

---

# Concurrency, distribution, client-server and the Internet

*L*ike humans, computers can team up with their peers to achieve results that none of them could obtain alone; unlike humans, they can do many things at once (or with the appearance of simultaneity), and do all of them well. So far, however, the discussion has implicitly assumed that the computation is *sequential* — proceeds along a single thread of control. We should now see what happens when this assumption no longer holds, as we move to *concurrent* (also known as *parallel*) computation.

Concurrency is not a new subject, but for a long time interest in it remained mostly confined to four application areas: operating systems, networking, implementation of database management systems, and high-speed scientific software. Although strategic and prestigious, these tasks involve only a small subset of the software development community.

Things have changed. Concurrency is quickly becoming a required component of just about every type of application, including some which had traditionally been thought of as fundamentally sequential in nature. Beyond mere concurrency, our systems, whether or not *client-server*, must increasingly become *distributed* over networks, including the network of networks — the *Internet*. This evolution gives particular urgency to the central question of this chapter: can we apply object-oriented ideas in a concurrent and distributed context?

Not only is this possible: object technology can help us develop concurrent and distributed applications simply and elegantly.

## 30.1 A SNEAK PREVIEW

As usual, this discussion will not throw a pre-cooked answer at you, but instead will carefully build a solution from a detailed analysis of the problem and an exploration of possible avenues, including a few dead ends. Although necessary to make you understand the techniques in depth, this thoroughness might lead you to believe that they are complex; that would be inexcusable, since the concurrency mechanism on which we will finally settle is in fact characterized by almost incredible simplicity. To avoid this risk, we will begin by examining a summary of the mechanism, without any of the rationale.

*Warning:*  
**SPOILER!** (The next section is 30.2, page

If you hate “spoilers”, preferring to start with the full statement of the issues and to let the drama proceed to its dénouement step by step and inference by inference, ignore the one-page summary that follows and skip directly to the next section.