Development of an Organic Computing Architecture for Robot Control *

Bojan Jakimovski, Marek Litza, Florian Msch, Adam El Sayed Auf

Institut für Technische Informatik
Universität Lübeck
Ratzeburger Allee 160
D-23538 Lübeck, Germany
{bojan,litza,fmoesch,elsayedauf}@iti.uni-luebeck.de

Abstract: Motivated by the vision of Organic Computing, a concept is proposed towards building a more robust robot control architecture. An experimental setup is described that allows to develop, implement and test new approaches and their practical realizations on a walking robot demonstrator. These experiments demonstrate that normal walking of the robot can be distinguished from the walking with a defective leg or against an obstacle by observing appropriate sensor signals from the legs. Based on this, the properties of self-adaptation and self-reconfiguration in case of malfunctions of some parts or in case of obstacle avoidance can be realized.

1 Tendency towards organic computing systems

Now more than ever, the constantly increasing complexity of computer systems is heavily influencing all segments of IT and embedded systems industry. The need for approaches which can effortlessly cope with such type of complexity is becoming demanding. Several initiatives exist today, which pursue the development of new paradigms for autonomic systems. One of them is the IBM Manifesto for Autonomic Computing [IBM01], proposing several key elements for autonomic systems and their characteristics such as: self-configuring, self-healing, self-optimizing, self-protecting. Complementary to this, the Organic Computing initiative [Ger04], inspired by information processing in biological systems, proposes means of achieving such self-x properties by a novel concept of software architecture.

Such a software architecture will help achieving the anticipated level of reliability, availability and maintainability of future systems. The focus of our research is on employing an Organic Computing approach for realizing more robust, self-reconfiguring, self-adapting robots.

---

*This work is supported by DFG (associated to SPP 1183, MA 1412/7-1).

2 A robot system and its software control architecture

2.1 Robot-demonstrator

For carrying out experiments in the domain of Organic Computing, we have built a hexapod robot demonstrator. Our robot OSCAR (Organic Self Configuring and Adapting Robot), shown on Figure 1, is constructed out of: a round aluminium body, six aluminium legs with 60 degrees between each leg, three servo motors per leg and onboard control hardware. The diameter of the body is 17.5 cm and its fully span length is 47.5 cm.

Figure 1: Hexapod robot OSCAR (Organic Self Configuring and Adapting Robot)

Other dimensions of the robot parts and their schematic representation can be found in [SMM06]. Servo motors, depending on their position, are responsible for: protraction and retraction; elevation and depression; extension and flexion. Due to its constitution, the OSCAR robot is found to be a very good testbed for experimenting with various walking behaviours for planar and rough terrain. There has already been done a lot of work on gait control for walking robots, including biologically inspired approaches (see e. g. [SDK^{+}01]), but only little research so far has been carried out on walking robots still working in case of malfunctions. Our focus is to use Organic Computing principles in this case.

2.2 Software architecture inspired by the vision of Organic Computing

The mission of the Organic Computing initiative is to develop new approaches which will exhibit more adaptive and self-x properties of complex systems. One of the anticipated challenges Organic Computing should handle with is: implementing trustworthy systems which do not show undesired global behaviour [Sch05]. In that perspective, a development of such generalized robust Organic Computing software architecture would be the ultimate goal of this research. The research should also demonstrate the usability of the methodology of multi-agents [BK05] and modules, which can autonomously cooperate like peers [MMS^{+}06] with each others. Furthermore, such architecture can reveal the most wanted characteristic of the system of being easily adaptive. In addition, the system will consider the synergy effect of combination of concurrency and configurability [Wal04] within the building blocks of the architecture. For a realization of the Organic Computing inspired control architecture for engineering applications, the observer/controller concept [MS04] has been proposed. The objective is to achieve a controlled emergence property of the system. Namely, the system should be able to adapt its behaviour, to learn, and in the same time not to exceed the constraints defined by its specification. One of such Organic Computing inspired architectures is the ORCA-software architecture, proposed in [BMM05] for robot control applications. The basic architecture is shown in Figure 2. It is built from Organic Computing Units (OCUs) and Basic Computing Units (BCUs). The characteristics for that architecture are: scalability and flexibility to property changes of the constituent OCU and BCU units.

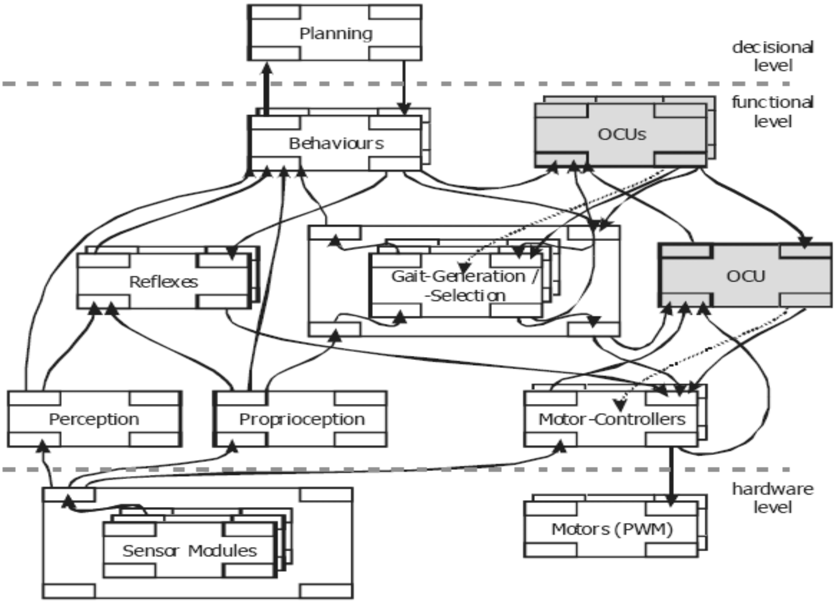


Figure 2: Schematic representation of standard ORCA architecture

As main building blocks of this architecture, BCUs, implement several tasks regarding the robot's control. These tasks can e. g. be defined as: simplified sensor acquisition, sensor information pre-processing and sensor fusion. BCUs can be also used for example, to control a complete robot leg segment. OCUs on the other hand are responsible for monitoring the correct behaviour of the BCUs and to influence their outputs or to change their behaviour in case of malfunctions.

2.3 Step towards implementation of a modified ORCA software architecture

As a step towards to a more robust control software architecture, we are modifying the concept of the ORCA architecture in a way of generalizing it, extending it and modifying the functionality of the software modules. Such a modified ORCA architecture is shown in Figure 3. As can be seen from the figure, the proposed architecture has a generalized representative character rather than initially pre-defined connections between the units. This approach where connections between the units will be dynamically built up during the runtime of the system, will further support emergence. The software units in the figure are grouped in different contexts (represented by different boxes). Units within a box have more cooperation and influence among each other than with the other units in the overlapped box. The units within the overlapped box also cooperate with each other, but with lower cooperation influence, opposed to cooperation within their own box. The units are grouped with respect to the type of behaviour, whether it is more reflexive or more cognitive. Modifying the ORCA architecture, furthermore, involves placing the "proprioception" module as an input for the "perception" module. The "perception" module is also re-arranged to be positioned 'higher' in the hierarchy of decision making, since it can implement more cognitive-like properties. Both of them, including the re-defined "behaviour" modules can closely cooperate with the OCU modules, although the "proprioception" module can have the same importance for the "reflex" modules and "gait generation" modules. Additionally, "reflex" modules can closely collaborate with "sensor" modules and "motor controller" modules.

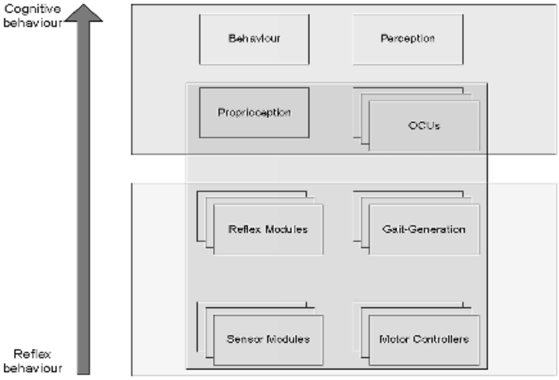


Figure 3: Schematic representation of a modified ORCA architecture

In the same way, the "gait generation" modules will also establish a closer cooperation with the "motor controller" modules, in order to successfully implement the walking behaviour of the robot. Generally, the redesigned ORCA architecture should contribute towards increasing the robustness and functionality of our robot system. For achieving the proposed emergent behaviour in future stages of our design, we are planning to implement soft computing techniques and self-learning algorithms. In order to incrementally build up the modified ORCA architecture, we have developed several steps towards achieving this goal. Namely we have started to experiment with designing the OCU modules as well as the "Reflex" and "Gait-Generation" modules. For such experiments, we have considered the case, in which the robot should decide autonomously upon some malfunction or complex interaction with the environment. Therefore, we have built a complex experimental setup, in order to analyze the delicate problems which the robot would face during its mission. A robot control architecture claiming to be robust has to deal with various sorts of unpredictable situations such as: mechanical problems, electronic problems, problems with sensors. In the last mentioned case, the robot is left to make proper decisions from analyzing the data from the still correctly functioning sensors and parts. An example of such a complex situation would be the following: The robot is having malfunction problems with sensors for visual/ultrasonic obstacle sensing. In the same time, the robot should detect if there is an obstacle in front of the legs or if one of the legs is damaged. In order to study such complex scenarios, we have developed an experimental setup, which is explained in the next section.

2.4 Experimental setup

The structure of our experimental setup, is represented in Figure 4. The control loop, consists of two parts: computer system, and a robot demonstrator. The reason why we have particularly chosen this kind of setup is that it enables us to measure and record in real-time many inputs from the robot simultaneously. These simultaneous inputs are essential for our observations, so we can make proper decisions and plans regarding to the development of our OCUs. The simultaneous inputs are: 18 servo positions, 18 servo motor currents, and 6 foot sensor values. For the measurements, we have been using National Instruments acquisition hardware, placed at the computer level, and its inputs connected with the robot parts. The measured values were further transported through our specially made software interface to our experimental information processing block running a fuzzy logic engine. The information processing part is intended to be a prototyping block, where we are going to build up the implementation of the OCUs. The information processing block communicates via a serial link with the robot's on-board controller, and therefore has direct control on the robot's actuators.

2.5 Performed experiments

We are aiming to develop self-adapting units which can cope with complicated fault situations. In those situations the robot has to rely on just on a few sensor's information streams. In our experiment we have tried to implement a robot's behaviour which is able in such hard situations to distinguish between a leg malfunction or perhaps an obstacle in its way. The robot is assumed to receive only inputs such as: servo current measurements, servo motor positions, analogue foot sensor measurements (see Figure 5). For the

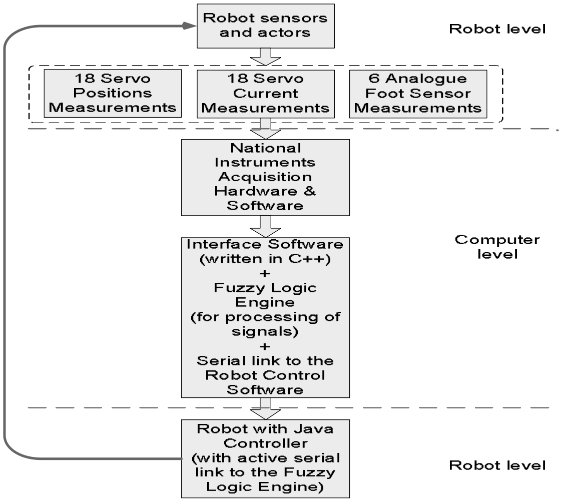
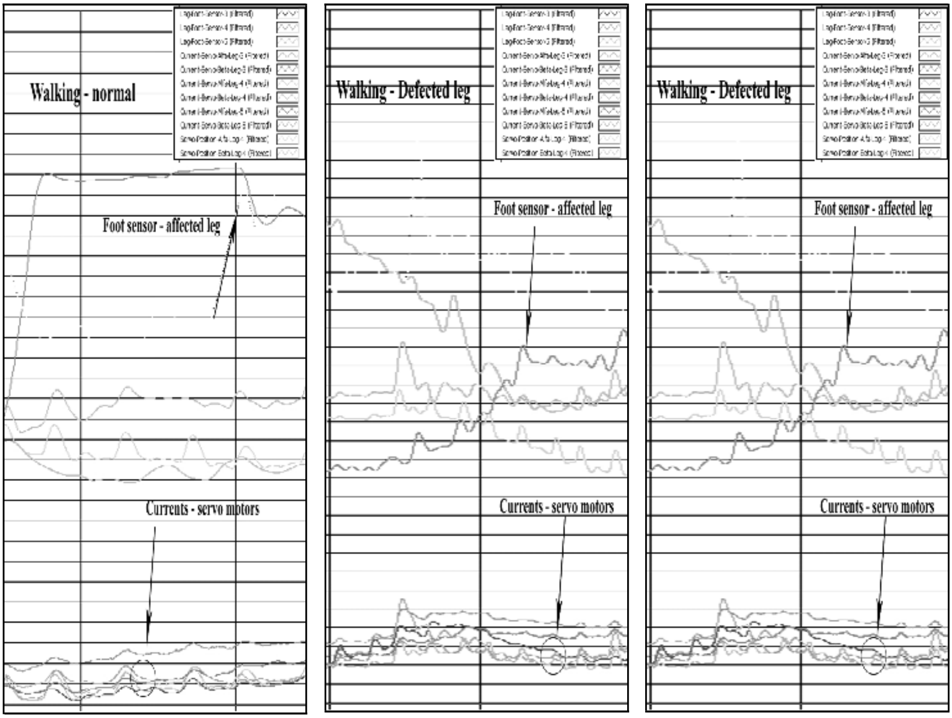


Figure 4: Structure of experimental setup

leg, we assume that several parts can induce its malfunction (screws, material bending, material fractures, servo gear problems, etc). In the further stages of research we are also planning to introduce an attribute “level of pain”, to characterize the transitory robot malfunction. The proposed emergent behaviour of the robot in the future stages of the research is supposed to be achieved with the following methodology. First, we assume the situation when the robot has an obstacle in front of it, and the robot is working with fully functional obstacle detection sensors. In several stages the robot should “learn” correlation factors between sensor values from the non-obstacle detection case and the actual presence of an obstacle. Later, when the robot will be in a malfunction situation, it should be able to employ the “learnt” correlation for self-assessment of the situation, indicating that something is wrong. Additionally, in that situation the robot should also be able to “learn” the correlation sensor factors from the new malfunction situation, and use it perhaps later. Therefore, we have started our experiments with observation on the correlation factors between the robot sensors information. For the first case, we have recorded and analyzed the typical sensor values during the normal walking of our robot, without obstacles in its way (Figure 5(a)). The second recorded case represents defected part of the leg, which was emulated with loosing a screw on one of the servo motors on the leg (Figure 5(b)).

The third recorded case represents a robot leg hitting some obstacle in its way (Figure 5(c)). In the center of all the figures, the positions of the servo motors are represented. At the bottom of the figures, the currents from the leg servo motors are shown. From the analysis of the plots we can observe strong correlation between the several sensor signals: foot sensor of the affected leg and currents of servo motors of other neighbouring legs of the affected leg. As we can see in the normal walking case (Figure 5(a)), the foot sensor shows a typical pattern when the leg is lifted during the walking phase. In case of a defected leg (Figure 5(b)), we can see a different pattern of the foot sensor signal when the damaged leg is tried to be lifted. Furthermore, the currents of servo motors of the affected leg, and the neighbouring legs, show the tendency of increasing their values. In the situation, where the leg is properly functioning and there is an obstacle in the way (Figure 5(c)), we can observe a different pattern of the signals. Namely, the foot sensor signal is showing oscillatory values, caused by the fact that the foot at that time hits the obstacle in front of it. Due to that, the current of the affected leg increases, but there is also an increased current at the neighbouring legs, which at that time, try to keep the robot in the same position. This is because the leg hitting the obstacle tries to reverse the movement of the robot in interaction with the object.



(a) Walking normal (b) Walking with defected leg (c) Walking with obstacle

Figure 5: Servo current measurements, servo motor positions, analogue foot sensor measurements

Being able to vaguely recognize the more common from an abnormal walking situation will help us to build a fuzzy processing engine for this purpose. This will be the main factor for properly implementing the correlation between the sensor data and self-estimation. In later phases we are going also to incorporate the learning characteristics, which will further enable promoting the emergence of the system.

3 Conclusion

We have proposed the modified ORCA software architecture, which will direct us towards development of more robust robotic systems. The redefining of the architecture was done on how the units will cooperate with each other and how to reach an emergent property. In order to implement the defined concept, we have built up an experimental testbed setup, which will be used for the development of the constituent units of the software architecture. In order to prove our concepts, we have conducted first experiments with a robot demonstrator. These experiments will guide us towards implementing the proposed self-x properties within our Organic Computing inspired software architecture for robot control.

References

[BK05] B. Bauer and H. Kasinger. Combining Multi-Agent-System Methodologies for Organic Computing Systems. In *Sixteenth International Workshop on Database and Expert Systems Applications Proceedings*, 2005.

[BMM05] W. Brockmann, E. Maehle, and F. Mösch. Organic Fault-Tolerant Control Architecture or Robotic Applications. In *4th IARP/IEEE-RAS/EURON Workshop on Dependable Robots in Human Environments*, Nagoya, Japan, 2005.

[Ger04] German Science Foundation (DFG). priority program SPP 1183 Organic Computing. http://www.organic-computing.de/spp, 2004.

[IBM01] IBM Research. Autonomic Computing. http://www.ibm.com/research/autonomic, 2001.

[MMS$^+$06] F. Mösch, Litza M., A. E. Sayed, Maehle E., Grosspietsch E. K., and Brockmann W. ORCA-Towards an Organic Robot Control Architecture. to appear in: International Workshop on Self-Organizing Systems, New Trends in Network Architectures and Services, Passau, Germany, 2006.

[MS04] C. Müller-Schloer. Organic computing: On the feasibility of controlled emergence. In *International Conference on Hardware/Software Codesign and System Synthesi*, 2004.

[Sch05] H. Schmeck. Organic computing - a new vision for distributed embedded systems. In *8th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing, ISORC*, 2005.

[SDK$^+$01] Josef Schmitz, Jeffrey Dean, Thomas Kindermann, Michael Schumm, and Holk Cruse. A Biologically Inspired Controller for Hexapod Walking: Simple Solutions by Exploiting Physical Properties. *The Biological Bulletin*, 200(2):195–200, 2001.

[SMM06] A. E. Sayed, F. Mösch, and Litza M. How the Six-Legged Walking Machine OSCAR Handles Leg Amputations. to appear in: Workshop on Bio-inspired Cooperative and Adaptive Behaviours in Robots, Rome, Italy, 2006.

[Wal04] K. Waldschmidt. Adaptive System Architectures. In *18th IEEE International Parallel and Distributed Processing Symposium Proceedings*, 2004.