The Case for Handling Inconsistency caused by Errors

Wilhelm Hasselbring

Software Engineering Group, University of Oldenburg
26111 Oldenburg, Germany
http://se.informatik.uni-oldenburg.de
hasselbring@informatik.uni-oldenburg.de

Abstract. Errors are a fact of life. In requirements engineering, for instance, it is well accepted that we have to live with inconsistent specifications; we need measures to handle inconsistency caused by errors.
Database textbooks generally explain that integrity constraints should be satisfied at all times because they capture the set of all legal databases. Nevertheless, data inconsistency is a phenomenon that often occurs in practice. The most common reason for inconsistency is the need to integrate heterogeneous, independent data sources: different databases that are consistent by themselves can contain inconsistent information about the same real-world object.
The conflicts are revealed only when the data is brought together in an integrated database. In such situations, it is of practical importance to know how to deal with violations of integrity constraints. In general, there is no single best way to restore consistency, leaving us with a multitude of possible repairs.
Domain-specific approaches are required: in hospital information systems, for instance, data is often added, but only seldom changed. Delays in inserting data may cause incomplete, but not contradictory data.
Fault-tolerance and self-healing/self-stabilizing systems address the problem of handling errors, i.e. inconsistent states. EAI patterns emphasize asynchronous updates. In Software Engineering, software architectures with redundancy for enabling fault tolerance are designed. Programming languages provide mechanisms for exception handling. In this presentation, I discuss various issues of handling errors, and some related topics that are investigated in our graduate school TrustSoft (http://trustsoft.uni-oldenburg.de) [1].

Keywords. Inconsistency, Error Handling, Trustworthy Software Systems

References

1. Hasselbring, W., Reussner, R.: Toward trustworthy software systems. IEEE Computer **39** (2006) 91–92