processor means at the same time high power consumption, for this reason two different main processors are integrated into the robot: *Core* and *Shadow* processors as shown in Fig. 4. The separation is also necessary in order to distribute tasks according to their power, calculation, and memory requirements among different boards. Depending on the intended tasks, boards are able to be switched OFF or ON and thereby allow to reduce power consumption.

The *Core* processor is a Cortex M3 microcontroller based on 32-bit ARM architecture. It has 256 Kbyte of internal SRAM and is able to reach frequencies up to 50 MHz. The main role of this board is to control basic sensorimotor activities of the robot as well as to manage the inter-robot communication. A lot of important sensors (IR, microphones, accelerometer, temperature, humidity etc.) are accessible from this board i.e. the raw data can be directly linked to the processor and accelerates the sensor data processing. For these reasons behavioral, neural and hormone-based controllers as well as Artificial Immune System and sensor fusion tasks are intended to be run here. In order to avoid memory lack which is usually a problem on embedded systems, an external SRAM device of 1Mbit as well as one micro SD card (HD) of a few Gbit are mounted on this board. Genomes, hormones, raw and processed sensor data can be stored here as well as a back copy of obsolete data which can important for evolution and learning tasks.

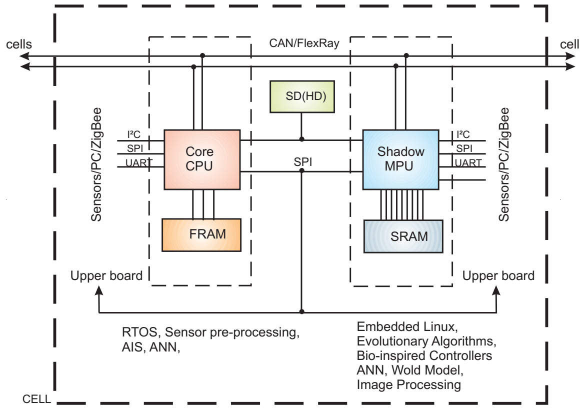


Fig. 4. *Distribution of tasks on two processors in a single cell. Tasks running on Core Processor: Real-Time Operating System (RTOS), Artificial Immune System (AIS), Artificial Neural Network (ANN) and Sensor Pre-Processing. Tasks running on Shadow Processor: tasks with high demand of computational and memory resources like Evolutionary and Bio-inspired controllers, Image Processing etc.*

A fast and stable communication in an organism is a key factor in order to achieve good results during evolutionary process. To speed up the data transfer between cells and hence to allow faster parameter adaptation, sensor data transfer, hormones exchange etc. a multi master wired bus system (CAN) is used for inter-robot communication, see Fig. 5). The data rate of the CAN bus goes up to 1MBit/sec and is extremely stable against disturbances caused by motors, docking elements, as shown in Fig. 1, and other passive components of the PCB.

The external communication with other swarm robots, other organisms or with a basis station is possible since a global wireless communication board (ZigBee) is mounted on each robot and enables together with a UWB-based localization sensor the detection of position in 3D space (described later). For short distances up to 20cm a lot of infrared sensors (IR) are equipped for local communication as well as for collision avoidance.

The second main PCB (*Shadow*) of the robot is equipped

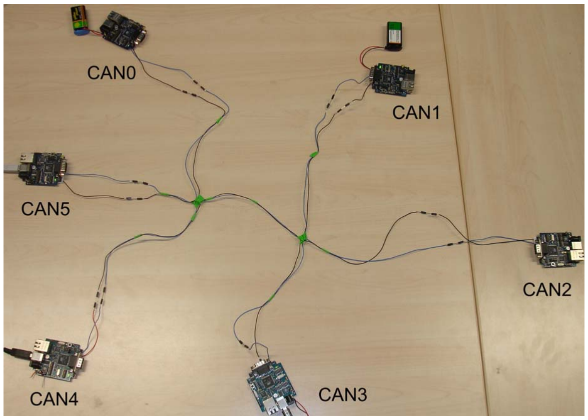


Fig. 5. *Principle of sharing common communication bus (CAN) for different topologies in multi-robot organisms. Demonstrated are physical intersections (corresponding lines of different CAN-bus "branches" are soldered here), which will later represent joints of the organism.*

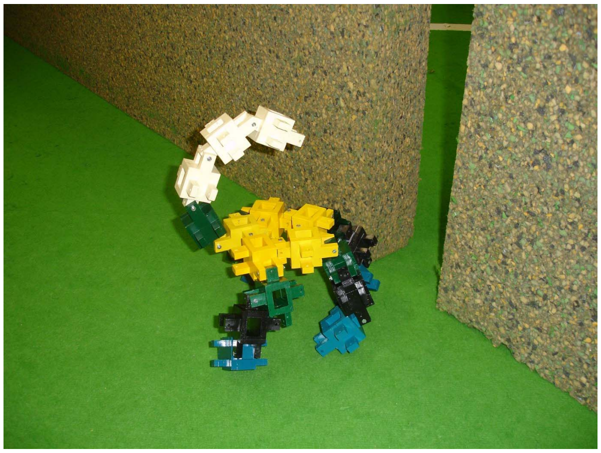


Fig. 6. *The gap in a wall can be detected by local sensors (a topological model is shown). However, no information is available regarding the situation behind the wall. It means that the local fitness of some structural configuration may be very high before the wall, but it can get very low after passing the gap.*

with more powerful processor (Blackfin BF537) runs up to 500MHz and with external 32MByte of SRAM. This board allows to equip robots with a vision system (camera, laser scanner) and gives the opportunity to run middle or long-term evolution tasks. A world model which requires a lot of memory and calculation resources is also intended to run on this board.

Even if applications are running as distributed process on both processors, as well as on processors of different modules within the organism, the exchange of data between the *Core* and the *Shadow* board is managed by using Serial Peripheral Interface (SPI) with a bit rate up to 2 Mbps, see Fig. 3. In addition, since the robots are autonomous, energy consumption (caused by a running software) should be balanced with the available energy on board which should be shared among the robot units, see Fig. 3).

V. CAPABILITIES OF ON-BOARD FITNESS MEASUREMENT

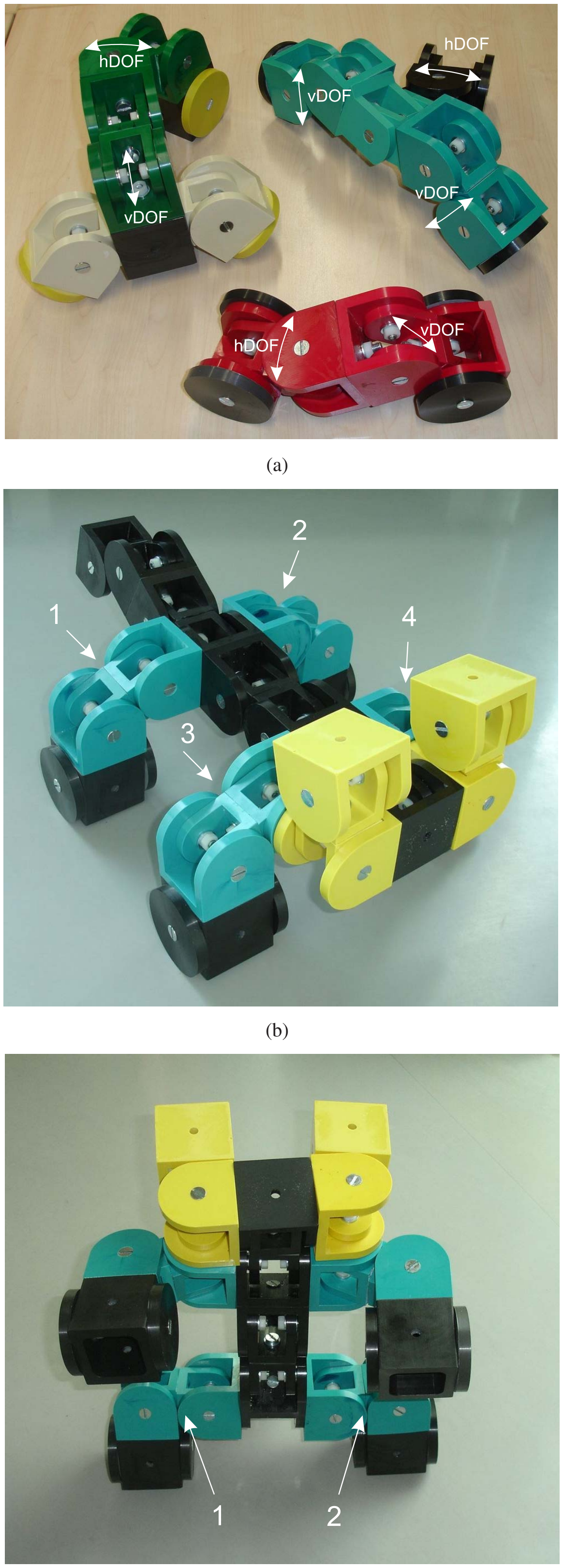
For the application of evolutionary approaches, the platform should provide a measurement of how robots do fit to environment. From a conceptual viewpoint, the following four ways are available to measure the fitness: approximation of a global state by local sensors, perception of local environment by on-board sensors, different measurements during robot-robot interaction, and finally, measurements of internal states. These factors are summarized in Table I.

1. Approximation of a global state by local sensors. For an application of evolutionary strategies the most appropriate feedback may be provided when knowing a global state of the environment, including internal states of other robots. However, such information is not available for individual robots due to practical reasons. Nevertheless, the global state can be approximated when using the world model and several sensor-fusion approaches. Example of global states are map-related values, like explored/unexplored area, coverage of some territory, position of robots in 3D space.

2. Sensing a local environment. Perception of local environment by on-board sensors is the primary way of receiving the information about the environment. However, the problem is that this local information does not reflect the global state. This is well-know problem in robotics, but it has interesting consequence for a fitness function. For instance, the robot shown in Fig. 6 can perceive the wall based on camera/rangeFinder sensors (like IR-based or laser). It can find the gap in the wall and estimate the geometry. This information is needed to decide whether a reconfiguration is needed in order to pass the gap. However, the robot has no information about environment behind the wall. The fitness estimated before the gap will be very high, however it will rapidly drop after passing a wall. In this way, fitness landscape based on the local sensors is not smooth, a gradient-based navigation on such a fitness landscape will be very challenging.

3. Information provided by a robot-robot interaction and communication. Robot-robot interaction is very important source of fitness measurement from the two different viewpoints. First of all, the relationship between structure and function is non-trivial [7]. Robots of the same functionality but of different structures, see Fig. 7, are expected to have a different fitness. One way to estimate fitness of different structurally-shaped organisms is to let them interact with each other and with environment. Another reason for robot-robot interaction in fitness measurement is to consider a Constraint Satisfaction Problem (CSP) for self-reconfiguring robots, where environment plays the role of a limiting factor. In this case CSP solution can be found by a reconfiguration in real environment. Robot-robot communication is important to fuse local information from different robots. This is related not only to environmental values, but also to internal states of robots.

4. Internal states of robot organisms. There are different



(a)

(b)

(c)

Fig. 7. ***(a)** Three topological models of the same functionality (2DOF moving small-size-organisms), but of different structures.. The horizontally placed **(b)** and vertically placed **(c)** organism "Dog". Numbers 1-4 denote active joints (gears+motors).*

Type/Name	Can be used for
Approximation of a global state	
Exploration of a global map	Goodness of behavior
Global coverage	Goodness of behavior
Position in 3D space	Success of kinematic transform.
Local environment	
Gradient of light	Environmental feedback
Gradient of temperature	Environmental feedback
Number of neighbors	Feedback of collaborat. strategies
Number of collisions	Goodness of behavior
Distances to objects	Goodness of behavior
Specific signals in environment	General feedback
Explored area	Goodness of behavior
Robot-Robot	
Internal states of another robot	Feedback of collaborat. strategies
Number of received "eggs"	Feedback of collaborat. strategies
Trophallaxis exchange	Feedback of collaborat. strategies
Internal	
Energy Level	Individual fitness/activities
Distribution of energy	Individual fitness
Number of internal failures	Individual fitness

TABLE I

Different types of on-board fitness measurement.

internal sources of information: energy-based, mechanical, load on buses, number of internal failures (e.g. in CAN bus), CPU/Memory usage and other. The energy-based values, discussed in Sec. III, are very useful for many purposes, e.g. in estimation of the most efficient structure of organisms depending on the locomotion principle. For example, the energy consumption in active joints of the topological models shown in Fig. 7 depends not only on the position of organisms, but also on the locomotion principle. The balanced organism in Fig. 7(c) is expected to consume less energy with a moving locomotion, however the organism Fig. 7(b) is expected to consume less energy with a legged locomotion. Generally, the number of internal sensors can be very high, however only a specific combination of them can provide enough information to make decisions. It is expected to use artificial immune network to monitor data from internal sensors as well as to use them for control purposes. To give a reader an overview about sensing capabilities of the platform, we collect in Table II a brief description of on-board sensors.

A. Spatial Awareness in Fitness Estimation

The spatial awareness [13] of the robot swarm and robot organism is a very important issue in fitness estimation as it defines interactions towards the environment and internal decisions as well. The consortium develops a new concept for spatial awareness which involves two key components: accurate ultra-wideband (UWB) 3D localization and wireless sensor network based on ZigBee and FRID technologies.

Sensor	Name	Interface
Environmental		
Light	ADPS9002	analog
Air Pressure	SCP1000	I2C
Directional Sound	SPM0208HD5	analog
Humidity/Temper.	SHT15	I2C
IR-reflective	TCRT1000	analog
Imaging Sensor	OV7660FSL	PPI
Laser (in the Range Finder)	LS-1-650	digital
RFID sensor	Lux	no
Sonar sensor	SRF08(or 10)	I2C
Laser RangeFinder	URG-04LX	RS232/USB
Detecting motion in environment	AMN34111	analog
Hall effect (magnetic)	US4881EUA	analog
Locomotion		
3D Acceleration	LIS3L02AL	I2C
WTL laser mouse	ADNS-7530	SPI
3D Localization	Ubisense	digital
Orientation-Sensor	SFH 7710	SPI
IR-docking sensor	IR-based	analog
Force measurement sensor	tbd	analog
Joint angle sensor	2SA-10-LPCC	analog
Internal, Indirect Sensors		
Voltage, Current	BQ29312/bq2084	SMBus
Bus Load Sensor	no	software
Center of mass	no	software
Energy-docking sen.	no	software

TABLE II

On-board sensors of artificial organisms.

A ZigBee [14] network has been used for transmitting and exchanging data among the single robotic cells. As depicted in Fig. 8, the robotic modules are equipped with UWB localization tags providing very accurate 3D indoor positioning. The position of each robot has been achieved via the Angle of Arrival (AoA) and Time Difference of Arrival (TDoA) information derived from each ultra-wideband transceiver situated in the corners of the room [15]. A location platform monitoring station connected to the network, computes and transmits the localization data over TCP/IP to the Zigbee coordinator (connected to the same network), which forwards these geometrical coordinates to each robotic cell.

The technology of spatial awareness can be used in a wide spectra of learning and evolving approaches. Each individual robot and well as the whole robot organism know moving trajectories of all its parts/robots. This can be used, for example, in generating a feedback/reward in learning algorithms.

VI. HARDWARE SIMULATOR

The well-know problem in large projects is that a software development goes behind a hardware development. Software partners have to wait until some hardware version become available in order to start programming – it wastes resources of projects. This problem was solved in the following way: with very early specifications of the targeted processors and basic hardware architecture, evaluation boards with the Cortex MCU are ordered for software partners. These Cortex boards can communicate with PC via USB and with each other via CAN. The software simulators, running on the PC side, simulate the robots, the objects and the sensors in a 2D (for Player/Stage [16]) or 3D (for Delta3D) environment. A controller for each simulated robots is situated in the Cortex evaluation board. It reads sensor data from the simulated environment and sends actuator commands to the virtual robots. The structure of such a hardware simulator is shown in Fig. 9(a), images of a screenshot and connected boards are shown in Fig. 9(b,c).

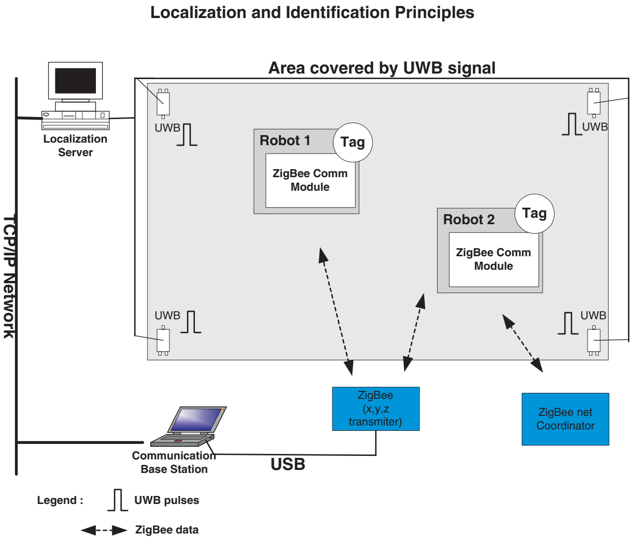


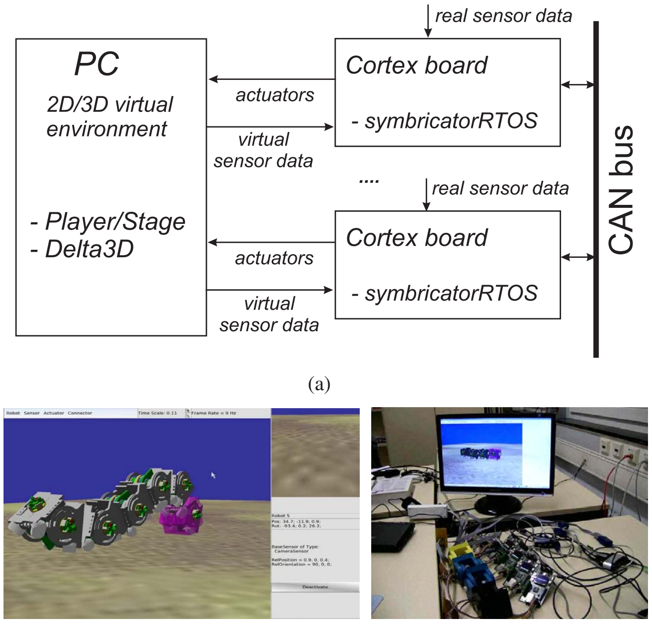
Fig. 8. *Localization principle of the robot swarm and artificial organism.*

Meanwhile, many different real sensors can be connected to the Cortex evaluation boards to provides extra information for the controllers, see Fig. 10. In this way a software development can be started even on very early stage, later on the software can be transferred without serious modifications to the robot platform. The idea of mixture of virtual robots with real sensors also provides the hardware partners a fast way to test their prototyped sensors.

In connections with real sensors, the hardware simulator provides several advantages for the software/evolutionary community. It is enough to order boards (cost about 70 euros each from different manufacturers) and to install a open-source virtual environment to have a possibility to experiment with on-board/on-line evolutionary experiments. Moreover, the community can download and run software developed by consortium, and vice versa, we can run on real self-reconfigurable robots the algorithms developed by community. In this way everybody can take part in the real-world evolutionary experiments. Details of the hardware simulator are described on www.symbrion.eu.

VII. CONCLUSION

In this paper we presented the current development of the reconfigurable robotic platform which is capable of



(a)

(b)

(c)

Fig. 9. ***(a)** Structure of the hardware simulator. **(b)** Virtual organism in Delta3D visualization. **(c)** Cortex boards connected to PC with running Delta3D. Images (b,c) are taken by L. Winkler, University of Karlsruhe.*

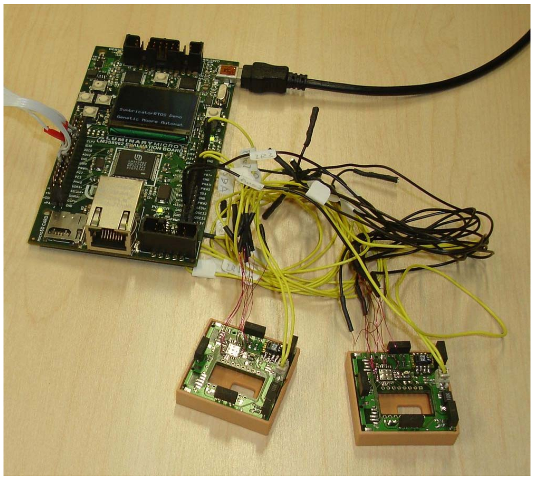


Fig. 10. *Different sensors (on two sensor boards), connected to the Cortex board with running symbricatorRTOS.*

working as independent robot swarm as well as aggregated organisms. We have indicated three key capabilities of the platform: autonomous morphogenesis, performing on-line and on-board evolving approaches and on-board fitness measurement. Finally, we showed the hardware simulator, which can help everybody to participate in evolutionary experiments performed in consortiums.

The main focus of this paper lies on advantages and new concepts which this development provides to software and evolutionary communities. We think about a new kind of "symbiotic experiments", which are a combination of swarm and reconfigurable robotics. In these experiments not only a collective/swarm behavior can be investigated, but also we can explore a relationship between structures and functions, between collective behavior of autonomous individuals and collective behavior of symbiotic co-individuals. More generally, we think about new morphogenetic approaches performed in real systems, which can contribute to our understanding of the phenomenon of artificial evolution.

ACKNOWLEDGEMENT

The "SYMBRION" project is funded by the European Commission within the work programme "Future and Emergent Technologies Proactive" under the grant agreement no. 216342. The "REPLICATOR" project is funded within the work programme "Cognitive Systems, Interaction, Robotics" under the grant agreement no. 216240. Additionally, we want to thank all members of projects for fruitful discussions.

REFERENCES

[1] Dario Floreano, Phil Husbands, and Stefano Nolfi. Evolutionary Robotics. In *Handbook of Robotics*. Springer Verlag, Berlin, 2008.

[2] S. Kornienko, O. Kornienko, A. Nagarathinam, and P. Levi. From real robot swarm to evolutionary multi-robot organism. In *Proc. of the CEC2007, Singapure*, pages 1483–1490, 2007.

[3] E. Tuci, C. Ampatzis, and M. Dorigo. Evolving neural mechanisms for an iterated discrimination task: a robot based model. In *Proc. of the Eighth European Conference on Artificial Life (ECAL 2005)*, pages 231–240, Septemeber 2005.

[4] D. Floreanoand M. Suzukiand C. Mattiussi. Active vision and receptive field development in evolutionary robots. *Evolutionary Computation*, 13(4):527–544, 2005.

[5] L. Koenig, K. Jebens, S. Kernbach, and P. Levi. Stability of on-line and on-board evolving of adaptive collective behavior. In *Proc. of the EUROS 2008, Prague, Czech Republic*, 2007.

[6] Ch. Schwarzer. *Investigation of Evolutionary Reproduction in a Robot Swarm*. Master Thesis, University of Stuttgart, Germany, 2008.

[7] Serge Kernbach. *Structural Self-organization in Multi-Agents and Multi-Robotic Systems*. Logos Verlag, Berlin, 2008.

[8] L. Koenig. *A model for developing behavioral patterns on multi-robot organisms using concepts of natural evolution*. Master Thesis, University of Stuttgart, Germany, 2007.

[9] SYMBRION. *SYMBRION: Symbiotic Evolutionary Robot Organisms, 7th Framework Programme Project No FP7-ICT-2007.8.2*. European Communities, 2008-2012.

[10] REPLICATOR. *REPLICATOR: Robotic Evolutionary Self-Programming and Self-Assembling Organisms, 7th Framework Programme Project No FP7-ICT-2007.2.1*. European Communities, 2008-2012.

[11] D. McFarland and E. Spier. Basic cycles, utility and opportunism in self-sufficient robots. In *Robotics and Autonomous Systems*, pages 179–190, 1997.

[12] S. Kernbach, L. Ricotti, J. Liedke, P. Corradi, and M. Rothermel. Study of macroscopic morphological features of sambiotic robotic organisms. In *Proceedings of the workshop on self-reconfigurable robots, IROS08, Nice*, pages 18–25, 2008.

[13] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. Stability of on-line and on-board evolving of adaptive collective behavior. In *Proceedings of the* 5^{th} *annual ACM/IEEE international conference on Mobile computing and networking*, pages 59–68, 1997.

[14] Fred Eady. *Hands-On ZigBee: Implementing 802.15.4 with Microcontrollers*. Elsevier, Burlington, USA, 1 edition, 2007.

[15] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personnel Communications Magazine*, pages 42–47, 1997.

[16] Brian P. Gerkey, Richard T. Vaughan, and Andrew Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *In Proceedings of the 11th International Conference on Advanced Robotics*, pages 317–323, 2003.