- Also relevant in practice are issues of expressiveness and notation. They are taken into account to the extent that they affect the key criteria of architecture and design. For the most part, however, the discussion considers semantics rather than syntax.

Two more preliminary notes. First, terminology: by default, the term "contract" refers to financial contracts, relevant to the application domain of the article and presentation, and not to be confused with the software notion of Design by Contract* (the idea [Meyer 1997] of including elements of specification such as preconditions, postconditions, or invariants). In case of possible ambiguity, the terms used here will be *financial contracts* and *software contracts*.

Second, a semi-apology: when the discussion moves to OO territory in its second half, it includes more references to and repetitions from the author's previous publications than discretion would command. The reason is that the wide spread of object technology has been accompanied by the loss of some of its more subtle but (in our opinion) critical principles, such as command-query separation (see "State Intervention" later in this chapter); this makes some brief reminders necessary. For the full rationale behind these ideas, see the cited references.

## The Functional Examples

The overall goal of the article and presentation is to propose a convenient mechanism for describing and handling financial contracts, especially modern financial instruments that can be very complicated, as in this example from the presentation (in whose numerical values one can hear the nostalgic echo of a time when major currencies enjoyed a different relationship):

> "Against the promise to pay USD 2.00 on December 27 (the price of the option), the holder has the right, on December 4, to choose between:
>
> - Receiving USD 1.95 on December 29, or
> - Having the right, on December 11, to choose between:
>   - Receiving EUR 2.20 on December 28, or
>   - Having the right, on December 18, to choose between:
>     - Receiving GBP 1.20 on December 30, or
>     - Paying immediately one more EUR and receiving EUR 3.20 on December 29"

(Throughout this section, extracts in quotes are direct citations from the presentation or the article. Elements not in quotes are our interpretations and comments.)

As a pedagogical device to illustrate the issues, the presentation starts with a toy example: *puddings* rather than contracts. From the precise description of a pudding, it should be possible

* Design by Contract is a trademark of Eiffel Software.