# Data Flow Diagram

# *Flow-Oriented Modeling*

Represents how data objects are transformed at they move through the system

A data flow diagram (DFD) is the diagrammatic form that is used, considered by many to be an ‘old school’ approach.

Flow-oriented modeling continues to provide a view of the system that is unique.

It should be used to supplement other analysis model elements

# *Flow-Oriented Modeling*

Use a Data Flow Diagram (DFD) to show the relationships among the business processes within an organization to:

- external systems,
- external organizations,
- customers,
- other business processes.

# Flow Modeling Notation

external entity

process

data flow

data store

# The Flow Model

Every computer-based system is an information transform ....

# External Entity

**A producer or consumer of data**

Examples: a person, a device, a sensor

Another example: computer-based system

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

# Process

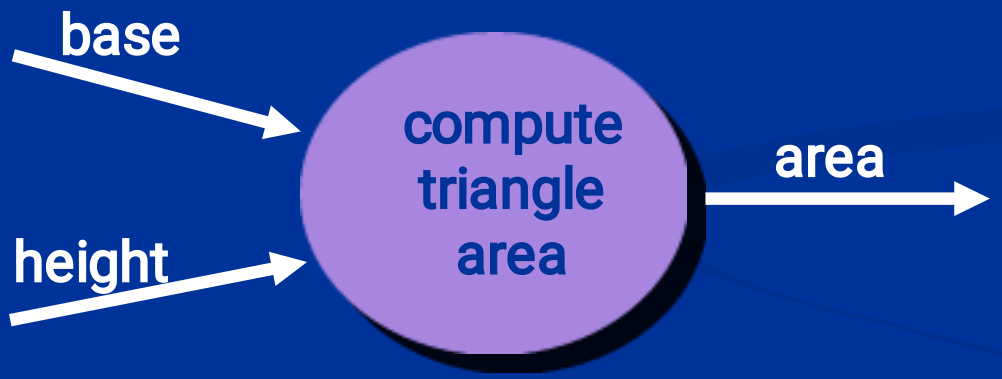
**A data transformer (changes input to output)**

Examples: compute taxes, determine area, format report, display graph

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

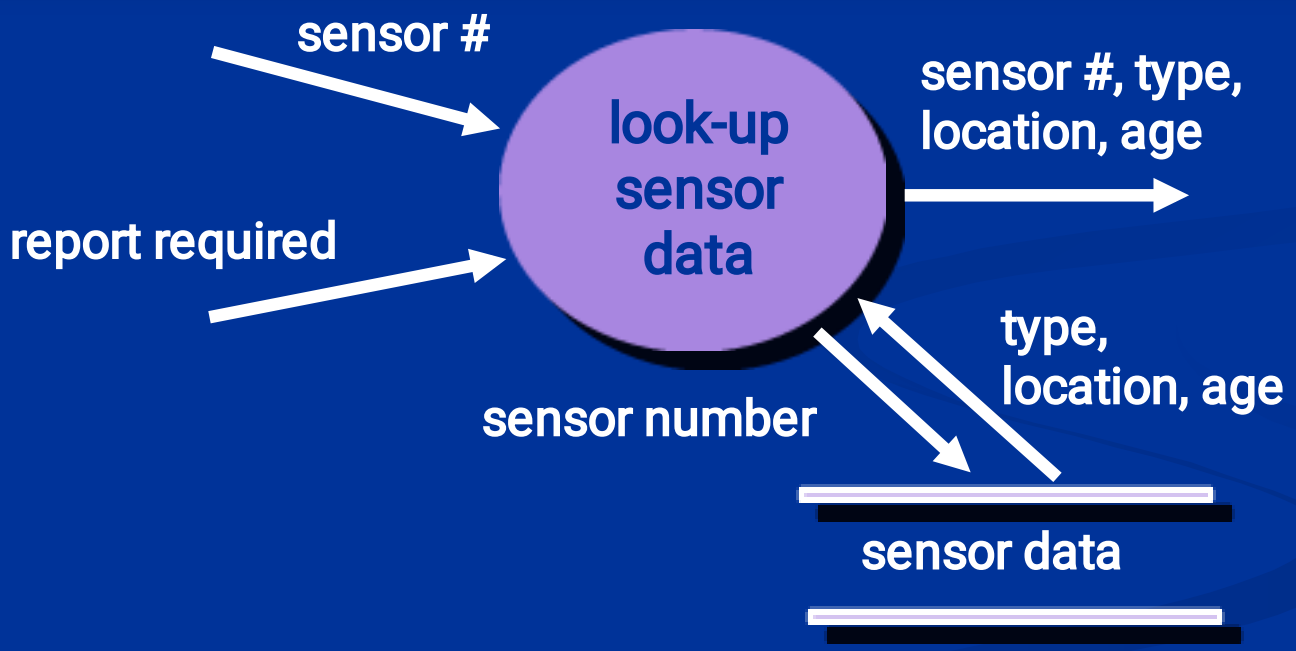
# Data Flow

Data flows through a system, beginning as input and be transformed into output.



# Data Stores

Data is often stored for later use.



# *Methods*

Data flow diagrams are used to describe how the system transforms information.

They define how information is processed and stored and identify how the information flows through the processes.

# *Methods*

When building a data flow diagram, the following items should be considered:

1. where does the data that passes through the system come from and where does it go,

2. what happens to the data once it enters the system (i.e., the inputs) and before it leaves the system (i.e., the outputs),

3. what delays occur between the inputs and outputs (i.e., identifying the need for data stores)

.

# STEPS TO DRAW A DATA FLOW DIAGRAM

1. Start from the context diagram. Identify the parent process and the external entities with their net inputs and outputs.

2·Place the external entities on the diagram. Draw the boundary.

3·Identify the data flows needed to generate the net inputs and outputs to the external entities.

4·Identify the business processes to perform the work needed to generate the input and output data flows.

5· Connect the data flows from the external entities to the processes.

6· Identify the data stores.

# STEPS TO DRAW A DATA FLOW DIAGRAM

7· Connect the processes and data stores with data flows.

8· Apply the Process Model Paradigm to verify that the diagram addresses the processing needs of all external entities.

9· Apply the External Control Paradigm to further validate that the flows to the external entities are correct.

10· Continue to decompose to the nth level DFD. Draw all DFDs at one level before moving to the next level of decomposing detail.

# Modelling Rules

1. All processes must have at least one data flow in and one data flow out.
2. All processes should modify the incoming data, producing new forms of outgoing data.
3. Each data store must be involved with at least one data flow.
4. Each external entity must be involved with at least one data flow.
5. A data flow must be attached to at least one process

# Data Flow Diagramming: Guidelines

- all icons must be labeled with meaningful names
- the DFD evolves through a number of levels of detail
- always begin with a context level diagram (also called level 0)
- always show external entities at level 0
- always label data flow arrows
- do not represent procedural logic

# Level 0 - Context Diagram

- models system as one process box which represents scope of the system
- identifies external entities and related inputs and outputs
- review the data model to isolate data objects and use a grammatical parse to determine “operations”
- determine external entities (producers and consumers of data)
- create a level 0 DFD

# Level 1 - overview diagram

- gives overview of full system
- identifies major processes and data flows between them
- identifies data stores that are used by the major processes
- boundary of level 1 is the context diagram
- write a narrative describing the transform
- parse to determine next level transforms
- “balance” the flow to maintain data flow continuity
- develop a level 1 DFD

# Level 2 - detailed diagram

- level 1 process is expanded into more detail
- each process in level 1 is decomposed to show its constituent processes
- boundary of level 2 is the level 1 process