Heterogeneity of Complex Systems

Hessam S. Sarjoughian

Arizona Center for Integrative Modeling & Simulation
School of Computing and Informatics
Arizona State University, Tempe, Arizona, USA
hss@asu.edu

A system can be defined to have different kinds of attributes. Existing concepts focus on scalability and complexity aspects. Scalability is concerned with how well a system design scales with respect to its size. For example, given an algorithm for finding an optimal path through a network, will the computation time remain constant or grow linearly or exponentially relative to the scale of the network. Complexity is a property of the direct and indirect interactions among a system's parts. An example of a system that has complex dynamics is the Internet. A variety of computer hardware, software applications, and instruments are connected together in such a way to allow arbitrary interactions to give rise to complex behavior. Complex systems are defined in terms of five attributes [Booch1994].

(A.I) "Frequently, complexity takes the form of a **hierarchy**, whereby a complex system is composed of interrelated subsystems that have in turn their own subsystems, and so on, until some lowest level of elementary components is reached."

(A.II) "The choice of what components in a system are **primitive** is relatively arbitrary and is largely up to the discretion of the observer of the system".

(A.III) "Intra-component linkages are generally stronger than inter-component linkages. This fact has the effect of separating the **high-frequency** dynamics of the components – involving the internal structure of the components – from **low-frequency** dynamics – involving interaction among components".

(A.IV) "Hierarchic systems are usually composed of only a **few different kinds of subsystems** in various combinations and arrangements."

(A.V) "A complex system that works is invariably found to have evolved from a **simple system** that worked …. A complex system designed from scratch never works and cannot be "patched up" to make it work. One has to start over, beginning with a simple working system."

Aside from the above attributes, it is useful to also consider *heterogeneity* as a distinguishing attribute of complex systems. Heterogeneity is defined to refer to a system's parts having different kinds. Simply stated, as the number of kinds of parts of a system increases, the system complexity will increase. Since some parts of a system have structure and behavior, it is difficult to define heterogeneity from a purely quantitative view. Therefore, it is desirable to understand heterogeneity as much as possible qualitatively – i.e., the higher the degree of heterogeneity, the more complex the software becomes.

The heterogeneity of a software system depends on its target application domain. For example, consider the Spirit Mars rover – a robot geologist [NASA2003]. This rover is part of a complex system in which different instruments are responsible for collecting data from the Red planet. To collect data, the robotic arm, communication transmitters, and vision software of the Spirit rover need to carry out their specialized tasks. For NASA's mission engineers to receive data from Spirit's, specialized communication software is needed. Data repositories are used for storing and using a very large amount of data. Software applications are then used to process different kinds of data and generate information that can be disseminated to the scientific communities and the public at large. The complexity of such a system is much greater as compared with the control system of an aircraft, for example. The control system is responsible for keeping the aircraft cabin pressure within some lower and upper bound limits. In comparison with the Spirit rover, the different kinds of parts of a control system (e.g., sensors and processors, and actuators of a driverless vehicle) are not only individually complex but also, critically, they interact with one another in non-trivial settings. Another example is the Jet Propulsion Laboratory Planetary Data System [Mattmann2006]. This system also is built from different kinds of sub-systems.

In view of the examples mentioned above and examining the five attributes of complex systems, it can be seen that heterogeneity is not accounted in its own right in any of the attributes. Below is a definition for heterogeneity in the context of complexity.

(A.VI) A system's complexity has a direct relation to the heterogeneity of its sub-systems.

Each of the five attributes is briefly examined in terms of *heterogeneity*. The *A.I* attribute does not directly account for the heterogeneity of the parts since a system's decomposition is difficult without considering classifying sub-system types. Similarly, the *A.II* attribute is concerned with identifying primitive components that may or may not involve distinguishing among component types (primitive or not). The emphasis of this attribute is determining the most concrete component in a hierarchy. The *A.III* attribute is also not concerned whether or not the components have different kinds; instead it emphasizes the impact of individual component behavior on the intra- and inter-linkage communications. The *A.IV* attribute is concerned with the evolution of a complex system from simpler systems. This attribute also does not account directly for heterogeneity. With a system having a high degree of heterogeneity, there can be many different simpler systems and thus affecting a complex system's evolution.

The consideration of heterogeneity has a direct impact on a system's complexity as described above and illustrated in the figure below. The complexity is related to the degree of each component's heterogeneity and the number of the heterogeneous components. As the heterogeneity of a system increases, its complexity grows proportionally. The impact of heterogeneity on software complexity is driven by the number of the different kinds of subs-systems and their degrees of interaction. The complexity of a system decreases proportionally to the degree of component homogeneity. For example, given a few different kinds of sub-systems and the degree of heterogeneity is low, the impact of heterogeneity is smaller as compared with the case where the degree of heterogeneity is high (refer to the figure below).

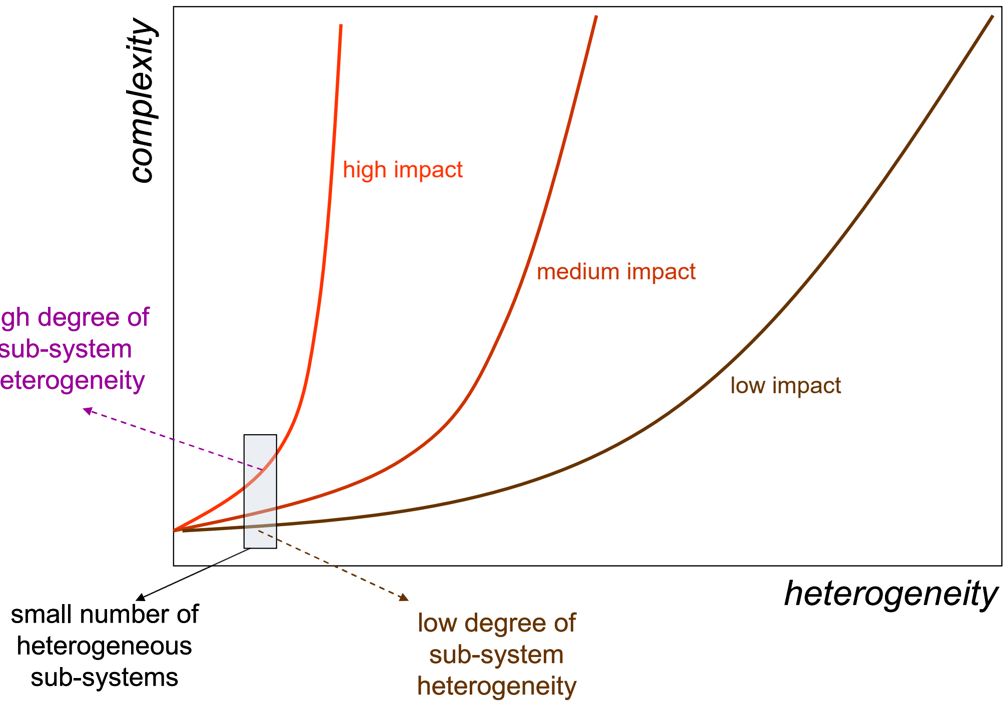

Figure 1: An illustration of syb-system heterogeneity impact on system complexity

References

[Booch1994] Booch, G., (1994), Object-Oriented Analysis and Design with Applications, 2nd Edition, Benjamin Cummings, 1994.

[Gates2005] Gates, B., (2005), Building Software That Is Interoperable By Design.

[Mattmann2006] Mattmann, C.A., Crichton, D.J., Medvidovic, N., and Hughes, S., (2006), "A Software Architecture-Based Framework for Highly Distributed and Data Intensive Scientific Applications", Proceeding of the 28th international conference on Software engineering, p. 721 – 730, Shanghai, China.

[NASA2003], Mars Exploration Rover Mission, https://www.nasa.gov/mission_pages/mer/index.html [visited January 2007] *This reference is updated on August 26, 2021.*