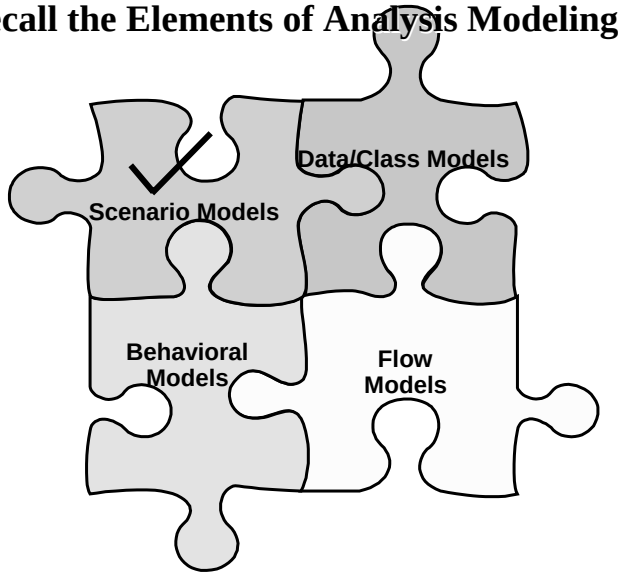
Software Engineering I CS-382

- Lecture 8
- What we will cover: (Details of Analysis Modeling)
  - Chapter 8 Sections 8.1, 8.2, and 8.6 in Pressman
  - Goal is to understand the methods and tools available from ***Structured Analysis*** for analysis modeling

Recall the Elements of Analysis Modeling



Recall the Tools We Will be Using for the Remaining Analysis Modeling

- **Structured Analysis-based Techniques**
  - Use Data Flow Diagrams, Control Flow Diagrams for the Flow Models and State Transition Diagrams for the Behavior Models
  - develop a Data Dictionary for the Data Models
  - Finish with process narratives (PSPECS) and control specifications (CSPECS) to define low level requirements.

Recall the Tools We Will be Using for the Remaining Analysis Modeling II

- **Object Oriented Analysis-based Techniques**
  - Derive the details of Class Models (using CRC Cards) and hierarchies (Class Diagrams)
  - Derive the Behavioral Models with Sequence Charts
  - Finish with Object Constraint Language (OCL) specifications of the methods

Modeling Rules of Thumb

- The model should focus on requirements that are visible within the problem or business domain.
  - The level of abstraction should be relatively high. (If out-sourcing then typically need more detail)
  - Be certain that the analysis model provides value to all stakeholders.

Modeling Rules of Thumb II

- Each element of the analysis model should add to an overall understanding of software requirements
  - Provide insight into the information domain, function and behavior of the system.
  - Delay consideration of infrastructure and other non-functional models until design.
- Keep the model as simple as it can be.

Analysis Modeling via Structured Analysis

- **Structured Analysis** views a system as a sequence of ***transformations*** operating on the input data and leading to the final outputs
  - Represents how data objects are transformed at they move through the system
- While considered "old fashioned" it can be helpful in defining end-to-end system operation
  - The data at the flow inputs can even help define object classes if used with OO methods or can be useful for describing processing within some of the methods

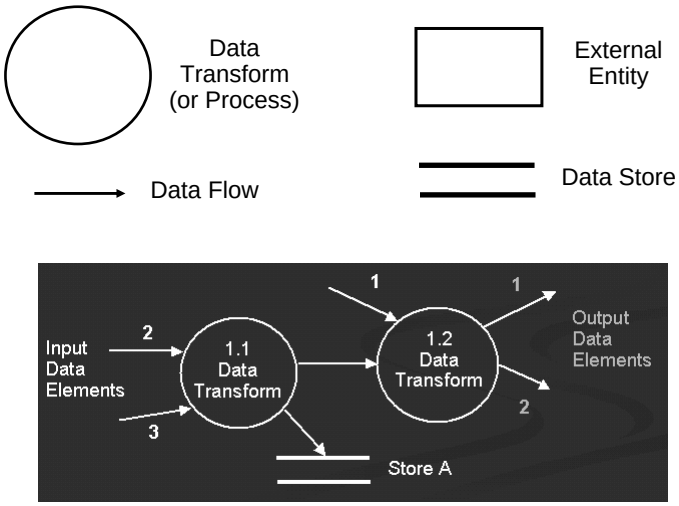
Steps for Analysis Modeling via Structured Analysis

- Starts with the **Context Diagram**
- Continue with **Data Flow Diagrams** to define all the data transformations
- Down to the lowest level uses **Process Specifications (PSPECS)** (narratives) to define the data transformations required at the lowest desired level of modeling
- The control of the system is defined through companion Control Flow diagrams
  - **State Transition diagrams** are then used to define the low level control modeling

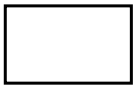
Flow Modeling with the Data Flow Diagram

- Elements of Data Flow Model
  - **Input Data** – the inputs to the current transformation
  - **Data Transformation** – the processing to be executed on input data
  - **Output Data** – the outputs from the current transformation
- The data flows are generated in a top-down process
  - Stop when the process defined in a single process bubble is easily explained
  - Then use **Process Specifications (P-SPECS)** to define the process

The Elements of the Data Flow Diagram



External Entity

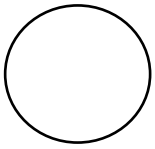


A producer or consumer of data

Examples: a person, a device, a sensor, a computer-based system

Data must always originate somewhere and must always be sent to something

Process (or transformation)



A data transformer (changes input to output)

Examples: Compute Taxes, Determine Area, Format Report, Display Graph

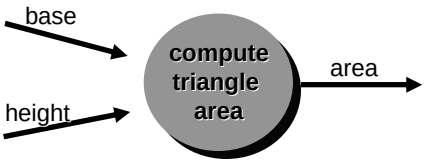
Data must always be processed in some way to achieve system function

Data Flow

Data flows through a system

Begin as input and be transformed into output

Examples: salary, grades, students enrolled, video, etc.

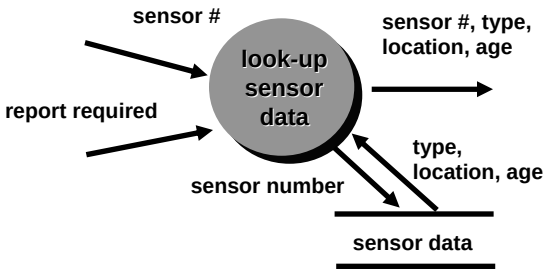


Data Stores

Data is often stored for later use.

Still may not be persistent (permanent) data but is needed for later processing.

Examples: last video frame for Safehome motion detection, data calculated to generate a report.



Data Flow Diagramming: Guidelines

- All icons must be labeled with meaningful names
- The Data Flow Model evolves through a number of levels of detail
  - Always begin with a context level diagram
  - Always show ***all*** external entities at level 0
- Always label data flow arrows
- Do not represent procedural logic
  - This will be in the detailed PSPECS of each transformation

Steps for Constructing a DFD

- Write a narrative describing each transform
- Parse to determine next level transforms
  - Verb phrases define the transforms to be performed
- "Balance" the flow to maintain data flow continuity
  - Use a 1:5 (approx.) expansion ratio
  - Try to maintain no more than 7 transforms at any given level

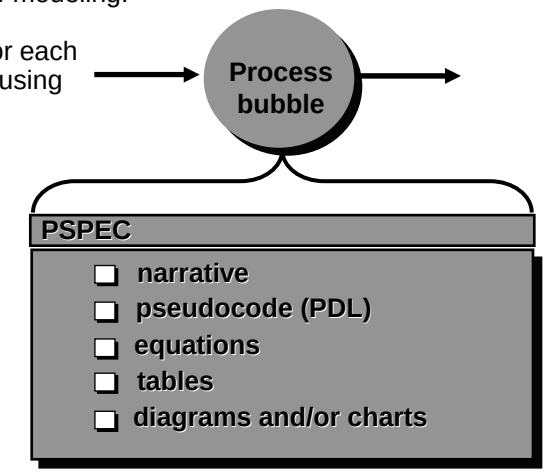
Steps for Constructing a DFD II

- Each bubble is refined until it does just one thing
  - Once a transform is expanded down to a set of atomic transforms then use **Process Specifications (P-SPEC)**
- Data flow item (arrow) may also be expanded
  - Often at higher levels the data flows can be quite complex and the results of multiple transforms being combined

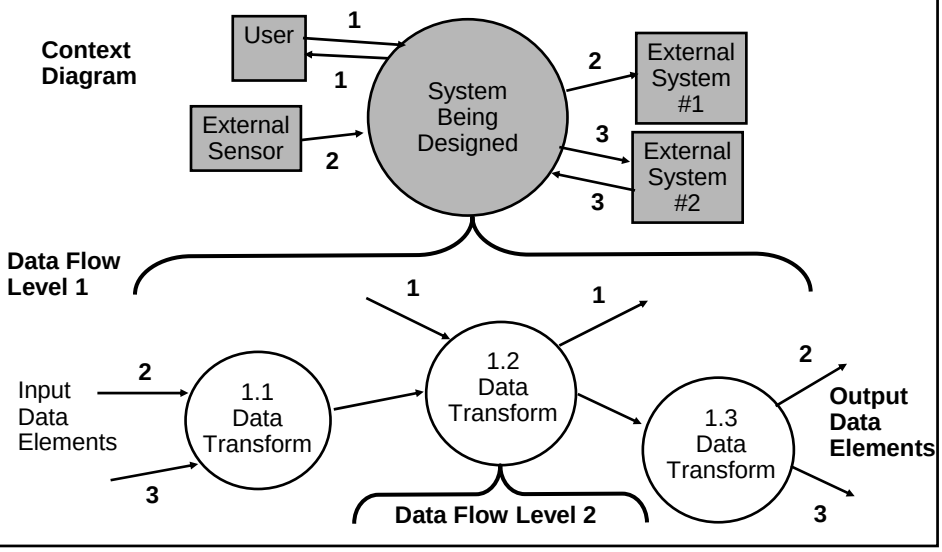
Process Specification (PSPEC)

•The last stage in the Data Flow modeling.

•The required processing for each process is defined in detail using the PSPEC.

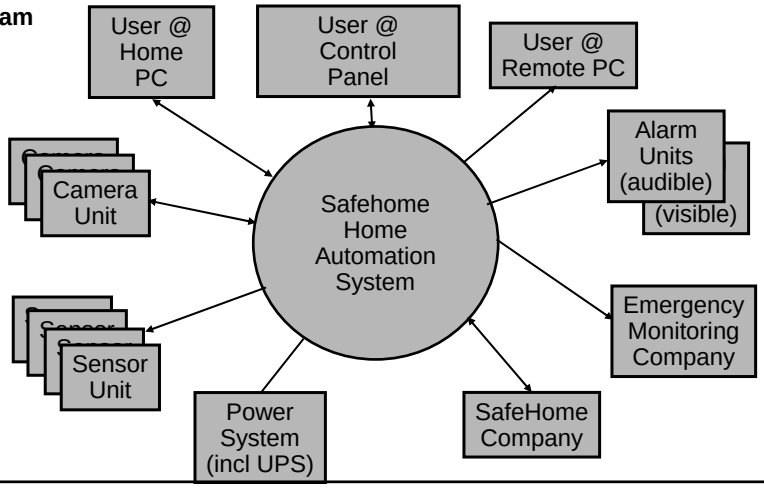


Data Flow Diagram Process



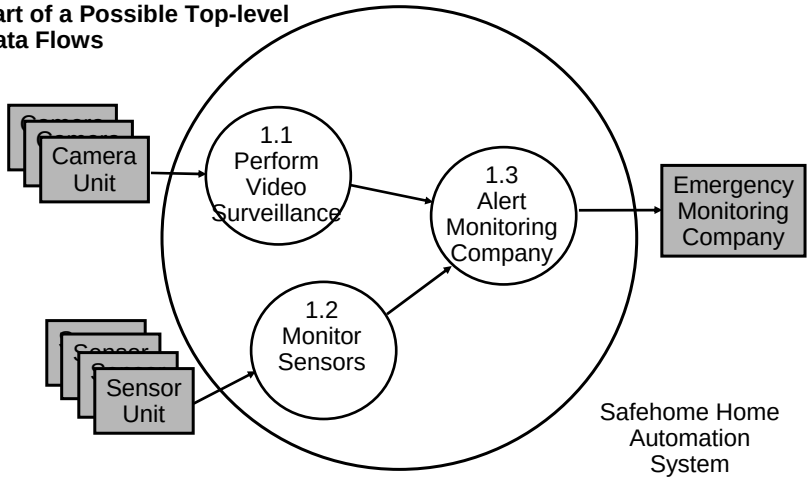
In-class Data Flow Diagram

Recall our Safehome Context Diagram



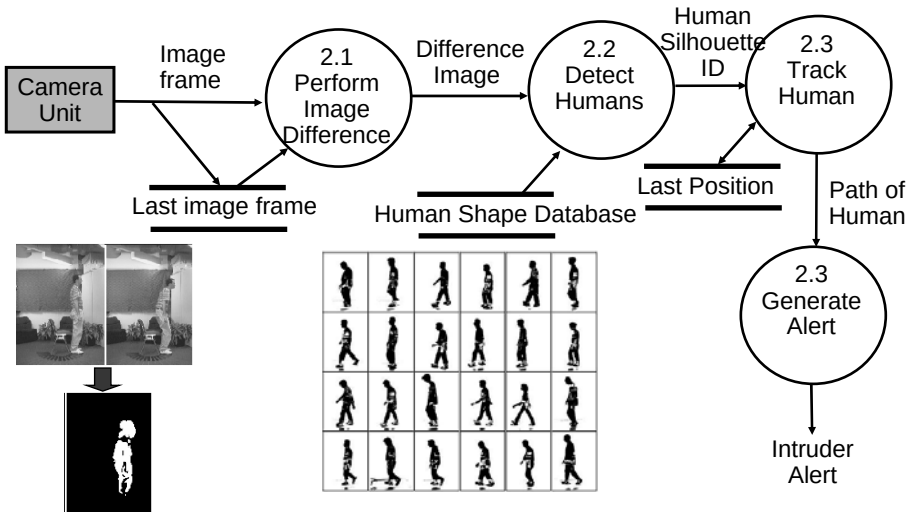
In-class Data Flow Diagram

Part of a Possible Top-level Data Flows



In-class Data Flow Diagram

Now define some data flows: (Perform Video Surveillance)



In-class Data Flow Diagram

Recall our work on the traction control system

1) First lets re-draw the Context Diagram

2) Then try to identify 4-5 transforms that would comprise the next level of the DFD

In-class Process Specification

Now define some of the P-Specs for a couple of these transforms:

For Next Class

- Continue to Study Chapter 8 Sections 8.4, 8.7, and 8.8 in Pressman (OO-based Analysis Modeling)