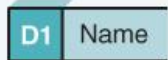
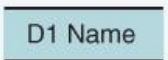


DFD Diagrams summary

B HOLLAND

The main elements

Data Flow Diagram Element	Typical Computer-Aided Software Engineering Fields	Gane and Sarson Symbol	DeMarco and Yourdon Symbol
Every <i>process</i> has - a number - a name (verb phrase) - a description - at least one output data flow - at least one input data flow	Label (name) Type (process) Description (what is it) Process number Process description (structured English) Notes		
Every <i>data flow</i> has - a name (a noun) - a description - one or more connections to a process	Label (name) Type (flow) Description Alias (another name) Composition (description of data elements) Notes		
Every <i>data store</i> has - a number - a name (a noun) - a description - one or more input data flows - one or more output data flows	Label (name) Type (store) Description Alias (another name) Composition (description of data elements) Notes		
Every <i>external entity</i> has - a name (a noun) - a description	Label (name) Type (entity) Description Alias (another name) Entity description Notes		

The Process Symbol

A process is an activity or a function that is performed for some specific business reason.

Processes can be manual or computerized when described in our logical DFD.

Every process should be named starting with a **verb and ending with a noun** (e.g., “**Create shop work order**”).

Names should be short, yet contain enough information so that the reader can easily understand exactly what they do.

In general, each process performs only one activity, so to avoid using the word “and” in process names because it suggests that the process performs several activities. I

In addition, every process must have at least **one input data flow and at least one output data flow**.

The Data Flow

A data flow is a single fact, such as Send in order ID (sometimes called a data element), or a logical collection of several facts (e.g., new shop work order).

Every data flow should be named with a **noun**. The description of a data flow lists exactly what data elements the flow contains. For example, the order details data flow would list customer information, order date, and the custom components included in the order as its data elements.

Data flows are the glue that holds the processes together. **One end of every data flow will always come from or go to a process**, with the arrow showing the direction into or out of the process.

Data flows show **what inputs go into each process and what outputs each process produces**.

Every process must create at least **one output data flow**, because if there is no output, the process does not do anything. Likewise, each process has at least one input data flow, because it is difficult, if not impossible, to produce an output with no input.

Data Store

A data store is a collection of data that is stored in some way (which is determined later when creating the physical model). Every data store is named with a noun and is assigned an identification number and a description.

Data stores form the starting point for the data model and form the logical connection between the process model and the data model.

Data flows coming out of a data store indicate that information is retrieved from the data store. For example we could say Retrieve Order Details indicating data flow that we have retrieved the Order Details data flow from the Custom Orders data store.

Data flows going both into and out of a data store indicate that information in the data store is change.

All data stores must have at least one input data flow or else they never contain any data.

Interface / External Entity

An external entity is a person, organization, organization unit, or system that is external to the system, but interacts with it (e.g., customer, clearinghouse, government organization, accounting system).

The external entity typically corresponds to the primary actor that we identified in the as a system user.

External entities **provide data to the system or receive data from the system**, and serve to establish the system boundaries.

Every external entity has a name and a description. The key point to remember about an external entity is that it is external to the system, but may or may not be part of the organization.

People who use the information from the system to perform other processes or who decide what information goes into the system are documented as external entities (e.g., managers, staff).

Context Diagram

The top-level DFD in every business process model, whether a manual system or a computerized system, is the context diagram. As the name suggests, the context diagram shows the entire system in context with its environment.

All process models have one context diagram.

The context diagram shows the overall business process as a single process (i.e., the system itself) and shows the data flows to and from external entities.

Data stores are not included on the context diagram.

If the system uses data from or supplies data to another independent information system, that system should be shown as an external entity, not a data store.

Level 0 DFD

The next level in the DFD hierarchy is the diagram called the level 0 diagram or level 0 DFD .

The level 0 diagram shows all the major processes at the first level of numbering (i.e., processes numbered 1 through n (usually no more than 6 or 7)).

The data stores, external entities, and data flows among them. The purpose of the level 0 DFD is to show all the major high-level processes of the system and how they relate to each other and to stored data. **All process models have one and only one level 0 DFD.**

The context diagram deliberately hides some of the system's complexity in order to make it easier for the reader to understand. Only after the reader understands the context diagram does the analyst “open up” process 0 to display its internal operations by decomposing the context diagram into the level 0 DFD, which shows more detail about the processes, data stores, and data flows inside the system.

Another key principle in creating sets of DFDs is balancing. Balancing means ensuring that all information presented in a DFD at one level is accurately represented in the next-level DFD.

Level 1 DFD

In the same way that the context diagram deliberately hides some of the system's complexity, so does the level 0 DFD.

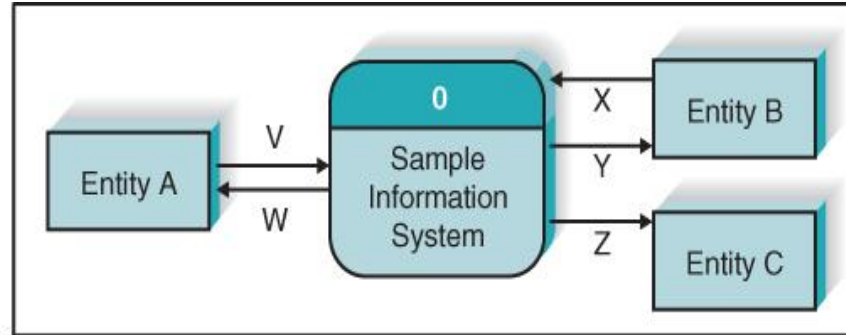
The level 0 DFD shows only how the major high-level processes in the system interact. Each process on the level 0 DFD must be decomposed into a more explicit DFD, called a level 1 diagram, or level 1 DFD, which shows how it operates in greater detail.

In general, all process models have as many level 1 diagrams as there are processes on the level 0 diagram; every process in the level 0 DFD would be decomposed into its own level 1 DFD.

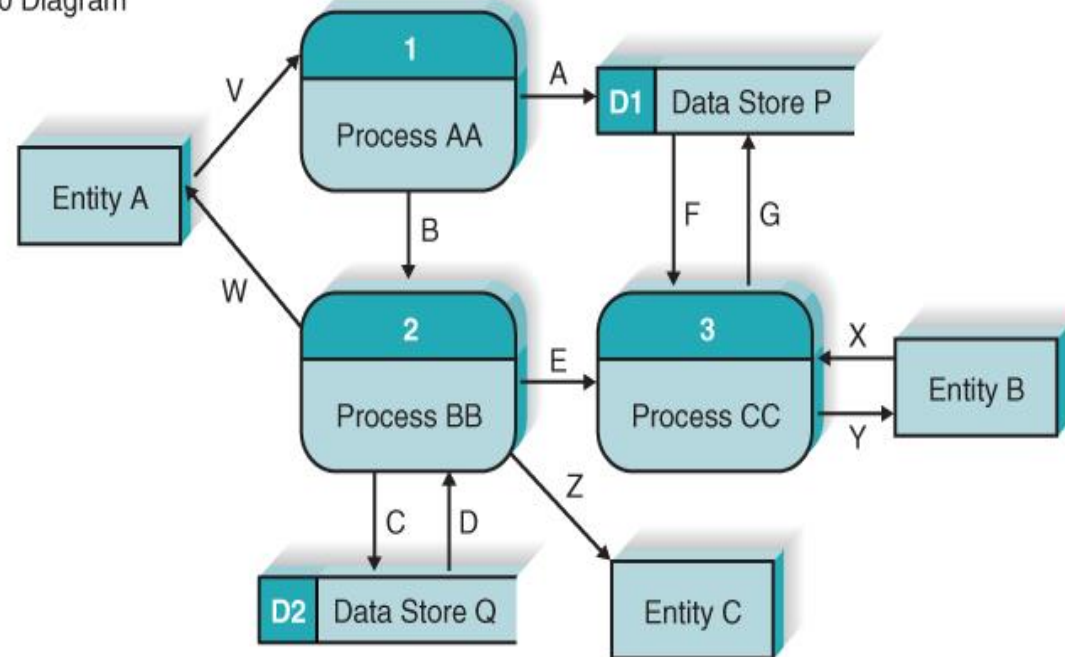
Once again, it is very important to ensure that the level 0 and level 1 DFDs are balanced.

Example Diagram layout

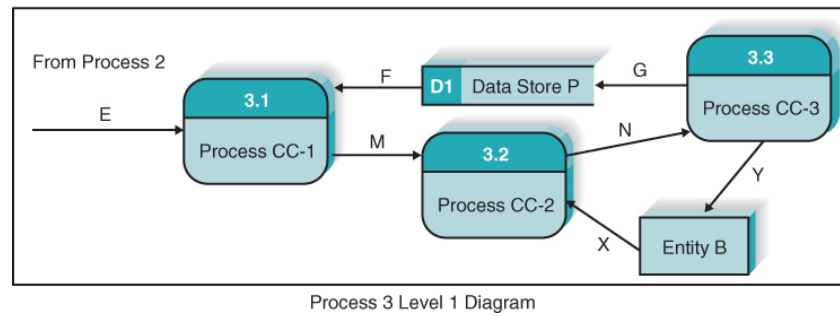
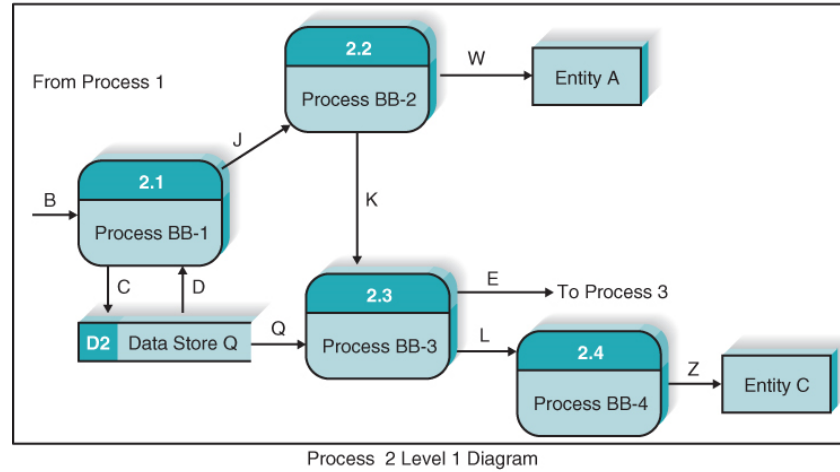
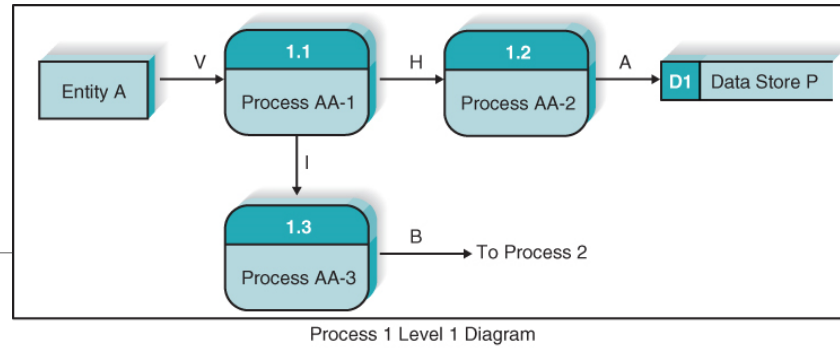
Context
Diagram



Level 0 Diagram



Example Level 1 DFDs



Reference

