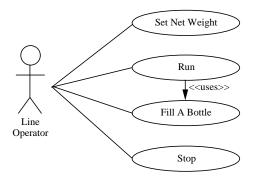# Interaction Diagrams

- Basic Concepts

- A Small Case Study

- Collaboration Diagram

- Case Study Collaborations

- Sequence Diagram

- Case Study Sequence Diagram

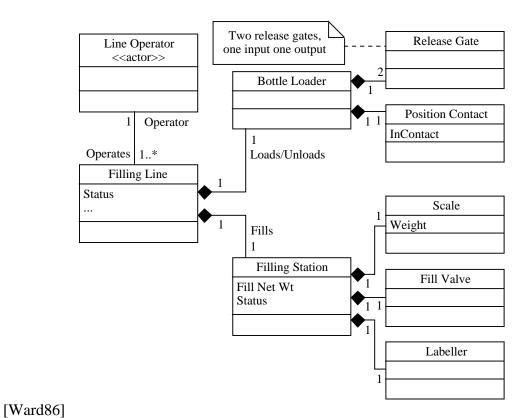- Comparing and Contrasting: Collaboration and Sequence

Basic Concepts

• We are now starting to take a look at system dynamics

• This will be a mid-level view
   - We will look at interactions between (classes of) objects
   - We will not look inside the (classes of) objects themselves

• Generally speaking, there should be one interaction diagram for every use case

• There are two kinds of interaction diagrams
   - Collaboration Diagrams
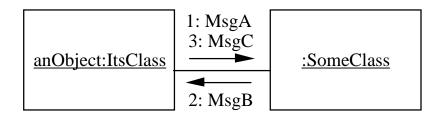   - Sequence Diagrams

A Small Case Study

- Use Cases

- Class Model

[Ward86]

Paul Ward and Steve Mellor, Structured Development for Real-time Systems, Volume 3, Yourdon Press, 1986

Collaboration Diagram

- Collaboration diagrams are centered on objects
    - Use UML's `UnderlinedName` convention to emphasize
    - Objects are named `<an object name>:<its class>`
    - Either `<an object name>` or `<a class name>` can be omitted
        * If `<an object name>` is omitted, use `:<a class name>`

- A Collaboration is a set of objects that communicate
    - Collaborations are shown by lines between objects

- That communication is made up of messages
    - Messages are shown as labelled arrows
    - Numbers show sequence

- Notation

```
┌──────────────────┐  1: MsgA   ┌──────────────────┐
│                  │  3: MsgC   │                  │
│                  │  ───────►  │                  │
│ anObject:ItsClass│            │   :SomeClass     │
│                  │  ◄───────  │                  │
│                  │  2: MsgB   │                  │
└──────────────────┘            └──────────────────┘
```
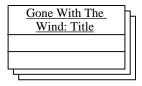
- In some cases it can be better to use sequence numbers of the form `x.y.z`
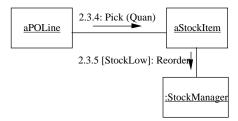
- For example `3.2.5, 3.2.6,` ...

Collaboration Diagram (cont)
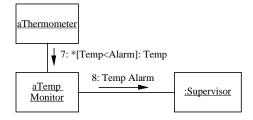
- Multiobject
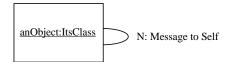
- Conditional messages
  - Conditional messages mean that under certain conditions the message will be sent and under other conditions it won't
  - The message is sent when the condition in the square brackets is true

- Repetitive messages
  - The message repeats while the condition in the square brackets is true
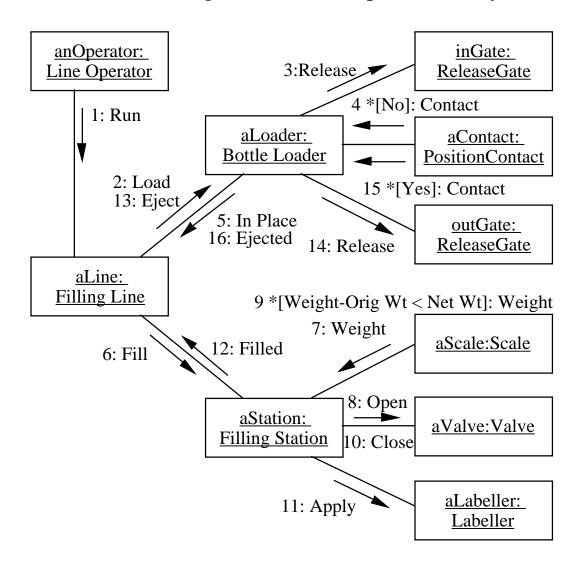
- Objects will sometimes send messages to themselves

anObject:ItsClass   N: Message to Self

Case Study Collaborations

• A collaboration diagram for the sample case study
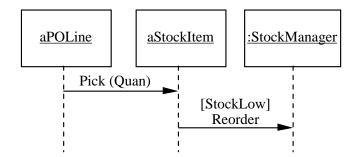
Sequence Diagram

• Sequence diagrams are also centered on objects
 - Objects are named the same way

• A Lifeline shows the time-ordered history of the interaction
 - Lifelines run either vertically or horizontally
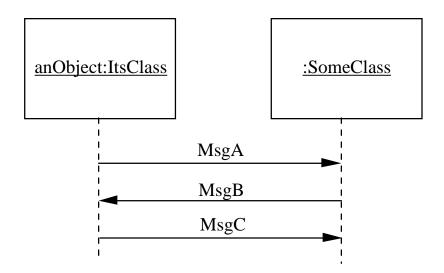 - The usual style is vertical

• Messages are shown in time-order along the object lifelines

• Notation

```
┌─────────────────────┐        ┌─────────────────────┐
│                     │        │                     │
│  anObject:ItsClass  │        │     :SomeClass      │
│                     │        │                     │
└──────────┬──────────┘        └──────────┬──────────┘
           ┆                              ┆
           │            MsgA              │
           ├─────────────────────────────▶│
           │            MsgB              │
           │◀─────────────────────────────┤
           │            MsgC              │
           ├─────────────────────────────▶│
           ┆                              ┆
```
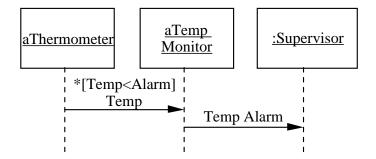
Sequence Diagram (cont)

- Conditional messages

```
┌──────────┐   ┌──────────┐   ┌──────────────┐
│ aPOLine  │   │aStockItem│   │:StockManager │
└──────────┘   └──────────┘   └──────────────┘
     ┆    Pick (Quan)  ┆              ┆
     ┆───────────────▶ ┆              ┆
     ┆             ┆  [StockLow]      ┆
     ┆             ┆   Reorder        ┆
     ┆             ┆────────────────▶ ┆
     ┆             ┆              ┆
```

- Repetitive messages

```
┌──────────────┐   ┌──────────┐   ┌──────────────┐
│aThermometer  │   │  aTemp   │   │ :Supervisor  │
│              │   │ Monitor  │   │              │
└──────────────┘   └──────────┘   └──────────────┘
      ┆  *[Temp<Alarm]  ┆              ┆
      ┆      Temp       ┆              ┆
      ┆──────────────▶  ┆              ┆
      ┆             ┆  Temp Alarm      ┆
      ┆             ┆────────────────▶ ┆
      ┆             ┆              ┆
```

- Messages to self
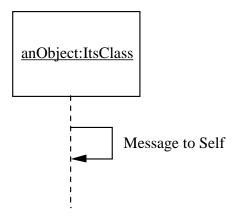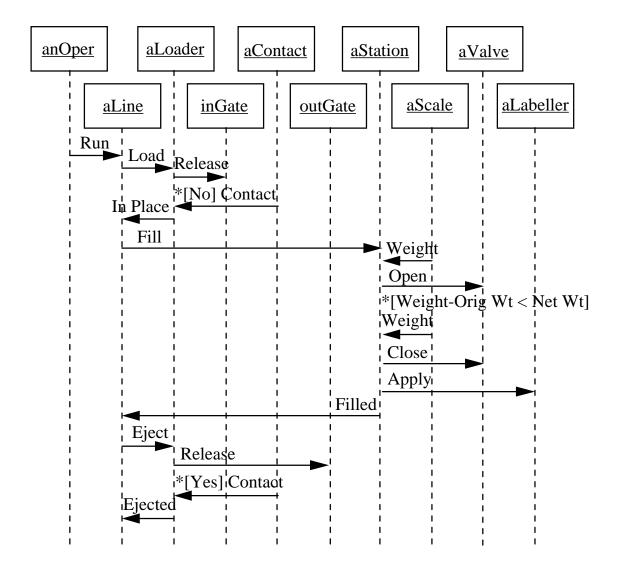
```
┌────────────────────┐
│ anObject:ItsClass  │
└────────────────────┘
          ┆
          ┆ ┌──────┐
          ┆ │      │  Message to Self
          ◀─┘      │
          ┆
```

Case Study Sequence Diagram

• The sequence diagram for the sample case study

anOper	aLoader	aContact	aStation	aValve

aLine	inGate	outGate	aScale	aLabeller

Run
Load
Release
*[No] Contact
In Place
Fill
Weight
Open
*[Weight-Orig Wt < Net Wt]
Weight
Close
Apply
Filled
Eject
Release
*[Yes] Contact
Ejected

• Some modelers prefer to include the use case descriptive text along the time axis of a sequence diagram
   - This can help correlate the use case to the interaction

Comparing and Contrasting: Collaboration and Sequence

- Both diagrams show the same information
    - Objects/classes
    - Messages
    - Sequence
    - Conditional
    - Repetition
    - Messages to self

- Collaboration diagrams emphasize who-is-talking-to-who
    - But the time-ordering of the messages gets obscured

- Sequence diagrams emphasize time-ordering
    - But the who-is-talking-to-who gets obscured

- Use the diagram that you are most comfortable with
    - A good CASE tool really ought to be able to generate one given the other

Key Points

• We are now looking at system dynamics

• This is a mid-level view
  - We looked at interactions between (classes of) objects
  - We did not look inside the (classes of) objects
    themselves

• Both diagrams show the same information
  - Objects/classes
  - Messages
  - Sequence
  - Conditional
  - Repetition
  - Messages to self

• Collaboration diagrams emphasize who-is-talking-to-who
  - But the time-ordering of the messages gets obscured

• Sequence diagrams emphasize time-ordering
  - But the who-is-talking-to-who gets obscured