

As these simple examples suggest, good architects don't construct their designs by ad-hoc, brute force means, and they certainly try to avoid clumsy hacks. In response to the complex demands of site and context, climate, the activities to be accommodated, supply chains of materials and components, construction processes, and budgets, they design commensurately varied and complex buildings. But they try to do this with conceptual elegance—following principles of economy of means, and rigorously applying their own version of Occam's Razor. Underlying the apparent variety and complexity of beautiful works of architecture, then, you can usually discover some simple, elegant principles of functional organization and formal order. Discovering these principles takes intellectual engagement—which is a crucial part of the experience and pleasure of architecture.

If you *can* figure out these principles, you can construct models of these works with a few equally elegant lines of code in some standard programming language, or (in a less traditional programming environment) a few shape rewriting rules. You might even be able to generalize, and write code that produces designs, following the same principles, that respond appropriately to ranges of varied conditions and requirements. But if you can't figure out these principles, then you are condemned to write lengthier, less insightful code.

The evident complexity of a building derives from the complexity of the requirements to which the architect has responded, and it is measured by the length of its brute force, point-by-point description. The principles that the architect followed to produce that response can usually be captured in a much shorter sequence of code that executes to generate all that complexity. To a pretty good approximation, the lower the ratio of this short description to the long description, the more beautiful the building.

Architects, then, admire the beauty of buildings that employ a few simple, elegant principles to accomplish many complex things. Similarly, software architects and programmers admire the beauty of code that cleanly and concisely (without sacrificing readability or maintainability) performs many complex tasks. And scientists admire the beauty and explanatory power of simple laws that describe widely varied phenomena. These are all particular cases of a beautiful general principle.