**Chapter 14:**

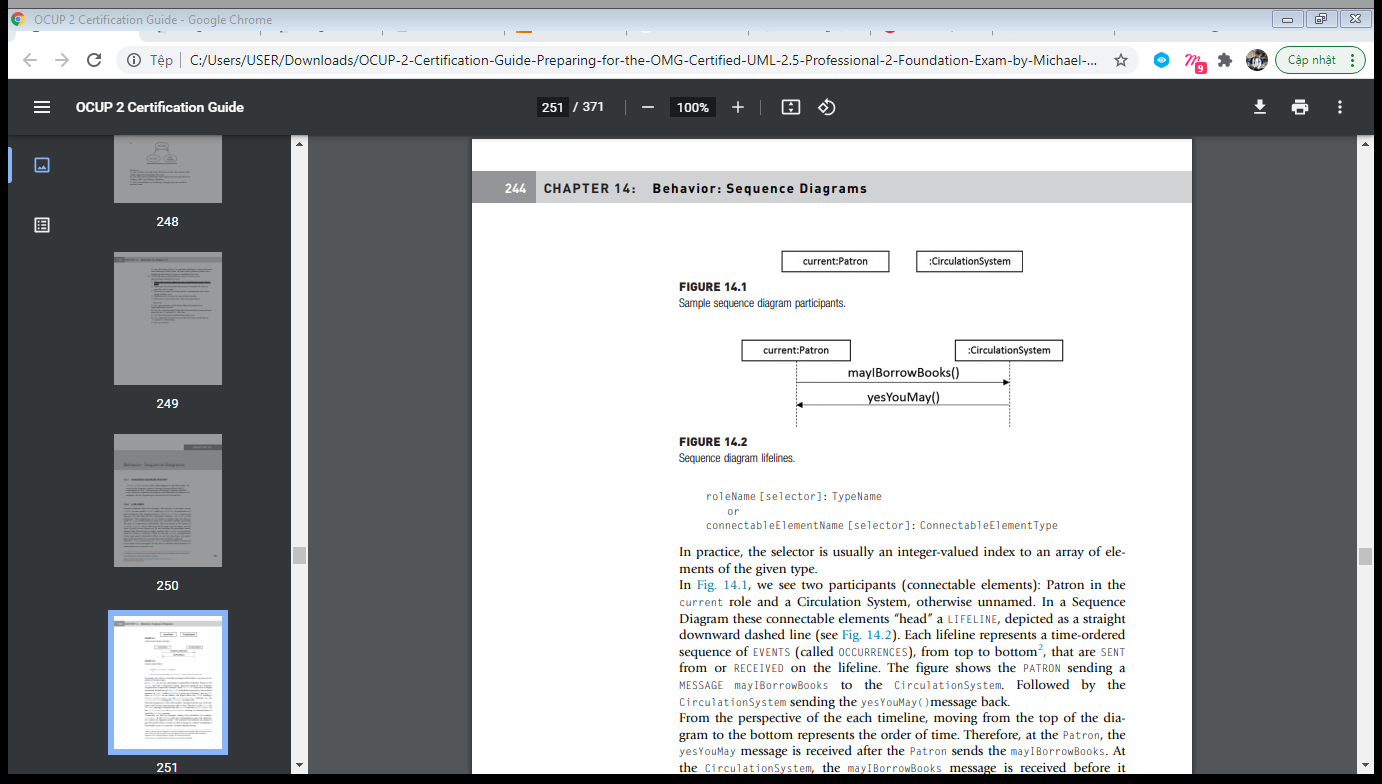
**Hành vi: Sơ đồ trình tự**

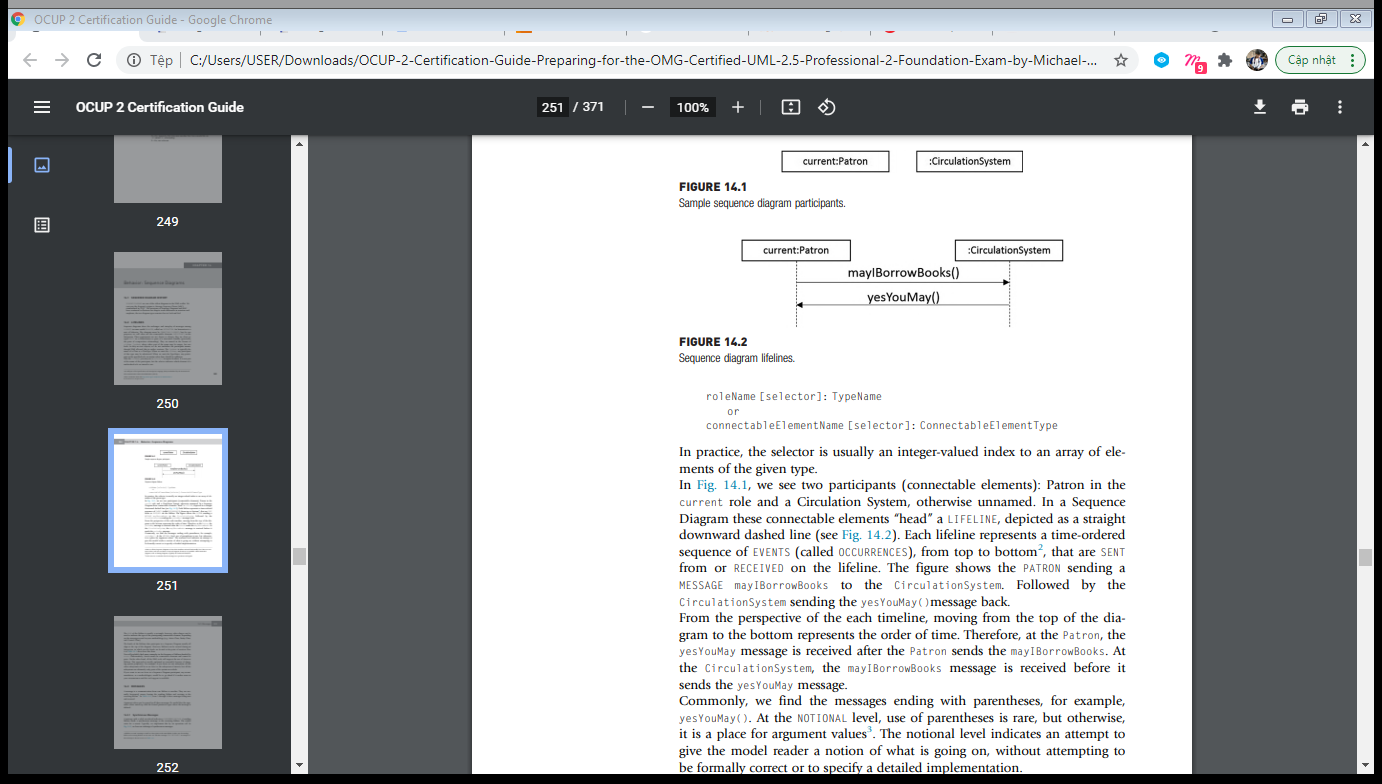
**14.1 Lịch sử sơ đồ trình tự**

SƠ ĐỒ trình tự là một trong những sơ đồ lâu đời nhất trong bộ công cụ UML. Chúng ta có thể theo dõi nguồn gốc của sơ đồ tới Biểu đồ trình tự thông báo (MSC) tiêu chuẩn hóa năm 1993. Ký hiệu của Sơ đồ trình tự và MSC đã tiếp tục dao động nhưng mặc dù có một số khác biệt trong ký hiệu và nhấn mạnh, hai loại sơ đồ vẫn gần giống nhau trong giao diện.

**14.2 Lifeline**

Biểu đồ trình tự cho thấy sự trao đổi và tác động lẫn nhau của các thông điệp giữa các CÁC YẾU TỐ trong một số HÀNH VI hữu ích, được gọi là SỰ TƯƠNG TÁC. Một sự tương tác là một đơn vị của hành vi. Các phần tử phải là các phần tử có thê KẾT NỐI, nhưng đối với mục đích của chúng ta, chúng ta thường sẽ gọi các phần tử có thể kết nối là những thành phần tham gia vào sự tương tác. Những thành phần tham gia này không phải là lớp hoặc đối tượng; họ gần hơn với những vai trò đã được nhập trong sự cộng tác hoặc các phần của bộ phân loại có cấu trúc (các bộ phận thực tế của mối quan hệ thành phần). Chúng được đặt tên theo định dạng roleName: TypeName, trong đó một trong hai phần của tên có thể để trống, nhưng không thể cả hai. Vì chúng không phải là đối tượng, chúng tôi không gạch dưới tên người tham gia, mặc dù UML đã cho phép điều này trong các phiên bản trước đó. TypeName thường là tên của một Lớp hoặc một Kiểu dữ liệu. Khi chúng ta bỏ qua roleName, bất kỳ thành phần nào loại đó có thể được thay thế. Khi chúng ta bỏ qua TypeName, bất kỳ thành phần nào trong vai trò cụ thể, bất kể lớp nào, đều phải đủ. Sau roleName, bạn có thể thấy [selector] trong dấu ngoặc vuông. Nó không phải là một phần của tên của thành phần tham gia, nhưng bộ chọn cho biết phần tử nào của vai trò đa giá trị mà chúng tôi dự định sử dụng.





Trong thực tế, bộ chọn thường là một chỉ số có giá trị nguyên cho một mảng các phần tử của kiểu đã cho. Trong Hình 14.1, chúng ta thấy hai thành phần tham gia (các phần tử có thể kết nối): Người đại diện trong vai trò hiện tại và Hệ thống lưu thông, nếu không thì không được đặt tên. Theo sơ đồ trình tự các phần tử có thể kết nối này “đứng đầu” một lifeline, được mô tả là một đường thẳng đường đứt nét xuống (xem Hình 14.2). Mỗi dòng đại diện cho một thứ tự thời gian chuỗi SỰ KIỆN (được gọi là OCCURRENCES), từ trên xuống dưới , được gửi SENT từ hoặc được nhận NHẬN trên dòng. Hình cho thấy người đại diện (patron) đang gửi một thông báo mayIBorrowBooks tới Hệ thống lưu thông (CirculationSystem). Tiếp theo là CirculationSystem gửi lại thông báo yesYouMay (). Từ quan điểm của mỗi dòng thời gian, việc di chuyển từ trên cùng của biểu đồ xuống dưới cùng thể hiện thứ tự thời gian. Do đó, tại Patron, thông báo yesYouMay được nhận sau khi Đại diện gửi mayIBorrowBooks. Tại Circulation System, thông báo mayIBorrowBooks được nhận trước khi nó gửi thông báo yesYouMay.

Thông thường, chúng ta thấy các thông báo kết thúc bằng dấu ngoặc đơn, ví dụ: yesYouMay(). Ở mức độ lý thuyết, hiếm khi sử dụng dấu ngoặc đơn, nhưng ngược lại, nó là một nơi cho các giá trị đối số. Mức độ lý thuyết cho thấy nỗ lực cung cấp cho người đọc mô hình một khái niệm về những gì đang xảy ra, mà không cố gắng để chính thức đúng hoặc để chỉ định một cách thực hiện chi tiết.

Đầu của lifeline thường là hình chữ nhật; tuy nhiên, các hình dạng khác có thể được sử dụng để chỉ ra loại phần tử có thể kết nối tham gia, tùy thuộc vào trên các khuôn mẫu được sử dụng trong phương pháp luận của bạn (ví dụ: Lớp hoạt động, Lớp thực thể, và Lớp Kiểm soát).

Những đầu của những lifeline tham gia vào Sơ đồ trình tự thường là tất cả căn chỉnh ở đầu sơ đồ. Tuy nhiên, các lifeline có thể được tạo ra trong một Sự tương tác, trong trường hợp đó các đầu được đặt tại điểm tạo. Hàng 6 của Bảng 14.1 cho thấy điều này được thực hiện như thế nào.

Bạn có thể sẽ tìm thấy nhiều ví dụ trong tài liệu về các lifeline do Các chủ thể là đầu. Thật không may, Các chủ thể không thể là các thành phần có thể kết nối và không thể là các phần. Mặt khác, tất cả các công cụ UML sẽ đáp ứng công dụng của các chủ thể như là các lifeline. Cách tiếp cận này thường được giải thích là hợp lý vì thay đổi quan điểm hệ thống. Ví dụ: nếu bạn tập trung vào một hệ thống con, tất cả các hệ thống con khác sẽ là một Chủ thể đối với hệ thống con quan tâm, nhưng tất cả hệ thống con cuối cùng cũng chỉ là một phần của toàn bộ hệ thống.

Nếu bạn muốn sử dụng Chủ thể làm thành phần tham gia Sơ đồ trình tự, đề xuất của tôi, với tư cách là một nhà phương pháp, sẽ tiếp tục nếu điều đó có ý nghĩa trong hoàn cảnh của bạn và công cụ hỗ trợ có sẵn.

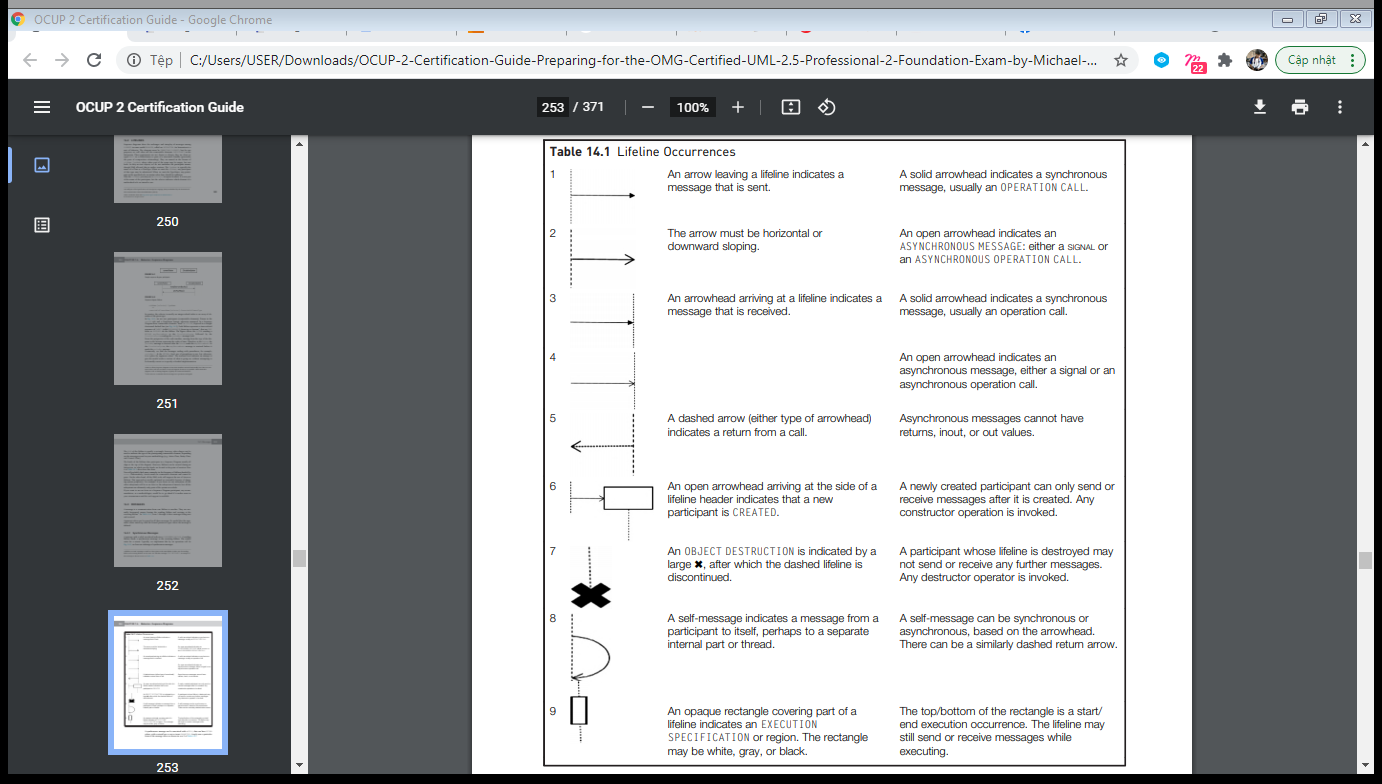
**14.3 Thông báo**

Một thông báo là một thông tin giao tiếp từ lifeline này đến lifeline khác. Chúng thường là những mũi tên nằm ngang rời khỏi lifeline gửi đi và đến lifeline tiếp nhận. Trong Bảng 14.1, các hàng từ 1 đến 4 hiển thị các thông báo được gửi và nhận được.

Giá trị đối số có thể được chuyển trong tất cả các thông báo này. Hãy cẩn thận rằng các giá trị đối số khớp với các loại tham số chính thức nơi thông báo được xác định.

**14.3.1 Thông báo đồng bộ**

Thông báo có đầu mũi tên liền mạch cho biết THÔNG BÁO ĐỒNG BỘ. Một lifeline gửi gửi một thông báo đồng bộ đến lifeline nhận. Người gửi chờ đợi một sự phản hồi. Thông thường, chúng tôi thực hiện điều này bằng một lệnh gọi. Trong Hình 14.2, chúng ta có sự trao đổi các thông điệp đồng bộ.



Một thông báo đồng bộ có thể được liên kết với một TRẢ LỜI, có thể có các giá trị phản hồi, với returnType hoặc out hoặc inout PARAMETERS. Một câu trả lời sử dụng một dạng của mũi tên thông báo như trong hàng 5 của Bảng 14.1

Trong khi người gửi đợi cho đến khi nhận được phản hồi, điều đó không có nghĩa là người gửi nhất thiết phải bị chặn gửi hoặc nhận các thông báo khác. Hành vi không chặn nhưng đồng bộ được phép để giải thích cho khả năng người gửi có thể có nhiều phần hoặc luồng, và chờ đợi có thể chỉ chặn một.

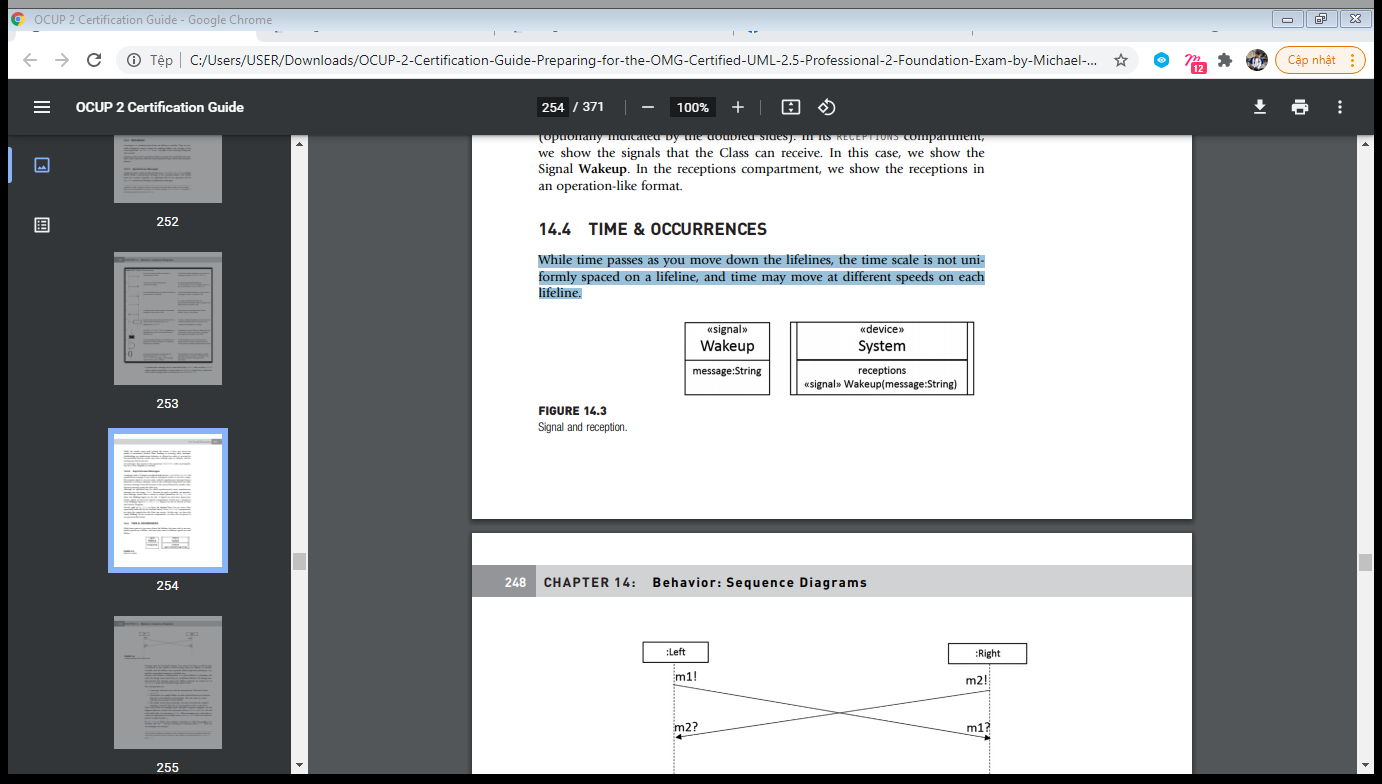
Dưới dạng các hoạt động, chúng xuất hiện trong các ngăn cách hoạt động của lifeline nhận trên Sơ đồ lớp (hoặc tương tự).

**14.3.2 Thông báo không đồng bộ**

Một thông báo có đầu mũi tên hình chữ V cho biết Thông báo không đồng bộ. Một thông báo không đồng bộ được gửi mà không khiến người gửi phải đợi trả lời. Người nhận phải là một lớp đang hoạt động, với thông báo không đồng bộ là phần cứng hoặc phần mềm gián đoạn. Hầu hết các tương tác dựa trên web là các thông báo không đồng bộ từ trình duyệt đến máy chủ, theo sau là một thông báo không đồng bộ khác theo hướng khác.

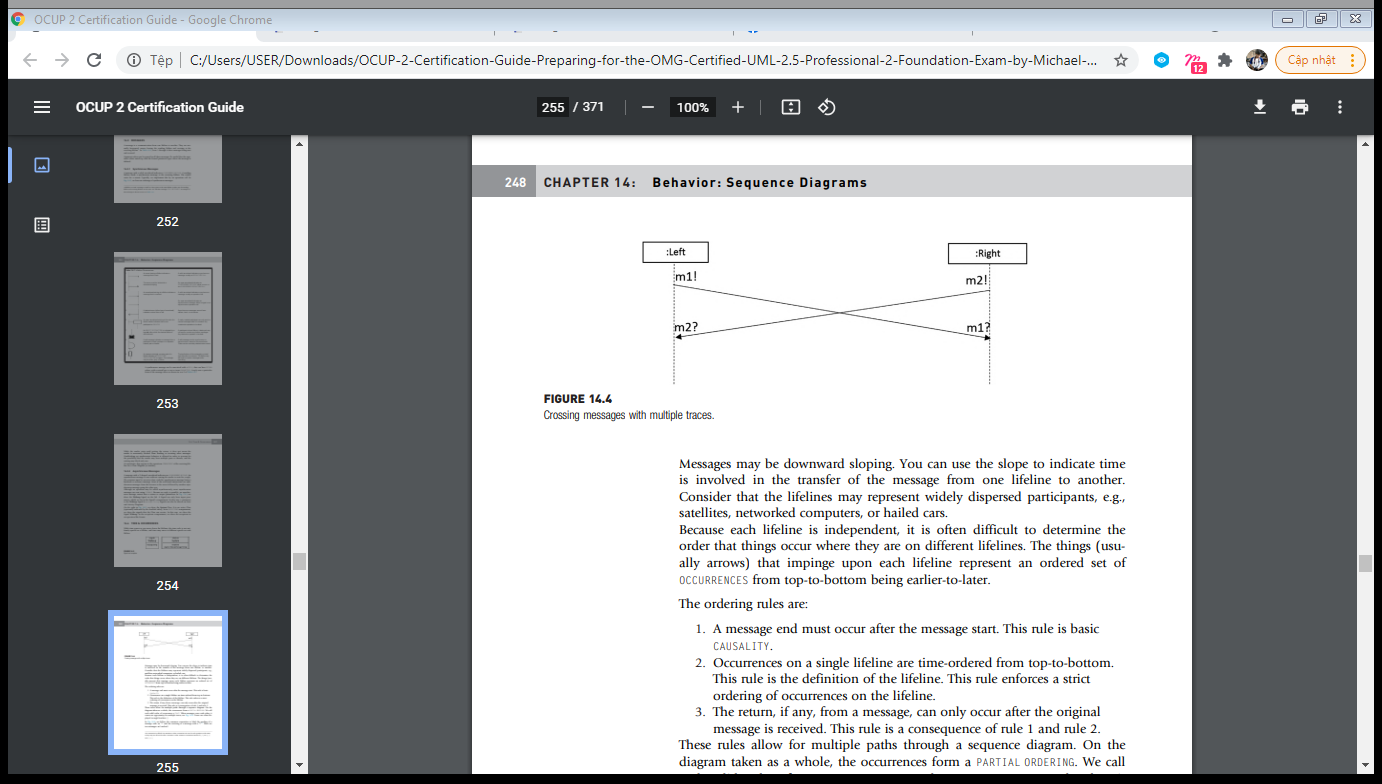
Mặc dù một hoạt động có thể được gọi là không đồng bộ, hầu hết thông báo không đồng bộ được gửi bằng tín hiệu. Bởi vì không thể trả lời, một thông báo không đồng bộ không thể có tham số trả về hoặc đầu ra. Trong hình 14.3, chúng tôi hiển thị Tín hiệu Đánh thức ở bên trái. Một tín hiệu chỉ có thể có các tham số đầu vào mà chúng tôi liệt kê trong ngăn Tín hiệu. Trong trường hợp này, một tham số đến Tín hiệu Đánh thức là message:String. Tín hiệu cũng có thể được hiển thị trên State và Sơ đồ hoạt động.

Ở bên phải trong Hình 14.3, chúng tôi hiển thị Lớp Hệ thống. Đó là một Lớp đang hoạt động (tùy chọn được chỉ định bởi các mặt được nhân đôi). Trong ngăn tiếp tân của nó, chúng tôi hiển thị các tín hiệu mà Lớp có thể nhận được. Trong trường hợp này, chúng tôi hiển thị tín hiệu đánh thức. Trong ngăn tiếp tân, chúng tôi hiển thị các chiêu đãi trong một định dạng giống như hoạt động.



**14.4 Thời gian & tần suất**

Trong khi thời gian trôi qua khi bạn di chuyển xuống các lifeline, thang thời gian không được cách đều nhau trên lifeline và thời gian có thể di chuyển với tốc độ khác nhau trên mỗi lifeline

.

Cân nhắc rằng các lifeline có thể đại diện cho những thành phần tham gia phân tán một cách rộng rãi, ví dụ, vệ tinh, máy tính nối mạng hoặc ô tô được tung hô. Bởi vì mỗi lifeline là độc lập, thường rất khó xác định thứ tự rằng mọi thứ xảy ra ở những nơi chúng ở trên các lifeline khác nhau. Những thứ (thường là mũi tên) tác động lên mỗi lifeline đại diện cho một tập hợp có thứ tự Tần suất từ trên xuống dưới sớm hơn đến muộn hơn.

Các quy tắc yêu cầu là:

1. Kết thúc thông báo phải xảy ra sau khi thông báo bắt đầu. Quy tắc này là NGUYÊN NHÂN cơ bản.

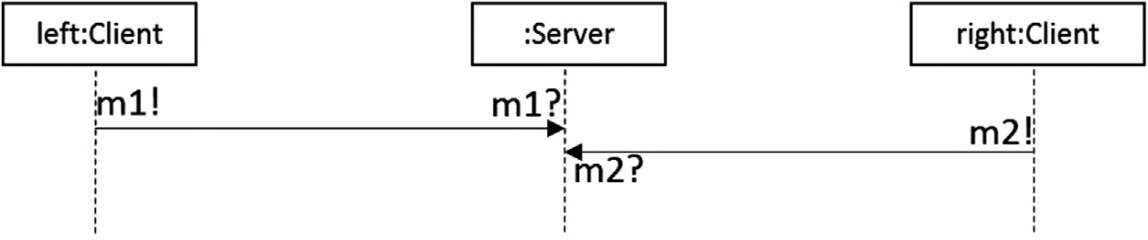
2. Các lần xuất hiện trên một lifeline được sắp xếp theo thứ tự thời gian từ trên xuống dưới.

Quy tắc này là định nghĩa của lifeline. Quy tắc này thực thi nghiêm ngặt thứ tự của các lần xuất hiện trên lifeline.

3. Việc trả lại, nếu có, từ một thông báo, chỉ có thể xảy ra sau thông báo gốc đã được nhận. Quy tắc này là hệ quả của quy tắc 1 và quy tắc 2.

Các quy tắc này cho phép nhiều đường dẫn thông qua một sơ đồ tuần tự. Trên sơ đồ tổng thể, các lần xuất hiện tạo thành một LỆNH BÊN TRONG. Chúng tôi gọi mỗi thứ tự xuất hiện hợp lệ một dấu vết. Khi các thông báo chuyển qua nhau, nó gây cơ hội cho nhiều dấu vết; xem Hình 14.4. Các dấu vết thường được hiển thị trong dấu ngoặc nhọn.

Trong hình 14.4, chúng tôi tuân theo quy ước chung để gắn nhãn việc gửi một thông báo có dấu “!” và nhận được thông báo có dấu “?”. Có hai thông báo: m1 và m2.



**Hình 14.5** *Nhiều dấu vết từ ba lifeline.*

Theo quy tắc 1, chúng ta biết rằng m1! phải xảy ra trước m1? và m2! phải xảy ra trước m2 ?. Ngoài ra, m1! phải xảy ra trước m2? và m2! phải xảy ra trước m1? dựa trên quy tắc 2.

Tuy nhiên, không có quy tắc gói gọn m1! và m2 !. Vì vậy, có hai khả năng lệnh (m1 !, m2! ) hoặc (m2 !, m1! ) . Tương tự, không có quy tắc nào điều chỉnh thứ tự của m2? và m1 ?. Vì vậy, có hai đơn đặt hàng có thể là (m2 ?, m1? ) và (m1 ?, m2?). Bây giờ chúng ta cùng nhau có bốn dấu vết khả dĩ cho chuỗi này biểu đồ.

( m1 !; m2 !; m2 ?; m1? ) ( m1 !; m2 !; m1 ?; m2?)

( m2 !; m1 !; m2 ?; m1? ) ( m2 !; m1 !; m1 ?; m2?)

Không có thông điệp vượt qua, chỉ có một đường kẻ thẳng khả thi khi có chỉ có đường sống . Tuy nhiên, khi có nhiều hơn hai đường sống , bạn có thể cũng nhận được nhiều đường kẻ thẳng . Mặc dù UML cho phép THÔNG ĐIỆP BỊ CHÉO, nhưng nó không cho phép vượt qua hoặc uốn cong đường sống .

Trong hình 14.5, chúng ta có ba đường sống và nhiều đường kẻ thẳng . Hãy xem xét rằng ở đó không có quy tắc nào điều chỉnh thứ tự của m1! và m2 !. Bạn có thể có bao nhiêu đường kẻ thẳng tìm trong hình?

Có nhiều loại lần xuất hiện, nhưng chỉ có một số lượng hạn chế có thể xuất hiện trong kỳ thi OCUP 2 Foundation. Xem Bảng 14.1.

1. Mũi tên liền khối có điền đầy (đặc) là việc gửi một thông điệp đồng bộ. Nếu đây là một sơ đồ trình tự , thì nó chỉ có nghĩa là người gửi được đảm bảo rằng người nhận nhận được

Vì các quy tắc bao gồm tất cả các thử thách có thể có khác ngoại trừ thứ tự của m1! và m2 !, chỉ có

hai đường kẻ thẳng . (m1 !, m1 ?, m2 !, m2? ) ( m2 !, m1,! m1 ?, m2? )

Lưu ý rằng chúng tôi chưa xác định mất bao lâu để thông điệp được chuyển từ khách hàng đến người phục vụ. Đây là nguồn gốc cơ bản của sự không chắc chắn trong các đường kẻ thẳng . Xin nhắc lại, chúng tôi xây dựng một sơ đồ danh nghĩa khi chúng tôi dự định truyền đạt một khái niệm (một ý tưởng) về bức tranh chung về tình hình và hành vi. Nó được thiết kế để chính xác về mặt pháp lý, nhưng nó thường sẽ loại bỏ các chỉ số tùy chọn về chi tiết thực hiện.

thông điệp. Nếu không, điều đó có nghĩa là người gửi đang yêu cầu một hoạt động về người nhận và người gửi sẽ đợi phản hồi. Việc lập sơ đồ phản hồi là tùy chọn, nhưng nếu nó xuất hiện, nó sẽ giống như ví dụ ở hàng 5. Tên thông điệp được đặt phía trên mũi tên và tuân theo cú pháp của một phép toán như đã định nghĩa trước đó. Người được gọi hoạt động phải là thành viên của OPERATIONS của người nhận hoặc được kế thừa từ lớp cha của người nhận. Các thông số của hoạt động có thể được cung cấp các giá trị phù hợp với tình huống.

2. Mũi tên liền khối có đầu mũi tên hình chữ V là sự gửi đi của một thông báo không đồng bộ. Nếu đây là một biểu đồ trình tự , thì nó chỉ có nghĩa là người gửi không đợi phản hồi. Lớp người nhận phải là một lớp hoạt động, mặc dù thực tế đó không phải lúc nào cũng được mô hình hóa trong bất cứ cách nào .

Các thông báo không đồng bộ này thường là TÍN HIỆU và chúng phải là thành viên TIẾP NHẬN của người nhận hoặc được thừa kế từ lớp cha của người nhận. Các tín hiệu có cùng một cú pháp như nhưng không có giá trị TRẢ VỀ hoặc tham số RA hoặc VÀO, như chúng yêu cầu phản hồi từ người nhận. Các thông báo không đồng bộ này cũng có thể là các hoạt động được gọi không đồng bộ.

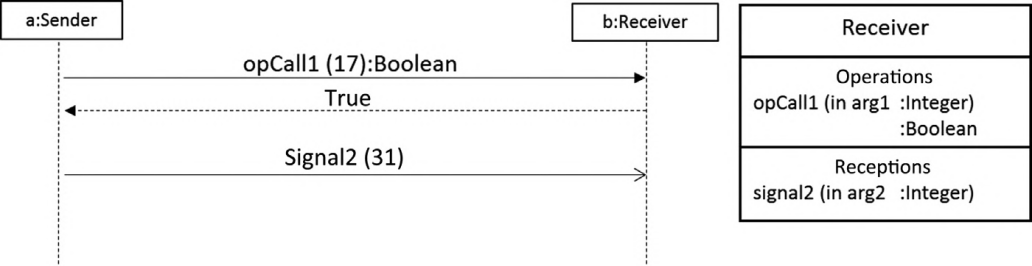
3. Nhận được một đầu mũi tên rắn có nghĩa là một hoạt động đã được gọi trên người nhận. Nếu một hoạt động có tên đó không được xác định trên người nhận, các lớp cha được tìm kiếm một định nghĩa. Nếu không có định nghĩa nếu được tìm thấy sau đó, hành vi là không xác định.

4. Nhận được một đầu mũi tên hình chữ V cho biết một tín hiệu đã được người nhận nhận được không đồng bộ. Nếu một sự tiếp nhận của tên đó không được xác định trên người tham gia, các lớp cha là đã tìm kiếm chuỗi kế thừa để tìm định nghĩa. Nếu không có định nghĩa nếu được tìm thấy, thì hành vi là không xác định. Người nhận phải thuộc loại ACTIVE và không được trả lại.

5. Một mũi tên nét đứt biểu thị sự trở lại từ một lệnh gọi hoạt động. Trong mô hình không chính thức, đây có thể là một lợi nhuận không tưởng, nhưng khi mô hình hóa chính thức, nó phải khớp với kiểu trả về của lệnh gọi hoạt động. Nó cũng có thể sử dụng định dạng phức tạp hơn cho biết lệnh gọi hoạt động, mặc dù điều này định dạng sẽ không có trong kỳ thi OCUP2 Cấp độ 1.

Trong Hình 14.6, chúng tôi hiển thị các ví dụ của Bảng 14.1 hàng 1..5, tức là gửi và nhận các thông điệp đồng bộ và không đồng bộ và một thông điệp trả về từ một lệnh gọi hoạt động.

6. Một mũi tên với đầu mũi tên hình chữ V mở đến hộp tiêu đề cho biết người tham gia đã được tạo, như được mô tả trong Hình 14.7. Tên này của thông báo có thể là tên của hàm tạo với các tham số điền vào, hoặc nó có thể chỉ là hư cấu. Nếu nó không phải là một cuộc gọi rõ ràng đến



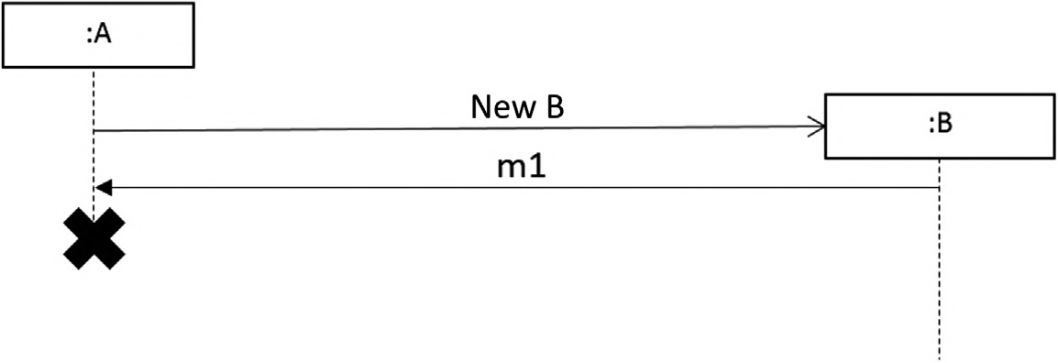
**Hình 14.6** Biểu đồ trình tự mẫu và lớp người nhận



**Hình 14.7** Tạo mới đường

phương thức khởi tạo, phương thức khởi tạo được gọi. Sau khi TẠO ĐỐI TƯỢNG, người tham gia bây giờ có thể tham gia bình đẳng với các lifeline đã tồn tại trước đó. Tất cả các lifeline không nằm ở đầu sơ đồ phải được tạo trước khi họ có thể tham gia. Trong hình ví dụ, Đường dây nóng A tạo ra đường lifeline B. Sau khi B được tạo, nó đồng bộ gửi thông điệp M1 đến A (hoặc gọi hoạt động M1 trên A). Sau đó, A gửi đồng bộ thông điệp M2 đến B (hoặc gọi Hoạt động của M2 trên B).

7. Dấu ✖ lớn ở cuối đường sống cho biết đường sống là bị hủy và không thể tiếp tục tham gia thêm hành động nào . Các đường sống không được kéo dài thấp hơn ✖. Một đường sống có thể bị hủy trong phản hồi một thông điệp đến hoặc nó có thể quyết định tự hủy. Trong Hình 14.8, chúng tôi mô tả sự THẤT BẠI của đường sống A trông giống như được kích hoạt bởi m1. Bất kỳ đường sống nào bắt đầu ở đầu Sơ đồ trình tự phải là tồn tại trước tương tác . Bất kỳ đường sôgngs nào không đạt đến dưới cùng của Sơ đồ trình tự không tồn tại sau tương tác.



**Hình 14.8** Tạo và phá hủy đường cứu sinh.

Trong hình, đường sống A tồn tại từ trước nhưng bị phá hủy vào cuối Tương tác, trong khi huyết mạch B được tạo và vẫn có sẵn sau tương tác đã hoàn tất.

8. Ký hiệu này tự chỉ ra một thông điệp . Nó tuân theo tất cả các quy tắc giống nhau quản lý các thông điệp, nhưng nó là từ một thời điểm trên một đường sống khác thời gian (một thời gian sau) trên cùng một đường sống .

9. Một hộp mờ đục đặt trên đường sống cho biết một Thông Số Kỹ Thuật Thực Thi , nghĩa là, đường sống đang thực hiện một số hành vi hoặc chờ một hành vi cấp dưới kết thúc. Như một đường sống có thể chứa nhiều luồng thực thi, một đường sống vẫn có thể gửi hoặc nhận thông báo bổ sung trong khi được bao phủ bởi một đặc tả thực thi.Ngoài ra, có thể có nhiều thông số kỹ thuật thực thi chồng chéo.Một đặc tả thực thi thường được gọi là KÍCH HOẠT.

**NHỮNG ĐIỂM CẦN NHỚ**

● Sơ đồ trình tự cho thấy cách các phần tử tham gia trao đổi thông điệp theo thời gian trong một phần tử.

● Phần tử tham gia phải là phần tử có thể kết nối.

● Không có đối tượng (phiên bản) nào được cho phép bởi Đặc tả UML 2.5.

● Không cho phép Actors (mặc dù thường được sử dụng trong thực tế).

● Các yếu tố tham gia được hiển thị dưới dạng dây sống.

● Đường sống được hiển thị dưới dạng đường đứt nét dọc dưới tiêu đề.

● Thường thì tiêu đề là hình hộp chữ nhật, mặc dù nó có thể có hình dạng khác nếu có

khuôn mẫu liên quan.

● Tên trong tiêu đề có định dạng roleName: TypeName

● Có hai loại thông báo chính: Đồng bộ và Không đồng bộ.

● Tin nhắn đồng bộ là các cuộc gọi thao tác và có thể trả về kết quả cho người gọi.

● Tin nhắn đồng bộ được hiển thị bằng đầu mũi tên liền mạch.

● Các hoạt động có thể được hiển thị trong ngăn hoạt động của điểm đến

người tham gia.

● Thông báo không đồng bộ không cho phép các đối số trả về, ra hoặc bỏ đi

Thông báo không đồng bộ được hiển thị với đầu mũi tên hình chữ V

● Một tin nhắn không đồng bộ có thể là một cuộc gọi hoạt động không đồng bộ và được hiển thị với các hoạt động khác trong ngăn hoạt động của người tham gia đích.

● Tin nhắn không đồng bộ có thể là một tín hiệu. Một tín hiệu có thể được hiển thị trong các cuộc chiêu đãi ngăn của người tham gia đích.

● Có thể tạo hoặc phá hủy đường dây nóng.

● Một đường cứu sinh đã tạo hiển thị một mũi tên hình chữ V nối với mặt bên của tiêu đề.

● Lifeline bị phá hủy kết thúc bằng dấu ✖ lớn.

● Tin nhắn đi từ người gửi đến người nhận và có thể nằm ngang hoặc dốc xuống.

● Một thông điệp tự có thể đi từ huyết mạch đến chính nó, miễn là nó đi đến điểm muộn hơn (thấp hơn).

● Sự tự quay trở lại cũng có thể đi từ huyết mạch đến chính nó sau khi xuất hiện thông điệp bản thân.

● Sự kiện xảy ra là bất kỳ điểm nào trên đường cứu sinh, bao gồm:

● Một tin nhắn đang được gửi đi

● Một tin nhắn đang được nhận

● Gửi trả lại

● Đã nhận được lợi nhuận

● Bắt đầu một thông số kỹ thuật thực thi

● Kết thúc một đặc tả thực thi

● Gửi một lệnh tạo (mới) để tạo một huyết mạch mới

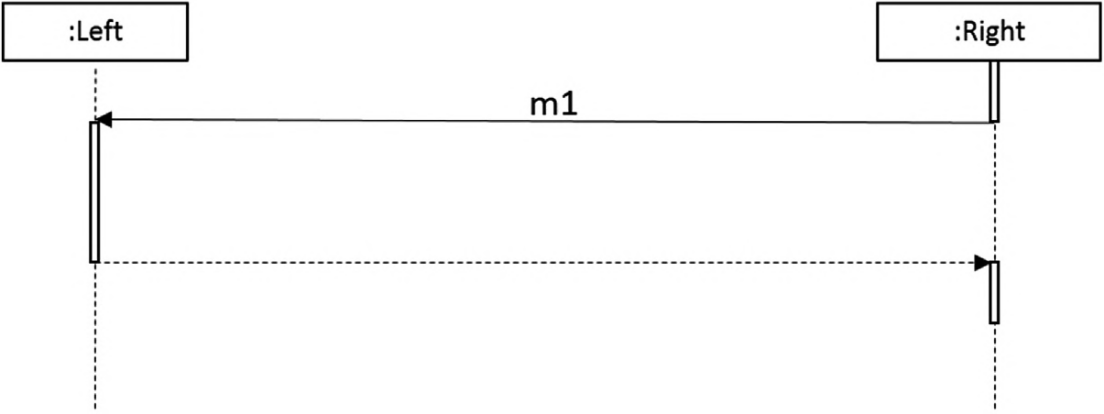
● Lifeline mới đang được tạo

**14.5 THÔNG SỐ KỸ THUẬT THỰC THI**

Thông số kỹ thuật thực thi là một hình chữ nhật đứng, mờ đục và phủ lên một phần của đường sống để cho biết rằng đường sống đang bận thực hiện một số hành vi. Thông số kỹ thuật thực thi là tùy chọn và thường được sử dụng trong các sơ đồ phức tạp nơi ngăn xếp thực thi hữu ích để xem hoặc nơi kết thúc của một số hành vi được quan tâm. Tuy nhiên, nếu bạn sử dụng nó trên bất kỳ đường sống nào trong Sơ đồ trình tự, bạn nên sử dụng nó trên tất cả các đường sống trong sơ đồ. Bạn có thể đặt tên là Thông số kỹ thuật thực thi. Để không làm lộn xộn sơ đồ, các tên thường được dành riêng cho những trường hợp mà tên của hành vi đang thực thi không khớp với tên của thông báo.

Trong Hình 14.9, Đường sống bên phải bắt đầu bận rộn; sau đó nó sẽ gửi thông điệp m1 một cách đồng bộ đến dòng thời gian Left, dòng thời gian này trở nên bận thực hiện phản hồi thông báo. Trong khi đó, dòng thời gian Bên phải sẽ tạm dừng cho đến khi quá trình thực hiện kết thúc và dòng thời gian Bên trái trở về bên phải. Sau đó, dòng thời gian Bên phải tiếp tục thực hiện trước đó của nó.

Như trong tất cả các Sơ đồ trình tự, việc vẽ trả về một cách rõ ràng là tùy chọn, đặc biệt là nếu nó chỉ chuyển giao quyền kiểm soát trở lại người gọi. Trong hình trên, nếu chúng tôi giải quyết được việc trả lại, bạn vẫn có thể xác định thời điểm nó xảy ra.



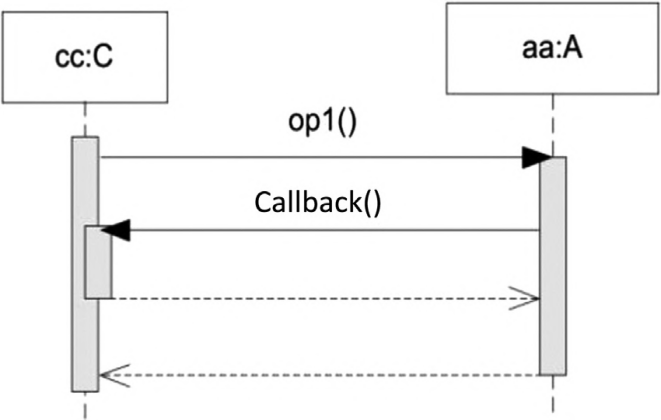
**Hình 14.9** Đặc tả thực thi.

Bạn nhìn thấy bao nhiêu lần xuất hiện trên sơ đồ này? Trước khi bạn đếm, hãy nhớ rằng hai lần xuất hiện ở cùng một vị trí trên một đường sống vẫn là những lần xuất hiện riêng biệt.

Có một số cách tiếp cận để sử dụng Đặc tả Thực thi. Trong hình 14.9, chúng tôi cho thấy đường sống Bên phải không bận bằng cách bỏ qua hình chữ nhật trong khi Bên trái bận thay cho Bên phải. Tuy nhiên, bạn có thể thấy các mô hình hiển thị cả hai đường sống là bận rộn trong thời gian này, vì Bên phải bận rộn, bận rộn, bận rộn chờ đợi. Chúng tôi đưa ra một ví dụ về cách tiếp cận này dưới đây.

Trong hình 14.10, đường sống C đang bận ngay cả khi đang đợi đường sống A quay lại. Trong quá trình A thực hiện, nó gọi một lệnh gọi lại trên C, giải thích đặc điểm thực thi chồng chéo trên C. Khi lệnh gọi trở lại A, quá trình thực thi đó kết thúc.

Sau đó, quá trình thực thi của A kết thúc bằng việc quay trở lại Lifeline C.



**Hình 15.10** Các thông số kỹ thuật thực thi chồng chéo. Đặc điểm kỹ thuật UML 2.5 Hình 17.2.

**14.5 THÔNG SỐ KĨ THUẬT THỰC THI**

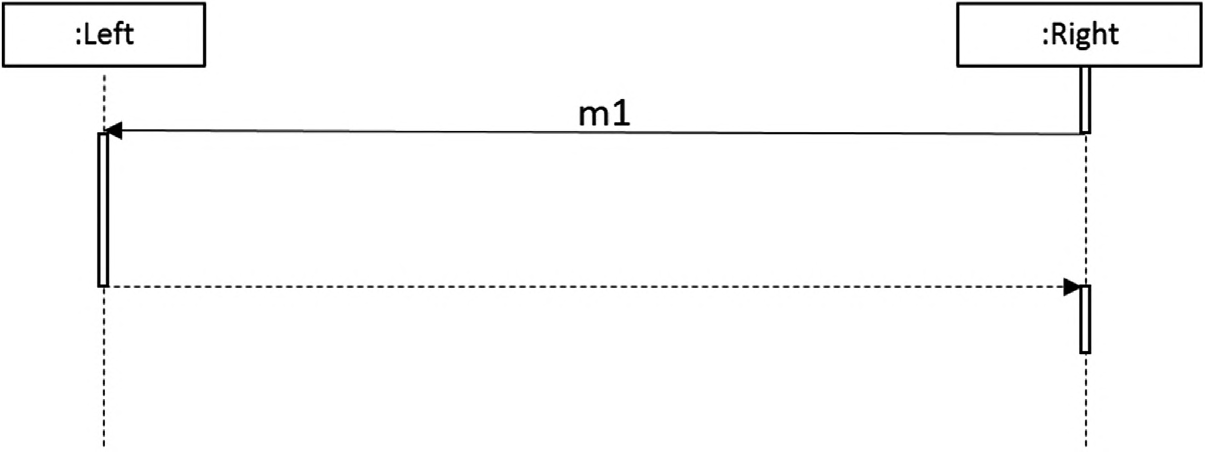
Thông số kỹ thuật thực thi là một hình chữ nhật đứng, mờ đục và có các lớp phủ trên một phần của ĐƯỜNG SỐNG để chỉ ra rằng ĐƯỜNG SỐNG đang bận thực hiện một số hành vi.   
Thông số kỹ thuật thực thi là tùy chọn và thường được sử dụng sơ đồ phức tạp nơi ngăn xếp thực thi hữu íchđể xem hoặc nơi kết thúc của một số hành vi được quan tâm. Tuy nhiên, nếu bạn sử dụng nó trên bất kỳ lifeline nào trong một Trình tự Sơ đồ, bạn nên sử dụng nó trên tất cả các lifeline trong sơ đồ.

Bạn có thể đặt tên là Thông số kỹ thuật thực thi. Để không làm lộn xộn sơ đồ, tên thường được dành riêng cho những trường hợp mà tên của thực thi hành vi không khớp với tên của thông báo.

Trong Hình 14.9, Đường Sống bên phải bắt đầu bận rộn; sau đó nó gửi tin nhắn m1 đồng bộ với dòng thời gian Bên trái, dòng thời gian này trở nên bận rộn thực thi trong trả lời tin nhắn. Trong khi đó, dòng thời gian Bên phải sẽ tạm ngừng cho đến khi quá trình thực hiện kết thúc và đường sống Bên trái trở lại Bên phải. Sau đó, dòng thời gian Bên phải tiếp tục thực hiện việc trước đó của nó.

Như trong tất cả các Sơ đồ trình tự, việc vẽ trả về một cách rõ ràng là tùy chọn, đặc biệt là nếu nó chỉ chuyển giao quyền kiểm soát trở lại người gọi. Ở trên hình, nếu chúng tôi giải quyết được việc trả lại, bạn vẫn có thể xác định khi nào xảy ra.

**HÌNH 14,9 Đặc tả thực thi.**



Bạn nhìn thấy bao nhiêu lần xuất hiện trên sơ đồ này? Trước khi bạn đếm, hãy nhớ rằng hai lần xuất hiện ở cùng một vị trí trên lifeline vẫn còn các lần xuất hiện riêng biệt.

Có một số cách tiếp cận để sử dụng Đặc tả Thực thi. Trong hình 14.9, chúng tôi cho thấy rằng Đường sống bên phải không bận bằng cách bỏ qua hình chữ nhật trong khi Bên trái bận thay cho Bên phải. Tuy nhiên, bạn có thể thấy các mô hình cho thấy cả hai đường là bận rộn trong thời gian này, vì Bên phải đang bận, rằng là: bận chờ đợi. Chúng tôi đưa ra một ví dụ về cách tiếp cận này bên dưới

Trong hình 14.10, Đường dây nóng C đang bận ngay cả khi đang chờ Đường dây A quay trở lại. Trong quá trình thực thi của A, nó sẽ gọi lại trên C, giải thích sự chồng chéo đặc tả thực thi trên C. Khi gọi lại trở về A, việc thực thi đó kết thúc.

Sau đó, quá trình thực thi của A kết thúc bằng việc quay trở lại Lifeline C.

**NHỮNG ĐIỂM CẦN NHỚ**

● Thông số kỹ thuật thực thi chỉ ra rằng một số hành vi đang thực thi trong lifeline.

● Đó là một hình chữ nhật mờ đục che khuất một phần của lifeline, có điểm bắt đầu cho biết hành vi bắt đầu.

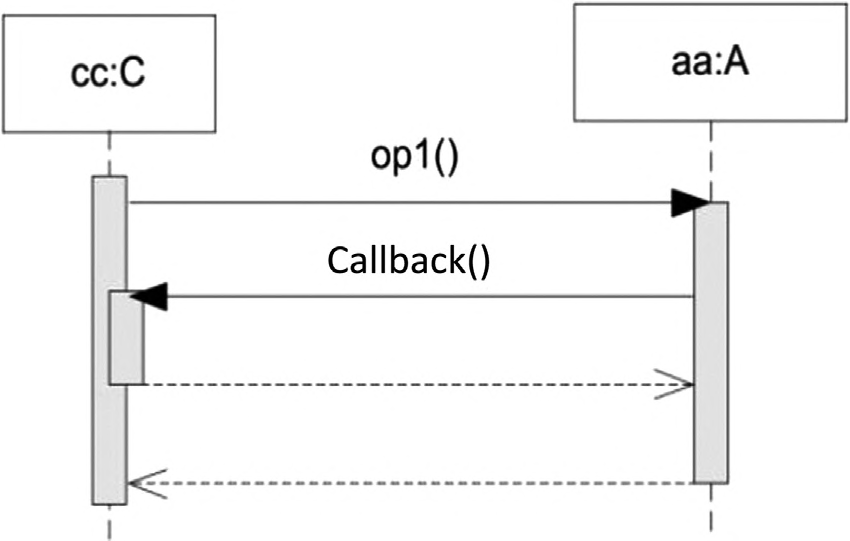
● Điểm kết thúc cho biết sự kết thúc của hành vi.

● Chờ một lifeline khác kết thúc có thể được hiển thị dưới dạng một khoảng trống trong hình chữ nhật.

● Lifeline có thể có các thông số kỹ thuật thực thi chồng chéo nếu lifeline nhận được một cuộc gọi khác trong khi thực hiện

**HÌNH 14.10 Các thông số kỹ thuật thực thi chồng chéo.**

***Đặc điểm kỹ thuật UML 2.5 Hình 17.2***



**14.6 Biểu đồ trình tự**

Biểu đồ trình tự cho thấy sự trao đổi và tác động lẫn nhau của các thông điệp giữa các những người tham gia vào một số hành vi hữu ích. Chúng tôi gọi nội dung này là TƯƠNG TÁC. Tương tác là một đơn vị của HÀNH VI tập trung vào việc trao đổi thông tin giữa CÁC YẾU TỐ CÓ THỂ KẾT NỐI. Như một Hành vi, một Tương tác là một loại Lớp và do đó, là một loại NAMESPACE

Biểu đồ trình tự có cùng một loại khung và tiêu đề cho tất cả UML sơ đồ có, một tiêu đề hình ngũ giác với loại không gian tên sơ đồ trong in đậm (sd) theo sau là tên không gian tên. Trong Sơ đồ trình tự, tên không gian tên là tên của Tương tác Xem Hình 14.11.

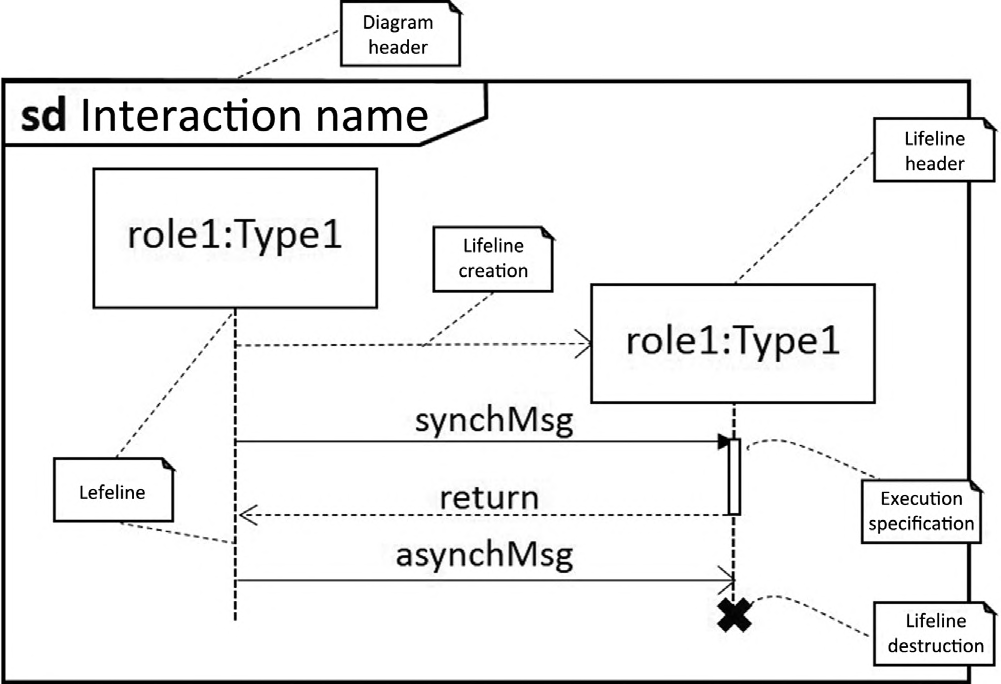
Vì lý do lịch sử, mặc dù loại không gian tên là một Tương tác trong tiêu đề sơ đồ loại là sd. Sd này cũng là loại được sử dụng cho thời gian Sơ đồ, Sơ đồ giao tiếp và Sơ đồ Tổng quan về Tương tác.

Các sơ đồ này cũng được coi là Sơ đồ tương tác, nhưng chúng sẽ không đang trong kỳ thi Foundation.

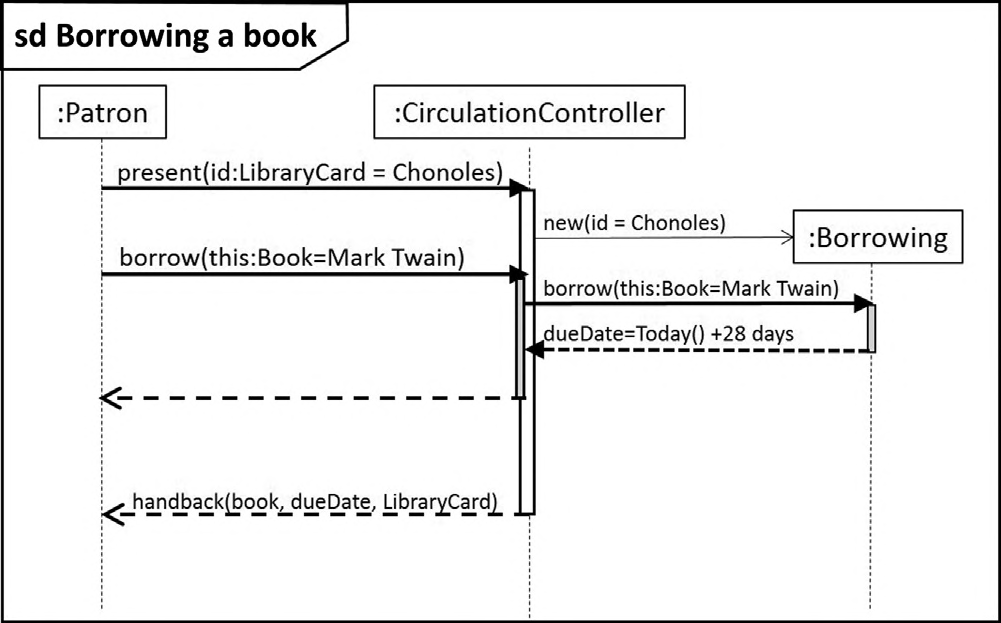
Trong một mô hình, nếu hai Sơ đồ tương tác có cùng tên, chúng xem vào cùng một không gian tên Tương tác. Một số công cụ UML sẽ thực thi đúng điều này và cho phép các sơ đồ chia sẻ các phần tử. Các công cụ khác sẽ chỉ coi các sơ đồ được đặt tên giống nhau là độc lập. Những người khác sẽ ngăn chặn sự trùng lặp.

Hộp ranh giới sơ đồ có khả năng quan trọng vì nó có ý nghĩa khi thông điệp đến từ hoặc đi đến ranh giới. Trong kỳ thi đầu tiên, chúng tôi sẽ không bao gồm các tính năng này.

***HÌNH 14.11 Các phần chính của sơ đồ tuần tự.***



***HÌNH 14.12 Mượn kịch bản sách.***



**14.7 SƠ ĐỒ PHÂN TÍCH THỰC TIỄN**

Trong chương này, chúng tôi đã đề cập đến các tính năng của Sơ đồ trình tự là được bao phủ bởi kỳ thi OCUP 2 đầu tiên. Sử dụng các tính năng này, chúng ta có thể vẽ đơn giản đường dẫn một lần mà không cần lặp lại. Trong khi có thể có nhiều dấu vết vì được mô tả trong chương này, chúng tôi vẫn thấy rằng tất cả các thông báo mô hình hóa phải được gửi và nhận mà không thất bại.

Đặc tả Sơ đồ trình tự hoàn chỉnh mạnh hơn nhiều. Nó bao gồm một tập hợp cấu trúc điều khiển mạnh mẽ: tương đương với tùy chọn, if then-else, alt, vòng lặp, if, break, sinh sản song song, vùng quan trọng và hỗ trợ cho hành vi và thành phần cấu trúc.

Ngay cả khi không có cấu trúc điều khiển bổ sung, Sơ đồ trình tự có thể minh họa cách một kịch bản hành vi hoạt động với các giá trị đối số cụ thể được hỗ trợ cho các tham số chính thức.

Trong Hình 14.12, chúng ta thấy một phiên bản đơn giản của kịch bản Mượn sách. Đối với một số tham số chính thức, chúng tôi đã cung cấp các giá trị đối số cụ thể choKịch bản. Ở đây, chúng ta thấy Patron Chonoles hiển thị thẻ LibraryCard của mình và sau đó mượn Sách Mark Twain. Ngày đến hạn được tính toán, và một bản ghi hàng ngang được tạo. Sau đó là Thẻ thư viện, Sách và Ngày đến hạn đều được trao lại cho Người bảo trợ.

Trong chương này, chúng tôi đã trình bày những kiến thức cơ bản về UML 2.5 rất mạnh mẽ Sơ đồ trình tự. Bây giờ bạn có thể vượt qua các câu hỏi trong chương tiếp theo và làm tốt phần Tương tác trong kỳ thi OCUP 2.