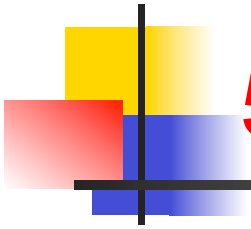
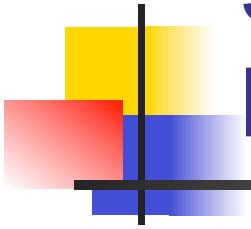


5. Products

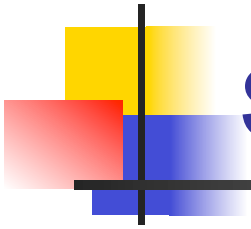
Overview

- 5.1 Requirements Definition Document
- 5.2 Software Requirements Specification

5.1 Requirements Definition Document

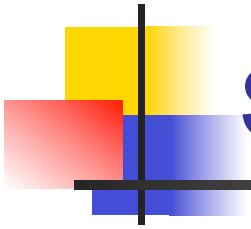


- Point of origin
  - elicitation activity
- Purpose
  - reveal and clarify project objectives
- Focus
  - corporate level concerns
  - communication venue
- Nature
  - high level
  - technically incomplete
- Usage
  - trade-offs resolution
  - starting point for technical specifications
  - starting point for technical studies



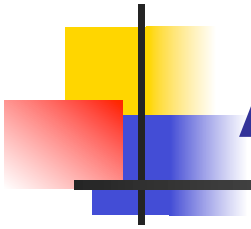
Sample Table of Contents

1. Introduction
   - *the mind set*
2. Definition of terms
   - *the basis for accurate communication*
3. Objectives
   - *the central issue*
4. Overall system organization
   - *the context*
5. Interfaces
   - *the environment refined*
6. Capabilities
   - *the outline for a solution*
7. Constraints
   - *the bounds placed on the solution space*
8. Additional documentation
   - *attached or included by reference*



Special Cases

- Variations in the nature of projects often leads to specialized documentation
- One needs to be make sure that specialization does not lead to omissions
- Sample specializations and likely limitations
  - user manual (§5 and §6)
  - product specification (§4 and §5)
  - marketing specification (§4 and §7)
  - interface specifications (§5)

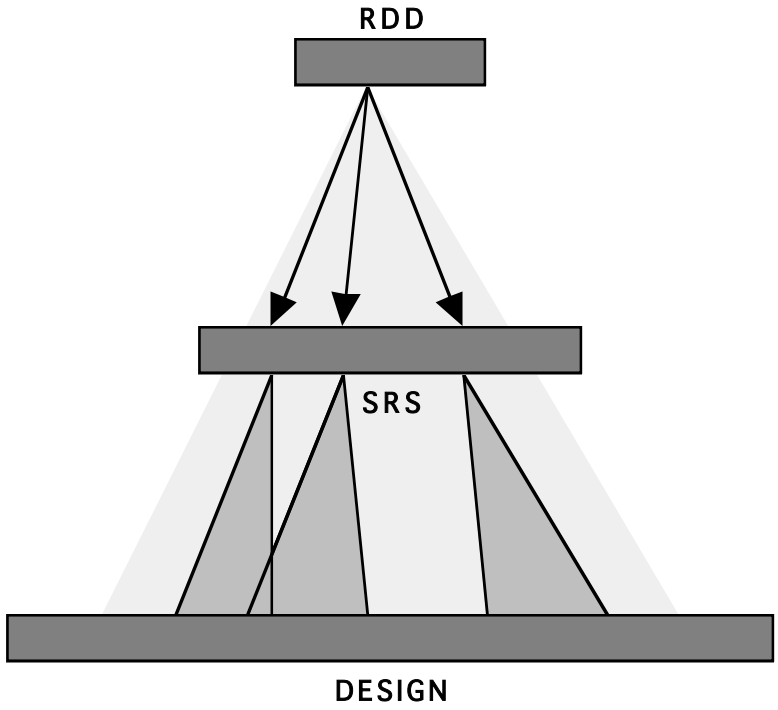


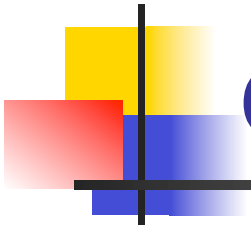
A Question of Quality

- The Requirements Definition Document is a live document
  - subject to changes in the requirements
  - interacting with the SRS to define and redefine the baseline
  - justifying or invalidating design choices
- Completeness
  - completeness can be judged only in connection with the SRS
- Control
  - procedures are identical to the SRS

Traceability

- Objectives, capabilities, and constraints in the RDD are traceable to the SRS
  - having to trace the less-technical RDD to design can become counterproductive
  - one objective may affect everything
- Using the RDD only to construct the SRS is feasible for some projects
  - objectives and rationale should not be lost

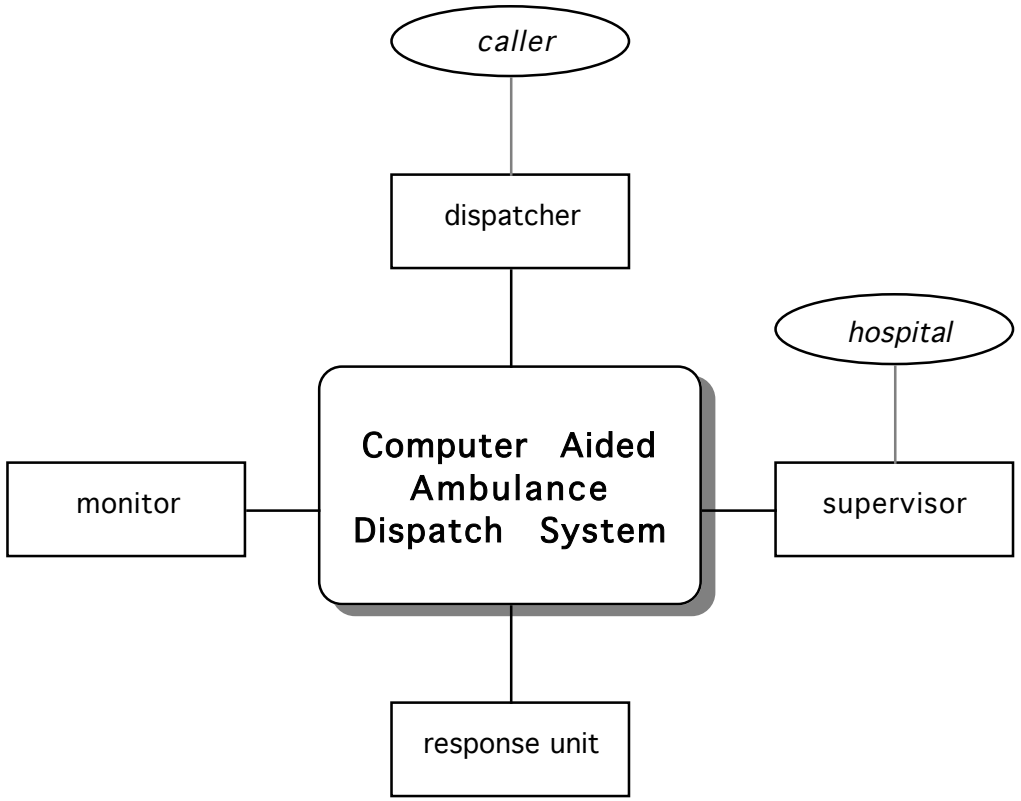


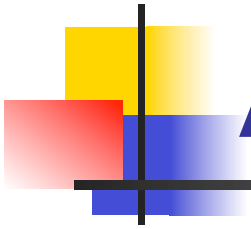


Case Study: Ambulance Dispatch

- The ambulance dispatching system consists of dispatch centers and response units.
- A call is first handled by a dispatcher in the dispatch center.
- The dispatcher enters the incident data into the system and sends a response unit to the incident site.
- After the response unit is dispatched, communication between the unit and the dispatch center is handled by a monitor until the incident is complete.
- Response units communicate with the dispatchers and monitors via mobile data terminals located in each ambulance.
- Supervisors oversee the entire system operation.

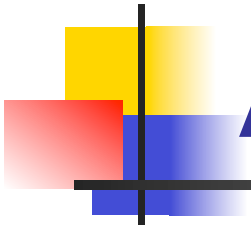
Ambulance Dispatch





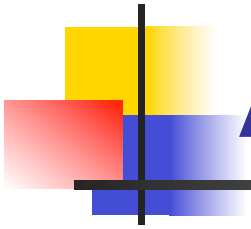
Ambulance Dispatch RDD

1. Introduction
   - ambulance dispatching process
   - summary of the requirements
   - report organization
2. Definition of terms
   - ambulance, crew, call, duplicate call, incident, incident record, open incident, dispatch record, open record, etc.
3. Objectives
   - maximize response to the incident
   - minimize human error
   - provide dependable operation



Ambulance Dispatch RDD

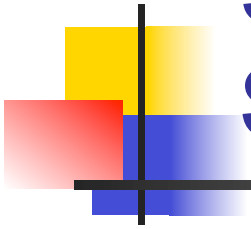
4. Overall system organization
   - identify interfaces and the nature of the interactions
   - identify communication outside the system (voice)
5. Interfaces
   - dispatcher, monitor, supervisor, response unit
6. Capabilities
   - call life cycle (e.g., dispatch records)
   - incident monitoring (e.g., incident records)
   - response unit life cycle (e.g., accepting an assignment, reporting location, updating status, etc.)
   - resource management (e.g., personnel, crews, ambulances, maps, hospitals, stations, etc.)
   - supervision tasks (e.g., reports and statistics)



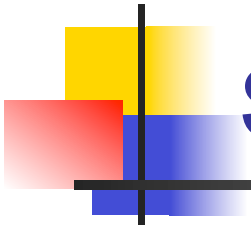
Ambulance Dispatch RDD

7. Constraints
   - reliability
   - loads
   - archive
   - survivability
8. Additional documentation
   - tbd

5.2 Software Requirements Specification



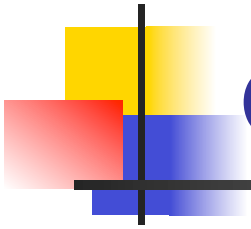
- Point of origin
  - elicitation or allocation activity
- Purpose
  - provide a baseline for all software development activities
- Focus
  - software/environment interactions
  - technical reformulation of constraints
- Nature
  - highly technical
- Usage
  - design
  - testing
  - technical studies



Sample Table of Contents

(ANSI/IEEE STD-830-1984)

1. Introduction
2. General description
3. Specific requirements
   - 3.1 Functional requirements
     - input/processing/output
   - 3.2 External interface requirements
     - interface specification
4. Performance requirements
5. Design constraints
6. Attributes
7. Other requirements



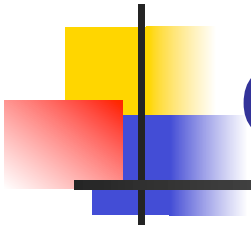
Guiding Principles

- The choice of specification method is determined by the need to maximize communication quality and speed
- Whether formal or informal in style, an appropriate underlying model is required for the specification task
- The choice of presentation style must recognize the realities of change

- simplicity
- clarity
- precision

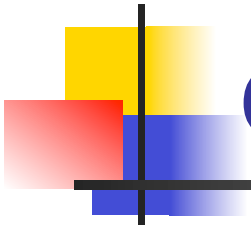
- design independence
- soundness
- completeness

- modifiability
- traceability
- testability



Communication Challenges

- Applying technical writing skills in context is a very difficult art to master
- The document needs to consider the audience and the local culture
- There are many opportunities for errors and misunderstandings in a lengthy document
- It is often forgotten that a document is written once and read many times

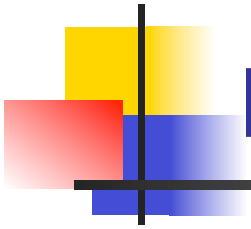


Case Study: Duplicate Calls

- The RDD for an ambulance dispatch system includes the requirement

  "Detect and consolidate duplicate calls."

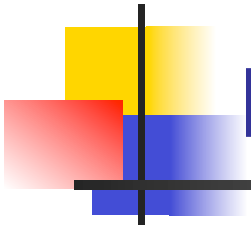
- The SRS might refine this capability in terms of the following requirements
  - Multiple calls regarding the same incident may arrive
  - Only one ambulance should be dispatched
  - No calls should be lost
- The SRS must be specific.
  - How are potential duplicates detected? (criteria)
  - Who decides on the duplication? (system or user)
  - Is the decision reversible?
  - For how long?
  - Who assumes dispatching responsibility?
  - Can we guarantee that someone is always responsible for the call?



Duplicate Calls

Solution 1:

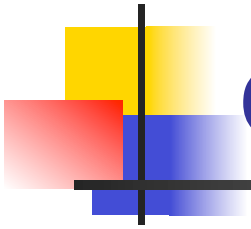
- Candidate duplicates are selected (by matching addresses, names and phone numbers) by the system and presented to the dispatcher
- Any dispatch record may be marked as duplicate of any other record by anybody who observes the duplication
- Any duplication can be nullified by anyone who has information that invalidates the duplication contention



Duplicate Calls

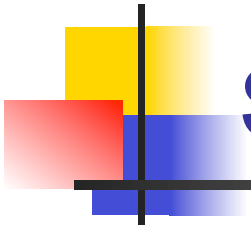
Solution 2:

- Candidate duplicates are selected (by matching addresses, names and phone numbers) by the system and presented to the dispatcher
- The dispatcher marks the current dispatch record as duplicate of another (one only)
- Disjoint sets of duplications are created and no cycles are present
- The dispatcher can change his/her mind at any time before an ambulance is dispatched and an incident record is created
- Any other dispatcher may mark the duplication as questionable and defer to the monitor for a final decision



Other Sources of Complexity

- The information available is often
  - too low level—interface specifications
  - too complex—human interfaces
- The propensity to change
  - a test of quality is the ability to localize the effects of changes in the document
- The difficulty to achieve completeness and precision
  - a test of completeness and precision is the ability to construct a rapid prototype from the document alone

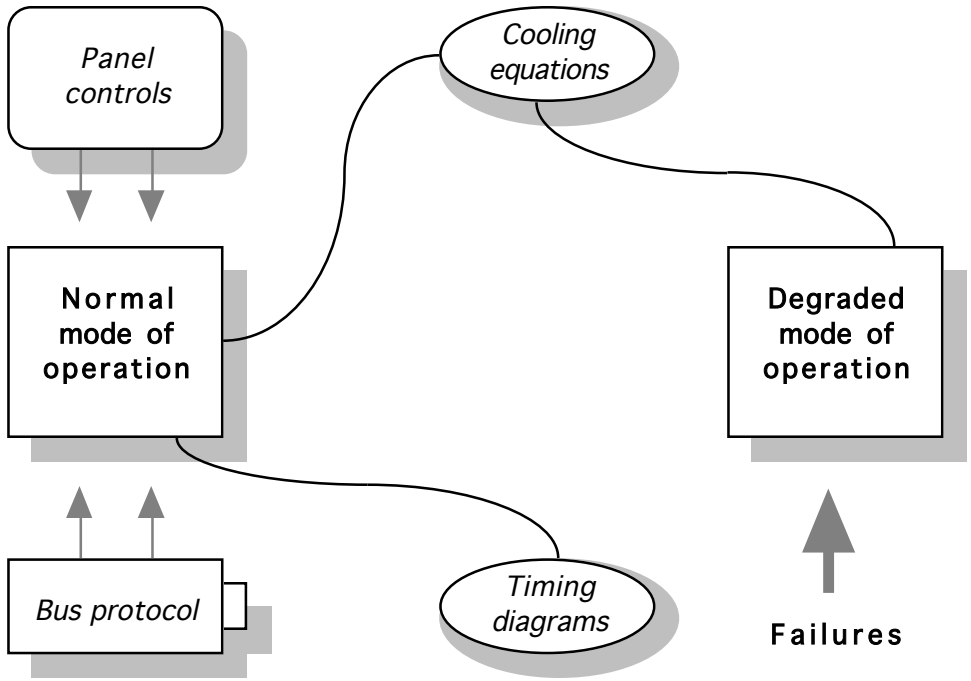


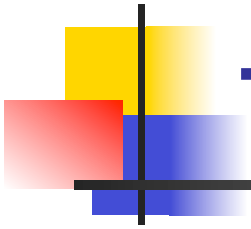
Some Technical Difficulties

- Requirements often require multiple views to achieve a complete specification
  - specialized models
  - multiple modes of operation
  - multiple specialized versions
- Certain concepts have no direct representation in the specification models used today
  - time is absent from most models
  - it is hard to discuss response to failures before doing any design

Response to Failures

- Consider an application involving the annealing of special glass
- Safety demands that certain functions be performed even in the presence of major faults (e.g., when neither manual nor remote control is possible)

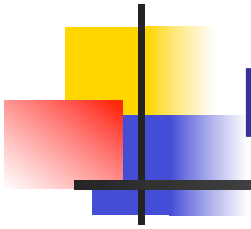




Traceability

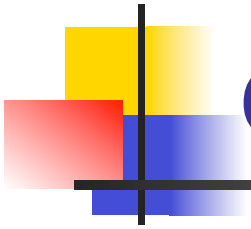
- Design verification requires one to show that all requirements have been factored in the design
- Requirements changes must be assessed with respect to the design areas affected
- Trade-off decisions must be recorded and justified

- Capabilities and constraints must be cataloged and their impact must be tracked
  - requirements level
    - label
    - type (mandatory / optional, stable / volatile)
    - status (active / eliminated)
    - reason
  - design level
    - impacted area
    - design rationale



Model Selection Difficulties

- Selecting the right model for the task requires highly specialized skills
- Standardizing across the organization builds corporate expertise but may cause problems for individual projects
- There is a natural tendency to design solutions rather than conceptualize problems



Case Study: Library

- Basic concepts
  - book — call number, author, title
  - client — library card number, name, address
- Library data
  - book holdings
  - client list
  - checked out listings
- Initial operations
  - check out
  - return

- **Question**: How can we minimize the impact of changes?

- Later additions
  - search for books
  - place a hold
  - pay fine
  - list client data
  - add clients and books