One of the most frequent sources of confusion in the UML is aggregation and composition. It's easy to explain glibly: Aggregation is the part-of relationship. It's like saying that a car has an engine and wheels as its parts. This sounds good, but the difficult thing is considering what the difference is between aggregation and association.

In the pre-UML days, people were usually rather vague on what was aggregation and what was association. Whether vague or not, they were always inconsistent with everyone else. As a result, many modelers think that aggregation is important, although for different reasons. So the UML included aggregation (Figure 5.3) but with hardly any semantics. As Jim Rumbaugh says, “Think of it as a modeling placebo” [Rumbaugh, UML Reference].
As well as aggregation, the UML has the more defined property of composition. In Figure 5.4, an instance of Point may be part of a polygon or may be the center of a circle, but it cannot be both. The general rule is that, although a class may be a component of many other classes, any instance must be a component of only one owner. The class diagram may show multiple classes of potential owners, but any instance has only a single object as its owner.

You'll note that I don't show the reverse multiplicities in Figure 5.4. In most cases, as here, it's 0..1. Its only other possible value is 1, for cases in which the component class is designed so that it can have only one other class as its owner.

The "no sharing" rule is the key to composition. Another assumption is that if you delete the polygon, it should automatically ensure that any owned Points also are deleted.

Composition is a good way of showing properties that own by value, properties to value objects (page 73), or properties that have a strong and somewhat exclusive ownership of particular other components. Aggregation is strictly meaningless; as a result, I recommend that you ignore it in your own diagrams. If you see it in other people's diagrams, you'll need to dig deeper to find out what they mean by it. Different authors and teams use it for very different purposes.

**Derived Properties**

Derived properties can be calculated based on other values. When we think about a date range (Figure 5.5), we can think of three properties: the start date,