

# UML

---

## A Selected Overview of the **Unified Modeling Language**

Brought to you by:

Ray Buse



# What is UML?

---

- ❑ Standardized Graphical Notation
- ❑ Object Modeling
- ❑ Created by Object Management Group [omg.org] in 1997
- ❑ Current Specification: 2.0

# UML Topics

---

- Modeling a System
  - What aspects would we like to model?
- Types of Diagrams
  - What things does UML model?
- UML Diagram Constructs
  - How does UML work?
- Working with UML
  - How should UML [not] be used?

# Modeling a System

---

- ❑ **Object Model**

Showcases the structure and substructure of the system using objects, attributes, operations, and associations.

- ❑ **Functional Model**

Showcases the functionality of the system from the user's Point of View.

- ❑ **Dynamic Model**

Showcases the internal behavior of the system.

# Types of Diagrams

---

13 types in UML 2.0

## □ **Structure Diagrams**

- emphasize what things must be in the system being modeled

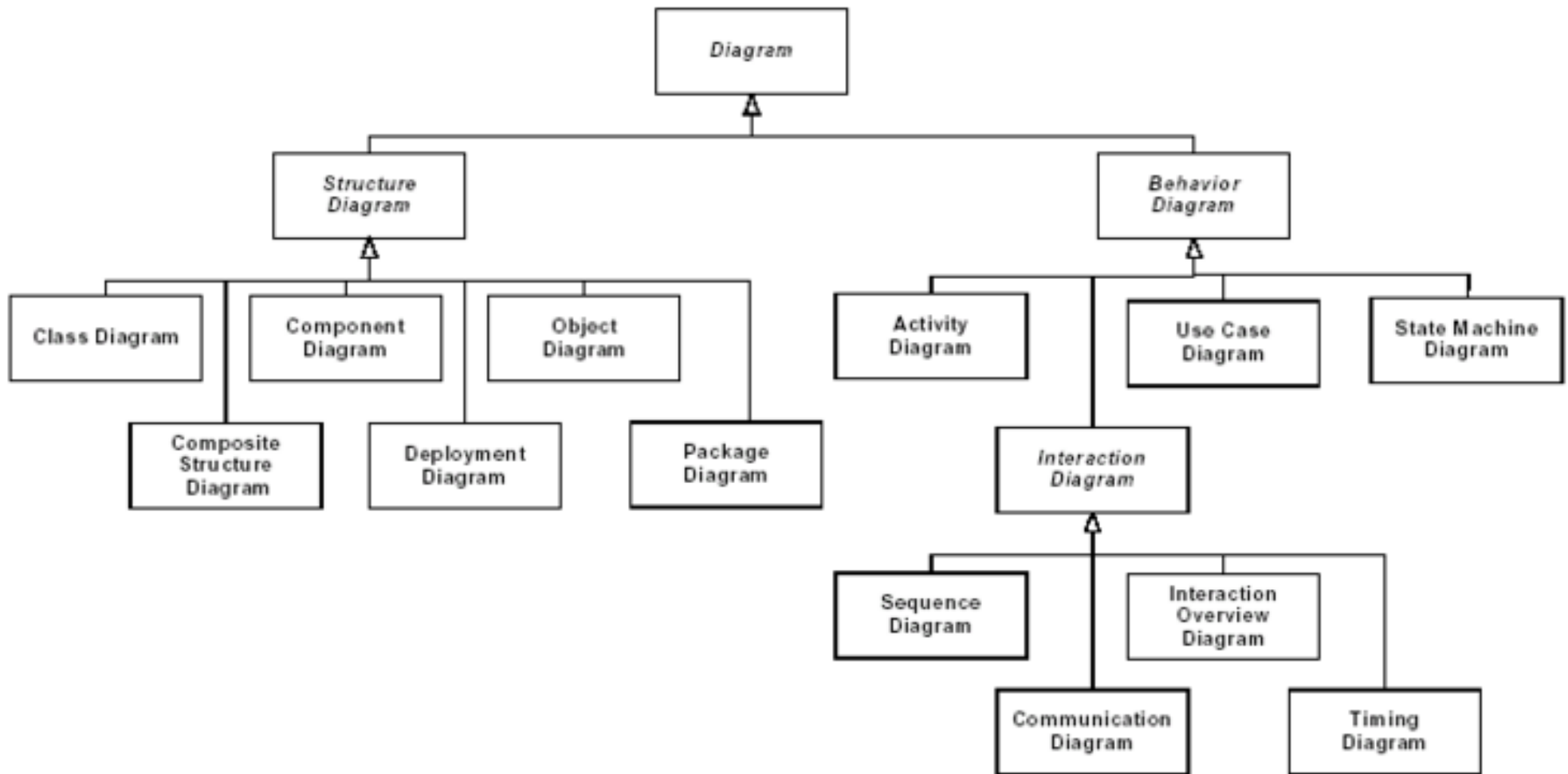
## □ **Behavior Diagrams**

- emphasize what must happen in the system being modeled

## □ **Interaction Diagrams**

- emphasize the flow of control and data among the things in the system being modeled

# Types of Diagrams



# Diagrams to know

---

- Class Diagram
- Activity Diagram
- Sequence Diagram

# UML Class Diagram

---

Static Class Structure

# Class

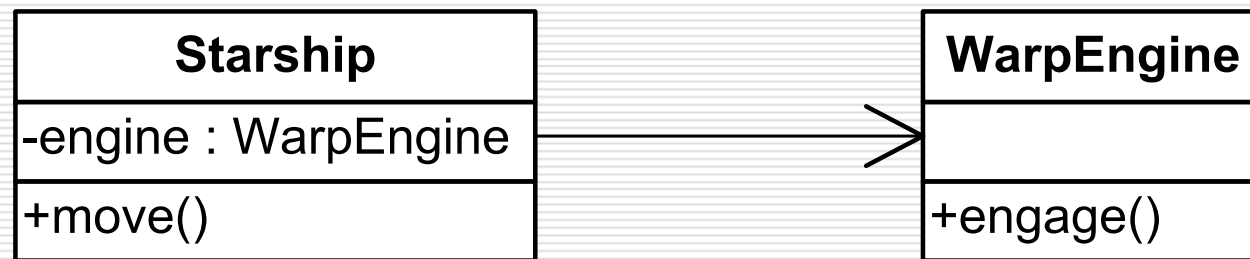
---

Class
-private
+public : int
#protected : object
+method() : return_type

# Associations - Navigability

---

## □ Directed

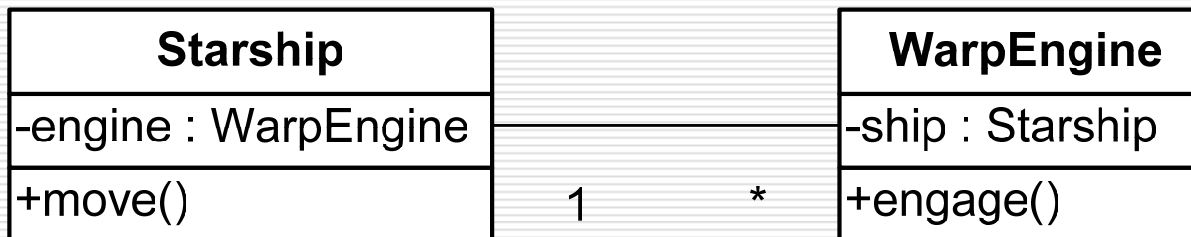
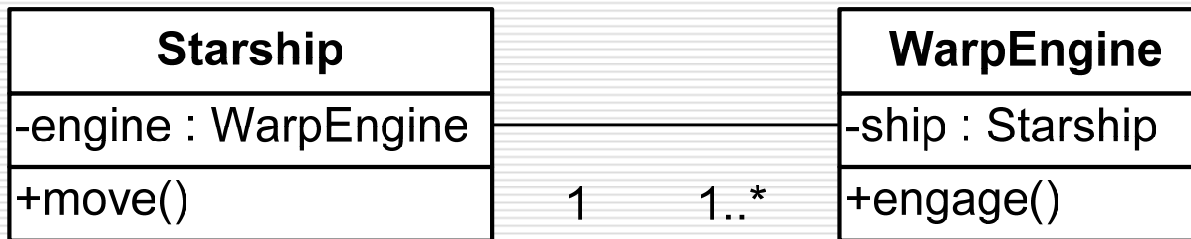
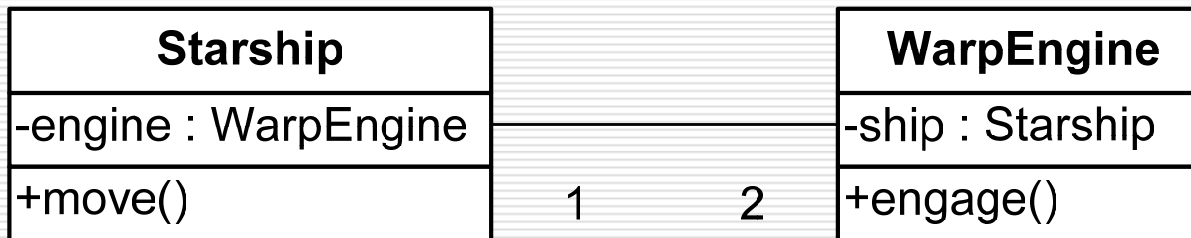


## □ Undirected



# Associations – Multiplicity

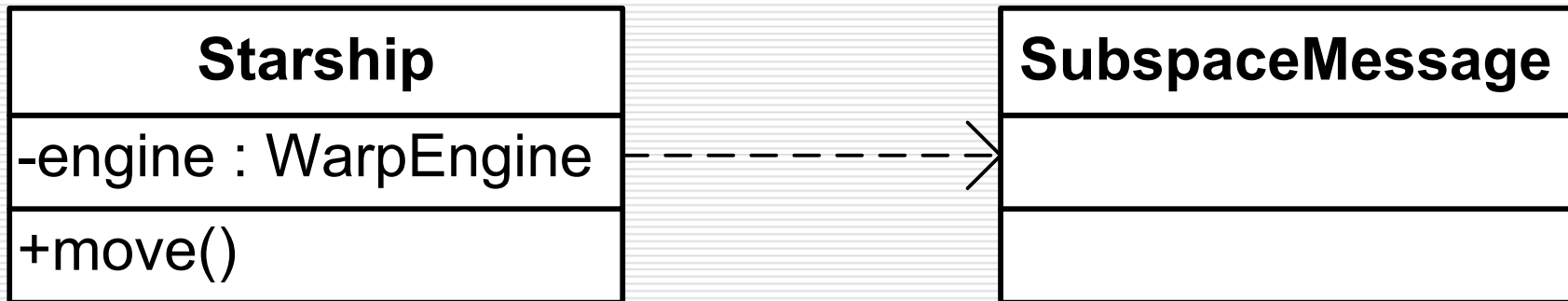
---



# Dependency

---

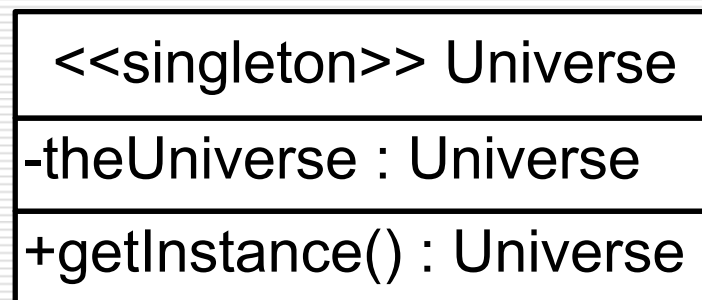
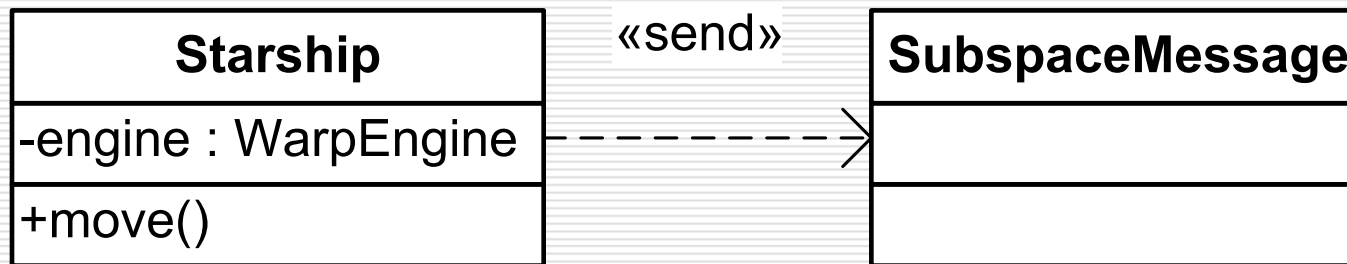
- SubspaceMessage is used by Starship but it is not a field in Starship



# Stereotypes

---

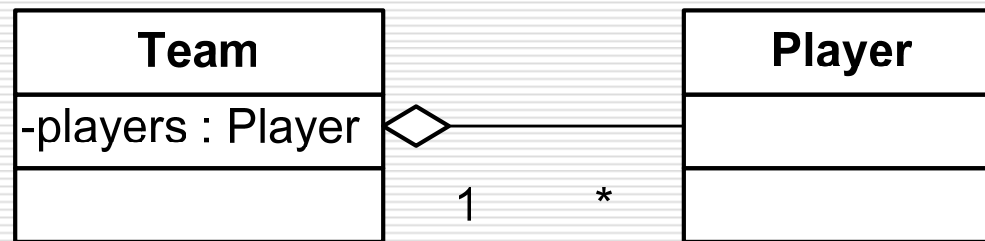
- qualifies the symbol it is attached to



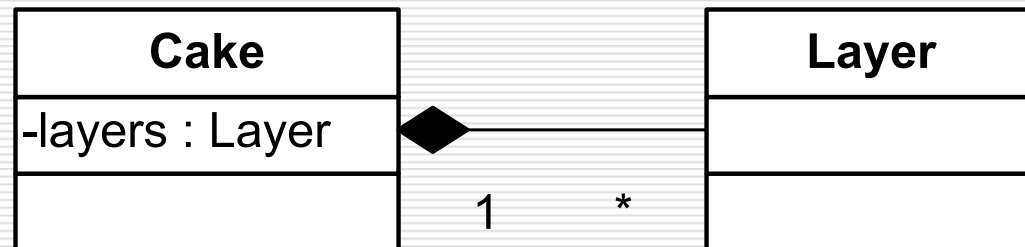
# Aggregation & Composition

---

## □ Aggregation

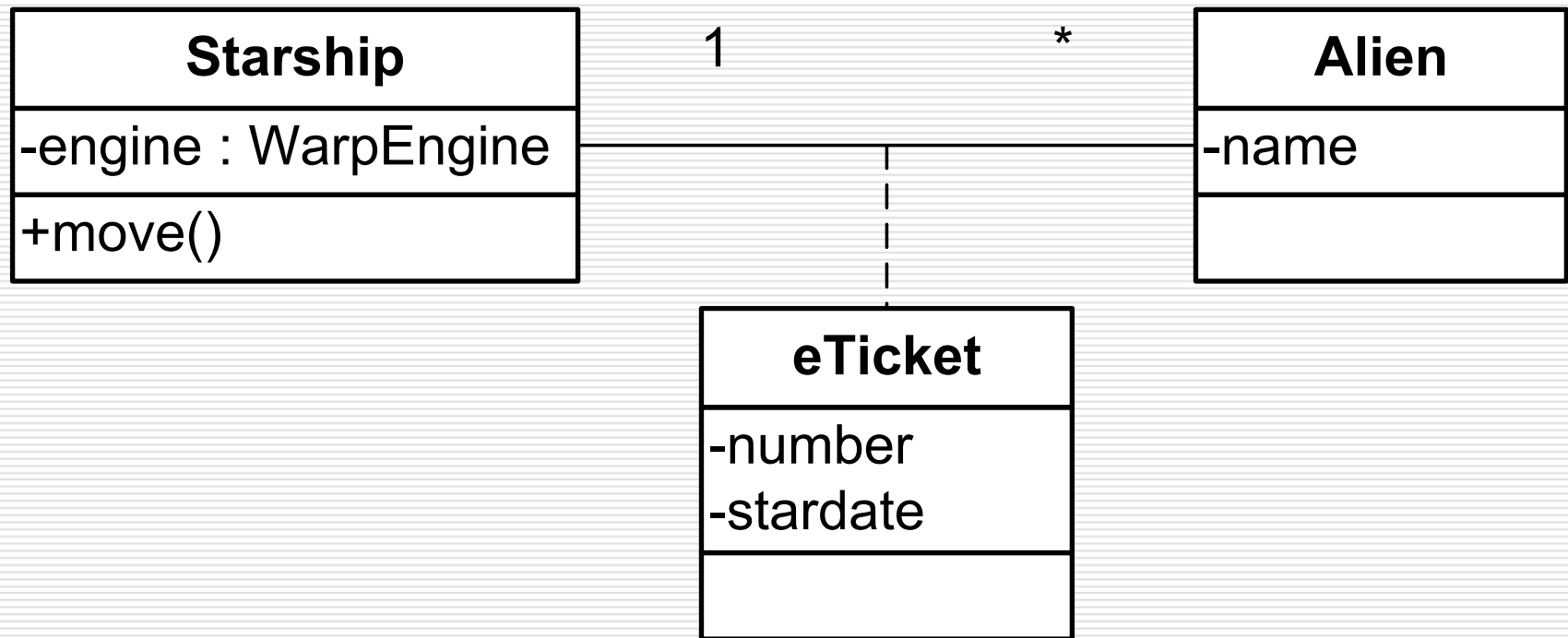


## □ Composition – life cycle connection



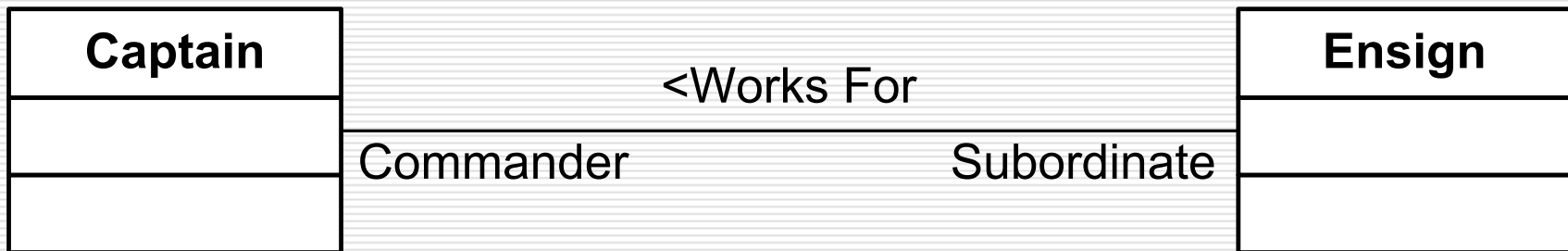
# Association Class

---



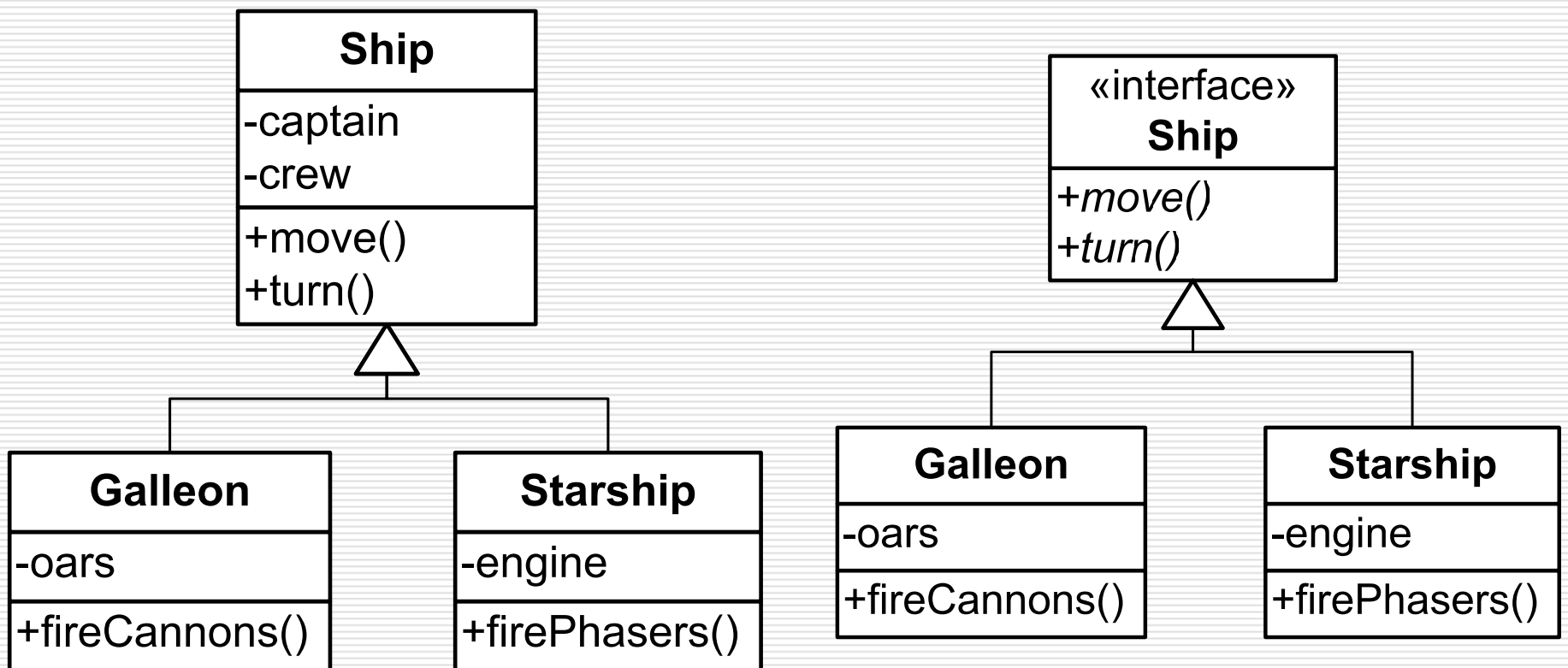
# Association role

---



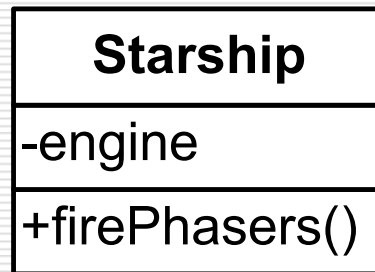
# Generalization (Inheritance)

---



# Notes

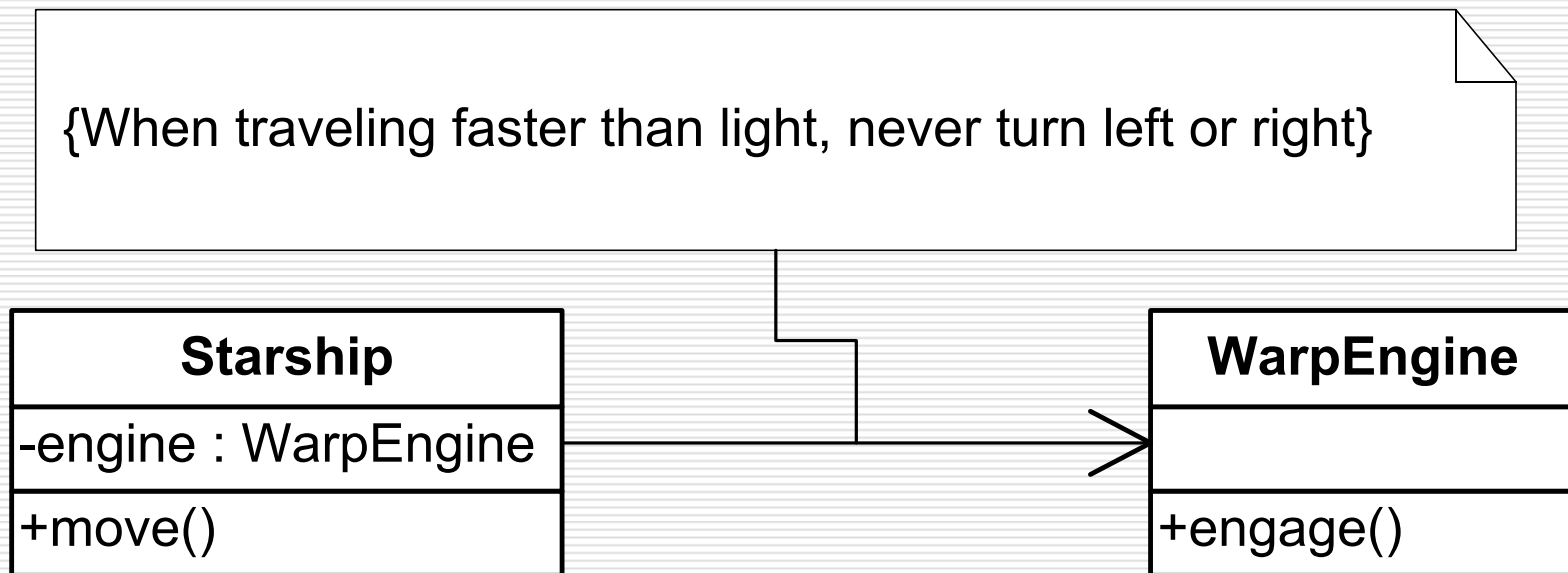
---



You can't fire phasers when at warp speed

# Constraints

---

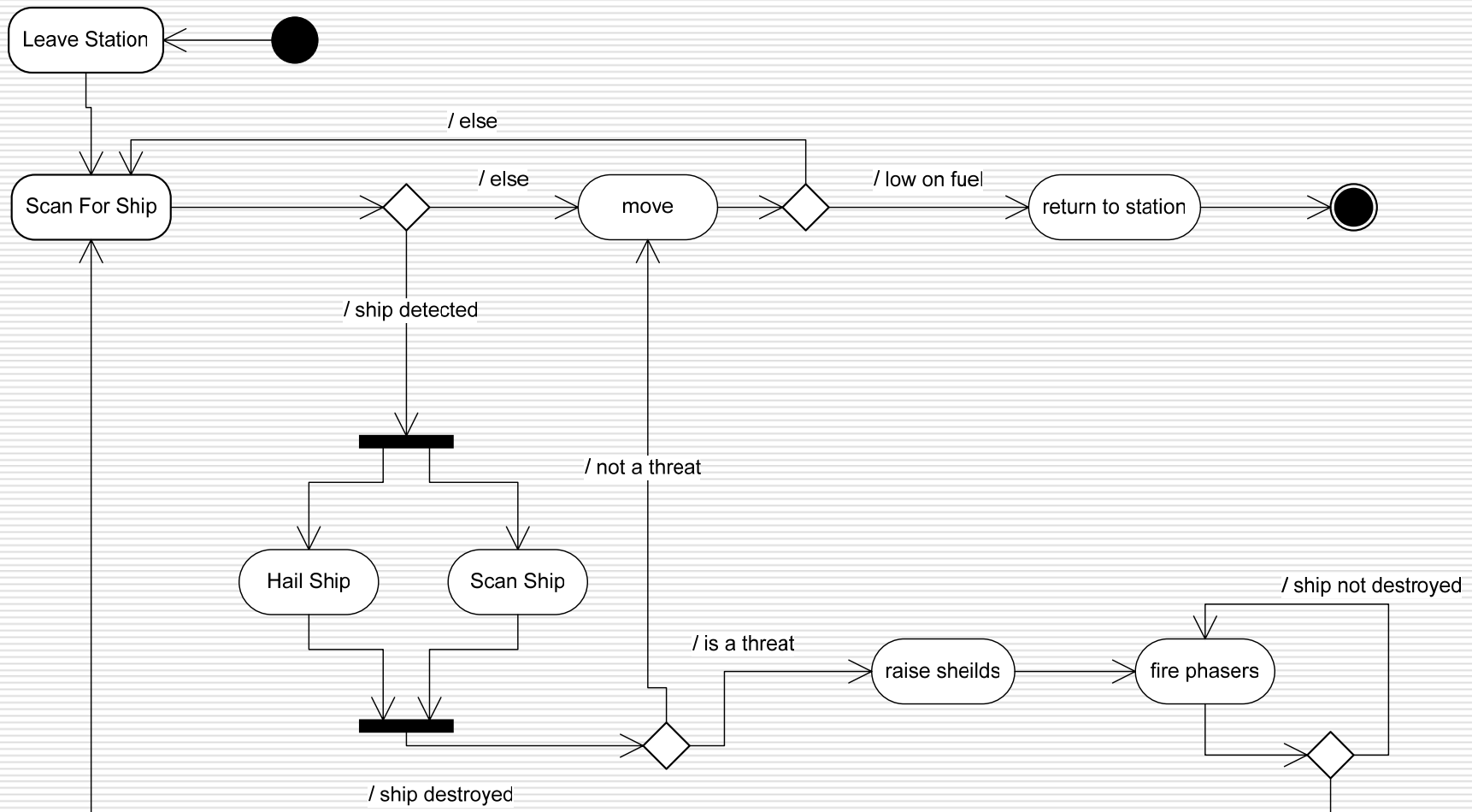


# UML Activity Diagram

---

Operational Workflow  
(State Machine)

# Activity Diagram

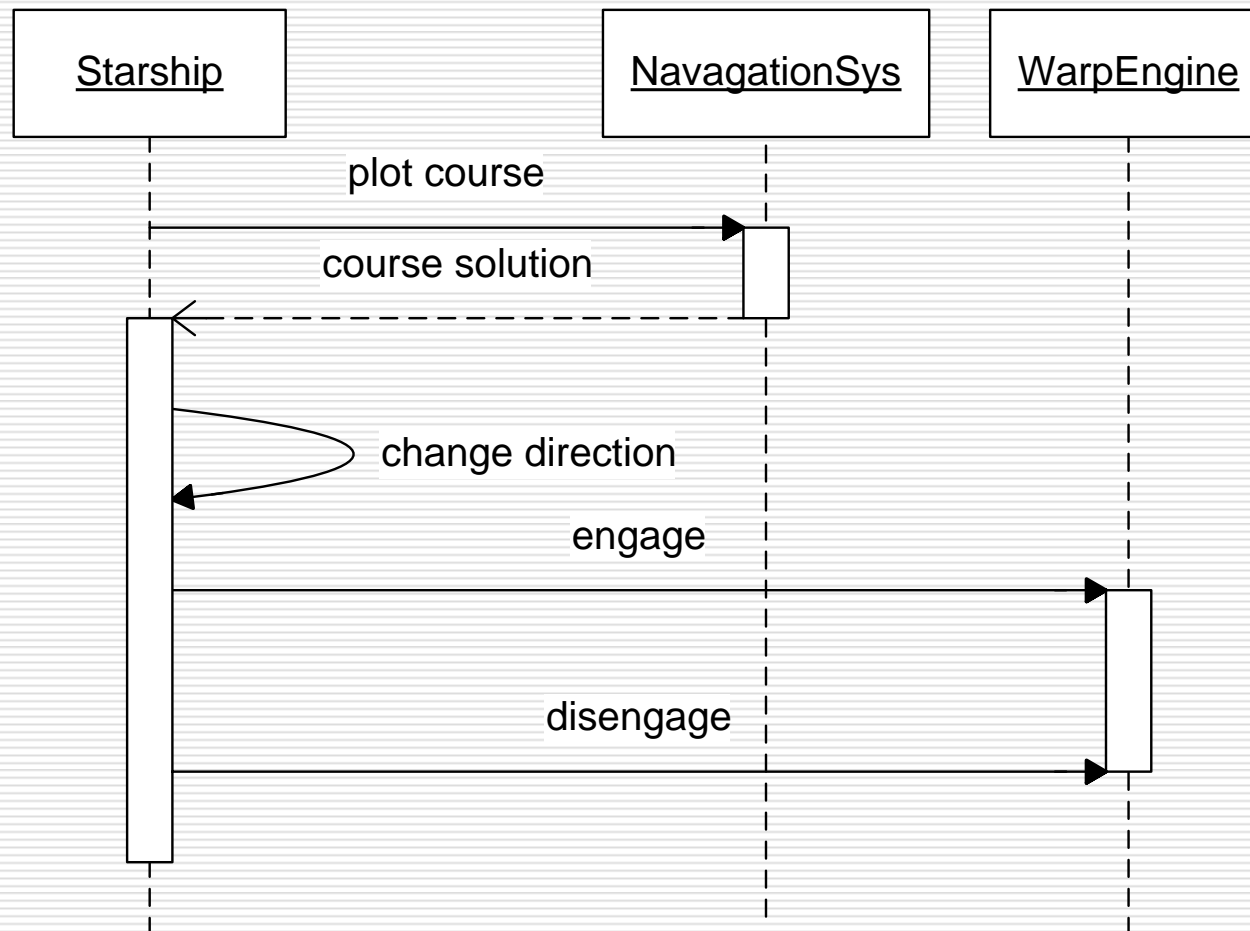


# UML Sequence Diagram

---

Internal Behavior

# Sequence Diagram



# Working with UML

---

Tools

Criticisms

# UML Tools

---

- Code Generation
  - UML -> Code
- Reverse Engineering
  - Code -> UML
- Round Trip Engineering
  - UML -> Code -> UML -> Code ....

# UML Criticisms

---

- Imprecise Semantics
- Doesn't apply well to distributed systems
- Can be over-used

# Imprecise Semantics

---

- ❑ UML is sometimes too flexible
- ❑ leads to subjective interpretation and difficulties in the formal test phases of development
- ❑ Can always use notes / documentation

# Distributed Systems

---

- Hard to model with UML
  - serialization
  - message passing
  - persistence

# Over-Use

---

- Details best captured in source code may be difficult to capture using UML
  - people often try to do so anyway.
- The 80-20 rule
  - a small part of UML is adequate for most of the modeling needs
  - many aspects of UML cater to some specialized or esoteric usages.