Requirements Specifications

Requirements Engineering

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Tao Xie, Professor
Computer Science @ Illinois
Learning Objectives

By the end of this video, you will be able to

• Explain main purposes of requirements specifications.
• Classify a given statement to be a descriptive or prescriptive statement.
• Choose among “shall,” “should,” or “will” to use for a given requirement.
• Name example notations for requirements specifications.
Purposes of Specifications

Contracts

Communication

Evaluation
Requirements Specifications

(Zave & Jackson, 1997)

Problem World/Application Domain
(Problem: environment)

Shared Phenomena
(Interface)

Machine
(System solution: software + hardware)
Statement Types in Specifications

(Lamsweerde, 2009)

• *Descriptive* statements
  • Properties about the system that hold regardless of how the system behaves (e.g., due to natural law or physical constraint)
  • E.g., “*The same book copy cannot be borrowed by two different persons at the same time.*”

• *Prescriptive* statements
  • Desirable properties about the system that may hold or not depending on how the system behaves
  • E.g., “*A person may not borrow more than three books at the same time.*”
Domain Properties, Assumptions, Definitions

- **Domain property**: *descriptive* statement about the problem world (holds regardless of the system/software-to-be), e.g., physical law

- **Assumption**: (generally) *prescriptive* statement to be satisfied by the environment (of the software-to-be)

- **Definition**: statement providing a precise, complete, and agreed meaning to domain concepts or auxiliary terms

(Lamsweerde, 2009)
Natural Language Mood

• **Shall** (== *is required to*): mandatory requirements
• **Should** (== *is recommended that*): desirable requirements
• **Will**: a reference to the future, not under the control of the specified system

“I shall drown. No one will save me.”

“I will drown. No one shall save me.”
Many Notations

• Traditional (e.g., “The system shall ...”)

• UML, e.g., use cases, state diagram

• User stories

• Finite state machine

• Data flow diagram

• Pseudocode
References


The End