The Definition of a System: Computational or Otherwise

Abhinav Varre

April 8, 2022

1 Introduction

The definition of a system has been in contention as the English Language grows and evolves. Domains ranging from biology and mathematics to computation and engineering display fundamentally different understandings and interpretations of what one would consider the "definition" of a system. Therefore, before one goes about attempting to synthesize and develop a definition for a system, it is important to elucidate the context in which the definition is being applied. To begin, this essay will attempt to address the definition of systems from a general perspective and then narrow this definition down so that it aligns with a definition for a "computational system." The literature on this topic is diverse and varies from domain to domain, but this essay will take three separate sources into consideration in order to supplement the argument for the definition.

2 Abstract System

What is a system? In the most general, abstract sense, what constitutes what one may consider a system such that the semantic definition can be applied across a number of domains? Russel Ackoff, in his paper "Systems Thinking and Systems" lists four attributes of a system that form a holistic definition:

1. A "system is a whole containing two or more parts." [3]
2. Each of these parts "can affect the performance or properties of the whole." [3]
3. "No subgroup of [the parts] can have an independent effect on the whole. " [3]
4. A system "cannot be divided into independent parts or subgroups of parts." [3]

The word "part" as Ackoff uses it is functionally defined as an independent component of the overall system for the purposes of this paper. In any case, the fundamental definition of the word "system" in this essay places a heavy emphasis on the idea of a "whole" being comprised of a number of individual parts. However, this definition is incomplete in the sense that it implies that the meaning of the system is derived from the aggregate functions of its parts - or in more colloquial terms, the sum of its parts. A more complete description would have to encapsulate a very important property of a system - a system is formed by the interaction of its individual components, rather than just a sum of these components. In other words, if one were to lay out all the components of a system in a discrete manner, the system ceases to exist. The individual components must interact with one another in order to satisfy the definition. Ackoff describes this when explaining that "a system is whole that cannot be divided into independent parts." However, it is important to note that the interaction between the parts is what constitutes a system, rather than all of the components being part of the same organizational structure.

3 Computational Systems

At this point, the abstract definition of the word system that has been developed is a collection of components or parts that interact with one another. The next section of this essay focuses on narrowing this definition in order to better represent the domain of computation. The handbook “Hints for Computer Systems Design” [2] by Butler W. Lampson expands upon the idea of how a computational system is comprised of “interfaces,” which can be considered systems themselves based on the definition outlined earlier. In the paper, Lampson describes one of the integral attributes of a computer system - the function that the system provides [2]. Lampson provides a number of axioms to adhere to when designing a computer system, but one of the most important ones is to decide on a solid purview of the functionality that a computer system should offer. [2] This will serve as one of the distinguishing factors from the general definition of a system to one aligned towards computational systems. A computational system is designed to, and performs in order to, provide a certain functionality or range of functionality. This definition would not apply to the abstract definition of a system - the domain of biology provides a prime example. An ecosystem is not designed to provide a certain function. It is simply a distinction that is drawn for purposes of classification. Therefore, a computational system is a system that was designed and is used to implement some sort of predetermined functionality.

4 Computational Systems - Components

At this point, the definition of a computer system has been extended to include the role functionality plays. However, there is one more important distinction necessary to supplement and complete the definition of a computational system. A relevant example to illustrate this is the paper “Organizational Structure” [1] by Ahmady et. al. Ahmady describes that one of the qualities of “systemic thinking” is organizational structure. Organizational structure is defined as “a set of methods dividing the task to determining duties and coordinates them.” [1] This highlights a key aspect of computational systems - the explicit designation of every component in the system. This is another specialized attribute of a computational system. Every component is designed to function and interact in a manner consistent with executing the function of the system, and nothing more. What this means is that there are no extraneous components, in the sense that the component exists because it provides an integral function to the overall system. Each component is also carefully chosen to adhere to other constraints, such as efficiency and size. The third attribute that will be added to the working definition is that all of the components of a computational system must be necessary to the functioning of the system, or its optimization. There can be no extraneous components in order for a system to be computational. Nothing in the system is arbitrary - every component plays a role and it matters.

5 Computational Systems - Computation

The definition for a system is now narrow enough to specify computational systems. However, the definition still does not address the computational part of the definition. Therefore, the term computation must be defined in order to encompass both definitions. Computation, in its simplest form, refers to some sort of calculation. The type of calculation can vary, but common examples include arithmetic and logical - both of which are applicable for a computational system. Therefore, the final stipulation to add for the definition of a computational system is that it must undertake some sort of computation, as defined above.

6 Rubric Design

In order for something to be defined as a system, it must satisfy two simple requirements.

- A system must be comprised of components.
- These components must interact with one another.

This rubric, while seemingly trivial, encompasses what this paper defines a system as - A whole comprised of multiple parts that interact with one another. It is important that this definition is broad because it must encompass all systems in all domains. For instance, at first glance it might seem prudent to add the specification that all systems must produce something or function in a certain manner - however, this would exclude certain systems, such as an ecosystem. From here, the rubric can be made more specific by adding three more specifications to approach the working definition of a computational system:

- The system must perform a predetermined function.
- Every component must play a role.
- There cannot be any extraneous solutions to the functions.

These specifications reflect the points in the earlier definition of a computational system. These requirements ensure that only a system that was designed can qualify - however, this rubric is still somewhat general. For instance, a pulley/lever system designed to move an object would qualify, and this would certainly not be computational. Therefore, there must be further requirements in the rubric to ensure that the system is computational. What the rubric requires is to incorporate the concept of computation - whether it be logical, mathematical, or otherwise.

- The system must perform some form of computation.

This rubric, by definition, addresses both parts of a "computational system." Therefore, adding the stipulation that the system must compute something ensures the appropriate amount of exclusivity in the rubric.

References

[1] Gholam Ali Ahmadya, Maryam Mehrpour, and Aghdas Nikooravesh. Organizational structure.

[2] Butler W. Lampson. Hints for computer system design. *SIGOPS Operating Systems Review*, 17(5):33–48, October 1983.

[3] Russell L. Ackoff. Systems thinking and thinking systems. *System Dynamics Review*, 10(2–3):175–188, Summer–Fall 1994.