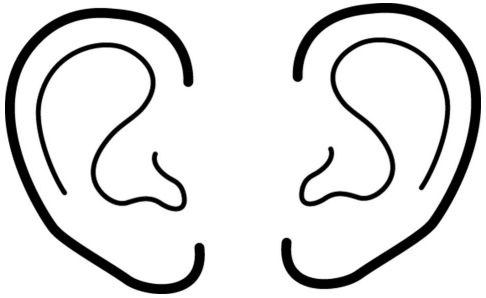
Lecture 6
Communicating the Requirements: EARS



(not in textbook; also work through posted A. Mavin slides)

Com S/SE 409/509
Robyn Lutz
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
Homework 2 due Sept. 17
Exam 1, Sept. 24

Upcoming

Reminder: Requirements Discovery Homework 1 (9/3 due)

HW

- Practice 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

- 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

Q: Why are requirements so hard to write?

[text in slides is adapted from Mavin, *IEEE Software*, 2012; Mavin & Wilkinson, 2010]

1. You have to figure out what's needed, & that's hard. (Lectures 1-5)
2. **Today: You have to figure out a clear way to express it**, & that's hard

- Fact: most requirements are written in text (English, etc.)
- Frequent problems: too wordy, ambiguous, vague, hard to understand, partial, premature implementation decisions, untestable
- Projects need clear & concise textual specification of functional requirements
- Technique that works: **EARS: Easy Approach to Requirements Syntax**

Specifying requirements using EARS

- EARS classifies all requirements into **5 basic templates**
- Results from industry show that requirements improve with EARS use
  - Problems/cost/developer misunderstandings are reduced
  - Helps projects build the right product

5 templates:

1. Ubiquitous

- A ubiquitous requirement is something that the system must **always do** (unconditional & continuously active)
- Ex: "The <system name> shall comply with regulation XXX."
- Template: The <system name> shall <response>

Template

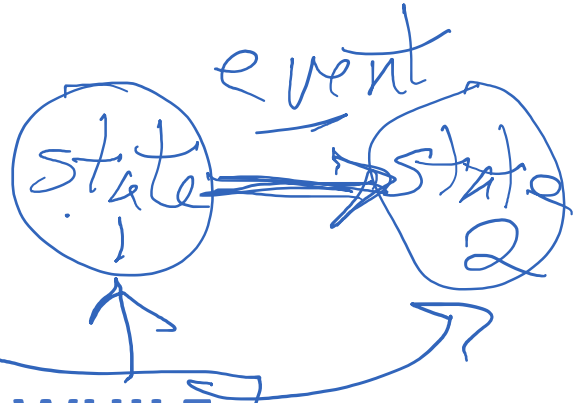
2. Event-driven

Event-driven keyword: **WHEN**

- System response is initiated by a triggering event the system detects at the system boundary
- Ex: WHEN commanded by the aircraft, the Engine Control System shall dry crank the engine."
- Template: WHEN <trigger> the <system name> shall respond.

Template

3. State-driven



Event: atomic

State: duration

State-driven keyword: **WHILE**

- Active while a particular state or states remain true,
- continuous as long as the state holds
- Ex: "WHILE the aircraft is in flight and the engine is running, the Engine Control System shall maintain engine fuel flow above *x* lbs./sec."
- Template: WHILE <in specific state>, the <system name> shall <response>

Template

4. Option

Option keyword: **WHERE**

- A system response is needed only in applications that include a particular feature
- used as a simple way to handle product or system variation, and
- Ex: "WHERE electronic components are used in the Engine Control System, they shall comply with DO-254." → standard
- Template: WHERE <feature is included>, the <system name> shall <response>

(won't use too much)

Template

5. Unwanted Behavior

- Unwanted Behavior keyword: **IF/THEN**
- Required system response to unwanted events (such as failures, disturbances, and any unexpected behavior of interacting systems or users)
  - variation of the event-driven requirement
- Ex: "IF the engine fails to start during a third attempt, THEN the Engine Control System shall terminate the autostart sequence."
- IF <unwanted trigger>, THEN the <system name> shall <response>

Template:

Combining these 5 templates

? state ?

event-driven

- Can state more complicated requirements by combining templates
- Ex: "**WHILE** the aircraft is on the ground, **WHEN** reverse thrust is commanded, the Engine Control System shall enable deployment of the thrust reverser"
- Ex: "**WHILE** the aircraft is in flight, **IF** reverse thruster is commanded, **THEN** the Engine Control System shall inhibit thrust reverser deployment."

unwanted behavior

Benefits of EARS & more examples

- Using EARS templates gives you simple, clear statements of the requirements
- EARS is a structured way to write better textual requirements

Assignment pre-HW2:

Check your understanding: Work through the examples in A. Mavin's EARS tutorial slides, posted on Canvas