










Object Oriented Analysis & Design

Engr. Abdul-Rahman Mahmood

DPM, MCP, QMR(ISO9001:2000)

 armahmood786@yahoo.com
 alphapeeler.sf.net/pubkeys/pkey.htm
 pk.linkedin.com/in/armahmood
 www.twitter.com/alphapeeler
 www.facebook.com/alphapeeler
 abdulmahmood-sss  alphasecure
 armahmood786@hotmail.com
 <http://alphapeeler.sf.net/me>

 alphasecure@gmail.com
 <http://alphapeeler.sourceforge.net>
 <http://alphapeeler.tumblr.com>
 armahmood786@jabber.org
 alphapeeler@aim.com
 mahmood_cubix  48660186
 alphapeeler@icloud.com
 <http://alphapeeler.sf.net/acms/>

Entity Control & Boundary Classes

Rules for Analysis / Robustness

Classes / ECB Pattern

- **Structural restrictions for analysis classes**
 - Entity: only attributes (+get/set/find methods)
 - Control: only methods: (at least) one method / UC
 - Boundary: both attributes and methods
- **Relationship between analysis classes (Layers)**
 - Actors access only boundaries
 - One boundary class for each Actor-UC relation
 - Entities are only accessed by control objects
 - Control objects may communicate with all entities, boundaries, and control objects

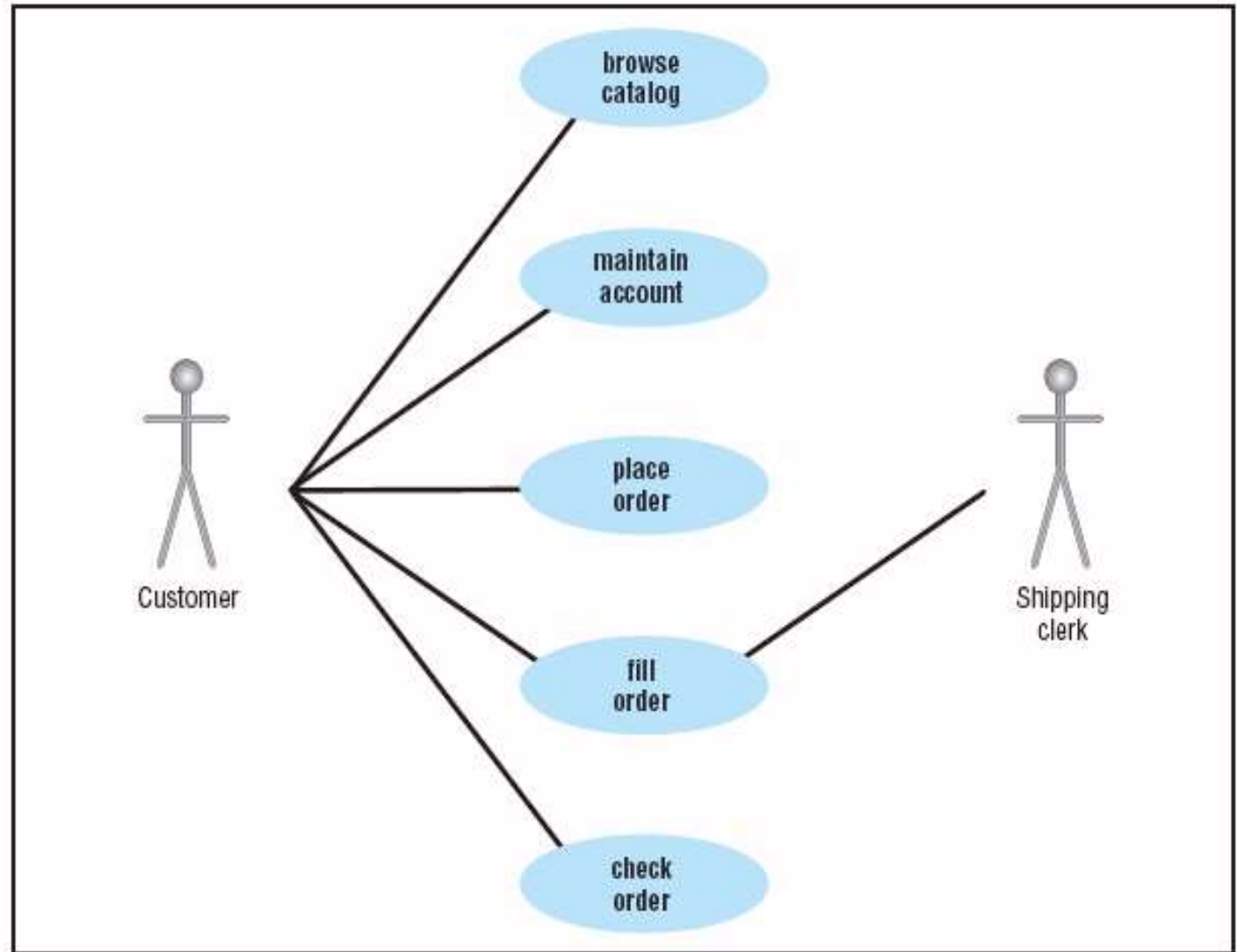
Communication allowed:

| | Entity | Boundary | Control |
|----------|--------|----------|---------|
| Entity | | | X |
| Boundary | | | X |
| Control | X | X | X |

Communication allowed:

| | Entity | Boundary | Control |
|----------|--------|----------|---------|
| Entity | X | | X |
| Boundary | | | X |
| Control | X | X | X |

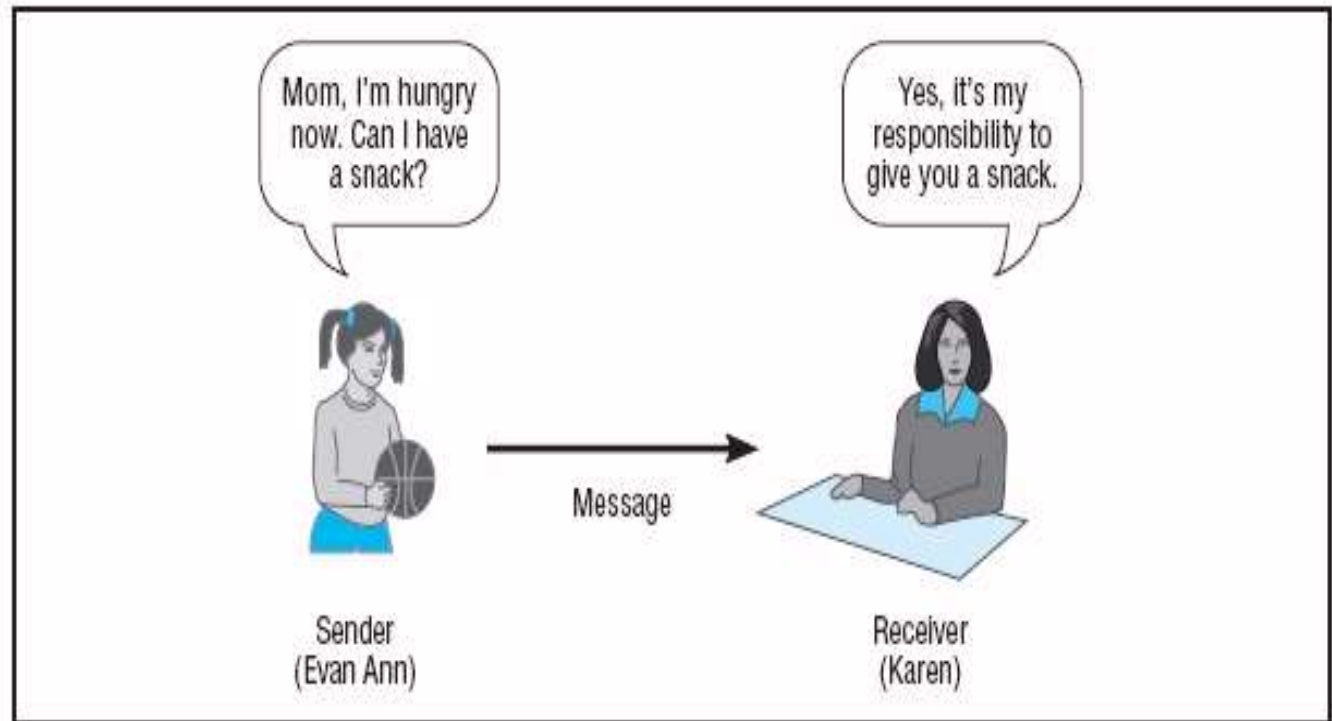
Figure 9.4 The Use-Case Diagram of the WebStore System



What Is a Sequence Diagram?

- A UML diagram that shows the interaction between objects to perform critical pieces of use case behavior in a time-ordered manner
- Interactions are in the form of *messages*
- Behavioral *responsibilities* are assigned to message recipients

Figure 9.5
Message Passing



Messages are passed between objects, and may contain parameters

Symbols in a Sequence Diagram

- Box or class stereotype symbol represents objects.
- Dotted vertical line represents object's lifetime.
- Thin bars represent focus of control, periods of time when object is behaving (fulfilling responsibilities).
- Labeled horizontal lines represent messages passing between objects.

Figure 9.6
Message Passing in
a Sequence Diagram

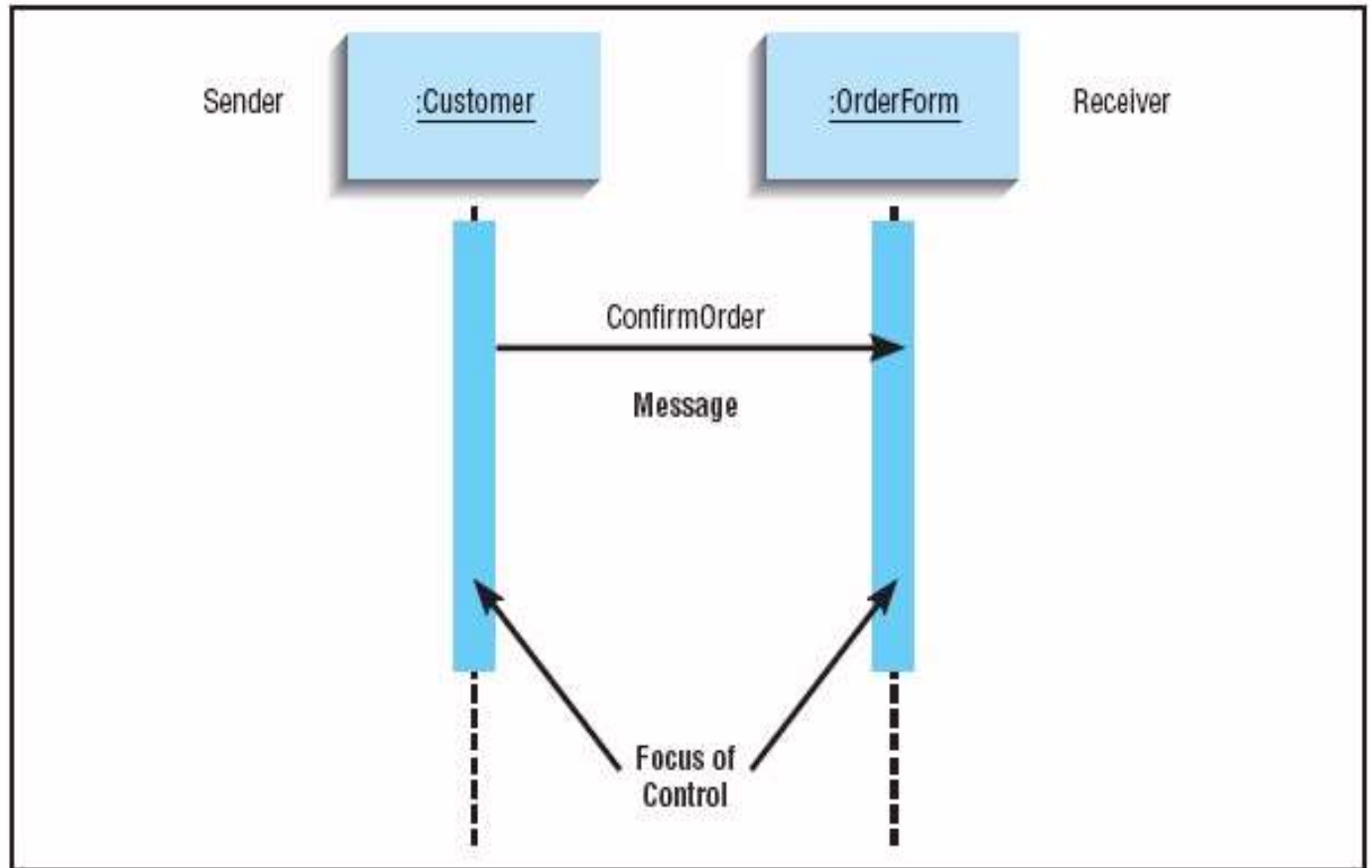
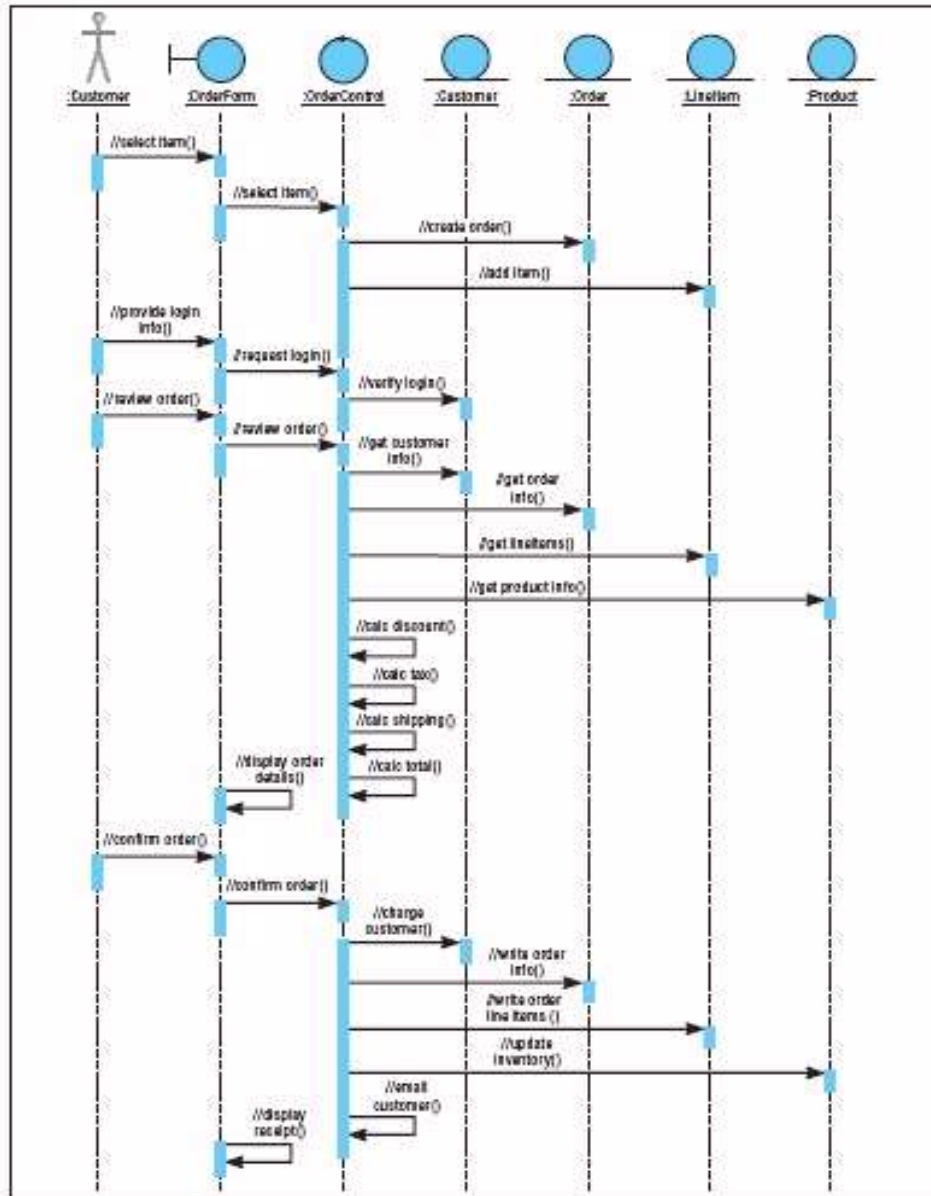


Figure 9.7 Sequence Diagram of the Use Case Place Order

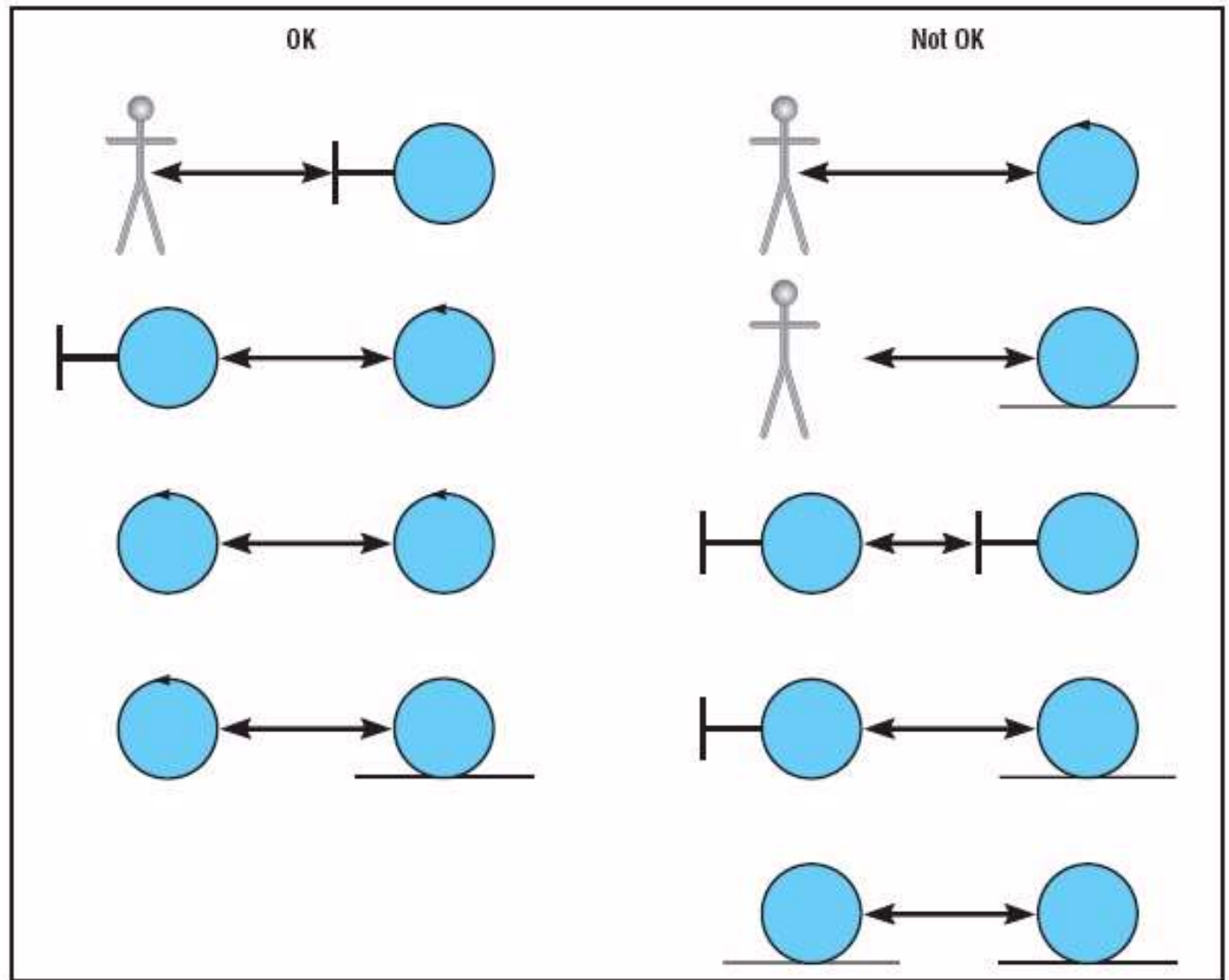


Typically, actors interact with boundary classes, which in turn interact with control classes, which in turn interact with entity classes.

What Is Robustness Analysis?

- Involves analyzing the narrative text of each of the use cases and identifying a first-guess set of the objects into entity, boundary, and control classes
- Requires completeness checks and adherence to diagramming rules

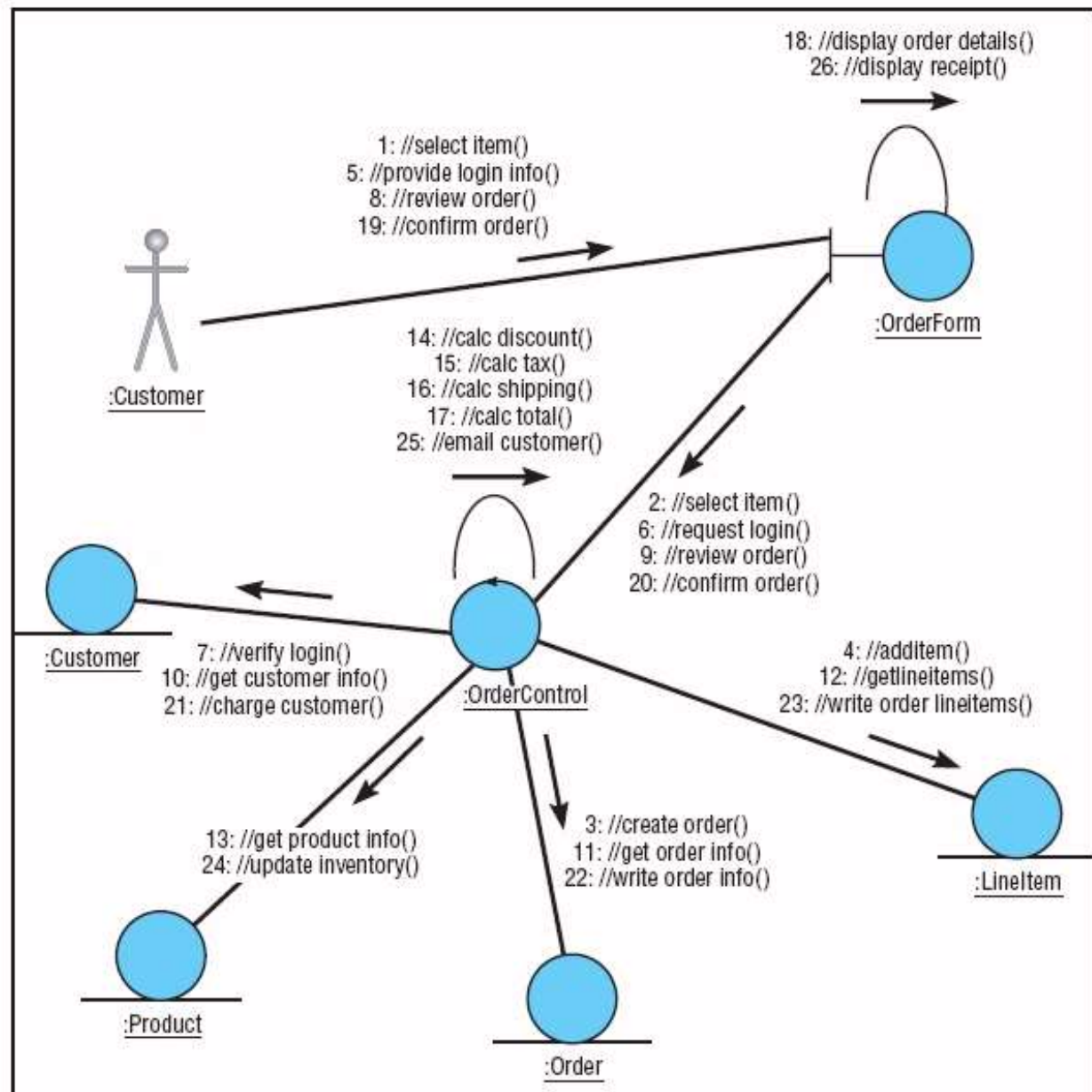
Figure 9.9
Sequence Diagram
Rules



What Is a Communication Diagram?

- A UML diagram that shows the interactions between objects to perform critical pieces of the use case behavior
- Unlike sequence diagrams, communication diagrams have no spatial representation of time; sequences of messages are shown by numbering.

Figure 9.10
Communication
Diagram of the
WebStore Case



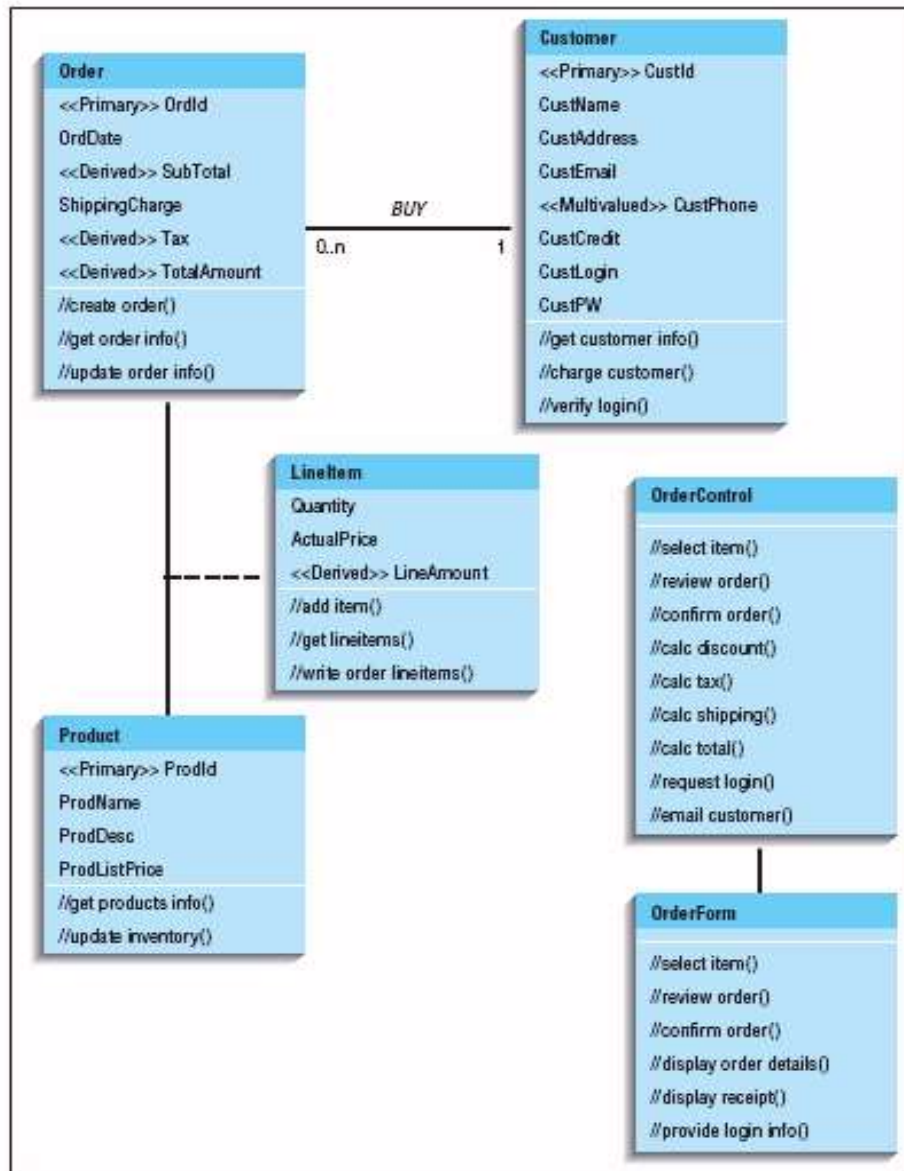
Drawing Analysis Class Diagrams

- Analysis classes are completed by including operations, which fulfill the responsibilities taken by the analysis class.
- In addition, attributes and relationships from the conceptual data model are retained and refined.

Figure 9.11
Analysis Class
Diagram Showing
Structure and
Behavior of Classes

Operations (behaviors) are depicted in the lower third of a class box.

Operations fulfill the class's responsibilities.



EXAMPLE

- The UNIVERSITY OF KARACHI registration system is briefly described thus:
- You have been asked to streamline, improve, and automate the process of assigning professors to courses and the registration of students such that it takes advantage of prevailing web technologies for on-line real time, location independent access.
- The process begins by professors deciding on which courses they will teach for the semester. The Registrar's office then enters the information into the computer system, allocating times, room, and student population restrictions to each course. A batch report is then printed for the professors to indicate which courses they will teach. A course catalogue is also printed for distribution to students.

EXAMPLE

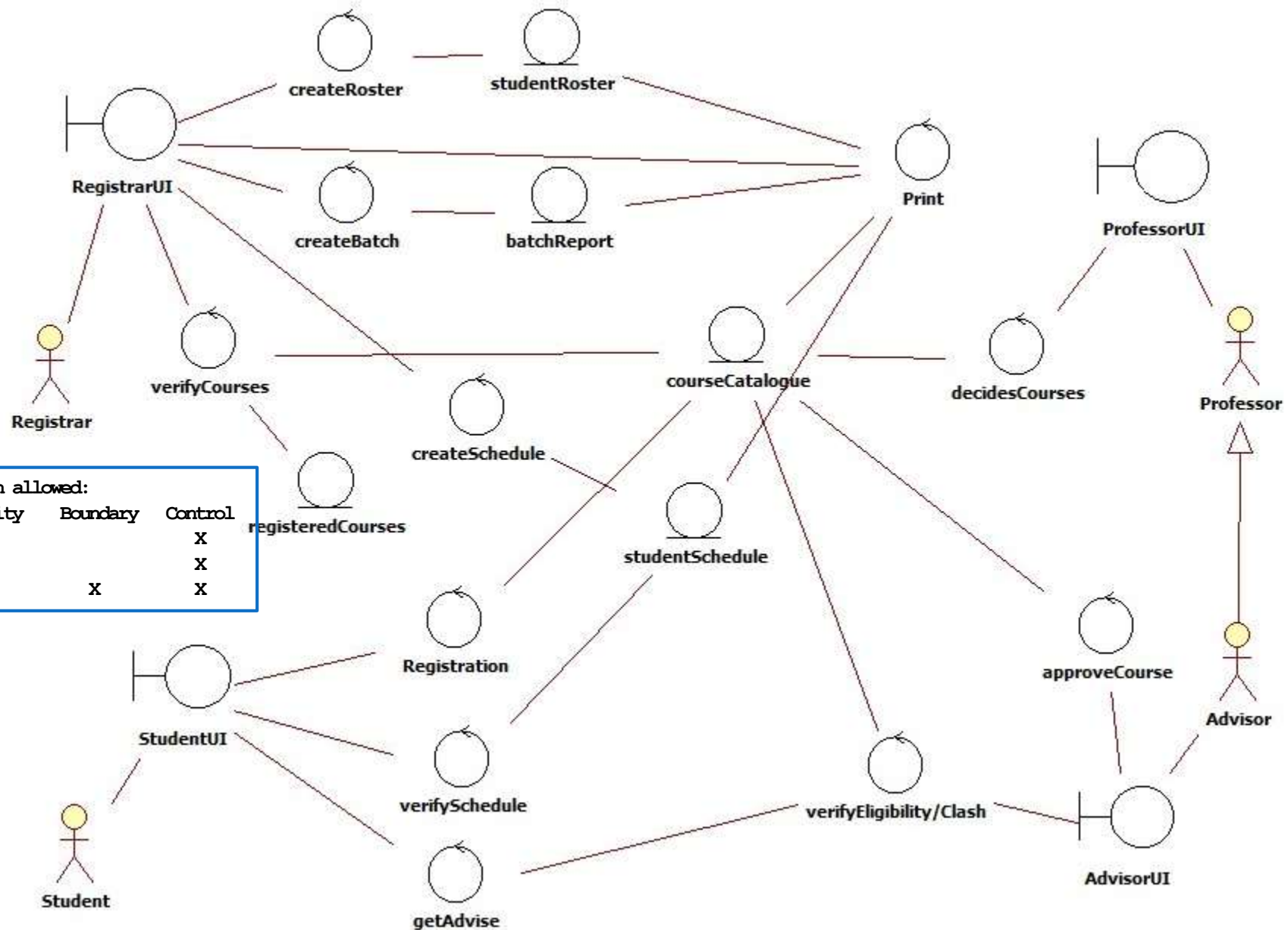
Students then select what courses they desire to take and indicate these by completing paper-based course advising forms. They then meet with an academic advisor who verifies their eligibility to take the selected courses, that the sections of the courses selected are still open, and that the schedule of selected courses does not clash. The typical student load is four courses. The advisor then approves the courses and completed the course registration forms for the student. These are then sent to the registrar who keys them into the registration system – thereby formally registering a student. If courses selected are not approved, the student has to select other courses and complete the course advising forms afresh.

EXAMPLE

Most times students get their first choice, however, in those cases where there is a conflict, the advising office talks with the students to get additional choices. Once all students have been successfully registered, a hard copy of the students' schedule is sent to the students for verification. Most student registrations are processed within a week, but some exceptional cases take up to two weeks to resolve.

Once the initial registration period is over, professors receive a student roster for each class they are scheduled to teach.

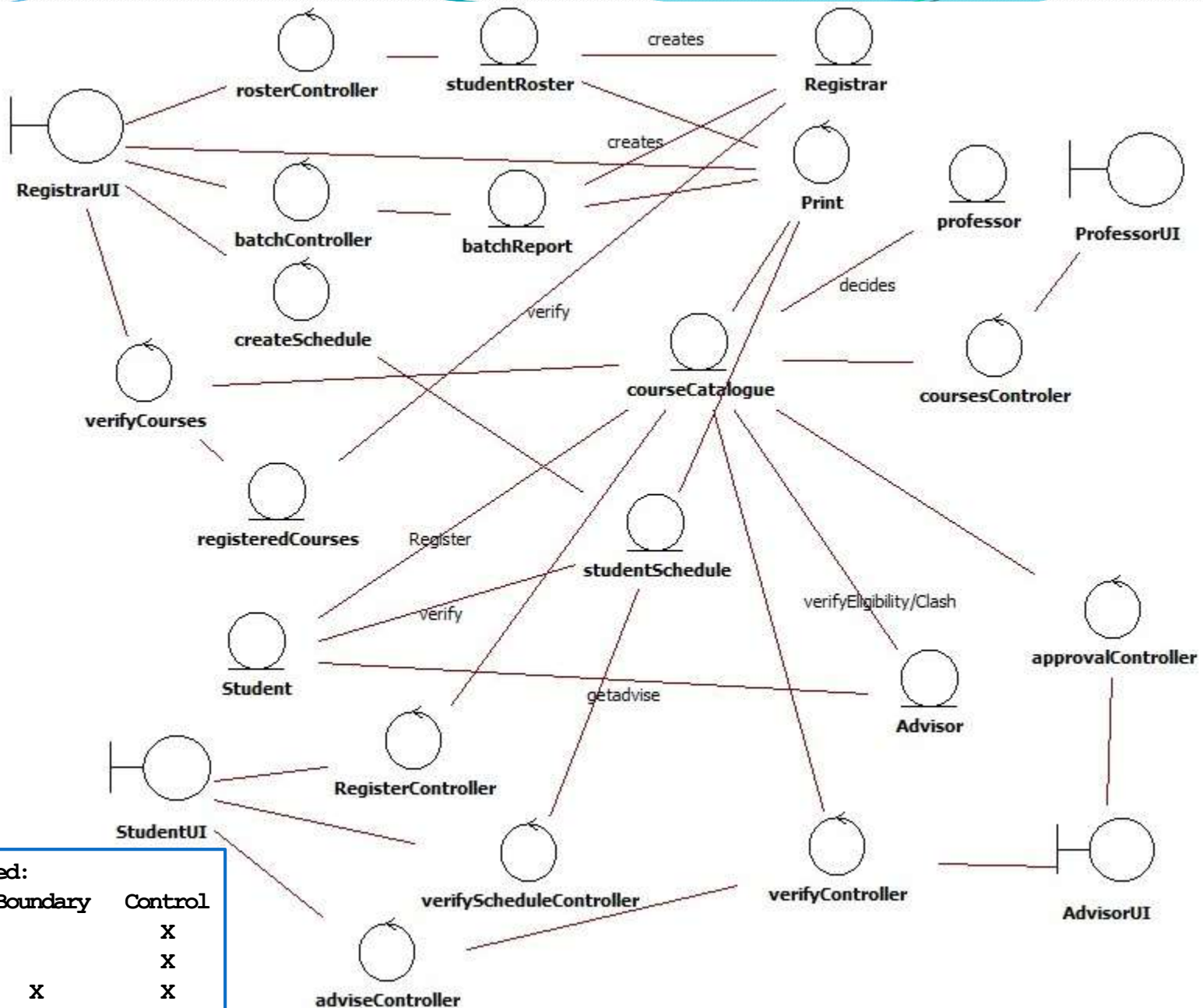
Robustness Diagram – ECB Pattern



Communication allowed:

| | Entity | Boundary | Control |
|----------|--------|----------|---------|
| Entity | | | X |
| Boundary | | | X |
| Control | X | X | X |

Robustness Diagram – ECB Pattern



Here, actors (As Entity classes) can communicate with other entities

Communication allowed:

| | Entity | Boundary | Control |
|----------|--------|----------|---------|
| Entity | X | | X |
| Boundary | | X | X |
| Control | X | X | X |