Abstract

Many of the interesting problems in Computer Science related to Software Engineering have yielded only to partial solutions or not at all. In the related areas of research denoted Communications Software Engineering and Protocols, most corresponding problems are on their way to a solution. In this presentation, we will explore some similarities and differences between these research areas by giving examples of problems which appear to be tractable in one area and not the other, and conclude with some ideas for building conceptual bridges.

1. Introduction

One of the central problems in both theoretical and applied computer science is the problem we will denote the conformance problem, namely (in simplest terms):

Given a specification $S$ of desirable process behaviours, and an implementation $I$ derived from $S$, verify that $I$ exhibits only behaviours which exactly correspond to behaviours included in $S$.

A difficulty with such a definition is of course the ambiguity of many of the key terms such as "derived from", "verify", "exhibits", and "exactly corresponds". This ambiguity is long-standing in mainstream computer science despite valiant attempts by both theoretical computer scientists and software engineers to develop a "pragmatically rigorous" foundation for
software engineering. Such a foundation would provide cost-effective tools and engineering techniques based on realistic, yet mathematically precise models.

In the area of communications protocols, however, substantive progress has been made towards what might be termed "foundations of protocol engineering". An abstraction of the protocol engineering process is given in Figure 1.

![Figure 1. Current Protocol Engineering Process](image)

A major contribution towards this progress has been the development of a number of standardized specification languages intended for communication protocols and services, called **formal description techniques** or FDTs. [Bo89] These FDTs vary from FSM-based (SDL and Estelle) to