

# Foreword

*Stephen J. Mellor*

**THE CHALLENGES OF DEVELOPING HIGH-PERFORMANCE, HIGH-RELIABILITY,** and high-quality software systems are too much for ad hoc and informal engineering techniques that might have worked in the past on less demanding systems. The complexity of our systems has risen to the point where we can no longer cope without developing and maintaining a single overarching architecture that ties the system into a coherent whole and avoids piecemeal implementation, which causes testing and integration failures.

But building an architecture is a complex task. Examples are hard to come by, due to either proprietary concerns or the opposite, a need to “sell” a particular architectural style into a wide range of environments, some of which are inappropriate. And architectures are big, which makes them difficult to capture and describe without overwhelming the reader.

Yet beautiful architectures exhibit a few universal principles, some of which I outline here:

## *One fact in one place*

Duplication leads to error, so it should be avoided. Each fact must be a single, nondecomposable unit, and each fact must be independent of all other facts. When change occurs, as it inevitably does, only one place need be modified. This principle is well known to database designers, and it has been formalized under the name of *normalization*. The principle also applies less formally to behavior, under the name *factoring*, such that common functionality is factored out into separate modules.