Requirements Specifications
Representations and Notations

Remember, Requirements Specification should be:
- Complete
- Consistent
- Unambiguous
- Precise
- Feasible
- Even
- Understandable
- Testable
- Free from Implementation Bias

Some notations work better than others

Common Approaches
- Natural Language (English)
- Structured/Disciplined Natural Language
- Diagrams
  or some combination of all of the above

Natural Language
- Write requirements in “plain English”
- Build upon universal understanding of natural language
- Create a glossary of terms and their definitions
  - Don’t introduce a new term if a previously defined term will do
  - Use modifiers to distinguish related terms
    - E.g., “Registered Customer” and “Anonymous Customer”
    - Can use “Customer” to refer to either

Requirement Template
- Header
- Glossary of terms
- High-level overview (often a diagram)
- Functionality for each functional component
  - Functional description
  - Interfaces
  - Error handling
  - Potential future modifications/extensions
  - Constraints
  - Risk Assessment
  - Test plan

Requirement Template
- Structure the document in sections by functional description
  - Separate concerns, usually by related functionality
Requirement Template Revisited

- Header
- Glossary of terms
- High-level overview (often a diagram)
- F1: functionality for F1 (may have a diagram to help explain the decomposition)
  - F1.1
  - F1.2
- F2: functionality for F2 (may have a diagram to help explain the decomposition)
  - F2.1
  - F2.2
  - F2.3
    - F2.3.1
    - ...
- F3

Commonly Used Diagrams

- Data Flow Diagrams
- Flow Charts
- Activity Diagrams
- Finite State Machines
- State Diagrams
- Message Sequence Charts
- Use Cases

- Each good for some things, not for others

Adding More to a DFD

- Use of “open boxes” to indicate data stores
  - Not a typical computation function

Data Flow Diagram Example

- LIBRARY
- Buy new books
- Update Request List
- New Request
- Get copy of Request List
- New Request
- Delete Request
- Satisfied
- Turn on light
- Button
- Press

Control Flow Diagram

```plaintext
total, value, count, maximum : int;
total := 0;
count := 1;
read maximum;
while (count <= maximum) do
  read value;
total := total + value;
count := count + 1;
endwhile;
print total;
```
More Abstract Control flow Diagram

Activity Diagram

Finite State Machine for Digital Watch

More Finite State Machine Detail

State Transition Diagram

Message Sequence Charts
Use Cases

Selecting the “right” notation
- Use diagrams that seem to fit the purpose
- Be consistent in the notation
- Define the notation

“Almost” a data flow diagram

Requirement Template
- Header
- Glossary of terms
- High–level overview (often a diagram)
- Functionality for each functional component
  - Functional description
  - Interfaces
  - Error handling
  - Potential future modifications/extensions
  - Constraints
  - Risk Assessment
  - Test plan

Header
- Title
  - Access Control
- Authors:
- Date and Version, with version history
  - 9/28/02 v1 preliminary draft
  - 10/02/02 v1.1 preliminary draft with modified glossary
  - 11/05/02 v2 improved user interface
- ...

Glossary of terms
- Define any major “noun” that is introduced
- These are the objects/resources that will be used
- Define the conceptual structure of these objects if they are major “nouns” in the requirements
  - Not the implementation
**Functional Description**

- **Describe proposed behavior**
  - Short, precise statement of each required function that needs to be supported

- **Model how information flows through the system with these functions**
  - Often use diagrams

- **Sometimes show the current functionality AND the proposed functionality to highlight differences**

**Interfaces**

- **User interfaces**
  - kinds of information and style

- **Component interfaces**
  - other tasks/subsystems that will be used by this task
  - Brief explanation of what will be needed
  - other tasks/subsystems that are expected to use this task

- **Data interfaces**
  - type, quantity, frequency, reliability
  - operational profile (expected scenarios)
  - stress profile (worse case scenarios)

- **Hardware interfaces**

**Error handling**

- **errors that are anticipated and the expected response**
  - entering an incorrect password results in a denial of access
  - entering an incorrect password more than three times in a row results in a message being sent to security

**Potential future modifications**

- While doing the requirements, some extensions will become apparent

- Often these are generalizations of the proposed functionality
  - Processing more than one type of form
  - Allowing an administrator to define the types of forms
  - Allowing the format of the forms to change dynamically

- Indicate the expected difficulty and approach to support these extensions

**Constraints**

- Performance, hardware, security and reliability, etc.

**Risk assessment**

- Areas where there is uncertainty
  - Not sure if certain resources will be available or affordable
  - Not sure if certain approaches will be able to meet constraints
  - Alternatives to be explored to address these risks
Validation plans

- provide a high level test plan
- use cases/activity diagrams can be used to describe the major flow of information through the system
- Each flow through the one of these diagrams corresponds to a test case

Hierarchical Representation

- Header
- Glossary of terms
- Functionality
  - functional description
  - Interfaces
  - Error handling
  - Potential future modifications/extensions
  - Constraints
  - Risk Assessment
  - Test plan

For each (sub)function

- Include only the information that is appropriate
- Function F_i
  - Functionality
  - [Interfaces—only those that exist ]
  - [Error handling]
  - [Potential future modifications/extensions]
  - [Constraints]
  - [Risk Assessment]
  - Test plans

Plan your presentation

- 30 minutes is a very short time
- Plan and rehearse what you will say
- Use of visuals is very important
  - Not too many, though
- 1–2 speakers
  - Everyone on your team will make a presentation at some time during the semester
- Other team members provide support
  - Help in the preparation of the slides
  - Answering questions as they arise
- The purpose is to teach and learn
  - Be prepared to do both

Presentations

- Usually best to present material top down
  - Say what you are going to say
  - Say it
  - Review what you said

Typical Outline

- Provide a high–level overview of the problem you are address
- Provide a high–level overview of your approach
- Provide an outline of what you are going to discuss based on that overview
  - For each piece (or a selected subset):
    - Provide an overview of the problem
    - Provide an overview of the approach
    - If appropriate, drill down to the next level of detail
- Return to the high–level approach and show how it all fits together