helps answer the defining questions for one of the concerns. Key structural decisions divide the product into components and define the relationships among those components (Bass, Clements, and Kazman 2003; Booch, Rumbaugh, and Jacobson 1999; IEEE 2000; Garlan and Perry 1995). For any given product, there are many structures that need to be designed. Each must be designed separately so that it is viewed as a separate concern. In the next few sections we discuss some structures that you can use to address the concerns on our list. For example, the Information Hiding Structures show how the system is organized into work assignments. They can also be used as a roadmap for change, showing for proposed changes which modules accommodate those changes. For each structure we describe the components and the relations among them that define the structure. Given the concerns on our list, we consider the following structures to be of primary importance.

The Information Hiding Structures

COMPONENTS AND RELATIONS: The primary components are Information Hiding Modules, where each module is a work assignment for a group of developers, and each module embodies a design decision. We say that a design decision is the secret of a module if the decision can be changed without affecting any other module (Hoffman and Weiss 2000, chaps. 7 and 16). The most basic relation between the modules is "part of." Information Hiding Module A is part of Information Hiding Module B if A's secret is a part of B's secret. Note that it must be possible to change A's secret without changing any other part of B; otherwise, A is not a submodule according to our definition. For example, many architectures have virtual device modules, whose secret is how to communicate with certain physical devices. If virtual devices are organized into types, then each type might form a submodule of the virtual device module, where the secret of each virtual device type would be how to communicate with devices of that type.

Each module is a work assignment that includes a set of programs to be written. Depending on language, platform, and environment, a "program" could be a method, a procedure, a function, a subroutine, a script, a macro, or other sequence of instructions that can be made to execute on a computer. A second Information Hiding Module Structure is based on the relation "contained in" between programs and modules. A program P is contained in a module M if part of the work assignment M is to write P. Note that every program is contained in a module because every program must be part of some developer's work assignment.

Some of these programs are accessible on the module's interface, whereas others are internal. Modules may also be related through interfaces. A module's interface is a set of assumptions that programs outside of the module may make about the module and the set of assumptions that the module's programs make about programs and data structures of other modules. A is said to "depend on" B's interface if a change to B's interface might require a change in A.

The "part of" structure is a hierarchy. At the leaf nodes of the hierarchy are modules that contain no identified submodules. The "contained in" structure is also a hierarchy, since each