EEC 521: Software Engineering

Analysis Modeling - 2

Data Flow Diagrams

Structured Analysis

- Models data elements
  - Attributes
  - Relationships
- Models processes that transform data

modeled using

modeled using

Modeling Tools

- Data object diagrams
- ERD diagrams
- Data flow diagram
- Process narrative

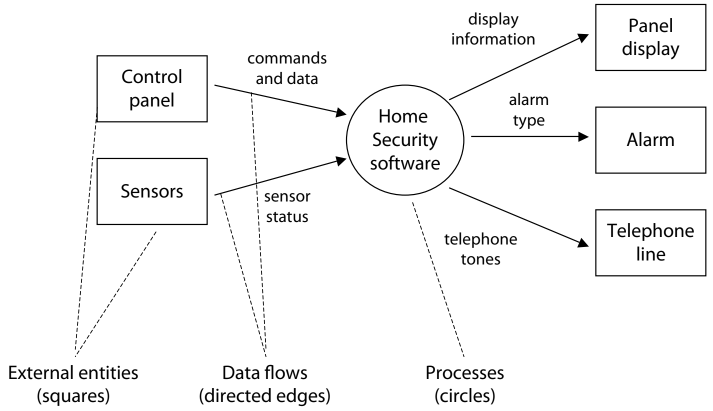
A data flow diagram describes information flow among *a* ***set*** of processes and actors.

1

*

A process narrative describes how *a* ***single*** process transforms input data to output data.

DFD: A Basic Example



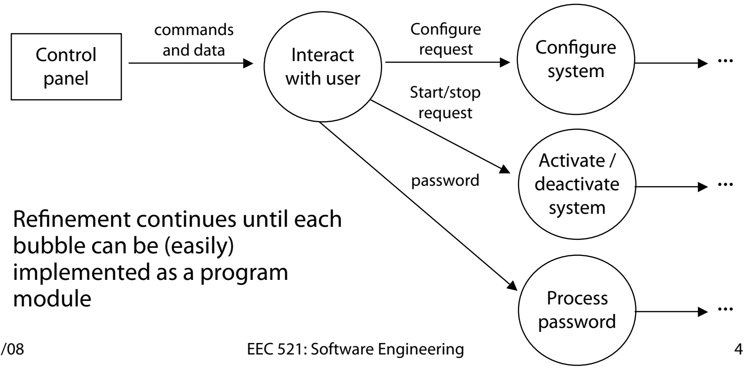
Notice that the system is represented as a single bubble

This is known as a level 0 DFD, or a context diagram

DFDs and Progressive Refinement

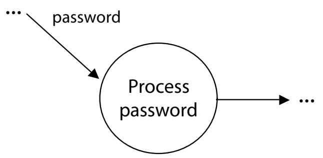
Each DFD reveals progressively more detail than the DFD that preceded it

Level 1 DFD:



Refinement continues until each bubble can be (easily) implemented as a program module

Process Narrative

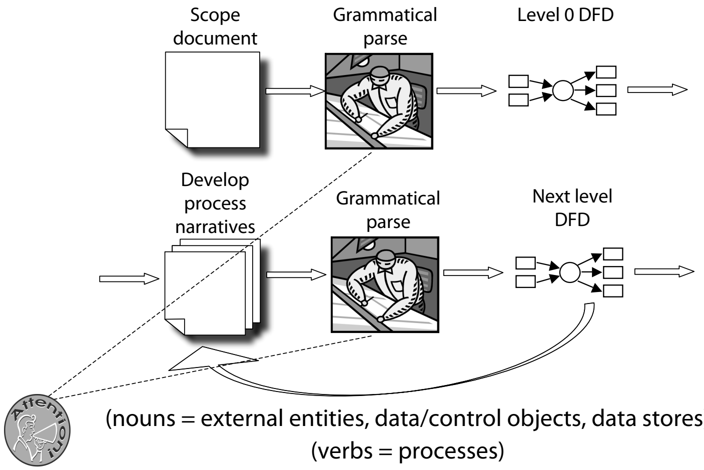


A process specification describes all of the flow processes in the final (most detailed) DFD.

"The ***process password*** transform performs password validation at the control panel for the SafeHome security function. Process password receives a four-digit password from the ***interact with user*** function. The password is first compared to the master password stored within the system …"

A process specification can be represented as a collection of process narratives.

DFD Construction



(nouns = external entities, data/control objects, data stores)

(verbs = processes)

Note that nouns and verbs are associated with one another

Some Guidelines

- Level 0 DFD must contain only a single bubble
- All arrows and bubbles should be meaningfully labeled
- Refinement begins by isolating next level processes, data objects, and data stores
- Refine only one bubble at a time
- Data flow continuity must be maintained between levels

Exercise: Data Flow Diagrams

Create a level 0 data flow diagram for a basic automated teller machine (ATM).

You can ignore administrative scenarios.

What would the level 1 data flow diagram look like?

A Different View

Object-Oriented Analysis

- Models analysis classes
  - Data
  - Processes
- Models class collaborations

Modeling Tools

modeled using →
- Class diagrams
- Packages

modeled using →
- CRC cards
- Sequence diagrams

Object-oriented analysis results in an analysis model that describes a system of collaborating ***objects***

Every object ***encapsulates*** a set of data elements, and ***exports*** a set of operations for working with those elements

CRC Cards

- Class-responsibility-collaborator (CRC) cards provide a way to *organize* a system's classes
  - Each class is represented by an index card that identifies:
    - The class
      - Just a simple name
    - Its responsibilities
      - Description of attributes and operations
    - Its collaborators
      - Other classes that help satisfy responsibilities

CRC Cards: Example

Class: FloorPlan	
Description: The FloorPlan class ...	
Responsibilities:	Collaborators:
Defines floor plan name/type	
Manages floor plan positioning	
Scales floor plan for display	
Incorporates walls, doors, and windows	**Wall**, **Door**, **Window**
Shows position of video cam	**Camera**

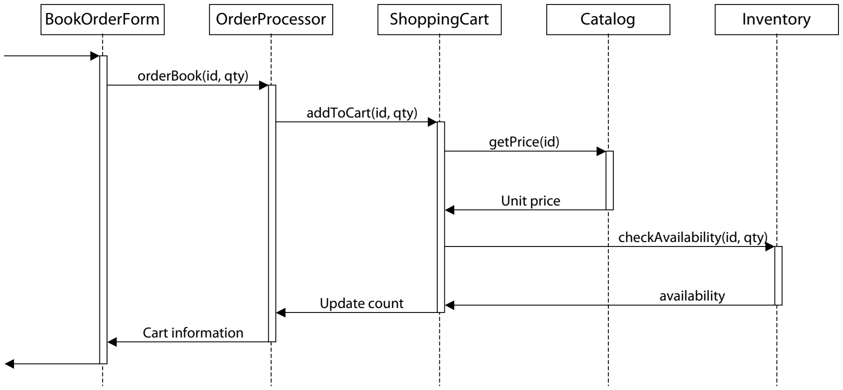
Model Validation with CRC Cards

- A CRC model can be validated by the project team using the following procedure:
  - All participants are given a set of CRC cards
    - Cards that collaborate should be separated
  - Each use-case is read by the facilitator
    - When a named class is reached, a token is passed to the person holding the corresponding CRC card
  - The token holder reads the responsibilities on their card
    - The team determines whether the class's responsibilities satisfy the use-case
    - Card responsibilities and collaborations are validated
  - Responsibilities and collaborations are updated

UML Sequence Diagrams

- CRC cards provide one way to model collaborations
  - Easy to develop and maintain
  - Easy for stakeholders to understand
  - High-level view of responsibilities and interactions
  - Acid-test validation
- *UML sequence diagrams* provide an alternative view, focusing on method interactions
  - More difficult to develop and maintain
  - Not always easy for stakeholders to understand
  - More detailed view of interactions
  - More detailed validation

UML Sequence Diagram: Example



Sequence Diagram Exercise

- Draw the sequence diagram for “Checkout”