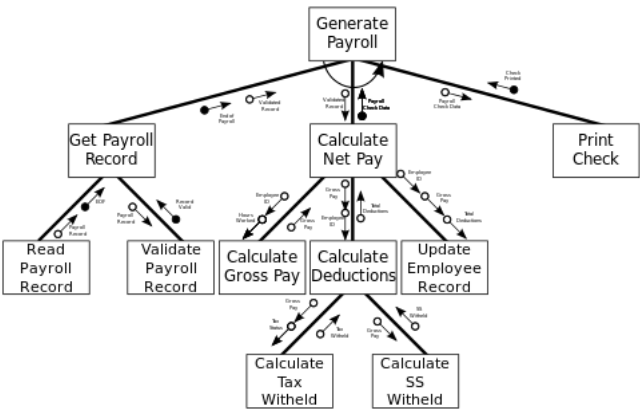


Structure chart

A **Structure Chart** (SC) in software engineering and organizational theory is a chart which shows the breakdown of a system to its lowest manageable levels.^[2] They are used in structured programming to arrange program modules into a tree. Each module is represented by a box, which contains the module's name. The tree structure visualizes the relationships between modules.^[3]



Example of a Structured Chart.^[1]

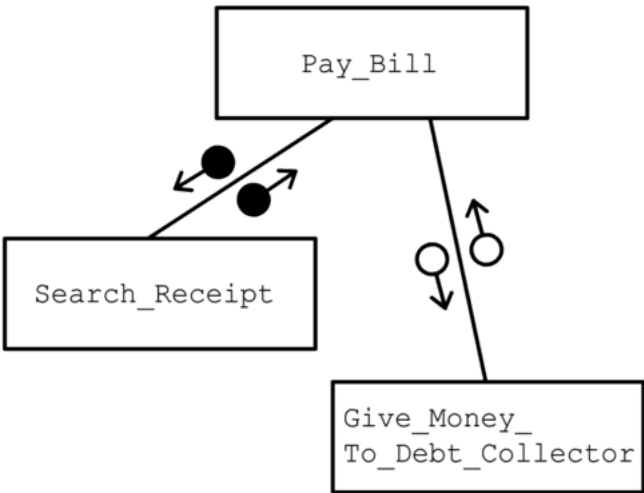
Contents

- Overview
- Structure chart construction
- See also
- References
- Further reading
- External links

Overview

A structure chart is a top-down modular design tool, constructed of squares representing the different modules in the system, and lines that connect them. The lines represent the connection and or ownership between activities and subactivities as they are used in organization charts.^[4]

In structured analysis structure charts, according to Wolber (2009), "are used to specify the high-level design, or architecture, of a computer program. As a design tool, they aid the programmer in dividing and conquering a large software problem, that is, recursively breaking a problem down into parts that are small enough to be understood by a human brain. The process is called top-down design, or functional decomposition. Programmers use a structure chart to build a program in a manner similar to how an architect uses a blueprint to build a house. In the design stage, the chart is drawn and used as a way for the client and the various software designers to communicate. During the actual building of the program (implementation), the chart is continually referred to as "the master-plan".^[5]



This hierarchy chart represents data passing between two modules. When the module Pay_Bill is executed, the pseudocode checks if the bill is already paid by searching for the payment receipt (execute Search_Receipt). If the receipt is not found then it will execute the module Give_Money_To_Debt_Collector to finish the job.

A structure chart depicts^[2]

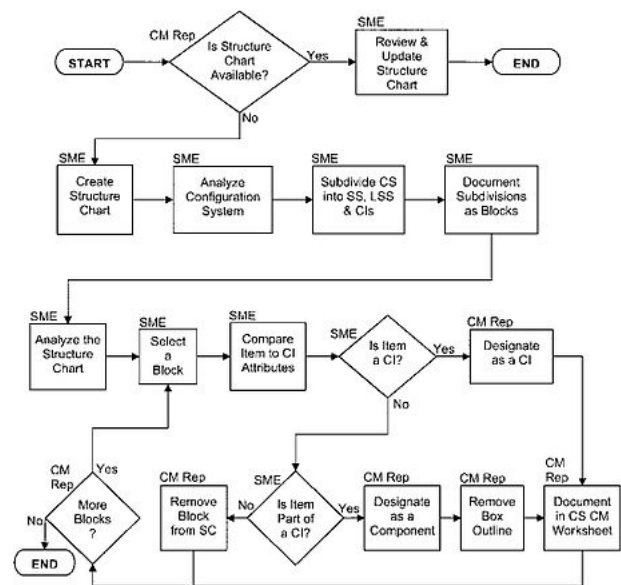
- the size and complexity of the system, and
- number of readily identifiable functions and modules within each function and
- whether each identifiable function is a manageable entity or should be broken down into smaller components.

A structure chart is also used to diagram associated elements that comprise a run stream or thread. It is often developed as a hierarchical diagram, but other representations are allowable. The representation must describe the breakdown of the configuration system into subsystems and the lowest manageable level. An accurate and complete structure chart is the key to the determination of the configuration items (CI), and a visual representation of the configuration system and the internal interfaces among its CIs^(define CI clearly). During the configuration control process, the structure chart is used to identify CIs and their associated artifacts that a proposed change may impact.^[2]

Structure chart construction

According to Wolber (2009), "a structure chart can be developed starting with the creating of a structure, which places the root of an upside-down tree which forms the structure chart. The next step is to conceptualize the main sub-tasks that must be performed by the program to solve the problem. Next, the programmer focuses on each sub-task individually, and conceptualizes how each can be broken down into even smaller tasks. Eventually, the program is broken down to a point where the leaves of the tree represent simple methods that can be coded with just a few program statements".^[5]

In practice, see figure, first it is checked if a Structure Chart has been developed already. If so an expert needs to review it to ensure it represents the current structure and if not, updates the chart where needed..^[2]



A process flow diagram describing the construction of a structure chart by a so-called Subject Matter Experts (SME).^[2]

See also

- Computer-aided software engineering
- Context diagram
- Decomposition diagram
- Function model
- HIPO
- Structured Analysis and Design Technique
- Warnier/Orr diagram
- Work breakdown structure

References

1. Sandia National Laboratories (1992). Sandia Software Guidelines Volume 5 Tools, Techniques, and Methodologies (<http://www.prod.sandia.gov/cgi-bin/techlib/access-control.pl/1985/852348.pdf>) SANDIA REPORTS 85-2348qUC-32
2. IRS (2008) "Configuration Management" In: *IRS Resources Part 2. Information Technology Chapter 27. Configuration Management*. Accessed at [IRS.GOV \(https://www.irs.gov/\)](https://www.irs.gov/) 14 Nov 2008. No longer online Nov 8, 2009.