In the final three chapters we will try to understand types of concurrent systems for which CSP is not generally thought appropriate, and at the same time see how it is often possible to translate automatically from these other notations into CSP.

We will spend most time studying the model of concurrency that the invention of CSP was a reaction against, namely shared-variable concurrency. In this, processes communicate by writing to and reading from shared variables rather than via explicit communication. Chapter 18 shows how we can both understand this type of concurrency and develop an automated verification engine called SVA for it using CSP and FDR.

Chapter 19 then examines shared-variable concurrency more deeply, for example by looking at how to define refinement over shared-variable programs and examining how to prove results about certain sorts of unbounded systems using SVA. Much of this is done through examples.

We have already studied timed concurrency in Chaps. 14 and 15. Chapter 20 looks at two other ways of extending the expressive power of CSP, namely to prioritised systems, where we can express a preference for which actions happen (e.g. \( b \) only if \( a \) is not available), and to mobile systems, in which processes and channel names can migrate.

Not only are shared variables, priority and mobility interesting in their own right, the fact that we can express and reason about these concepts in CSP and FDR may inspire others to explore further types of concurrent systems. For example, we can handle systems with a mixture of these ideas, as in Statecharts, (Sect. 20.2.1) which use time, priority, ordinary communication and shared variables.

Further work of interest includes incorporation of CSP and FDR into integrated tool sets. For example see Anderson et al. (An environment for integrating formal methods tools, 1997), the work of Verum (www.verum.com) (Hopcroft and Broadfoot, Automated Software Engineering, 2004) on integrating CSP/FDR with the Box Structure Development Method, and the integration of CSP and the B-method (Schneider and Treharne, Integrated Formal Methods, Springer, Berlin, 2004).