



UNIFIED MODELING LANGUAGE™



WE SET THE STANDARD™

8.

Focusing on Interaction Links: Communication Diagrams

Shaoning Zeng, <http://zsn.cc>

What we learnt?

- ▶ 6. Bringing Your Classes to Life: Object Diagrams
 - ▶ 6.1. **Object** Instances
 - ▶ 6.2. Links
 - ▶ 6.3. Binding Class Templates
- ▶ 7. Modeling Ordered Interactions: Sequence Diagrams
 - ▶ 7.1. **Participants** in a Sequence Diagram
 - ▶ 7.2. **Time**
 - ▶ 7.3. Events, Signals, and **Messages**
 - ▶ 7.4. **Activation** Bars
 - ▶ 7.5. Nested Messages
 - ▶ 7.6. Message **Arrows**
 - ▶ 7.7. Bringing a Use Case to Life with a Sequence Diagram
 - ▶ 7.8. Managing Complex Interactions with Sequence Fragments



8. Focusing on Interaction Links: **Communication** Diagrams 会话图

- ▶ 8.1. Participants, Links, and Messages
- ▶ 8.2. Fleshing out an **Interaction** with a Communication Diagram
- ▶ 8.3. Communication Diagrams Versus Sequence Diagrams



8.1. Participants, Links, and Messages

- ▶ A communication diagram is made up of three things:
 - ▶ **participants**,
 - ▶ the **communication links** between those participants,
 - ▶ and the **messages** that can be passed along those communication links

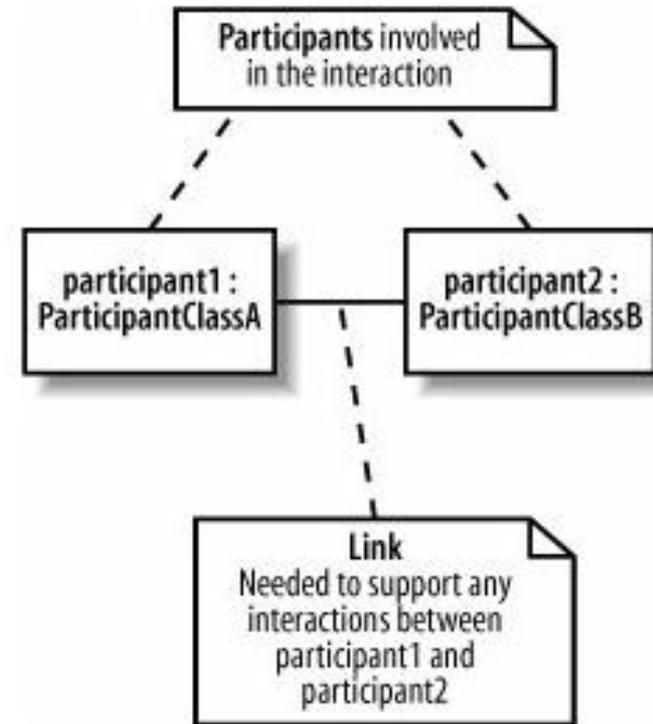


Figure 8-1. Much simpler than sequence diagrams, communication diagrams are made up of participants and links



8.1. Participants, Links, and Messages

- ▶ **Participants** on a communication diagram are represented by a **rectangle**. 矩形
 - ▶ The participant's **name** and **class** are then placed in the middle of the rectangle.
 - ▶ A participant's name is formatted as **<name> : <class>**, similar to participants on a sequence diagram. You need to specify either the participant's name or class (**or both**).
 - ▶ If, for some reason, you do not have both the name and class information, sometimes a participant is anonymous and does not have a name, then either the class or the name **can be left out**.
- ▶ A **communication link** is shown with a single line that connects two participants. 参与者之间的联系 (消息)
 - ▶ A link's purpose is to allow **messages** to be passed between the different participants; without a link, the two participants cannot interact with each other.



Figure 8-2. Two **messages** are passed along the link between **participant1** and **participant2**

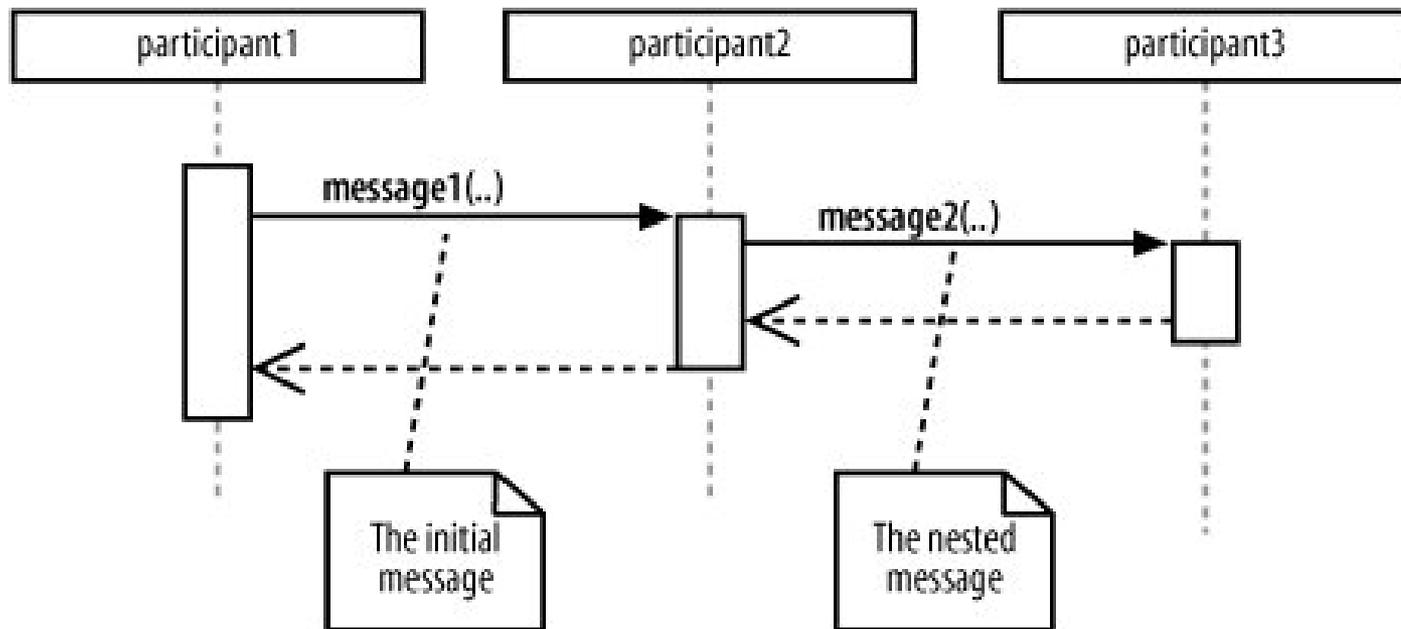


8.1. Participants, Links, and Messages

- ▶ A message on a communication diagram is shown using a filled arrow from the message sender to the message receiver.
 - ▶ **Similar** to messages on a sequence diagram, a message's signature is made up of the message **name** and a list of **parameters**.
[消息名+参数]
 - ▶ However, **unlike** sequence diagrams, the message **signature** alone is not enough for a communication diagram, you also need to show the **order** in which the messages are invoked during the interaction.
[顺序]
- ▶ Communication diagrams do not necessarily flow down the page like sequence diagrams; therefore, message order on a communication diagram is shown **using a number** before each message. [使用数字表示顺序]
- ▶ Things get more **complicated** when a message sent to a participant directly causes that participant to **invoke another message**.



Figure 8-3. **Nested messages** on sequence diagrams are easy to see; when the initial message, message1(..), is invoked on participant2, participant2 then invokes the nestedmessage2(..) on participant3



8.1.1. Messages Occurring at the Same Time

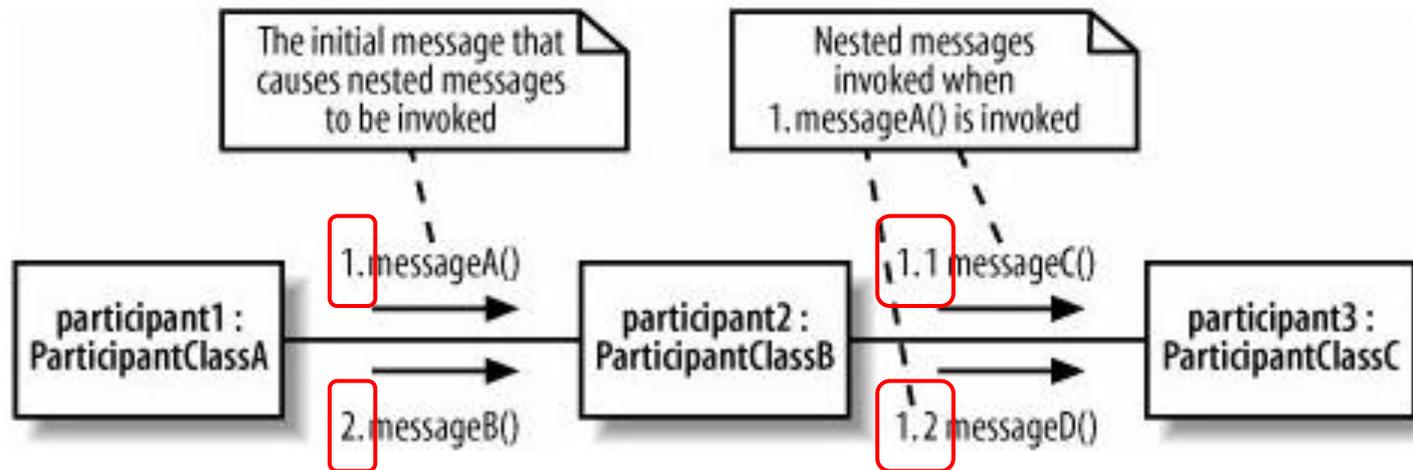


Figure 8-4. messageA() directly leads to nested 1.1 messageC(), followed by nested messageD(), before message 2 is invoked



8.1.1. Messages Occurring at the Same Time

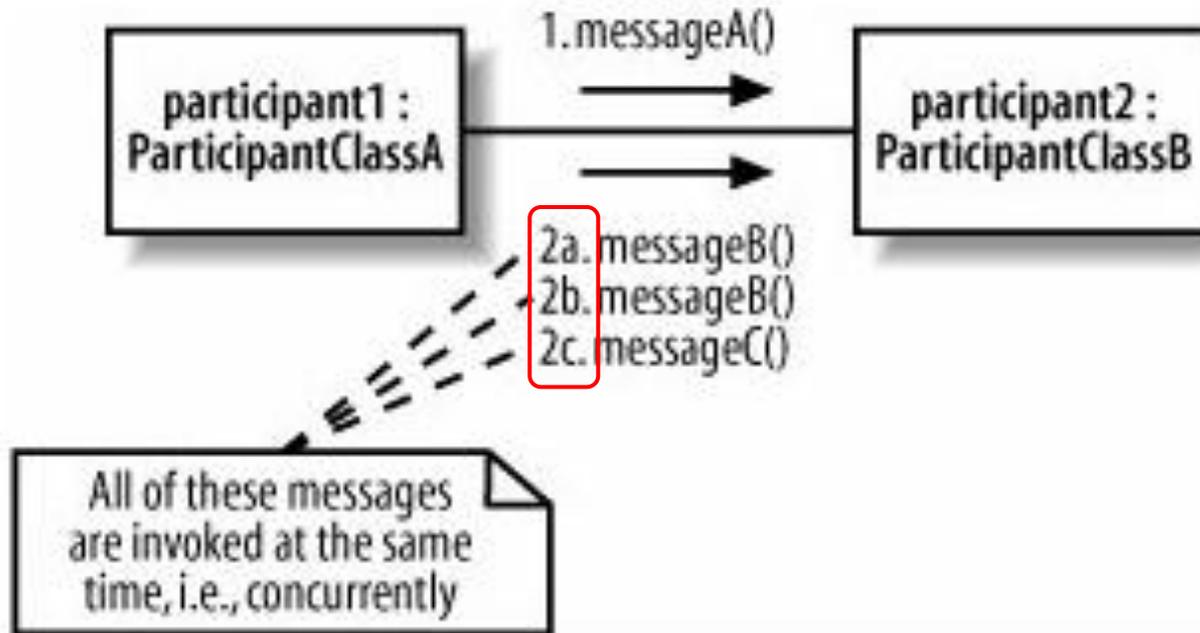


Figure 8-5. Messages 2a. messageB(), 2b. messageB(), and 2c. messageC() are all invoked at the same time after 1. messageA() has been invoked



8.1.2. Invoking a Message Multiple Times

- ▶ This is similar to showing that your messages will be invoked in a **for(..) loop** if you are mapping your communication diagram's participants to software.
- ▶ Although UML does not actually dictate how a communication diagram can show that a message is invoked a number of times, it does state that an **asterisk** should be used before a looping constraint is applied. 星号
 - ▶ This rather complicated statement means that the following example is a safe way to specify that something is going to happen 10 times:
 - ▶ `*[i = 0 .. 9]`
 - ▶ In the above looping constraint, *i* represents a counter that will count up from 0 to 9, doing whatever task is associated with it 10 times.



8.1.3. Sending a Message Based on a Condition

- ▶ Sometimes a message should be invoked **only if a particular condition** is evaluated to be true. 条件
 - ▶ For example, your system might have a message that should be invoked only if the previous message has executed correctly.
 - ▶ Just as with sequence diagram fragments, communication diagram messages can have **guards** set to describe the conditions that need to be evaluated before a message is invoked.
- ▶ A **guard condition** is made up of a logical Boolean statement. 守护条件
 - ▶ When the guard condition evaluates to true, the associated message will be invoked, otherwise, the message is skipped.



Figure 8-6. The addition of a new looping constraint to 1. messageA() means that the message will be invoked 10 times before the next set of messages 2a, 2b, and 2c can be invoked

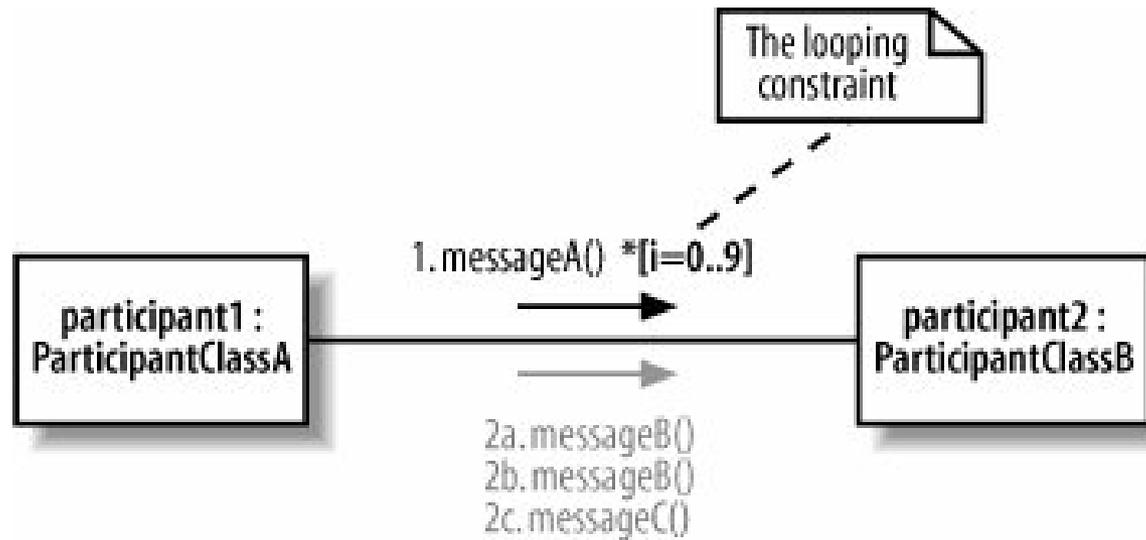
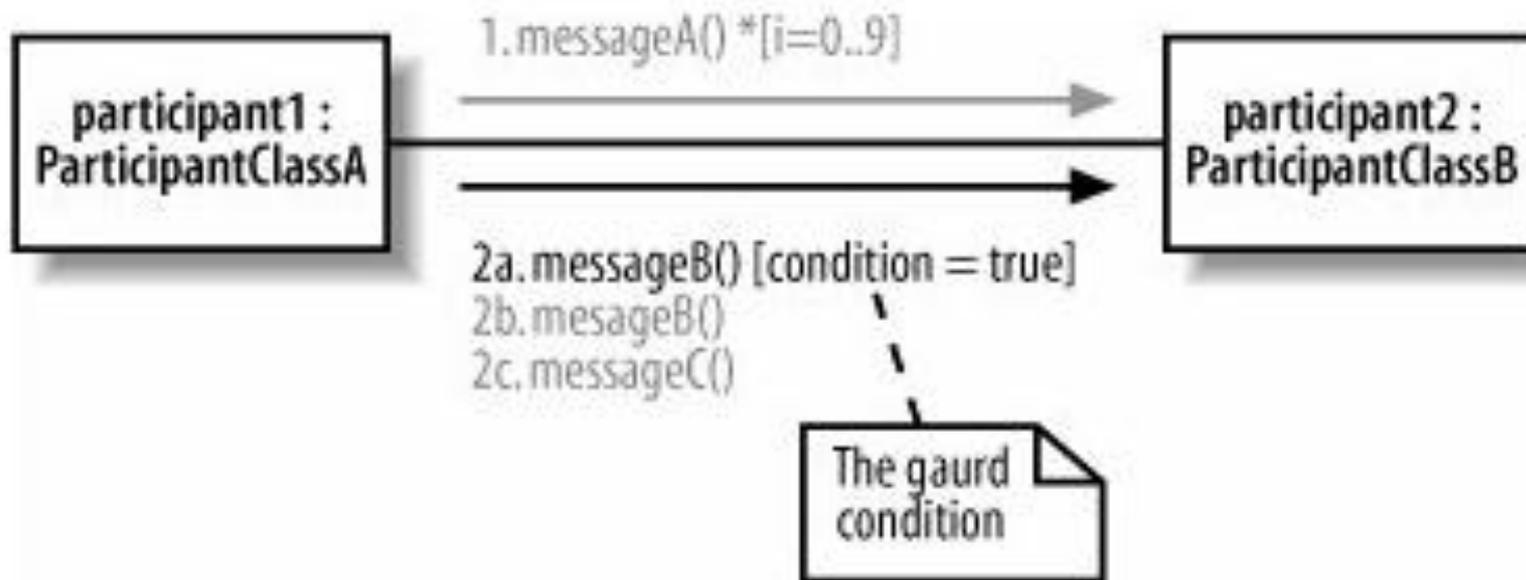


Figure 8-7. a. messageB() will be invoked only at the same time as 2b. messageB() and messageC() if the expression condition == true is evaluated as true; if condition == true returns false, then 2a. messageB() is not invoked, but message 2b. messageB() and 2c. messageC() are



8.1.4. When a Participant Sends a Message to Itself

- ▶ A participant talking to itself may sound strange at first, but if you think in terms of **a software object making a call to one of its own methods**, you might start to see why this form of communication is needed (and even common). 调用自己

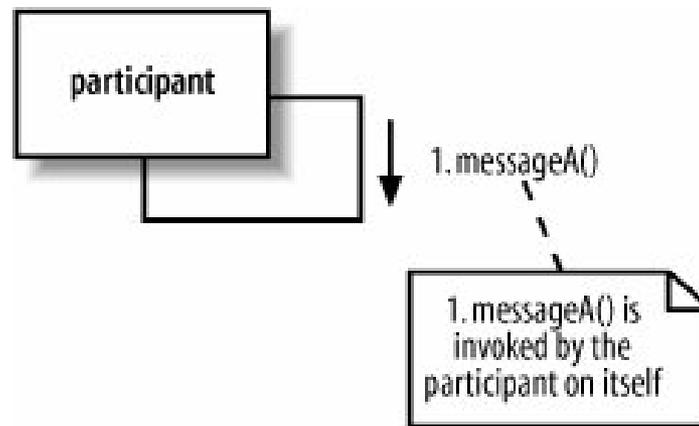


Figure 8-8. The participant can invoke 1. messageA() on itself because it has a communication line to itself



8.2. Fleshing out an Interaction with a Communication Diagram

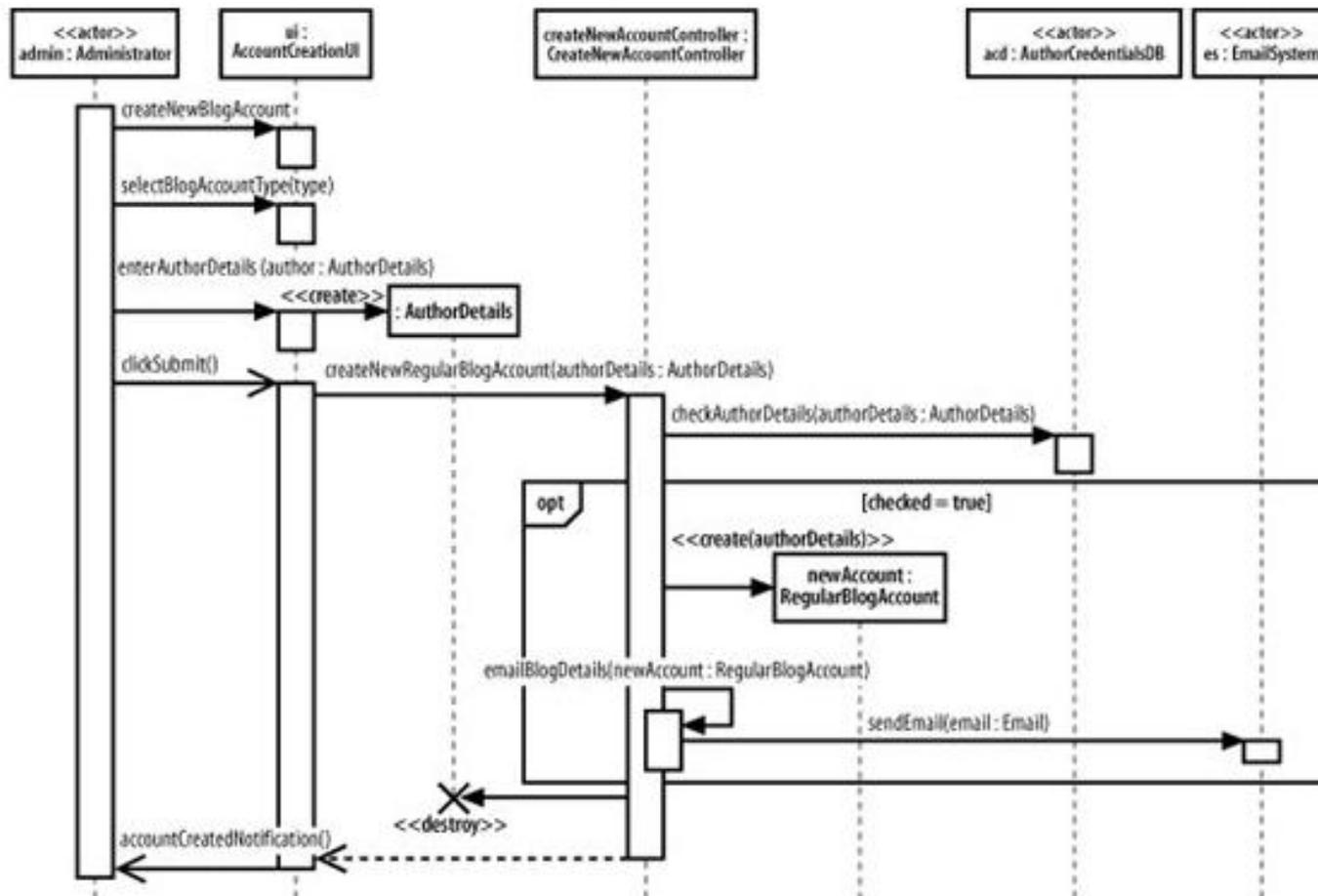


Figure 8-9. This sequence diagram describes the **interactions** that take place in a CMS when a new regular blog account is created

Figure 8-10. The **participants** involved in an interaction are often the first pieces added to a communication diagram

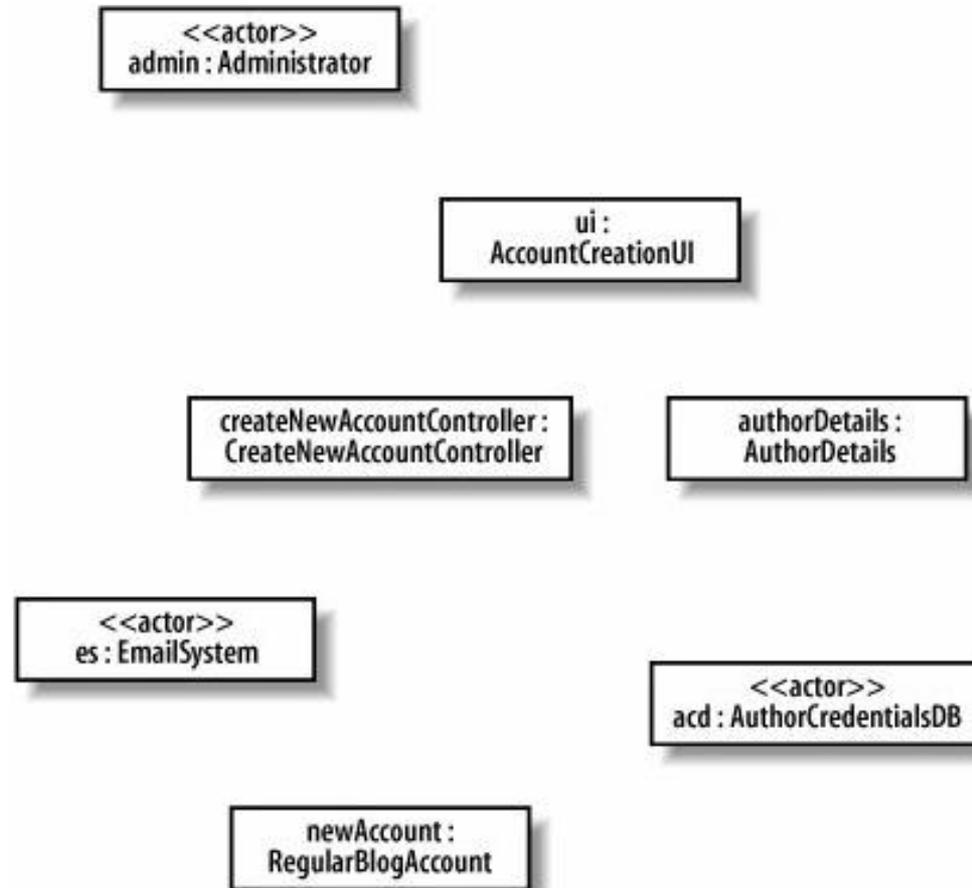


Figure 8-11. By looking at the sequence diagram in [Figure 8-9](#), the links required to support the message passing can be added to the communication diagram

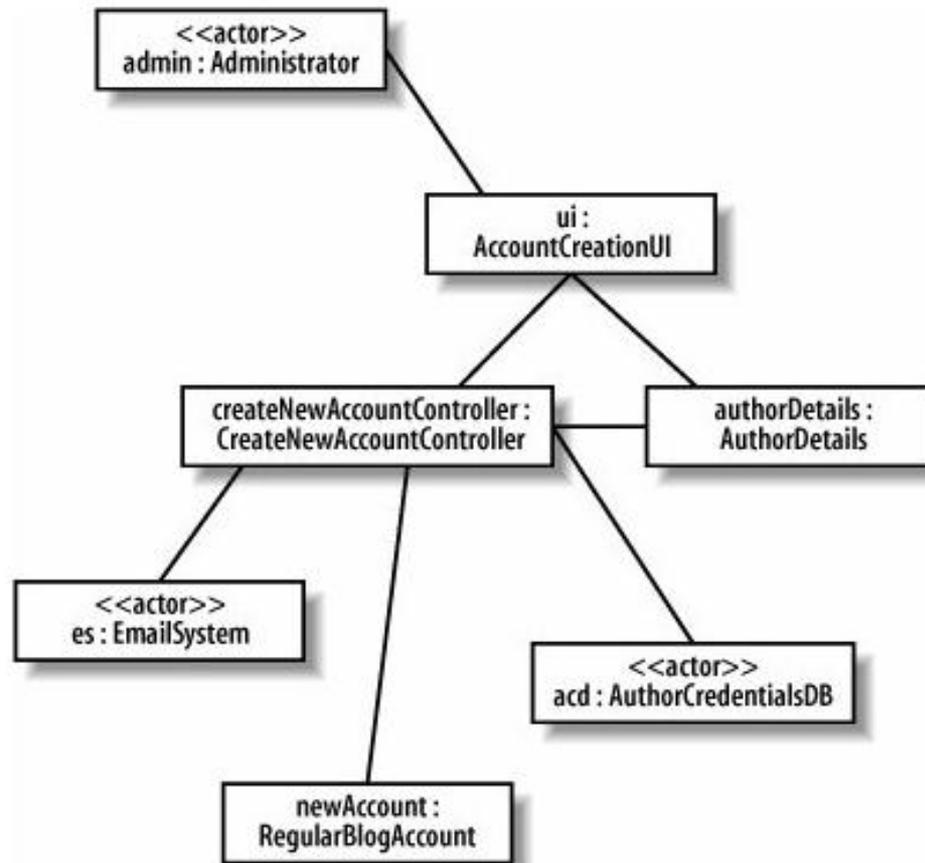


Figure 8-12. The Administrator actor starts things off by passing three separate messages to the ui:AccountCreationUserInterface participant

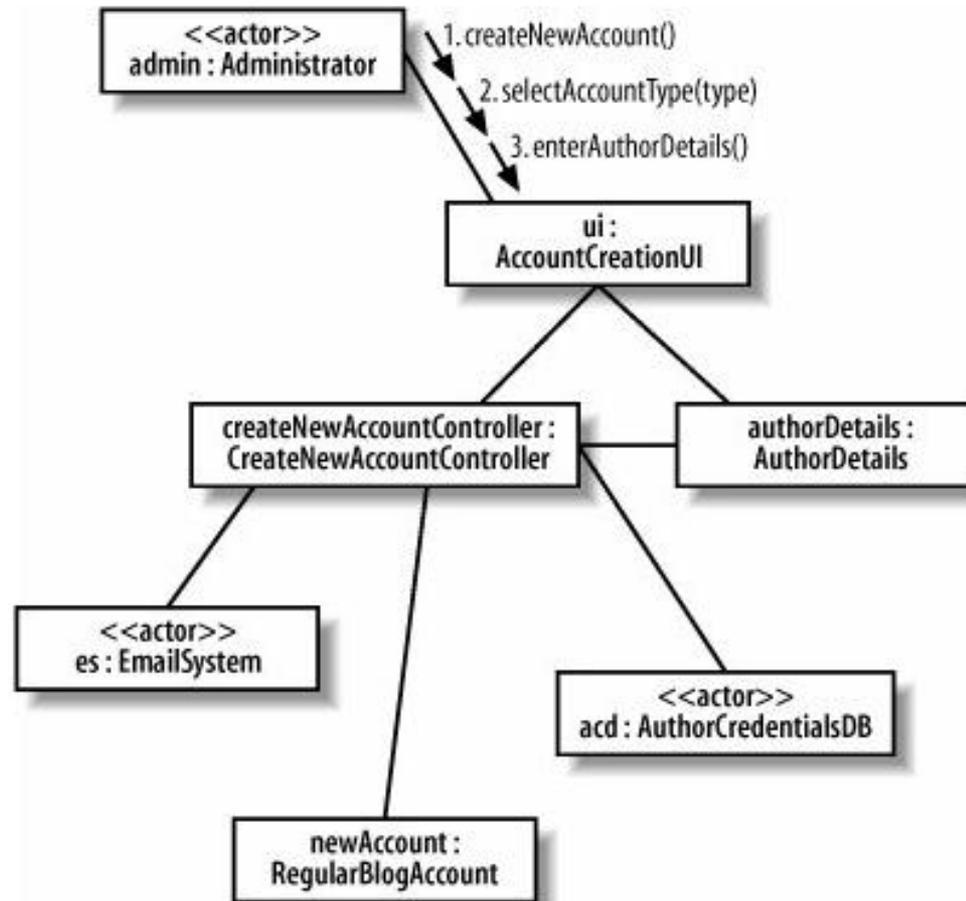


Figure 8-13. When the <<create>> message is added to the communication diagram, its message order number is set to 3.1. to show that it is nested inside the 3. enterAuthorDetails() message

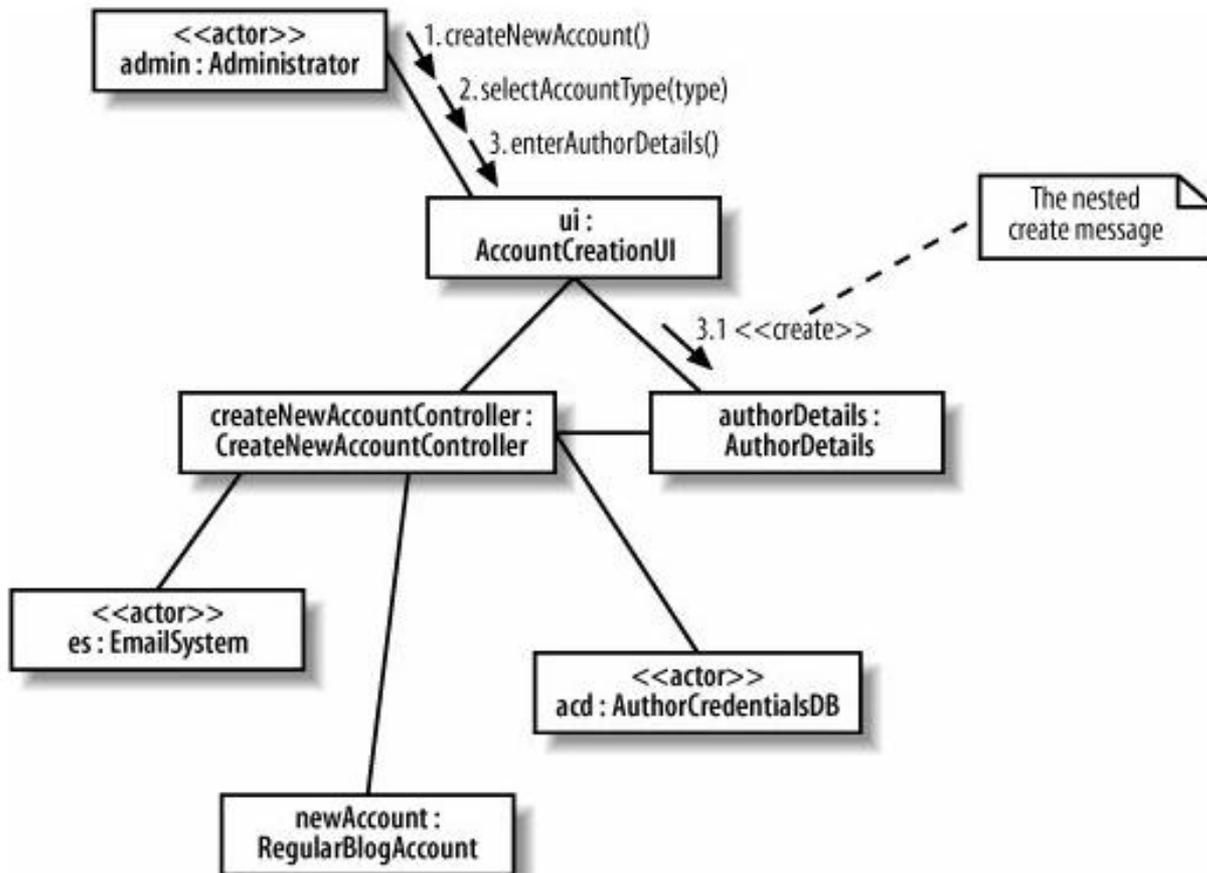
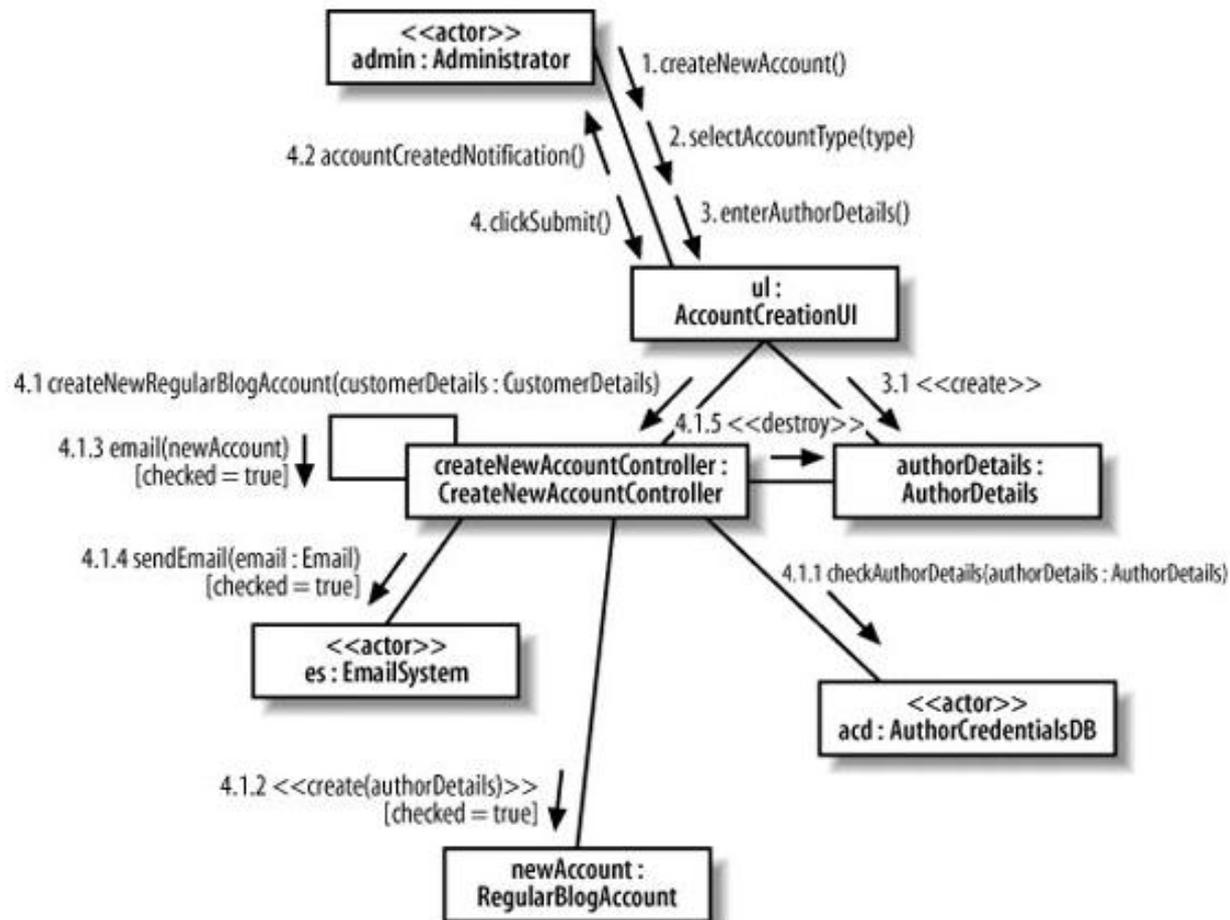
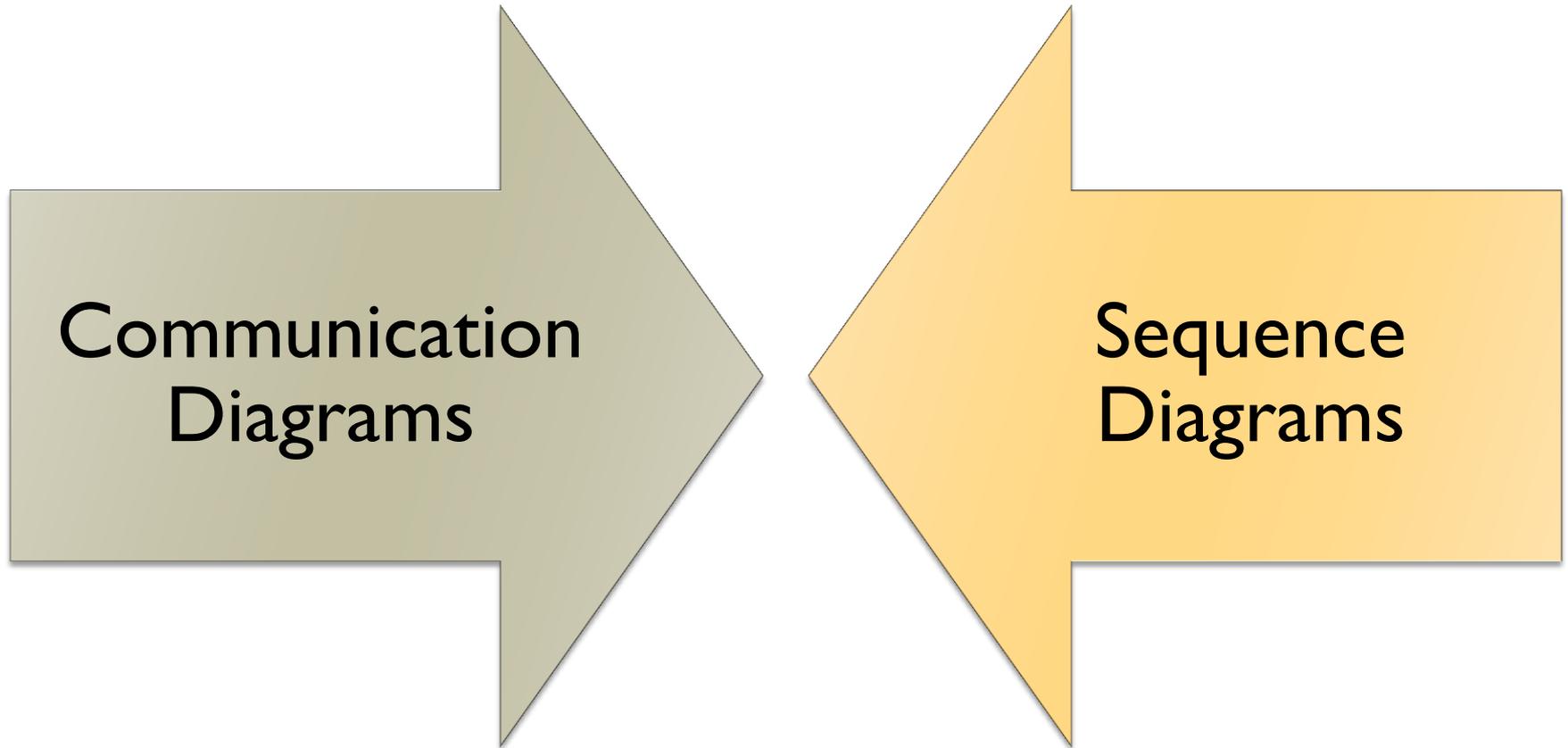


Figure 8-14. The finished communication diagram shows the complete set of messages within the **Create a new Regular Blog Account** interaction according to those shown on the original sequence diagram shown in [Figure 8-9](#)



8.3. Communication Diagrams Versus Sequence Diagrams



8.3.1. How the Fight Shapes Up

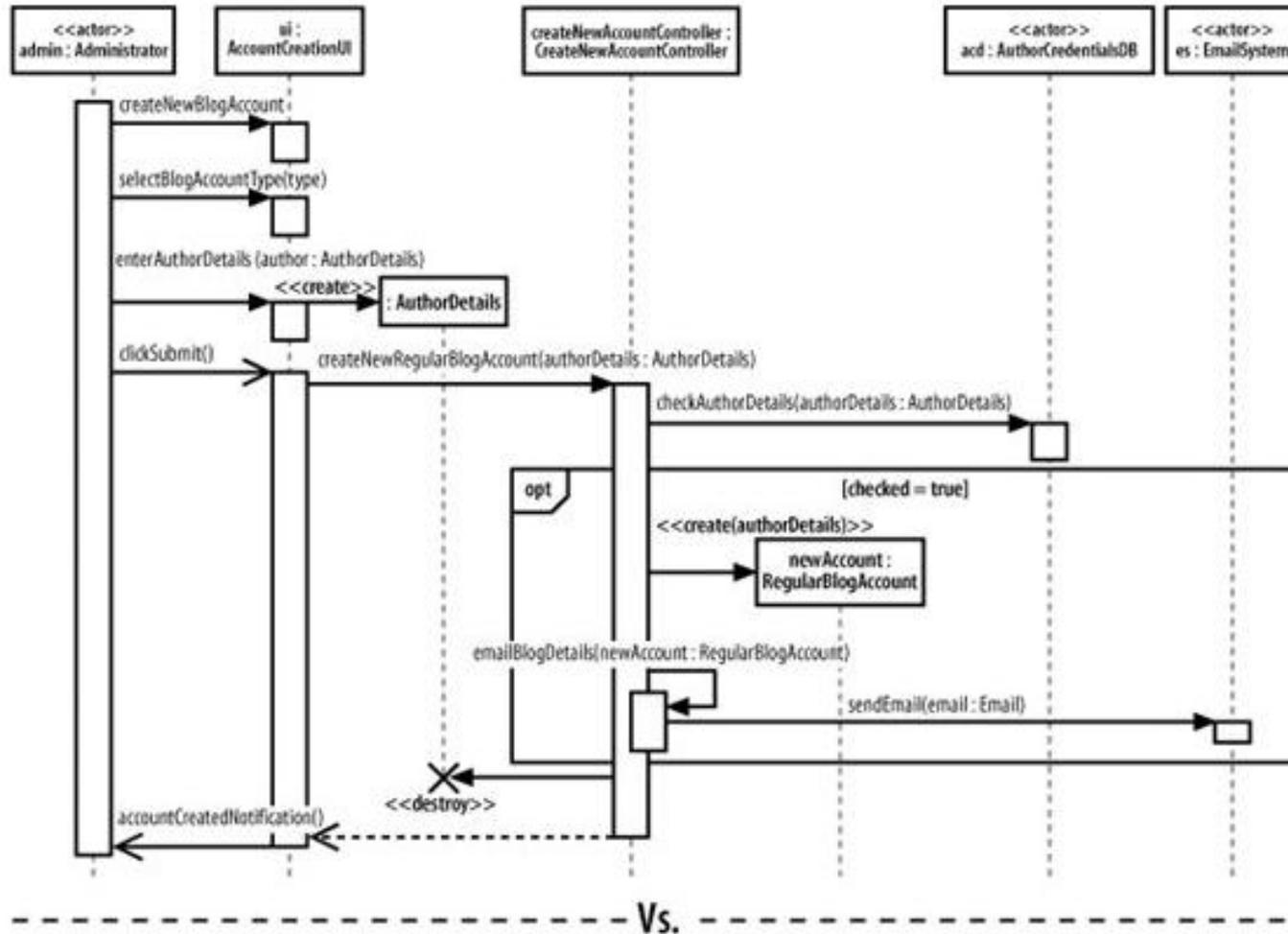


Figure 8-15. The **Create a new Regular Blog Account** interaction can be modeled using a sequence diagram and a communication diagram

8.3.1. How the Fight Shapes Up

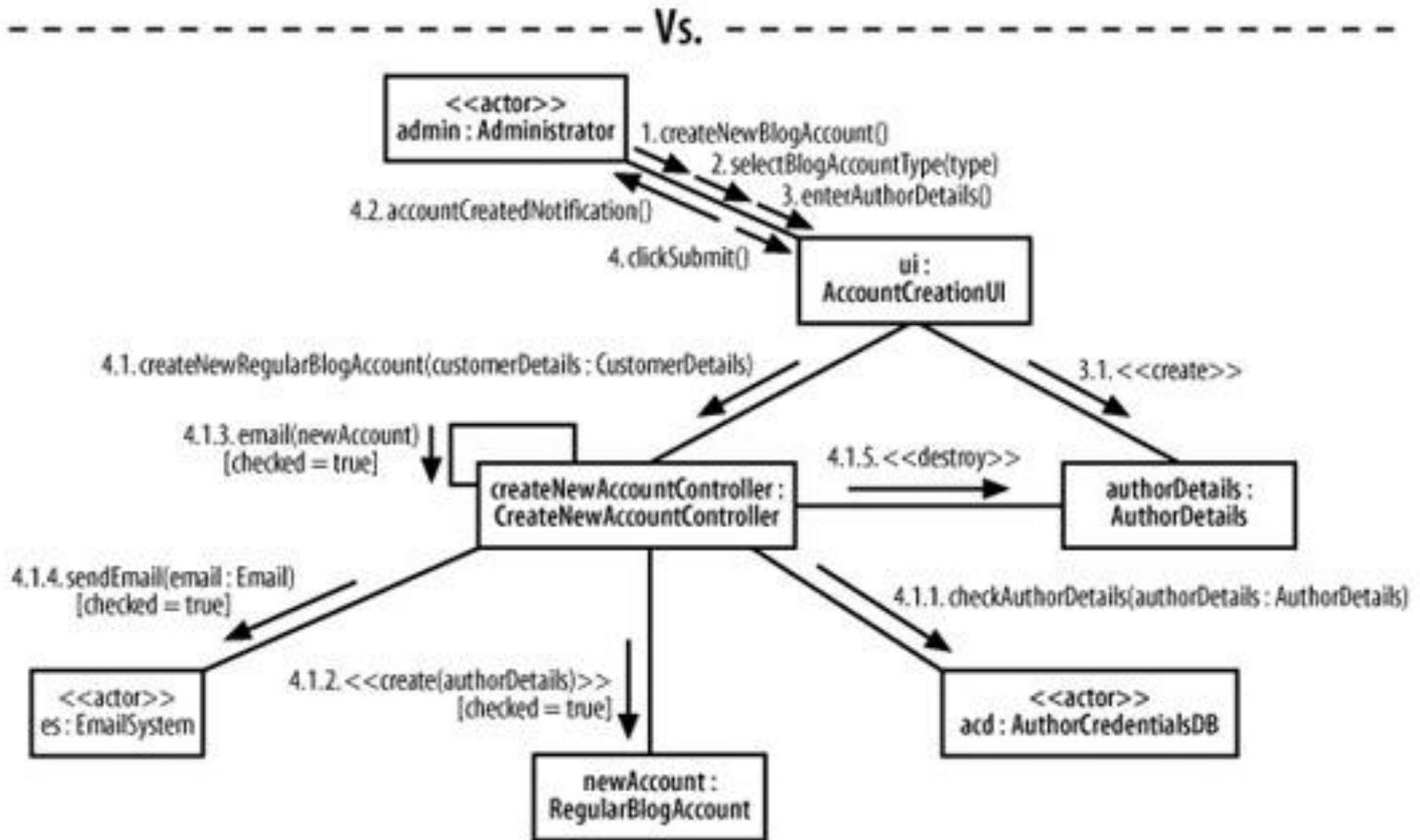


Figure 8-15. The **Create a new Regular Blog Account** interaction can be modeled using a sequence diagram and a communication diagram

8.3.2. The Main Event 各种指标对比

▶ Feature:

- ▶ Shows participants effectively 清晰显示参与者

▶ Sequence diagrams:

- ▶ Participants are mostly arranged along the top of page, unless the drop-box participant creation notation is used. It is easy to gather the participants involved in a particular interaction.

▶ Communication diagrams:

- ▶ Participants as well as links are the focus, so they are shown clearly as rectangles.

▶ The result:

- ▶ **Communication** diagrams barely win. 会话图更佳
- ▶ Although both types of diagram can show participants as effectively as each other, it can be argued that communication diagrams have the edge since participants are one of their main focuses.



8.3.2. The Main Event

- ▶ Feature:

- ▶ Showing the links between participants 显示参与者的链接

- ▶ Sequence diagrams:

- ▶ Links are **implied**. 隐含
- ▶ If a message is passed from one participant to another, then it is implied that a link must exist between those participants.

- ▶ Communication diagrams:

- ▶ **Explicitly** shows the links between participants. In fact, this is the primary purpose of these types of diagram. 显式

- ▶ The result:

- ▶ **Communication** diagrams win because they explicitly and clearly show the links between participants. 会话图更佳



8.3.2. The Main Event

- ▶ Feature:

- ▶ **Showing message signatures 显示消息签名**

- ▶ Sequence diagrams:

- ▶ Message signatures can be fully described.

- ▶ Communication diagrams:

- ▶ Message signatures can be fully described.

- ▶ The result:

- ▶ **Draw! 平手**

- ▶ Both types of diagram can show messages as effectively as each other.



8.3.2. The Main Event

- ▶ Feature:

- ▶ Supports asynchronous messages (fire and forget) 异步消息

- ▶ Sequence diagrams: 支持

- ▶ Achieved using the asynchronous arrow.

- ▶ Communication diagrams: 不支持

- ▶ Communication diagrams have no concept of the asynchronous message since its focus is not on message ordering.

- ▶ The result:

- ▶ **Sequence** diagrams are a clear winner here because they explicitly support asynchronous messages.



8.3.2. The Main Event

- ▶ Feature:

- ▶ Easy to read message ordering 消息顺序

- ▶ Sequence diagrams: 直接支持

- ▶ This is a sequence diagram's forte.

- ▶ Sequence diagrams clearly show message ordering using the vertical placement of messages down the diagram's page.

- ▶ Communication diagrams: 使用数字

- ▶ Shown using the number-point-nested notation.

- ▶ The result:

- ▶ **Sequence** diagrams are a clear winner here since they really show off message ordering clearly and effectively.



8.3.2. The Main Event

- ▶ Feature:
 - ▶ Easy to create and maintain the diagram 画图方便性
 - ▶ Sequence diagrams:
 - ▶ Creating a sequence diagram is fairly simple.
 - ▶ However, maintaining sequence diagrams can be a nightmare unless a helpful UML tool is being used.
 - ▶ Communication diagrams:
 - ▶ Communication diagrams are simple enough to create;
 - ▶ however, maintenance, especially if message numbering needs to be changed, still ideally needs the support of a helpful UML tool.
 - ▶ The result:
 - ▶ This is a difficult one to judge and is largely based on personal preference.
 - ▶ However, **communication** diagrams do have the edge on the ease-of-maintenance stakes.
-



How to choose?

- ▶ Use sequence diagrams if you are mainly interested in **the flow of messages** throughout a particular interaction.
主要关注于消息流
- ▶ Use communication diagrams if you are focusing on **the links between the different participants** involved in the interaction.
主要关注于参与者之间的关系



Summary

- ▶ 8. Focusing on Interaction Links: **Communication Diagrams** 会话图
 - ▶ 8.1. Participants, Links, and Messages
 - ▶ 8.2. Fleshing out an **Interaction** with a Communication Diagram
 - ▶ 8.3. Communication Diagrams Versus Sequence Diagrams



See you ...

