Part III

Representing Specific Subsystems

Many successful modelling studies are conducted without venturing beyond the techniques described in Part II. In other words, the system characteristics considered in these studies are restricted to those that can be represented directly using the parameters of separable queueing networks.

There are, of course, situations in which the analyst will wish to represent a specific subsystem in greater detail than is possible within the confines of separable networks. Techniques for doing so are the subject of Part III.

The efficient evaluation that is characteristic of separable networks is mandatory in analyzing contemporary computer systems. For this reason, *non-separable* networks typically are evaluated by "mapping" them onto (perhaps several) separable networks. Since this mapping necessarily is approximate, the techniques for doing so traditionally have been referred to as *approximate solution techniques*. This phrase is not really meaningful, though, since we use approximate techniques to evaluate even separable networks, and since any queueing network model is only an approximate representation of an actual system.

As yet there is no unifying theory underlying these techniques. There is, however, a small set of ideas on which they are based. Among these ideas are:

- *iteration* — making an initial guess at the value of a parameter, then iteratively refining this value, in a manner analogous to that of the MVA-based iterative approximate solution techniques for separable networks, described in Chapter 6;
- *load concealment* — representing the effect of a workload component or system characteristic indirectly, by "inflating" the service demands of those workload components that are represented explicitly, in a manner analogous to the calculation of performance measures for closed classes in mixed separable networks, described in Chapter 7;

- *decomposition* — evaluating a subsystem in isolation, perhaps using a heuristic, and incorporating the results of this analysis in a flow equivalent service center that can be included in a high-level model, as described in Chapter 8.

Not all of the techniques are fully general. We will see that *homogeneity assumptions* frequently are introduced in some aspect of a model to facilitate the detailed representation of a subsystem. As a specific example, in order to evaluate multiple class memory constrained queueing networks we will assume that the throughput of each class is dependent only on its own central subsystem population and the *average* central subsystem population of every other class.

We have organized our discussion into three chapters, which consider the representation of memory, disk I/O, and processors. Just as with the algorithms for evaluating separable queueing networks presented in Part II, the techniques presented in Part III generally will be incorporated in a queueing network analysis package at a level not visible to the analyst. While it is possible to use these techniques without understanding them, achieving such an understanding is important for two reasons: so that they can be used confidently and appropriately, and so that the analyst can devise related techniques when confronted with novel situations. Some examples of such novel applications will be given in Part V.