

## Lesson 8

# Data Flow Diagrams

### OVERVIEW

As described in Lesson 4 – Structured Modeling Techniques, a data flow diagram (DFD) is used for process modeling. This modeling technique shows the flow and transformation of the data without regard to the details of the data structure or type. It clearly represents where the transactions and transformations occur in your system.

A DFD is not the same as a flow chart, although there are certain similarities. A flow chart is much less specific with regard to how pieces of data are broken down, combined, and moved around the system than is a DFD. On the other hand, a flow chart is much more specific and physical than a DFD with regard to how processing is performed. A data flow diagram is more flexible and has a more general applicability than does a flow chart.

Data flow diagramming is not designed to show materials flow, just data. For example, if you were modeling a bookstore, how all of the receipts, invoices, inventory counts and financial transaction items are handled would be shown on your diagrams; but the books themselves would not. The books are materials, and their movement from the publisher to the store's loading dock to the shelves to the bag in the customer's hand is materials flow and not a part of data flow diagramming.

In any structured analysis methodology, the first task is to draw a top-level diagram, a simple summary of the overall system. It shows the system environment and major inputs and outputs, and is sometimes referred to as the basic problem statement. This is usually much less specific than the way most people picture a system because so many details are omitted. It should involve only one, two, or three processes and a very few external entities (source/sinks). In the example that follows, you use only one process and two source/sinks, though a top-level diagram could contain a few more of each. You break down (decompose) these top-level elements into more specific processes and flows. Some methodologies and analysts like to use a single process to represent the highest level of the data flow diagram. This is called a context diagram, and only one process is allowed on a context diagram to designate the entire system. For child diagrams, though, you can have multiple processes on any diagram.

The idea behind creating a general top-level diagram is twofold:

- To ensure agreement and understanding of the fundamental, overall mission of the system. There is confusion on this more often than is realized, and the details can rarely work well if the overall mission is unclear.
- To make explicit the source and derivation of the more detailed operations of the system. Often it is the second or third level of design that is the taken-for-granted starting point. Making the derivation explicit is important both for the design discipline itself and for the completeness of the resulting documentation. If you start Visible Analyst at the highest level, the tracking of all subsequent derivations automatically results from the data repository documentation.

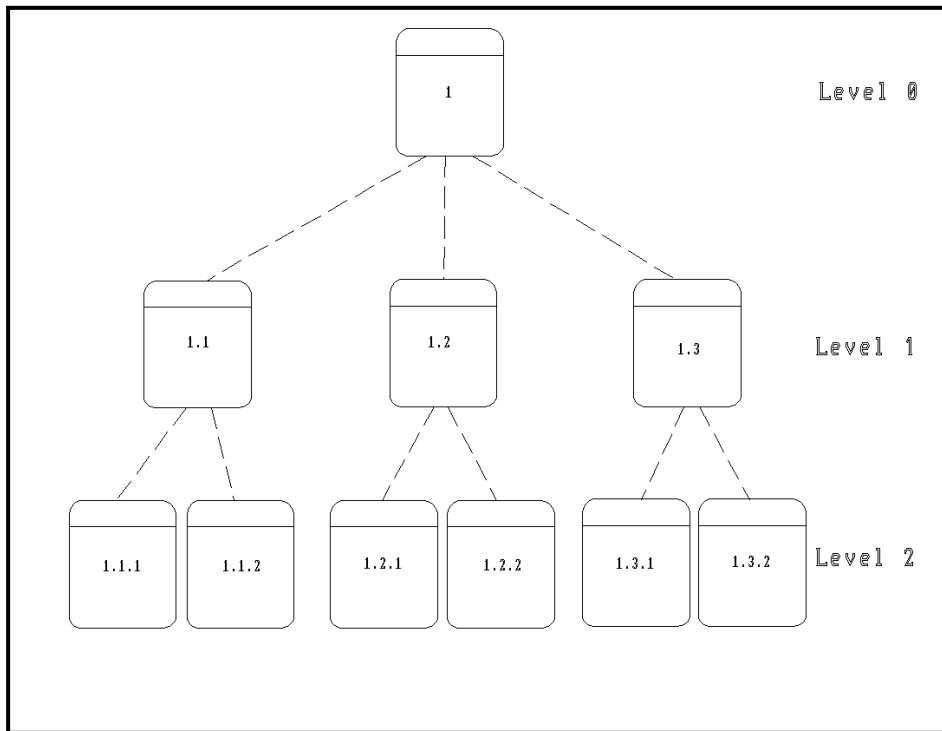
There are four meaningful objects that appear on data flow diagrams:

### *Process*

The process symbol is accessed with the first symbol button on the control bar. If you are using Yourdon rules, a process is represented by a circle. For Gane & Sarson rules, a process is represented by a rounded-corner square.

A process signifies that something is happening to transform data. At the highest level you could show the whole bookstore as a single process.

After creating the context (or high-level) diagram, you then break that diagram down into processes representing the various departments of the store, then into processes representing the functions of the departments, then into subdivisions of these processes, and so forth to as fine a level of description as you wish. This is done by “nesting” or decomposing a process and creating a child diagram at a greater level of detail, one that shows all of the inputs and outputs to the parent process and allows you to show what is going on inside it. Processes have numbers, and those numbers reflect the decomposition hierarchy, as shown in Figure 8-1.



**Figure 8-1 A Process Numbering Scheme**

*Data Store (or File)*

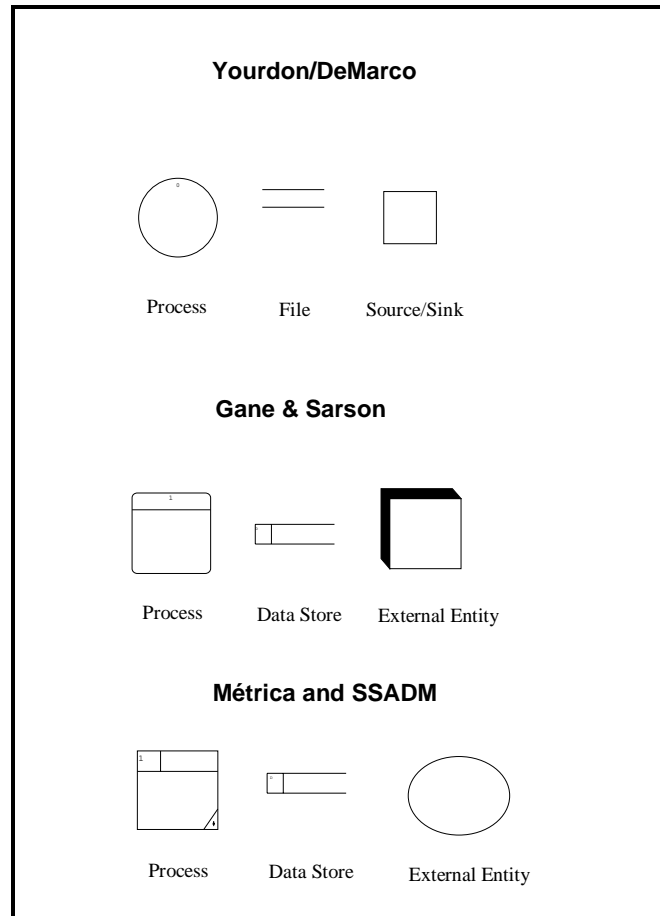
A data store or file is accessed using the second symbol button on the control bar. If you are using Yourdon rules, a file is represented by two horizontal parallel lines. For Gane & Sarson rules, a data store is shown as a rectangle with the right side open, and it has a number. A data store is a place where data is kept while it is not actively being processed. Your process model does not show how it is stored, whether encoded on magnetic disk or scribbled on the back of an envelope, just that it is stored. Data can only enter a data store from a process and can only leave a data store to a process. The detailed data element in a data store can be defined in the Visible Analyst repository.

### *External Entity (or Source/Sink)*

An external entity is accessed using the last symbol button on the control bar. It is represented by a large square under Yourdon rules, or a square drawn with relief under Gane & Sarson rules. An external entity is something outside the boundary of the system you are modeling that either sends data to your system or receives data from it. It is effectively a black box, in that what happens inside the external entity is not material to your system description. It is only there to make clear some of the environment in which your system resides. External entities are optional. A net input data flow can just as well be shown coming from nowhere as from an external entity. Note that an external entity has no relation to the entity that is a part of entity relationship modeling. It is simply an unfortunate duplication of terminology.

### *Data Flow*

A data flow depicts the movement of one to many items of data. Data can enter a system from outside, such as the entries that appear on a publisher's invoice or a packing list. (The invoice data flow is shown entering a process—it *must* enter a process—where it is examined and acted upon.) This process might send some of the data to be stored, some to be printed, some to be ignored. These invoice data elements may or may not be combined with elements from other input data flows and may then exit the process as parts of other data flows. To draw a data flow line, click on a line type in the control bar.



**Figure 8-2 Data Flow Diagramming Symbols**

**Note**

- ❏ In Yourdon methodology, names of data flows contain hyphens instead of spaces. When you enter a space in a data flow name, Visible Analyst uses a hyphen.

This lesson leads you through the diagram creation process for a Gane & Sarson-based process model. Basic drawing and decomposing a process into a subordinate “child” diagram are shown. Also, you see how the system is validated using the rules capabilities of Visible Analyst. You build errors into your diagram to demonstrate the types of errors that can be identified by the Analyze function.

## CREATING AND POPULATING A TOP-LEVEL DIAGRAM

The basic procedure for creating a top-level DFD is the same as creating a new diagram for the unstructured diagram type. The only difference is that if you choose a context diagram, by clicking the box at the bottom of the New Diagram screen, a process symbol number 0 is automatically placed on the diagram; and you are prompted for its name. A context diagram is permitted only one process symbol. You can add data flows and other symbols to the diagram.

This diagram has already been created for you so that you do not have to draw the diagrams and enter repository information. It is named DMV System and is shown in Figure 8-9, the top-level diagram of the DFDs you spawned from your FDD. This diagram also has one child diagram called Driver's Licensing System. You can display a list of diagrams by selecting **Open Diagram** from the **File** menu, or by clicking the **Open** button on the control bar. When a diagram type has a plus sign next to it, it means that diagrams of that type have been created. Click the plus sign to display the list of existing diagrams, then double-click the diagram you would like to open. (You can hide the list again by clicking the minus sign next to the diagram type name.)

To close a diagram, click the control button in the top left corner of the diagram window and select **Close**, double-click the control button, or choose **Close Diagram** from the **File** menu.

## NESTING A PROCESS

In this unit, you structurally decompose a process symbol. This is also called “nesting” or “exploding” a process. The **File** menu contains the **Nest** function for this purpose; the submenu contains the **Explode** function. **Explode** can also be found by clicking the *right* mouse button on a process symbol that you want to model in more detail to display its **Object** menu, and then selecting **Explode**.

If the process has not previously been decomposed, this generates a “child” diagram from this “parent” process. All of the data flows attached to the parent process are automatically “dragged down” to the child diagram by the **Nest** function. These flows can be attached to the lower-level processes that you create on the child diagram. Those lower-level processes can then be nested further to increase the level of detail. In the current example, the child diagram was created by the **Spawn** function that you executed in Lesson 6 – Functional Decomposition Diagrams, and the processes you added to the FDD were placed on it.

*Open the Diagram:*            1            From the **File** menu or the open diagram button on the control bar, open the data flow diagram DMV System, if it is not still open from a previous lesson. This is the context diagram for this project.

- Select a Process:* 2 Click the right mouse button on the process Driver's Licensing System to open its **Object** menu, and choose **Explode**. This opens the existing child diagram Driver's Licensing System and is an alternate way to navigate between the diagrams of your project, avoiding the **File** menu.
- Explode It:* 3 Click the *right* mouse button on the process labeled Issue License and choose **Explode**. The flows attached to the parent process are dragged down to the spawn-created diagram entitled Issue License, where the three process symbols from the functional decomposition diagram were placed. Maximize the diagram. The dragged-down flows are lined up on the sides of the child diagram, input flows on the left, output flows on the right. (See Figure 8-3.) Since you did not move the symbols on the diagram before nesting, it is possible that the dragged-down flows were drawn over a symbol.

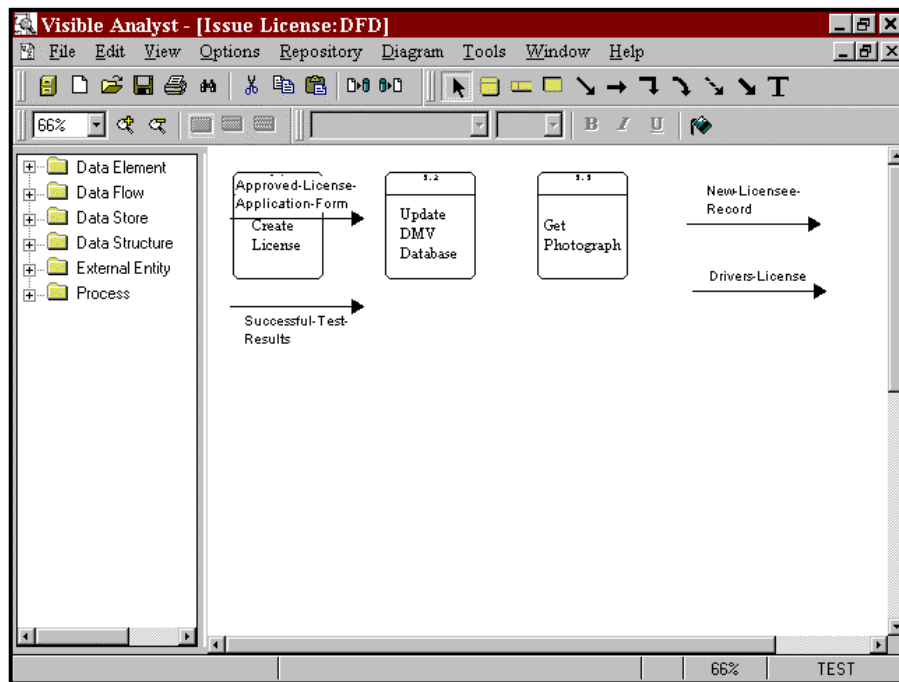


Figure 8-3 Child Diagram With Dragged-Down Flows

*Edit the Diagram:* 4 Move the symbols and attach the flows as shown in Figure 8-4. To move a symbol, click and drag it with the *left* mouse button. To attach the lines, click one endpoint. Then click the *left* mouse button on the middle of the data flow and drag the line so that it is positioned correctly. When the data flow is selected, it changes color; and the line becomes a dashed line as it is moved on the diagram. Do the same for the other data flows. (Ignore for now the other flows you see in Figure 8-4; you add them later in this section.)

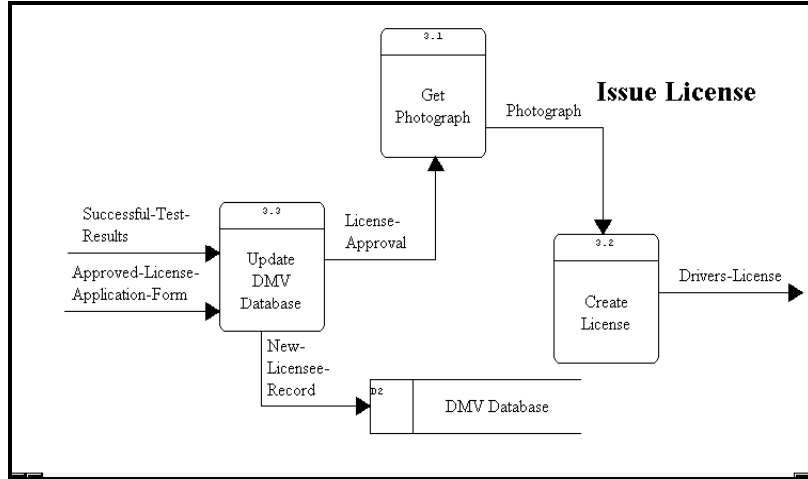


Figure 8-4 The Completed Diagram

**Note**

- When you want to show a data flow line (or another type of line) as attached to a process (or another type of symbol), you must drag the end of the line into contact with the symbol. With the Auto Connect option on, Visible Analyst redraws your connection at the outer edge of the symbol.

*Add Flows and Text:* 5 Add two new data flows: License-Approval and Photograph. Since these are internal flows, as opposed to net input or net output data flows<sup>7</sup> appearing on the parent, the Nest function could not create them.

<sup>7</sup>For a full explanation of net input and output flows, please see the Visible Analyst *Operation Manual* or the online help.



- Change Line Format:*      6      You may want to change the data flow lines from straight to elbow. This can be done by highlighting the line and then selecting **Line Settings** from the **Options** menu and changing the line orientation to elbow. Or click the elbow line button on the control bar before drawing the lines.
- To change the orientation of an elbow, position the cursor over the line segment handle to change and click, but do not release, the left mouse button. Move the mouse slightly until the line changes from solid to dashed, and then press the right mouse button. Release the left mouse button to save the change.
- 7      Click the **T** button on the control bar to add the caption text “Issue License” to display the diagram title on the diagram. Note that there is a way to do this automatically by using boilerplates. You can read about this in the *Visible Analyst Operation Manual* or in the online help system. (Boilerplates are not available in the Education Editions of Visible Analyst.)
- Add a File:*              8      If you wish, you can add the file **DMV Database** to the diagram. Since it appears on the context diagram, this is not necessary, but some people feel that showing it on a lower-level diagram adds clarity. Move the flow **New Licensee-Record** to attach it to the file **DMV Database**. A symbol is considered attached to a line when the endpoint of the line is touching the edge of the symbol. (It does not automatically connect to the symbol.)
- Save:*                      9      Select **Save** from the **File** menu.

## CREATING A NEW DIAGRAM

In previous sections of this lesson, you worked with diagrams that were either supplied by Visible Systems or created by the **Spawn** function. In this unit, you create and populate a new diagram yourself and practice more data flow diagramming techniques. You structurally decompose the process **Administer Road Test** (drawing a level-three diagram) that details what occurs within the process **Administer Road Test**.

If you have any other diagrams open, you should maximize the window by clicking the **Maximize** button in the upper right corner of the window.



- Save:* 7 Select **Save** from the **File** menu and click OK. The title of your diagram defaults to the name of the parent process. Visible Analyst indicates that it is saving *both* diagrams. This is because they are involved in a nest relationship and both ends of the nest relationship must be saved in the repository.


### **Adding Processes to a Child Diagram**

Now add processes to the child diagram named Administer Road Test. The processes contained in this diagram are the individual processes that make up the parent process Administer Road Test. This diagram is a more detailed representation of the transformations and interactions that occur to the data flows within the parent process.

- Add Processes:* 1 Click the first symbol button, process, in the control bar.
- 2 Add and label three processes: Validate Applicant, Test Vehicle Knowledge, and Test Driving Capabilities.
- Save:* 3 Select **Save** from the **File** menu.

### **Attaching Data Flows to Symbols**

The input data flows on the left side of the diagram and the output data flows on the right side of the diagram were dragged down to the child diagram with the **Nest** function. It is necessary to attach the data flows to the appropriate processes on the child diagram. To attach a flow to a symbol:

- Select a Line:* 1 Put the cursor in selection mode by clicking the  button on the control bar.
- 2 Select the data flow Learners-Permit. The line handles appear.
- Drag It Into Position:* 3 Drag it to the edge of the process symbol labeled Validate Applicant, as shown in Figure 8-6.
- Repeat for Other Flows:* 4 Attach the other dragged-down flows as shown in Figure 6-6.
- Add New Flows:* 5 Click the first line button on the control bar.

- 6 Add a flow from process Validate Applicant to Test Vehicle Knowledge and label it “Valid-Applicant.”
  - 7 Click the straight line button on the control bar and add an input flow into the process Test Driving Capabilities and label it “Test-Criteria.”
  - 8 Add a flow from process Test Vehicle Knowledge to process Test Driving Capabilities, but leave it unlabeled by clicking Cancel or pressing ESC when you are prompted to enter a name. (This deliberate error is added to demonstrate the capabilities of the Analyze function.)
  - 9 Add an unattached data flow labeled “Driving-Criteria.” (Remember, you must double-click to end the line when it is not attached to a symbol.) This demonstrates the ability to select an existing flow from the diagram when a flow is split.
- Save:*
- 10 From the File menu select Save.

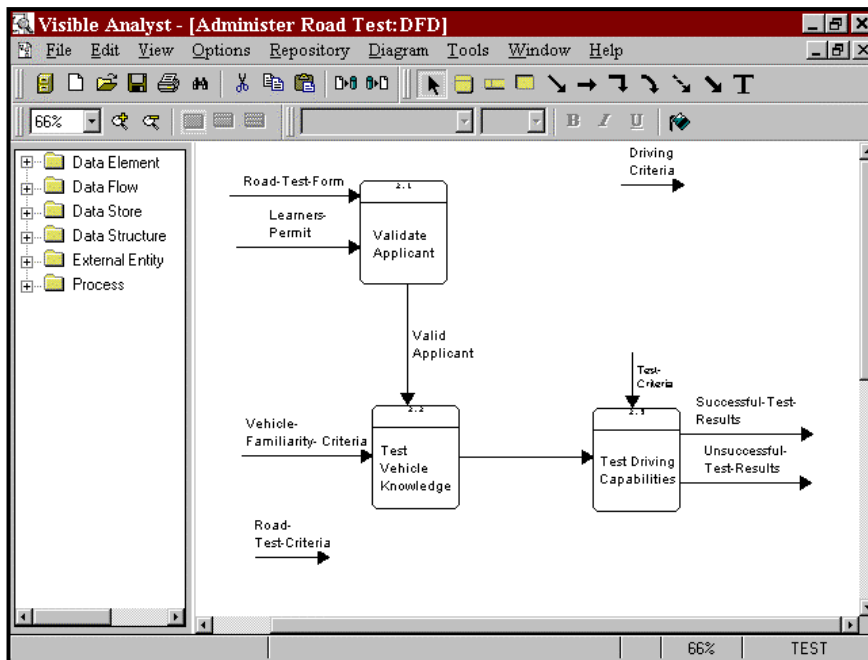
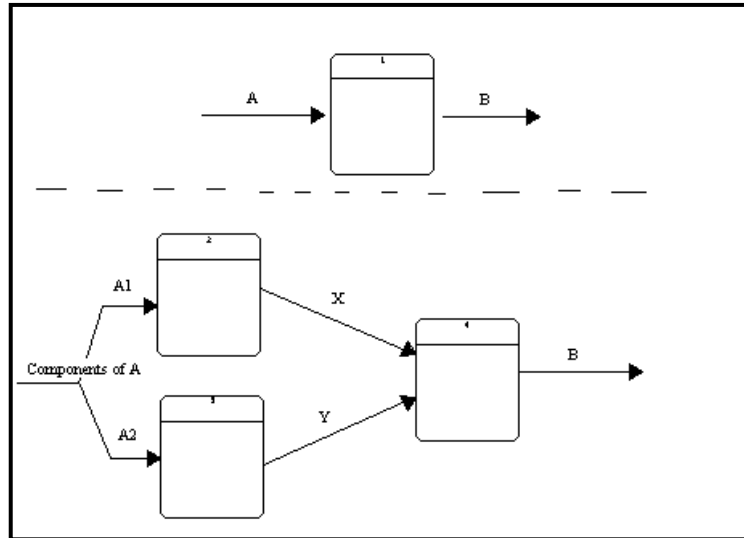



Figure 8-6 Child Diagram with Processes and Flows

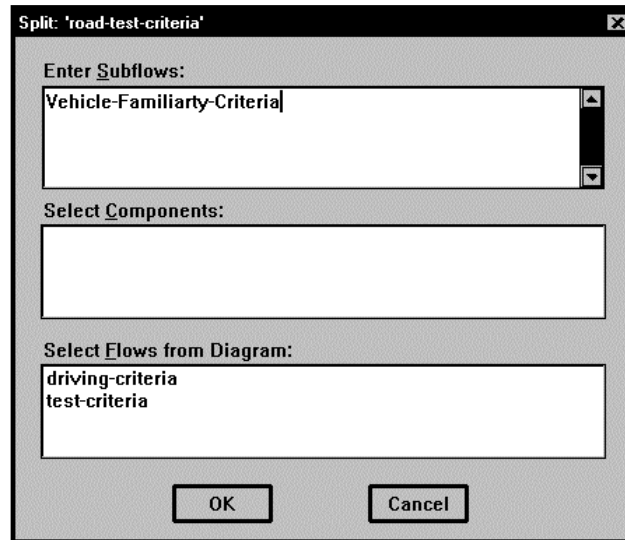
### Splitting Data Flows

Decomposing, or “splitting,” a data flow offers the capability to divide net input or net output data flows into subflows, creating more detailed representation on lower-level diagrams. The concept is illustrated in Figure 8-7. This capability greatly aids in the system analysis process by showing more complex data flows at high levels of the structured specification and smaller or even atomic data flows at the lower levels of the structured specification. This decomposition capability provides a better understanding of the entire system and its parts. Whenever a dragged-down data flow is split into subflows, the original flow is erased from the current diagram and replaced by the selected or created subflows.



**Figure 8-7 Splitting a Data Flow**

- Select a Flow to Split:*
- 1 Click the  button on the control bar to put the cursor into selection mode.
  - 2 Display the **Object** menu for the data flow labeled Road-Test-Criteria by clicking on one end of it with the *right* mouse button.
- Start the Split:*
- 3 Choose Split Data Flow.
  - 4 In the box labeled Enter Subflows, type in “Vehicle-Familiarity-Criteria.” This option draws a new flow (a subflow of Road-Test-Criteria) on the diagram with this label. See Figure 8-8. If you want to add more than one name in this box, press ENTER to place the cursor on a new line.



**Figure 8-8 Split Data Flow Dialog Box**

- 5 In the box marked Select Flows from Diagram, click Driving-Criteria. This option allows you to select an existing flow to be a subflow of Road-Test-Criteria.
- 6 Click OK. Notice that the flow Road-Test-Criteria is no longer on the diagram and that the flow Vehicle-Familiarity-Criteria has been added to the diagram.
- 7 Attach the flow Driving-Criteria to the process Test Driving Capabilities with the arrow pointing *away* from the process symbol. (This error is made deliberately; it is explained during the discussion of Analyze.)
- 8 Ignore the data flow Vehicle-Familiarity-Criteria, as another test for Analyze.

## **ANALYZING FOR BALANCE AND COMPLETENESS**

As a project goes through a number of nested decompositions (nests), data flow splits, various object moves and other edit procedures, there is a significant possibility that various data flows are incorrectly used, or that objects are forgotten, etc. For a large project with many symbols and flows, this is a real probability; and the errors are not easily detected by visually

checking the diagrams yourself. The **Analyze** function, found on the **Diagram** menu, is designed to warn you of completeness and logic errors. The function checks diagrams for:

- Labels on all objects.
- Unattached objects.
- At least one input flow and one output flow for each process.
- Data flow balance, which implies that an input flow is used everywhere as an input flow rather than an output flow and that data flows are properly accounted for at all levels of the diagram hierarchy.

The diagram is now analyzed for adherence to the rules of the Gane & Sarson methodology. Those rules are outlined in the *Visible Analyst Operation Manual* and in the online help.

- Analyze the Diagram:*
- 1 Select **Analyze** from the **Diagram** menu.
  - 2 Select **Current Diagram** and click **OK**.

Visible Analyst displays the errors found. To display the errors full screen, click the **Maximize** button in the upper right corner of the error window. If an error message extends beyond the box, use the scroll bar at the bottom of the box to scroll the text to the left. There should be five messages.

<i>Data Flow labeled 'Vehicle-Familiarity-Criteria' is dangling.</i>	This indicates that Vehicle-Familiarity-Criteria is not attached to a process.
<i>There are 1 unnamed Data Flow(s).</i>	This is the data flow that you left unlabeled on the diagram.
<i>Net input Data Flow 'Test-Criteria' is not shown attached to parent Process.</i>	This indicates that the data flow Test-Criteria has been added to the child diagram but is not accounted for on the parent diagram.
<i>'Driving-Criteria' should be shown as a net input Data Flow.</i>	This indicates that Driving-Criteria is being used as a net output flow on the diagram, while it is used as a net input flow on the parent.
<i>Input Data Flow 'Road-Test-Criteria' on parent is not shown.</i>	This message is a result of the fact that Vehicle-Familiarity-Criteria, a child flow of Road-Test-Criteria is not <i>attached</i> to a process as a net input flow, even though it appears on the diagram.



**Note**

- ☐ Analysis error dialog boxes allow you to keep them on the screen while you carry on various Visible Analyst activities. This is to make it easier for you to correct the errors found by Analyze. If you don't want to keep the box open, press ESC or click Cancel to close it.

**Fixing the Errors**

- Correct the Data Flows:*
- 3 Attach the data flow Vehicle-Familiarity-Criteria to the process Test Vehicle Knowledge, as shown in Figure 8-9.
  - 4 Reverse the direction of Driving-Criteria, so that it becomes an input flow to Test Driving Capabilities (see Figure 8-9), by dragging the endpoints.
  - 5 Delete Test-Criteria by clicking on the line and pressing the DELETE key.
  - 6 Label the unlabeled data flow "Vehicle-Knowledge" by clicking on the line with the *right* mouse button and selecting Change Item from the Object menu. Then enter the label and click OK
- Analyze Again:*
- 7 Select Analyze from the Diagram menu again.
  - 8 Choose Current Diagram and click OK. The diagram should now be correct.

**Note**

- ☐ It is unnecessary to save a diagram after Analyze has been performed because Visible Analyst automatically saves it for you before analysis begins.

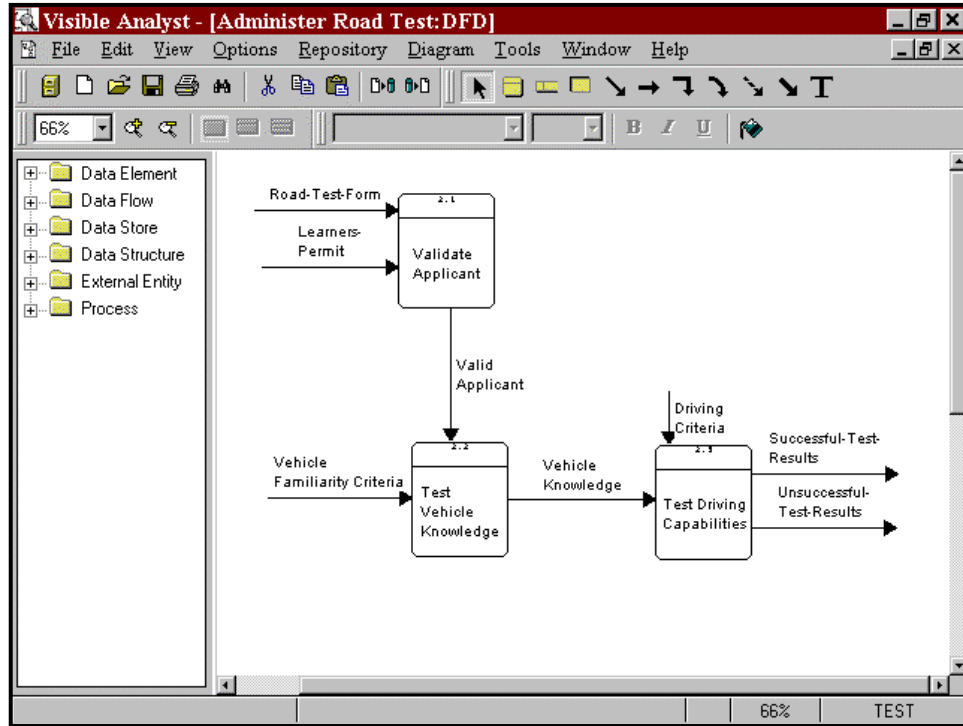


Figure 8-9 The Completed Diagram

## GENERATING A PROCESS DECOMPOSITION MODEL

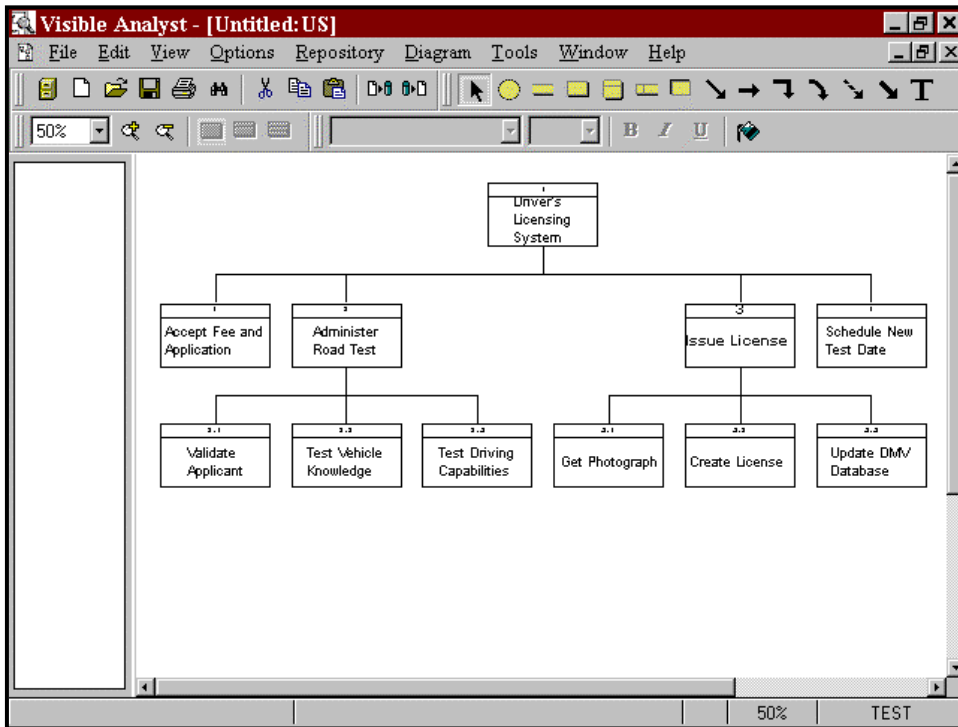
A process decomposition model for a process shows you the hierarchical structure of a decomposed process that has been nested.

### Note

- A *process* decomposition diagram is very different from a *functional* decomposition diagram. The former is simply an unstructured diagram displaying the hierarchy of processes that are descendants of an indicated process. The latter, discussed in Lesson 4 -Functional Decomposition Diagrams, is a full diagramming methodology for performing business planning.

- Open the Diagram:*            1            From the Window menu, click “DMV System: DFD.”
- Select a Process:*                2            Click the process Driver’s Licensing System with the *right* mouse button.

- Create the Decomposition:* 3 Select Decompose. An unstructured diagram is generated showing the hierarchical structure of the process.
- Save the New Diagram:* 4 Select Save from the File menu. Label the diagram "Process Decomposition" and click OK.



**Figure 8-10 Process Decomposition Diagram**