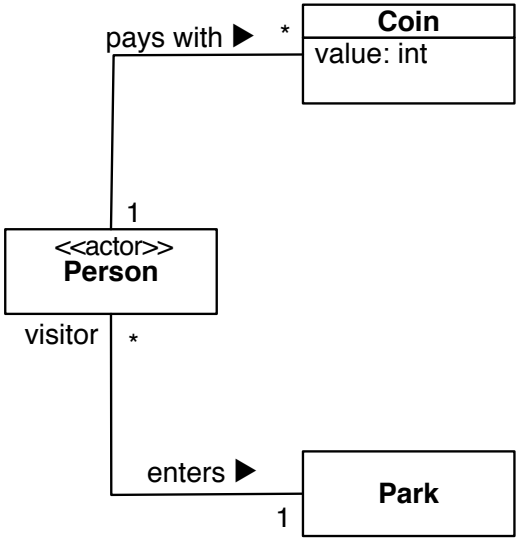
CS445 / ECE451 / CS645 / SE463

Software Requirements Specification & Analysis

Specification Domain Models

Domain Models

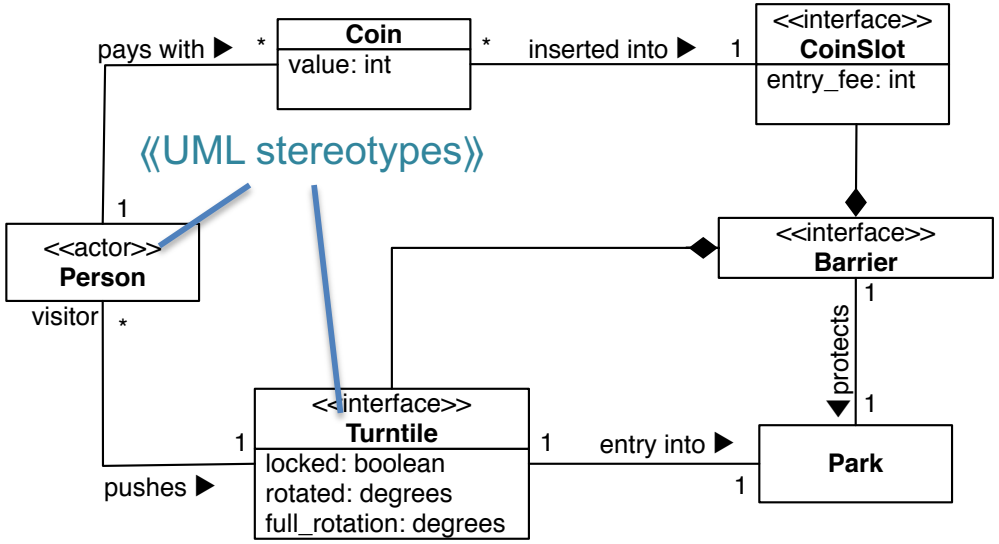
Requirements



Collect $1 from each visitor on entry to the park

Ensure that anyone who has paid can enter the park

Specifications



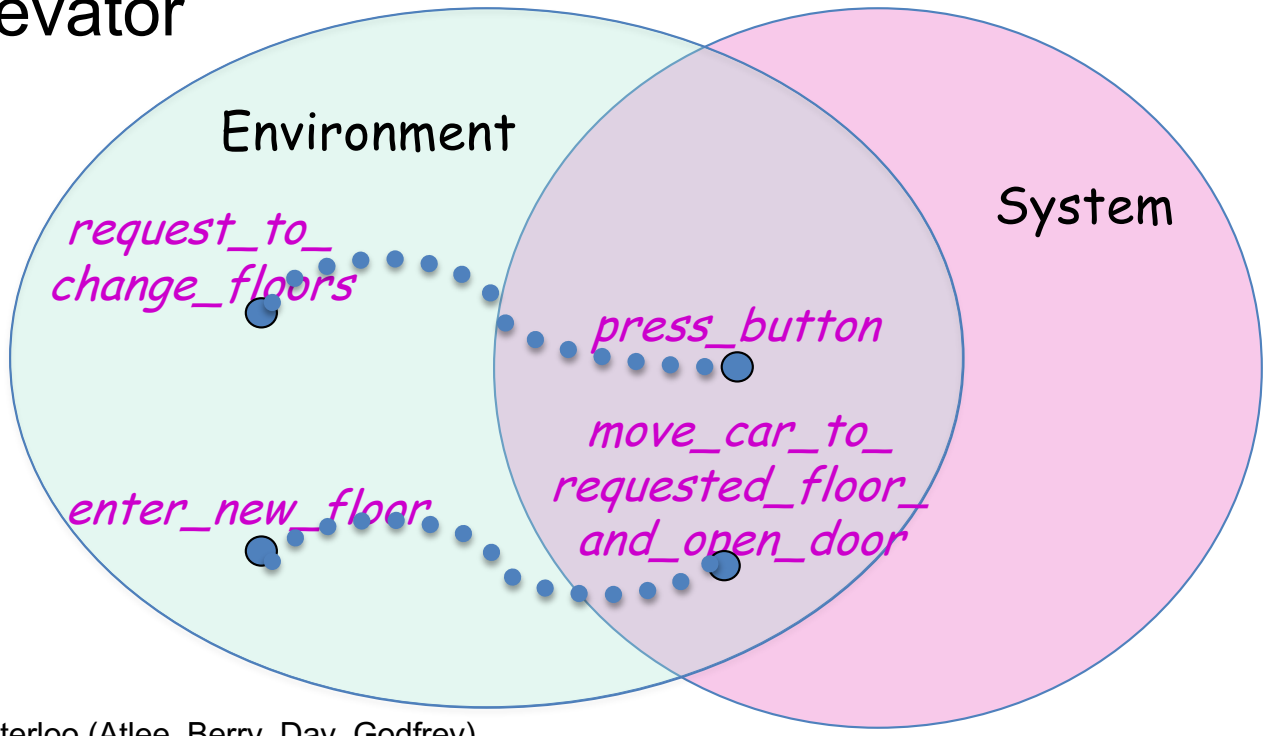
Collect $1 in the CoinSlot for every rotation of the Turnstile

Whenever the CoinSlot receives $1, unlock the Turnstile

Cyber-Physical Systems

Interface entities are likely to be interface devices (sensors and actuators), sensor readings, and actuator commands

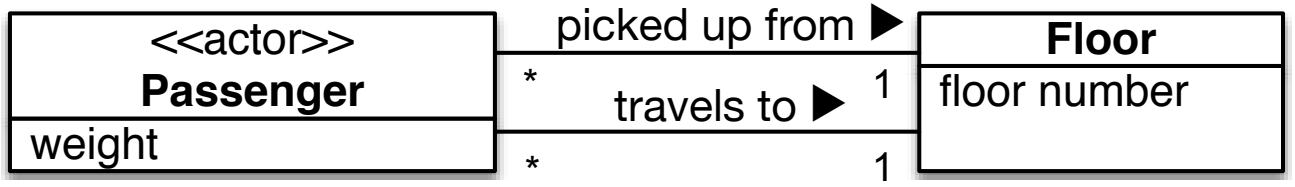
Example: Elevator



Requirements Domain Model

Example:

R: Passenger is transported from her current floor to her desired floor



Interface Entities

Actors

- passengers

Interface objects

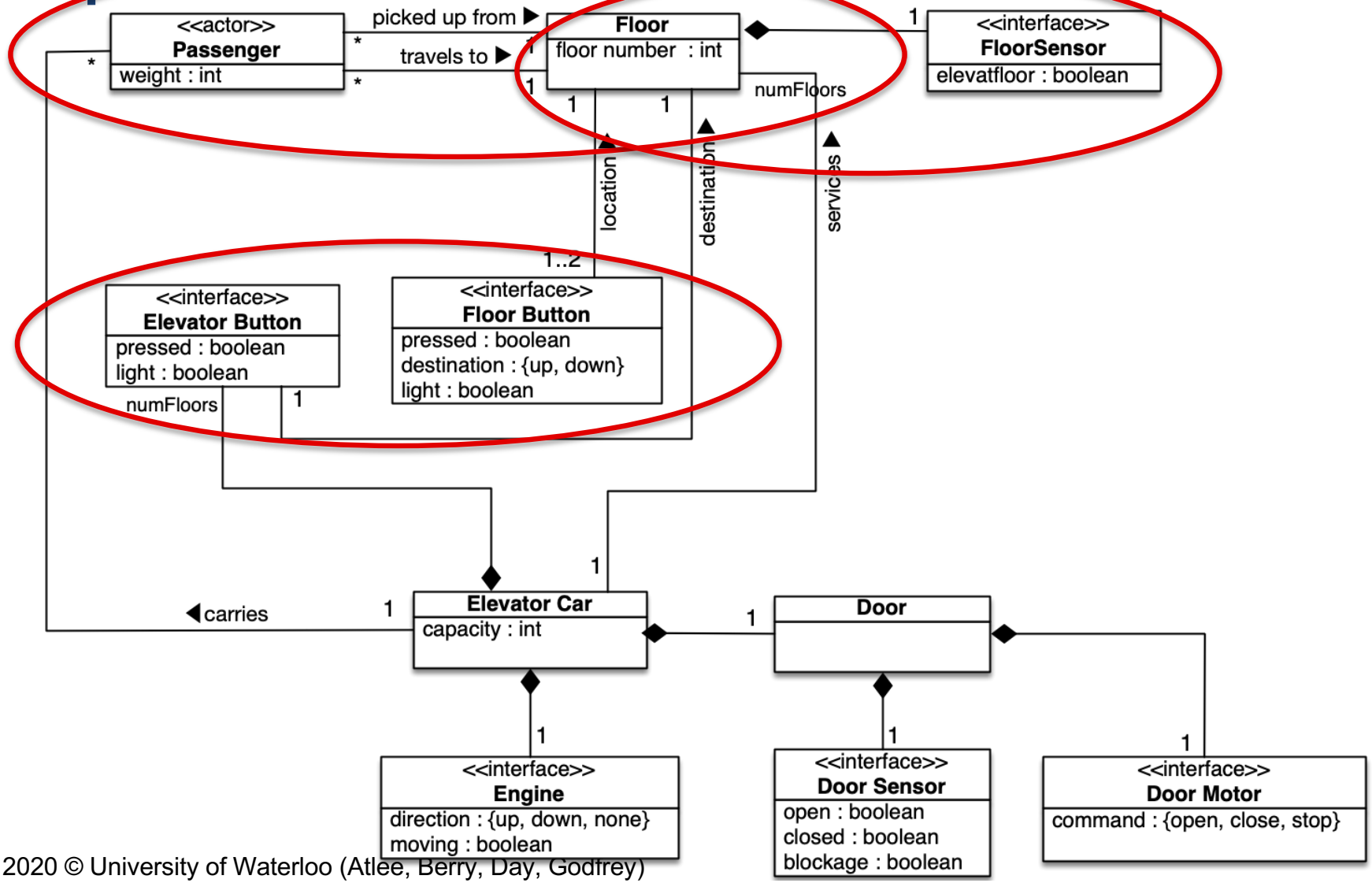
- request buttons
- button lights

Domain objects

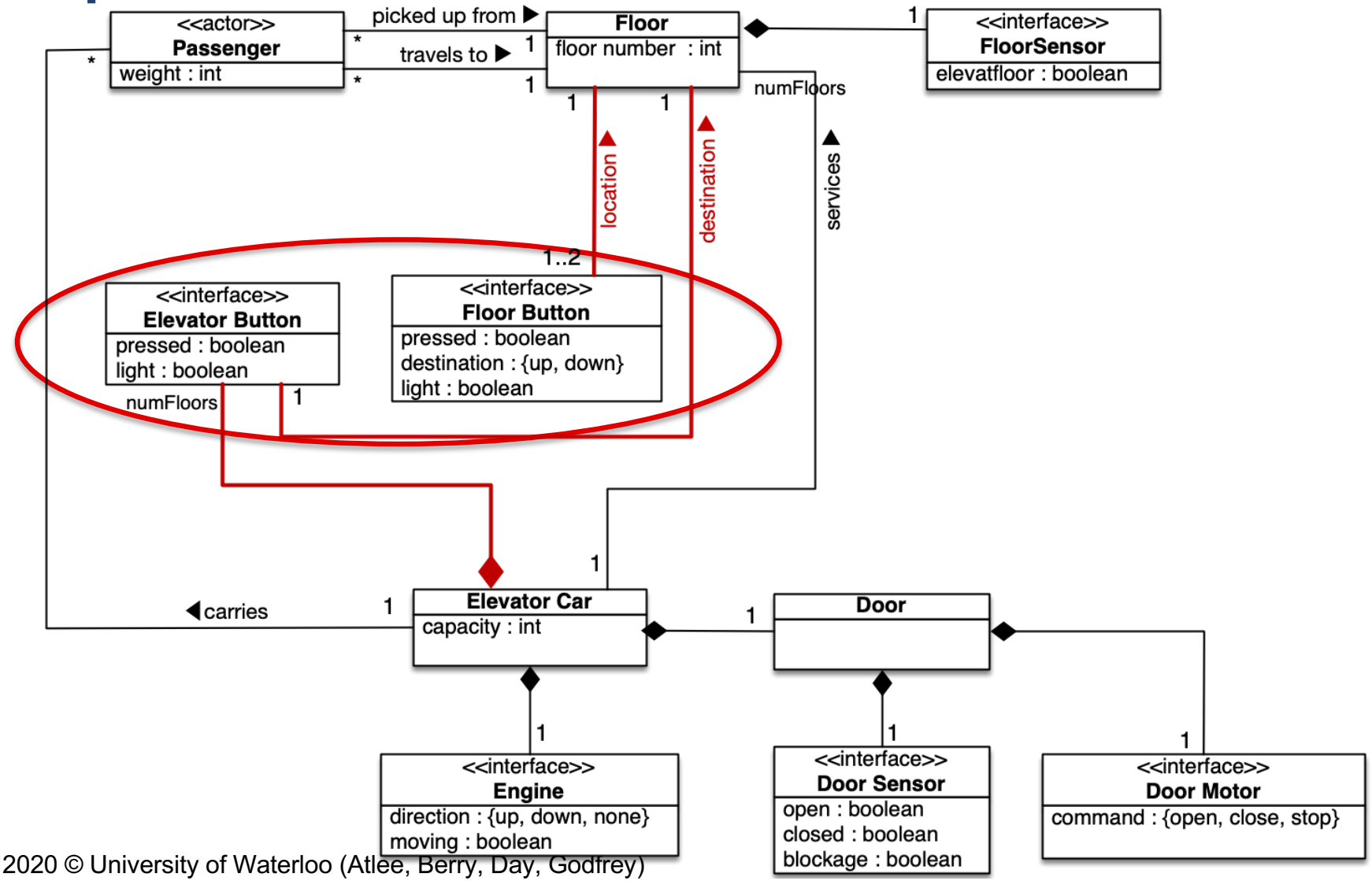
- floors
- elevator car

- floor sensors
- elevator motor
- elevator door motor
- elevator door sensor

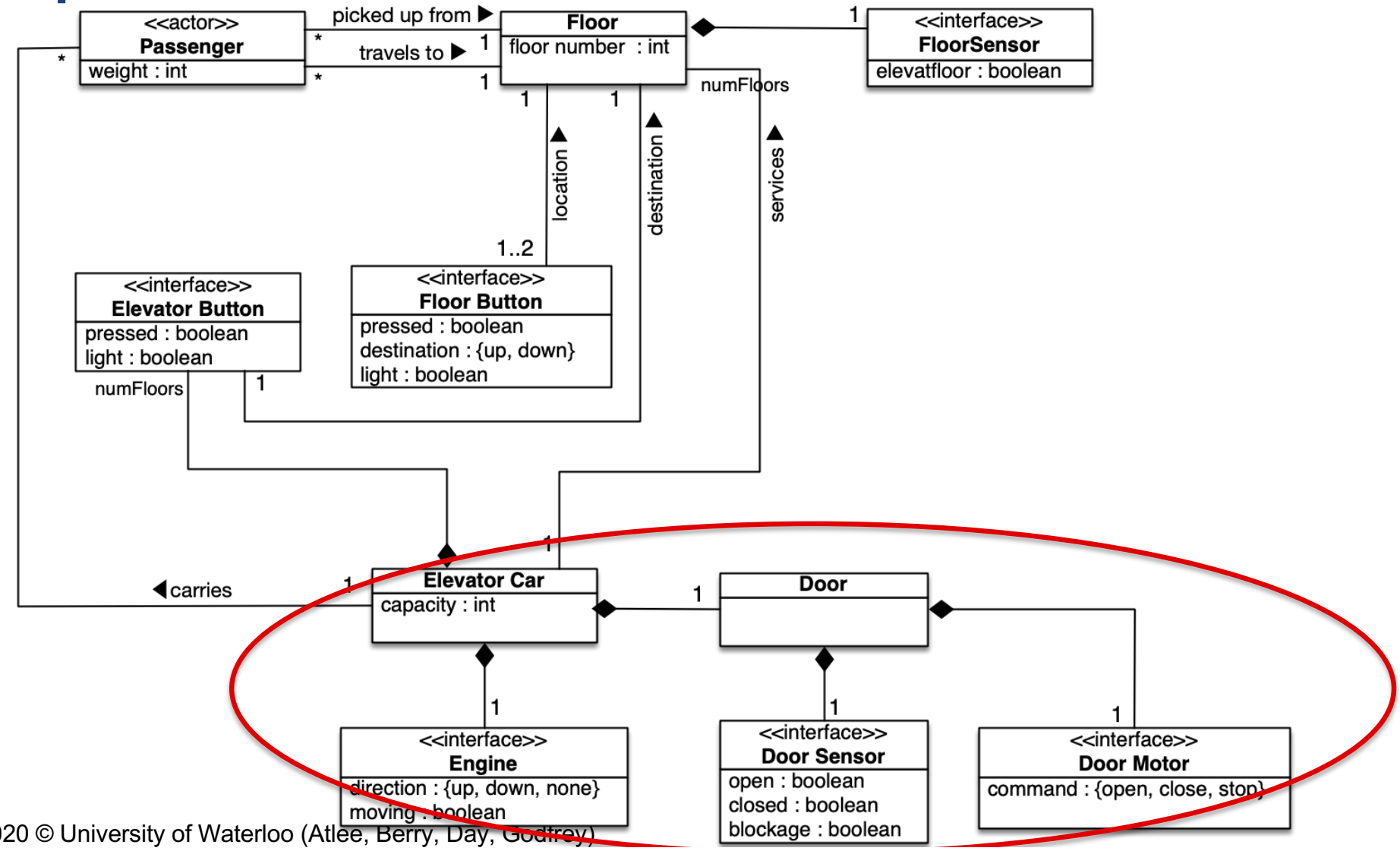
Specification Domain Model



Specification Domain Model



Specification Domain Model

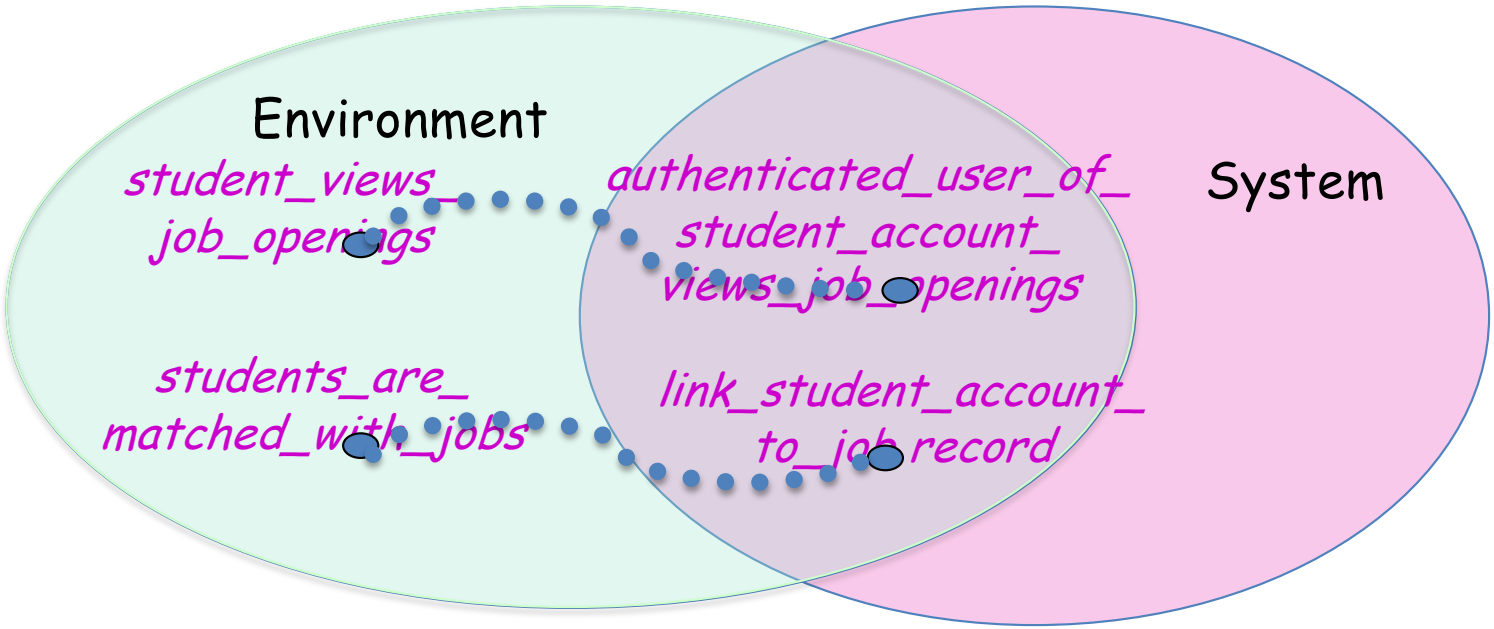


Cyber Systems

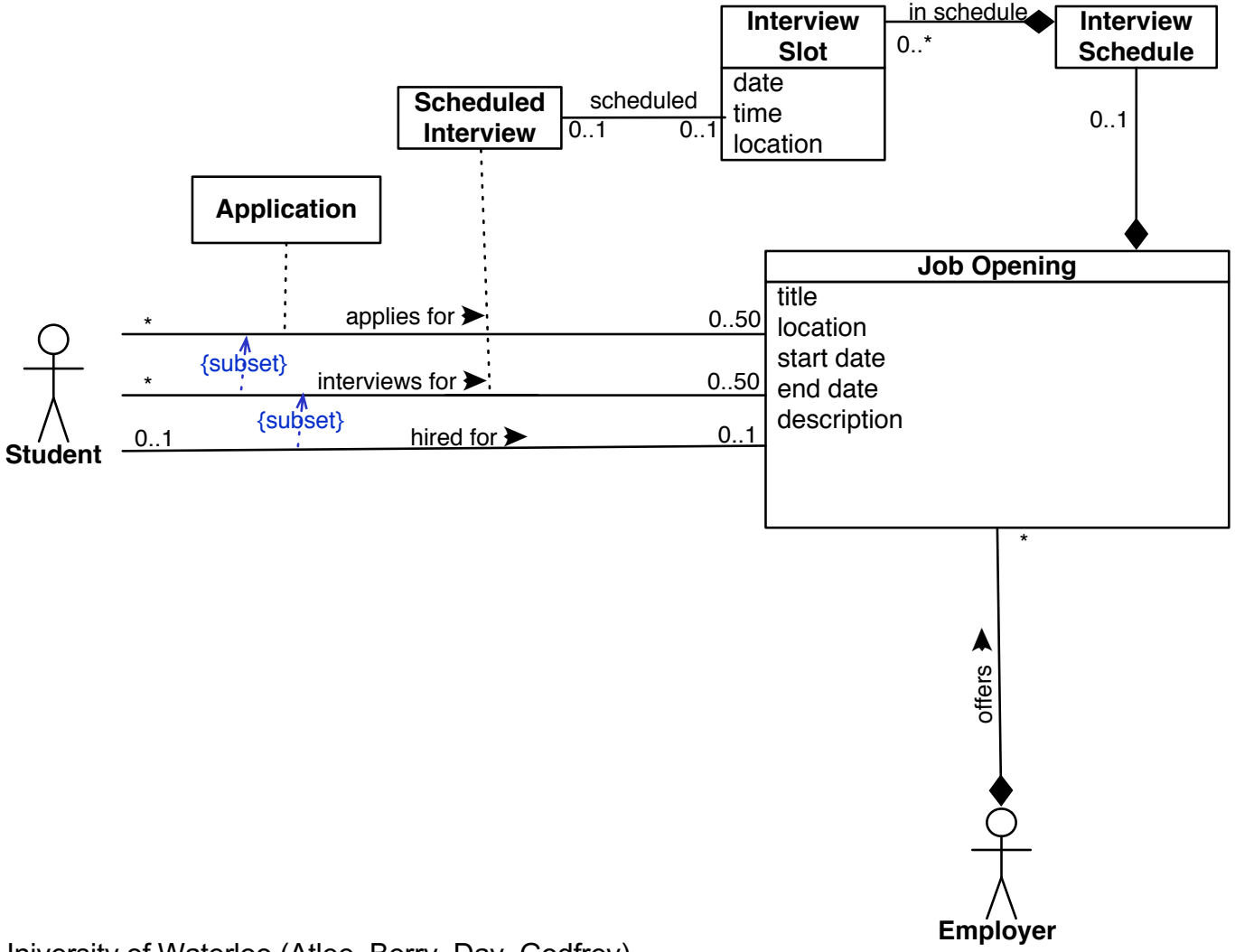
Interface entities are likely to be cyber entities:

- Accounts, user ids, passwords, records, record ids, Web forms, keystrokes, mouse clicks, screen displays, popup messages

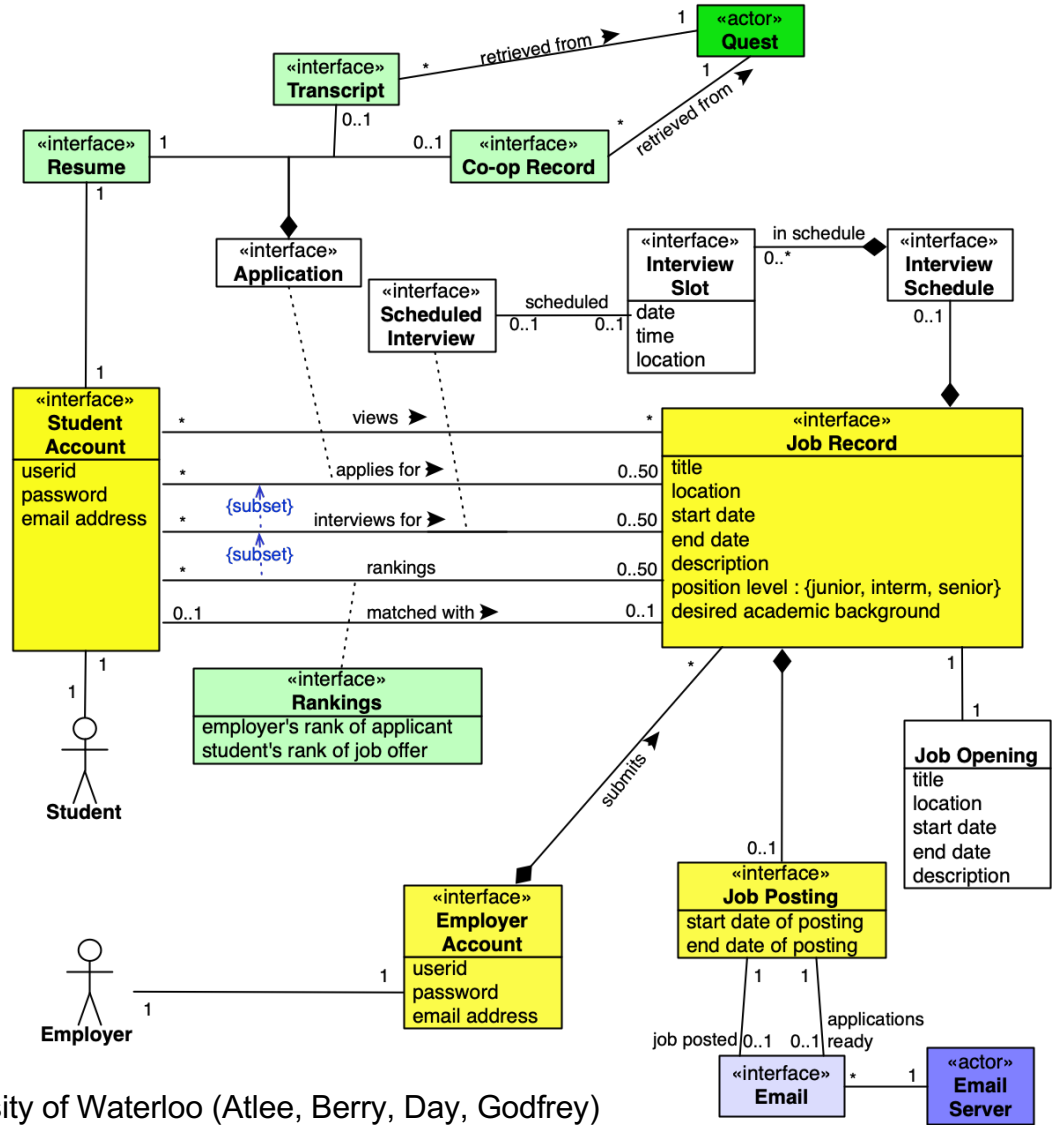
Example: WaterlooWorks



Requirements Domain Model



Specification Domain Model



Key:

- **Electronic record**
- **Input**
- **Input Service**
- **Output**
- **Output Service**

Domain Model Checklist

- «interface» classes in specification domain models
  - associations between «interface» classes and corresponding environment classes being sensed or controlled
- Attributes and Associations
  - attribute types, initial values if they are known and required
  - effective use of generalizations, compositions, aggregations, association classes
- Association names or role names on associations
  - except on compositions and aggregations
- Multiplicities on all associations (including “1” multiplicities)

Domain Model Anti-Checklist

Avoid design-level details such as

- Class-level operations or methods
- Visibility annotations (i.e., private, protected, public)
- Navigability arrows
- Object construction and destruction functions

Watch your language!

- The goal is to create a *conceptual* model:
  - Models of real-world entities (customers, accounts, bills) and not of system entities (databases, sw components)
- Focus on the information/artifacts that the system will input, transform, analyze, display, etc; physical and conceptual