

Defining a System

Grant Martin glm2367

October 21, 2021

Throughout the various sciences and a multitude of perspectives, the term system is used quite liberally in various ways. However, there are certain traits of these definitions that overlap. Taking this overlap into consideration, a system can be defined as a typically hierarchical group of multiple parts that interact with each other to make a whole. Changes to these parts, therefore, change and evolve the system in its entirety. In this paper, I will cross reference various scientific papers' proposals of the definition of system to synthesize the previously stated definition. Then, I will provide my own proposal for a necessary feature of systems. Finally, taking both the papers and my own thoughts into consideration, I will list the criteria for a group to be considered a system.

Crossreferencing Ackoff's claim of a system's parts' dependence as well as Lampson's implication of dependence, we can conclude that a system must consist of interdependent parts. In Russell L. Ackoff's paper, "Systems thinking and thinking systems," Ackoff states that a system "cannot be divided into independent parts or subgroups of parts." [5] Because they cannot be divided into independent parts, they must, therefore, be dependent upon each other in order to be considered a part of a system. In Butler W. Lampson's paper, "Hints for Computer System Design," Lampson implies this previously mentioned trait of dependency, stating that, since an interface, which is a system, "embodies assumptions which are shared by more than one part of a system, and sometimes by a great many parts, it is very desirable not to change the interface." [4] He claims that, through too much interaction and, therefore, reliance between parts of an interface, difficulties arise when attempting to change the interface, since changing one part of the system may require changes to all the other parts that rely on it.

Based on both Lampson's and Reason's implication of a system's reliance on its parts, we can conclude that a system's parts change and evolve the system as a whole. In Butler W. Lampson's paper, "Hints for Computer System Design," Lampson states that it can be "unclear about how one choice will limit freedom to make other choices, or affect the size and performance of the entire system." [4] Lampson then goes on to describe how some programming languages mitigate these issues, claiming that with Mesa, a programming language that includes type-checking and support interfaces, it "becomes much easier to change interfaces without causing the system to collapse." [4] Similarly, In James Reason's paper, "Human error: models and management", Reason claims that a single "strategic [decision]" made by a member of a system has "the potential for introducing pathogens into the system." [3]

Based on Lampson's example, Parnas' suggestion, and Simon's outright claim of requirement of the hierarchical nature of complex systems, we can conclude that, if complex, which systems most often are, systems are hierarchical with potentially many layers of subsystems. In Lampson's paper, "Hints for Computer System Design", he provides such an example, stating that a computer system "has much more internal structure, and hence many internal interfaces." in comparison to an algorithm [4] This idea of internal structure/interfaces is indicative of the hierarchical property of complex systems. Furthermore, in David Lorge Parnas' paper, "Software Aspects of Defense systems", Parnas states that "The system should be divided

into modules using information-hiding (abstraction) before writing the program begins.”[1] In Herbert A. Simon’s article, “The Architecture of Complexity,” he thoroughly depicts the hierarchical nature of complex systems, stating that a complex system consists of “interrelated subsystems, each of the latter being, in turn, hierarchic in structure until we reach some lowest level of elementary subsystem.” [2]

As a man of science, I tend to follow the definitions that the relevant scientific literature suggests. Thus, I believe that, as synthesized in the previous paragraphs, a system can be defined as a typically hierarchical group of multiple parts that interact with each other to make a whole, and that changes to these parts, therefore, change and evolve the system in its entirety. However, one property of systems that I believe that scientific literature fails to mention in their proposed definitions is that some independence between parts is necessary for something to be considered as a system. For example, computers have a cpu, a gpu, a motherboard, a power supply, a case (typically), ram, and a hard drive, each of which are distinguishable. Thus, computers meet this criteria of a system. However, I would not consider a puddle of water a system. Sure, a puddle of water consists of many water molecules (a group of parts) interacting with each other to make a whole. It also would be affected as a whole if one molecule were to be changed or removed (it’s volume would decrease). However, puddles of water do not have distinguishable parts. All water molecules are molecularly identical with no specialization whatsoever. Therefore, I think it would be more accurate to call it a body of water molecules rather than a system. In contrast, a cup of water would indeed be a system. It contains two distinguishable parts: a cup, and a body of water. Additionally, If the water was not contained within the cup, it would not be a cup of water. Thus, these two parts interact with each other to make the cup of water as a whole.

In conclusion, for something to be considered a system, it must meet the following criteria:

- It must consist of multiple parts
- Its parts must interact with each other
- Its parts must be able to be grouped into one whole
- Its parts must be dependent enough so that changes to an individual part will affect the system in its entirety
- Its parts must be independent enough so that each part is distinguishable

References

- [1] David Lorge Parnas. Software aspects of strategic defense systems. *Communications of the ACM*, 28(12):1326–1335, December 1985.
- [2] Herbert A. Simon. The architecture of complexity. *Proceedings of the American Philosophical Society*, 106(6):467–482, 12 December 1962.
- [3] James Reason. Human error: models and management. *BMJ*, 320:768–770, 2000.
- [4] Butler W. Lampson. Hints for computer system design. *SIGOPS Operating Systems Review*, 17(5):3348, October 1983.
- [5] Russell L. Ackoff. Systems thinking and thinking systems. *System Dynamics Review*, 10(2–3):175–188, Summer–Fall 1994.