to “compute the sugar content,” “estimate the time to make” the pudding, and obtain “instructions to make it.” A “bad approach” would be to:

- “List all puddings (Trifle, lemon upside-down pudding, Dutch apple cake, Christmas pudding)
- For each pudding, write down sugar content, time to make, instructions, etc.”

Although the presentation does not state why the approach is bad, we can easily surmise the reasons: as a collection of ad hoc descriptions, it has no reusability, since it does not take advantage of the property that different kinds of pudding may share the same basic parts; it has no extendibility, since any modification of a pudding part will require reworking all the puddings that rely on that part.

The pudding is a metaphor for the examples of real interest, contracts, but since it is easily understandable without a specialized knowledge domain, we continue with it. A “good approach” is to:

- “Define a small set of ‘pudding combinators.’
- Define all puddings in terms of these combinators.
- Calculate sugar content from these combinators too.”

A combinator is an operator that produces a composite object from similar objects. The tree shown in Figure 13-1, from the presentation, illustrates what the combinators may be in this example.

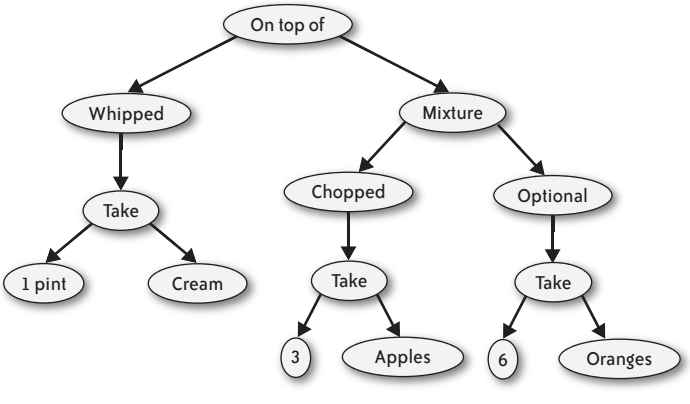

FIGURE 13-1. Ingredients and combinators describing a pudding recipe