Designing Self-Healing Application System

Goutam Kumar Saha
Email: gksaha@ieee.org, sahagk@gmail.com

Software technique based on validating application semantics assertions can be a useful toward designing a self-healing application system. Self-Healing is a very new area of research that deals with fault tolerance for dynamic systems. Self-healing deals with imprecise specification, uncontrolled environment, and reconfiguration of systems according to their dynamics. The term "Self-healing" denotes the capability of a software system in dealing with bugs. Fault Tolerance for dependable computing is about providing the specified service through rigorous design whereas self-healing is about run-time issues. Software which is capable of detecting and reacting to its malfunctions, is called self-healing software. Such software system has the ability to examine its failures and to take appropriate corrections. Self-Healing system must have knowledge about its expected behavior in order to examine whether its actual behavior deviates from its expected behavior in relation of the environment.

Self-healing categories of aspects include Fault-model or fault hypothesis, System-response, System-completeness and Design-context. A fault-model of Self-Healing system is to state what faults or injuries to be self-healed including fault duration, fault source such as, operational errors, defective system requirements or implementation errors etc. System-response includes the aspects of fault detection, degree of degradation, fault response and an attempt to recovery action or compensation for a fault. Fault detection approaches involved in a self-healing system include application system's semantics-driven assertions, supervisory checks, examining the computing answers, comparison of replicated components, online self testing etc. System-completeness aspect deals with reality of knowledge limits, incompleteness in specifications and designs thereof. It also deals with the problem of system self-knowledge, system evolution etc. Handling the architectural incompleteness for example, of third-party components or of various patches during or after system deployment is really a challenging issue in developing a self-healing system.

Designers of Self-Healing application system should have a thorough knowledge about their application systems' semantics. Design-context addresses the problems on abstraction level, component-level homogeneity, system linearity, system-scope, pre-deterministic behaviors, user involvement aspects etc. Application semantics driven assertions approach helps to run-time validate the various assertions that are derived from application semantics for designing a Self-healing system. The assertions are derived from the application domain specific knowledge of the system's expected behavior and Self-Healing achieved by examining whether the system's actual behavior deviates from its expected behavior in relation of the system's environment.

Further Reading:

- P. Koopman, "Elements of the Self-Healing System Problem Space," Proceedings of the ICSE WAD03, 2003.
- Goutam Kumar Saha, "Application Semantic Driven Assertions toward Fault Tolerant Computing," ACM Ubiquity, Vol. 7, No. 22, pp. 1 - 27, ACM Press, 2006, USA.

- Goutam Kumar Saha, “Software Fault Tolerance through Run – Time Fault Detection,” ACM Ubiquity, Vol. 6, No. 46, pp. 1-5, ACM Press, 2005, USA.
- Goutam Kumar Saha, “Software Based Fault Tolerance – a Survey,” ACM Ubiquity, Vol. 7, No. 25, pp. 1-17, ACM Press, 2006, USA
- Goutam Kumar Saha, “Transient Fault Tolerance through Algorithms,” IEEE Potentials, Vol. 25, No. 5, pp. 25-30, IEEE Press, Sep-Oct 2006, USA.