Formal Analysis of Fault Tolerance in Self-Organizing Systems

Jan Calta
first supervisor: Prof. Dr. Holger Schlingloff
second supervisor: Prof. Dr. Miroslaw Malek
in METRIK since: 2008-03-15

Problem statement

Self-organization is used as one possible approach to assure fault tolerance in a distributed system. Fault tolerance is the ability of a software system to recover from an undesirable state caused by a fault. If a fault affects the system and the system is tolerant to this kind of fault, it adapts to this situation by recovering to the intended functionality. Tolerance to a specific class of faults is assured in a system to improve its robustness and reliability.

A self-organizing system consists of uniform members with limited capabilities. However, if the members establish some kind of interaction (via physical connection, communication, etc.) and a specific structure is formed, an advanced functionality emerges in the system. Moreover, this added quality is achieved automatically and locally without any intervention from outside the system. An example of a system where self-organization is beneficial is a sensor network for detecting natural disasters or a multi-agent robot system for performing rescue operations.

Formal analysis of the key properties of software systems for disaster management is often required. In case of fault tolerance, the information about the set of faults to which the system is tolerant is particularly interesting. Another important characteristic is the partial functionality which the system provides during the recovery from a specific fault. For such an analysis fault tolerance in a specific system and the system itself must be described formally. A model of the system must be constructed from this description. The information about fault tolerance must be extracted from the model.

Approach

I defined a general schema of adaptation to faults - recoverability and correctness properties - in modal logic. Alternating-Time Temporal Logic (ATL) is a modal logic designed for reasoning about open systems, i.e., about systems interacting with their environment. With ATL it is possible to specify to a certain extent a behavior of the system environment and thus also describe potential faults caused by the environment.

There have been several attempts to extend ATL for the purpose of reasoning about multi-agent systems, however, so far none of them proved to be applicable to real-world problems. Instead of extending the logical language to cover the large domain of multi-agent systems, I take the approach of adjusting the semantics of ATL so that it is applicable to adaptive distributed systems.

I proposed a classification of adaptive systems. The introduced classes determine means of expressivity that are required for formal description of the respective systems. After the mapping of system classes to fragments of ATL is finished, I'll further focus on the subclass of self-organizing systems.

This approach allows to use ATL-based analysis in several stages of system development – from the initial design of algorithms to the final verification.

Future work

- Research on the expressivity of the ATL-based logic appropriate for adaptive systems corresponding to the proposed classification must be finished.
- A tool for the formal analysis of adaptive systems based on model checking for the introduced logic must be provided. It will be further developed to meet the specifics of self-organizing systems.
- Since self-organizing systems are designed to be scalable, verification results valid for any size of the analyzed system would be welcome. Research on formal techniques applicable to model checking of scalable systems must be done.

Publications

- Calta, J.: A Taxonomy of Adaptive Systems, to appear in EUMAS 2009: 7th European Workshop on Multi-Agent Systems, December 2009
- Calta, J. and Malek M.: Formal Analysis of Fault Recovery in Self-Organizing Systems, to appear in DASC-09: 8th International Conference on Dependable, Autonomic and Secure Computing, December 2009
- Calta, J.: Representation of Temporal Logic Formulae in Zing, CS&P 2008: Concurrency, Specification and Programming, September 2008