The Definition of a System: Computational or Otherwise

Abhinav Varre

February 18, 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 computational 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 [3] lists four attributes of a system that form a holistic definition: The first is that a system is a whole containing two or more parts. [3] The second states that each of these aforementioned parts can affect the performance or properties of the whole. [3] The third attribute states no subgroup of [the parts] can have an independent effect on the whole. [3] Ackoff concludes his definition by stating that a system cannot be divided into independent parts or subgroups of parts. [3] Additionally, 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, Lampston 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 Information Management: A Proposal [1]by Tim Berners-Lee. Berners-Lee includes a diagram of a number of components that make up a system that he was writing about - the World Wide Web. [1] This diagram 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.

References

[1] Tim Berners-Lee. *Information Management: A Proposal*, chapter 2. 1989.

[2] Butler W. Lampson. Hints for computer system design. *SIGOPS Operating Systems Review*, 17(5):3348, October 1983.

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