WIKIPEDIA

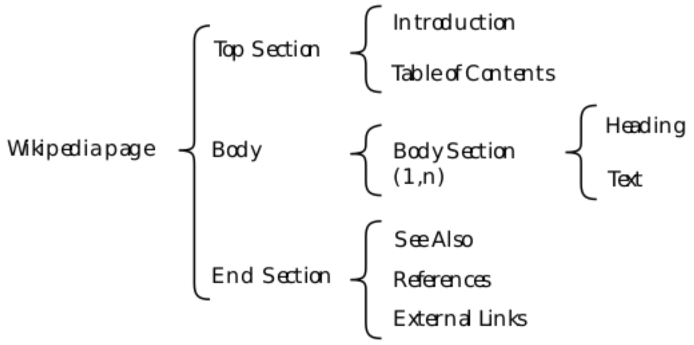
Warnier/Orr diagram

A **Warnier/Orr diagram** (also known as a logical construction of a program/system) is a kind of hierarchical flowchart that allows the description of the organization of data and procedures. They were initially developed in France by Jean-Dominique Warnier and in the United States by Kenneth Orr. This method aids the design of program structures by identifying the output and processing results and then working backwards to determine the steps and combinations of input needed to produce them. The simple graphic method used in Warnier/Orr diagrams makes the levels in the system evident and the movement of the data between them vivid.

Contents

Basic Elements

Warnier/Orr diagrams show the processes and sequences in which they are performed. Each process is defined in a hierarchical manner i.e. it consists of sets of subprocesses, that define it. At each level, the process is shown in bracket that groups its components.



Sample Warnier Orr Data diagram illustrating structure of a Wikipedia page.

Since a process can have many different subprocesses, Warnier/Orr diagram uses a set of brackets to show each level of the system. Critical factors in s/w definition and development are iteration or repetition and alteration. Warnier/Orr diagrams show this very well.

Using Warnier/Orr diagrams

To develop a Warnier/Orr diagram, the analyst works backwards, starting with systems output and using output oriented analysis. On paper, the development moves from the set to the element (from left to right) . First, the intended output or results of the processing are defined. At the next level, shown by inclusion with a bracket, the steps needed to produce the output are defined. Each step in turn is further defined. Additional brackets group the processes required to produce the result on the next level.

Warnier/Orr diagrams offer some distinct advantages to systems experts. They are simple in appearance and easy to understand. Yet they are powerful design tools. They have advantage of showing groupings of processes and the data that must be passed from level to level. In addition, the sequence of working backwards ensures that the system will be result oriented. This method is useful for both data and process definition. It can be used for each independently, or both can be combined on the same diagram.

Constructs in Warnier/Orr diagrams

There are four basic constructs used on Warnier/Orr diagrams: hierarchy, sequence, repetition, and alternation. There are also two slightly more advanced concepts that are occasionally needed: concurrency and recursion.

Hierarchy

Hierarchy is the most fundamental of all of the Warnier/Orr constructs. It is simply a nested group of sets and subsets shown as a set of nested brackets. Each bracket on the diagram (depending on how you represent it, the character is usually more like a brace "{" than a bracket "[", but we call them "brackets") represents one level of hierarchy. The hierarchy or structure that is represented on the diagram can show the organization of data or processing. However, both data and processing are never shown on the same diagram.

Sequence

Sequence is the simplest structure to show on a Warnier/Orr diagram. Within one level of hierarchy, the features listed are shown in the order in which they occur. In other words, the step listed first is the first that will be executed (if the diagram reflects a process), while the step listed last is the last that will be executed. Similarly with data, the data field listed first is the first that is encountered when looking at the data, the data field listed last is the final one encountered.

Repetition

Repetition is the representation of a classic "loop" in programming terms. It occurs whenever the same set of data occurs over and over again (for a data structure) or whenever the same group of actions is to occur over and over again (for a processing structure). Repetition is indicated by placing a set of numbers inside parentheses beneath the repeating set.

Typically there are two numbers listed in the parentheses, representing the fewest and the most number of times the set will repeat. By convention the first letter of the repeating set is the letter chosen to represent the maximum.

While the minimum bound and maximum bound can technically be anything, they are most often either "(1,n)" as in the example, or "(0,n)." When used to depict processing, the "(1,n)" repetition is classically known as a "DoUntil" loop, while the "(0,n)" repetition is called a "DoWhile" loop. On the Warnier/Orr diagram, however, there is no distinction between the two different types of repetition, other than the minimum bound value.

On occasion, the minimum and maximum bound are predefined and not likely to change: for instance the set "Day" occurs within the set "Month" from 28 to 31 times (since the smallest month has 28 days, the largest months, 31). This is not likely to change. And on occasion, the minimum and maximum are fixed at the same number.

In general, though, it is a bad idea to "hard code" a constant other than "0" or "1" in a number of times clause—the design should be flexible enough to allow for changes in the number of times without changes to the design. For instance, if a company has 38 employees at the time a design is done, hard coding a "38" as the "number of employees" within company would certainly not be as flexible as designing "(1,n)".

The number of times clause is always an operator attached to some set (i.e., the name of some bracket), and is never attached to an element (a diagram feature which does not decompose into smaller features). The reason for this will become more apparent as we continue to work with the diagrams. For now, you will have to accept this as a formation rule for a correct diagram.

Alternation

Alternation, or selection, is the traditional "decision" process whereby a determination is made to execute one process or another. The Exclusive OR symbol (the plus sign inside the circle) indicates that the sets immediately above and below it are mutually exclusive (if one is present the other is not). This diagram indicates that an Employee is either Management or Non-Management, one Employee cannot be both. It is also permissible to use a "negation bar" above an alternative in a manner similar to engineering notation. The bar is read by simply using the word "not".

Alternatives do not have to be binary as in the previous examples, but may be many-way alternatives.

Concurrency

Concurrency is one of the two advanced constructs used in the methodology. It is used whenever sequence is unimportant. For instance, years and weeks operate concurrently (or at the same time) within our calendar. The concurrency operator is rarely used in program design (since most languages do not support true concurrent processing anyway), but does come into play when resolving logical and physical data structure clashes.

Recursion

Recursion is the least used of the constructs. It is used to indicate that a set contains an earlier or a less ordered version of itself. In the classic "bill of materials" problem components contain parts and other sub-components. Sub-components also contain sub-sub-components, and so on. The doubled bracket indicates that the set is recursive. Data structures that are truly recursive are rather rare.

See also

- Structure chart

References

External links

- Warnier (http://www.davehigginsconsulting.com/warnier.htm)