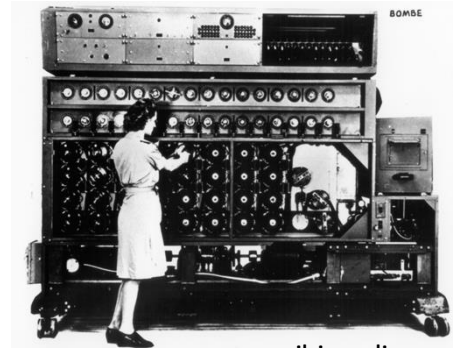


Lecture 4

Requirements Discovery & Analysis

Scenarios



wikimedia

Com S/SE 409/509

Robyn Lutz

rlutz@iastate.edu

Office hours:

Robyn: 11 Tues, 230 Atanasoff; 3 Fri (<https://iastate.zoom.us/my/rlutz>), & by appointment

Carter: cwunsch@iastate.edu, 12 Tues (0112 Pearson); 12 Thurs (<https://iastate.webex.com/meet/cwunsch>), & by appointment

Arushi: arushi17@iastate.edu, 1 Wed (0112 Pearson) & by appointment


Homework 1 due Thurs, 9/19

Requirements Discovery, Homework 1 (9/19, on Gradescope)

- 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 **lead-role** responsibilities (note: group works together on each HW problem)
- **Project Description**: software for health monitoring
- Groups' products will be a product family:
 - Shared core requirements
 - Some customized, group-specific variations
- **509: foundational research paper posted**: extra HW question for grad students

Reading assignment:

Chap. 6, “Scenarios”

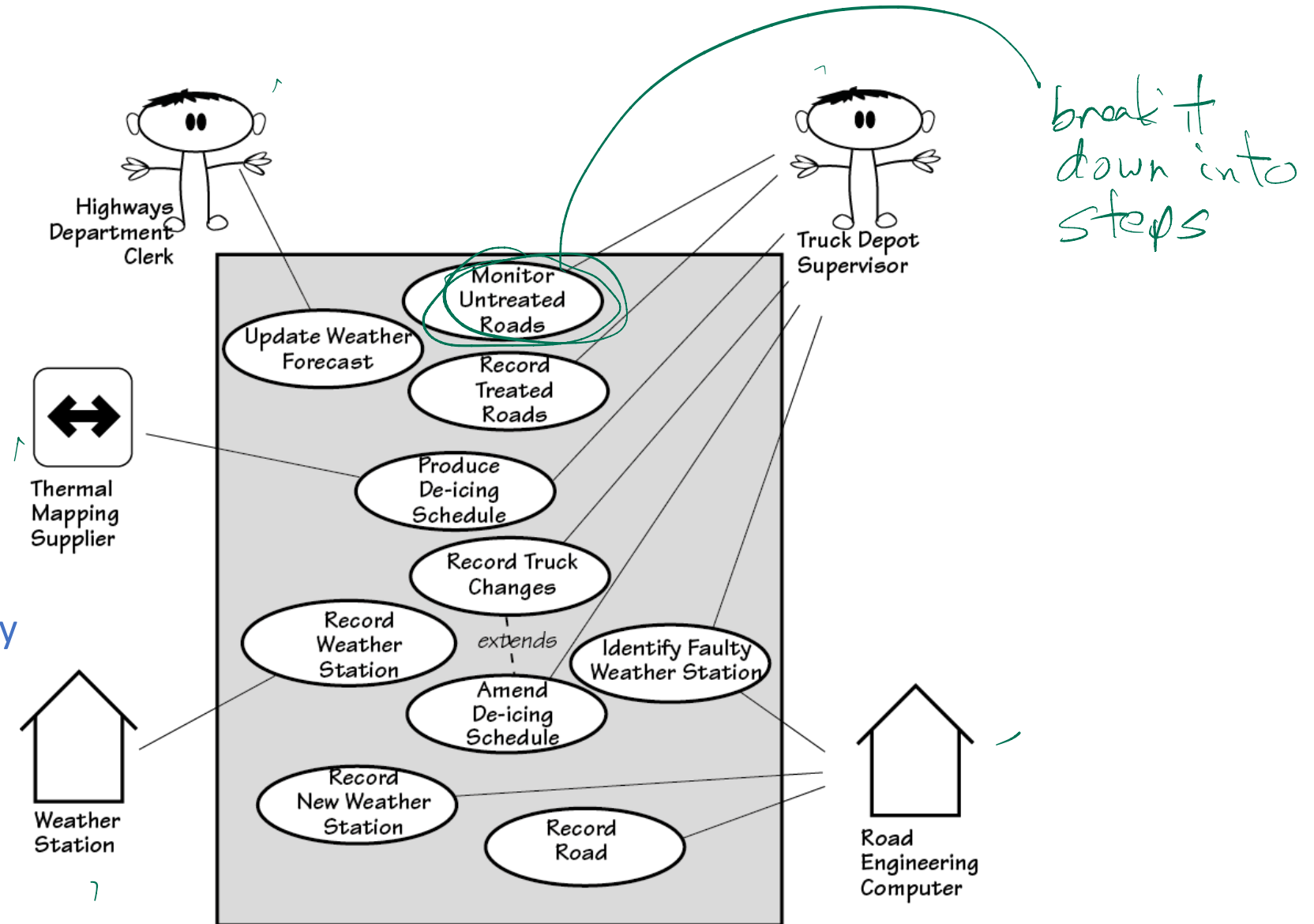
 Chap. 8, pp. 196-199, “Product Use Case Scenarios”

Also: pp. 239-240, 329-333 (scenarios & user stories)

(we’ve skipped Chap. 5 for now & will pick it up later on)

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.)



Review: IceBreaker's product use case diagram

Use cases are units of functionality the product needs

Actors are adjacent systems that Interact with the product

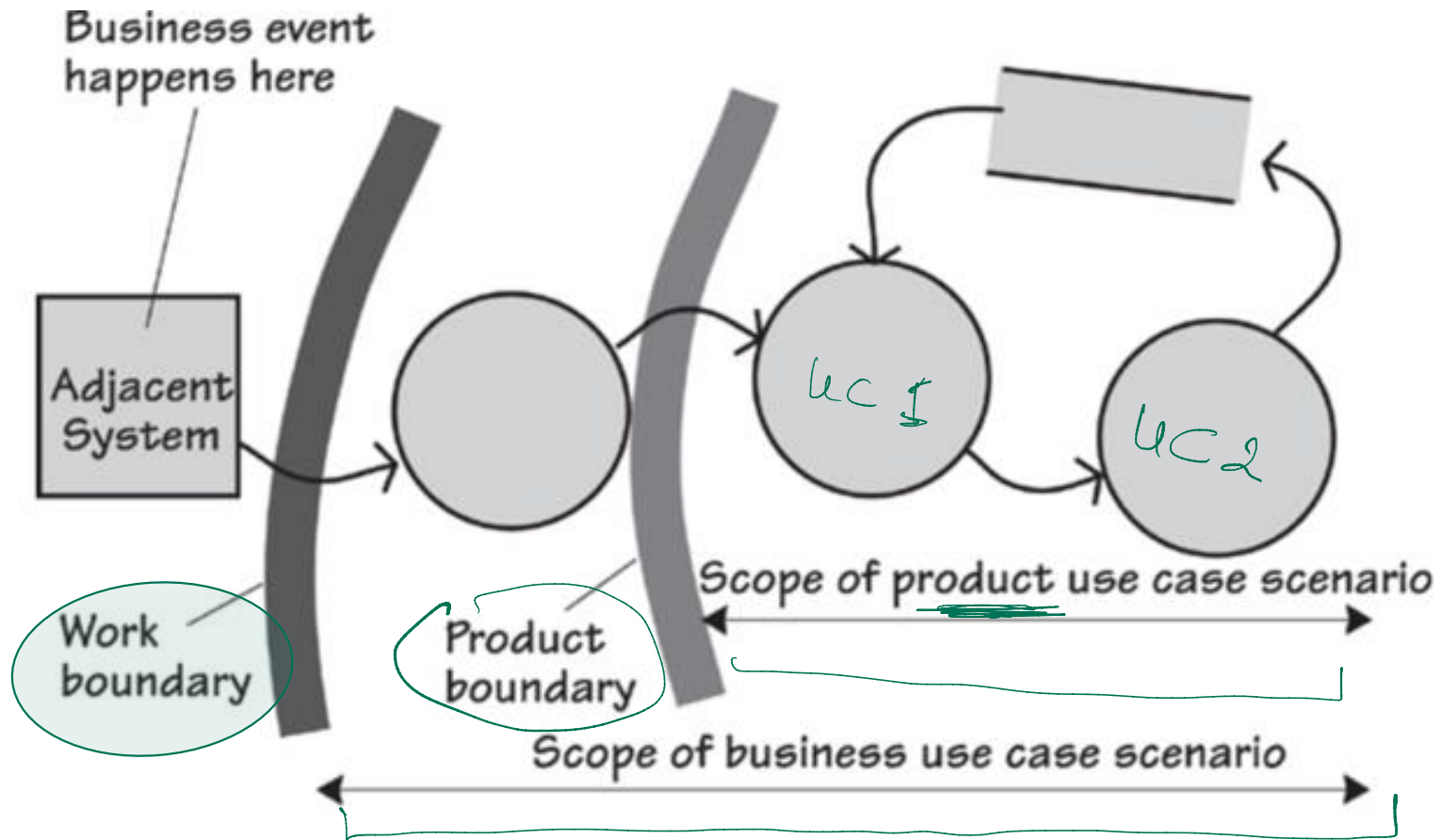


Figure 8.10. This functional model depicts the processes that respond to a business event, as does the business use case scenario; the product use case scenario shows the functionality to be included in the product.

Determine how much of the business use case is to be implemented as the product. The result is the product use case, which you can describe using a product use case scenario.

Product Use Cases → *Scenarios* → Functional Requirements

- Chapter 6: Use Case → Scenario

Example: Chapter 8 (toward the end): “Product Use Case Scenarios”

- What is a scenario?
 - It “tells the story” of a use case
 - Scenarios break down each product use case into a series of stakeholder-recognizable **steps** (Glossary)
- * • Build **at least one scenario for each product use case**
- Advantage:
 - Fast way to find/record/revise functional requirements in meetings with stakeholders

Product Use Case (PUC) Scenario

Example

Template & example: ~end of Chap. 8 (pp. 196-199)

PUC name: passenger checks into airline flight

***Trigger:** passenger activates the machine

Preconditions: ...

Interested stakeholders: ...

***Actor:** passenger

***Steps:** (1) software asks for passenger's name

...

(9) software prints baggage tag

Outcome:

* 3-10 steps recommended

Beyond normal scenarios

At each step, ask:

Alternative paths: are there alternative scenarios?

Actor has choices, so branch at that point

Ex: buy now OR save cart for later

Failures: are there exceptional/abnormal/off-nominal cases?

Unwanted deviations from normal scenario

Ex: buyer forgets password

Stopping criteria: Jackson's principle of commensurate care:

Required care = probability of an exception × seriousness of damage if it occurs

Ex: Scenarios for a product use case



Product use case:

Lab manager requests a chemical from Chemistry Stores

- 🌀 **Normal flow scenario [when it's in the storeroom]**
- 🌀 **Alternative flow scenario [when it has to be ordered]**
- 🌀 **Exception scenario [when it's not available]**

Scenario template:

✓ **PUC name:** passenger checks into airline flight

✓ ***Trigger:** passenger activates the machine

✓ **Preconditions:** . . .

✓ **Interested stakeholders:** . . .

✓ ***Actor:** passenger

✓ ***Steps:** (1) software asks for passenger's name

. . .

(9) software prints baggage tag

✓ **Outcome:**

Limitations of scenarios

Q: What are the drawbacks/limitations of scenarios?

Limitations of scenarios [Wiegiers & Beatty]

Partial coverage of possible system behaviors
Thus, misses some functional requirements

Too-early use risks over-specification & unnecessary constraints on order,
parallelization of steps

Doesn't capture the "why" of interactions (rationales)

Doesn't capture non-functional requirements (quality attributes)

reason why this requirement is needed

Beyond scenarios

- Storyboard: shows the steps of a scenario as a sequence of pictures
 - Good for User Interfaces, displays, dashboards
- Check your understanding: work through the PUC example at end of Chap. 8
 - the software to be developed will automate checking a passenger in for a flight

(software) product use case

User stories (pp. 239-240, 329-333)

User stories can be derived from business use cases

Especially used in agile approaches

As a [role] I want [functionality/feature]
so that [reason for use/use of the functionality]
rationale

Ex: As a [moviegoer] I want [to have my tickets sent to my phone]
so that [I can avoid the box-office queue].

In other words: persona + need + purpose [Rehkopf, Atlassian '22]

Stories often about features the product should have & reside in backlog until chosen

When user story is chosen for development, it's essentially refined into functional & nonfunctional requirements

* Good stories “Look at the real underlying need & not the guessed-at solution” (p. 331)