Hierarchy

Hierarchical organization is the simplest structure for visualizing and understanding complexity.

Increasing the visibility of the hierarchical relationships within a system is one of the most effective ways to increase knowledge about the system. Examples of visible hierarchies are book outlines, multi-level software menus, and classification diagrams. Perception of hierarchical relationships among elements is primarily a function of their relative left-right and top-down positions, but is also influenced by their proximity, size, and the presence of connecting lines. Superordinate elements are commonly referred to as *parent* elements, and subordinate elements as *child* elements. There are three basic ways to visually represent hierarchy: trees, nests, and stairs.[1]

Tree structures illustrate hierarchical relationships by locating child elements below or to the right of parent elements, or through the use of other strategies indicating hierarchy (e.g., size, connecting lines). Tree structures are effective for representing hierarchies of moderate complexity, but can become cumbersome for large or complex hierarchies. Tree structures grow large quickly, and become tangled when multiple parents share common child elements. Tree structures are commonly used to represent overviews or high-level maps of system organization.

Nest structures illustrate hierarchical relationships by visually containing child elements within parent elements, as in a Venn diagram. Nest structures are most effective when representing simple hierarchies. When the relationships between the different levels of the hierarchy become too dense and complex to be clearly distinguishable, nest structures become less effective. Nest structures are most commonly used to group information and functions, and to represent simple logical relationships.

Stair structures illustrate hierarchical relationships by stacking child elements below and to the right of parent elements, as in an outline. Stair structures are effective for representing complex hierarchies, but are not easily browsed, and falsely imply a sequential relationship between the stacked child elements. Interactive stair structures found in software often deal with the former problem by concealing child elements until a parent element is selected. Stair structures are commonly used to represent large system structures that change over time.[2]

Hierarchical representation is the simplest method of increasing knowledge about the structure of a system. Consider tree structures when representing high-level views of hierarchies of moderate complexity. Consider nest structures when representing natural systems, simple hierarchical relationships, and grouped information or functions. Consider stair structures when representing complex hierarchies, especially if the volatility and growth of the system represented is unpredictable. Explore ways to selectively reveal and conceal the complexity of hierarchical structures to maximize their clarity and effectiveness.[3]

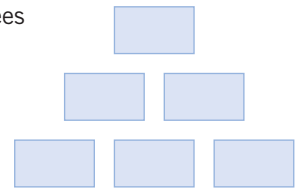
See also Advance Organizer, Alignment, Five Hat Racks, Layering, and Proximity.

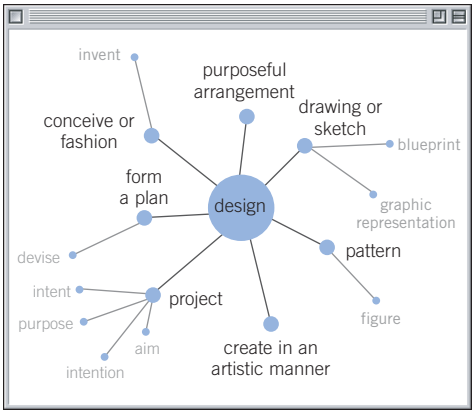
[1] The seminal works on hierarchy are "The Architecture of Complexity," Proceedings of the American Philosophical Society, 1962, vol. 106, p. 467–482; and The Sciences of the Artificial, MIT Press, 1969, both by Herbert A. Simon.

[2] Note that stair hierarchies in software are often referred to as tree hierarchies.

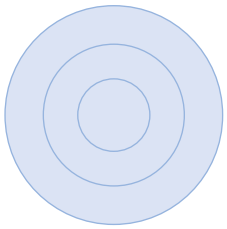
[3] Representing these structures in three-dimensional space improves little in terms of clarity and comprehensibility—though it does result in some fascinating structures to view and navigate. See, for example, "Cone Trees: Animated 3D Visualizations of Hierarchical Information" by George G. Robertson, Jock D. Mackinlay, Stuart K. Card, *Proceedings of CHI '91: Human Factors in Computing Systems*, 1991, p. 189–194.

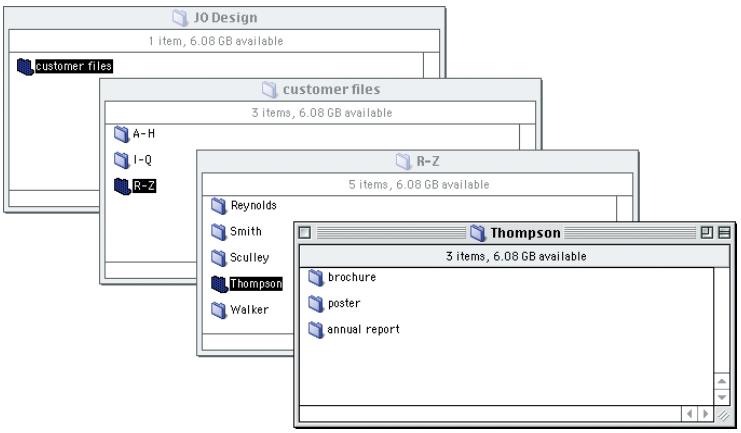
Trees





Nests





Stairs