**ITEC 630**

*Information Systems Analysis, Modeling, and Design*

***Lecture Notes***

**Process-Oriented Analysis**

**Data Flow Diagrams and Data Dictionaries**

Created by Daniel LeRevised by Daniel Le 07/28/2016

**Learning objectives**

1. Learn to use data flow diagrams (DFDs) to coherently represent the information gathered as part of requirements determination.
2. Learn how to draw and revise data flow diagrams.
3. Learn about what to do and what not to do when drawing DFDs
4. Learn how to perform balancing and decomposition of DFDs
5. Understand rules to develop a context diagram, a set of balanced data flow diagrams, and a data dictionary
6. Understand how to use DFDs as part of the analysis of an information system and as a tool for supporting business process reengineering

**Overview**

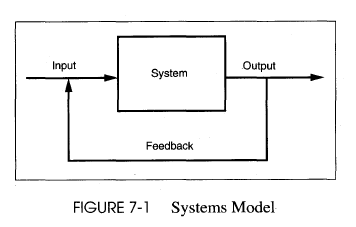
In this week lecture, we review process-oriented analysis using structured analysis following DeMarco [1979], Yourdon [1989], and McMenamin and Palmer [1985]. The fact-finding techniques presented earlier are used to identify human information requirements. We will use that information to develop the system logical model in the system analysis phase and to build the system physical model later in the system design phase. The logical model shows what the system does while the physical model focuses on how the system will be constructed in a specific hardware/software environment. Data modeling concepts and tools used to develop these system models include data flow diagrams and data dictionaries. A data flow diagram describes graphically how data moves through the system while data dictionary consists of information about the system’ data.

***Note #1: All links provided in this lecture can be activated with a "Ctrl + Click"; however, you can also activate these links by copy and paste the link content to the Web browser address bar, just in case.***

***Note #2: To access links associated with "http://library.books24x7.com.ezproxy.umuc.edu", you might have to log into UMUC Library and activate the link "Books24x7" first.***

**STRUCTURED SYSTEMS ANALYSIS CONCEPTUAL FOUNDATIONS**

1. Functions of an information system are the processes that transform application data.
2. Processes and the flows of data into and out of those processes are emphasized
3. Structured systems analysis is based on systems theory, which assumes inputs are fed into  
   processes to produce outputs.
4. Some sort of feedback is included to eliminate system entropy, that is, to keep the system from 'running down.'

****

**[Source: Conger Sue (2008), The New Software Engineering, a Creative Commons Attribution 3.0 License]**

* ***Conceptual Foundations***

[**https://learn.saylor.org/mod/resource/view.php?id=12352**](https://learn.saylor.org/mod/resource/view.php?id=12352)

**Click “The New Software Engineering.pdf” link to view the file.**

**Read the "CONCEPTUAL FOUNDATIONS" section in Chapter 7 (pages 227-228).**

**PROCESS MODELING**

1. Process modeling involves graphically representing the functions or processes, that capture, manipulate, store, and distribute data between a system and its environment and between components within a system.
2. The primary deliverables from process modeling are a set of coherent, interrelated data flow diagrams

* ***Process Modeling***

[**http://library.books24x7.com.ezproxy.umuc.edu/assetviewer.aspx?bookid=30713&chunkid=582801836&rowid=331&noteMenuToggle=0&leftMenuState=1**](http://library.books24x7.com.ezproxy.umuc.edu/assetviewer.aspx?bookid=30713&chunkid=582801836&rowid=331&noteMenuToggle=0&leftMenuState=1)

**STRUCTURED SYSTEMS ANALYSIS CONCEPT TERMS**

***1. Assumptions in using the "structured system analysis"***

* Focusing on what the application is to do.  
  That is, what are its functions or processes?  
  A function or process is some activity that transforms an input data flow into an output data flow.
* Treating the problem in a top-down manner.

In top-down analysis, we analyze the external interfaces of the application first, then high level functions, and finally, lower level functions.

***2. The scope of project activity and a context diagram***

* The scope defines the boundaries of the project
* What is in the project and what is outside of the project.
* The scope of the project is defined in a context diagram
* A context defines a setting or environment.
* In structured systems analysis, the context diagram defines the interactions of the application with the external world. It consists of an application and external entities that are connected together by arrows.

***3. Data Flow Diagram (DFD)***

* A data flow diagram is a graphic representation of the application's component parts.
* Elements of an DFD consist of processes, external entities, data flows, and data stores (data files or databases)
  + A function or process is some activity that transforms an input data flow into an output data flow.
  + An external entity is a person, place, or thing with which the application interacts
  + A data flow is data or information that is in transit. Data flows in a diagram are directed arrows that depict data movement from one place to another.
  + Differences between data flows and files or data stores

1. Data flows are distinct from data stores in their time orientation.

* Data flows are temporary and cease to exist once they are acted upon by a process.
* Files or data stores are persistent and maintained over time.

1. Files or data stores may represent one or more data structures.

* Quality assurance for DFD completeness and consistency can be performed by comparing the context to the data flow.
* There are several analysis levels of DFDs and DFDs start at a high level of abstraction to summarize the processing taking place. At successively more detailed levels, procedural and data are added to describe the processing in more detail.
* DFD balancing is the act of checking entities, data flows, and processes across the levels of the diagram set. After each analysis, the current level of DFDs should be balanced with the previous level.

***4. Data Dictionaries***

* The data dictionary compiles detailed definitions for each element in a DFD.
* The dictionary entries for processes contain details of how to accomplish the process.

***5. Structured Systems Analysis Activities***

The specific activities in structured systems analysis are:

* Develop a context diagram
* Develop a set of balanced data flow diagrams
* Develop a data dictionary
* ***Structured Systems Analysis Concept Terms***

[**https://learn.saylor.org/mod/resource/view.php?id=12352**](https://learn.saylor.org/mod/resource/view.php?id=12352)

**Click “The New Software Engineering.pdf” link to view the file.**

**Read the "SUMMARY OF STRUCTURED SYSTEMS ANALYSIS TERMS" and   
“STRUCTURED SYSTSEMS ANALYSIS ACTIVITIES” sections in Chapter 7 (pages 228-234).**

**DEVELOP CONTEXT DIAGRAM**

The context diagram summarizes the scope of the project.

***Rules for Developing Context Diagram***

1. Define the boundaries (i.e., scope) of the application. Specifically, define what the application will do *and* what it will not do. Draw the circle identifying the application and write the application name in the center.

2. Using the application boundary as a starting point, identify all external entities with which the application must interact. For each entity, draw one square on the diagram and label the square.

3. For each entity, create a definition in the data dictionary.

4. For each external entity, identify the specific data flows that define the interface.

5. For each data flow, create a definition and list of tentative contents in the data dictionary.

* ***Develop Context Diagram***

[**https://learn.saylor.org/mod/resource/view.php?id=12352**](https://learn.saylor.org/mod/resource/view.php?id=12352)

**Click “The New Software Engineering.pdf” link to view the file.**

**Read the "Develop Context Diagram" section in Chapter 7 (pages 234-241).**

**DEVELOP DATA FLOW DIAGRAM**

***Rules for Developing a Data Flow Diagram***

To develop a DFD, iterate through the following steps until a primitive level is reached:

1. Define the processes.

2. Define the files and other data flows required to support the processes.

3. Draw a Level 0 DFD. At level 0, ignore trivial error paths and data stores. If you define a validation process, you must eventually identify an error path. Define the error path at the primitive level. Similarly for data stores, define files when they are shared between processes. Introduce files that are only used within a given process at the level at which the file is shared between two or more subprocesses.

4. Balance the DFD with the context diagram. Compare the net inputs and outputs to external entities on the DFD to the net inputs and outputs on the context diagram. There should be a one-to-one correspondence between the diagrams.

5. Iterate through this procedure until the primitive level of DFD is reached for all processes. Always balance the current level DFD's net inputs and outputs with those of the previous level.

**The DFD syntax rules are:**

1. *All* processes are *connected* to something else.
2. *All* process have *both inputs and outputs.*
3. *No* processes have only outputs or only inputs.
4. Processes may connect to anything: other processes, data stores, or entities.
5. All processes have a unique name and number.
6. Each process number is used once in the diagram set.
7. Only sub-processes of a process shall follow the numbering scheme of the parent process.
8. Entities and data stores may connect *only* to processes. Another way to state this is that each data flow must have at least one end connected to a process.
9. Data flows are the only legal type of connection between entities, processes, and data stores.
10. Make sure there are no dangling arrows.
11. The net data flows to and from context diagram external entities *must* balance, that is, be present, in each level of DFDs.
12. Trivial errors and exceptions are not handled until Ll or lower in the DFD set.
13. Trivial data stores show up in the diagram set the first time they are referenced by a process.

* ***Develop Data Flow Diagram***

[**https://learn.saylor.org/mod/resource/view.php?id=12352**](https://learn.saylor.org/mod/resource/view.php?id=12352)

**Click “The New Software Engineering.pdf” link to view the file.**

**Read the "Develop Data Flow Diagram" section in Chapter 7 (pages 241-261).**

**DEVELOP DATA DICTIONARIES**

The data dictionary (or repository) is used to maintain definitions of all DFDs and other analysis information, including files, fields, flows, and external entities, in addition to processes.

1. Definitions for each element in a DFD are detailed in their data dictionaries.
2. There are five set of data dictionary contents and rules for entities, processes, data stores, data flows, and attributes.

* ***Develop Data Dictionary***

[**https://learn.saylor.org/mod/resource/view.php?id=12352**](https://learn.saylor.org/mod/resource/view.php?id=12352)

**Click “The New Software Engineering.pdf” link to view the file.**

**Read the "Develop Data Dictionary" section in Chapter 7 (pages 261-270).**

**DATA FLOW DIAGRAMMING MECHANICS**

**1. Definitions and Symbols**

There are two different standard sets of data flow diagram symbols: one is developed by Gane and Sarson (1979) and the other is developed by DeMarco (1979) and Yourdon (Yourdon and Constantine, 1979).

* ***Data Flow Diagramming Mechanics (go to section 7.3.1: Definitions and Symbols)***

[**http://library.books24x7.com.ezproxy.umuc.edu/assetviewer.aspx?bookid=30713&chunkid=945265424&rowid=337&noteMenuToggle=0&leftMenuState=1**](http://library.books24x7.com.ezproxy.umuc.edu/assetviewer.aspx?bookid=30713&chunkid=945265424&rowid=337&noteMenuToggle=0&leftMenuState=1)

**2. Developing DFDs: An Example**

An example using Roop Chand restaurant, a fictional restaurant in New Delhi, India, owned by Aman and Vansh Dixit, to illustrate how DFDs are used to model the logic of data flows in information systems.

* ***Data Flow Diagramming Mechanics (go to section 7.3.2: Developing DFDs: An Example)***

[**http://library.books24x7.com.ezproxy.umuc.edu/assetviewer.aspx?bookid=30713&chunkid=945265424&rowid=337&noteMenuToggle=0&leftMenuState=1**](http://library.books24x7.com.ezproxy.umuc.edu/assetviewer.aspx?bookid=30713&chunkid=945265424&rowid=337&noteMenuToggle=0&leftMenuState=1)

**3. Rules of Data Flow Diagramming**

There is a set of rules that we must follow when drawing data flow diagrams and then evaluating DFDs for correctness. In addition, some illustrations are presented to show correct and incorrect ways to draw DFDs as they are applied to these rules.

* ***Data Flow Diagramming Mechanics (go to section 7.3.3: Rules of Data Flow Diagramming)***

[**http://library.books24x7.com.ezproxy.umuc.edu/assetviewer.aspx?bookid=30713&chunkid=945265424&rowid=337&noteMenuToggle=0&leftMenuState=1**](http://library.books24x7.com.ezproxy.umuc.edu/assetviewer.aspx?bookid=30713&chunkid=945265424&rowid=337&noteMenuToggle=0&leftMenuState=1)

**4. Decomposition of DFDs**

Functional decomposition is an iterative process of breaking the description of a system down into finer and finer detail, which creates a set of charts in which one process on a given chart is explained in greater detail on another chart.

* ***Data Flow Diagramming Mechanics (go to section 7.3.4: Decomposition of DFDs)***

[**http://library.books24x7.com.ezproxy.umuc.edu/assetviewer.aspx?bookid=30713&chunkid=945265424&rowid=337&noteMenuToggle=0&leftMenuState=1**](http://library.books24x7.com.ezproxy.umuc.edu/assetviewer.aspx?bookid=30713&chunkid=945265424&rowid=337&noteMenuToggle=0&leftMenuState=1)

**5. Balancing DFDs**

Conservation is the conservation of inputs and outputs to a data flow diagram process when that process is decomposed to a lower level. During the modeling processes, we decompose DFDs from one level to the next and it is required that we must conserve inputs and outputs to a process at the next level of decomposition through an activity called balancing DFDs.

* ***Data Flow Diagramming Mechanics (go to section 7.3.5: Balancing DFDs)***

[**http://library.books24x7.com.ezproxy.umuc.edu/assetviewer.aspx?bookid=30713&chunkid=945265424&rowid=337&noteMenuToggle=0&leftMenuState=1**](http://library.books24x7.com.ezproxy.umuc.edu/assetviewer.aspx?bookid=30713&chunkid=945265424&rowid=337&noteMenuToggle=0&leftMenuState=1)

**USING DATA FLOW DIAGRAMMING IN THE ANALYSIS PROCESS**

**1. Guidelines for Drawing DFDs**

Additional guidelines as given for drawing DFDs that cover completeness, consistency, timing considerations, the iterative nature of drawing DFDs, and primitive DFDs

* DFD completeness is the extent to which all necessary components of a data flow diagram have been included and fully described.
* DFD consistency is the extent to which information contained on one level of a set of nested data flow diagrams is also included on other levels.
* Primitive DFD is the lowest level of decomposition for a data flow diagram.
* ***Using Data Flow Diagramming in the Analysis Process  
  (go to section 7.5.1: Guidelines for Drawing DFDs)***

[**http://library.books24x7.com.ezproxy.umuc.edu/assetviewer.aspx?bookid=30713&chunkid=565686837&rowid=371&noteMenuToggle=0&leftMenuState=1**](http://library.books24x7.com.ezproxy.umuc.edu/assetviewer.aspx?bookid=30713&chunkid=565686837&rowid=371&noteMenuToggle=0&leftMenuState=1)

**2. Using DFDs as Analysis Tools and in Business Process Reengineering**

We have learned that DFDs can be used to model systems but DFDs can also be used as analysis tools to discover discrepancies between sets of DFDs and to model processes in Business Process Reengineering (BPR).

* ***Using Data Flow Diagramming in the Analysis Process  
  (go to sections 7.5.2: Using DFDs as Analysis Tools)  
  (go to section 7.5.3: Using DFDs in Business Process Reengineering)***

[**http://library.books24x7.com.ezproxy.umuc.edu/assetviewer.aspx?bookid=30713&chunkid=565686837&rowid=371&noteMenuToggle=0&leftMenuState=1**](http://library.books24x7.com.ezproxy.umuc.edu/assetviewer.aspx?bookid=30713&chunkid=565686837&rowid=371&noteMenuToggle=0&leftMenuState=1)

**AUTOMATED SUPPORT TOOLS**

1. A large number of CASE tools to support structured analysis are available on the market.
2. All of the tools support DFDs and dictionaries.

* ***Automated Support Tools***

[**https://learn.saylor.org/mod/resource/view.php?id=12352**](https://learn.saylor.org/mod/resource/view.php?id=12352)

**Click “The New Software Engineering.pdf” link to view the file.**

**Read the "AUTOMATED SUPPORT TOOLS" section in Chapter 7 (pages 270-276).**

**References**

1. Conger, Sue (2008). The New Software Engineering. A Creative Commons Attribution 3.0 License.
2. J.B. Dixit and Raj Kumar (2007). Structured System Analysis and Design, Laxmi Publications. <http://library.books24x7.com.ezproxy.umuc.edu/toc.aspx?bookid=30713>

**Disclaimer:** Some articles or sites I select or refer you to may include materials or opinions associated with particular political or other ideological positions. My pointing to these sites in no way suggests that I am encouraging a particular ideology or position on this or any other related topic; these are simply some of the more interesting and informative sites I have found to cover the topic(s) at hand. As always, you must consider the source when reading any materials that may reflect a particular ideological point of view on an issue.