

Process and Data Modeling

- We talked about an information system being composed of:

Information

Methods & Procedures

People

Technology



- As ISYEs, we don't have to design people and technology.
We just have to select people and the technology to suit a project.
- But we have to design the information storage and retrieval mechanism.
- We have to capture the methods and procedures involved in using, creating, accessing and storing information.
- For capturing information, its structure, and relationship, we have to perform what is known as Data Modeling.
- For capturing methods and procedures, we have to perform what is known as Process Modeling.
- Process Modeling and Data Modeling are two of the most important design steps.

Process and Data Modeling (Cont.)

- What is a data model?

A data model describes the data used, their meaning, relationship with other data, and the rules imposed on them.

Data modeling is the process of developing a data model.

- Why is data modeling needed?

1. It forces system designers to look at data needs and create better system designs.
2. It is necessary for integrating a system with other systems in an organization. Therefore, we need some consistency in data representation, usage, definitions, etc.
3. Data modeling is needed to understand what the user wants and how the system should provide it.
4. It provides a standard.

- What is a process model?

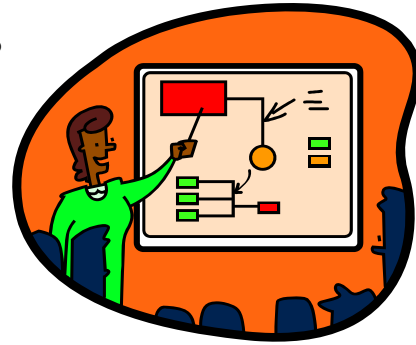
A process model describes the various processes that create, use, and manipulate data in a system.

Process modeling is the methodology for developing a process model.

Process and Data Modeling (Cont.)

- Why is process modeling needed?

1. Unless we have a way of capturing a process, it is difficult to understand what goes on in a process and implement it in a system.
2. Documentation and communication.
3. Understand data modeling needs.
4. Validation purposes.



- Process versus data characteristics

Process Model

1. Dynamic
2. Definition changes frequently
3. Principal user is the programmer
4. Shows movement of data, algorithm, flow of control

Data Model

1. Static
2. Definition changes rarely
3. Principal user is the database designer.
4. Shows data definition and relationships.

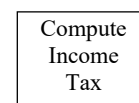
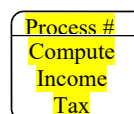
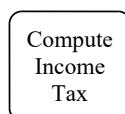
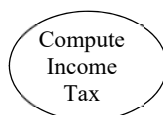
Process Modeling (Cont.)

- For large scale information systems development there are tools available commercially to develop process models and data models.
- We will look at one particular technique for process modeling here.
- This process modeling techniques is known as Data Flow Diagrams (DFD)
- A data flow diagram (or a process model) can lead to a data model. A process model will show you where to begin for developing the data model.

• **Therefore, the process model should be developed before designing a data model.**

- Terminology and notation for drawing data flow diagrams:

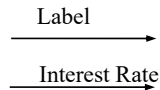
1. Process – a part of the system that transforms inputs into outputs; shows action and the description should include a verb.



We will use the third notation from the left as a standard in this course for representing processes. Note that this notation requires a process # on top and a description in the box below the process #.

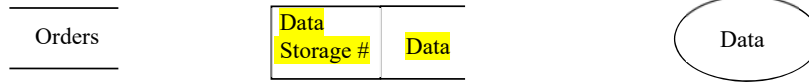
Process Modeling (Cont.)

2. Flow of Data



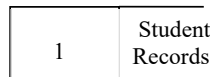
Example:

3. The Store or Data Storage

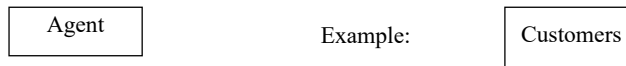


We will use the middle notation for representing data storage which shows a data storage # and a description of data stored.

Example:



4. External Agents (sources and sometimes terminators)



External agents are external to the information system and are usually the source or recipients of information. External agents should be drawn on the periphery of data flow diagrams and external agents should exist only on the Level 0 or Context Diagram (to be explained next)

Process Modeling (Cont.)

• Guidelines for drawing DFDs

1. Choose meaningful names for processes, flows, data stores, and agents.
2. Number the processes and data stores as indicated in the symbols for processes and data storages.
3. Redraw the DFD as many times as possible for both appearance and logic.
4. Avoid overly complex DFDs.
5. Each level should not have more than 3 to 7 processes. But this is only a guideline.
6. Make sure the data flows are appropriate and label all the data flows.

In some cases, the flow may show both read and write and in some cases it may be just read or write.

If it is read and write, you can show two arrows or two sided arrows as appropriate.

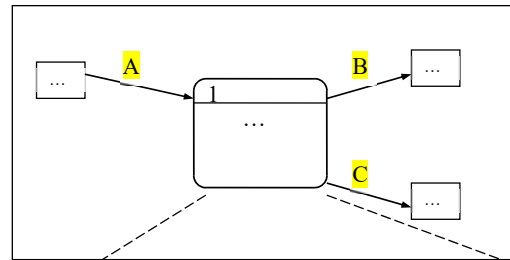
7. It is usually preferable to draw a context diagram to show what the DFD is modeling in the system.

A context diagram should indicate one process that encompasses the entire information system.

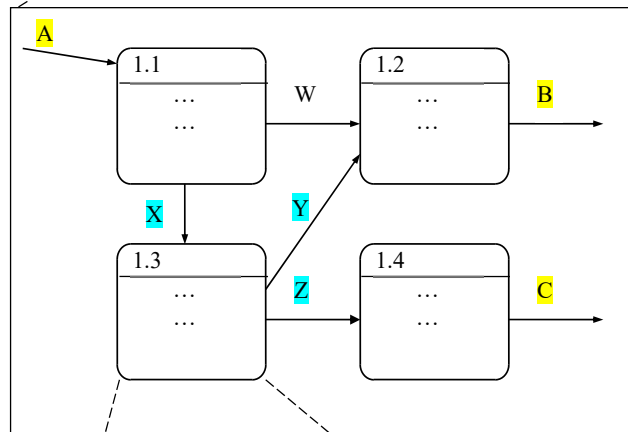
8. The DFD should be balanced.

That is the inputs and outputs at various levels should be consistent.

Process Modeling (Cont.)



Context (Level 0) Diagram example.
Assume the three external agents are named and that Process 1 has a meaningful description written inside the process symbol.



Level 1 Diagram: This is a hierarchical decomposition of Process 1 shown above.

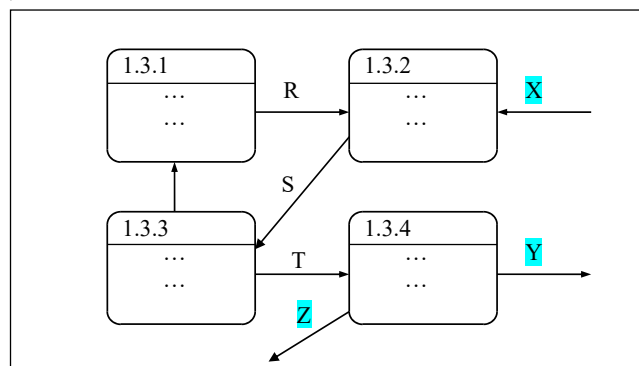
Sub-processes 1.1, 1.2, 1.3, and 1.4 are detailed descriptions of what goes on inside Process 1 shown in the context diagram.

Note the inputs and outputs in the Level 0 diagram match the inputs and outputs in Level 1.

Process Modeling (Cont.)

The Level 2 diagram below shows the details of Process 1.3 in the Level 1 diagram. Note that the inputs and outputs of Process 1.3 match in the Level 0 and Level 1 diagrams.

Level 2



Similarly, there will be level 2 diagrams for processes 1.1, 1.2 and 1.4.

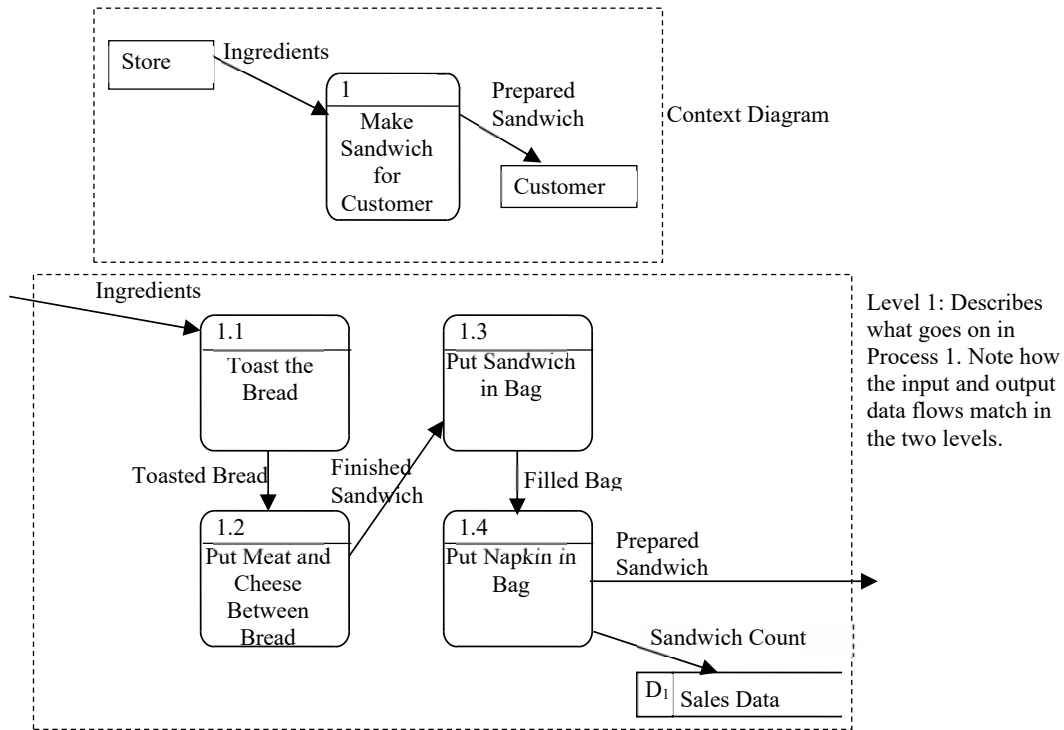
Similarly, decompositions of processes 1.1, 1.2, and 1.4 and their details can be shown.

- An unbalanced DFD may have inputs and outputs shown in one level but not shown in another level.

It is very important to make sure the input and output data flows match at all levels of a DFD.

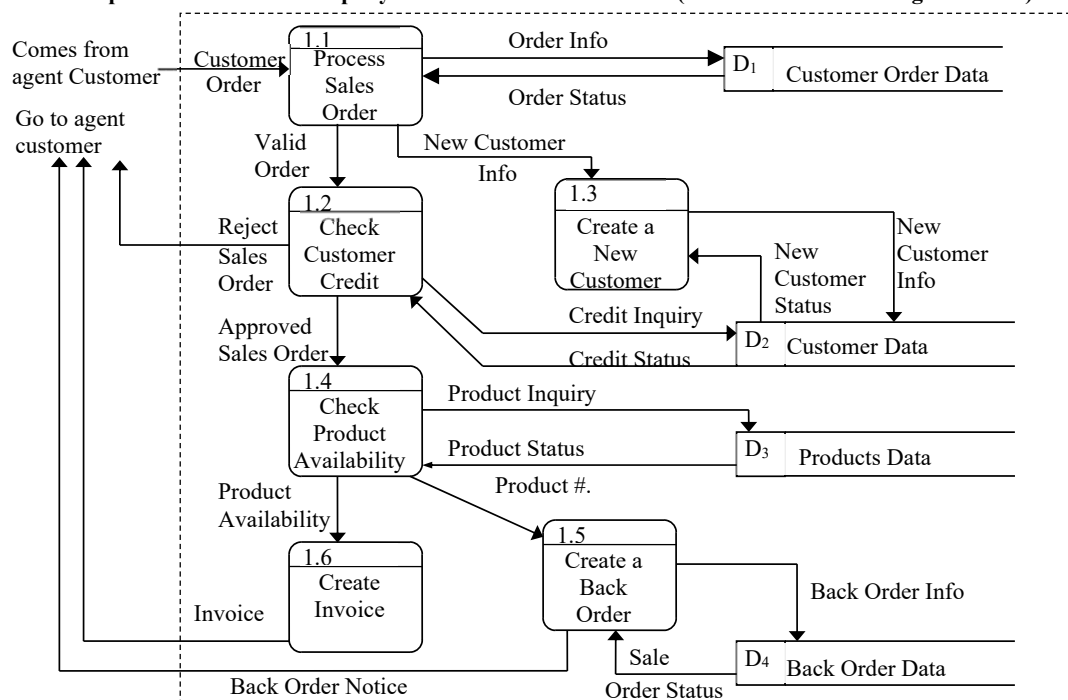
Process Modeling (Cont.)

- A simple DFD:



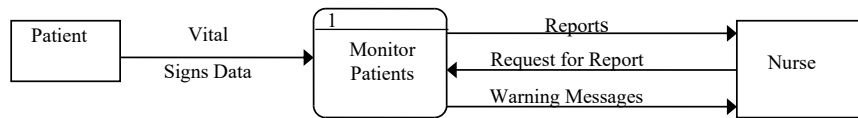
Process Modeling (Cont.)

- Draw a process model of a company's order fulfillment function (assume the context diagram exists):

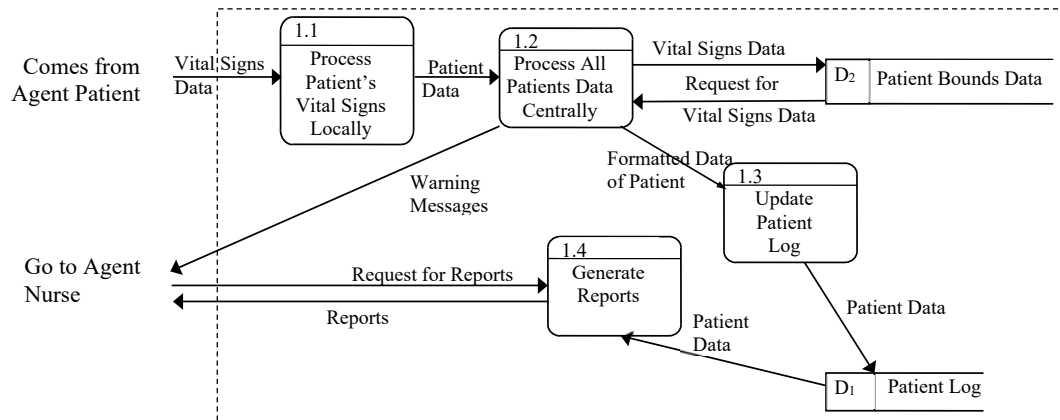


Process Modeling (Cont.)

- Another example of a DFD: Context diagram for a patient monitoring system for a hospital.

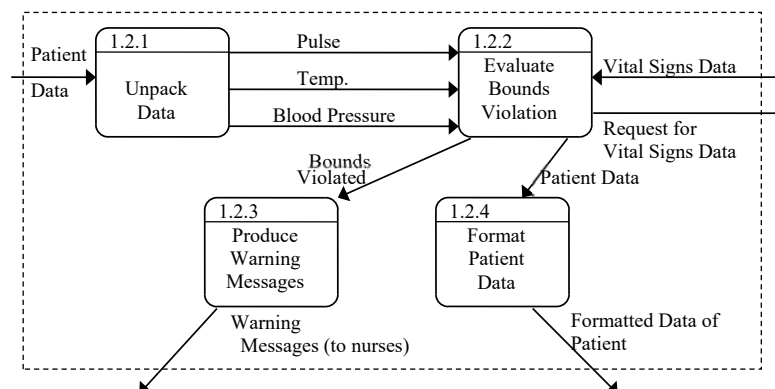


- Process 1 of the above DFD above can be enhanced as follows at the next level (Level 1):



Process Modeling (Cont.)

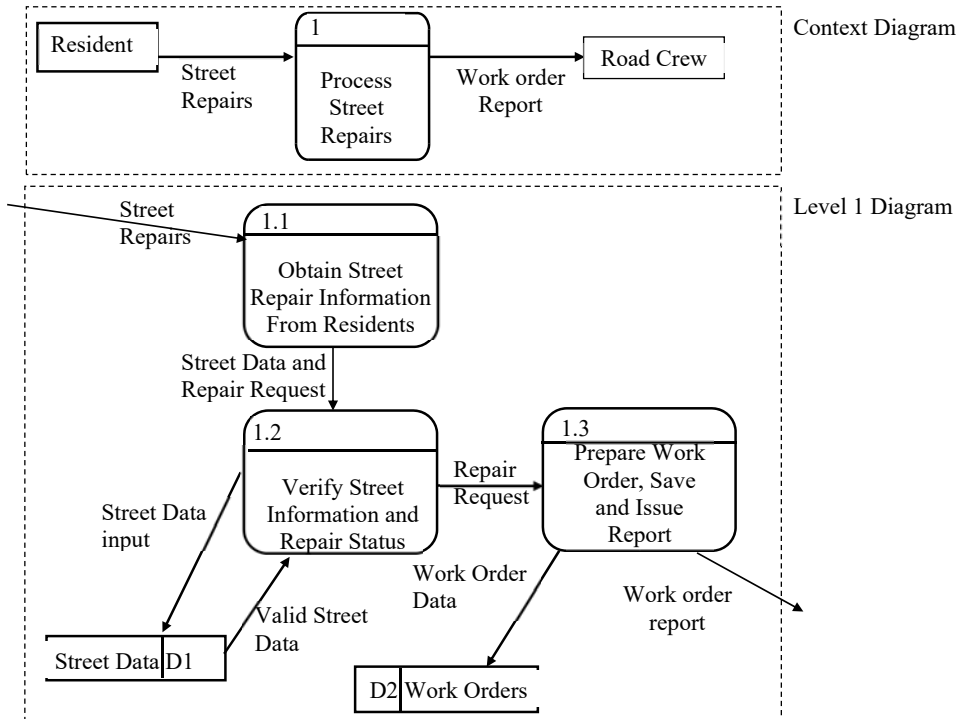
- Process 1.2 can be further elaborated as follows at the next level:



- To develop a data flow diagram or a process model, we need to understand the methods and procedures followed in the system very well.
- The scope of the process model can also dictate the scope of an information system.
- The process model should be validated with people familiar with the methods and procedures since what is modeled in a data flow diagram may not be correct as far as methods and procedures are concerned.

Process Modeling (Cont.)

- Another example of a DFD: Context diagram for a city street road repair needs reporting system



Process Modeling (Cont.)

- Common DFD mistakes

- Process name should be an action (a verb indicating an action).

Data flow between external agents cannot be stipulated.

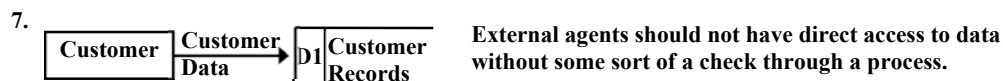
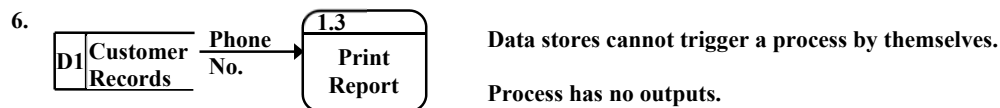
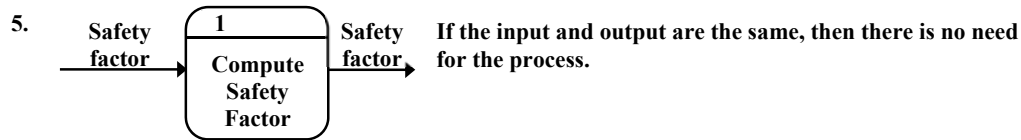
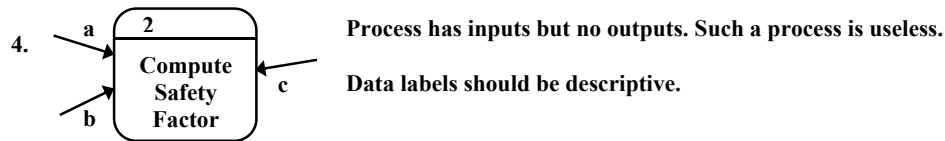
Data labels should be more descriptive than just x, y, and z.
- External agent should be a noun, such as a person or another system, and should not be an action.

Process should have an output.
- Process names are missing.

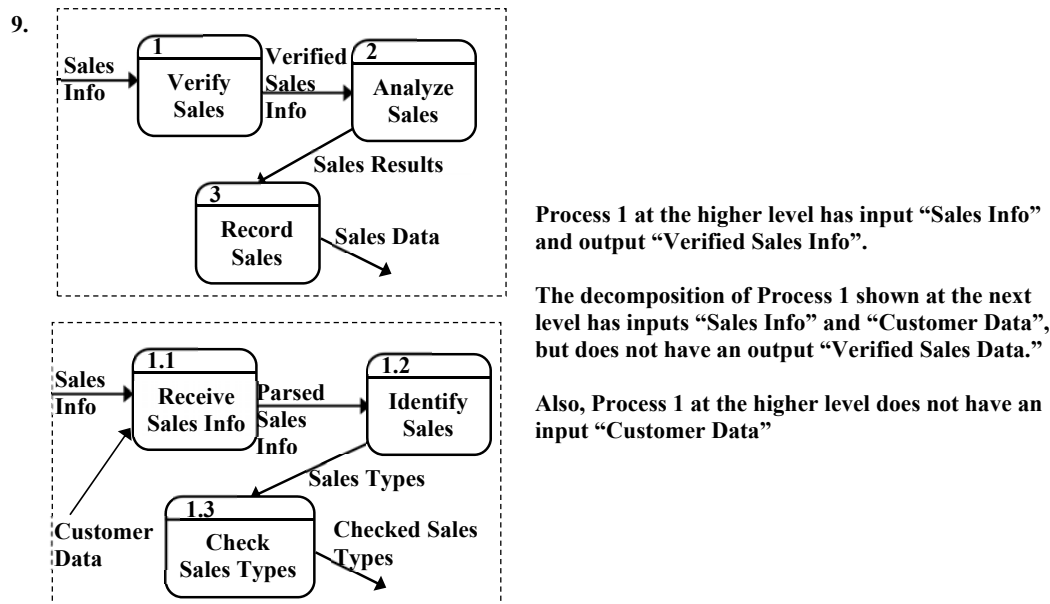
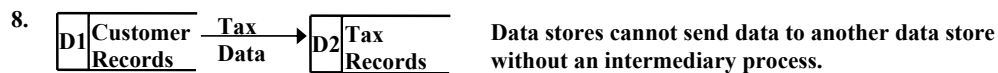
A data store by itself cannot trigger or execute a process.

Processes should have outputs.

Process Modeling (Cont.)



Process Modeling (Cont.)



Process Modeling (Cont.)

- Common DFD mistakes (Cont.)

10. Including external agents in levels other than the Level 0 or Context Diagram. Note: Levels from Level 1 onwards show what happens inside a particular process in the previous level and so an external agent cannot be inside a particular process in the previous level.
11. Drawing a DFD without any data storage. If a DFD does not have a data storage then no data records have to be stored in the database system, and in such case, there is no need for a database system.
12. Repeating the same data storage # in more than one level.
13. Data flows without data labels or data labels that read like process step. The labels can be worded appropriately to avoid this confusion.

For example, if a data flow is for obtaining, say, student ID then instead of writing the data label as “Request Student ID”, which reads like a process, it can be written as “Student ID Request” with the noun appearing first on the label

14. Data storages without labels or labels that read like process step.
15. Data flow diagram without external agents.

Remember: Process modeling (data flow diagram) should be completed before data modeling and the data stores identified in a data flow diagram indicate data needed in the system

Check the web for examples of data flow diagrams and note the symbols are not standardized!