

Data Flow Diagram (DFD)

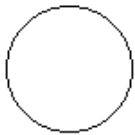
Overview

Data flow diagram (DFD) represents the flows of data between different processes in a business. It is a graphical technique that depicts information flow and the transforms that are applied as data move from input to output. It provides a simple, intuitive method for describing business processes without focusing on the details of computer systems. DFDs are attractive technique because they provide what users do rather than what computers do.

Representation of Components

DFDs only involve four symbols. They are:

- Process
- Data Object
- Data Store
- External entity



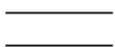
Process

Transform of incoming data flow(s) to outgoing flow(s).



Data Flow

Movement of data in the system.



Data Store

Data repositories for data that are not moving. It may be as simple as a buffer or a queue or as sophisticated as a relational database.



External Entity

Sources of destinations outside the specified system boundary.

Relationship and Rules

Relationship

The DFD may be used for any level of data abstraction. DFD can be partitioned into levels. Each level has more information flow and data functional details than the previous level.

Highest level is Context Diagram. Some important points are:

- 1 bubble (process) represents the entire system.
- Data arrows show input and output.
- Data Stores NOT shown. They are within the system.

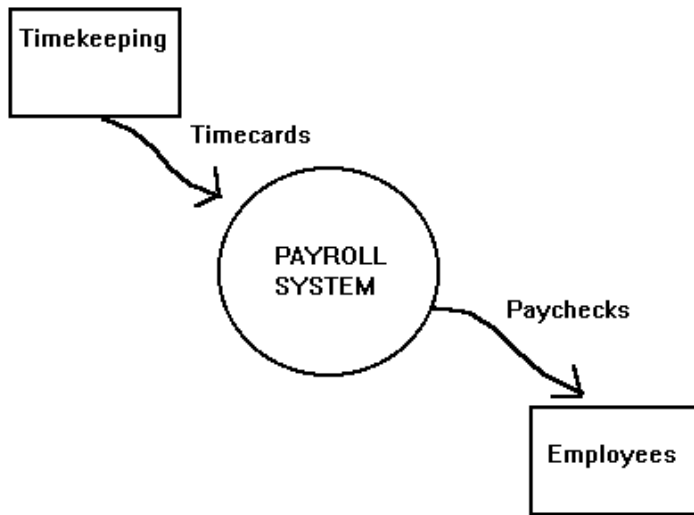


Diagram above is an example of Context Level DFD

Next Level is Level 0 DFD. Some important points are:

- Level 0 DFD must balance with the context diagram it describes.
- Input going into a process are different from outputs leaving the process.
- Data stores are first shown at this level.

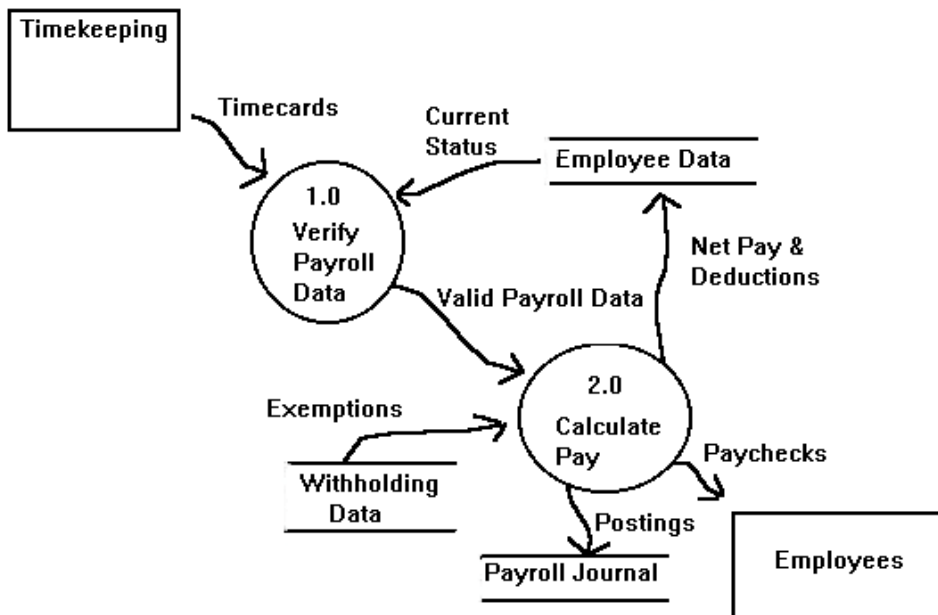


Diagram above show an example of Level 1 DFD

Next level is Level 1 DFD. Some important points are:

- Level 1 DFD must balance with the Level 0 it describes.

- Input going into a process are different from outputs leaving the process.
- Continue to show data stores.

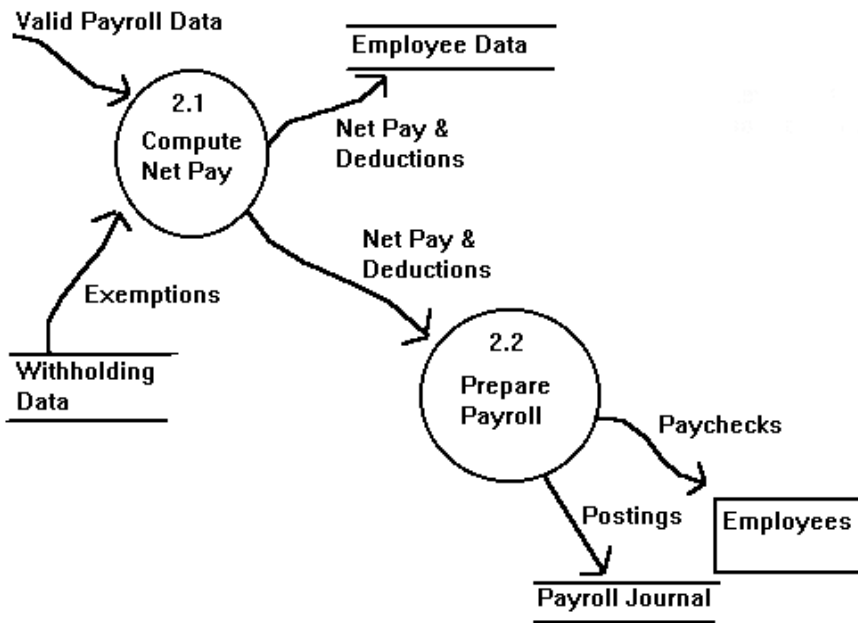


Diagram above show an example of Level 1 DFD

A DFD may look similar to a flow chart. However, there is a significant difference with the data flow diagram. The arrows in DFDs show that there is a flow of data between the two components and not that the component is sending the data that must be executed in the following component. A component in DFD may not continue execution when sending data and during execution of the component receiving the data. The component sending data can send multiple sets of data along several connections. In fact, a DFD node can be a component that never ends.

Rules

- In DFDs, all arrows must be labeled.
- The information flow continuity, that is all the input and the output to each refinement, must maintain the same in order to be able to produce a consistent system.

Strengths and Weaknesses

Strengths

- DFDs have diagrams that are easy to understand, check and change data.
- DFDs help tremendously in depicting information about how an organization operations.
- They give a very clear and simple look at the organization of the interfaces between an application and the people or other applications that use it.

Weaknesses

- Modification to a data layout in DFDs may cause the entire layout to be changed. This is because the specific changed data will bring different data to units that it accesses. Therefore, evaluation of the possible of the effect of the modification must be considered first.
- The number of units in a DFD in a large application is high. Therefore, maintenance is harder, more costly and error prone. This is because the ability to access the data is passed explicitly from one component to the other. This is why changes are impractical to be made on DFDs especially in large system.

Appropriate and Inappropriate Domain of Application and Example

Appropriate Domain of Application

- DFDs are excellent guide for validating the compatibility of the process and designs of the system. This is because in order to design applications successfully, especially large ones, the design of both the processes and the data stores is important. In addition, the data must be consistent with each other. For example, there must be process to store the data in the data stores and the data stores must supply the data views accessed by the processes. Since DFDs depict the relationships between processes, data store, and dataviews, this made DFD the perfect guide for validating compatibility.
- DFDs are appropriate diagrams for designing high-level application architecture. This is because it is a fact that the larger the application is to be developed the more important the architecture is. For example, building a box does not need an architect but a 10-story building does. In most architectural design, they are represented as diagrams because diagrams are the best way to depict multiple relationships among multiple components. This is applicable to software design, too and DFDs helps tremendously in showing the architecture design of the system r application.
- DFDs are especially useful for depicting system flow charts. DFDs are used to show the flows of data among batch-job steps.

Inappropriate Use

DFDs are inappropriate to use in a large system because if changes are to be made on a specific unit, there is a possibility that the whole DFD need to be changed. This is because the change may result in different data flow into the next unit. Therefore, the whole application or system may need modification too.

Tools Related to DFD

Data Flow Diagram Tool (DFDT) is one of Integrated Software Software Development System (ISDS) that enforces Software Engineering Principles. DFDT is set to be the second most important tool after Project Management Tool (PMT) in ISDS. DFDT contains processes, data flows, external entities and data store. In order to design a consistent DFD, there are some rules that need to be followed in DFDT.

- In Context Diagram, the process could be considered as the project itself.
- In Level 0 of DFD, the processes could be considered as the module(s) in the system.
- In Level 1 of DFD, the processes could be considered as the sub-module(s) of function(s) of the project or module.
- Level 2 or so on, similar to Level 1.

Other tools and software that are realted with DFDs modeling tools are:

- **Information Based Modeling Tool (IBMS)**

IBMS is a educational freeware CASE Tool for database design. It is based on its own modeling methodology - The Two-Stage Entity-Relationship (TSER) approach which comes with a DFD modeling.

- **FirstSTEP**

FisrtSTEP is a business Management Tool Software that allows allowing decision makers the power to design, document, analyze, refine, incrementally improve and redesign business processes.

Beside these, other CASE Modeling Tools that support data flow diagram modeling are:

- ER/Studio
 - ERwin
 - Infomodeler
 - Oracle Designer/2000
 - Power Desingner
-

