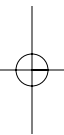


IV

Strategic Design



As systems grow too complex to know completely at the level of individual objects, we need techniques for manipulating and comprehending large models. This part of the book presents principles that enable the modeling process to scale up to very complicated domains. Most such decisions must be made at team level or even negotiated between teams. These are the decisions where design and politics often intersect.

The goal of the most ambitious enterprise system is a tightly integrated system spanning the entire business. Yet the entire business model for almost any such organization is too large and complex to manage or even understand as a single unit. The system must be broken into smaller parts, in both concept and implementation. The challenge is to accomplish this modularity *without losing the benefits of integration*, allowing different parts of the system to interoperate to support the coordination of various business operations. A monolithic, all-encompassing domain model will be unwieldy and loaded with subtle duplications and contradictions. A set of small, distinct subsystems glued together with ad hoc interfaces will lack the power to solve enterprise-wide problems and allows consistency problems to arise at every integration point. The pitfalls of both extremes can be avoided with a systematic, evolving design strategy.

Even at this scale, domain-driven design does not produce models unconnected to the implementation. Every decision must have a direct impact on system development, or else it is irrelevant. Strategic design principles must guide design decisions to reduce the interdependence of parts and improve clarity without losing critical interoperability and synergy. They must focus the model to capture the conceptual core of the system, the “vision” of the system. *And they must do all this without bogging the project down.* To help accomplish these goals, Part IV explores three broad themes: context, distillation, and large-scale structure.

Context, the least obvious of the principles, is actually the most fundamental. A successful model, large or small, has to be logically consistent throughout, without contradictory or overlapping definitions. Enterprise systems sometimes integrate subsystems with varying origins or have applications so distinct that very little in the domain is viewed in the same light. It may be asking too much to unify the models implicit in these disparate parts. By explicitly defin-

ing a **BOUNDED CONTEXT** within which a model applies and then, when necessary, defining its relationship with other contexts, the modeler can avoid bastardizing the model.

Distillation reduces the clutter and focuses attention appropriately. Often a great deal of effort is spent on peripheral issues in the domain. The overall domain model needs to make prominent the most value-adding and special aspects of your system and be structured to give that part as much power as possible. While some supporting components are critical, they must be put into their proper perspective. This focus not only helps to direct efforts toward vital parts of the system, but it keeps the vision of the system from being lost. Strategic distillation can bring clarity to a large model. And with a clearer view, the design of the **CORE DOMAIN** can be made more useful.

Large-scale structure completes the picture. In a very complex model, you may not see the forest for the trees. Distillation helps, by focusing the attention on the core and presenting the other elements in their supporting roles, but the relationships can still be too confusing without an overarching theme, applying some system-wide design elements and patterns. I'll give an overview of a few approaches to large-scale structure and then go into depth on one such pattern, **RESPONSIBILITY LAYERS**, to explore the implications of using such a structure. The specific structures discussed are only examples; they are not a comprehensive catalog. New ones should be invented as needed, or these should be modified, through a process of **EVOLVING ORDER**. Some such structure can bring a uniformity to the design that accelerates development and improves integration.

These three principles, useful separately but particularly powerful taken together, help to produce good designs—even in a sprawling system that no one completely understands. Large-scale structure brings consistency to disparate parts to help those parts mesh. Structure and distillation make the complex relationships between parts comprehensible while keeping the big picture in view. **BOUNDED CONTEXTS** allow work to proceed in different parts without corrupting the model or unintentionally fragmenting it. Adding these concepts to the team's **UBIQUITOUS LANGUAGE** helps developers work out their own solutions.

