Online-Diagnosis with Organic Computing based on Artificial DNA

Uwe Brinkschulte*, Roman Obermaisser†, Simon Meckel†, Mathias Pacher*
*Goethe University of Frankfurt am Main, Germany
†University of Siegen, Germany

***Abstract*—Organic Computing leads to significant advantages for complex dynamic systems like reduced development efforts, increased adaptability and robustness. However, for safety-critical systems which have to maintain functionality even in the presence of faults or failures (fail-operational) further properties are necessary. This includes the maintenance of the major core functionality even if non-redundant system resources fail, the Organic Computing runtime environment is harmed or the remaining resources are insufficient to maintain all services. These failure scenarios require semantic knowledge of the system combined with fault-diagnosis and adaption techniques to properly degrade and reconfigure the system.**

This paper highlights the research gaps towards active diagnosis based on artificial DNA and proposes solutions including semantic description methods, optimization algorithms for diagnostic models and adaptation techniques. Semantic description methods for Organic Cmputing systems with artificial DNA are the foundation for higher semantic-based failure detection and adaptation techniques. Diagnosis techniques for Organic Computing systems with artificial DNA can exploit the semantic descriptions to automatically build diagnosis models. Furthermore, these models can be optimized by evolutionary algorithms to improve their failure detection rates. Adaptation techniques modify the artificial DNA based on the recognized failures and the semantic description to realize the reconfiguration and degradation concepts.

***Keywords*—Organic Computing, artificial DNA, online-diagnosis**

I. Introduction

Embedded systems are growing very complex because of the increasing chip integration density, larger number of chips in distributed applications and demanding application fields e.g. in autonomous cars. Bioinspired techniques like self-organization and emergence are a key feature to handle this complexity. Organic Computing [8] has been introduced as a research focus to adapt such techniques to embedded systems. It leads to significant advantages like reduced development efforts, increased adaptability and robustness. However, for safety-critical systems which have to maintain functionality even in the presence of faults or failures (fail-operational) further properties are necessary. This includes the maintenance of the major core functionality even if non-redundant system resources fail, the Organic Computing runtime environment is harmed or the remaining resources are insufficient to maintain all services. These failure scenarios require semantic knowledge of the system combined with fault-diagnosis and adaption techniques to properly degrade and reconfigure the system.

The project presented in this paper addresses the corresponding research gaps by combining online-diagnosis with Organic Computing principles based on artificial DNA (ADNA) [3]. ADNA is a concept to include the entire building plan of the target system on each computing node. Using this building plan, the system is able to build up itself at runtime and to autonomously adapt to changes in the computation environment. The project encompasses three major research goals:

1) **Semantic description methods** will be added to the ADNA. In its current form, the ADNA is a blueprint of the system describing its components, parameters and interconnections. With a semantic description, the ADNA will be additionally aware of its own properties and the functional meaning of its components. This enables several new possibilities: the ADNA can judge its own quality, specifically modify itself, deliver information on failures and failure sources for the online-diagnosis and be the basis for automatic graceful degradation concepts.
2) **Diagnosis techniques** with a high level of automation will be developed based on the ADNA. The semantic description can be used to extract information on the relationships between physical values and signal properties. This information is utilized by synthesis algorithms to build diagnosis models for the detection and identification of conventional failures such as losses of sensors, actors, communication or computation resources, as well as specific failures of the Organic Computing system itself (ADNA failures).
3) **Adaptation techniques** will counteract the failures detected by the diagnosis. The ADNA will be changed at runtime to initiate system reconfigurations either by using alternative resources or by gracefully degrade the system to keep the safety-critical functionalities alive as long and as best as possible with the available resources.

II. Related Work

Self-organization has been a research focus for several years. Publications like [10] deal with basic principles of self-organizing systems, like emergent behavior and reproduction, amongst others. In the field of computer science IBM's and DARPAS's Autonomic Computing project [6] deals with self-organization of IT servers in networks. Several so-called self-X properties like self-optimization, self-configuration, self-protection and self-healing have been postulated. The German

Organic Computing Initiative as well as the ESOS workshop [1] dealt with self-organization for embedded systems. In [9] the concept of trust is used in self-organizing systems as a mechanism to enable agents to decide if they execute tasks for other agents. An agent will rather execute a task for an agent it has trust in than for an agent it has no trust relation to. So the trust concept helps agents in self-organization. However, this concept is completely different to our approach since we do not address trust but fault-diagnosis as well as semantic concepts. Self-aware computing systems (SACS) [7] are a new research topic and deal with two properties: (1) A SACS learns models by collecting information on itself and its environment and (2) it uses the model to plan actions according to its goals. The ADNA is a special type of a SACS, however, it provides a blueprint to build a system at runtime even if failures occur. In addition, it does not need abstract models on itself and the environment.

The trend towards open dynamic embedded systems requires diagnostic models to evolve with the systems. This is especially important in safety-critical application domains where the systems need to be fail-operational, i.e. they have to come up with strategies able to overcome faults or failures even in cases of no redundant hardware.

In the state-of-the-art numerous active fault-diagnosis techniques exist. Moreover, some methods for an automatic generation of diagnostic mechanisms have been described recently. [4] gives an introduction into the wide area of fault-diagnosis systems. The major fault-detection principles are based on mathematical models of the process to be diagnosed, on signal analysis techniques, or data-driven approaches that extract diagnosis models from an analysis of the process data. The subsequent fault identification techniques include classification methods as well as inference methods. The former include pattern recognition, statistical classification and artificial intelligence methods like neural network classifiers, the latter comprehend binary and approximate reasoning with predicate logic or fuzzy logic.

None of the existing diagnostic techniques directly supports open, dynamic and highly reliable Organic Computing systems. Problems are caused by the fact that the underlying diagnostic models do not support a dynamically changing system structure (e.g. fault trees or PMC) or by the fact that it is extremely computational-intensive to adapt the models in open real-time systems (e.g. existing rule-based fault inference methods). Also, current fault diagnosis methods using learning methods, classifiers or neural networks are not very variable if the underlying system changes and they mostly refer to specific use cases [5].

In order to guarantee essential system core functionalities even in case of loss of non-redundant system resources within open distributed embedded systems we propose the combination of Organic Computing based on artificial DNA and online-diagnosis principles, where a semantic description of the building blocks allows a self adaptation of the system. Moreover, an optimization algorithm implemented as a genetic algorithm, supports the character of the artificial DNA.

III. ARTIFICIAL DNA

The concept of ADNA has been chosen as the Organic Computing based technology for this work. This has several reasons: the goals described in Section I require the system to know its own structure, properties and the interaction of its components. The structure has to be modifiable at runtime to change and optimize diagnosis models and to overcome failures detected by diagnosis. Furthermore, the description of the system structure must not be a single-point-of-failure. Therefore, ADNA represents the best base, since each computation node knows the building plan of the entire system, the system dynamically creates itself at runtime from this building plan and therefore changes in this plan directly lead to changes in the system. The building plan describes the structure and components as well as their properties and interactions.

In the following, the concept of the ADNA is briefly introduced. For detailed information see [3] and [2].

The ADNA approach is based on the observation that, in many cases, embedded systems are composed of a limited number of basic elements, e.g. controllers, filters, arithmetic/logic units, etc. This is a well known concept in embedded systems design. If a sufficient set of these basic elements is provided, many embedded real-time systems can be completely built by simply combining and parameterizing these elements. Figure 1 shows the general structure of such an element. It has two possible types of links to other elements. The *Sourcelink* is a reactive link, where the element reacts to incoming requests. The *Destinationlink* is an active link, where it sends requests to other elements.

Each basic element is identified by a unique *Id* and a set of parameters. The sourcelink and the destinationlink of a basic element are compatible to all other basic elements and may have multiple channels.

The *Id* numbers can be arbitrarily chosen, it is only important that they are unique. Figure 2 gives an example for a PID controller which is often used in closed control loops. This element has the unique *Id* = 10 and the parameter values for P, I, D and the control period. Furthermore, it has a single sourcelink and destinationlink channel.

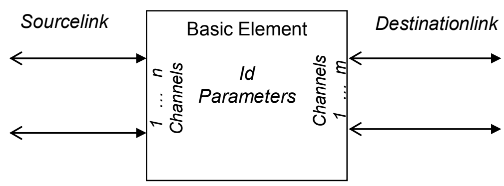


Fig. 1. Structure of a basic element (task)

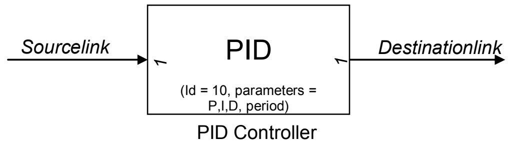
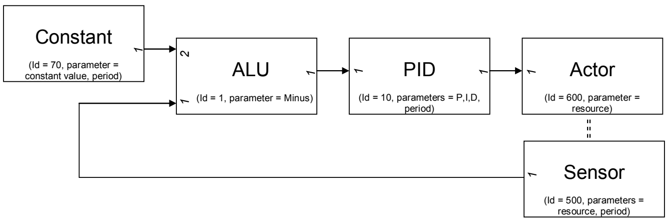


Fig. 2. Sample basic element

Embedded systems can be composed by using these basic elements as building blocks. Figure 3 shows a simple example of a closed control loop based on basic elements. An actor (defined by its resource id, e.g. a motor) is controlled by a sensor (also defined by its resource id, e.g. a speed sensor) applying a constant setpoint value.




```
1  = 70  (1:2.2) 100 25    // constant setpoint value, period 25 msec
2  = 1   (1:3.1) -         // ALU, control deviation (minus)
3  = 10  (1:4.1) 4 5 6 25  // PID (4, 5, 6), period 25 msec
4  = 600         1         // actor, resource id = 1
5  = 500 (1:2.1) 2 25      // sensor, resource id = 2, period 25 msec

ADNA line:       linenumber = id destinationlink parameters //comment
destinationlink: (destchannel:destlinenumber.sourcechannel...)
```

Fig. 3. A closed control loop consisting of basic elements and its netlist

If a sufficient set of standardized basic elements with unique *Id*s is available, an embedded system will no longer be programmed, but composed by connecting and parametrizing these elements. The building plan of the system can be described by a compact netlist containing the basic elements, its parameters and interconnections. This netlist can be stored in each processor of the system. It therefore represents a digitial artificial DNA which allows to partition and build the system at runtime. Detailed examples and a very memory efficient format to store an ADNA are presented in [2] and [3].

IV. Semantic description

The ADNA presented in the previous section describes the building blocks, parameters and interconnections of the target system. This allows a self-organized creation the the system at runtime and an autonomous adaption to changes in the computational environment (loss or accrual of computing nodes). Enriching the ADNA by a semantic description of its building blocks offers completely new possibilities: The system now becomes able to purposfully modify its own building plan since it knows the meaning and purpose of its building blocks due to the semantic description . This can be organized hierarchically on different levels of abstraction. It allows the system to optimize itself at runtime as soon as better building blocks become available. A goal-oriented evolutionary process of components becomes possible, i.e. dedicated addition, removement and replacement of building blocks as well as modification of interconnections. Since the semantic description contains the requirements and properties of building blocks, the quality of these modifications can be automatically evaluated. In case of failures, the semantic knowledge can be used to specifically modify the building plan for diagnostic purposes. Furthermore, failing building blocks can be automatically replaced by alternates as its properties

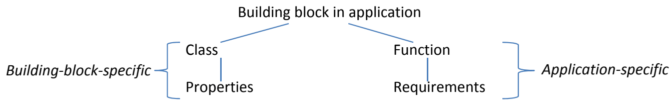


Fig. 4. Semantic description of an ADNA building block

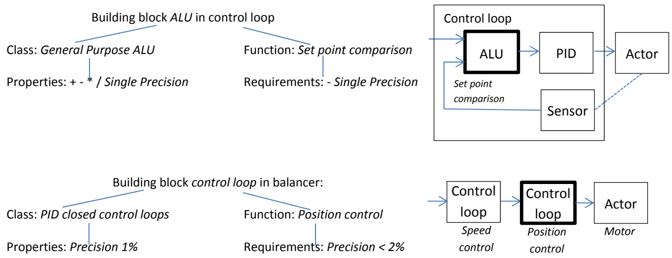


Fig. 5. Example of a semantic description in an ADNA

and requirements are well known from the semantic description. If the semantic description also contains the importance of components for the overall system functionality (e.g. the anti-locking-brake component in a car is more important than the entertainment component), an automatic graceful degradation becomes possible if too many failures prevent the maintainance all components.

Figure 4 shows the structure of the semantic descrition for a building block. It consists of a building block specific part which describes its class and properties. As soon as a building block is used within an application, an application specific part comes along. It describes the purpose and function of the building block in the application as well as the application's requirements for the building block.

Figure 5 shows a simple example for a hierarchic semantic description within an ADNA of a self-balancing vehicle (for more details of this vehicle see [3]). First, the building block specific class (general purpose ALU) and properties (available operations, precision, ...) of an ALU (arithmetic logic unit) building block are described. Application specific, this building block is responsible for the set point comparison in a closed control loop (see also Figure 3). Here, the application requirements (operation - (subtract), single precision, ...) for the building block are specified. One level above, this control loop itself becomes a building block having a class (PID closed control loop) and properties (control precision, ...). Application specific, this builing block is now part of a cascaded closed control loop to balance the vehicle. As an example, the position control component with its requirements for control precision are shown. The control loop for speed can be described in the same way. Often it is possible to automatically derive the requirements of building blocks on lower hierarchical levels from the builing blocks on higher levels (e.g. control precision $\rightarrow$ ALU precision).

Thus, an ontologie for the ADNA is created, which allows an automatic intervention for active online-diagnosis and runtime modifications. Functional areas for optimization and diagnosis can be purposefully identified, building blocks can be replaced (e.g. a PID control loop by a fuzzy control loop) and properties can be compared to requirements. If one of the requirements is the importance of a functional area (e.g. importance of position control > importance of speed display), this also forms the base for automatic graceful degradation.

V. Diagnosis and Adaptation Techniques

Building up an embedded system based on artificial DNA in combination with semantic descriptions of the basis elements inherently supports limited online-diagnosis and recovery strategies through the ability to communicate failures of computation nodes to neighboring nodes, which can take over lost tasks in these cases. The support for root-cause identification and specialized degradation functionalities, however, can be only realized if suitable diagnostic models are implemented. In our system architecture we consequently integrate the diagnosis into the Organic Computing environment.

In order to overcome the limitations of static diagnosis models and enable an adaptive character of the diagnosis to the self-establishing system, our work aims for *synthesis algorithms*, that automatically generate diagnosis models. For this, a synthesis algorithm maps continuous and discrete relationships between the semantic concepts onto instances of the physical quantities, state variables and resources of the system. This results in equations for plausibility checks of signals as part of a diagnostic model, e.g. differential equations, equations in the time and frequency domain as well as threshold values. In addition, different parameters such as signal-to-noise ratio, blur and the temporal dynamics are taken into account in the plausibility equations.

The diagnostic procedures are modeled as a directed acyclic graph. For its generation, methods from graph and network theory will be applied. In the graph, the input nodes are formed from directly accessible data such as sensor measurements or state variables. Throughout the graph, logical relationships and interdependencies between the building blocks are utilized for diagnostic checks (operations, tasks) to the end nodes which correspond to the identified root causes of faults. This yields fully integrated diagnosis techniques coded as an ADNA.

Due to the derivation of the diagnostic graph from the semantic descriptions, it offers significant optimization potential with respect to the diagnostic procedures themselves, parameters, calibration, integration of sensor fusion or placement of observation/measurement points.

In order to optimize the synthesized diagnostic graph with respect to diagnostic coverage on system level or confidence of diagnostic results, a *genetic algorithm* can be applied where the initial graph serves as a start solution for the algorithm. Such an evolutionary approach is especially advantageous in our work due to its ability to directly exploit the available semantic knowledge in the form of constraints or preparing new start solutions after a system modification. The genetic algorithm pursues several goals: First, evaluation of valid substitutes from the synthesis algorithm as the start solutions derived from the semantic descriptions only present some of many combination possibilities. Second, a continuous evaluation of redundant paths as a reaction to a potential failure of subsystems or components. Third, parameter optimization for a robust and performant diagnosis-ADNA.

Advanced concepts for a (graceful) degradation allow a system to keep core functionalities alive even if many processing units within the (distributed) system fail. We propose the usage of degradation concepts in combination with runtime modifications of the ADNA. In this way, for instance, certain sequences of the ADNA can have different priorities such that a sound reaction to a changed system configuration becomes possible. This goes beyond the inherent fault-tolerant possibilities of classical Organic Computing systems, where e.g. failing processing nodes can be compensated, but no program change is possible.

VI. Conclusion and Future Work

In this paper we motivate the combination of Organic Computing based on artificial DNA with adaptive online-diagnosis techniques to realize open distributed embedded systems, especially for safety-critical application domains, e.g. in the continuously growing field of industrial automation. Our approach implements a high-level semantic description layer for the embedded building blocks, enabling the system to establish itself according to the ADNA building plan. With the diagnosis and advanced degradation techniques, the system becomes able to supervise itself, grade its performance and, if necessary, overcome faults and failures through reconfigurations or degradations, keeping the core functionalities alive, even in the case of non-redundant hardware faults.

Future work comprises the development and implementation of the models and algorithms for the semantic description, diagnosis and adaptation with comprehensive experimental evaluations using realistic use cases.

References

[1] U. Brinkschulte, C. Müller-Schloer, and M. Pacher (editors). Proc. of the Workshop on Embedded Self-Organizing Systems, USA, 2013.

[2] Uwe Brinkschulte. An artificial dna for self-descripting and self-building embedded real-time systems. *Concurrency and Computation: Practice and Experience*, 2015.

[3] Uwe Brinkschulte. Prototypic implementation and evaluation of an artificial dna for self-descripting and self-building embedded systems. *Springer EURASIP Journal on Embedded Systems*, February 2017.

[4] Rolf Isermann. *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Science & Business Media, 2006.

[5] Rolf Isermann. *Fault-diagnosis applications: model-based condition monitoring: actuators, drives, machinery, plants, sensors, and fault-tolerant systems*. Springer Science & Business Media, 2011.

[6] Jeffrey O. Kephart and David M. Chess. The Vision of Autonomic Computing. *IEEE Computer*, pages 41–50, January 2003.

[7] Samuel Kounev, Jeffrey O. Kephart, Aleksandar Milenkoski, and Xiaoyun Zhu. *Self-Aware Computing Systems*. Springer Publishing Company, Incorporated, 1st edition, 2017.

[8] Christian Müller-Schloer, Hartmut Schmeck, and Theo Ungerer. *Organic Computing – A Paradigm Shift for Complex Systems*, volume 1 of *Autonomic Systems*. Springer Basel, 2011.

[9] Wolfgang Reif, Gerrit Anders, Hella Seebach, Jan-Philipp Steghöfer, Elisabeth André, Jörg Hähner, and Christian Müller-Schloer Theo Ungerer, editors. *Trustworthy Open Self-Organising Systems*. Springer International Publishing Switzerland, 1st edition, 2016.

[10] Randall Whitaker. Self-Organization, Autopoiesis, and Enterprises, 1995. https://tinyurl.com/26j85ej.