

Chapter 15

UML Interaction Diagrams

Larman, C. “Applying UML and
Patterns”. 3rd Ed.
Ed. Prentice-Hall: 2005.

Fig. 15.1

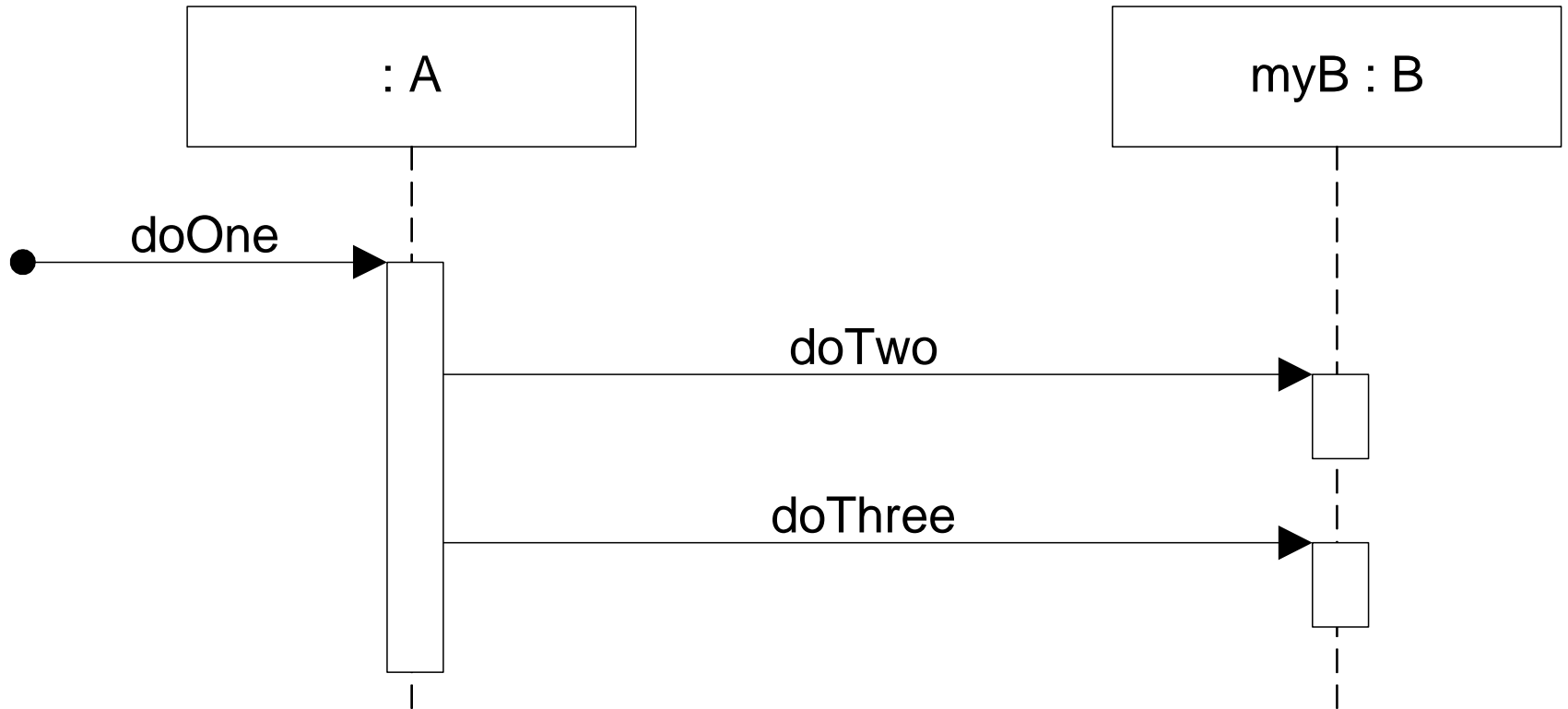


Fig. 15.2

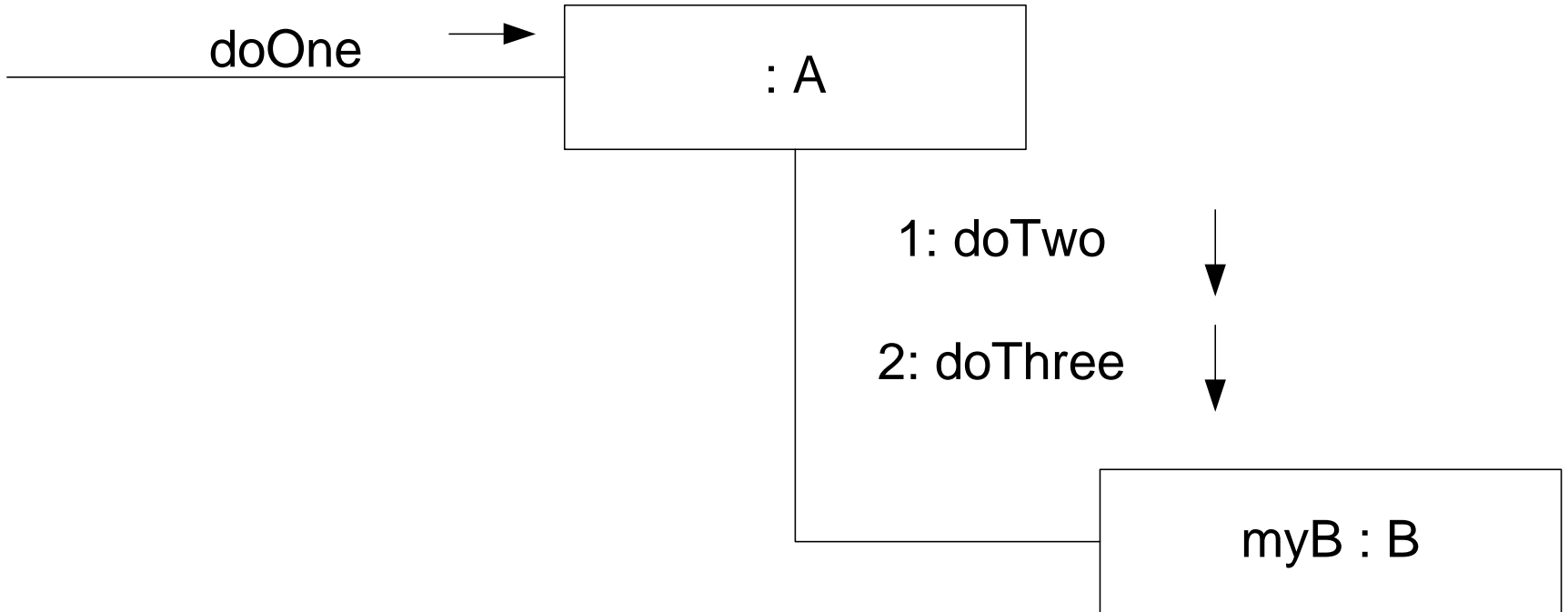


Fig. 15.3



Fig. 15.4

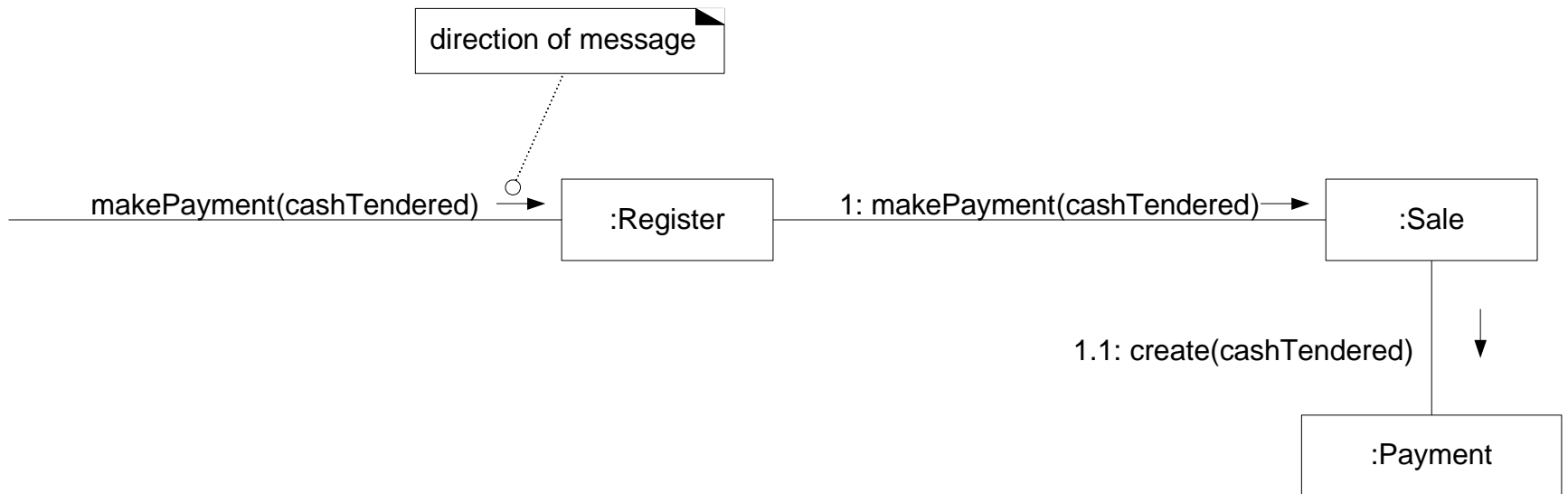


Fig. 15.5

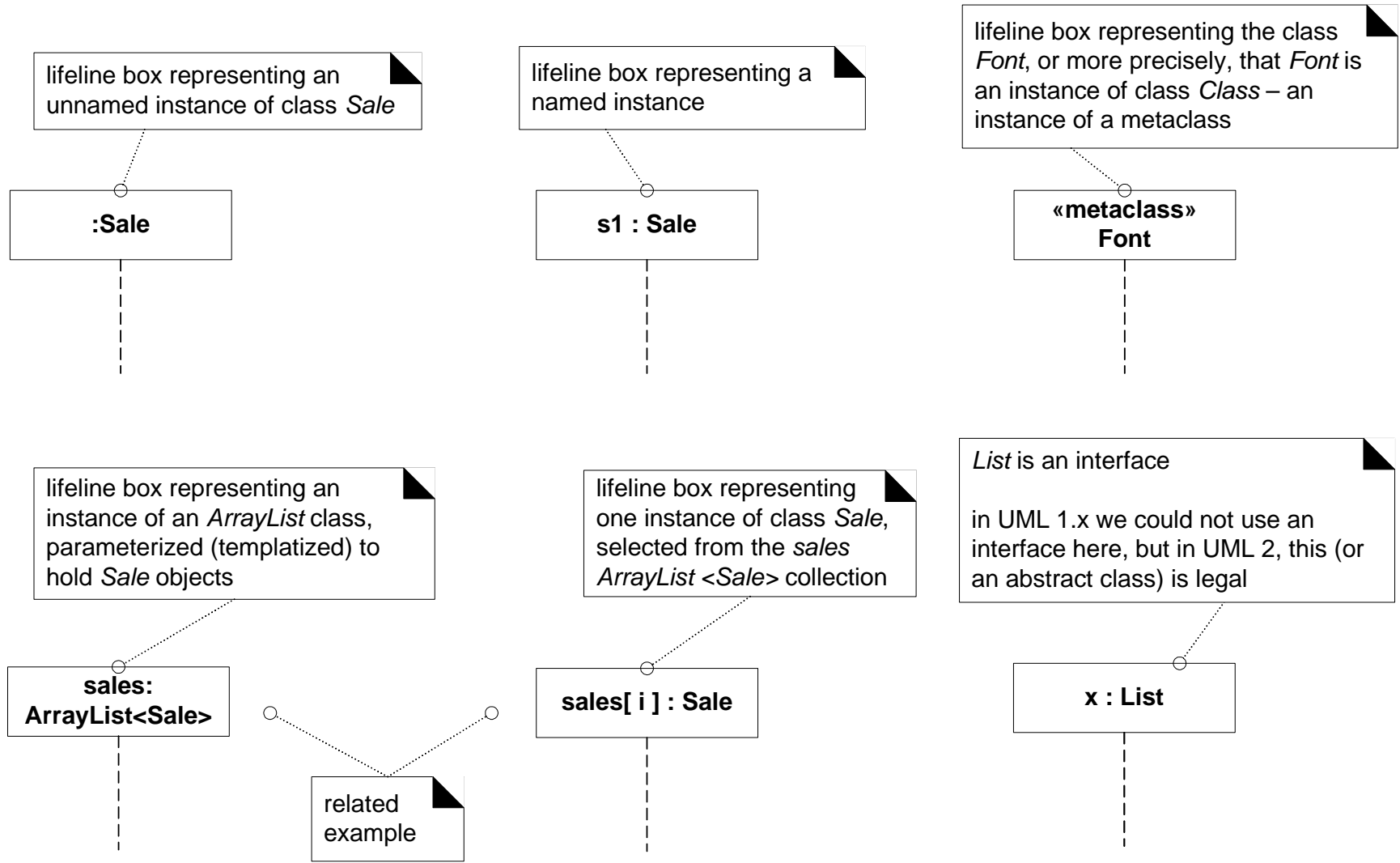


Fig. 15.6

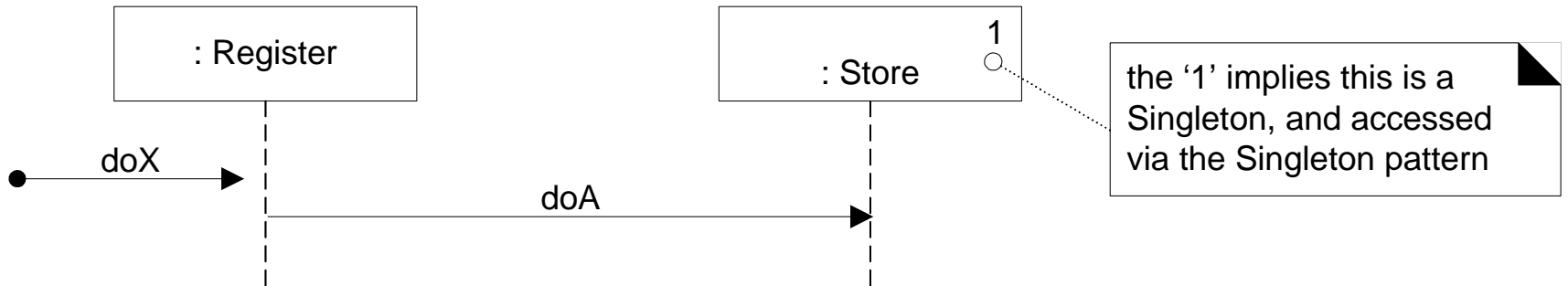


Fig. 15.7

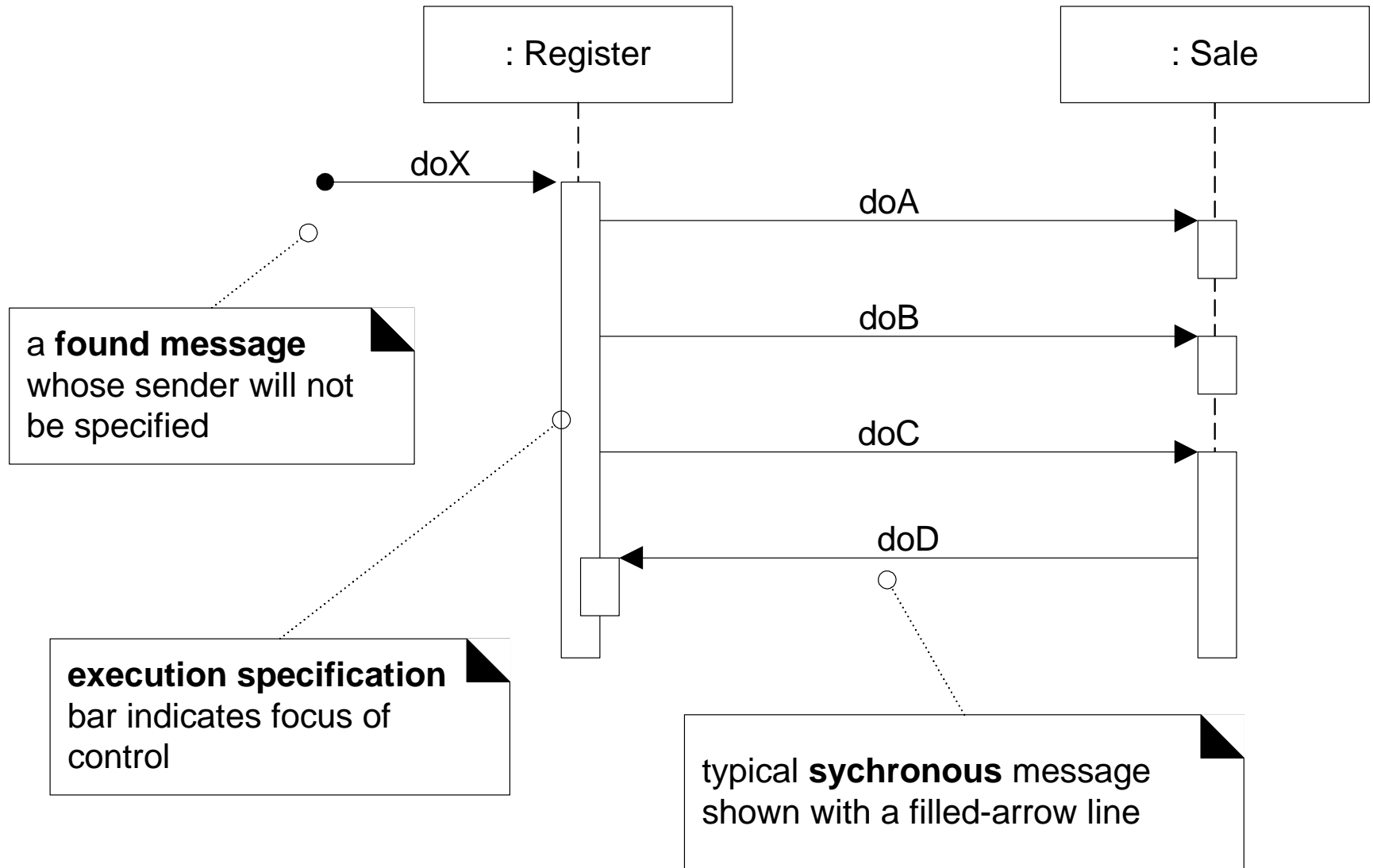


Fig. 15.8

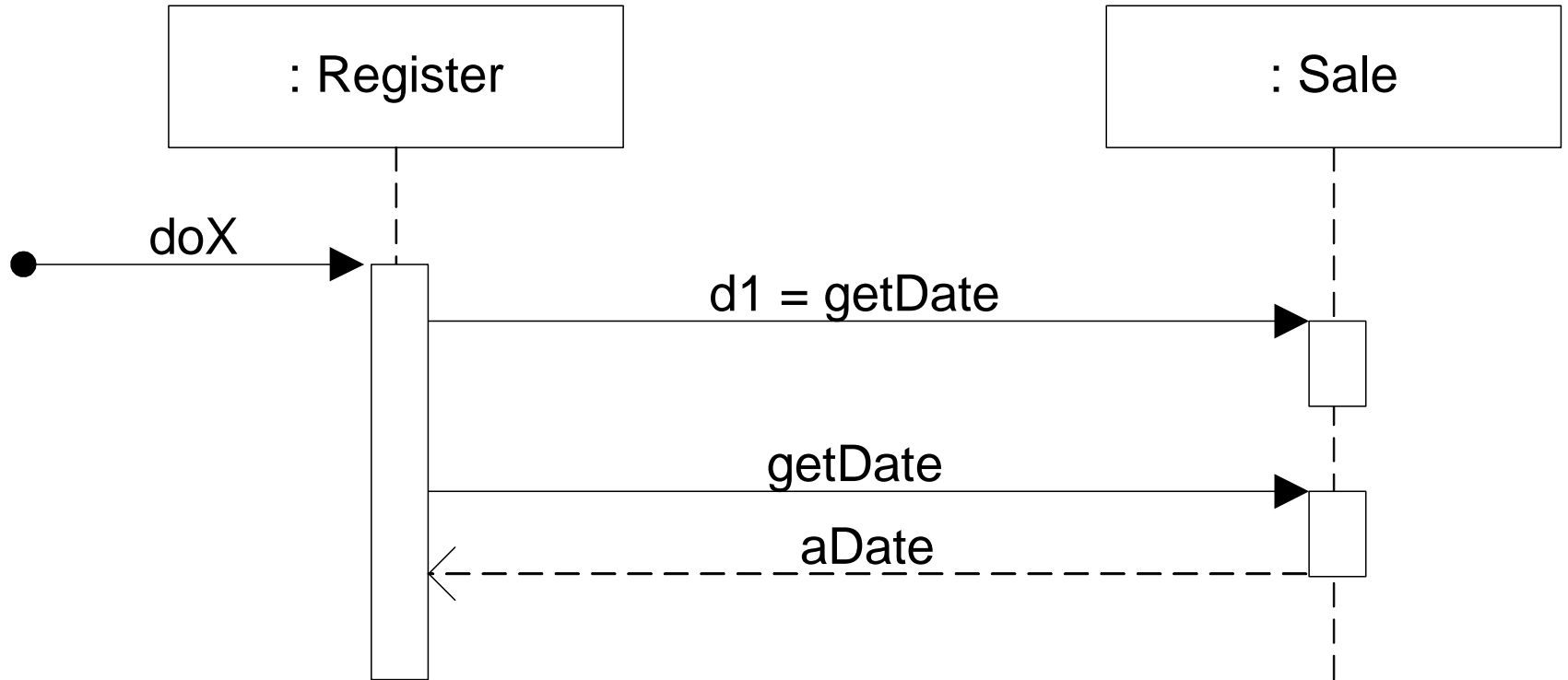


Fig. 15.9

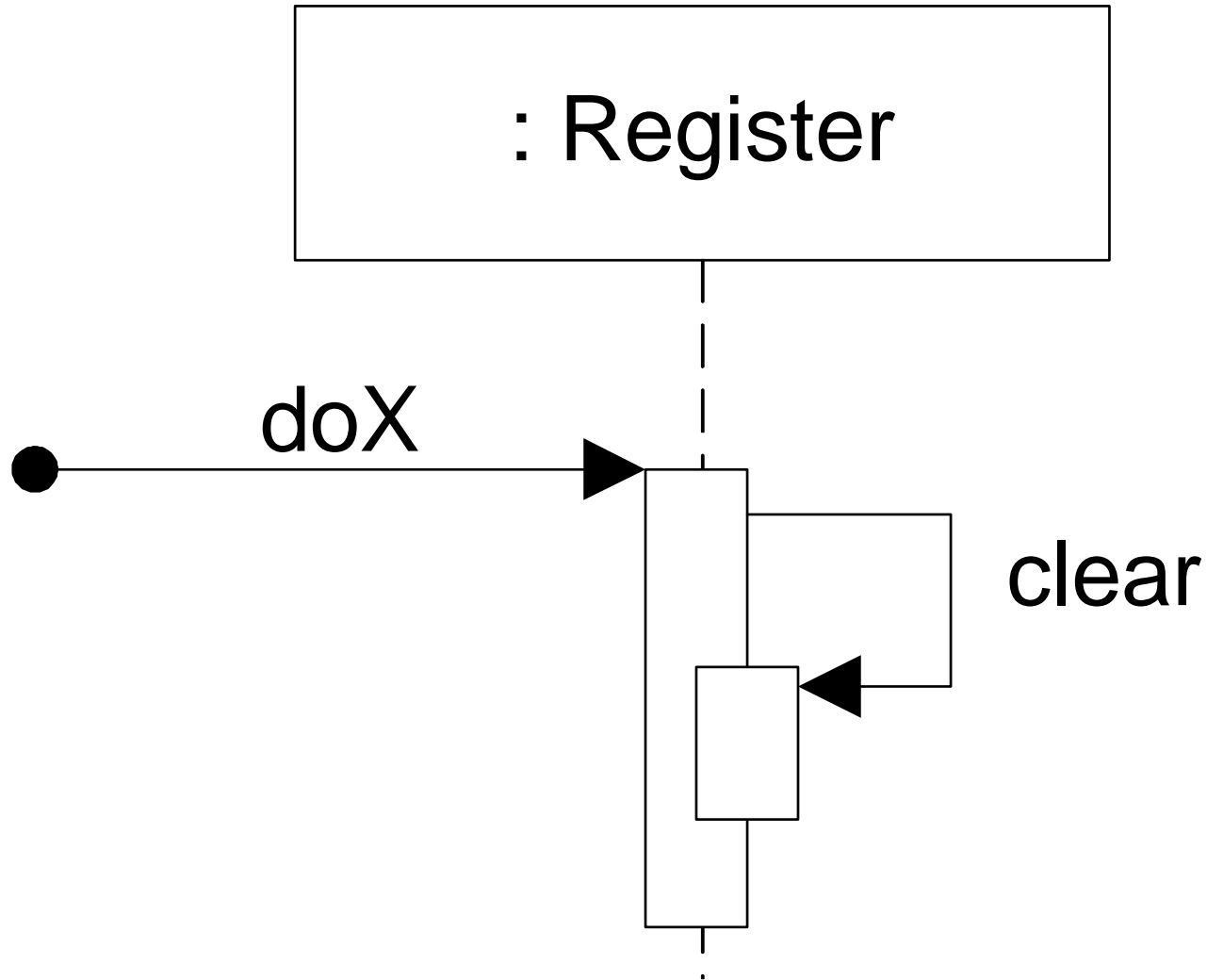


Fig. 15.10

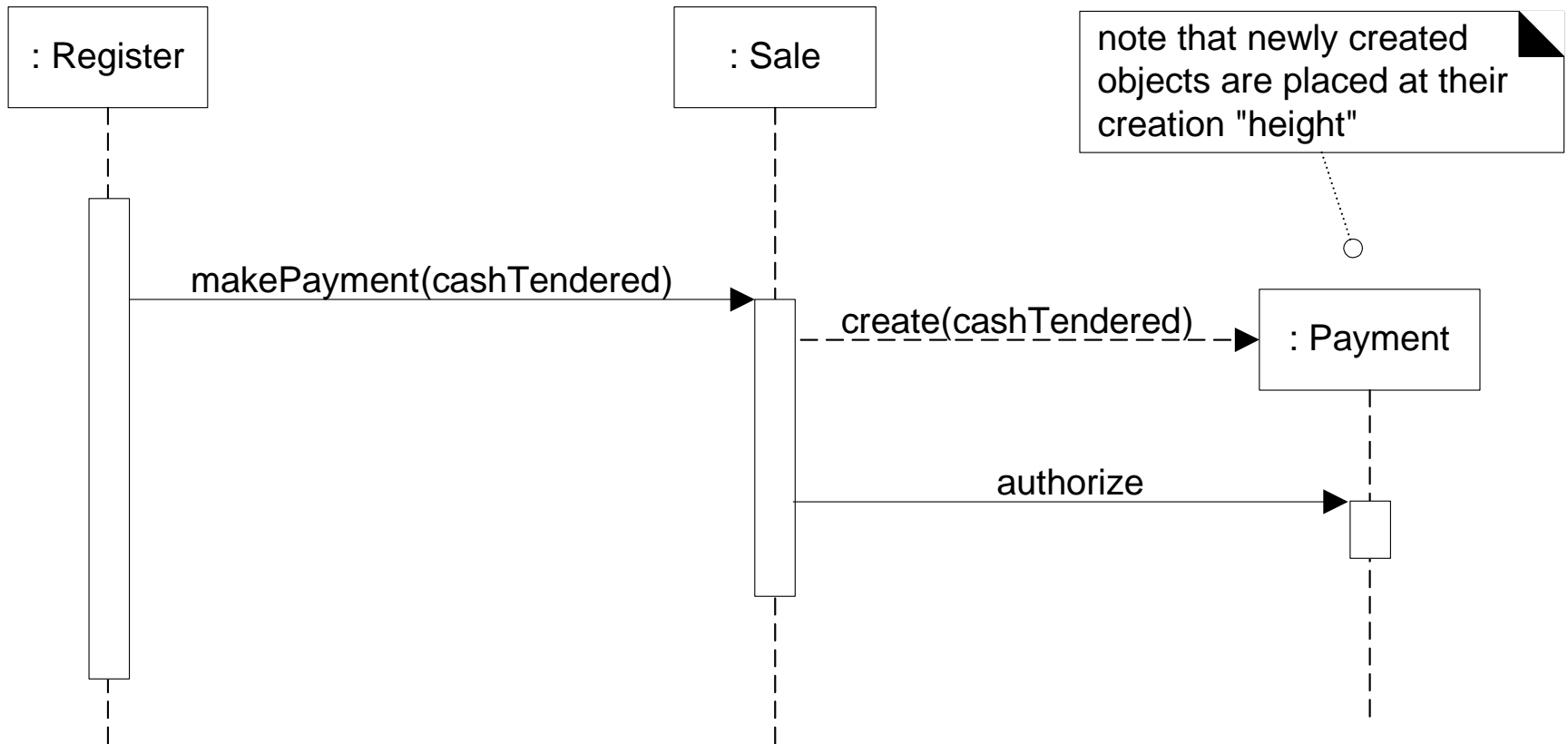


Fig. 15.11

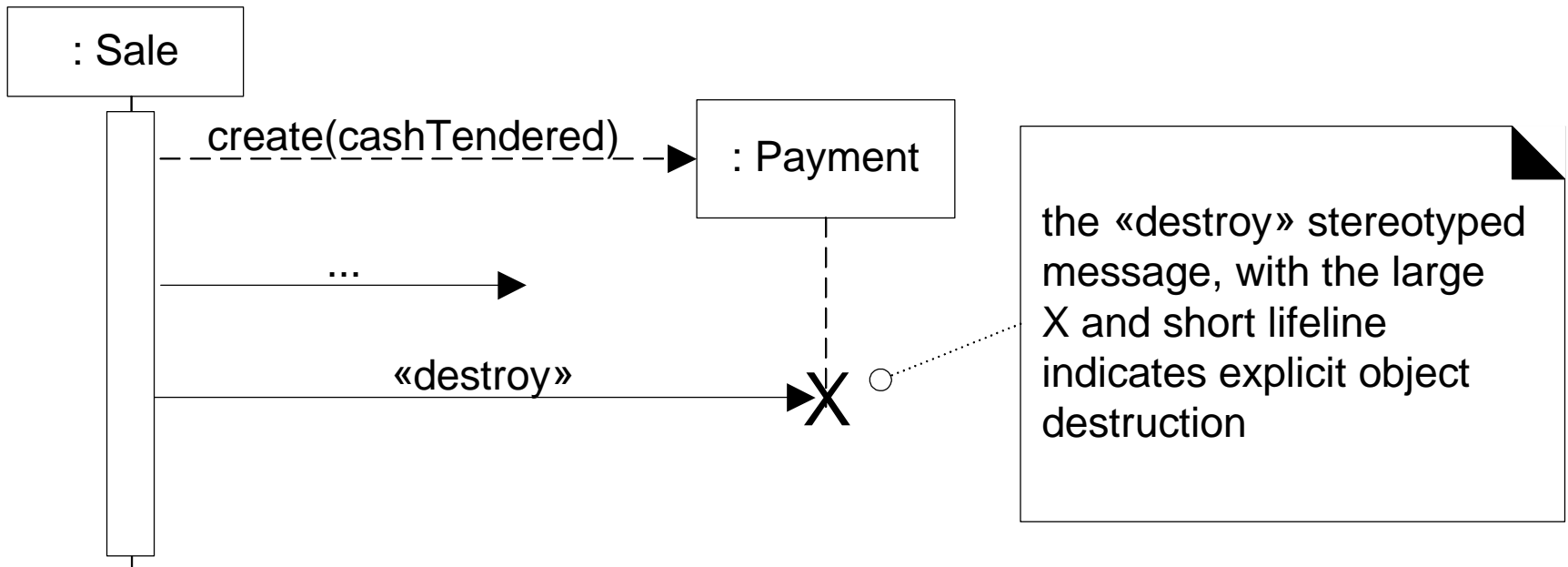


Fig. 15.12

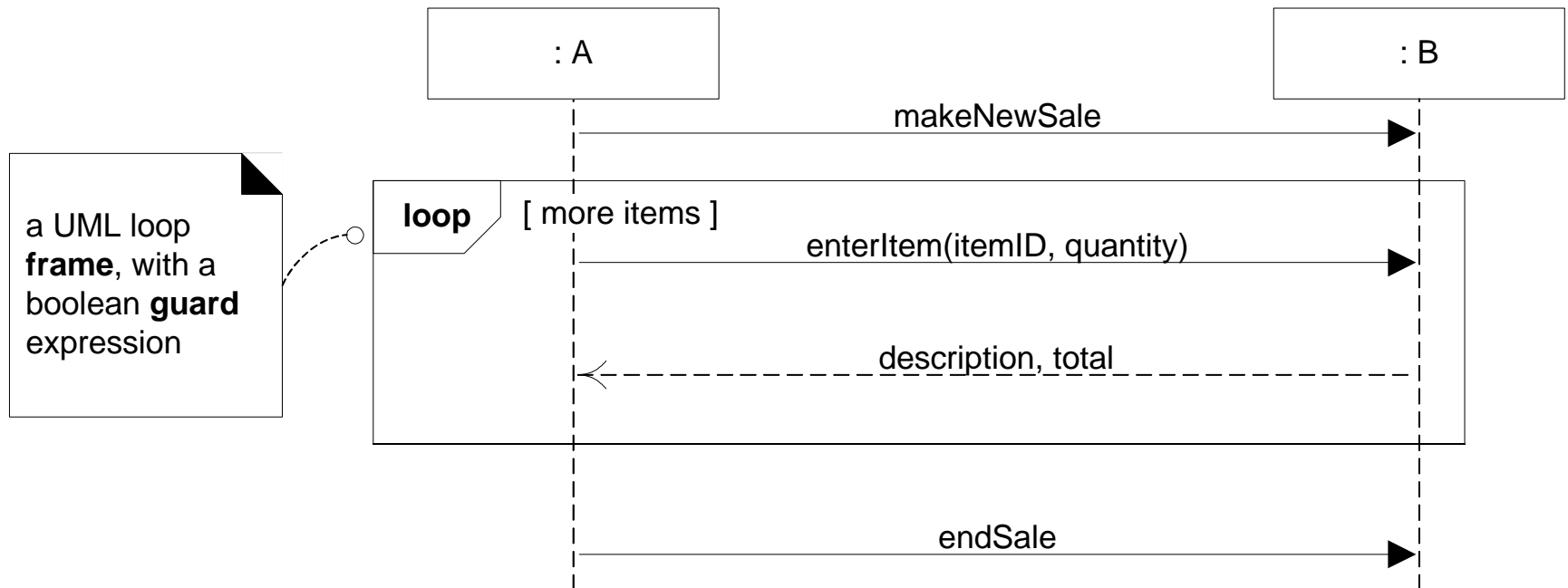


Fig. 15.13

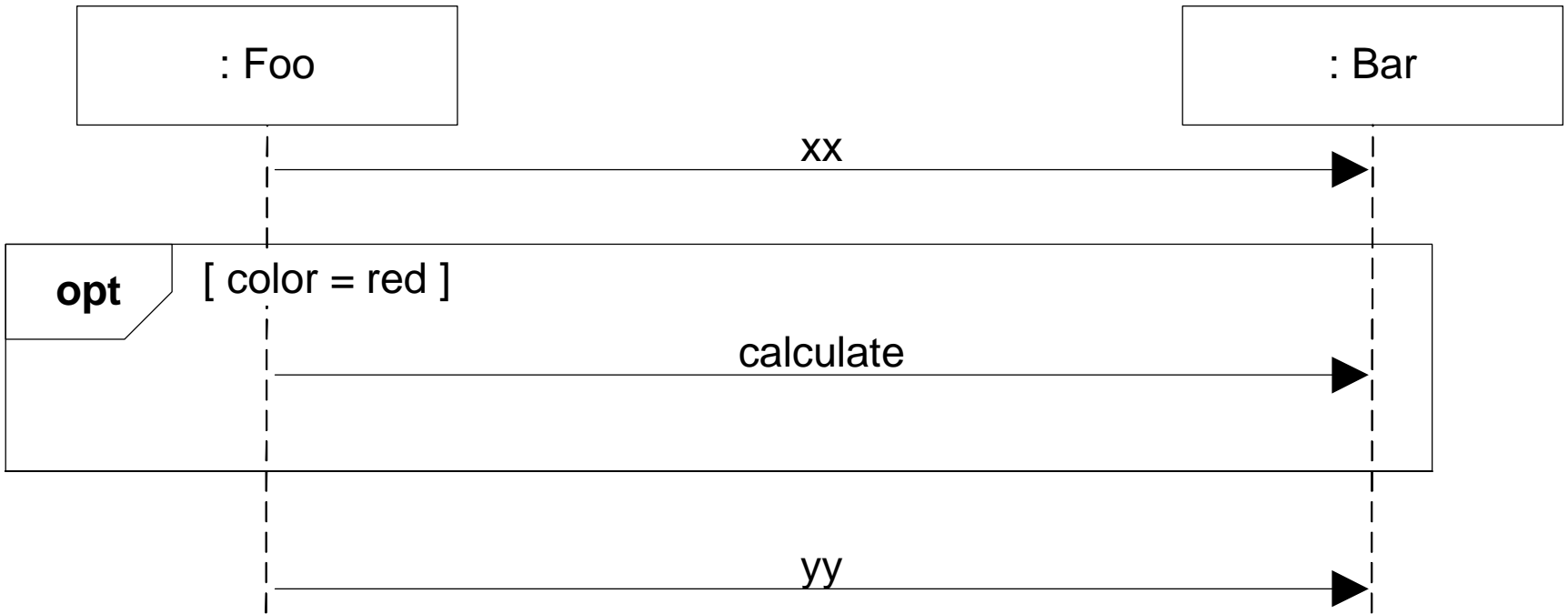


Fig. 15.14

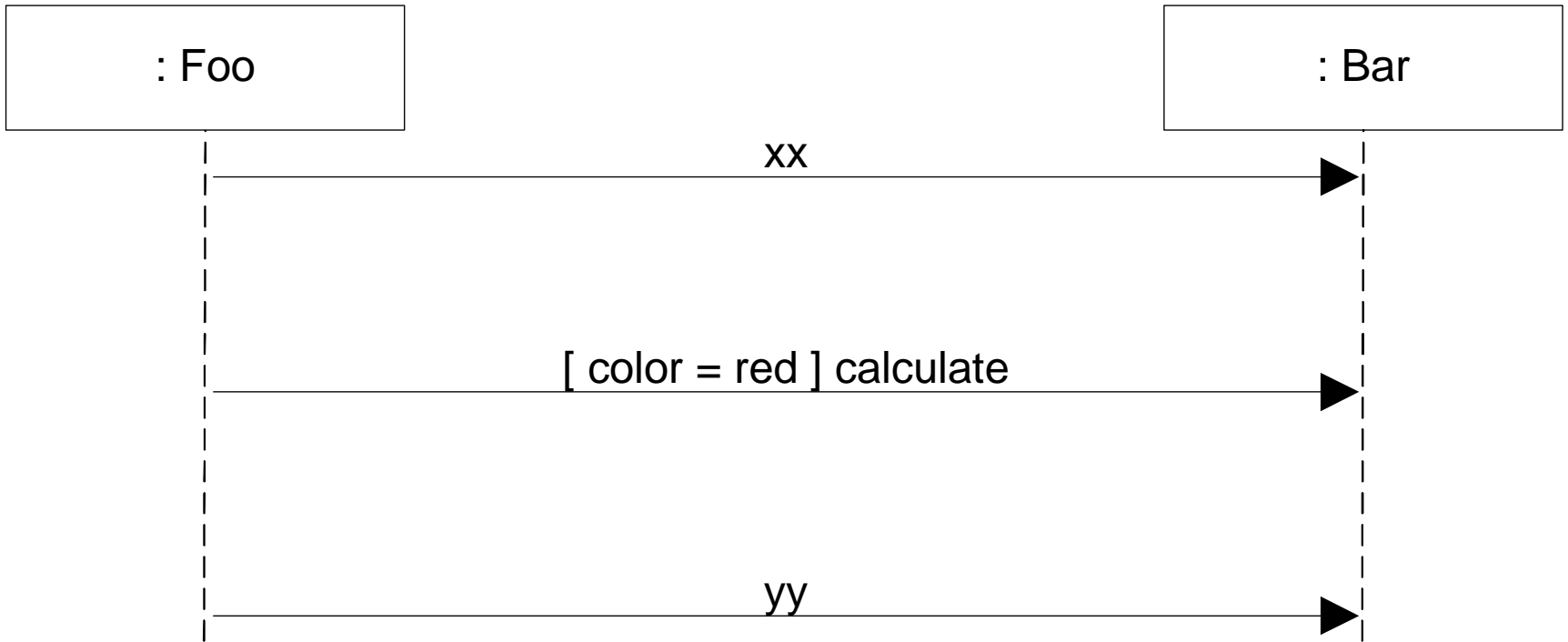


Fig. 15.15

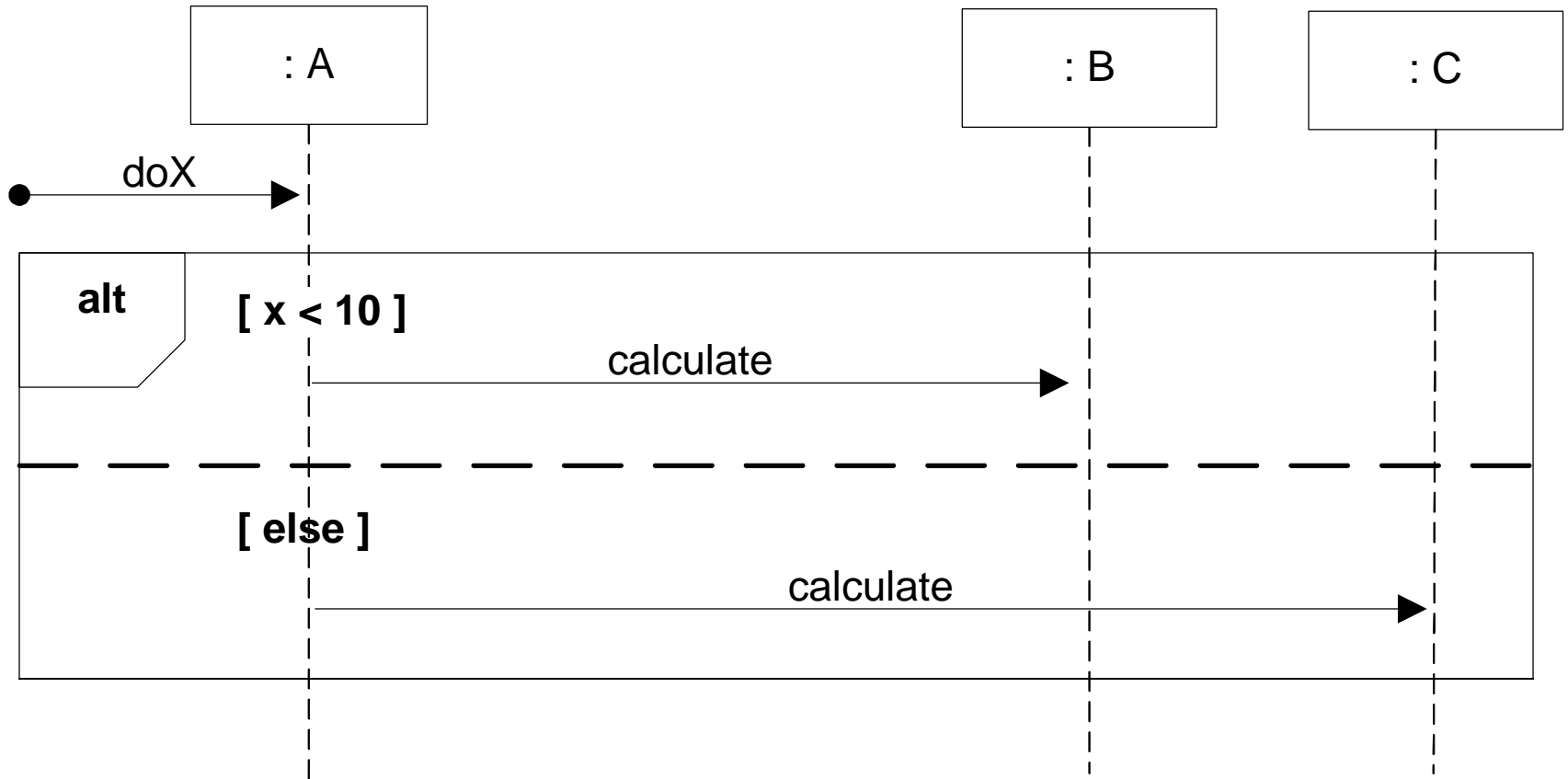


Fig. 15.16

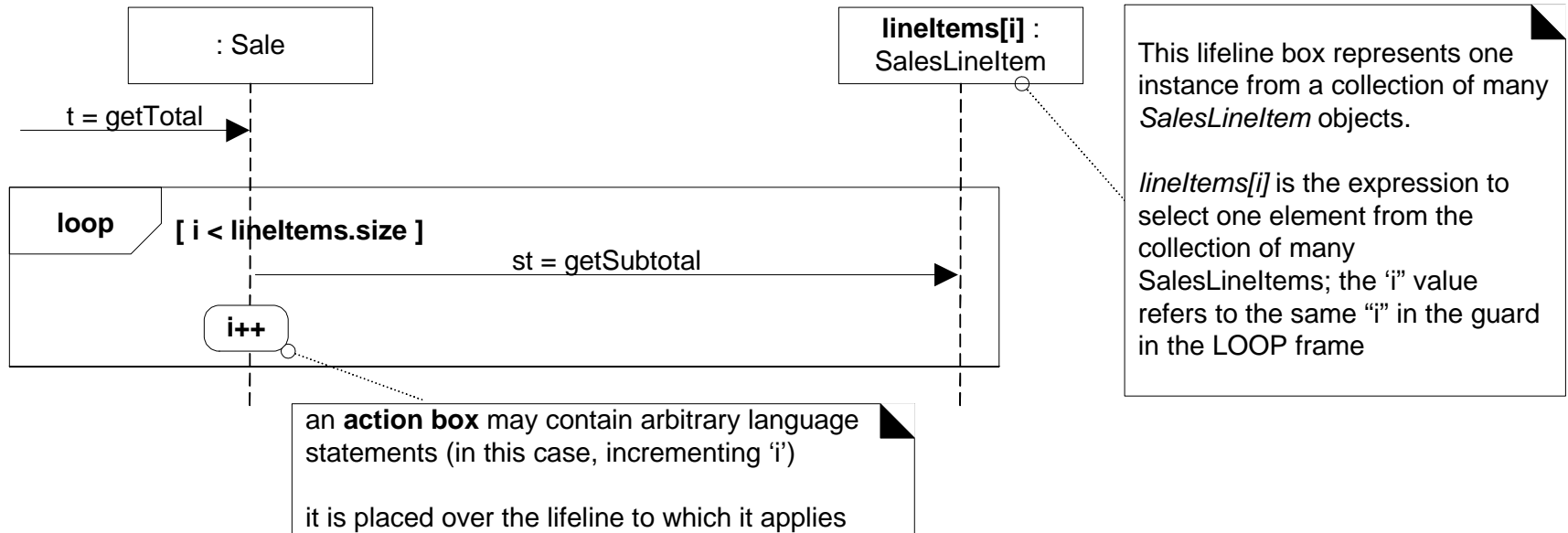


Fig. 15.17

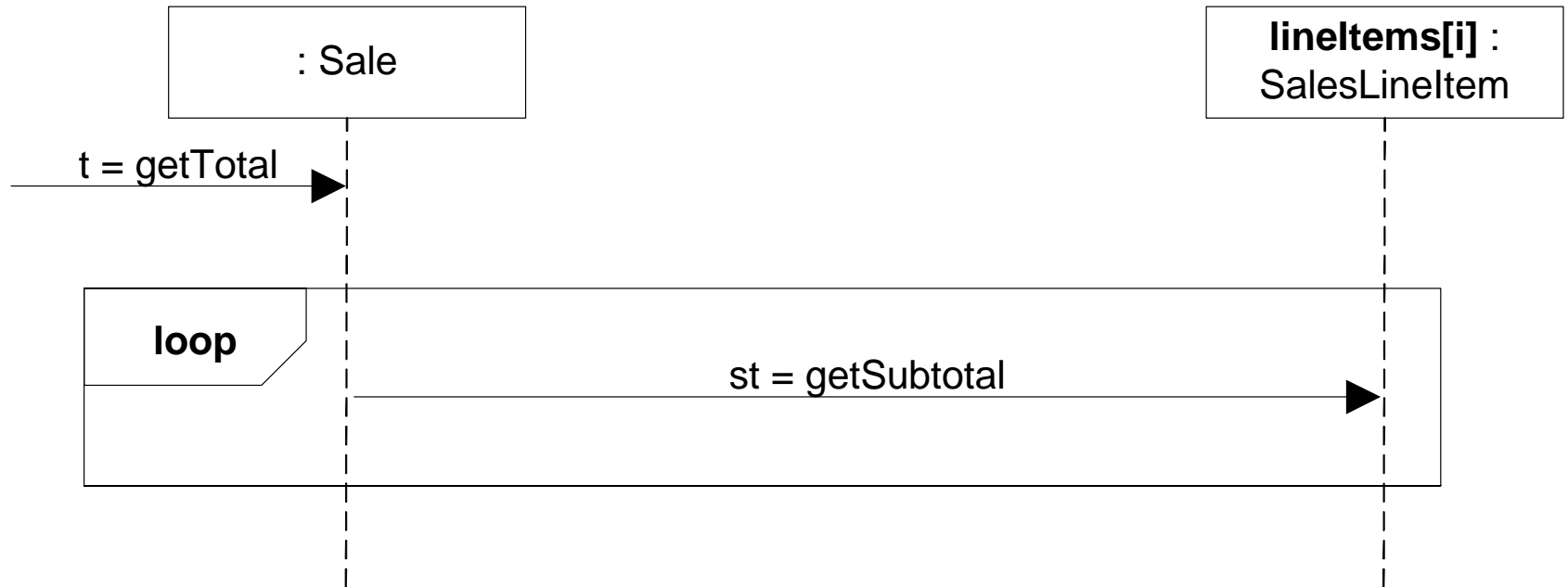


Fig. 15.18

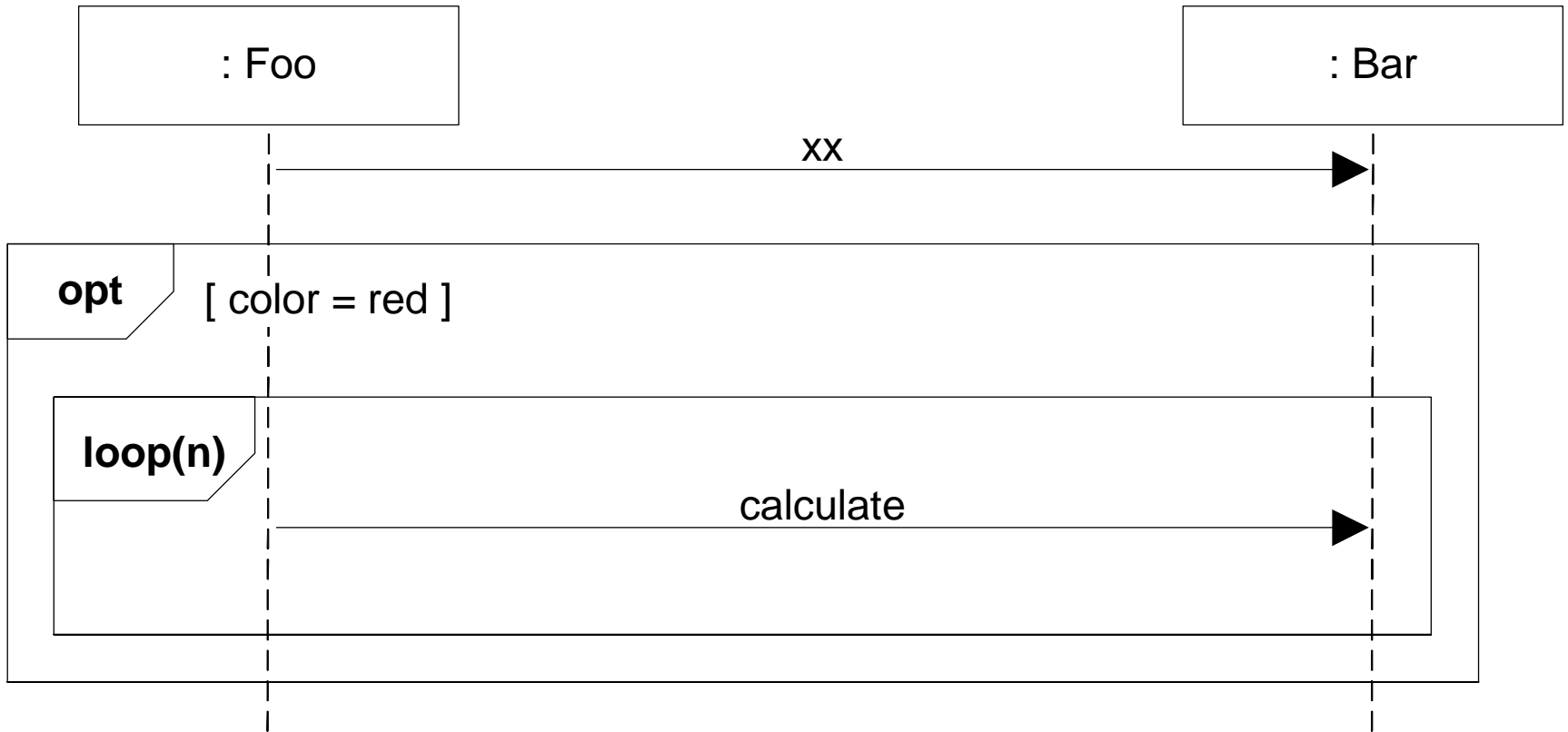
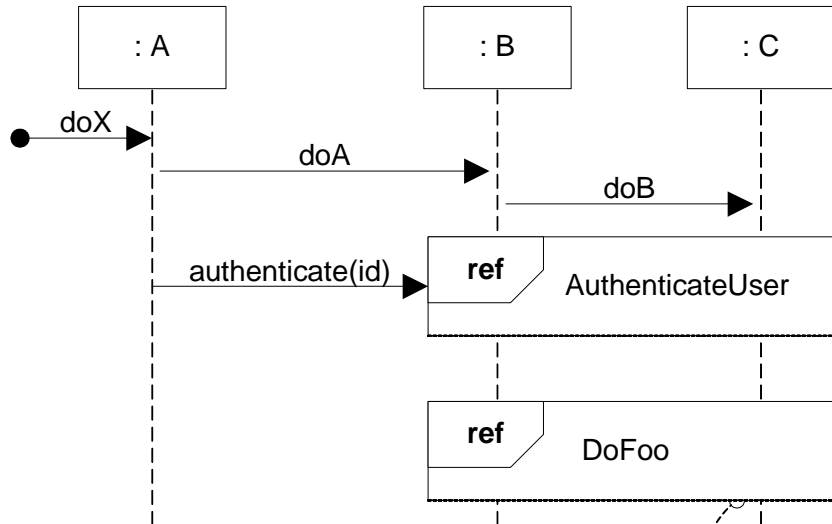


Fig. 15.19



interaction occurrence

note it covers a set of lifelines

note that the sd frame it relates to has the same lifelines: B and C

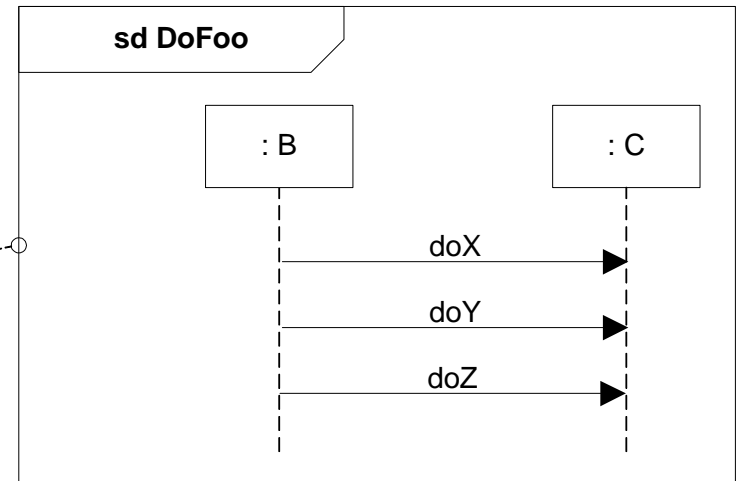
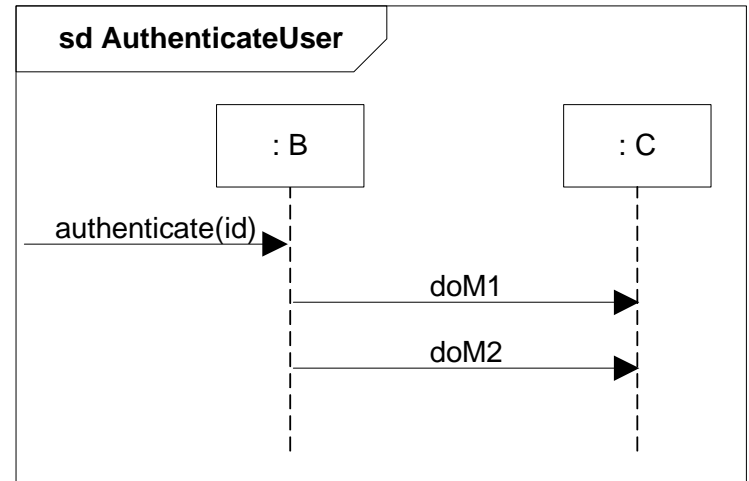


Fig. 15.20

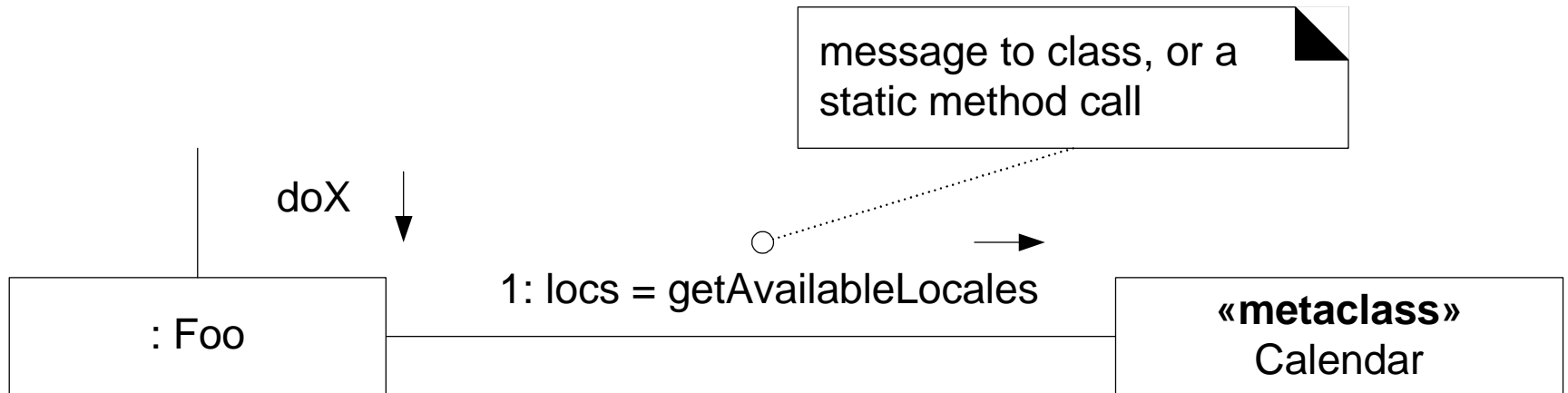


Fig. 15.21

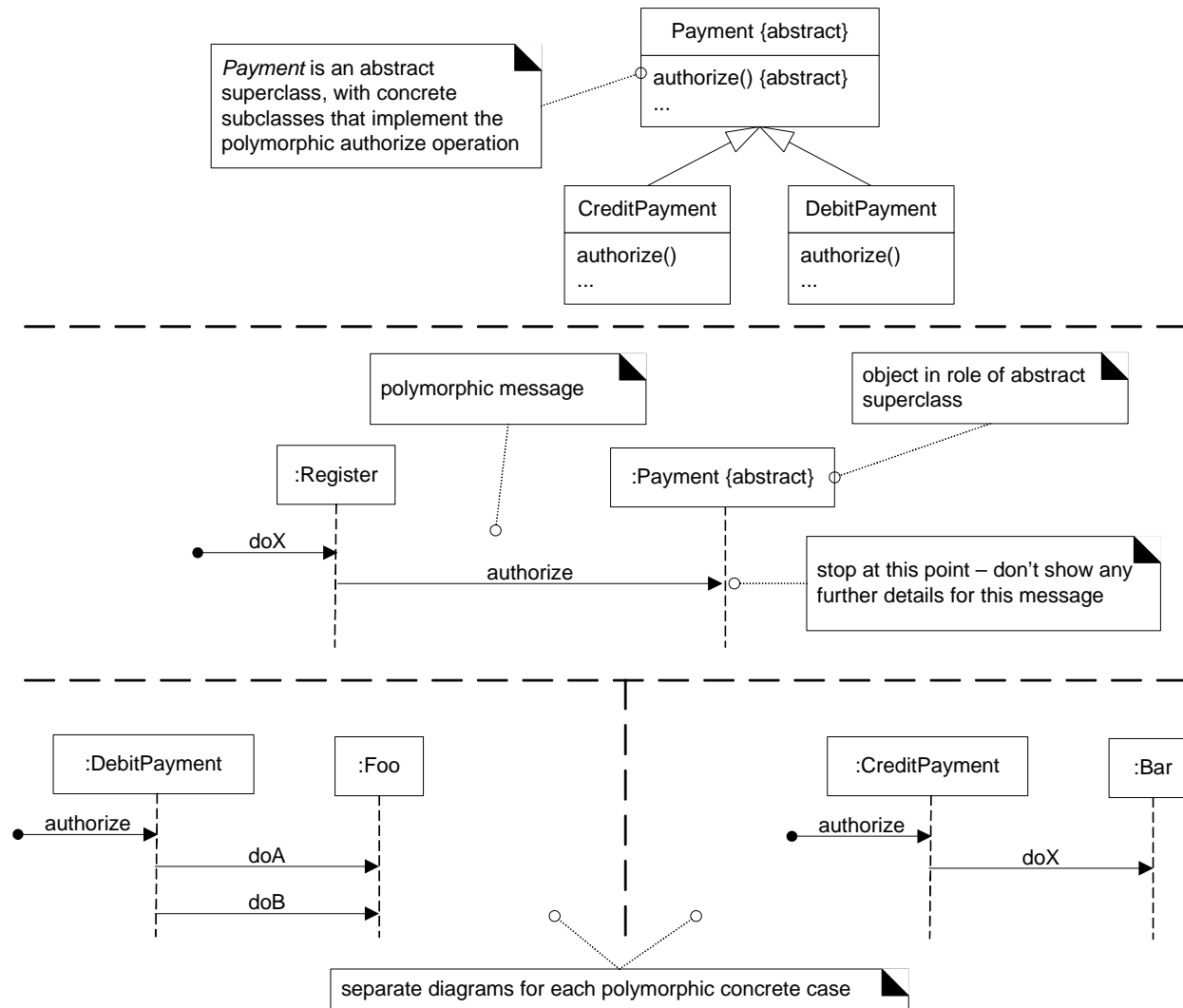


Fig. 15.22

a stick arrow in UML implies an asynchronous call

a filled arrow is the more common synchronous call

In Java, for example, an asynchronous call may occur as follows:

```
// Clock implements the Runnable interface  
Thread t = new Thread( new Clock() );  
t.start();
```

the asynchronous *start* call always invokes the *run* method on the *Runnable* (*Clock*) object

to simplify the UML diagram, the *Thread* object and the *start* message may be avoided (they are standard “overhead”); instead, the essential detail of the *Clock* creation and the *run* message imply the asynchronous call

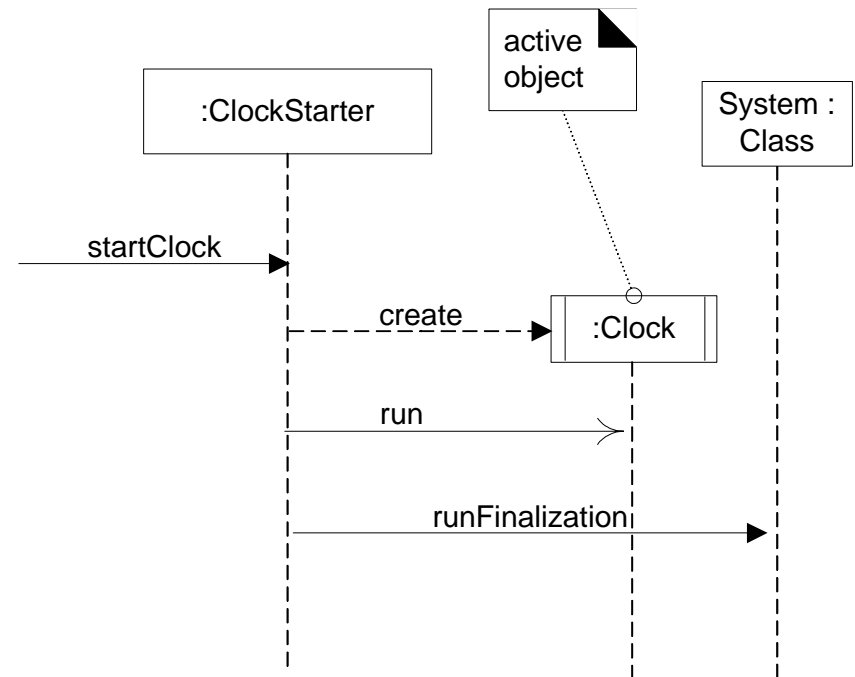


Fig. 15.23

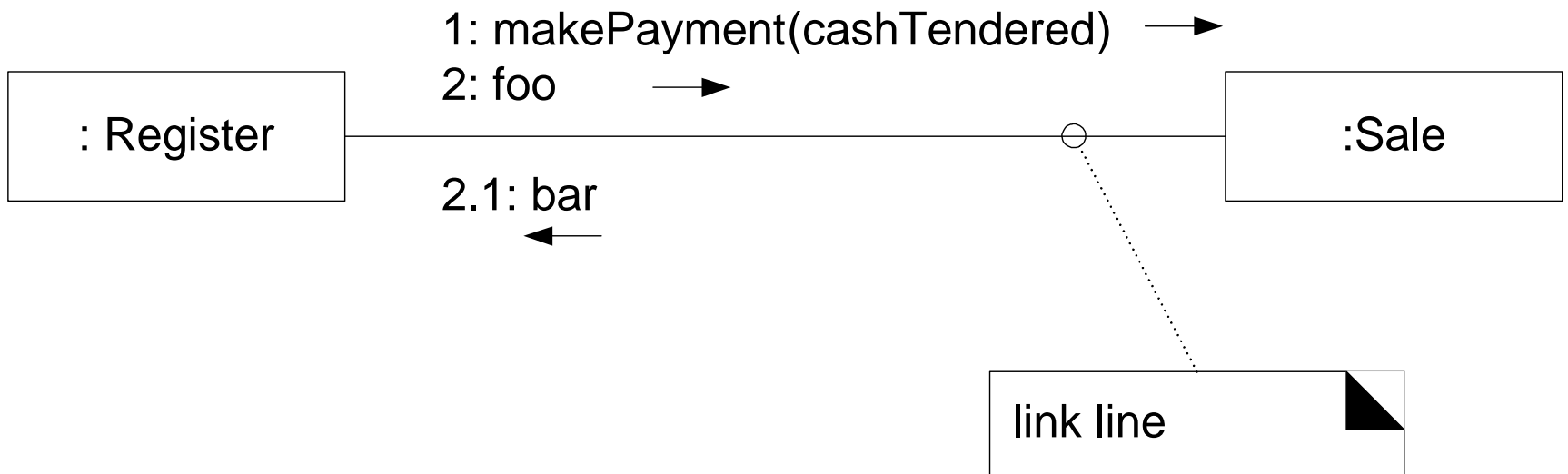


Fig. 15.24

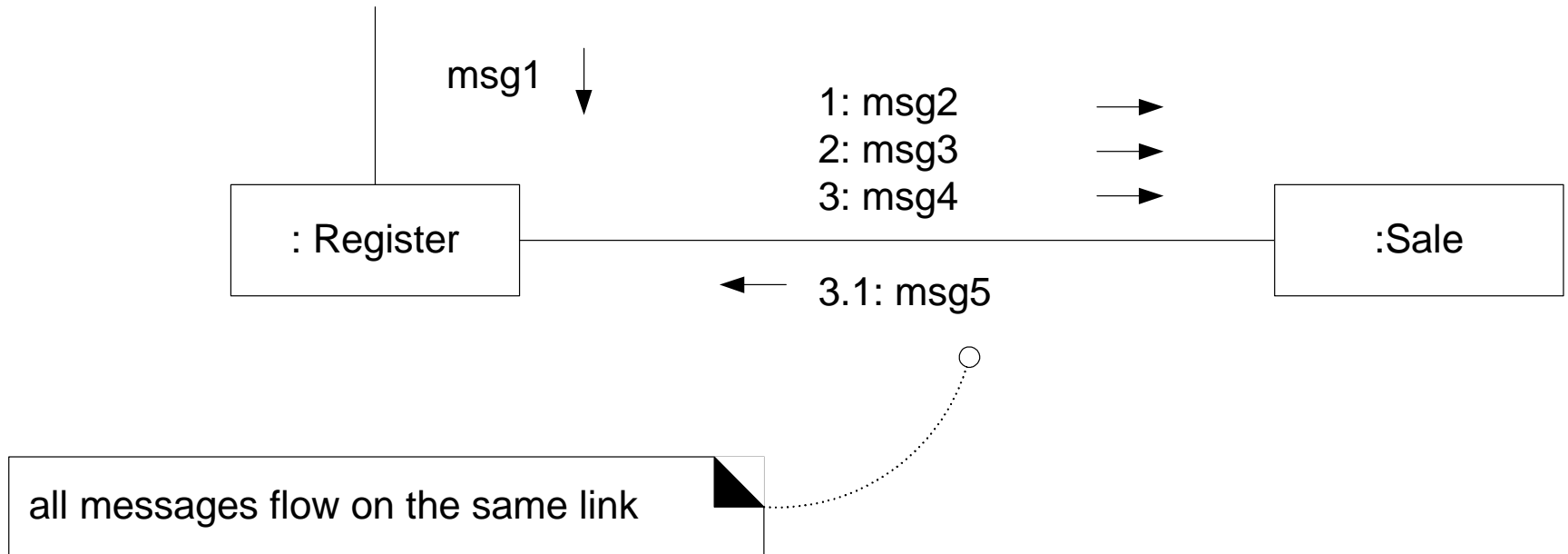


Fig. 15.25

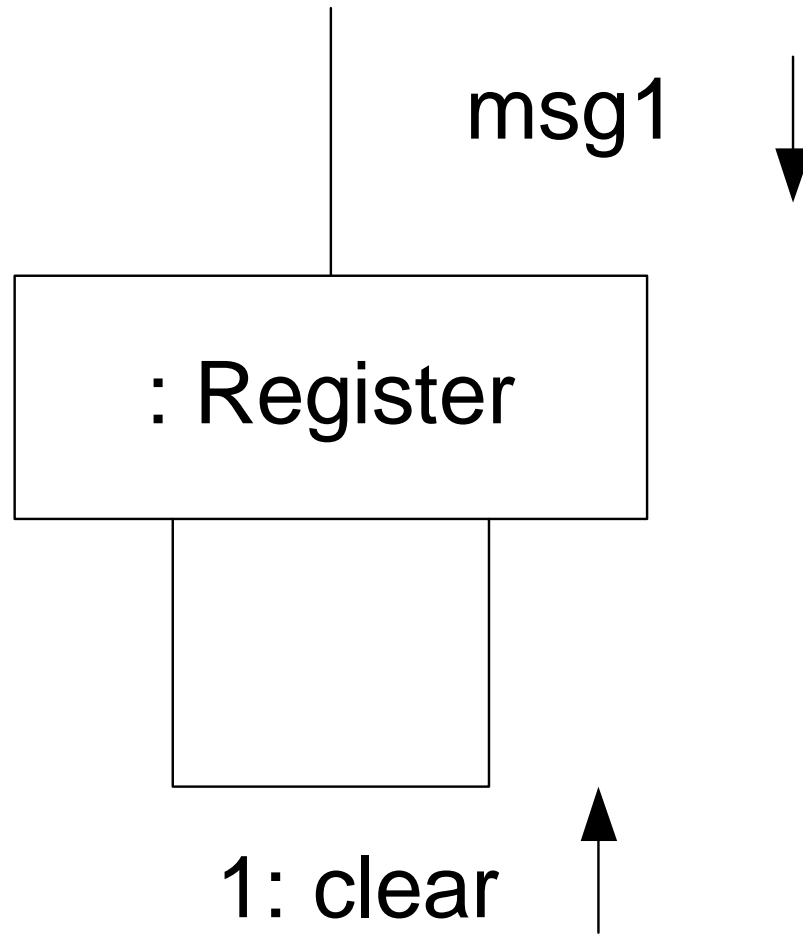
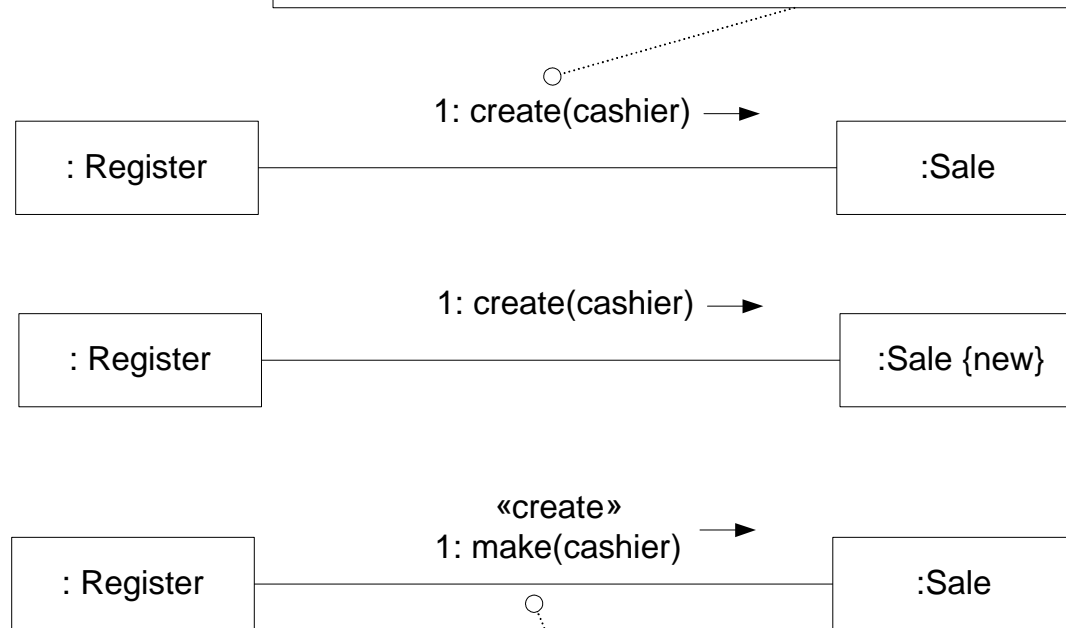


Fig. 15.26

Three ways to show creation in a communication diagram

create message, with optional initializing parameters. This will normally be interpreted as a constructor call.



if an unobvious creation message name is used, the message may be stereotyped for clarity

Fig. 15.27

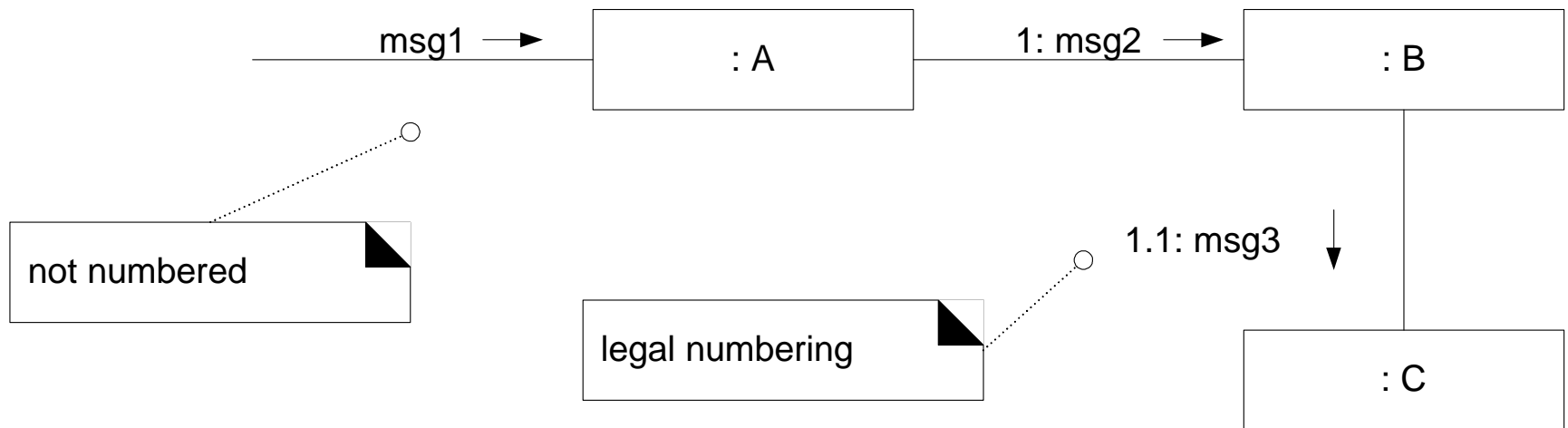


Fig. 15.28

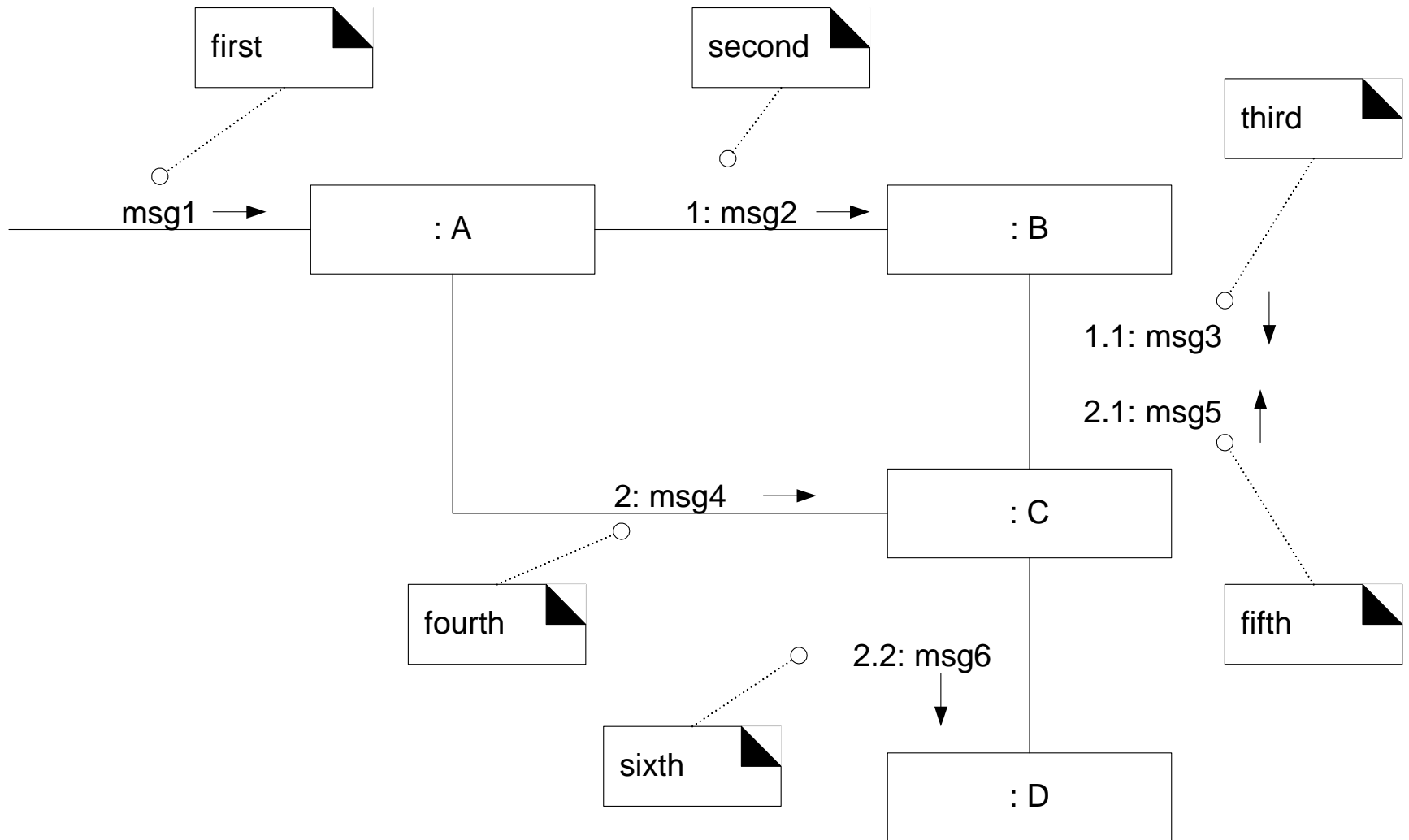


Fig. 15.29

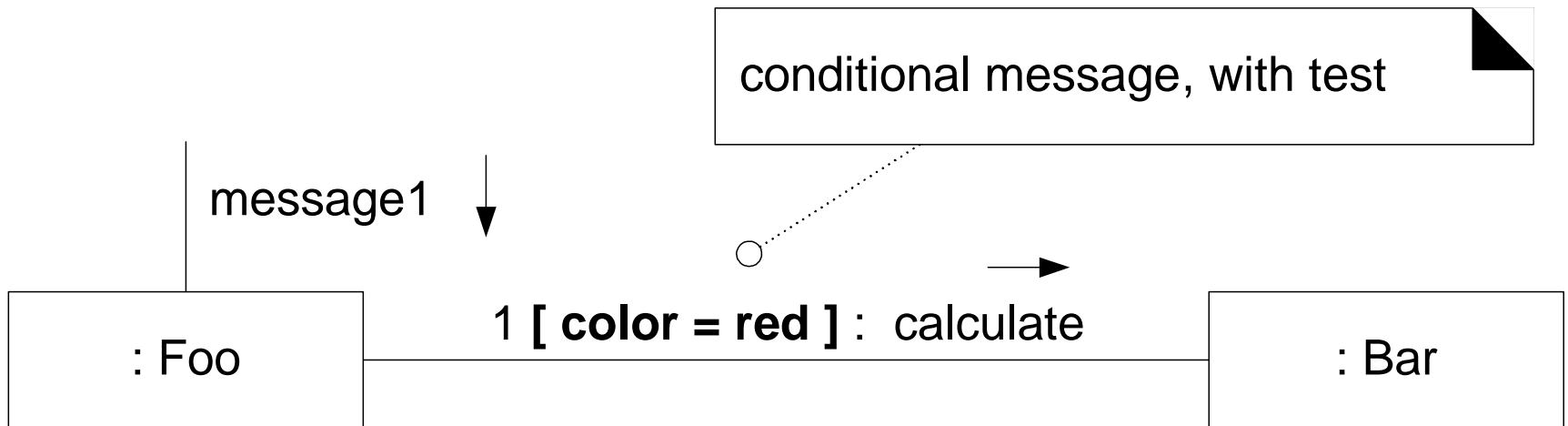


Fig. 15.30

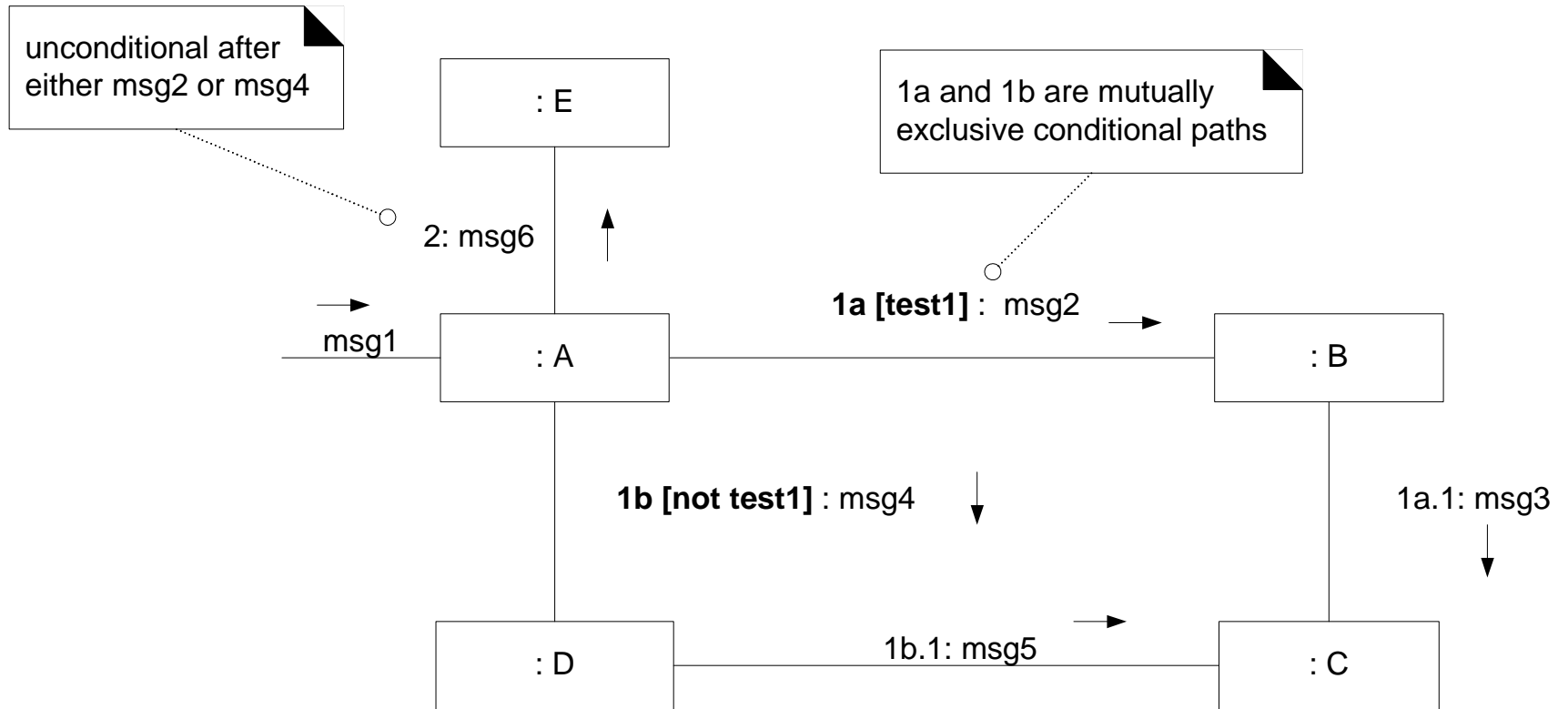


Fig. 15.31

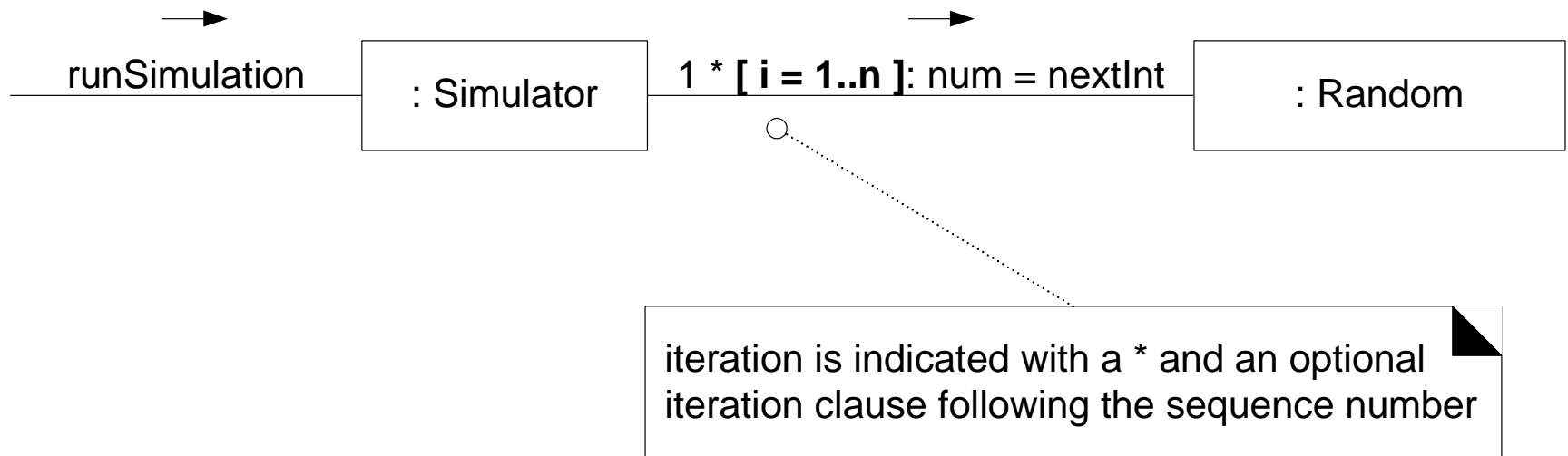


Fig. 15.32

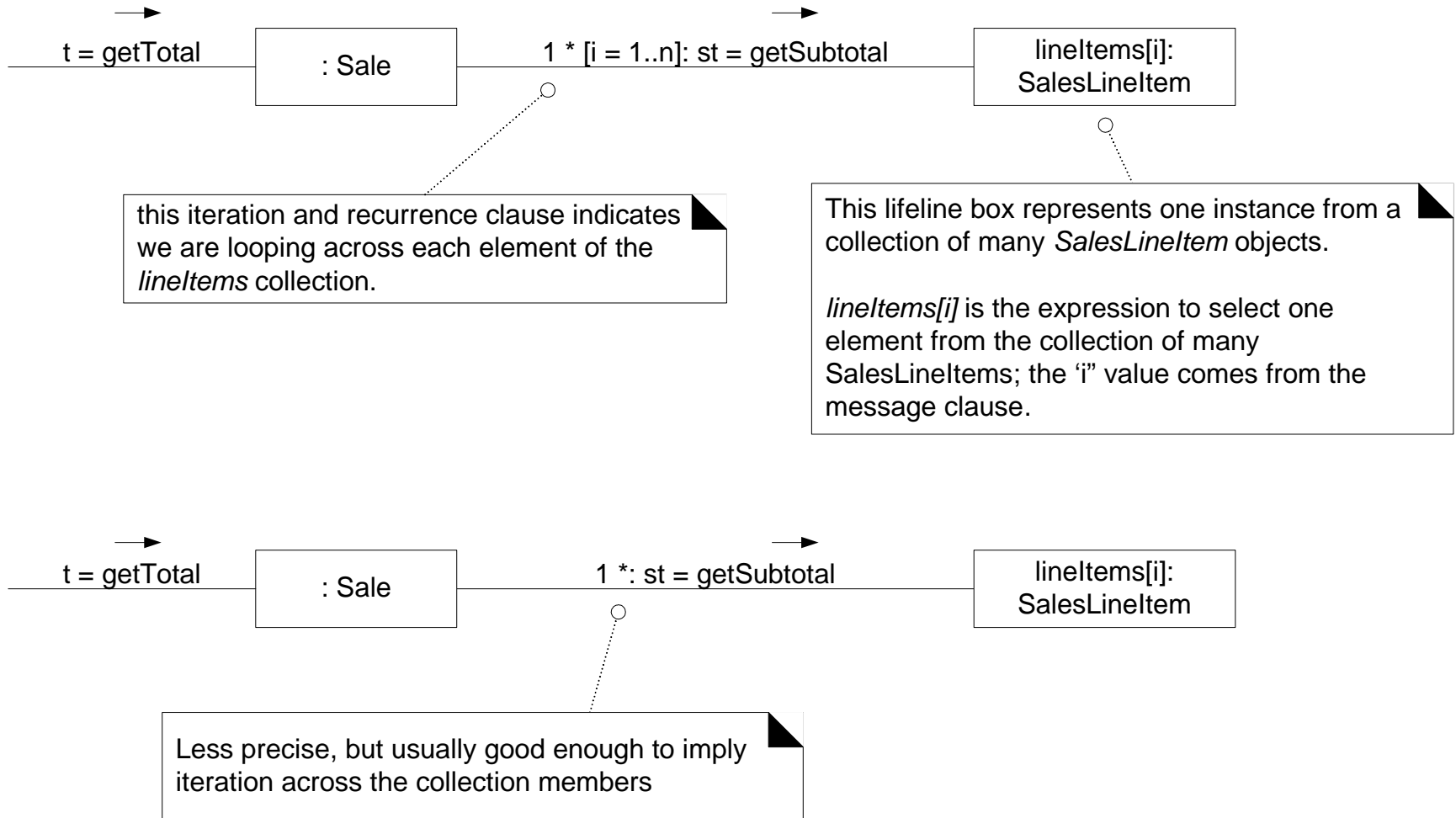


Fig. 15.33

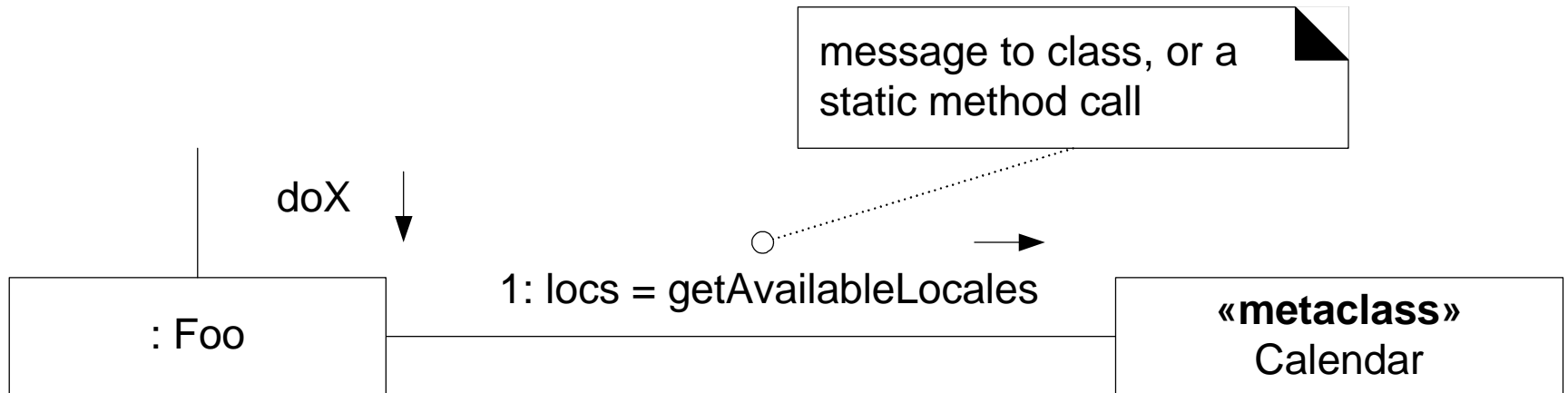


Fig. 15.34

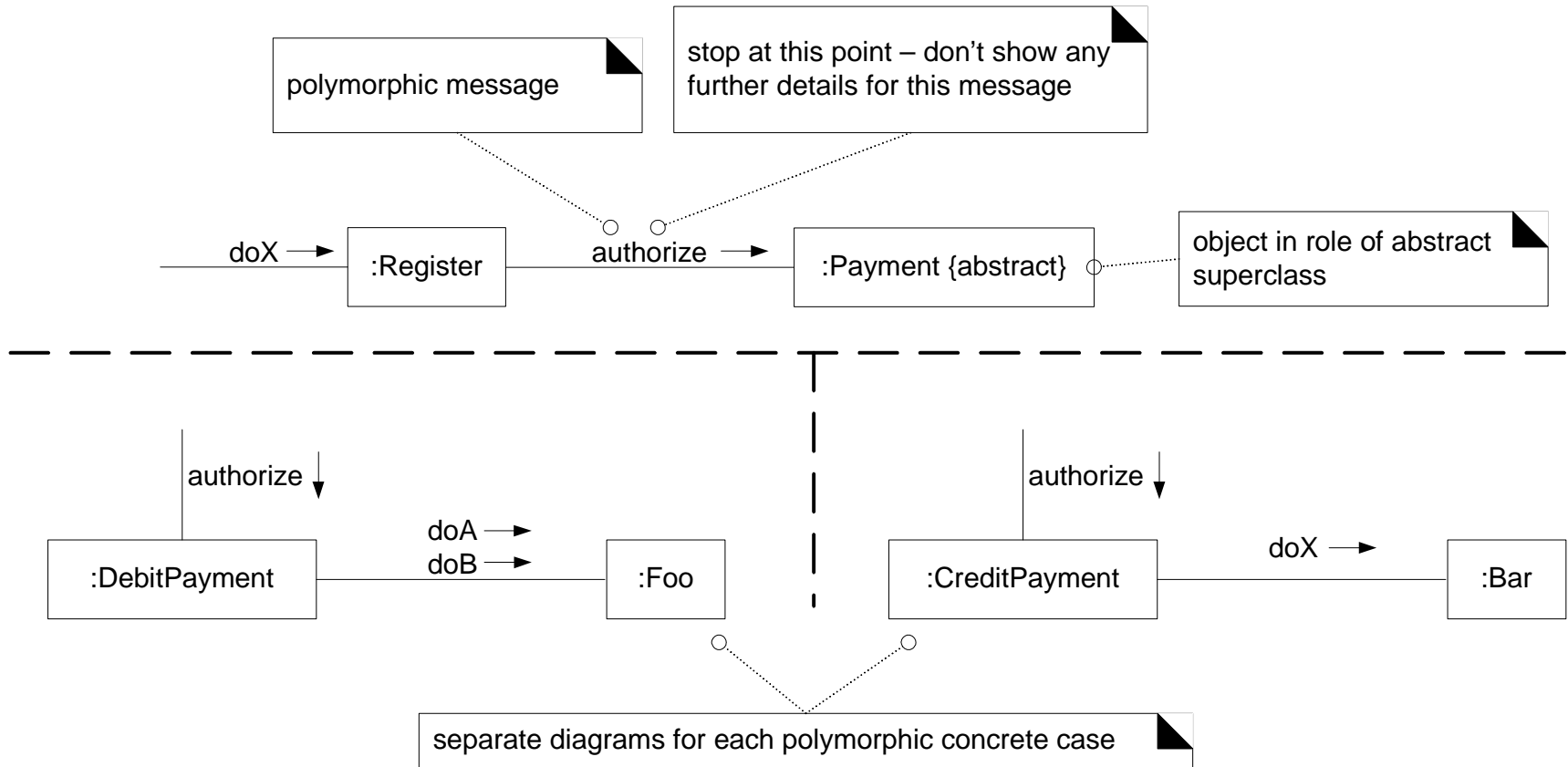


Fig. 15.35

