FRANZ J. RAMMIG

BIOLOGICALLY INSPIRED INFORMATION TECHNOLOGY: TOWARDS A CYBER BIOSPHERE[1]

1 Introduction

Engineers are interested in building highly efficient, highly reliable, and highly deterministic systems; they are interested to keep their systems completely under control under all potential circumstances. For this purpose the Embedded Systems community, especially the real-time researchers have developed sophisticated solutions: deterministic real-time scheduling techniques, schedulability analysis, collision-free communication protocols, time-triggered architectures, formal proof techniques, just to mention some of them. However, serious restrictions have to be respected when following such approaches. In most cases a closed world has to be assumed, all parameters must be controllable. This especially restricts the possibilities to distribute such systems, as in distributed systems synchronization among the components may cause problems. In fact, whenever there is a finite speed of communication which is "slow" with respect to the operational speed of the individual components, the effect of "relativity" is present.

Adapting inspirations from the biosphere, a world that seems to follow completely different approaches, appears to be a strange idea at the first glance. On the other hand engineers are impressed by the robustness of extremely complex biological systems. Robustness in this context means short-term robustness. Of course, in long terms any constituent of the biosphere shows a dramatically instable behavior, from appearance of a species or an instance of a species up to its disappearance. In short terms, however, the entire system and its constituents are remarkably stable. A human, made of billions of cells, interacting in a highly sophisticated manner, is continuously exposed to billions of enemies (antigens) which change their attacking strategies rapidly and in a non predictable manner. By simple *Mean Time Between Failures* (MTBF) calculations one would conclude that a human's lifetime should not exceed some hours. However, such a complex system survives in a hostile environment for up to 100 years or even longer. The same can be said for any

[1] This paper extends Franz J. Rammig, "Cyber Biosphere for Future Embedded Systems", in: Uwe Brinkschulte/Tony Givargis/Stefano Russo (eds.), *Software Technologies for Embedded and Ubiquitous Systems* (Lecture Notes in Computer Science Vol. 5287), Berlin, Heidelberg, 2008, p. 245–255.

kind of complex bio-conglomerates. So, biological systems have proven to be extremely robust even in dynamically changing hostile environments.

Of course engineers also are able to design highly complex systems. A today's complex microprocessor which is a sophisticated digital *System on Chip* (SoC) comprises a billion of transistors as well, and it runs reliably for a long time. Giant software systems like telephone switching systems are very reliable as well. What can be questioned, however, is the stability and robustness in case of dramatically changing environmental conditions or in case of unforeseen hostile circumstances. Classical engineering always makes rather strict assumptions concerning the environment, in most cases it even assumes the environment to be adapted to the operational conditions of the system to be built. For designing a car, e.g. the existence of well paved roads is assumed. Of course, biological systems can handle unforeseen situations also only to a certain amount. In cases beyond this level of flexibility the respective instance or even an entire species disappears. However, it seems that this limit of biological flexibility is much wider than in conventional technical artifacts. From this observation it does not surprise that one of the most stable, most robust and most adaptive complex technical artifact is the internet. In fact the internet follows a couple of basic principles of biological systems like distributive design, postponing decisions and actions into the operational phase, self-organization, emerging redundancy, just to mention some of them.

Common to the highly complex systems of the future are the following key characteristics:

- complex volatile networks in which components cooperate as well as possibly compete,
- decentralized control and components acting autonomously,
- an unobservable global system state and thus components with only local knowledge,
- optimization of own benefits being the driving force of a component's cooperation,
- adapting to and learning from environmental changes as a universal ability of components,
- limited availability of resources combined with security and safety requirements.

Characteristic No. 1 originates from the sheer fact of complexity. Above a certain threshold a conglomerate cannot be kept stable concerning its structure and behavior. Components will appear and disappear and it will not be able to keep always the same rules concerning these components' behaviors.

Characteristic No. 2 is closely related to the first one. If the complexity of a system grows towards infinity any attempt of global control would evolve towards infinite complexity, which in turn would convert it into a system of infinite complexity and so on. It is not surprising that e.g. centrally controlled economies tend to fail. So the individual components are forced to act autonomously to a certain amount, giving them the freedom to do this in cooperative or non cooperative manner.

It is the relativistic property of large distributed systems that results in characteristic No. 3. In a relativistic system a global state cannot exist as observation and even definition of a global state would imply exact synchrony of all local time lines. And exact synchrony cannot be achieved with finite communication speeds. This also implies that components of such a system can be aware only of their own state and the states of objects in their near vicinity. Near vicinity is defined as the set of components where communication is possible within a time period which is small with respect to the relevant speed of operation (in physics these are sub-systems where the error made when applying Newton's mechanics is negligible).

An unobservable global state brings us immediately to characteristic No. 4. As components of such a complex system cannot observe a global state, they also cannot precisely be aware of a global objective. A pragmatic way to overcome this problem is the assumption when trying to optimize the individual benefit (in whatever way this can be determined), the resulting “global” behavior will not be too far away from a theoretical global optimum.

Such systems can hardly show robust behavior if the constituents would be unable to adapt to changing environmental conditions. This makes learning capabilities necessary as undirected changes rarely result in beneficial adaptation (characteristic No. 5). New properties emerge while the network's components adapt to and learn from other components. This in turn makes resulting learning necessary.

Characteristic No. 6 is different in nature from the other ones. It is just a general restriction which we always find in real systems. There are never unlimited resources. Without providing safety, robustness would be harmed by the constituent components themselves, without security there would be no means to provide robustness in presence of potentially hostile other components. And as selfishness has to be the driving force of the individual components hostile behavior will always be present.

These fundamental characteristics raise a number of new research questions that need to be addressed in order to achieve any progress in this area. All the mentioned properties are present in biological systems as well. Therefore, it seems to be attractive looking for inspirations in this domain. Biological systems seem to follow optimal strategies (or at least near-optimal ones) in the presence of partial or even unreliable information. Biological components are able to “decide” which information is relevant and which need not be considered. They follow “algorithms” reaching stable, robust, and desirable behavior in a distributed network. Biological entities find out about their right option of interaction with cooperating other components or even such ones which they are in competition with. Nature “invented” clever, adaptive, and efficient communication principles. All this is done under restricted resources and even in case of failing parts. Nature transformed most of the decisions and actions into the operational phase of biological artifacts which results in highly adaptable systems. These systems reflect on both their own and their environment's behavior and consequently change themselves. Nature provides techniques that to a certain amount can ensure the correctness of emergent volatile systems.

To sum up: Highly complex systems behave like global economy. By their tradition engineers tend to organize their artifacts in the way of planned, centrally controlled economy. Nature is an economy driven by free enterprise of selfish agents. Such economies may be far away from optimality, they tend to locally show nondeterministic and undesired behavior at certain points of time (economic crisis). But they seem to be extremely robust on the long term. In this paper we would like to provide some hints why it could be wise for engineers to accept a certain amount of free economy as well.

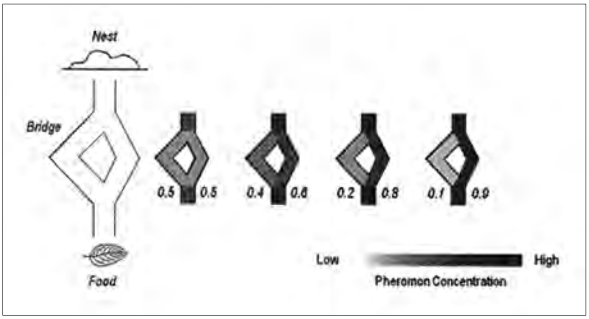
2 Ant Colony Algorithms

The total biomass of ants on earth is more or less the same as the biomass of mankind. Ants can be seen as one of the most advanced examples of social bio-systems. Ant colonies can be interpreted as a specific kind of an organism, forming an interesting compromise between simple swarms of single cell life and highly organized multi-cell systems (e.g. mammals) where most cells are fixed at a specific location and play a specific role. Differently from these two extremes, in an ant colony the individual constituent (an ant) is a multi-cell object, mobile, intelligent to a certain degree, but closely embedded into a global collaborative scheme. *Ant Colony Optimization* (ACO) is a cooperative meta-heuristic being successfully applied to various combinatorial optimization problems. Ants tend to find the shortest path from their nests to a food source in a relatively short time. In order to achieve this, they communicate in an indirect manner, called *stigmergy*. Moving ants deposit traces of pheromone on their trail. On the other hand, ants have the tendency to follow trails which are marked by pheromone. This establishes a positive feedback which makes a marked trail even more attractive. Evaporation of pheromone establishes a negative feedback. When alternative trails are chosen randomly in the beginning, with high probability the pheromone level of a path is inverse proportional to the path's length (see Fig. 1).

Marco Dorigo established this field of meta-heuristics. Dorigo et al.[2] was the first paper to apply ACO to graph-related optimization problems like the *Traveling Salesman Problem* (TSP). A more general theory has been devel-

[2] Marco Dorigo/Vittorio Manniezzo/Alberto Colorni, “The Ant System: Optimization by a Colony of Cooperating Agents”, in: *IEEE Transactions on Systems, Man, and Cybernetics, Part B* 26, 1 (1996), p. 29–41.

oped in his book *AntColony Optimization*[3], proceedings of dedicated conferences have been published as well.[4] An excellent introduction into biologically inspired techniques in sensor networks, including application of ACO is given by Dressler.[5] Here just some aspects, covered by research in Paderborn, shall be discussed.



1 – Pheromone estimation und adaptation (time line: left to right)

Ditze[6] and Ditze/Becker[7] describe the application of Dorigo's basic approach to the scheduling problem of MPEG streams via the 802.11e EDCA. For this purpose, the precedence-constrained MPEG scheduling has to be mapped onto a directed graph, expressing the precedence relationships of MPEG *Groups of Pictures* (GoP). This results in a cyclic graph consisting of the various I-, P-, and B-frames contained in the GoP being represented as nodes and the precedences as directed edges. A feasible solution represents a schedule of MPEG

[3] Marco Dorigo/Thomas G. Stützle, *Ant Colony Optimization* (A Bradford Book), Cambridge, MA, (et al.), 2004.

[4] Marco Dorigo/Mauro Birattari/Christian Blum, et al. (eds.), *Ant Colony Optimization and Swarm Intelligence. 4th International Workshop, ANTS 2004: September 5–8, 2004, Brussels, Belgium; Proceedings* (Lecture Notes in Computer Science Vol. 3172), Berlin, 2004, and Marco Dorigo/Luca Maria Gambardella/Mauro Birattari, et al. (eds.), *Ant Colony Optimization and Swarm Intelligence. 5th International Workshop, ANTS 2006: September 4–7, 2006, Brussels, Belgium; Proceedings* (Lecture Notes in Computer Science Vol. 4150), Berlin, 2006.

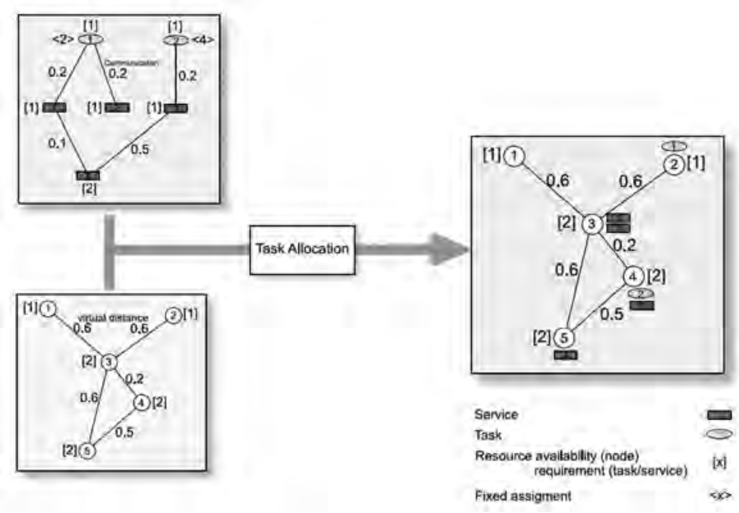
[5] Falko Dressler, *Self-Organization in Sensor and Actor Networks* (Wiley Series in Communications Networking Distributed Systems), Chichester (et al.), 2007.

[6] Michael Ditze, "Evaluation of an Ant Colony Optimization Based Scheduler for the Transmission of Multimedia Traffic in the 802.11e EDCA", in: *Proceedings of the 3rd ACM Workshop on Wireless Multimedia Networking and Performance Modeling: October 22nd 2007, Chania, Crete Island, Greece* [held in conjunction with the 10th ACM/IEEE International Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM)], New York, NY, 2007, p. 9–15.

[7] Michael Ditze/Markus Becker, "An Improved Adaptive ACO Meta Heuristic for Scheduling Multimedia Traffic Across the 802.11e EDCA", in: Reza Rejaie/Roger Zimmermann (ed.), *Multimedia Computing and Networking 2008: January 30–31, 2008, San Jose, California, USA; Proceedings, Electronic Imaging, Science and Technology*, Bellingham, WA, 2008, p. 6818–6820.

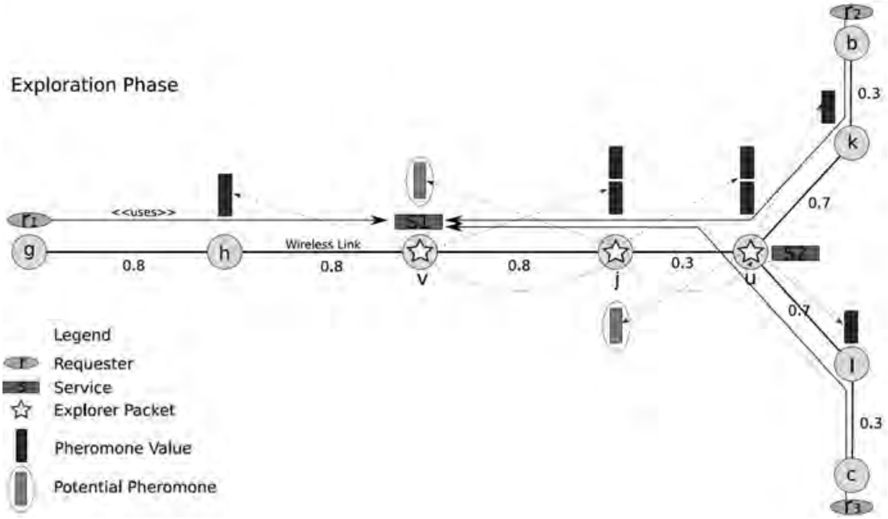
frames where each frame is expected to be transmitted within its (previously defined) delay bounds. On such a graph a colony of π ants is deployed. An ant of such a colony sitting on a “border” node of a partially feasible schedule selects an edge from this node to an attainable node according to a probabilistic function as in Dorigo's original work. A tour is said to be completed if all π ants of a colony have returned to the initial I-frame. Then the best selected path is evaluated by counting the number of timely scheduled frames. On each edge of this path the pheromone values are updated. The updated value is proportional to the ratio of the achieved solution and the optimal one (all frames of the GoP scheduled timely). As a result, near optimal solutions that entail higher concentration of pheromone will have a higher impact on the edge selection process in subsequent tours. In experiments this algorithm turned out to be nearly as efficient (concerning needed computation time) as a dedicated scheduling algorithm designed at our institute by the same author. However, it showed a much more robust behavior with respect to rapidly changing load and transmission distortions.

In our group we developed a fully distributed operating system, i.e. the various services offered by the OS are distributed over a network of nodes. Compared with providing all services at all nodes this results in a substantial reduction of memory footprint. However, additional communication costs occur whenever a service is requested which is not locally available. This implies the need for an optimized service distribution. See Fig 2. for the optimization problem. A problem graph describing service requests by applications has to be mapped onto a platform graph describing the nodes and their interconnection links in such a way that the global communication costs are minimized.



2 – Service allocation in a fully distributed operating system

Heimfarth/Janacik[8] apply ACO to this problem. Of course, the objective in this case is not to minimize path lengths but weighted communication costs. These costs are calculated in this case by path length measured in hops, link quality using a sophisticated combined link metrics, and the access frequency, i.e. the frequency a certain path is used due to requesting the considered service. In their use of ACOs services are the equivalent of food sources and service locations are the equivalent of shortest paths. Calls made by the requesters are modelled by ants while requesters are modelled by nests. Wireless links in between the platform nodes form the paths which the ants can use for movement. As common to ACOs, ants leave pheromone on the nodes while the requests are being routed to the destination service. This pheromone evaporates over time. In Fig. 3 this application of ACOs is illustrated. It shows a situation where service *S1* residing on node *v* is requested by applications r_1 on node *g*, r_2 on node *b*, and r_3 on node *c*. In this situation the service *S1* has a tendency migrating to node *u*. As this one might be overloaded due to service *S2* residing on this node, a migration to node *j* is selected. In addition (not shown in this example) geographically related paths are handled in such a way that they bundle attracting force into their direction.[9]



3 – Optimization of service allocation in a fully distributed operating system

[8] Tales Heimfarth/Peter Janacik, “Experiments with Biologically-Inspired Methods for Service Assignment in Wireless Sensor Networks”, in: Mike Hinchey/Anastasia Pagnoni, et al. (eds.), *Biologically-Inspired Collaborative Computing: IFIP 20th World Computer Congress, Second IFIP TC 10 International Conference on Biologically-Inspired Collaborative Computing: September 8–9, 2008, Milano, Italy*, Vol. 268, New York, NY, 2008, p. 71–84.

[9] For more details, see ibidem.

Up to now homogeneous systems have been considered. In our analogy this means ant colonies where all ants are alike. Nature, however, also "developed" colonies with heterogeneous populations. Social insects like *Pheidole Rea* form such colonies of heterogeneous objects. This allows a dedicated division of labor. Larger ants (the "majors" serve to protect the nest against enemies while the smaller ones ("minors") are doing daily work. In fact this division of labor reduces the overall energy need of the entire colony. Large ad hoc networks can be clustered following an approach based on division of labor in such colonies of social insects. The basic idea in this case is to treat each node of an ad hoc network either as a "major" ant or a "minor" one. A major represents a cluster head responsible for wider area communication. This means a higher workload while the minors are member nodes of clusters. The main power of the approach is originating from the built-in elasticity. Both types of species have a certain threshold to become major or minor. On the other hand they are stimulated by received signals. Whenever the strength of such signals is above a certain threshold the role of a major may change to a minor or vice versa. Typical stimuli signals are signal strengths of received messages, frequency of received messages, etc. Thresholds are established e.g. by the power reserve of a node. A cluster head with flattening power resources has a tendency to become a minor (member node), an "isolated" member node to become a cluster head.[10] This approach again shows enormous robustness against rapidly changing situations.

3 Artificial Hormone Systems

All biological systems can be seen just as a collection of individually operating cells which follow some collaborative principle of operation, based on some communication means. Electrical signaling via the nerve system constitutes a means of directed communication in the sense of single-cast or multicast. Controlled and centrally coordinated actions like contraction of specific muscles to enable movement may serve as an example. In other situations when an extremely high number of potentially receiving cells have to be addressed and if those cells are widely spread across a body, a multi-cast communication scheme is desirable. In bio-systems this is carried out by means of the hormone system which can be interpreted as a way of biological broadcasting. Specific chemicals are generated by the sending instance and cause reactions on the side of receiving cells. It is essential that different receiving cells can react in a different and specific manner. This specific reaction may

[10] For more details, see Tales Heimfarth/Peter Janacik/Franz J. Rammig, "Self-Organizing Resource-Aware Clustering for Ad Hoc Networks", in: *Proceedings of the 5th IFIP WG 10.2 International Conference on Software Technologies for Embedded and Ubiquitous Systems,* Berlin, Heidelberg, 2007, p. 319–328.

depend on cell type or even on a specific cell instance and its current environmental setting. Even the set of hormones may be specific for the different cells. Hormones unknown to a certain receiver are just ignored. So the intended communication is established only between processing elements that share a joint reservoir of hormones. By this concept multi-cast can be implemented easily. In addition it is a kind of “power line communication” as blood circulation serves as carrier for information transport. This simple basic principle thus can be tailored in numerous ways to result in the desired behaviors.

Von Renteln et al.[11] discuss an approach to apply concepts of artificial hormone systems to task allocation on heterogeneous processing elements. In their approach each of the processing elements and the tasks to be assigned may secrete “hormones” or may react on receiving ones. This approach strictly follows a decentralized approach. Each processing element may have an individual rule set for the secretion of hormones or how to react on receiving certain ones. The only common rules are given by some agreement what hormones to be used. In their approach the authors implement a distributed feedback controller by means of two principle types of hormones, so called accelerators (positive feedback) and so called suppressors (negative feedback). The first ones are sent out to indicate the willingness of a processing element to attract additional tasks, the second one to indicate the inability to do so. The approach results in a couple of self-x properties: *self-configuration* as there is no central control, *self-optimization* as there may be included rules to re-open the assignment “market” periodically or stimulated by some events, *self-healing* as a failing task or processing element is no longer sending hormones and by this disturbs the equilibrium which causes some re-allocation. The authors have built a flexible simulation environment which allows them to experiment with a variety of parameter settings. Brinkschulte/von Renteln[12] additionally examine very carefully conditions for stability in such systems.

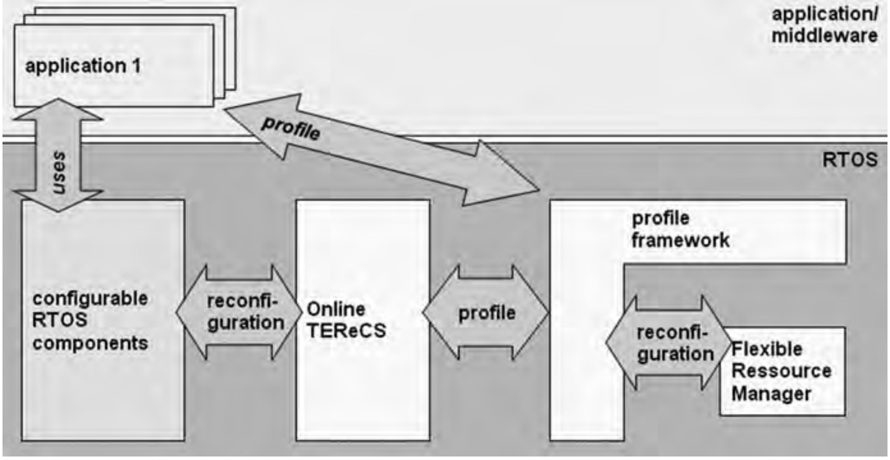
Stress response is a special version of a hormone system. The *“Fight-or-flight*”-theory by Walter Cannon[13] describes the reaction of humans and animals to threats. In such stress situations specific physiological actions are taking place by the sympathetic nervous system of the organism as an automatic regulation system without the intervention of conscious thought. For example, *epinephrine,* a hormone, is released which causes the organism to release energy to react on the threat (fight or flight). This concept is adopted to

[11] Alexander von Renteln/Uwe Bringschulte/Michael Weiss, “Analyzing the Behavior of an Artificial Hormone System for Task Allocation”, in: Juan González Nieto (ed.), *Proceedings of the 6th International Conference on Autonomic and Trusted Computing, ATC 2009: July 7–9, 2009, Brisbane, Australia,* Berlin, Heidelberg, 2009, p. 47–61.

[12] Brinkschulte/Renteln/Weiss (2009), Analyzing the Behavior.

[13] Walter B. Cannon, *Bodily Changes in Pain, Hunger, Fear and Rage: An Account of Recent Research into the Function of Emotional Excitement*, reprint of 2nd ed., New York, NY, 1929.

control the on-line reconfigurable real-time operating system DREAMS[14] (*Distributed Real-Time Extensible Application Management System*) which has been developed by our group. This RTOS (*Real-Time Operating System*) is able to manage system tasks and user tasks in the form of different "profiles" by means of a special resource manager FRM (*Flexible Resource Manager*)[15] (see Fig. 4).



4 – Flexible Resource Manager FRM

DREAMS has been tailored to the special demands of self-optimizing applications. The manager tries to optimize the resource utilization at run-time. The optimization includes a safe over-allocation of resources, by putting resources that are held back for worst-case scenarios by tasks at other tasks' disposal. The interface to the FRM is called *Profile Framework*. By means of the Profile Framework the developer can define a set of profiles per application. Profiles describe different service levels of the application, including different quality and different resource requirements. All states belonging to one profile build the state space that can be reached when the profile is active. The different profiles can be assigned to specific emergency categories using a generic monitoring concept for self-optimizing systems. The intent is to protect tasks systematically against hazards or faults. These hazards or faults might result from their self-optimizing behavior themselves, but self-optimizing behavior

[14] Recently a new version of DREAMS has been created, called *Organic Reconfigurable Operating System* (ORCOS). It can be downloaded from https://orcos.cs.uni-paderborn.de

[15] Simon Oberthür/Carsten Böke, "Flexible Resource Management – a Framework for Self-Optimizing Real-Time Systems", in: Bernd Kleinjohann/Guang R. Gao/Hermann Kopetz/ Lisa Kleinjohann/Achim Rettberg (eds.), *Design Methods and Applications for Distributed Embedded Systems: IFIP 18th World Computer Congress, TC10 Working Conference on Distributed and Parallel Embedded Systems (DIPES 2004): August 22–27, 2004, Toulouse, France*, Vol. 268, Boston, MA, (et al.), 2004, p. 177–186.

can also support the re-allocation of resources to handle threats. The concept distinguishes four different emergency categories:

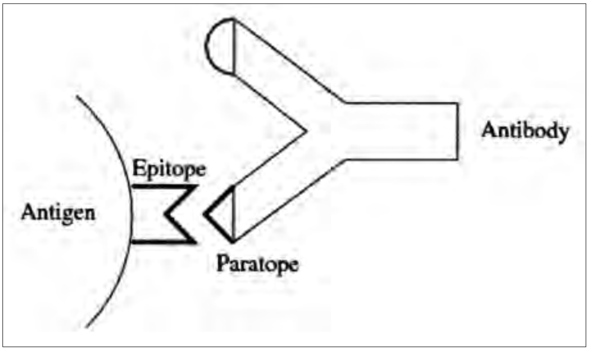
1) The system operates regularly and uses its self-optimization for the major system objectives.
2) A possible threat has been detected and the self-optimization is not only used to optimize the behavior but also to reach system states, which are considered to be safer than the current one.
3) A hazard has been detected that endangers the system. Fast and robust countermeasures, like a reflex, are performed to reach a safer state (1 or 2).
4) The system is no longer under control; the system must be immediately stopped or a minimal safe-operational mode must be warranted, to minimize damage.

The artificial hormone system is applied to ensure that the system can provide more resources to enable more efficient countermeasures whenever it experiences entering emergency category 2. The idea is, when a task of the system detects a threat for the system it releases virtual epinephrine. This distributed epinephrine forces non-critical tasks into a profile with lower resource consumption. By this, resources are released and this permits the critical task to handle the threat more appropriately by switching into a specific emergency handling profile which usually is more resource-hungry. The virtual epinephrine carries the information how much additional resources the epinephrine secreting task requires to activate its threat-handling profile. It is assumed that all tasks are sorted according to their safety critical nature. Like the cardiovascular system of an organism the resource manager broadcasts the epinephrine to the tasks. Tasks with the lowest safety level have the shortest reaction time. When the epinephrine is injected into such a task it can react by switching into a special profile with lower resource requirements. The task then updates the information inside the epinephrine how much resources are still required. This updated epinephrine then is secreted again, by this over-writing the hormone already received by tasks at higher safety levels which react more slowly. By this technique finally every task has information about the threat and can react accordingly. The complexity of this process is linear with respect to the number of tasks. The reaction of the tasks to the epinephrine (“consuming” it by update) is done in a short, constant time.[16]

[16] Details can be found in Holger Giese/Norma Montealegre/Thomas Müller/Simon Oberthür, “Acute Stress Response for Self-Optimizing Mechatronic Systems”, in: Yi Pan/Franz J. Rammig, et al. (eds.), *2006 – Biologically Inspired Cooperative Computing*, Vol. 216, p. 157–167.

4 Artificial Immune Systems

Immunocomputing intends to establish another kind of computing. The main idea is to copy the immune system's ability to identify abnormal objects ("*antigens*") with high separation precision and to attack such antigens using adapted means ("*antibodies*") in an extremely efficient manner. All this is done in a distributed but interlinked manner and is quickly adapted to varying situations (occurrence of previously unknown antigens) by a sophisticated learning ability. As biological immune systems are based on chemical reactions of proteins, immunocomputing is based on the "*formal protein*" as its basic element. A protein is an essential component of organisms and participates in every process within cells. It is composed of amino-acids arranged in a linear chain. Proteins constitute *epitopes* present in antigens and antigen presenting cells. Proteins constitute also *paratopes* present in antibodies. Both are made of around 10 amino-acids. An epitope is the minimum molecular structure that is able to be recognized by the immune system. One epitope matches with a paratope in molecular recognition (see Fig. 5). An antigen presenting cell is a cell that has digested an antigen and presents in its surface a respective epitope. The 3D shape or tertiary structure of the epitope is recognized by a paratope. It means an epitope is a kind of surface protein. That is why proteins will be seen as the basic element in immunocomputing.



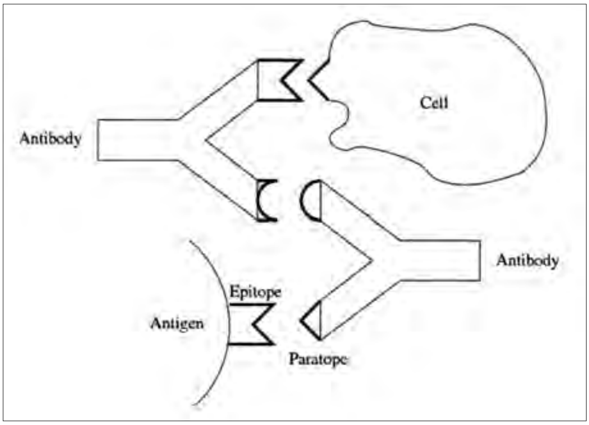
5 – Epitope of an antigen binding to a paratope of an antibody

Cytokines are introduced as an additional concept into immunocomputing[17] to establish collaboration. In biological systems cytokines are groups of proteins secreted by many types of cells. Each cytokine binds to a specific cell's surface receptor signaling a specific action i.e. differentiation into plasma cells,

[17] Alexander O. Tarakanov/Segei V. Kvachev/Alexander V. Sukhorukov, "A Formal Immune Network and Its Implementation for On-line Intrusion Detection", in: Vladimir Gorodetsky/ Igor Kotenko/Victor Skormin (eds.), *Computer Network Security* (Lecture Notes in Computer Science Vol. 3685), Berlin, Heidelberg, 2005, p. 394–405.

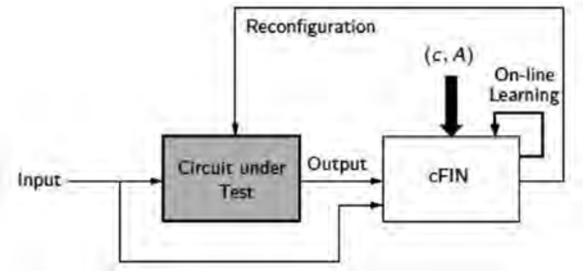
antibody secretion, or cell death. They bind also through own receptors constituted from proteins, too.

The basic entities in a biological immune system and therefore also in immunocomputing are so-called *B-cells*. B-cells in the immune system secrete antibodies, i.e. the actuators of immune reaction. On the other hand they also secrete cytokines in order to signal something to another cell. This introduces a positive feedback into the immune system. In Artificial Immune Systems, a B-cell will be taken as a generic cell V_i with two components expressed by $V_i = (c_i, P_i)$ where $c_i \in \mathbf{N}$ represents a cytokine (action to be carried out) and $P_i \in R^q = ((p_1)_i, ..., (p_q)_i)$ is a point in a q-dimensional space. P lies within a cube $max\{| (p_1)_i |, ..., | (p_q)_i |\} \leq 1$. It represents a protein transformed into the so-called FIN (*Formal Immune Network*) space (see Fig. 6). In biological terms it represents an antigen binding site (antigen detection) of an antibody or, simplifying, an antibody.



6 – Immune Networks

We applied cFIN (*cytokine FIN*) to build self-repairing reconfigurable hardware arrays (FPGAs), following a BIST (*Built-In Self-Test*) approach. The circuit under test receives a test pattern and the response is evaluated by means of cFINs. In this case, an antibody represents the expected output, transformed into the FIN space. An antigen is the response of the circuit under test. A cytokine represents the action to be taken for fault recovery purposes. It is important that the system has to be trained beforehand using a training matrix $V(c,A)$. $A = A_1,\ldots,A_n$ with $A_i = (Input_i, Output_i, Stimuli_i, State_i)$ is a matrix with information about expected responses under defined input patterns. Each expected or unexpected response then is linked to an action expressed by c with $c_i = (self_i, action_i)$. The first component indicates the differentiation between *self* and *not self*, the second one identifies the action to be taken. Using the cytokine communication system, on-line learning can take place during operation[18] (see Fig. 7).



7 – Essential part of cFIN-based self-repairing FPGA

5 Discussion

The three approaches presented here are just examples of a broad potential when getting inspiration from nature. Of course, these approaches include much more sophistication than the simple principles presented here just to initiate a discussion. In any case it is wise to collect more profound knowledge about biological systems before gaining real benefit out of them for engineering disciplines. Even the three sketches presented here, however, show some interesting similarities. So it seems that there are some basic principles worth to be studied as inspirations.

Inspiration 1:
Follow a cell-based approach.

Nature "invented" life by "inventing" cells. Cells include intelligent I/O (cell membrane), static code (nucleus), a reproduction mechanism, a complete chemical plant (cell plasma), energy management, and a motion mechanism. I.e. cells are self-contained with respect to information, energy, and material flow. For billions of years life did exist solely in form of single cell entities. So whatever emerged as biological system remains a collection of individual

[18] Details can be found in Norma Montealegre/Franz J. Rammig, "Immuno-Repairing of FPGA Designs", in: Mike Hinchey/Anastasia Pagnoni/Franz J. Rammig/Hartmut Schmeck (eds.), *Biologically-Inspired Collaborative Computing: IFIP 20th World Computer Congress [WCC 2008], Second IFIP TC 10 International Conference on Biologically-Inspired Collaborative Computing: September 8–9, 2008, Milano, Italy,* New York, NY, 2008, p. 137–149. For general readings on immunocomputing, see: Leandro N. de Castro/Jonathan Timmis, *Artificial Immune Systems. A New Computational Intelligence Approach*, London (et al.), 2002, and Jonathan Timmis/Peter J. Bentley/Emma Hart, *Artificial Immune Systems. Second International Conference (ICARIS): September 1–3, 2003, Edinburgh, UK; Proceedings* (Lecture Notes in Computer Science Vol. 2787), Berlin (et al.), 2003.

cells, a collection of cells which may cooperate very closely, a collection of cells where the cells may be differentiated into highly specialized ones.

Inspiration 2:
Follow a federated approach.

Biological cells never lost their property of autonomy even in highly complex multi-cell species. Biological systems are federated ones. The blood circulation can be interpreted as a trick to "simulate" the environment a cell needs to survive in such a federated multi-cell setting. Social insects may be seen as a copy of the same principle; now using more elaborate "macro cells". And this principle can be recursively extended. It may not be so surprising that the federation principle can be found using more and more complex "cells", a principle that reaches up to human societies. Federation seems to be a very useful principle to achieve robustness.

Inspiration 3:
Elastic division of labor.

Usually there is some dedication, some division of labor in federated systems. The degree of this division of labor increases by the complexity of the federal community. However, it can be observed that in most cases there is more or less elasticity. Components of a community dedicated to specific tasks can take over other tasks whenever they receive stimulations beyond their present threshold. This observation certainly is a valuable inspiration for future "Embedded Systems". Our own experiments in the areas of service migration, clustering, or real-time scheduling of media streams did show very robust and fault tolerant behavior when following this principle. Division of labor together with elasticity provides a good compromise between efficiency and avoidance of single points of failure.

Inspiration 4:
Make use of a broad variation of communication mechanisms.

Federated systems following the basic principle of delegation (distribute globally only what to do, let the individual components decide how to do) rely on an adequate communication scheme. It can be observed that nature created the entire range from uni-cast/multi-cast (nerve system) to multi-cast/broad-cast (secreting hormones/cytokines or pheromones) and from dedicated "cabling" (nerve system) via "*powerline communication*" (hormones/cytokines) to wireless (pheromone). Common to all these communication approaches is the fact that they are tailored for federated systems. All biological systems are made as a collection of cells and each single cell is equipped with sensors and actuators. All higher order constructions make use of this basic principle. By the same reason similarities can be observed between the different communication concepts. Nerve threads are made by sequences of nerve cells communicating via their synapses making use of the ability of any cell to cause and sense electrical potentials. Other capabilities of cells for sensing and acting are given by the ability to expose specific proteins on their surface and to sense the surface of proteins (necessary in any case as part of a cell's digestion system). This principle is used within the hormone system, in immune-networks via cytokines, and also when using pheromone for communication. An interesting aspect is the reuse of energy flows (cardiovascular system) to transmit messages. This is a kind of biological powerline communication. Stigmergy can be seen as transforming hormones or cytokines to a more general environment. An important principle in any case is a decay mechanism for messages, evaporation of pheromones in case of ant colony communication via stigmergy. Of course, the communication demands in technical systems differ. However, it is worth to consider biological communication techniques as inspiration as well.

Inspiration 5:
Delegation.

Common to all communication and control techniques in biological systems is the principle of delegation. It is up to a cell how to react on a sensed signal. This reaction may depend of the specific cell type or even cell instance (thus enabling multicast) or on actual environmental or state conditions of a cell. Large, complex systems need a certain degree of self-organization or, even less tight, self-coordination. Under such circumstances pre-planned communication systems seem to be no longer adequate. By the principle of delegation the amount of information to be communicated can be reduced dramatically.

More recently the discussion about *Cyber Physical Systems* (CPS) emerged. It is an attempt to combine two engineering cultures that up to now emerged in a relatively separated manner: highly predictable, hard real-time-driven Embedded Systems and the Internet, a communication infrastructure that is based on probabilistic principles. CPSs seem to be a promising approach to build globally distributed, collaborating technical artifacts. One of the fundamental arguments within this community is that the traditional separation into functional and non functional properties of computation seems to be no longer adequate when building the deeply embedded but widely distributed systems of the future. Especially abstracting away time which in most areas of computing is a common principle turns out to be a dangerous assumption. The solutions proposed include the usage of a strict and very precise global time source and then abstracting this source to a "*sparse time*"-model.[19] Based on such a

[19] Hermann Kopetz, "Embedded System Complexity", in: *Self-Optimizing Mechatronic Systems: Design the Future: Technologies for Tomorrow's Mechanical Engineering Products –*

model adequate OO (*Object Oriented*) architectures can be built, e.g. using the TMO (*Time-Triggered Message-Triggered Object*) approach of UC Irvine.[20] This approach seems to be completely different from the techniques of handling time in biological systems. They tend to follow an approach to approximate and correct afterwards if the approximation turns out to be wrong or not precise enough. It definitely makes no sense to look for inspirations from biology in an ideological manner. Technology opens potentials that were not available within evolution up to now and these potentials have to be used. Establishing a precise global time base was made possible by GPS and comparable systems and as it is available, it should be used. Other aspects addressed in CPS research, however, match relatively well with inspirations we can get from biological systems. As already mentioned several times in this paper, all biological systems are built bottom-up using a strict cell-based approach. These cells are more comparable to components than to objects in the OO sense. Communication is done by signaling values; it then is up to the components how to react. This basic principle of delegation constitutes much of the success of biological systems and should be considered as a basic principle for CPSs as well. Biological systems do not distinguish between functional and non functional properties. Nature always is aware of resources, is making use of what is available (considers the available "*platform*"), provides solutions how to handle lacking resources to a certain amount. This is another principle to be considered as inspiration when building CPSs. If such systems are built in a bottom-up manner by creating cells based on and closely adapted to available platforms, being sensitive for certain sets of rules, and being highly adaptive, capable of learning, then many of the CPSs' challenges might be solvable. Building a generic framework, a *Cyber Biosphere* (CBS) may be an attempt worth to be worked on.

6 Conclusion

In this paper some arguments are presented for taking inspirations from biology when designing the complex technical artifacts of the future. Using some

Dependability and Software Engineering – Design Methods and Tools; 7th International Heinz Nixdorf Symposium Self-Optimizing Mechatronic Systems: February 20–21, 2008, Heinz Nixdorf MuseumsForum, Paderborn, Germany, Paderborn, 2008, p. 469–486, and Hermann Kopetz, "The Complexity Challenge in Embedded System Design", in: *11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC): May 5–7, 2008, Orlando, Florida; Proceedings*, Piscataway, NJ, 2008, p. 3–12.

[20] Kanghee H. Kim, "Object Structures for Real-Time Systems and Simulators", in: *Computer* 30, 8 (1997), p. 62–70, and Kanghee H. Kim/Yuqing Li/Kee-Wook Rim/Eltefaat Shokri, "A Hierarchical Resource Management Scheme Enabled by the TMO Programming Scheme", in: *11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC), 2008: May 5–7, 2008, Orlando, Florida; Proceedings*, Piscataway, NJ, 2008, p. 370–376.

examples it has been shown that such inspirations may be helpful especially when the systems have to behave in a robust manner in rapidly changing environments. However, one should never make the mistake just to copy nature into technical artifacts. Our artifacts have to work in a dependable manner for some years or decades. Nature "thinks" in terms of millions of years, short-term behavior is of minor interest. Nature optimizes the long-term global performance; the specific entity is of no interest. Engineers have to consider the single entity they are liable for. Thus, taking inspiration from nature should always be an option but never more than an option among others.

References

Brinkschulte, Uwe/von Renteln, Alexander/Weiss, Michael, "Analyzing the Behavior of an Artificial Hormone System for Task Allocation", in: Juan González Nieto (ed.), *Proceedings of the 6th International Conference on Autonomic and Trusted Computing, ATC 2009: July 7–9, 2009, Brisbane, Australia*, Berlin, Heidelberg, 2009, p. 47–61.

Cannon, Walter B., *Bodily Changes in Pain, Hunger, Fear and Rage: An Account of Recent Research into the Function of Emotional Excitement*, reprint of 2nd ed., New York, NY, 1929.

Castro, Leandro N. de/Timmis, Jonathan, *Artificial Immune Systems. A New Computational Intelligence Approach*, London (et al.), 2002.

Ditze, Michael , "Evaluation of an Ant Colony Optimization Based Scheduler for the Transmission of Multimedia Traffic in the 802.11e EDCA", in: *Proceedings of the 3rd ACM Workshop on Wireless Multimedia Networking and Performance Modeling: October 22nd 2007, Chania, Crete Island, Greece* [held in conjunction with the 10th ACM/IEEE International Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM)], New York, NY, 2007, p. 9–15.

—/Becker, Markus, "An Improved Adaptive ACO Meta Heuristic for Scheduling Multimedia Traffic Across the 802.11e EDCA", in: Reza Rejaie/Roger Zimmermann (ed.), *Multimedia Computing and Networking 2008: January 30–31, 2008, San Jose, California, USA; Proceedings, Electronic Imaging, Science and Technology*, Bellingham, WA, 2008, p. 6818–6820.

Dorigo, Marco/Manniezzo, Vittorio/Colorni, Alberto, "The Ant System: Optimization by a Colony of Cooperating Agents", in: *IEEE Transactions on Systems, Man, and Cybernetics, Part B* 26, 1 (1996), p. 29–41.

—/Stützle, Thomas G., *Ant Colony Optimization* (A Bradford Book), Cambridge, MA, (et al.), 2004.

—/Birattari, Mauro/Blum, Christian, et al. (eds.), *Ant Colony Optimization and Swarm Intelligence. 4th International Workshop, ANTS 2004: September 5–8, 2004, Brussels, Belgium; Proceedings* (Lecture Notes in Computer Science Vol. 3172), Berlin, 2004.

—/Gambardella, Luca Maria/Birattari, Mauro, et al. (eds.), *Ant Colony Optimization and Swarm Intelligence. 5th International Workshop, ANTS 2006: September 4–7,* *2006, Brussels, Belgium; Proceedings* (Lecture Notes in Computer Science Vol. 4150), Berlin, 2006.

Dressler, Falko, *Self-Organization in Sensor and Actor Networks* (Wiley Series in Communications Networking Distributed Systems), Chichester (et al.), 2007.

Giese, Holger/Montealegre, Norma/Müller, Thomas/Oberthür, Simon/Schulz, Bernd, "Acute Stress Response for Self-Optimizing Mechatronic Systems", in: Yi Pan/ Franz J. Rammig, et al. (eds.), *2006 – Biologically Inspired Cooperative Computing*, Vol. 216, p. 157–167.

Heimfarth, Tales/Janacik, Peter, "Experiments with Biologically-Inspired Methods for Service Assignment in Wireless Sensor Networks", in: Mike Hinchey/Anastasia Pagnoni, et al. (eds.), *Biologically-Inspired Collaborative Computing: IFIP 20th World Computer Congress, Second IFIP TC 10 International Conference on Biologically-Inspired Collaborative Computing: September 8–9, 2008, Milano, Italy*, Vol. 268, New York, NY, 2008, p. 71–84.

—/Janacik, Peter/Rammig, Franz J., "Self-Organizing Resource-Aware Clustering for Ad Hoc Networks", in: *Proceedings of the 5th IFIP WG 10.2 International Conference on Software Technologies for Embedded and Ubiquitous Systems*, Berlin, Heidelberg, 2007, p. 319–328.

Kim, Kanghee H./Li, Yuqing/Rim, Kee-Wook/Shokri, Eltefaat, "A Hierarchical Resource Management Scheme Enabled by the TMO Programming Scheme", in: *11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC), 2008: May 5–7, 2008, Orlando, Florida; Proceedings*, Piscataway, NJ, 2008, p. 370–376.

—, "Object Structures for Real-Time Systems and Simulators", in: *Computer* 30, 8 (1997), p. 62–70.

Kopetz, Hermann, "Embedded System Complexity", in: *Self-Optimizing Mechatronic Systems: Design the Future: Technologies for Tomorrow's Mechanical Engineering Products – Dependability and Software Engineering – Design Methods and Tools; 7th International Heinz Nixdorf Symposium Self-Optimizing Mechatronic Systems: February 20–21, 2008, Heinz Nixdorf MuseumsForum, Paderborn, Germany*, Paderborn, 2008, p. 469–486.

—, "The Complexity Challenge in Embedded System Design", in: *11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC): May 5–7, 2008, Orlando, Florida; Proceedings*, Piscataway, NJ, 2008, p. 3–12.

Montealegre, Norma/Rammig, Franz J., "Immuno-Repairing of FPGA Designs", in: Mike Hinchey/Anastasia Pagnoni/Franz J. Rammig/Hartmut Schmeck (eds.), *Biologically-Inspired Collaborative Computing: IFIP 20th World Computer Congress [WCC 2008], Second IFIP TC 10 International Conference on Biologically-Inspired Collaborative Computing: September 8–9, 2008, Milano, Italy*, New York, NY, 2008, p. 137–149.

Oberthür, Simon/Böke, Carsten, "Flexible Resource Management – a Framework for Self-Optimizing Real-Time Systems", in: Bernd Kleinjohann/Guang R. Gao/Hermann Kopetz/Lisa Kleinjohanna/Achim Rettberg (eds.), *Design Methods and Applications for Distributed Embedded Systems: IFIP 18th World Computer Congress, TC10 Working Conference on Distributed and Parallel Embedded Systems (DIPES 2004): August 22–27, 2004, Toulouse, France*, Vol. 268, Boston, MA, (et al.), 2004, p. 177–186.

Rammig, Franz J., "Cyber Biosphere for Future Embedded Systems", in: Uwe Brinkschulte/Tony Givargis/Stefano Russo (eds.), *Software Technologies for Embedded* *and Ubiquitous Systems* (Lecture Notes in Computer Science Vol. 5287), Berlin, Heidelberg, 2008, p. 245–255.

Renteln, Alexander von/Brinkschulte, Uwe/Weiss, Michael, “Examinating Task Distribution by an Artificial Hormone System Based Middleware”, in: *IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC), 2008: May 5-7, 2008, Orlando, Florida; Proceedings*, Piscataway, NJ, 2008, p. 119–123.

Tarakanov, Alexander O./Kvachev, Sergei V./Sukhorukov, Alexander V., “A Formal Immune Network and Its Implementation for On-line Intrusion Detection”, in: Vladimir Gorodetsky/Igor Kotenko/Victor Skormin (eds.), *Computer Network Security* (Lecture Notes in Computer Science Vol. 3685), Berlin, Heidelberg, 2005, p. 394–405.

Timmis, Jon/Bentley, Peter/Hart, Emma (eds.), *Artificial Immune Systems. Second International Conference (ICARIS): September 1–3, 2003, Edinburgh, UK; Proceedings* (Lecture Notes in Computer Science Vol. 2787), Berlin (et al.), 2003.