ABSTRACT

This dissertation presents the *Protocol Complementation*, which is a new approach to protocol interworking problems. In the first part, it describes a systematic way to design a protocol converter from its service specifications and in the latter part, an attempt is made to implement the designed protocol converter using a Formal Description Technique (FDT).

The protocol complementation is realized by introducing additional layers, known as virtual layer, extra protocol functions, etc. A *Complementation* algorithm is developed and presented for the communicating finite state machine (CFSM) model of the protocols P and Q. The converter R_{PQ}, employing a new virtual layer, for protocols $P =< P_s, P_r >$ and $Q =< Q_s, Q_r >$ is viewed as a black box such that R_{PQ} is in between sender P_s and receiver Q_r. So the resulting protocol $< P_s, R_{PQ}, Q_r >$ provides an error free communication between P_s and Q_r.

The algorithm *Complementation* uses several other algorithms as follows. First, an algorithm, named as *Complement*, is employed to construct the complemented machines which makes use of another algorithm called *Project*. For this purpose, a state space is formed by the union of the state spaces of P_r and Q_s. From the set of allowed sequence of message exchanges, the traces are constructed in a FIFO order and are projected onto each other to generate the complemented CFSMs. Finally, the algorithm *Combine* constructs the composite state machine from the complemented machines. The converter is declared valid only when it succeeds the test for unspecified receptions, deadlocks and livelocks. Several practical examples are considered to demonstrate the protocol complementation results.

Extended State Transition Language (Estelle) is used to specify the protocol. The converter is implemented using a semiautomatic Estelle-C compiler and verified with the above examples. The semiautomatic implementation offers several advantages over the conventional manual implementation, including correctness and modularity in protocol implementation code, conformance to the specification and reduction in implementation time. The error free communication between the protocols P_s and Q_r proves the applicability of the *Complementation* algorithm as well as the correctness of implementation.