

# Lecture 3

## Requirements Discovery: Use Cases

Com S/SE 409/509

Robyn Lutz

[rlutz@iastate.edu](mailto:rlutz@iastate.edu)

Robyn's Office Hours: **Tues & Thurs, 9:30-10:45**

Wandi's Office Hours: **Mon 10 am. & Wed 7 pm**

Olukorede's Office Hours--**Wed 10**

**Homework 1 due Sept. 3**

Copyright: Robyn Lutz, 2020



# Requirements Discovery Homework 1 (9/3 due)

- 4 skills:

- 1) Create a **context diagram** for a new product (scope it)
- 2) Develop its **product use case diagram** (partition it)
- 3) Identify/elicit missing **domain knowledge** (know what you need to find out)
- 4) Decide **team** responsibilities

→ Chap. 4

context ; outside world

- **Project Description** posted: software controller for landscape watering

- Client's description

- Skill 3: answers to your questions in Problem 3 will be provided

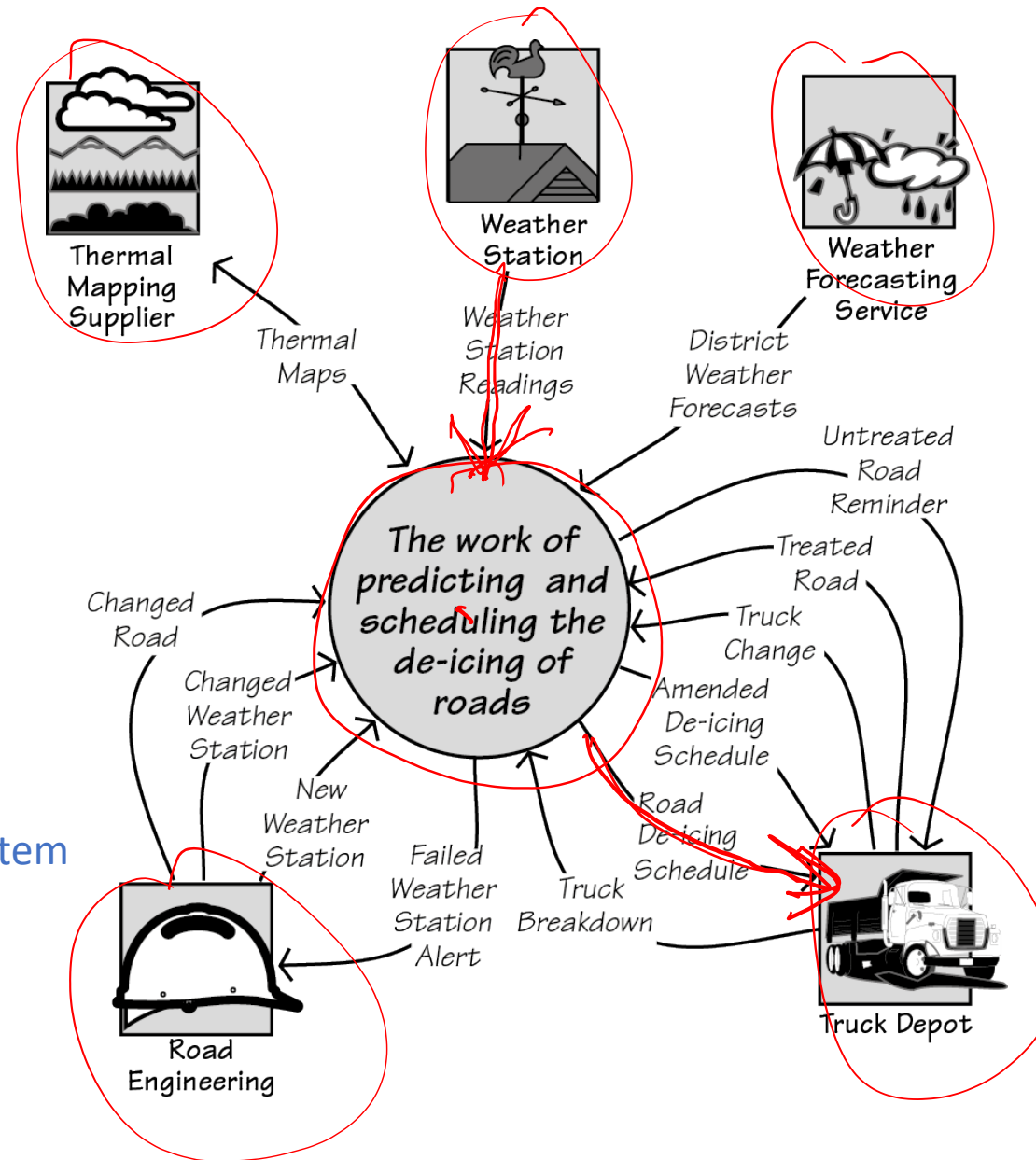
- Teams' products will be a product family:

- Shared core requirements
    - Some customized, team-specific variations

- 509: **research paper posted**: extra HW question for grad students

Figure 4.2

The context diagram showing the scope of the work. The central area of the diagram represents the work you are about to study, and the product you eventually build becomes part of this work. The outside world is represented by the adjacent systems—Weather Station, Truck Depot, and so on. The named arrows represent flows of information between the adjacent systems and the work.



## Review: Context diagram

Software product you'll build is within & *part of* the work's scope

Each arrow is a data flow event (input or output) from/to an adjacent system in the work's context

Data triggers the work, and perhaps the software, to do something

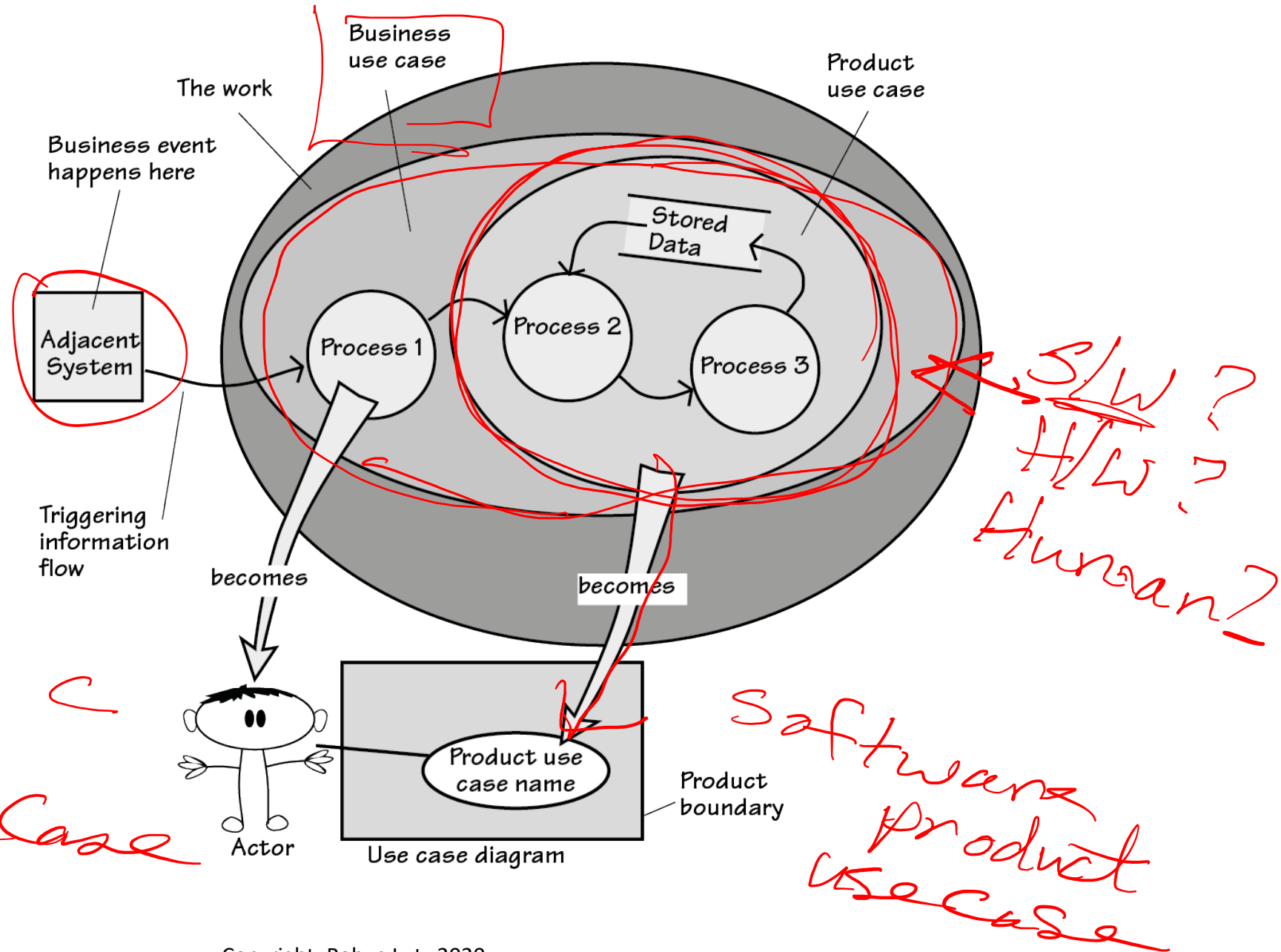
## Chap. 4: Business & Product Use Cases

- Scope the work to be done (“predicting & scheduling the de-icing of roads”) by partitioning it into use cases
- Use case: unit of functionality
- Use case diagram is a summary of the use cases
- Business use case: response of the work to a business event
  - Ex: Truck Depot reports a problem with a truck
  - Used in out-sourcing: client does Business Use Case diagram to describe needs; developer does Product Use Case diagram to describe what software does
  - Check your understanding: map each arrow (input/output information flow) in Fig. 4.8 (p. 79) forward to its triggering business event in Table 4.1 (p. 80).
- Ask: How much of the business use case is to be done by the software product use case?
  - Some of it may be done by humans or in software

*graphed*

Figure 4.11

The business event is some happening in the adjacent system. The resulting information flow notifies the work of the event and triggers a response (the business use case). After study, the requirements analysts and the interested stakeholders decide how much of the business use case is to be handled by the proposed product (the product use case). Whatever is immediately outside the scope of the product becomes the actor, who manipulates the functionality of the product use case within the product. A UML use case diagram is shown for comparison.



# Product Use Case diagram

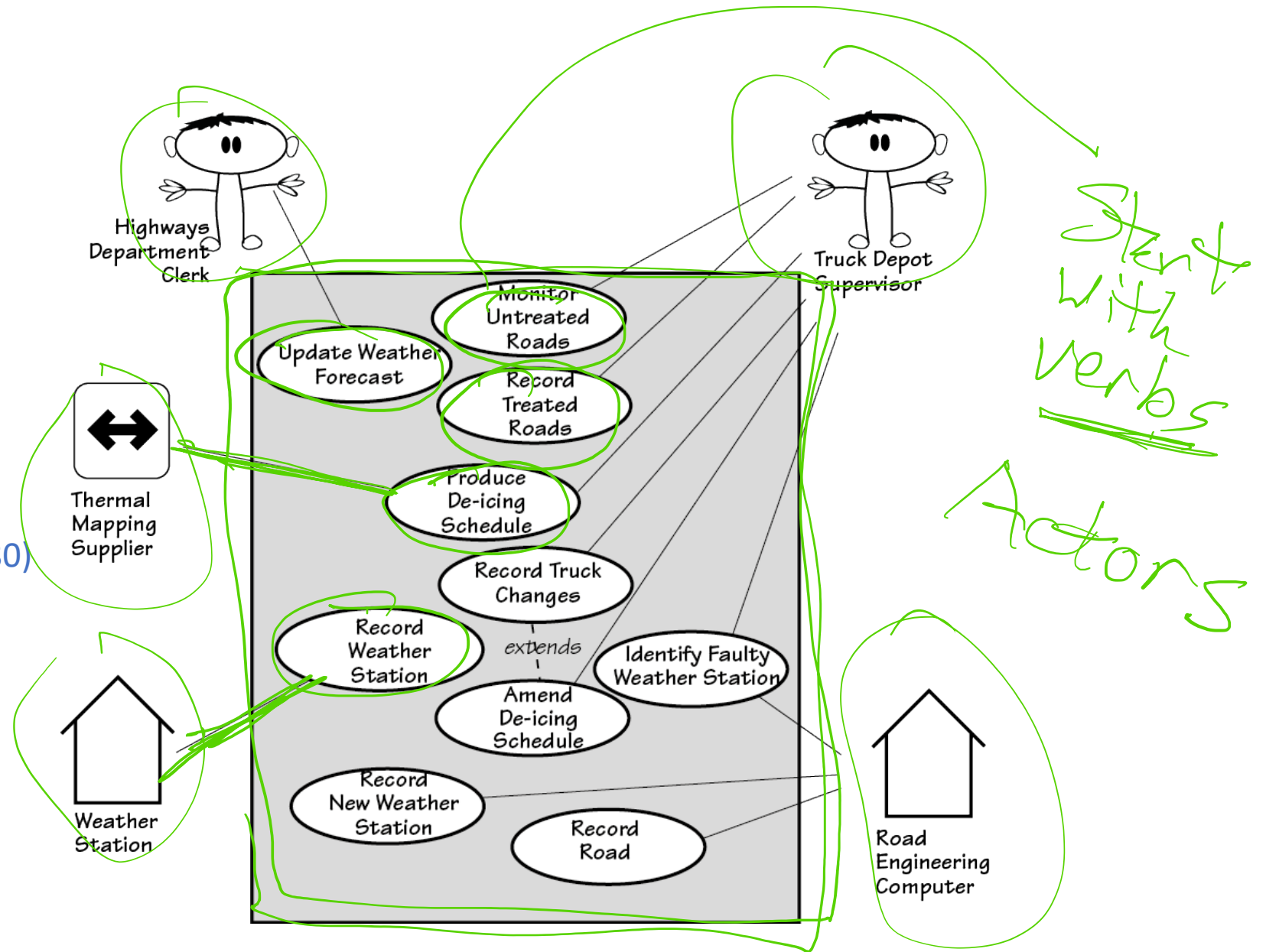
- Product use case:  
the part of the business use case handled by the automated system (the software product)
- Product use case diagram shows:
  - boundary between actors & software product to be built (rectangle)
  - product use cases (ellipses)
  - usage (lines)
  - actors: people or adjacent systems that interact with software product (pp. 190-194)
- Advantages:
  - High-level & informal model helps in discovering & reaching agreement on clients' needs
  - Popular & easy to prompt discussion of requirements with stakeholders
  - Some regulators require them
  - Helps in planning: builds, tests, prototypes, changes
- Limitations:
  - Some functional requirements will be missing
  - From the users' perspective, so internal processing requirements not included
  - Most non-functional requirements & constraints will be missing
- Solution [Wiegers & Beatty]:  
Use simple use case diagram, together with Functional Requirements

*eliciting*

*Graphical*

Figure 4.12

The product use case diagram for the IceBreaker product, showing the product use cases, the actors involved in each product use case, and the product's boundary. The different notation used for the actors indicates the way they interact with the product. (These distinctions are explained in Chapter 8, where we look at starting the product.)



Check your understanding:  
Map each Event in Table 4.1 (p. 80)  
forward to its Product Use Case  
in Fig. 4.12 (p. 84 & here).  
Which use case is missing?

Icons for adjacent systems:  
pp. 190-194

# Models

- We use models to help find the “real” requirements
- Models are abstractions focusing on a particular view of the product or the reality
  - Context diagram
  - Product use case diagram
- Simple models such as these are powerful ways to discover clients' & stakeholders' needs, as well as hidden constraints on your design