

Cognitive Systems Platforms using Open Source

Patrick Courtney(1), Olivier Michel(2), Angelo Cangelosi(3), Vadim Tikhanoff(3),
Giorgio Metta(4), Lorenzo Natale (5), Francesco Nori (5) and Serge Kernbach(6)

(1) Perkinelmer, Beaconsfield UK (2) Cyberbotics Ltd. Lausanne, Switzerland (3) Adaptive Behaviour & Cognition Group, University of Plymouth, UK (4) Italian Institute of Technology, & University of Genoa, Italy (5) Italian Institute of Technology (6) University of Stuttgart, Germany

ABSTRACT

This chapter reports to the development of the tools and methodologies that are in development within the EU, with an emphasis on the Open Source approaches with a view to performance analysis and comparison, and to provide an overview of cooperative research and especially on the use of Open platforms.

1 Introduction

A significant research programme under the banner of “cognitive systems” is now underway within the European Union with some €400M committed in the period to 2010. This programme focuses on developing the technology and the necessary scientific understanding to provide new levels of autonomy and decision making into computer-based systems. Active research approaches in the area range broadly, from traditional rule-based AI, through to connectionist, dynamical and emergent systems and include embodied systems combining computing and robotic systems. A major practical motivation for the development of cognitive systems is to overcome the problems faced by traditional computer systems in dealing robustly with the uncertainties and changing demands that characterise the real world. Potential applications cited span a very wide range and have included care-giver robots, and easier-to-use interfaces.

This chapter reports to the development of the tools and methodologies that are in development within the EU. One major approach that has emerged across the projects has been in the use of Open Source platforms in order to share experiences and run larger scale experiments. This clearly also has a role to play in facilitating performance analysis and comparison. Here we present 3 significant research efforts in cognitive robotics with varying degrees of development with respect to benchmarking¹:

¹ For a full list of past and current EU funded projects in Cognitive Systems and Robotics research, see http://cordis.europa.eu/fp7/ict/programme/challenge2_en.html

The Rats Life Benchmark

Rat's Life is a complete cognitive robotics benchmark that was carefully designed to be easily reproducible in a research lab with limited resources. It relies on two e-puck robots, some LEGO bricks and the Webots robot simulation software. This benchmark is a survival game where two robots compete against each other for resources in a previously unseen maze. Like the rats in cognitive animal experimentation, the e-puck robots look for feeders which allow them to live longer than their opponent. Once a feeder is reached by a robot, the robot draws energy from it and the feeder becomes unavailable for a while. Hence, the robot has to further explore the maze, searching for other feeders.

The iCub platform

The iCub is a new Open Source humanoid robot developed for research in embodied cognition. At around the size of a three and half year old child, it can crawl on all fours and sit up to manipulate objects. Its hands have been designed to support sophisticated manipulation skills. The iCub is distributed as Open Source following the GPL/FDL licenses. The entire design is available for download from the project homepage and repository². In the following, we concentrate on the description of the hardware and software systems, as well as an iCub simulator, each of which may be used in comparative studies.

The swarm platform of the Replicator and Symbrion projects

Cooperation and competition among stand-alone swarm agents can increase the collective fitness of the whole system. An interesting form of collective system is demonstrated by some bacteria and fungi, which can build symbiotic organisms. Symbiotic communities can enable new functional capabilities which allow all members to survive better in their environment. Here we present an overview of two large European projects dealing with new collective robotic systems which utilize principles derived from natural symbiosis. We also describe some of the typical hardware, software and methodological challenges arising, as well as prototypes and on-going experiments. The use of multiple robotic agents lends itself to the replication of experiments and thus benchmarking of behavior and functionality.

2 The Rat's Life Benchmark: Competing Cognitive Robots

2.1 Motivation

Most scientific publications in the area of robotics research face tremendous challenges: comparing the achieved result with other similar research results and hence convincing the reader of the quality of the research work. These challenges are very

² <http://www.robotcub.org>

difficult because roboticists lack common tools allowing them to evaluate the absolute performance of their systems or compare their results with others. As a result, such publications often fail at providing verifiable results, either because the studied system is unique and difficult to replicate or they don't provide enough experimental details so that the reader could replicate the system accurately.

This fact is unfortunately impairing the credibility of robotics research. A number of robotics researchers proposed to develop series of benchmarks to provide a means of evaluation and comparison of robotics research results [1, 3, 4, 9, 49, 50, 51]³.

2.2 Existing Robot Competitions and Benchmarks

Several popular robot competitions are organized on a regular basis, usually once a year. The Robocup soccer [6] is a robot soccer tournament with several categories (small size league, middle size league, standard platform league, simulation league, etc.). The Robocup Rescue is based on the Urban Search And Rescue (USAR) benchmark developed by the NIST [7] where robots have to search and rescue the victims of a disaster in a urban environment. MicroMouse⁴ involves wheeled robots solving a maze. The AAAI Robot Competition⁵ proposes different scenarios each year during the AAAI conference, but often lack clear performance metrics. The DARPA Grand Challenge and Urban Challenge⁶ and the European Land-Robot Trial⁷ focus on unmanned ground and sometimes aerial vehicles racing against each other.

Such competitions are useful as they can provide elements of comparison between different research results. However one of the major problem is that the rules often change across the different editions of the same competition. Hence it is difficult to compare the progress achieved over time. Also these competitions are very specific to particular problems, like Robocup is focused mostly on robot soccer and has arguably a limited interest for cognitive robotics [5].

Among all the benchmarks we reviewed which are mostly robot competitions, none of them provides both stable rules with advanced cognitive robotics challenges and an easy setup. In this paper, we propose a new robotics benchmark called "Rat's Life" that addresses a number of cognitive robotics challenges while being cheap and very

³ EURON Benchmarking. www.euron.org/activities/benchmarks/index.html
Robot Benchmark Euron website www.robot.uji.es/EURON/en/euron07.htm
NIST ISD website <http://www.isd.mel.nist.gov/>
Euron GEM SIG website <http://www.heronrobots.com/EuronGEMSig/>
Rawseeds web site <http://www.rawseeds.org/>
RoSta website <http://www.robot-standards.eu/>

⁴ Micromouse contest. <http://www.micromouseinfo.com>

⁵ AAAI Robot Competition and exhibition. <http://palantir.cs.colby.edu/aaai07>

⁶ DARPA Grand Challenge. <http://www.darpa.mil/grandchallenge>

⁷ European Land-Robot Trial ELROB. <http://www.elrob.org/>

easy to setup for any research lab. The aim of this benchmark is to foster advanced robotics and AI research.

2.3 Rat's Life Benchmark: Standard Components

The e-puck mobile robot

The e-puck mini mobile robot was originally developed at the Swiss Federal Institute of Technology in Lausanne (EPFL) for teaching purposes by the designers of the successful Khepera robot. The e-puck hardware and software is fully Open Source, providing low level access to every electronic device and offering unlimited extension possibilities. The robot is already equipped with a large number of sensors and actuators (figure 1). It is well supported by the Webots simulation software with simulation models, remote control and cross-compilation facilities. The robot is commercially available from Cyberbotics⁸ for about EUR 570.

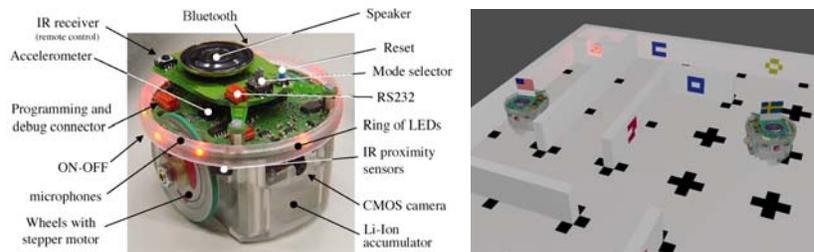


Figure 1: The e-puck robot **Figure 2: The Rat's Life maze: LEGO bricks, e-puck robots and feeder device**

LEGO bricks

The LEGO bricks are used to create an environment for the e-puck robot. This environment is actually a maze which contains "feeder" devices (see next sections) as well as visual landmarks made up of patterns of colored LEGO brick in the walls of the maze (see figure 2).

The Webots robot simulation software

Webots [8] is a commercial software package for fast prototyping and simulation of mobile robots. It was originally developed at EPFL from 1996 and has been continuously developed, documented and supported since 1998 by Cyberbotics Ltd. Over 500 universities and industrial research centers worldwide are using this software for research and educational purposes. Webots has already been used to organize robot programming contests (ALife contest and Roboka contest). Although Webots is a commercial software, a demo version is available from Cyberbotics's web site and includes the complete Rat's Life benchmark usable for free.

⁸ Cyberbotics Ltd. <http://www.cyberbotics.com>

2.4 Rat's Life Benchmark Description

This section does not claim to be a technical reference for the Rat's Life benchmark. Such a technical reference is available on the Rat's Life web site⁹.

Software-only Benchmark

The Rat's Life benchmark defines precisely all of the hardware necessary to run the benchmark (including the robots and their environment). Hence the users of the benchmarks don't have to develop any hardware. Instead, they can focus on robot control software development only. This is similar to the Robocup standard league where the robot platforms (Aibo robots) and the environment is fully defined and the competitors are limited to develop control software only. This has the disadvantage of preventing hardware research and is constraining the contest to the defined hardware only. However, it has the great advantage of letting the users focus on the most challenging part of cognitive robotics, i.e., the control software.

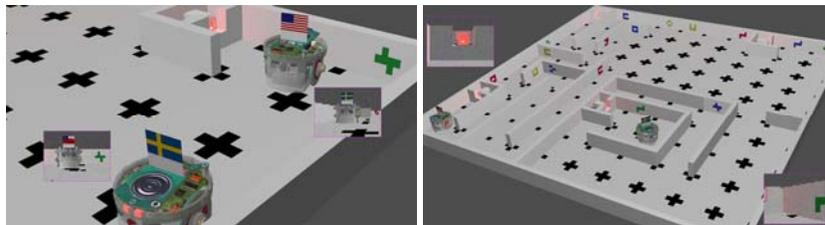


Figure 3: Closeup of the Rat's Life simulated robots in Webots (left) and general overview (right)

Configuration of the Maze

For each evaluation, the maze is randomly chosen among a series of 10 different configurations of the maze. In each configuration, the walls, landmarks and feeder are placed at different locations to form a different maze. Each configuration also has 10 different possible initial positions and orientations for the two robots, one of which is chosen at random. This makes 100 possible initial configurations. This random configuration of the maze prevents the robots from having prior knowledge of the maze, and forces them to discover their environment by exploring it. This yields to much more interesting robot behaviors. A possible configuration is depicted on fig 3.

Virtual Ecosystem

The Rat's Life benchmark is a competition where two e-puck robots compete against each other for resources in a LEGO maze. Resources are actually a simulation of energy sources implemented as four feeder devices. These feeder devices are depicted on figure 4. They are made up of LEGO NXT distance sensors which are controlled by a LEGO NXT control brick. They display a red light when they are full of virtual energy. The e-puck robots can see this colored light through their camera and have to move forward to enter the detection area of the distance sensor. Once the sensor de-

⁹ Rat's Life contest. <http://www.ratslife.org>

tests the robot, it turns its light off to simulate the fact that the feeder is now empty. Then, the robot is credited an amount of virtual energy corresponding to the virtual energy that was stored in the feeder. This virtual energy will be consumed as the robot is functioning and could be interpreted as the metabolism of the rat robot. The feeder will remain empty (i.e., off) for a while. Hence the robot has to find another feeder with a red light on to get more energy before its energy level reaches 0. When a robot runs out of virtual energy (i.e., its energy level reaches 0), the other robot wins.

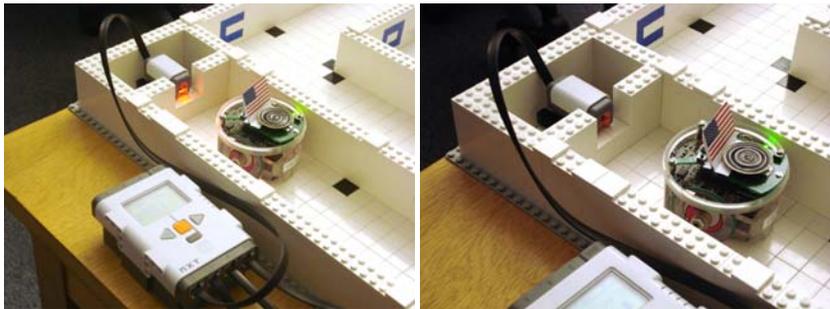


Figure 4: A full feeder facing an e-puck robot (left) and an empty one (right)

Robotics and AI Challenges

Solving this benchmark in an efficient way requires the following cognitive capabilities:

- Recognize a feeder (especially a full one) from a camera image.
- Navigate to the feeder and dock to it to grab energy.
- Navigate randomly in the maze while avoiding getting stuck.
- Remember the path to a previously found feeder and get back to it.
- Optimize energy management.
- Try to prevent the other robot from getting energy.

2.5 Evolution of the Competition over Time

Observing the evolution of the competition over days was very interesting and we decided to store all the simulation movies in a data base to be able to analyze this evolution afterwards. The movie database contains more than 2500 movies (totaling more than 50 GB of data) and is freely available online¹⁰. During the contest, several major performance breakthroughs could be observed simply by analyzing the behavior of the robots in the simulation movies. One could identify five major break-

¹⁰ <http://www.cyberbotics.com/ratslife/movies>

throughs which happened chronologically one after the other, bringing each time an improved performance:

Random Walkers The random walkers came actually from the very first version of the sample source code included with the contest software development kit, made available to all the competitors. This simple control algorithm similar to Braitenberg vehicles [2] let the robots move randomly while avoiding the obstacles. By chance some of them met a feeder from time to time, but this behavior is very inefficient and rely mostly on luck. Also, this very first version was not very efficient at navigating and often caused the robot to get stuck in some unexpected situations, like facing a corner.

Vision-Enabled Random Walkers The so called vision-enabled random walkers are an improved version of the original random walker making an extensive use of vision to recognize the feeders and adjust the trajectory of the robot to reach the feeder instead of simply moving randomly. This results in slightly more efficient robots who won't pass in front of a feeder without getting energy from it. A vision-enabled random walker is included in the sample code currently distributed to the competitors. This sample version has however been largely improved by different competitors over time.

Right Hand Explorers One of the problems with the random walkers is that a Braitenberg vehicle behavior is not very efficient at exploring extensively a maze and hence at finding the feeders. Maze exploration algorithms exist and are much more efficient. The right hand algorithm is one of the simplest and best known maze exploration algorithms. It consists in simply following the first wall found on the right hand side of the robot (this also works with the left hand side of course). Using this algorithm combined with some vision to reach efficiently the feeders, a significant performance breakthrough was reached. The first right hand explorer appeared on February 22nd, with a robot named Tony (which reached rank #1 of the hall of fame on February 22th very rapidly) and was rapidly copied by many other competitors as this behavior is both easy to understand and to re-program.

Energy-aware robots Getting energy from the feeder as soon as you find the feeder is nice, but there is an even better strategy: Once a robot finds a feeder, it can simply stop and sit in front of the feeder, thus preventing the other robot from reaching this feeder. In the meantime the robot sitting in front of the feeder should watch its energy level and decide to move to the feeder once its energy level reached a very low value, just enough to make that move to the feeder and refuel. During this waiting time, the other robot may be struggling to find a feeder and possibly loose the game if it runs out of energy. This kind of energy-aware robots appeared on February 28th, with a robot named Ratchou (which reached rank #1 thanks to this breakthrough). Similarly to the right hand explorer, it was rapidly copied by other competitors as it was easy to understand and to re-program.

SLAMers SLAM stands for Self Localization And Mapping. Compared to other techniques mentioned above, it involves a much more complicated algorithm and requires efficient image processing. SLAMer robots actually seems to use the right hand algorithm on a first stage to explore extensively the maze, but they build dynam-

ically a map of this maze while exploring it and eventually don't use the right hand algorithm at all. Their internal representation of the environment contains the walls, the feeders and likely the landmarks. This map is then used by the robot to get back to previously found feeders. It turned out to be very efficient and clearly outperformed the simpler reactive controllers. The first SLAMer robot is Ratatouille who implemented a first version of visual SLAM-based navigation on April 6th and reach rank #1. This first version was however probably not well tuned (or somehow buggy) and it happened to lose in rare occasions against lucky and efficient right-hand explorers. However, the author of Ratatouille continued to improve the performance of his SLAMer robot and finally sat steadily on the very top of the hall of fame for more than two months. The other competitors, including Tony among others, tried hard to implement such an efficient SLAMbased navigation controller, they were not very successful until June 5th. At this point a competitor with a robot controller named Gollum developed a pretty efficient SLAMer robot able to challenge Ratatouille. Golum reached rank #2 on June 5th and had a fierce and very interesting match against Ratatouille, but was not successful.

Super-SLAMers The author of Ratatouille actually never stopped from April 6th to improve his SLAM-based robot controller. A major improvement was probably the estimation of the status of the feeders, combined with an estimation of the time needed to travel the maze to reach the feeder. From the most recent simulation movies, Ratatouille seems to be able to anticipate that a mapped feeder will become available again: when the feeder is still red, Ratatouille starts to navigate towards this feeder and about one second before it reaches the feeder, the feeder becomes green again. This makes Ratatouille the most efficient robot controller currently on the Rat's Life benchmark. At this point, it is difficult to imagine a better behavior than the one exhibited by Ratatouille.

2.6 Discussion

Thanks to the Rat's Life benchmark, it becomes possible to evaluate the performance of various approaches to robot control for navigation in an unknown environment, including various SLAM and bio-inspired models. The performance evaluation allow us to make a ranking between the different control programs submitted, but also to compare the progresses achieved over a short period of time of research on this problem. However, this period of time could be extended and we could, for example, compare the top 5 controller programs developed in 2008 to the top 5 controller programs developed in 2012 to evaluate how much the state of the art progressed.

The control program resulting from the best robot controllers could be adapted to real world robotics applications in the areas of surveillance, mobile manipulators, UAV, cleaning, toys, etc. Also, interesting scientific comparisons with biological intelligence could be drawn by opposing the best robot controllers to a real rat (or a rat-controlled robot) in a similar problem. Similarly, we could also pit the best robot controllers against a human (possibly a child) remote controlling the robot with a

joystick and with limited sensory information coming only from the robot sensors (mainly the camera).

We hope that this initiative is a step towards a more general usage of benchmarks in robotics research. By its modest requirements, simplicity, but nevertheless interesting challenges it proposes, the Rat's Life benchmark has the potential to become a successful reference benchmark in cognitive robotics and hence open the doors to more complex and advanced series of cognitive robotics benchmarks.

3 The Open Source humanoid robot platform iCub

The RobotCub project has the two-fold goal of: i) creating an Open hardware/software humanoid robotic platform for research in embodied cognition, and ii) advancing our understanding of natural and artificial cognitive systems by exploiting this platform in the study of the development of cognitive capabilities.

The RobotCub stance on cognition posits that manipulation plays a fundamental role in the development of cognitive capability [10-13]. As many of these basic skills are not ready-made at birth, but developed during ontogenesis [14], RobotCub aims at testing and developing this paradigm through the creation of a child-like humanoid robot: the iCub. The small, compact size and high number of degrees of freedom combined with the Open Source approach distinguishes RobotCub from other humanoid robotics projects developed worldwide. The iCub is also an attempt at standardization and community-building, since several copies of the robot are being built and distributed to research laboratories in Europe and the USA. Together with Open Source licensing, this allows leveraging on the work of others, and possibly benchmarking. Software for the iCub developed at one site can be effectively tested and improved by other scientists.

On one hand this makes software development harder; it is expected that software modules will be tested in different environments and by researchers who will try to layer additional behaviors on top of them. Minimal standards and quality are required to guarantee interoperability. On the other hand, this initial difficulty can be the seed of more reliable and effective software for cognitive systems; systems that have to run multiple times instead of once.

3.1 The iCub

The iCub has been designed to allow manipulation and mobility. For this reason 38 degrees of freedom (DOF) have been allocated to the upper part of the body (including the waist). The hands, in particular, have 9 DOF each with three independent fingers and the fourth and fifth to be used for additional stability and support (only

one DOF). They are tendon driven, with most of the motors located in the forearm. The legs have 6 DOF each and are strong enough to allow bipedal locomotion.

From the sensory point of view, the iCub is equipped with digital cameras, gyroscopes and accelerometers, microphones, and force/torque sensors. A distributed sensorized skin is under development using capacitive sensor technology.

Each joint is instrumented with positional sensors, in most cases using absolute position encoders. A set of DSP-based control cards, designed to fit the small size of the iCub, takes care of the low-level control loop in real-time. The DSPs talk to each other via CAN bus. Four CAN bus lines connect the various segments of the robot.

All sensory and motor-state information is transferred to an embedded Pentium based PC104 card that handles acquisition, synchronization and reformatting of the various data streams. Time consuming computation is typically carried out externally on a cluster of machines. The communication with the robot occurs via a Gbit Ethernet connection.

The overall weight of the iCub is 22kg. The umbilical cord contains both the Ethernet cable and power to the robot. At this stage there is no plan for making the iCub fully autonomous in terms of power supply and computation (e.g. by including batteries and/or additional processing power on board).

The mechanics and electronics were optimized for size, starting from an evaluation and estimation of torques in the most demanding situations (e.g. crawling). Motors and gears were appropriately sized according to the requirements of a set of typical tasks. The kinematics was also defined following similar criteria. The controller cards were designed to fit the available space. Figure 5 shows the prototype of the iCub.

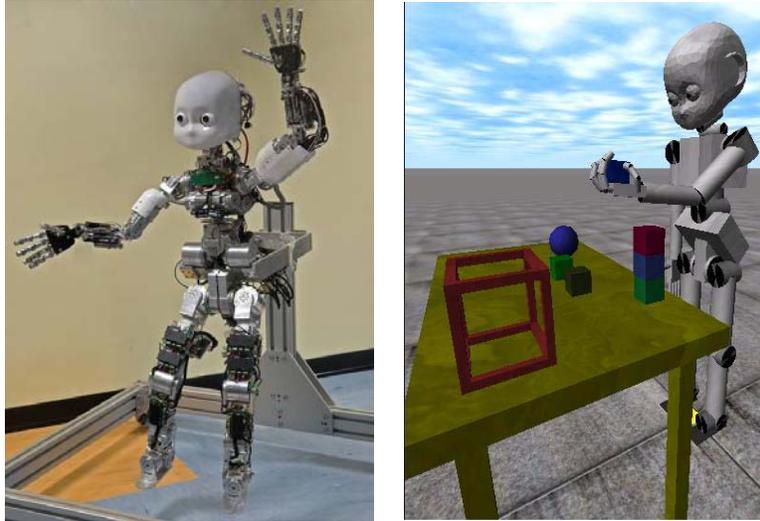


Figure 5: the complete prototype of the iCub (left) and simulated iCub looking at and manipulating an object in its environment (right)

Several copies of the iCub are now available in the EU. These have been developed as part of the RobotCub competitive call, which awarded seven iCubs free of charge to the six best project proposals. Many more copies of the iCub are also being constructed for other EU funded projects.

3.2 Mechanics

The kinematic specifications of the body of the iCub, the definition of the number of DOF, their actual locations as well as the actual size of the limbs and torso were based on ergonomic data and human X-ray images.

The possibility of achieving certain motor tasks is favored by a suitable kinematics and, in particular, this translates into the determination of the range of movement and the number of controllable joints (where clearly replicating the human body in detail is impossible with current technology). Kinematics is also influenced by the overall size of the robot which was imposed *a priori*. The size is that of a 3.5 years old child (approximately 100cm tall). This size can be achieved with current technology. The Sony QRIO is an example of a robot of an even smaller size although with less degrees of freedom. In particular, our task specifications, especially manipulation, require at least the same kinematics of QRIO with the addition of the hands and moving eyes. Also, we considered the workspace and dexterity of the arms and thus a three degree of freedom shoulder was included. This was elaborated into a proper list of joints, ranges, and sensory requirements.

Considering dynamics, the most demanding requirements appear in the interaction with the environment. Impact forces, for instance, have to be considered for locomotion behaviors, but also and more importantly, developing cognitive behaviors such as manipulation might require exploring the environment erratically. As a consequence, it is likely that high impact forces need to be sustained by the robot mechanical structure. This requires strong joints, gearboxes, and more in general powerful actuators and appropriate elasticity (for absorbing impacts). In order to evaluate the range of the required forces and stiffness, various behaviors were simulated in a dynamical model of the robot. These simulations provided the initial data for the design of the robot. The simulations were run using Webots and were later cross-checked by conventional static analysis.

At a more general level, we evaluated the available technology, compared to the experience within the project Consortium and the targeted size of the robot: it was decided that electric motors were the most suitable technology for the iCub, given also that it had to be ready according to the very tight schedule of the overall project. Other technologies (e.g. hydraulic, pneumatic) were left for a “technology watch” activity and were not considered further for the design of the iCub.

From the kinematic and dynamic analysis, the total number of degrees of freedom for the upper body was set to 38 (7 for each arm, 9 for each hand, and 6 for the head). For the legs the simulations indicated that for crawling, sitting and squatting a 5 DOF leg is adequate. However, it was decided to incorporate an additional DOF at the ankle to support standing and walking. Therefore each leg has 6 DOF: these include 3 DOF at the hip, 1 DOF at the knee and 2 DOF at the ankle (flexion/extension and abduction/adduction). The foot twist rotation was not implemented. Crawling simulation analysis also showed that for effective crawling a 2 DOF waist/torso is adequate. However, to support manipulation a 3 DOF waist was incorporated. A 3 DOF waist provides increased range and flexibility of motion for the upper body resulting in a larger workspace for manipulation (e.g. when sitting).

The neck has a total of 3 DOF and provides full head movement. The eyes have further 3 DOF to support both tracking and vergence behaviors.

The actuation solution adopted for the iCub is based on a combination of a harmonic drive reduction system (CSD series, 100:1 ratio for all the major joints) and a brushless frameless motor (BLM) from the Kollmorgen frameless RBE series. The harmonic drive gears provide zero backlash, high reduction ratios on small space with low weight while the brushless motors exhibit the desired properties of robustness, high power density, and high torque and speed bandwidths (especially when compared with conventional DC brushed motors). The use of frameless motors permits integration of the motor and gears in an endoskeletal structure that minimizes size, weight and dimensions. Smaller motors (brushed-DC type) were used for the hands and head joints.

3.3 The software: YARP

The iCub software was developed on top of Yarp [15]. RobotCub supported a major overhaul of the Yarp libraries to adapt to a more demanding collaborative environment. Better engineered software and interface definitions are now available in Yarp.

Yarp is a set of libraries that support modularity by abstracting two common difficulties in robotics: namely, modularity in algorithms and in interfacing with the hardware. Robotics is perhaps one of the most demanding application environments for software recycling where hardware changes often, different specialized OSs are typically encountered in a context with a strong demand for efficiency. The Yarp libraries assume that an appropriate real-time layer is in charge of the low-level control of the robot and instead takes care of defining a soft real-time communication layer and hardware interface that is suited for cluster computation.

Yarp takes care also of providing independence from the operating system and the development environment. The main tools in this respect are ACE [16] and CMake¹¹. The former is an OS-independent communication library that hides the quirks of interprocess communication across different OSs. CMake is a cross-platform make-like description language and tool to generate appropriate platform specific project files.

Yarp abstractions are defined in terms of protocols. The main Yarp protocol addresses inter-process communication issues. The abstraction is implemented by the port C++ class. Ports follow the observer pattern by decoupling producers and consumers. They can deliver messages of any size, across a network using a number of underlying protocols (including shared memory when possible). In doing so, ports decouple as much as possible (as function of a certain number of user-defined parameters) the behavior of the two sides of the communication channels. Ports can be commanded at run time to connect and disconnect.

The second abstraction of Yarp is about hardware devices. The Yarp approach is to define interfaces for classes of devices to wrap native code APIs (often provided by the hardware manufactures). Change in hardware will likely require only a change in the API calls (and linking against the appropriate library). This easily encapsulates hardware dependencies but leaves dependencies in the source code. The latter can be removed by providing a “factory” for creating objects at run time (on demand).

The combination of the port and device abstractions leads to remotable device drivers which can be accessed across a network: e.g. a grabber can send images to a multitude of listeners for parallel processing.

Overall, Yarp’s philosophy is to be lightweight and to be “gentle” with existing approaches and libraries. This naturally excludes hard real-time issues that have to be necessarily addressed elsewhere, likely at the OS level.

¹¹ <http://www.cmake.org>

3.4 Research with the iCub

One of the goals of the RobotCub project is to further our understanding of cognitive systems through the ontogenic development of a humanoid robot. That is, it is a program of enquiry into emergent embodied cognitive systems whereby a humanoid robot, equipped with a rich set of innate action and perception capabilities, can develop over time an increasing range of cognitive abilities by recruiting ever more complex actions and thereby achieving an increasing degree of prospection (and hence, adaptability and robustness) in dealing with the world around it.

Cognitive development involves several stages, from coordination of eye-gaze, head attitude, and hand placement when reaching, through to more complex — and revealing — exploratory use of action. This is typically achieved by dexterous manipulation of the environment to learn the affordances of objects in the context of one's own developing capabilities. Our ultimate goal is to create a humanoid robot — the iCub — that can communicate through gestures simple expressions of its understanding of the environment, an understanding that is achieved through rich manipulation-based exploration, imitation, and social interaction.

This program of research is carried out both by investigating natural cognition and by implementing some of these skills on the iCub. In particular, there is solid evidence that sustains the hypothesis of a significant involvement of the motor system in supporting processes traditionally considered as 'high level' or cognitive, such as action understanding, mental imagery of actions, objects perception and discrimination. The *biologically plausible stance* guiding the RobotCub project requires a deeper study of these processes not only because of their scientific value but also because their implementation on the iCub can provide better understanding of the corresponding neural processes.

A typical example of how sensorimotor integration is used by the brain in practical tasks is provided by a population of neurons in the monkey ventral premotor cortex (mirror neurons) that discharge both when the monkey performs a grasping action and when it observes the same action performed by other individuals [43].

A further example of the involvement of the motor system in cognitive functions is given by spatial attention. The view that we favor is that attention derives from the same circuits that are responsible for the control of eye movements. This view was originally proposed for visuospatial attention by Rizzolatti and Camarda [44]. Similar properties apply to the selection of visual stimuli even if not necessarily connected to eye movements. Craighero and colleagues [45] have shown that this is the case for other actions like for example grasping.

From the robotics side, RobotCub has started investigating several elements of these sensorimotor processes.

One first example is concerned with the basis of motor control for the iCub. In this case, we hypothesized the existence two basic types of motor primitives: i.e. discrete (aperiodic and finite) and rhythmic (periodic) movements [46]. We model these motor primitives as solutions of a dynamical system with a globally attractive fixed point

and an oscillator, respectively. Such an approach allows us to use the stability properties of dynamical systems to ensure a robust control of the movements. This controller allows the simultaneous execution of rhythmic and point to point movements, as required for example in locomotion. Separately, we developed a task-space based controller for the iCub that can take into account joint limits together with any number of constraints in generating appropriate kinematic inverses.

Finally, we would like to mention the implementation on the iCub of exploratory procedures for objects. Building on the existing sensorimotor behaviors, the iCub can touch and elicit movement from objects. Following the object behavior, the association of sensory and motor cues allows the autonomous acquisition of object affordances [47]. Object affordances are a powerful tool for a cognitive system, allowing the interpretation of scenes (because of the behavior of objects) as well as the imitation of other people's actions (as a consequence of the behavior of objects) [48].

Examples of these experiments are illustrated in Figure 6.



Figure 6: iCub experimenting with affordances (left) and exercising the reaching-rhythmic controller (right).

The Open Source approach used in the production and dissemination of the iCub platform supports the process of standardization and platform benchmarking in cognitive science research [9]. Currently, more than 10 iCub platforms are being produced to be used in various labs in the EU and USA. Moreover, the distribution of the Open Source iCub simulator (see next section) will further permit the use of the same robotic platform in many more labs, further contributing to the use of iCub as a benchmark platform.

4 The iCub simulator

Computer simulations play an important role in robotics research. Despite the fact that the use of a simulation might not provide a full model of the complexity present in the real environment and might not assure a fully reliable transferability of the controller from the simulation environment to the real one, robotic simulations are of great interest for cognitive scientists [17].

The iCub simulator has been designed to reproduce, as accurately as possible, the physics and the dynamics of the robot and its environment with the constraint of running approximately in real-time. The simulated iCub robot is composed of multiple rigid bodies connected via joint structures. It has been constructed collecting data directly from the robot design specifications in order to achieve an exact replication (e.g. height, mass, Degrees of Freedom) of the first iCub prototype developed by the RobotCub Consortium. The environment parameters on gravity, objects mass, friction and joints are based on known environment conditions. The iCub simulator presented here has been created using Open Source libraries in order to make it possible to distribute the simulator freely to any researcher without requesting the purchase of restricted or expensive proprietary licenses.

4.1 Physics engine

The iCub simulator uses ODE (Open Dynamic Engine)¹² for simulating rigid bodies and the collision detection algorithms to compute the physical interaction with objects. The same physics library was used for the Gazebo project and the Webots commercial package. ODE is a widely used physics engine in the Open Source community, whether for research, authoring tools, gaming etc. It consists of a high performance library for simulating rigid body dynamics using a simple C/C++ API. ODE was selected as the preferred Open Source library for the iCub simulator because of the availability of many advanced joint types, rigid bodies (with many parameters such as mass, friction, sensors...), terrains and meshes for complex object creation.

4.2 Rendering engine

Although ODE is a good and reliable physics engine, computing all the physical interaction of a complex system can take a good deal of processing power. Since ODE uses a simple rendering engine based on OpenGL, it has limitations for the rendering of complex environments comprising many objects and bodies. This can significantly affect the simulation speed of complex robotic simulation experiments. It was therefore decided to use OpenGL directly combined with SDL¹³, an Open Source cross-platform multimedia library. This makes it possible to render the scene with more ease and to carry out computationally-efficient simulation experiments.

¹² Open Dynamics Engine <http://opende.sourceforge.net/>.

¹³ SDL – Simple DirectMedia Layer <http://www.libsdl.org>

4.3 YARP protocol for simulated iCub

As the aim was to create a replica of the physical iCub robot, the same software infrastructure and inter-process communication will have to be used as those used to control the physical robot. As the physical iCub is based on YARP as its software architecture, the simulator and the actual robot share the same interface either when viewed via the device API or across network and are interchangeable from a user perspective. The simulator, like the real robot, can be controlled directly via sockets and a simple text-mode protocol; use of the YARP library is not a requirement. This can provide a starting point for integrating the simulator with existing controllers in esoteric languages or complicated environments. The user code can send and receive information to both the simulated robot itself (motors/sensors/cameras) and the world (manipulate the world). Network wrappers allow device remotization. The Network Wrapper exports the YARP interface so that it can be accessed remotely by another machine (Figure 7)

4.4 iCub body model

The iCub simulator has been created using the data from the physical robot in order to have a replica of it. As for the physical iCub, the total height is around 100cm, weighs approximately 22kg and has a total of 53 degrees of freedom (DoF). These include 12 controlled DoFs for the legs, 3 controlled DoFs for the torso, 32 for the arms and six for the head.

The robot body model consists of multiple rigid bodies attached through a number of different joints. All the sensors were implemented in the simulation on the actual body, such as touch sensors and force/torque sensors. As many factors impact on the torque values during manipulations, the simulator might not guarantee to be perfectly correct. However the simulated robot torque parameters and their verification in static or motion are a good basis and can be proven to be reliable [18].

All the commands sent to and from the robot are based on YARP instructions. For the vision we use cameras located at the eyes of the robot which in turn can be sent to any workstation using YARP in order to develop vision analysis algorithms.

The system has full interaction with the world/environment. The objects within this world can be dynamically created, modified and queried by simple instruction which are exactly those that YARP uses in order to control the real robot.

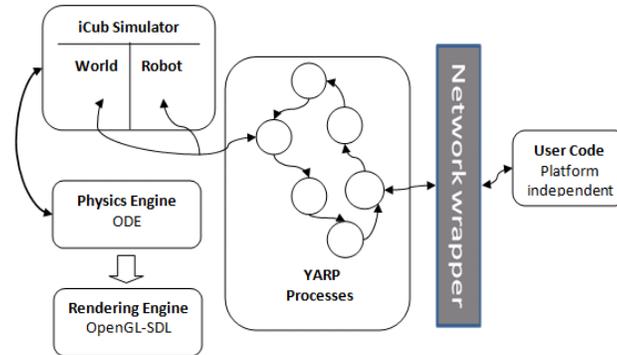


Figure 7: This shows the architecture of the simulator with YARP support.

4.5 Simulator testing and further developments

The current version of the iCub simulator has been used for preliminary testing by partners in the RobotCub and ITALK projects. The ITALK project [19] aims at the development of cognitive robotic agents, based among others on the iCub humanoid platform, that learn to handle and manipulate objects and tools autonomously, to cooperate and communicate with other robots and humans, and to adapt their abilities to changing internal, environmental, and social conditions. The main theoretical hypothesis behind the project is that the parallel development of action, conceptualisation and social interaction permits the bootstrapping of language capabilities, which on their part enhance cognitive development. This is possible through the integration and transfer of knowledge and cognitive processes involved in sensorimotor learning and the construction of action categories, imitation and other forms of social learning, the acquisition of grounded conceptual representations and the development of the grammatical structure of language.

In addition to being used for experiments on the development of controllers for the iCub robot, some groups have used the simulator to create a mental model [20] used by the robot to represent the current state of the environment.

Future plans on the simulator development will mostly involve the design of functionalities to model and interact with the physical environment. For example, this will allow the users to modify the objects in the world where the iCub resides, in order to allow different types of experiments. Finally, further work will focus on the systematic testing and replication of simulation studies with the physical robot. The latest version of the iCub simulation is available Open Source in the RobotCub/iCub repository¹⁴.

¹⁴ <http://www.robotcub.org/>

5 Symbiotic Robot Organisms: Replicator and Symbrion Projects

5.1 Introduction

Nature shows several interesting examples for cooperation of individuals. Most prominent examples of cooperation are found in social insects [21], where specialized reproductive schemes (in most cases just a few out of thousands of colony members are able to reproduce) and the close relationships of colony members favoured the emergence of highly cooperative behaviours [22]. However, also non-eusocial forms of cooperative communities evolved, like the collective hunting in predatory mammals [23] (e.g., lions, whales, ...) or the reciprocal trophallactic altruism in vampire bats. Such cooperative behaviours are mostly explained by reciprocal advantages due to the cooperative behaviours and/or by the close relationship among the community members. In contrast, cooperation sometimes arises also among individuals that are not just very distant in a gene pool, sometimes they do not even share the same gene pool: Cooperative behaviours between members of different species is called 'Symbiosis'. A non-exhaustive list of prominent examples includes the pollination of plants by flying insects (or birds), the cooperation between ants and aphids. Also lichens, which are a close integration of fungi and algae and the cooperation between plant roots and fungi represent symbiotic interactions.

A common pattern in all these above-mentioned forms of cooperation is that single individuals perform behaviours, which - at first sight - are more supportive for the collective of the group than for themselves. However, as these behaviours have emerged through natural selection, we can assume that these cooperative behaviours have their ultimate reasoning in a sometimes delayed and often non-obvious individual egoistic advantage.

From symbiotic forms of organization emerge new functional capabilities which allow aggregated organisms to achieve better fitness in the environment. When the need of aggregation is over, symbiotic organisms can dis-aggregate and exist further as stand-alone agents, thus an adaptive and dynamical form of cooperation is often advantageous.

Lately, technical systems mimic natural collective systems in improving functionality of artificial swarm agents. Collective, networked or swarm robotics are scientific domains, dealing with cooperation in robotics [24]. Current research in these domains is mostly concentrated on cooperation and competition among stand-alone robots to increase their common fitness [25]. However, robots can build a principally new kind of collective systems, when to allow them to aggregate into a multi-robot organism-like-forms. This *robot organism* can perform such activities that cannot be achieved by other kind of robotic systems and so to achieve better functional fitness.

To demonstrate this idea, we consider a collective energy foraging scenario for the Jasmine micro-robots [26]. Swarm robots can autonomously find an energy source and recharge. The clever collective strategy can essentially improve the efficiency of

energy foraging, but nevertheless a functional fitness of a swarm is limited. For instance, if the recharging station is separated from a working area by a small barrier, robots can never reach the energy source. However, if robots aggregate into more complex high-level organism which can pass the barrier, they will reach the docking station. In this way a cooperative organization of robotic system allows an essential increase of functional capabilities for the whole group. The large integrated project REPLICATOR¹⁵ deals with such issues as reconfigurability of sensors and actuators, adaptive control and learning strategies as well as working in real environments.

The cooperative (swarm-based or symbiotic) organization of the robotic system provides essential plasticity of used hardware and software platforms. The robot organism will be capable of continuously changing its own structure and functionality. Such an evolve-ability opens many questions about principles and aspects of long- and short-term artificial evolution and controllability of artificial evolutionary processes. The large integrated project SYMBRION¹⁶ is focused on evolve-ability, dependability and artificial evolution for such robot organisms based on bio-inspired and computational paradigms. Both projects are Open Science and Open Source.

Both projects, consortia and the European commission are closely cooperating to achieve the targeted goals. It is expected that results of both projects create new technology for making artificial robotic organisms self-configured, self-healing, self-optimizing and self-protecting from a hardware and software point of view. This leads not only to extremely adaptive, evolve-able and scalable robotic systems, but also enables the robot organisms to reprogram themselves without human supervision, to develop their own cognitive structures and, finally, to allow new functionalities to emerge.

5.2 New Paradigm in Collective Robotic Systems

Collective intelligence is often associated with macroscopic capabilities of coordination among robots, collective decision making, labor division and tasks allocation in the group [27]. The main idea behind this is that robots are achieving better performance when working collectively and so are capable of performing such activities which are not possible for individual robots. The background of collective intelligence is related to the capability of swarm agents to interact jointly in one medium. There are three different cases of such interactions:

1. In the first case agents communicate through a digital channel, capable for semantic messages exchange. Due to information exchange, agents build different types of common knowledge [28]. This common knowledge in fact underlies collective intelligence.

¹⁵ www.replicatores.eu

¹⁶ www.symbion.eu

2. The second case appears when macroscopic capabilities are defined by environmental feedback. The system builds a closed macroscopic feedback-loop, which works in a collective way as a distributed control mechanisms. In this case there is no need of complex communication, agents interact only by kinetic means. This case of interaction is often denoted as a spatial reasoning, or spatial computing.

3. The third case of interactions we encounter in nature, when some bacteria and fungi (e.g. *dictyostelium discoideum*) can aggregate into a multi-cellular organism when this provides better chances of survival [29]. In this way, they interact not only through information exchange or spatial interactions, they build the closest physical, chemical and mechanical interconnections, though the agents still remain independent from each other. The first two cases of interactions are objects of extensive research in many domains: robotics, multi-agents systems, bio-inspired and adaptive community and so on. However, the practical research in the last case represents essential technological difficulties and therefore is not investigated enough. Despite the similarities between a robot swarm and multi-robot organism, such as a large number of robots, focus on collective/emergent behavior, a transition between them is a quite difficult step due to mechanical, electrical and, primarily, conceptual issue. Now, we believe that research around the third case of interactions is concentrated on three important questions:

1. **Reconfigurability**, adaptability and learn-ability of the symbiotic systems. These issues include flexible and multifunctional sensors and actuators, distributed computation, scalability, modelling, control and other issues, which are closely related to the reconfigurable robotic research. The REPLICATOR project is focused on these points.

2. **Evolve-ability** of the symbiotic systems, which includes principles and aspects of long- and short-term artificial evolution and adaptivity as well as exploring and analogies to biological systems. The SYMBRION project is focused on these points.

3, **Embodiment** of evolutionary systems for different environments and medias as well as investigation of information properties of such systems. These points are covered by other research initiatives and projects.

In this way, the next step in further research within the collective robotic community can consist in investigation of multi-robot organisms or, in other words, a transition from robot swarm to multi-robot organisms. All further sections are devoted to demonstrate diverse aspects of such a transition.

5.3 Example: Energy Foraging Scenario

In this section we demonstrate the advantages of symbiotic organization of autonomous robotic systems. We choose for this purpose an example of energy homeostasis, because it is applicable to both living and robotic organisms and so we can draw several analogies between them.

The distinctive property of any living organism is energy homeostasis and, closely connected, foraging behavior and strategies [23]. The robots, equipped with on-board recharging electronics, can also possess its own energy homeostasis. In this way, when swarm robots get *hungry*, they can collectively look for energy resources and execute different strategies in a cooperative energy foraging [30]. In critical cases robots can even decide to perform individual foraging, competing with other robots for resources.

The need of energy is a perfect example of natural fitness. If robots that are performing individual strategies find enough energy, they can survive in the environment. In turn, this means that these strategies were sufficient enough to balance these robots energetic budgets. Simultaneously, other energetically die if their behavioral strategy was poor. Based on such energy foraging, many of evolutionary approaches for different robotic species can be developed, compared and tested.

However, if there are many robots foraging in the environment, several undesired effects can emerge: (1) the docking station can become a *bottleneck* resource that essentially decreases the swarm efficiency; (2) robots with a high-energy level can occupy the docking station and block low-energetic robots. These robots can energetically die (and so decrease the swarm efficiency); (3) many robots can create a *crowd* around a docking station and essentially hinder a docking approach. This can increase the total recharging time and makes worse the energetic balance of the whole swarm.

Robots, in pursuing their energetic homeostasis, have only two possible decisions to make: (1) to execute a current collective task or (2) to move for recharging. In balancing these two behaviours, a cooperative strategy may find the right timing and the right combination between these individual decisions of all robots. Lately, several strategies of energy foraging for a robot swarm up to 70 swarm agents are implemented, see Fig. 8. These cover different bio-inspired approaches [31, 32] and hand-coded strategies [33].



Figure 8: (a) Docking of a few robots for recharging. Shown is the two-line approach: the first line - recharging robots, the second line - robots waiting for recharging; (b) The "barrier problem" – robots separated from docking stations by a barrier; (c) Possible solution to the "barrier problem": swarm robots form a symbiotic multi-robot organism & collective pass the barrier.

In one of these experiments, a few robots died close to the docking station and blocked the recharge area (we *simulated* this in the Fig. 8b). Robots that were in front of this barrier (away from the docking station) finally also died. This is the limit of

functional fitness of swarm robots. There is no strategy, that allow swarm robots to overpass the barrier. Only when swarm robots would collectively emerge new functionality, like *pass the barrier*, they would solve the *barrier problem*.

Thus, an ideal solution for the *barrier problem* can be the aggregation of many single robots into one cooperative multi-robot organism. This way, they can reach the docking stations by "growing legs" and stepping over the barrier. In that case, the robots are helping each other in a cooperative manner, see Fig. 8c).

Obviously, such a robotic behaviour is extremely challenging from many viewpoints: Cooperative (symbiotic) robot systems have many similarities with known robotic research as e.g. mechanical self-assembling [34] or reconfigurable robotics [35]. However, the symbiotic form, show in Fig. 8 essentially differs from this robotic research, namely: (1) Robots should be capable for autonomous aggregation and disaggregation; (2) Robots in the disaggregate state should possess individual locomotion; (3) There is no central control neither for disaggregated state (swarm) nor for the aggregate state (organism); (4) Stand-alone robots should profit from the aggregation into organism.

The swarm-based approaches, which is underlying the aggregation processes, differs primarily from aggregated systems which are studied in the field of *reconfigurable robotics*. In the following we consider on-going work with aggregated (symbiotic) robot organisms.

5.4 Hardware and Software Challenges

The main feature of a modular robot consists in being composed of several independent modules, with limited complexity and capabilities, which are able to connect to each other in different configurations, in order to form a robot with greater capabilities. As a consequence, the overall functionalities and capabilities of a robotic modular organism are deeply related to the hardware structure and functions of its basic composing modules. At the current stage of development of the projects (both projects started in 2008), the development of the hardware represents one of the hardest issues. In general, the concept of hardware design is as follows:

1. Independence for separate robots, this includes capabilities for communication, computation and sensing as a stand-alone robot, as well as individual locomotion and energy management.
2. Large computational power of the organism, required for performing on-line and on-board evolutionary approaches.
3. Heterogeneity of individual robots, which allows their later specialization within the organism.
4. Rich sensing and communication capabilities of the organism. The more robots are joined in the organism, the more functional diversity the organism can demonstrate.

5. Possible higher independency from human in term of energy, support and maintenance.

The consortia considered many state-of-the-art reconfigurable solutions, such as superBot [36], M-Tran [37], Poly-Bot [38], molecube [39], HYDRA/ATRON [40] and others, even visited some of these labs for exchange of experience. Currently, we follow three different developmental lines, which will be later fused into one or two first prototypes.

The electronic design is also a huge challenge due to strong restrictions of the size of the robot and the complexity of the mechanical design. Each stand-alone robot is equipped with several 32 bit ARM-based microcontrollers and one main microprocessor with a large external memory. The breakdown in microcontrollers and microprocessor was deliberately intended to separate computational tasks within the single cell. The microcontrollers perform basic functionality (e.g. sensor pre-processing, running artificial immune system) and keep the robot alive. The microprocessor is mainly responsible for bio-inspired approaches like the genetic algorithms, sensor fusion, ANNs etc. and is more powerful in comparison to the microcontrollers. Due to higher computational power results in higher energy consumption, the microprocessor is able to run at different power-down modes when computational power isn't needed. One of the biggest challenges during the electronic design is the development of all modules in a tiny size as well as finding solutions for shared resources like memory, power and communication capabilities.

Beside the hardware challenges, the project is faced with many software and controller challenges. Because robots can either run independently, as a swarm, as an organism, or even as a swarm of organisms, the interaction has to be managed in an organized and efficient way. The whole software is divided into several layers. On the bottom layers, such as BIOS and real time operational system, there are different processes, which are able to communicate with each other and to control a low-level behavior of all components. To cope with the additional difficulties a swarm or an organism causes, a middleware-like system is necessary. On top of this abstraction layer high-level control mechanisms, distributed applications and genetic framework are integrated.

5.5 Towards Evolve-ability and Benchmarking of Robot Organisms

Within the projects, the creation of evolvable or otherwise adaptive software and hardware is the main focus. Achievement of evolve-ability for the robot organism is planned in two complementary ways, which we call bio-inspired (or bio-mimicking) and engineering-based approaches. Comparison between both approaches allows benchmarking the behavior and functionality of evolved organisms. It is also planned to use four benchmark tests on reconfigurability of the platform, sensing and sensor-fusion capabilities, behavioral and learning tasks as well as on evolving approaches. Evaluation can be performed qualitatively in terms of success in solving corresponding tasks and quantitatively by comparing running time and consumed energy.

Bio-inspired/bio-mimicking approach

Any bio-inspired approach is based on analogies to living organisms and is carried out by the biological partners in our consortia. Our bio-inspired control algorithms use neither any global point of information nor any form of complex knowledge. Our algorithms are stable to a wide range of environmental conditions and are extremely robust. Therefore, the bio-inspired strategies in projects are going to draw advantage from the well-known robustness/simplicity as well as from the plasticity/adaptability derived from natural systems. Our goal is to create stable, robust and adaptable robotic organisms. Here we will investigate a variety of concepts, such as:

Genome: All robotic organisms will carry one or several Genomes. A Genome is a collection of genes, which carry information about controller structure and controller dynamics. A gene can be a simple part of a blueprint, which "depicts" a part of the final controller. But a gene can also work as a rule, which is used to "construct" parts of the final controller. In the latter case, there can be interferences between different genes, thus competition or cooperation can arise also on the genetic level. A self-organized process can be established which will be able to create a flexible, but robust controller structure.

Controller: We will investigate several controller types, ranging from rules-based controllers, to Evolvable Artificial Neural Networks (EANN) and Artificial Immune Networks (AIS) to hormone-based controllers and to even hand-coded controllers that execute hand-optimized (modular) parts of the whole organism's behavioural repertoire.

Sexuality/Reproduction: We plan to enhance and to speed-up the dynamics of artificial evolution by implementing virtual-reproduction of robots. A separate process will allow removing controllers from the least fit robots and to re-initialize them with mixtures (interbreeds) of the controllers of more fit robots. We will also investigate the advantages of sexual reproduction in such scenarios.

Embryology: To allow well-ordered controllers to emerge from the information stored in the Genome, we will mimic embryological processes, driven by a virtual hormone system.

Engineering-based approach

The engineering-based approach is complementary to the bio-inspired one and focuses in such issues as learning, distributed decision making, navigation and so on. Generally, consortium focuses on three following approaches (these approaches are closely connected so that finally it will be a kind of hybrid framework):

On-line learning. On-line learning is based on the behavior level and uses automatically generated feedback. The feedback comes from internal, external and virtual sensors. Some direct feedback can be sensed through vision-based subsystem, by using FRID-based identification or localization technologies, by using smart laser scanner, sound, light, humidity, temperature, internal energy sensor and other sensors. It is intended to use middleware and sensor-fusion approach to generate complex

non-direct feedbacks through virtual sensors. Since off-line mechanisms can hardly be applied to real robots, the challenge of the proposed approach is to perform non-supervised learning without any off-line mechanisms (or at least with a minimum of them). This can be achieved by combining evolving computation with rewards/feedback/fitness calculated on-line. Therefore the whole approach can be named "on-line learning".

Evolutionary computation. High computational power of the system allows running on-line and on-board such well-known approaches as genetic programming (GP) (e.g. [41]), Genetic Algorithms (GA) (e.g. [42]). To avoid the problems posed by a huge search space, we intend to integrate limitations, originating from hardware platform. Another set of problems we are aware of are the fitness functions required for these algorithms. These fitness functions are very difficult to calculate based only on local sensor data. Moreover these functions are evaluated extremely delayed because the organism mostly assess their fitness after accomplishing the task.

Approaches from the domain of Distributed Artificial Intelligence (DAI). On-line learning as well as GA/GP include diverse aspects of DAI such as a distributed knowledge management, semantic information processing, navigation and actuation in the environment, planning, sensor fusion and others. Development and implementation of these approaches is an important step towards evolve-ability of the robot organisms.

5.6 Discussion

We provided an overview of two large European projects, dealing with a new paradigm in collective systems, where the swarm robots get capable of self-assembling into a single symbiotic multi-robot organism. We introduced an energy foraging scenario for both robot species and demonstrated that a transition between collective and symbiotic robot forms represents a very hard problem. It involves not only hardware and software issues, but also very basic questions being also open not only in biological but also in engineering sense. We demonstrated the main hardware and software challenges and the road-map how to achieve the evolve-ability of the robot organisms.

6 Other projects and future work

We have described three specific approaches to cognitive robotics, each with a significant Open Source aspect. The clearest of these is the Rat's Life benchmark which is applied to competitively evaluate the performance of various approaches to robot control for navigation in an unknown environment.

In the iCub project, Open Source forms the basis of a platform for experimental work in humanoid robotics that can be replicated across multiple labs. The motivation is primarily community building and leveraging the work of others, although as a con-

sequence, performance evaluation is also made possible. The relatively large number of iCub platforms being produced is an encouraging indicator of the willingness of research labs to cooperate at this level, with the distribution of the Open Source iCub simulator, an additional mechanism for sharing.

Within the field of swarm robotics, the creation of evolvable or otherwise adaptive software and hardware is the main focus. Here the Open Source aspect is perhaps more subtle, as it lies both in the creation and operation of multiple identical agents, as well as in the execution of the experiments which allow different approaches to be compared and replicated.

As these Open Source cognitive paradigms start to gain a foothold and the benefits within the research community spread, so we can expect the Open Source approach to be applied in a wide range of robotics research. Not surprisingly, a number of the other projects within the current EU programme also have an element of open source as listed in the table below:

Further details of these projects are available on the website¹⁷.

project	Open Source component
RoSta	architectures and standards for robotics
PHRIENDS	plans to deliver open-source code at the end of the project
DEXMART	plans to use a open source controller
SEARISE	implementation of software based on open-source modules
GRASP	Open source platform for benchmarking in grasping and dextrous manipulation. Open source development in detecting the unexpected and learning from it
SF	Use of multi-level neuronal simulation environment (iqr) released under GNU general public Licence
SCOVIS	Use of Open CV tool
POETICON	open resource for grounding action (movement) semantics

Figure 9: list of current projects with Open Source element

Acknowledgements

The Rat's Life benchmark was supported by the European Commission ICEA project¹⁸, while the work on iCub was supported by the European Commission FP6

¹⁷ www.cognitivesystems.eu

¹⁸ ICEA European Project (IST 027819) ICEA stands for Integrating Cognition, Emotion and Autonomy and is focused on brain-inspired cognitive architectures, robotics and embodied

Project RobotCub and FP7 Project ITALK¹⁹ within the Cognitive Systems and Robotics unit. The authors would like to thank the RobotCub Consortium. Paul Fitzpatrick is gratefully acknowledged for the continuous support to Yarp. The REPLICATOR and SYMBRION projects are funded by European Commission within the 7th framework program. The authors also acknowledge the support of the FP6 euCognition²⁰ Coordinated Action project funded under the same Cognitive Systems and Robotics unit.

REFERENCES

- [1] J. Baltes, 2000, A benchmark suite for mobile robots, IROS 2000 conference proceeding, 1101-1106 vol.2, ISBN: 0-7803-6348-5
- [2] V. Braitenberg, 1984, *Vehicles: Experiments in Synthetic Psychology* Cambridge, MA: MIT Press.
- [3] A. Serri, 2004, A Lego robot for experimental benchmarking of robust exploration algorithms, *Circuits and Systems*, iii - 163-6 vol.3, ISBN: 0-7803-8346-X
- [4] M. Eaton, J.J. Collins and L. Sheehan, 2001, Toward a benchmarking framework for research into bio-inspired hardware-software artefacts, *Artificial Life and Robotics*, Springer Japan, 5(1):40-45, ISSN: 1433-5298
- [5] J.J. Collins, M. Eaton, M. Mansfield, D. Haskett and S. O'Sullivan, 2004, Developing a benchmarking framework for map building paradigms, *Proceedings of the 9th International Symposium on Artificial Life and Robotics*, January, Oita, Japan, pp.614-617
- [6] H. Kitano, M. Asada, Y. Kuniyoshi et al, 1995, RoboCup: the robot world cup initiative, *IJCAI-95 workshop on entertainment and AI/ALife*.
- [7] A. Jacoff, B. Weiss and E. Messina, 2003, Evolution of a Performance Metric for Urban Search and Rescue Robots, *Proceedings of the 2003 Performance Metrics for Intelligent Systems (PerMIS) Workshop*, Gaithersburg, MD, September 16-18, 2003
- [8] O. Michel, 2004, Webots: Professional Mobile Robot Simulation, *Journal of Advanced Robotics Systems*, 1 (1): pp. 39-42
- [9] R. Dillmann, 2004, *Benchmarks for Robotics Research*, EURON, April.
- [10] L. Fadiga, L. Craighero and E. Olivier, 2005, Human motor cortex excitability during the perception of others' action, *Current Biology*, vol. 14 pp. 331-333.
- [11] L. Fadiga, L. Craighero, G. Buccino and G. Rizzolatti, 2002, Speech listening specifically modulates the excitability of tongue muscles: a TMS study, *European Journal of Neuroscience*, vol. 15, pp. 399-402.
- [12] G. Rizzolatti and L. Fadiga, 1998, Grasping objects and grasping action meanings: the dual role of monkey rostroventral premotor cortex (area F5), in *Sensory Guidance of Movement*, Novartis Foundation Symposium, G. R. Bock and J. A. Goode, Eds. Chichester: John Wiley and Sons, pp. 81-103.

cognition, bringing together cognitive scientists, neuroscientists, psychologists, computational modelers, roboticists and control engineers. It aims at developing a cognitive systems architecture integrating cognitive, emotional and bioregulatory (self-maintenance) processes, based on the architecture and physiology of the mammalian brain. <http://www.iceaproject.eu>

¹⁹ robotcub.org and italkproject.org

²⁰ <http://cordis.europa.eu/ist/cognition/index.html>

- [13] D. Vernon, G. Metta and G. Sandini, 2007, A Survey of Cognition and Cognitive Architectures: Implications for the Autonomous Development of Mental Capabilities in Computational Systems, *IEEE Transactions on Evolutionary Computation*, special issue on AMD, vol. 11.
- [14] C. von Hofsten, 2003, On the development of perception and action, in *Handbook of Developmental Psychology*, J. Valsiner and K. J. Connolly, Eds. London: Sage, pp. 114-140.
- [15] P. Fitzpatrick, G. Metta and L. Natale, 2008, Towards Long-Lived Robot Genes, *J. of Robotics and Autonomous Systems*, Special Issue on Humanoid Technologies, vol. 56.
- [16] S. D. Huston, J. C. E. Johnson and U. Syyid, 2003, *The ACE Programmer's Guide*: Addison-Wesley, 2003.
- [17] T. Ziemke, 2003, On the role of robot simulations in embodied cognitive science, *AISB Journal*, 1(4), 389-99
- [18] N. Nava, V. Tikhonoff, G. Metta and G. Sandini, 2008, Kinematic and Dynamic Simulations for The Design of RoboCub Upper-Body Structure *ESDA*.
- [19] A. Cangelosi, T. Belpaeme, G. Sandini, G. Metta, L. Fadiga, G. Sagerer, K. Rohlfing, B. Wrede, S. Nolfi, D. Parisi, C. Nehaniv, K. Dautenhahn, J. Saunders, K. Fischer, J. Tani and D. Roy, 2008, The ITALK project: Integration and transfer of action and language knowledge. In: *Proceedings of Third ACM/IEEE International Conference on Human Robot Interaction (HRI 2008)*, Amsterdam, 12-15 March.
- [20] P.F. Dominey, 2007, Sharing Intentional Plans for Imitation and Cooperation: Integrating Clues from Child Developments and Neurophysiology into Robotics, *Proc. AISB 2007 Workshop on Imitation*.
- [21] E. Bonabeau, M. Dorigo and G. Theraulaz, 1999, *Swarm intelligence: from natural to artificial systems*. Oxford University Press, New York.
- [22] S. Camazine, J-L. Deneubourg, N.R. Franks, J. Sneyd, G. Theraulaz and E. Bonabeau, 2003, *Self-Organization in Biological Systems*. Princeton University Press, NJ, USA.
- [23] D.W. Stephens and J.R. Krebs, 1987, *Foraging Theory*. Princeton University Press.
- [24] E. Sahin, 2004, *Swarm Robotics: From sources of inspiration to domains of application*. Springer-Verlag, Heidelberg, Germany.
- [25] S. Kernbach, R. Thenius, O. Kernbach and T. Schmickl, Re-embodiment of honeybee aggregation behavior in artificial micro-robotic system. *Adaptive Behavior* (in press).
- [26] S. Kornienko, O. Kornienko and P. Levi, 2005, IR-based communication and perception in microrobotic swarms. In *Proc. IROS 2005*, Edmonton, Canada.
- [27] G. Weiss, 1999, *Multiagent systems. A modern approach to distributed artificial intelligence*. MIT Press.
- [28] J.Y. Halpern and Y. Moses, 1990, Knowledge and common knowledge in a distributed environment. *J. of ACM*, 37(3):549–587.
- [29] H. Haken, 1983, *Synergetics: An introduction*, third edition. Springer-Verlag, New York.
- [30] S. Kornienko, O. Kornienko, A. Nagarathinam and P. Levi, 2007, From real robot swarm to evolutionary multi-robot organism. In *Proc. CEC2007*, Singapore.
- [31] D. Häbe, 2007, Bio-inspired approach towards collective decision making in robotic swarms. Master Thesis, University of Stuttgart, Germany.

- [32] T. Kancheva, 2007, Adaptive role dynamics in energy foraging behavior of a real micro-robotic swarm. Master Thesis, University of Stuttgart, Germany.
- [33] A. Attarzadeh, 2006, Development of advanced power management for autonomous micro-robots. Master Thesis, University of Stuttgart, Germany.
- [34] A. Ishiguro and T. Maegawa, 2006, Self-assembly through the interplay between control and mechanical systems. In Proc. of IEEE/RSJ06 Int. Conf. on Intelligent Robots and Systems, pp. 631–638, Beijing, China.
- [35] S. Murata, K. Kakomura and H. Kurokawa, 2006, Docking experiments of a modular robot by visual feedback. In Proc. of IEEE/RSJ06 Int. Conf. on Intelligent Robots and Systems, pp. 625–630, Beijing, China.
- [36] W-M. Shen, M. Krivokon, H. Chiu, J. Everist, M. Rubenstein and J. Venkatesh, 2006, Multimode locomotion for reconfigurable robots. *Autonomous Robots*, 20(2):165–177.
- [37] H. Kurokawa, K. Tomita, A. Kamimura, S. Kokaji, T. Hasuo and S. Murata, 2008, Distributed self-reconfiguration of m-tran iii modular robotic system. *Int. J. Rob. Res.*, 27(3-4):373–386.
- [38] A. Golovinsky, M. Yim, Y. Zhang, C. Eldershaw and D. Duff, 2004, Polybot and polykinetic/spl trade/ system: a modular robotic platform for education. In *IEEE ICRA*, 1381–1386.
- [39] V. Zykov, E. Mytilinaios, B. Adams and H. Lipson, 2005, Self-reproducing machines. *Nature*, 435(7039):163–164.
- [40] D.J. Christensen, E.H. Ostergaard and H.H. Lund, 2004, Metamodule control for the atron self-reconfigurable robotic system. In Proc. IAS-8, pp.685–692, Amsterdam,
- [41] J. Koza, 1992, Genetic programming: on the programming of computers by means of natural selection. MIT Press, Cambridge, Mass.
- [42] M. Srinivas and L.M. Patnaik, 1994, Genetic algorithms: A survey. *Computer*, 27(6):17–26.
- [43] V. Gallese, L. Fadiga, L. Fogassi and G. Rizzolatti, 1996, Action recognition in the premotor cortex. *Brain*, 119, 593-609.
- [44] G. Rizzolatti and R. Camarda, 1987, Neural circuits for spatial attention and unilateral neglect. In M. Jeannerod (Ed.), *Neurophysiological and neuropsychological aspects of spatial neglect* (pp. 289-313). Amsterdam: North Holland.
- [45] L. Craighero, M. Nascimben and L.Fadiga, 2004, Eye Position Affects Orienting of Visuospatial Attention. *Current Biology*, 14, 331-333.
- [46] S. Degallier, L. Righetti, L. Natale, F. Nori, G. Metta and A. Ijspeert, 2008, A modular bio-inspired architecture for movement generation for the infant-like robot iCub. In *Proceedings of 2nd IEEE RAS/EMBS Intl. Conf. on Biomed. Rob. And Biomech. (BioRob)*
- [47] L. Montesano, M. Lopes, A. Bernardino and J. Santos-Victor, 2008, Learning Object Affordances: From Sensory Motor Maps to Imitation, *IEEE Trans. Rob.*, vol. 24(1), Feb.
- [48] G. Metta and P. Fitzpatrick, 2003, Early Integration of Vision and Manipulation. *Adaptive Behavior*, 11(2), 109-128.
- [49] J.S. Albus, 2002, Metrics and Performance Measures for Intelligent Unmanned Ground Vehicles. In *Proceeding of the performance Metrics for Intelligent System Workshop*, NIST, USA

- [50] A.P. del Póbil, 2006, Why do We Need Benchmarks in Robotics Research?, International Conference on Intelligent Robot and Systems, Beijing, China
- [51] F. Bonsignorio, J. Hallam and A.P. del Póbil, 2007, Good Experimental Methodologies in Robotics: State of the Art and Perspectives, Workshop on Performance Evaluation and Benchmarking for Intelligent Robots and Systems, IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, USA