Object Oriented Design and Analysis
CPE 372

Lecture 7

Sequence Diagrams

Dr. Sally E. Goldin
Department of Computer Engineering
King Mongkut's University of Technology Thonburi
Bangkok, Thailand

Software Design:
Both Structure and Behavior

Data

Components

Relationships

Outputs

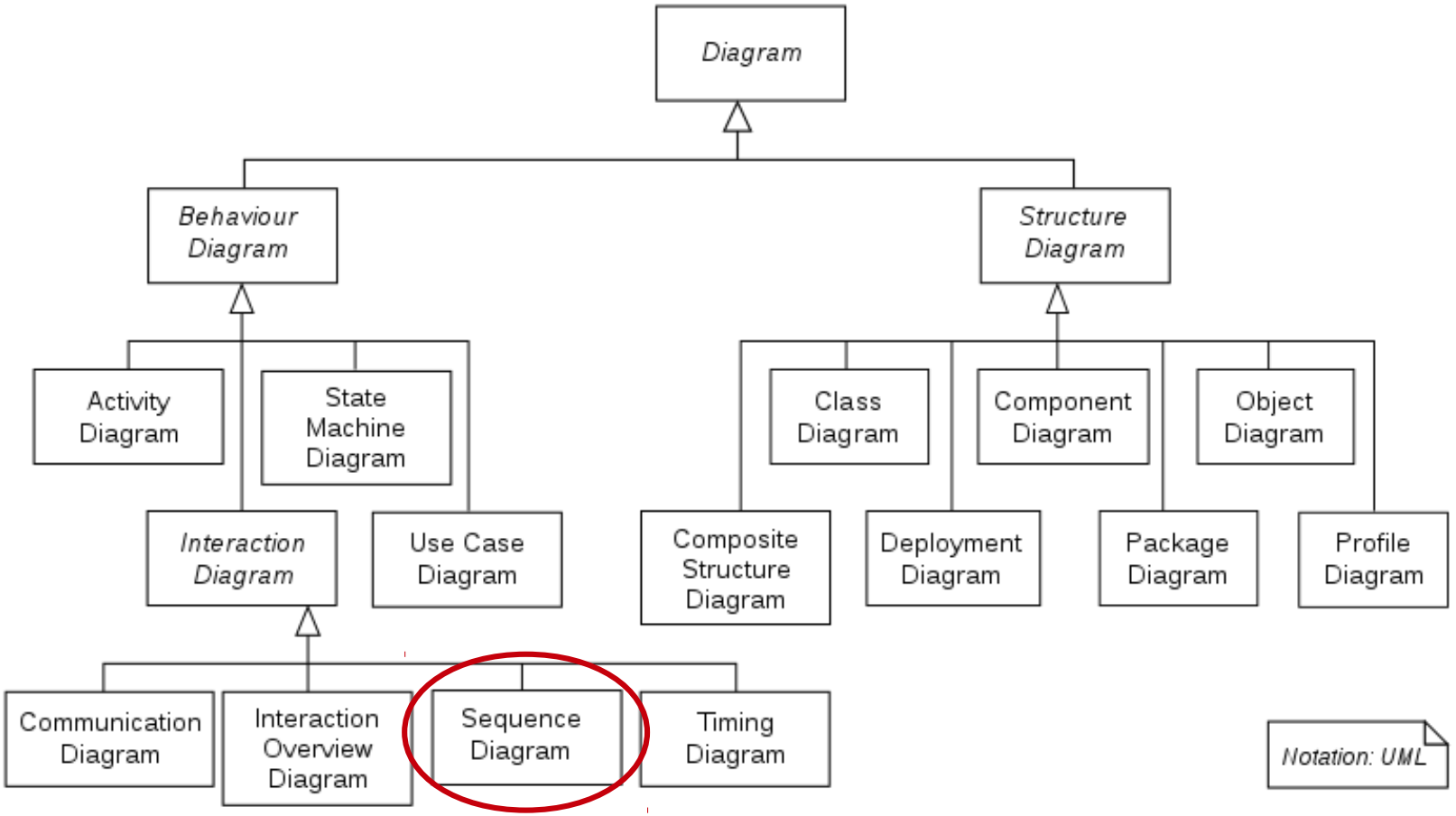
Actions

Interactions

Algorithms

State changes

UML Sequence Diagrams



What is a sequence diagram?

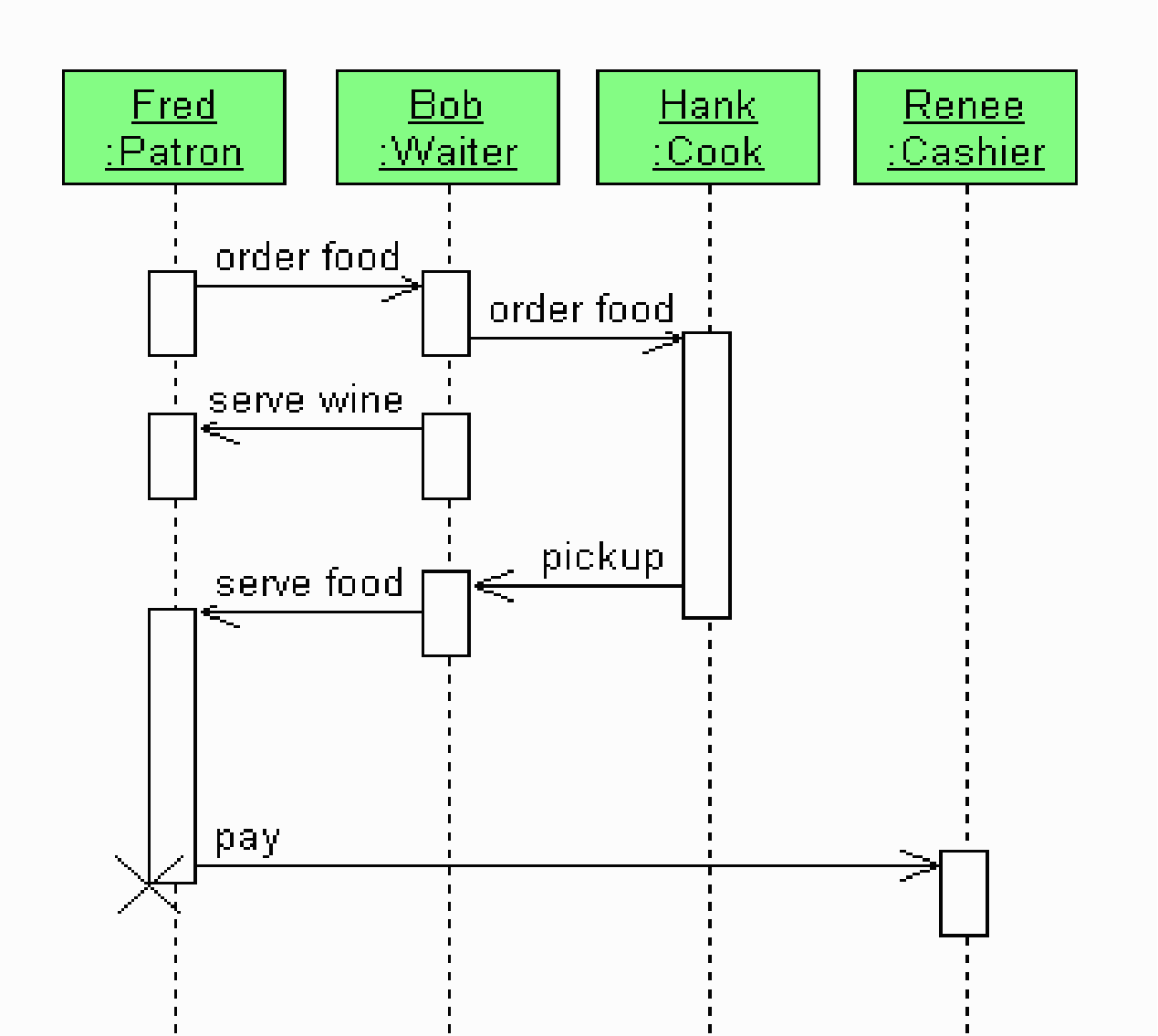
"**Interaction diagrams** describe how groups of objects collaborate in some behavior. The UML defines several forms of interaction diagrams, of which the most common is sequence diagrams.

Typically, a sequence diagram captures the behavior of a single scenario. The diagram shows a number of example objects and the messages that are passed between these objects within the use case." Martin Fowler, *UML Distilled 3rd Edition (2004)*

Important words:

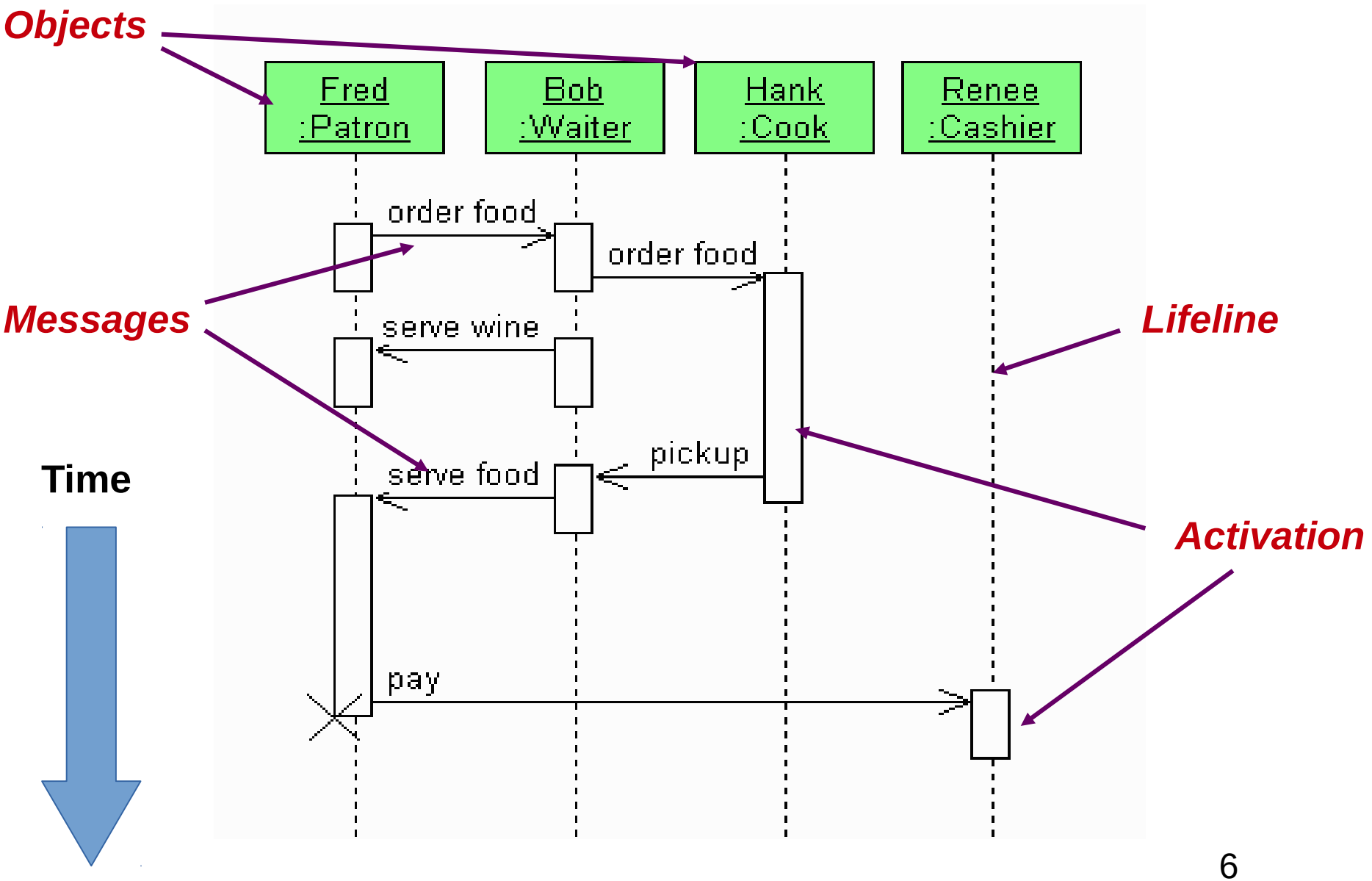
- *Objects (class instances)*
- *Messages (method calls)*
- *Collaborate (work together)*

A Simple Sequence Diagram

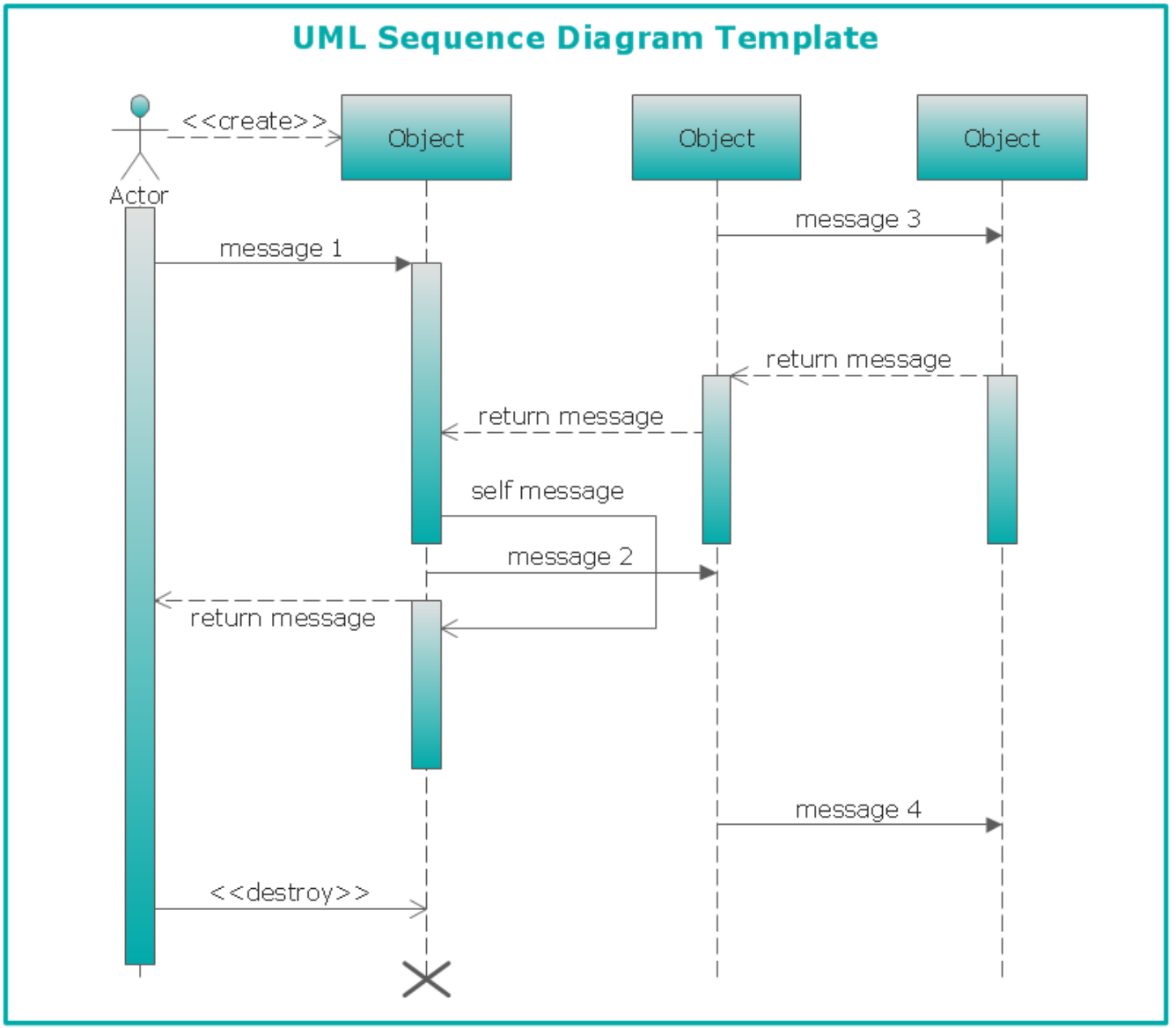


https://commons.wikimedia.org/wiki/File:Restaurant-UML-SEQ.gif

Some Sequence Diagram Components



More Components



http://www.conceptdraw.com/How-To-Guide/picture/UML-sequence-diagram-template.png

Why create sequence diagrams?

Use cases => interaction between actors and the "system"

Sequence diagrams => interaction between objects *within* the system

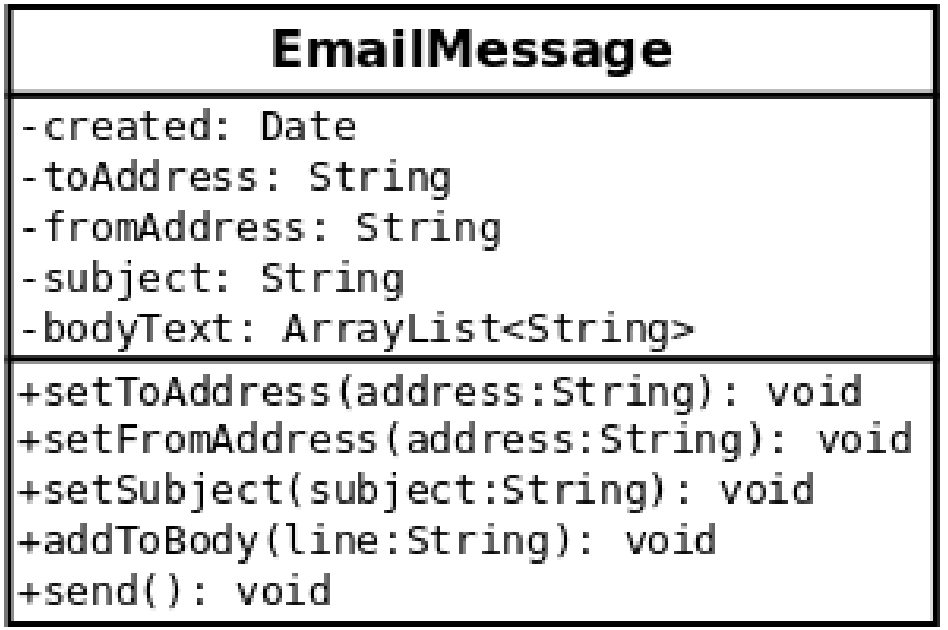
Expand behavioral description to the next levels of detail

Interaction reveals structure

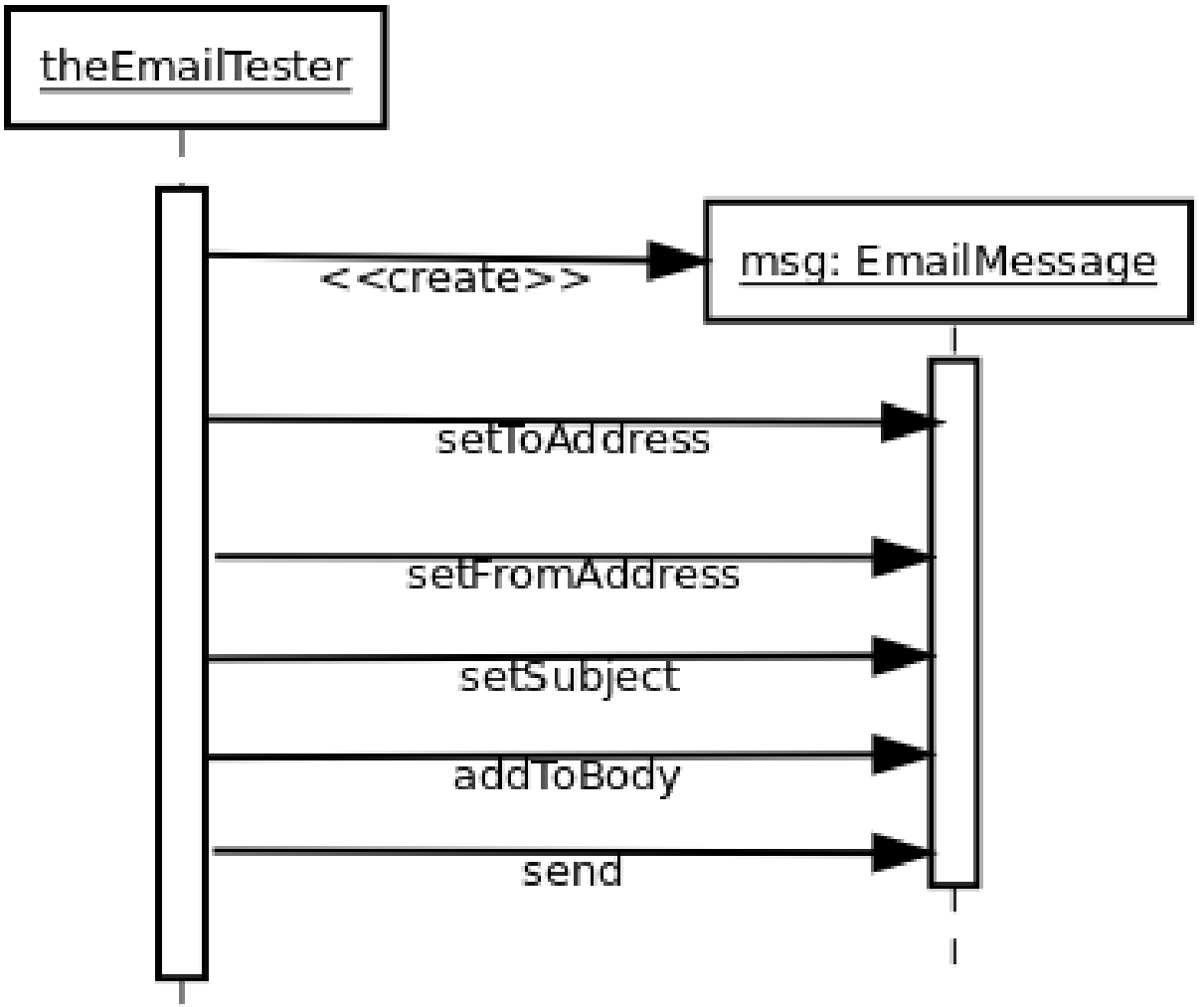
Creating sequence diagrams can help you to:

- *Discover new classes*
- *Identify new methods*
- *Refine your ideas about relationships*

Class Diagram for Exercise 1

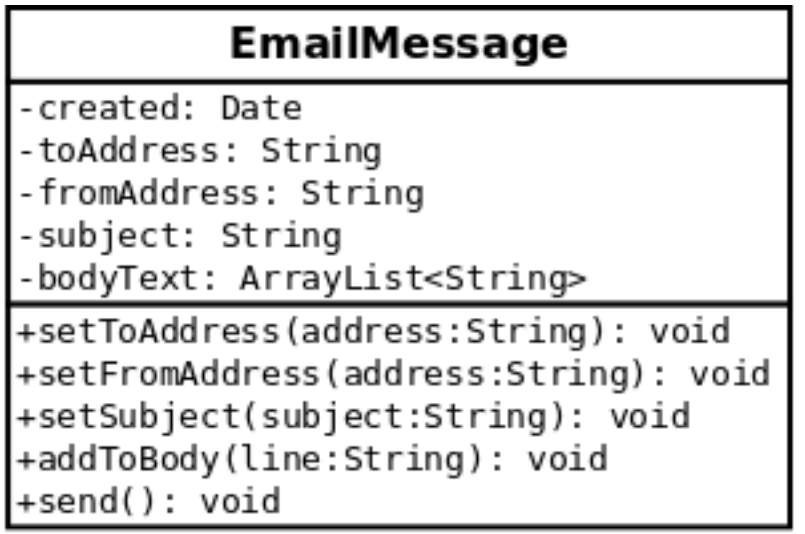


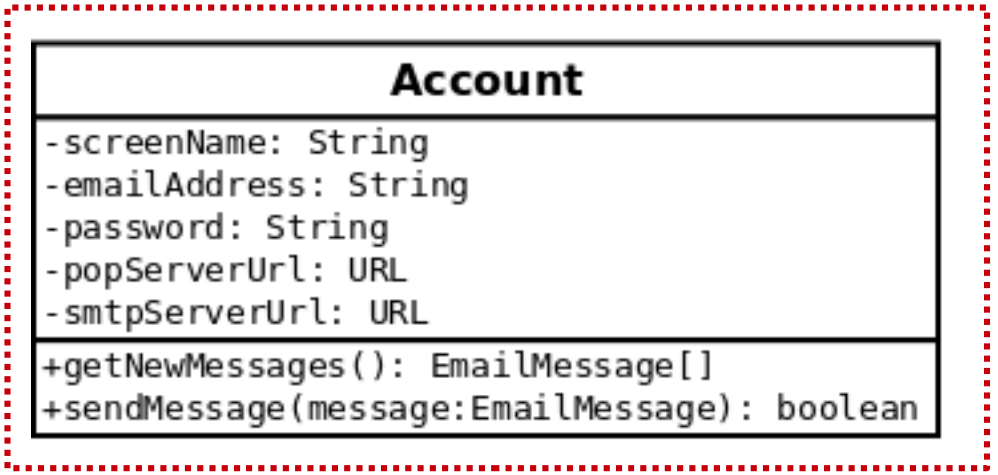
Sequence Diagram 1



This was fine for our simple Java exercise, but what if we want to start building an actual email application?

Add Account Class



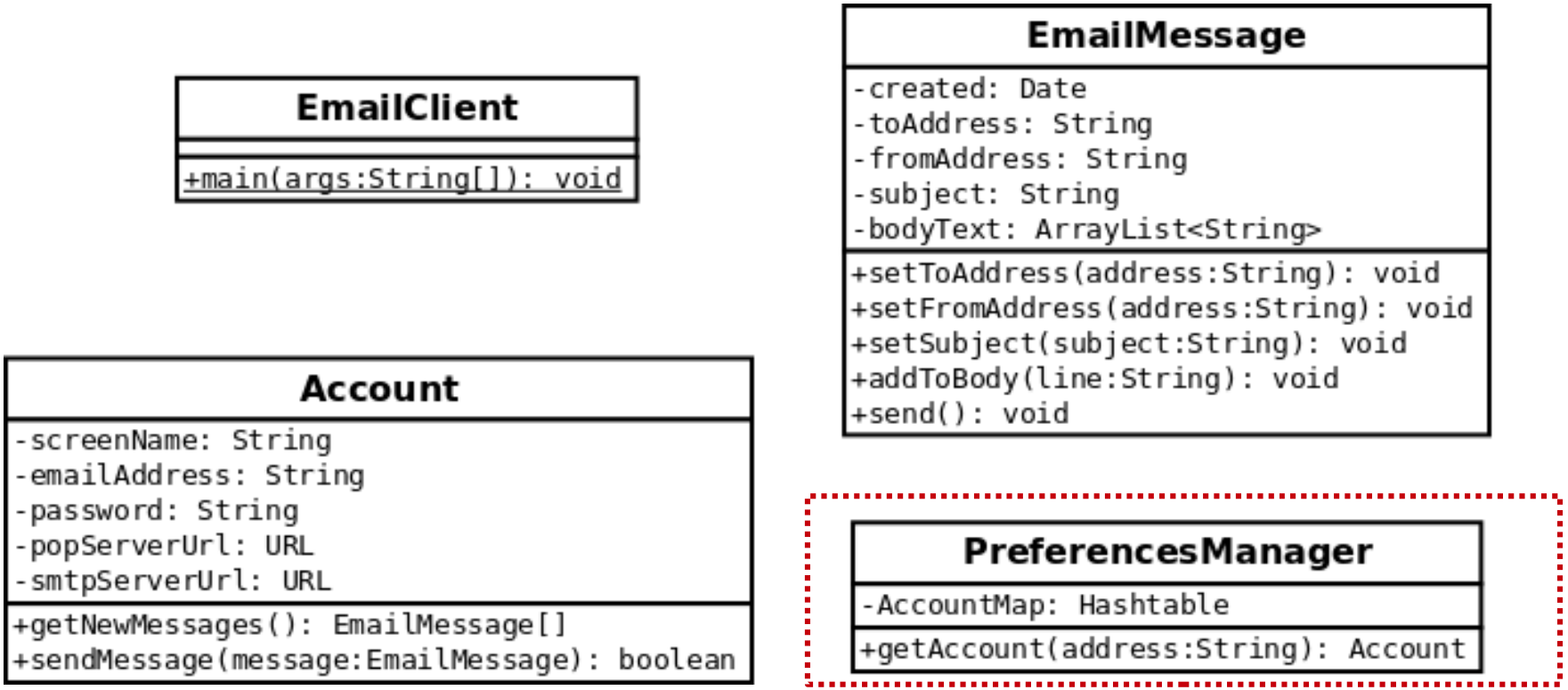


Introduced in example last week

Knows about servers associated with a particular email address

Knows how to send and receive emails via those servers

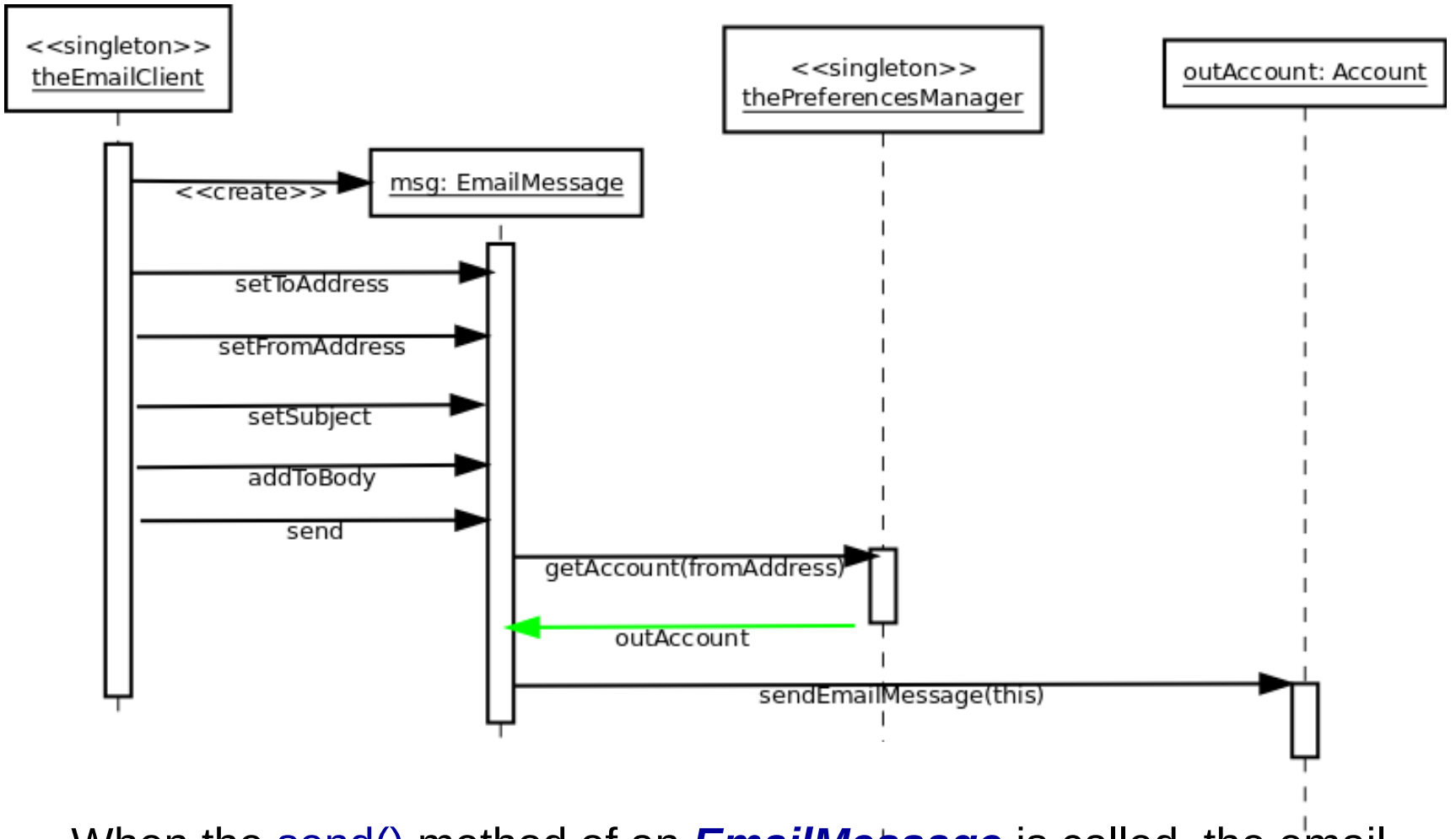
Who knows which account to use?



To implement the send() method on ***EmailMessage***, need the ***Account***

We introduce a new class, ***PreferencesManager***, to associate email addresses with accounts

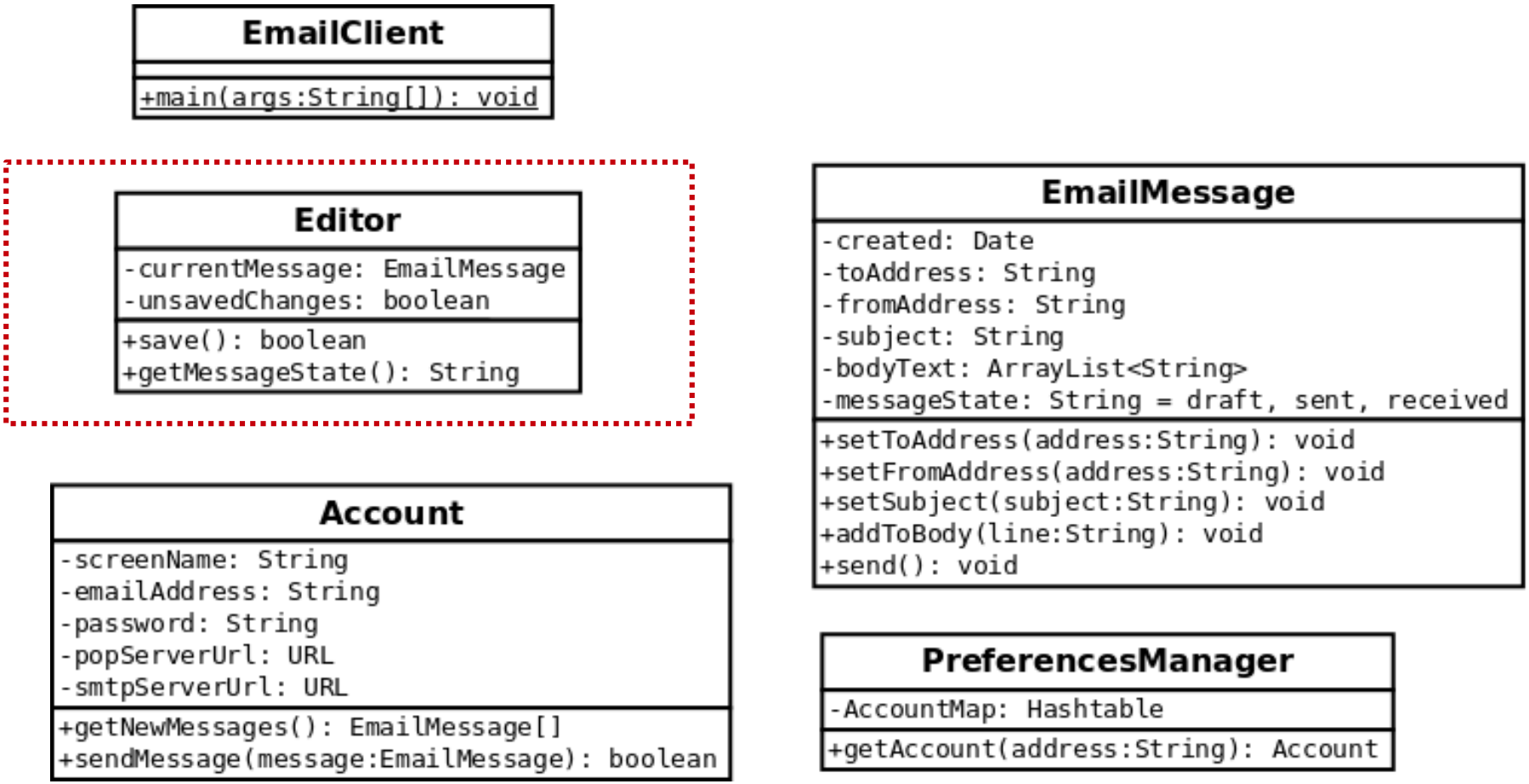
Sequence Diagram 2



When the send() method of an ***EmailMessage*** is called, the email

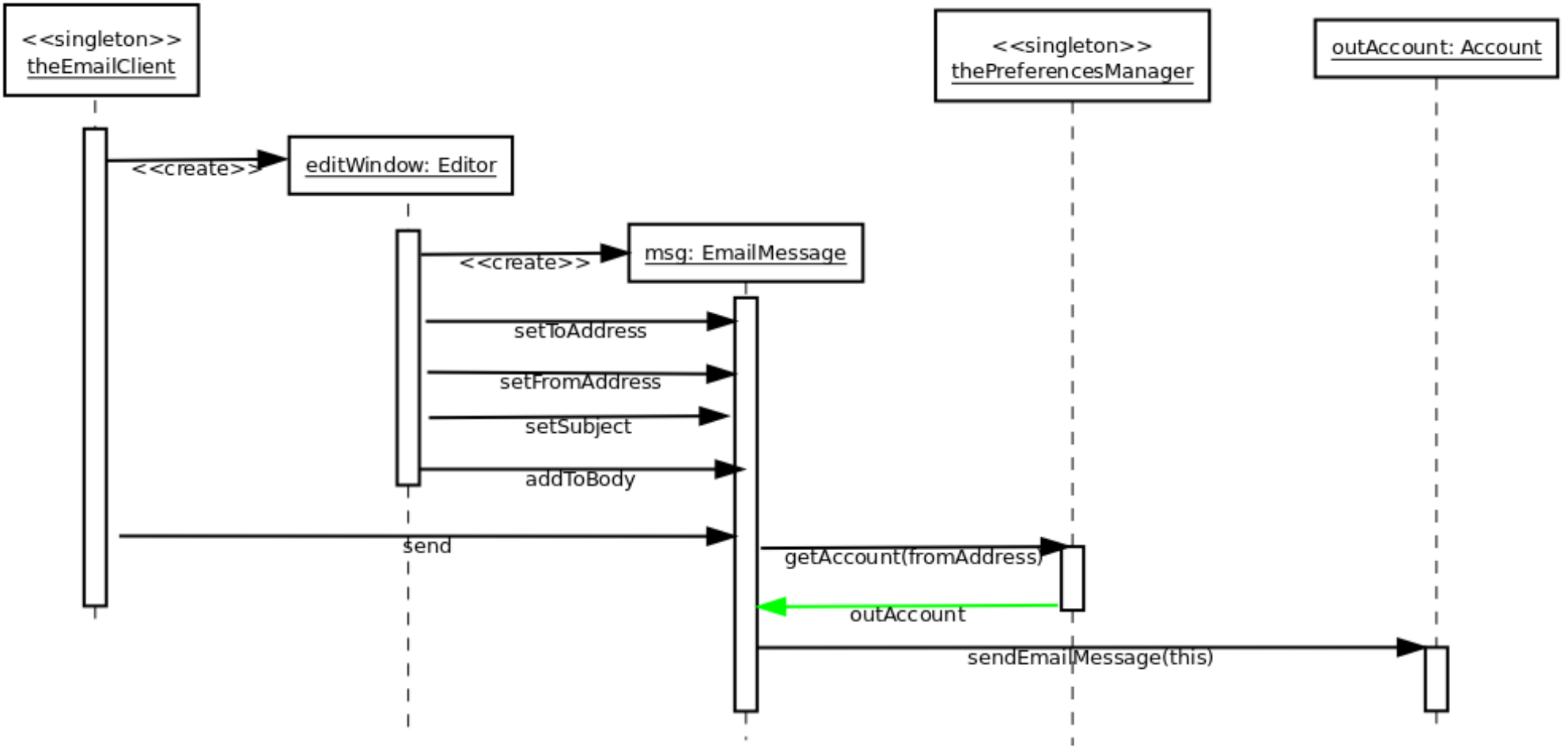
1. Calls getAccount() method on the *PreferencesManage*r
2. Gets the ***Account*** object back as a return value
3. Calls the sendEmailMessage() method on the account object

Should the ***EmailClient*** directly create ***EmailMessage*** objects?



Add an ***Editor*** class to enter and modify email text.

Sequence Diagram 3



The ***Editor*** controls the content of the ***EmailMessage***

However, the top level client is still responsible for the send() command

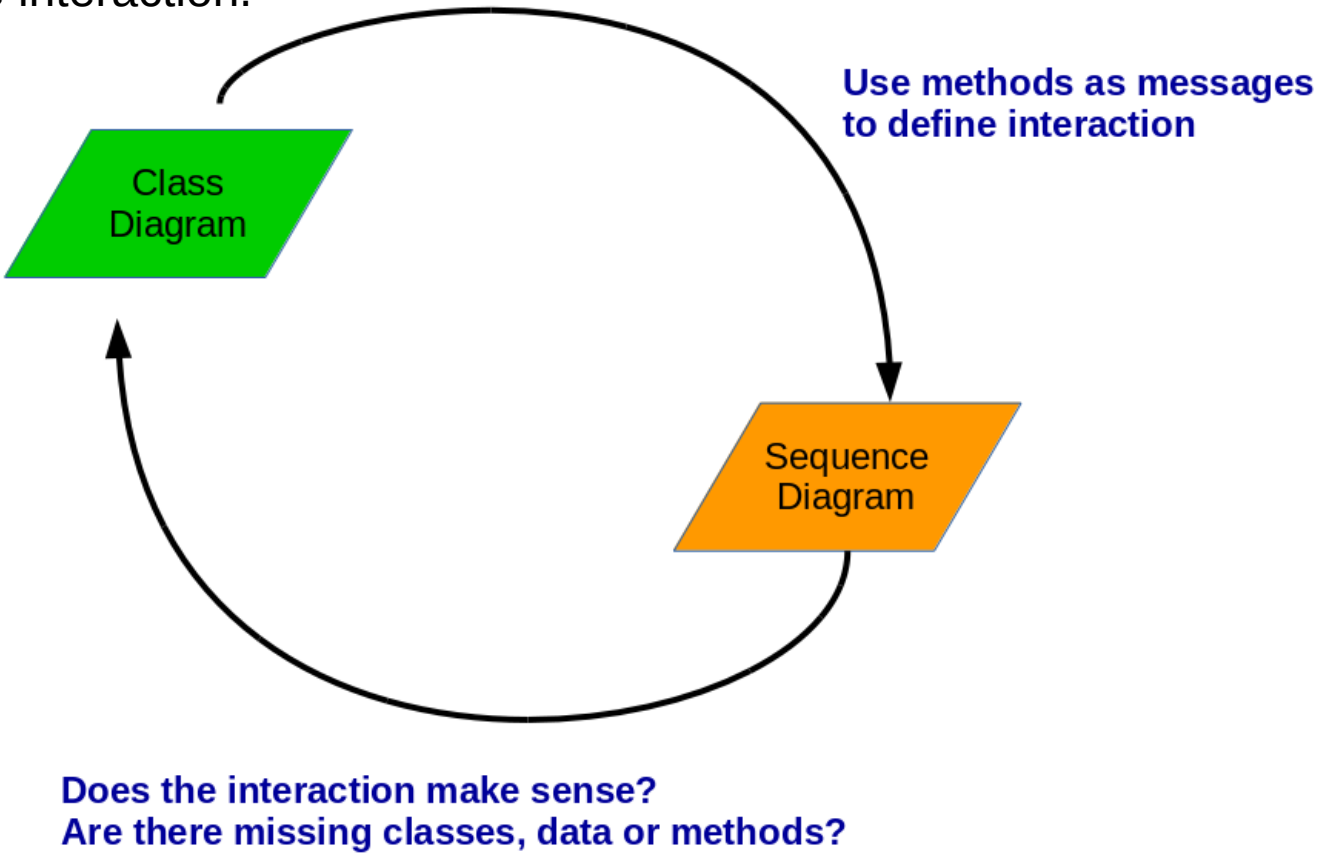
Iterating between Structure and Interaction

This example shows the true benefit of using UML.

The diagrams focus the designer's attention on specific aspects of the system design.

Studying interaction suggests changes to structure.

Modified structure changes the interaction.



More sequence diagram notation

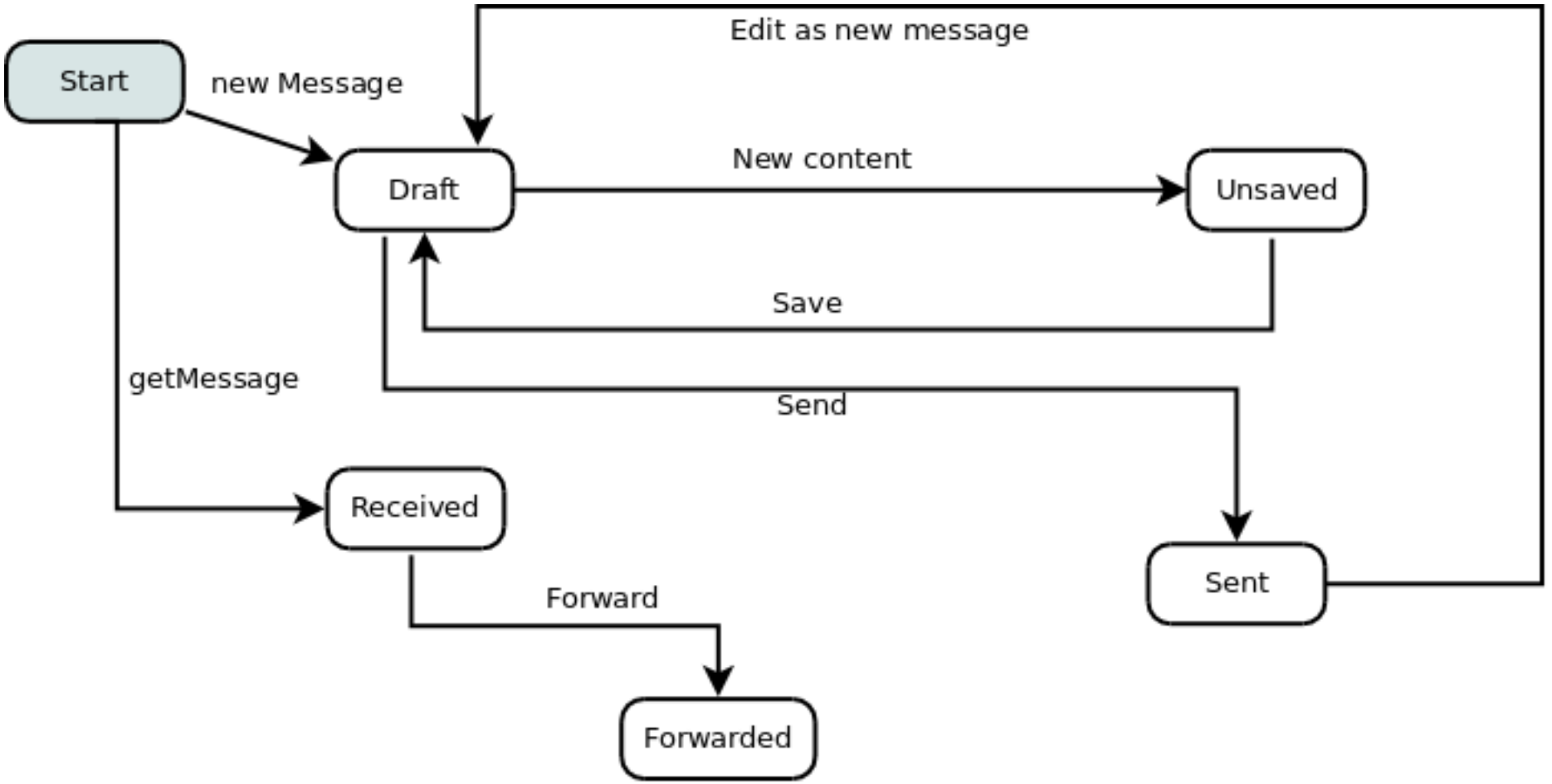
- Self-messages – when an object calls one of its own methods
- Iteration and conditional notations (different between UML 1 and UML 2)
- Synchronous versus asynchronous messages
- Notations for showing passed and returned information

Use what you think you need

Sequence diagrams in general are not good for capturing detailed logic

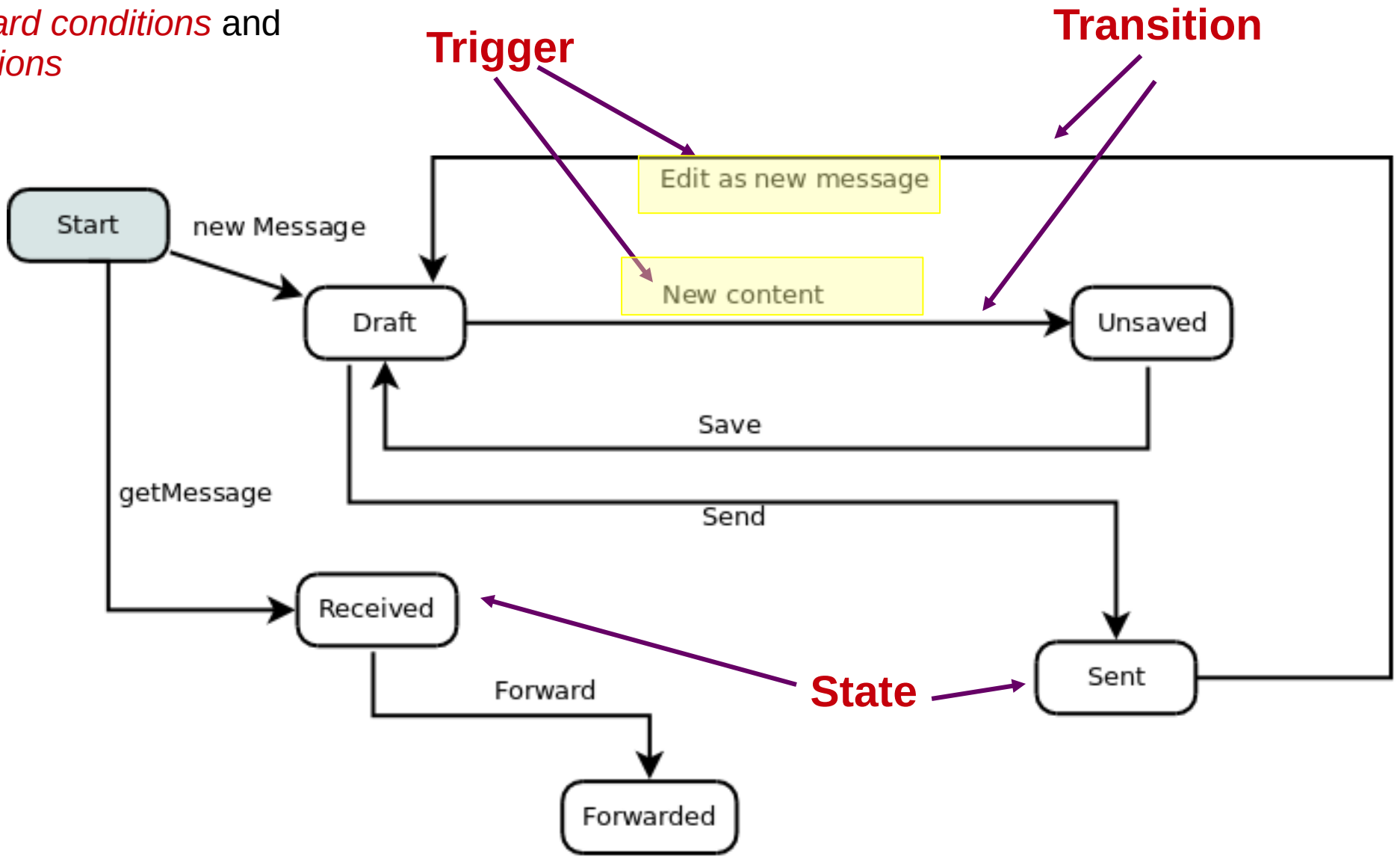
State Diagrams

State diagram for ***EmailMessage***

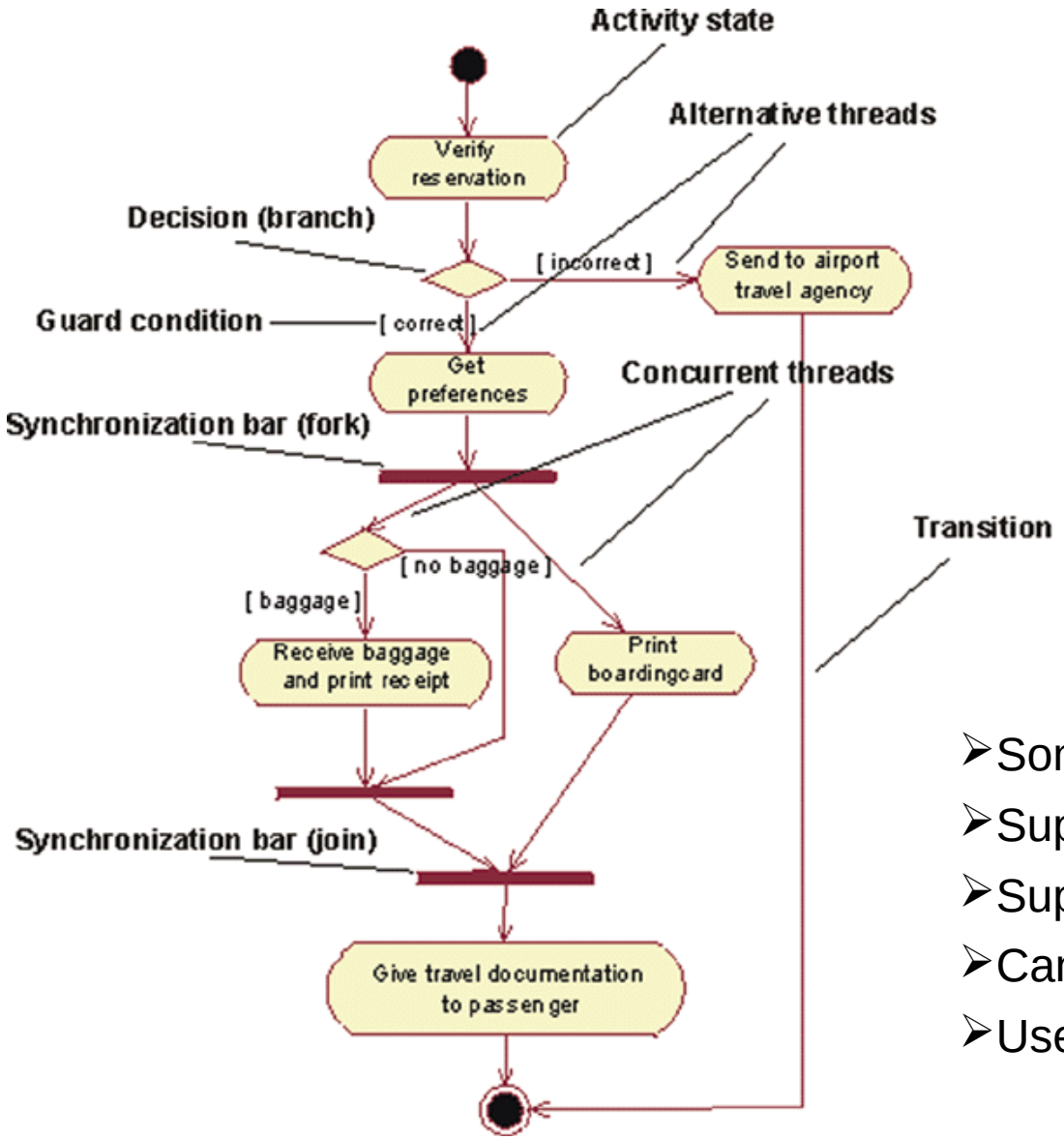


State Diagrams

Transitions can also have *guard conditions* and *actions*



Activity Diagrams



- Somewhat similar to flow charts
- Support concurrent activities
- Support external “signals”
- Can get complicated
- Used in many different ways

https://www.ibm.com/developerworks/rational/library/2802.html

Next Two Weeks

Midterm period – no class (and no exam)

Work on your project design plus assignments (next slide)

First draft of project design document due **October 17th** by noon (electronic submission)

PDF document with the following (in this order):

1. Title page – topic, team name, team members
2. Abstract – one to two paragraph description of your system
3. Use case diagram
4. Use case narratives (text) for each use case in diagram
5. Class diagram(s)
6. Sequence diagrams for main success scenario of each use case
7. A list of unresolved issues about your design – things you are not sure about

Please put titles on all diagrams. Please be sure all are readable.
No hand-drawn diagrams accepted.
Remember your classmates will be seeing and evaluating your design!

Assignments

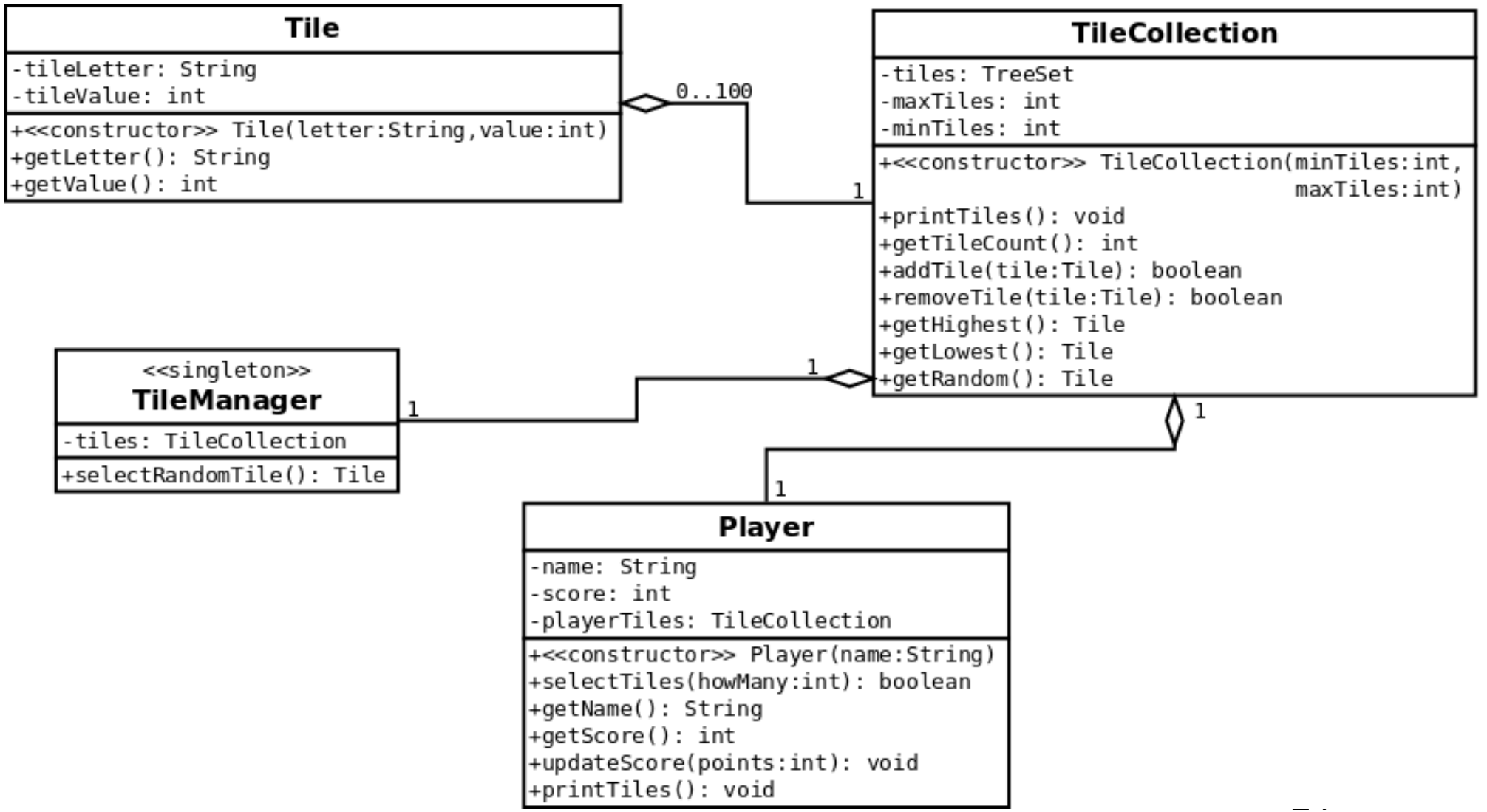
- **Read chapter 4 from *UML Distilled* (on web page)**
- **Given the class diagram and the sequence diagram on the following pages, *implement and test* these classes and methods in Java**

Due on Sunday, October 1st. Upload your Java files in a zip or tar (not rar) file.

Notes:

- Methods on *TileCollection* and *Player* that return boolean values will return true, unless the operation violates the limits (maxTiles, minTiles) of the *TileCollection*
- Create a main() method in the *Player* class to test your code. This method should call the Player() constructor to create an instance of a player, and then call the selectTiles() method on that instance. Then call printTiles() to show the tiles that were selected.
- Pass a value of 7 for the howMany argument to selectTiles(). This should also be the maximum limit on the playerTiles tile collection.

Class Diagram for Exercise



Sequence Diagram for Exercise