The Problem Frames Approach to Software Engineering

Michael Jackson
The Open University, UK

Abstract

Software-intensive systems are those in which the computer executing the software is only one of the parts of the system. Problem frames offer a conceptual structure for the development of such systems: that is, a coherent way of analysing the problem to be solved, identifying the concerns and difficulties that it poses, and working towards a solution. This tutorial will present the basic ideas of problem frames, illustrating them in the context of a small software-intensive system.

The basic ideas of the approach are:
1. to attend both to the hardware/software machine, which is to be developed, and to the other system parts, which constitute the problem world;
2. to distinguish the given properties of the problem world from the requirements, which are the properties that the machine must establish and maintain in the world;
3. to pay careful attention to the phenomena of the problem world;
4. to structure the development problem as a set of subproblems, and to consider each subproblem in isolation before considering the composition of the subproblems and their solutions;
5. so far as possible, to recognise each subproblem as a member of a recognised class for which a solution method is known and whose most important concerns have been identified.

Problem frames are not a notation or a calculus or a formalism; nor are they a development method or a process. They can fit with, into, or around specific techniques (such as agile or RUP) and specific notations (such as UML, Petri Nets, or DFDs). Problem frames do not promise a prescription for every problem; nor do they promise a complete prescription for any problem. They remind you of things you know already, but may not always pay enough attention to.

Biography

Michael Jackson wrote his first computer program at school in 1951, but did not start working in software until 1961. He developed the JSP program design method, chosen as the standard for UK Government software development; he led the development of JSD, a method of system specification and design based on entity histories represented as sequential processes. Since 1990 he has worked as an independent consultant and researcher in software development method, holding visiting posts at several universities and participating in research projects. His recent work has focused on the analysis and structure of software development problems, based on the idea of problem frames. From 1989 to 2001 he worked, in cooperation with Dr Pamela Zave, as a part-time researcher at AT&T Bell Laboratories (now AT&T Research Laboratories) in New Jersey USA. This work focused on the feature interaction problem and on principles and techniques for specifying telecommunications systems and services. The chief result of the work is the patented DFC architecture for specifying telecommunications services. He has described his work in many papers and in four books: Principles of Program Design (1974); System Development (1983); Software Requirements & Specifications (1995); and Problem Frames (2001).