Unit 7: Data Diagrams and Dashing Dreams

**Corey Crooks**

**Purdue University Global**

**IT460 – Systems Analysis and Design**

**Ted Witt**

**June 7th, 2023**

**What is the purpose of creating a logical process model and then a physical process model? How are they different? What information is found on the physical DFD that is not included on the logical DFD?**

There are many important distinctions between logical, and physical models. Logical models primarily focus on showing how data structures may move through a given system, while physical models show what exactly is happening to the data at these points with concerns for what technology the team will need to utilize in order to accomplish these goals (Mallikaarachchi, n.d.). Logical diagrams are essential for helping designers and business analysts understand the idealistic requirements of a design, and will aide in the team’s organizational capabilities while keeping to the project’s scope. Meanwhile, the physical model will elaborate on these topics to help teams understand the functional requirements of such a design—and will help organize tangible elements of the project. These diagrams may include specific functions or names of files and databases needed as well as what data is to be pulled from where, whereas a logical dataflow diagram would simply put a centralized idea in place of databases or files.

**List the steps necessary transform logical models to physical models** (Alan Dennis, 2012)**.**

1. Add Implementation References

At this step, it is important to start with adding references to how exactly the data will be handled. Specific data stores and human interaction should be noted and documented.

2. Draw a Human-Machine Boundary

In this step, separate human interactions with automated procedures. It is required to draw lines in order to document which steps in the data flow diagram require human input in which capacities, and which processes are to be performed by the system in automated functionality.

3. Add System-Related Data Stores, Data Flows, and Processes

This step handles additions and substitutions for logical processes that cannot be accommodated as described previously. Additional steps or data stores may be required in order to perform functions as outlined in project requirements. This step is where these ‘work-around’s may be added.

4. Update the Data Elements in the Data Flows

To fully incorporate the previous step, data elements may now be misaligned, or need updating in the Data Flow Diagram. This step will handle updating such elements to conform with the new layout as put forth in the previous three steps.

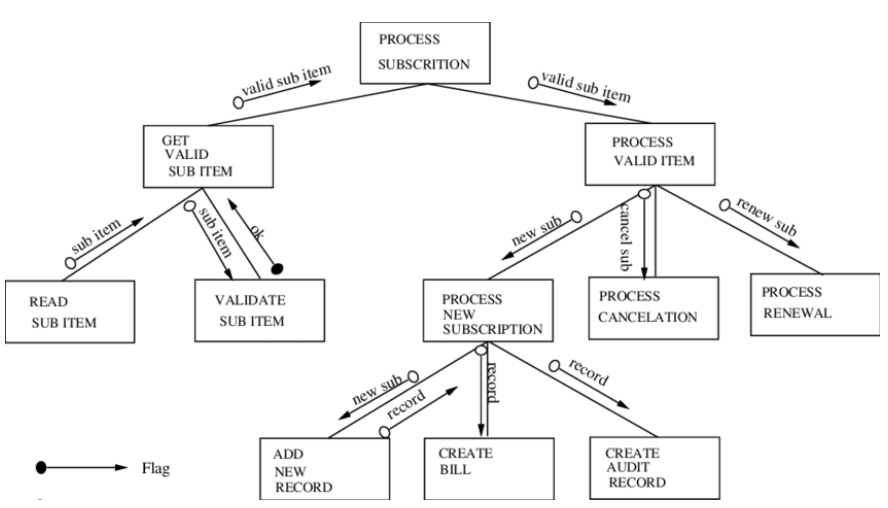
5. Update the Metadata in the Computer-Aided Software Engineering Repository

This step ensures the project is well documented with new information and analytics. Utilizing the CASE repository, the project’s team will update the information of this project to include automated procedures and requirements such as space and time restrictions as well as data growth rates.

**Define a structure chart and explain the importance of sequence, selection, and iteration**

Structure charts are important documents that list exactly that; the structure of a given system as it is intended to be implemented by designers in coherence with project requirements and goals (Alan Dennis, 2012). Sequence of such a chart gives the order in which specific components will perform their functions, Selection give conditions upon which components will operate the aforementioned functions, and Iteration describes at what rate these components operate, and how often these operations are required.

**Provide an example of a structure chart you have found online (citation necessary). Explain the purpose of this chart.**



(Ribeiro, 2000)

The above chart depicts a number of elements and information passages pertaining to a subscription system. This system will take a subscription request, process it, create a bill, and renew the subscription with a few criteria to meet.

**What does a data couple depict on a structure chart? What symbols are used to depict it? What does a control couple depict on a structure chart? What symbols are used to depict it?**

In Structure charts, couples depict the transfer of information between the specific models of the chart, shown by arrows between them (Alan Dennis, 2012). When these arrows have a solid circle at the back end, it is known as a Control Couple. These couples depict the passage of system information that is used for the program’s runtime.

**List the seven types of cohesion. Give an example of good cohesion and an example of bad cohesion.**

“Cohesion refers to how well the lines of code within each structure chart module relate to each other” (Alan Dennis, 2012). Essentially, cohesion details which elements of a program may effect or be affected by another element. The seven types of cohesion are: functional cohesion for modules to only perform a single tasks, sequential cohesion for the output to be used by the next module, communicational cohesion for elements to use similar inputs and outputs, procedural cohesion for elements to be performed sequentially but not share data, temporal cohesion for elements to relate in time, logical cohesion for elements to be scheduled outside the module, and coincidental cohesion for no apparent relationship between elements. Elements that perform minimal tasks may be more effectively in cohesion. Poor cohesion is characteristic by modules that run many complicated tasks and signal other modules to perform their tasks as well.

**What is meant by the characteristics of fan-in and fan-out?**

Fan out is characterized by the number of subordinates a certain module has. A high degree of fan-out is represented by a high degree of sub-modules that a given module controls (Alan Dennis, 2012). Fan-in represents the opposite factor. Fan-in dictates a low number of sub-modules per control module.

**Discuss three ways to improve the quality of a structure chart.**

Although there are many ways to improve structure chart quality that will depend on each individual chart, there are proper rules to stick by. Firstly, ensure the structure chart communicates how the program should be put together in a clear and concise picture. Secondly, ensure the modules of the structure chart are described in enough detail to remain effective throughout the planning and documentation phase of this project. Thirdly, ensure project heuristics are clearly labeled, and flow in an organized manor throughout the chart’s span. Project heuristics should relate to previous orders, and have a clear sequence of events from one to another.

# **References**

Alan Dennis, B. H. (2012). *System Analysis and Design, Fifth Edition*. Retrieved from John Wiley and Sons, Inc.: https://learning-oreilly-com.libauth.purdueglobal.edu/library/view/system-analysis-and/9781118057629/00\_cover.html

Mallikaarachchi, V. (n.d.). *Data Modeling for System Analysis*. Retrieved from University of Missouri - St. Louis: http://www.umsl.edu/~sauterv/analysis/Fall2010Papers/varuni/

Ribeiro, L. (2000). *ResearchGate*. Retrieved from A Survey on Semi-Formal Approaches to Software Development: https://www.researchgate.net/figure/Structure-Chart-for-a-Subscription-System\_fig3\_2356571