CONCLUSION

While the performance of computers and the speed of data networks continues to improve steadily, our ability to utilize these resources effectively does not. Data communications software has become a bottleneck in many high performance systems; it is often much slower than the hardware permits and it can be hard to establish its logical consistency.

It is notoriously difficult to write software for a distributed system. It is even harder to prove rigorously the correctness of such software. In its simplest form, the problem is to design methods that allow asynchronously executing communicating machines to exchange information quickly and reliably and to prove that these methods, or protocols, have certain desirable properties.

Today, most protocols are designed in an *ad hoc* matter. There is a known set of protocol standards, whose description is faithfully copied in most textbooks. There is, however, little understanding of why some protocols work and what is wrong with others. A designer needs to know how a correct protocol can be constructed from scratch and how that design can be matched to specific design and correctness criteria. The techniques that can be used to prove that a new protocol design is correct have long been considered too esoteric for real day-to-day use. This book is meant to show that the tools have come of age.

The design methods and tools that we have discussed allow the designer to attack fundamental process coordination problems in a rigorous and a practical manner. To design reliable protocols, no matter what your application is, you need tools to test your ideas. This book should convince you that the right tools are available. The capabilities of the new validation tools is sometimes justifiably regarded with scepticism. A generous number of pages is therefore devoted in this text to a detailed discussion of tools. For the first time, the complete source to these tools is now made available, both in this text and in electronic form. Your critical evaluation, experiments, applications, and comparisons are eagerly invited.