iRIS - Towards a Robotic Immune System

Dietmar Schreiner [1] **Jens Knoop** [1]
[1]Institute of Computer Languages, Compilers and Languages Group
Vienna University of Technology, Vienna, Austria
{schreiner,knoop}@complang.tuwien.ac.at

1 Introduction

Progress in robotics has led to complex autonomous robotic systems, fulfilling mission critical tasks in safety critical environments. Besides classical robotic tasks in automation and tele-operation, todays applications of robotic systems range from search and rescue, over explorative deep see or space exploration, up to autonomous navigating of vehicles in public areas. Unfortunately, increase in capabilities and thus complexity has also led to a dramatic increase in possible faults that might manifest in errors. Even worse, by applying robots with emerging behavior in non-deterministic real-world environments, faults may be introduced from external sources. Consequently, fault testing has become increasingly difficult. Both, software and hardware may fail or even break, and hence may cause a mission failure, heavy damage, or even severe injuries and loss of lifes. The ability of a robotic system to function in presence of such faults, so to become fault tolerant, is a continuously growing area of research.

Our work meets this challenge by developing a mechanism for robotic systems that is capable of detecting defects, selecting feasible counter measures, and hence keeping robots in a sane and and consequently safe state. Inspired by biology, we conceptually build an immune system [1] for a robot, which is able to detect anomalies, and which is able to autonomously counter them by appropriate means.

2 Methodology

In terms of computer science, a biological immune system is a robust, multi-layered, distributed system that is able to identify numerous pathogens. Its main responsibility is to counter act harmful effects to keep the organism in a sane state. An artificial immune system is meant to do exactly the same for computers and in our case robotic systems. Harmful effects can be detected and can be overcome autonomously.

In [2] one bio-inspired algorithm, the negative selection algorithm, is described, which is based on the detection of self from non-self. Dedicated immune cell types like lymphocytes have receptors that allow them to bind to specific proteins. During maturation these cells are ‘trained’ on proteins that are naturally present in the organism, the self-antigenes. Lymphocytes which spuriously bind to self-antigenes are destroyed immediately (apoptosis). After reaching maturity, the trained cells are spread over the organism. If they bind to a protein now, this is a clear indication of a non-self protein, a pathogen.

iRIS (innate Robotic Immune System) is structured in accordance to a biological immune system: Within the organism (the robot), distributed autonomous light-weight processes,

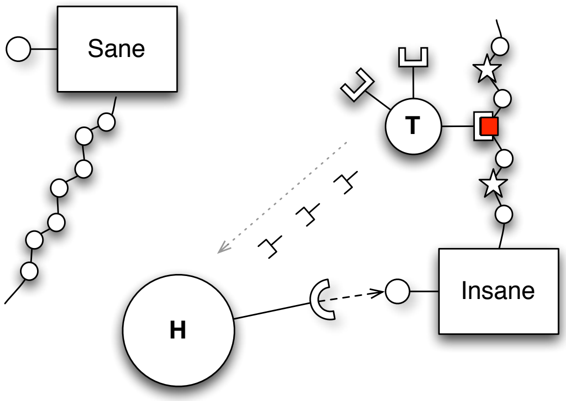

Figure 1: Immune reaction to faulty component

so called T-processes, constantly monitor the system's sanity via their receptors, e.g., software-sensors, execution monitors, or even hardware sensors. Antigenes are sequential representations of specific states within the robot, and may be defined and classified at development but also dynamically at run-time. Detection of non-self antigenes is achieved in two ways: (i) T-processes undergo a process of maturation, which means they are trained to detect self-antigenes on a running system. (ii) T-processes utilize 'genetic knowledge', represented by predefined rules and parameters. On detection of a non-self antigene, T-processes activate H-processes, which are in charge of eliminating anomalies (e.g., by restarting malfunctioning components of a robot, reinitializing affected structures, or recalibrating sensors).

Figure 1 depicts the conceptual idea behind an iRIS immune reaction: It shows two components, *Sane* and *Insane*, both exposing their associated antigene. As *Insane* is malfunctioning, the T-process T detects this non-self behavior and notifies specialized H-processes. One H-process finally docs at *Insane's* maintenance interface to counter-act the fault. In addition, our observations show that typical faults within robotic systems require complex repair actions at distributed subsystems. For that reason, H-processes may emit messenger antigenes to trigger additional repair activities at all affected subsystems.

Compared to existing work, iRIS covers aspects of autonomic computing like summarized in [3]. It does reflective computation [4, 5] at run-time in a bio-inspired way, using knowledge extracted from static analyses and system models at development and compile time. A model driven development methodology as much as static analyses come to use to extract the 'genetic knowledge' of iRIS. In addition, iRIS incorporates dynamic means of machine learning, also inspired by natural immune systems, to cope with unforeseen faults.

References

[1] J. D. Farmer, N. H. Packard, and A. S. Perelson, "The immune system, adaptation, and machine learning," *Physica D: Nonlinear Phenomena*, vol. 22, no. 1-3, pp. 187 – 204, 1986. Proceedings of the Fifth Annual International Conference.

[2] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri, "Self-nonself discrimination

in a computer," in *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy*, 1994.

[3] S. Dobson, R. Sterritt, P. Nixon, and M. Hinchey, "Fulfilling the vision of autonomic computing," *Computer*, vol. 43, no. 1, pp. 35 –41, 2010.

[4] P. Maes, "Concepts and experiments in computational reflection," in *OOPSLA*, pp. 147–155, 1987.

[5] M. Rodríguez, J.-C. Fabre, and J. Arlat, "Wrapping real-time systems from temporal logic specifications," in *EDCC* (F. G. 0002 and P. Thévenod-Fosse, eds.), vol. 2485 of *Lecture Notes in Computer Science*, pp. 253–270, Springer, 2002.