Subrata Dasgupta

Epistemic Complexity and the Sciences of the Artificial

Abstract

In 1962 Herbert Simon articulated the nature of complexity of both natural and artificial systems. A system, he said, is complex if it is composed of a large number of components that interact in nontrivial ways. I will label Simon's notion as *systemic complexity*. However, in the case of *artifacts* – things produced or conceived in response to some need or desire – there is another type of complexity which is especially relevant. This is the *richness of the knowledge embedded in an artifact*. I call this *epistemic complexity*. It comprises of the knowledge that both contributes to the creation of an artifact and the knowledge generated as a result of that creation.

Insofar as artifacts are what the *sciences of the artificial* are about, we might hope that the study of epistemic complexity might deepen our understanding of the sciences of the artificial and the nature of artifact creation.

In this paper I use examples from the history of technological artifacts to analyze aspects of epistemic complexity and its relation to systemic complexity.

1. Two types of complexity

In 1962, Herbert Simon articulated the nature of complexity as it is evident in both natural and artificial systems. A system, he said, is said to be complex if it is composed of a large number of components that interact in nontrivial ways. This means that even if one understands the properties of each component in isolation, one may not be able to interpret the properties of the system as a whole.[1]

I will label Simon's notion as *systemic complexity*. Now, *artifacts* – objects that are produced or conceived in response to some need or desire – are clearly more or less complex in this systemic sense. But there is another type of complexity which is especially relevant in the case of artifacts. And this is *the richness of the knowledge that is embedded in an artifact.* I will call this *epistemic complexity.* It comprises of the knowledge that both contributes to the creation of an artifact; and the knowledge that is generated as a result of that creation. Insofar as *arti-*

1 Herbert A. Simon, "The Architecture of Complexity", in: *Proceedings of the American Philosophical Society*, 106, 1962, pp. 467-482; Simon, *The Sciences of the Artificial.* Cambridge (Mass.): The MIT Press 1996.

H. Andersen et al. (eds.), *New Challenges to Philosophy of Science*, The Philosophy of Science in a European Perspective 4,
DOI 10.1007/978-94-007-5845-2_25,

facts are what the "sciences of the artificial" are about,[2] examination of epistemic complexity contributes, I believe, to our understanding of the nature of artifacts. Insofar as the systematic *study* of the nature of artifacts is what the "sciences of the artificial" are about, I will hope that shedding light on epistemic complexity will contribute to these sciences.

The nature of the complexity of artifacts has been of interest to me for many years reaching back to my study of the structure of design processes in the realm of computer system design and the design of languages to describe such systems.[3] This paper presents, somewhat briefly, some of the results of these studies especially as they relate to the epistemic complexity of artifacts and its relationship to systemic complexity.

Before I continue let me introduce a term of convenience. Henceforth, I will refer to any *practitioner* who creates artifacts as *artificer*. This is a somewhat archaic word but accurate nonetheless. It embraces inventors, designers, engineers, technologists. I will also use the collective term *artifactual creation* to include design, invention and making.

2. Technological knowledge and epistemic complexity

Artifactual creation is a *knowledge rich cognitive process*. The artificer is armed with a rich body of interconnected knowledge and beliefs which he or she brings to bear in any particular cognitive act of creation.[4] Some of this knowledge is shared by people in general, not just artificers, e.g., common rules of inference, or general mental tools for planning and problem solving. More specific artifactual knowledge is itself quite varied. It includes, e.g., mathematics, the basis sciences and engineering theory. But these types of knowledge have entered the artificer's mind relatively recently, mostly since the 18th century and the Industrial Revolution.[5] In the very long history of artifactual creation, reaching back to the origins of humankind itself, the dominant form of knowledge is what, following Michael Polanyi,[6] we may call *operational principles*. This term refers to all rules, proce-

2 Simon, *The Sciences of the Artificial*, *op cit.*

3 Subrata Dasgupta, "Computer Design and Description Languages", in: Marshall C. Yovits (Ed.), *Advances in Computers, vol. 21.* New York: Academic Press 1982, pp. 91-155; Dasgupta, *Design Theory and Computer Science*. Cambridge: Cambridge University Press 1991; Dasgupta, *Technology and Creativity*. New York: Oxford University Press; Janet Elias and Subrata Dasgupta, "A Cognitive Model of the Engineering Design Mind", in John S. Gero and Nathalie Bonnardel (Eds.), *Studying Designers '05.* Sidney: Key Centre for Design Computing and Cognition 2005, pp. 101-116.

4 Dasgupta, *Technology and Creativity*, *op cit.*

5 Albert E. Musson and Eric Robinson, *Science and Technology in the Industrial Revolution*. Manchester: University of Manchester Press 1969.

6 Michael Polanyi, *Personal Knowledge*. Chicago: The University of Chicago Press

dures, concepts and heuristics that facilitate the creation, manipulation and modification of artifacts.

We can now establish the concept of epistemic complexity in more precise terms. The process of conceiving and bringing into practical form an artifact (*any* artifact) involves the deployment, on the part of the artificer, of his or her knowledge base. Knowledge is, thus, an *input* to the process of artifact creation. But knowledge is also the *output* of that same act: a design embodies one or more operational principles. And in the case of true invention, when the artifactual form is *original* in some significant sense, the operational principles it encodes constitute genuinely *new* knowledge. Thus what distinguished invention or what the engineer-historian Walter Vincenti[7] called "radical design" from "normal design" (also a term Vincenti used) is characterized by two epistemic features: (I) The fact that genuinely new knowledge is produced, predominantly in the form of operational principles; and (II) The fact that old knowledge is put to use in unexpected or surprising way. What seems to most characterize invention or radical design in the realm of artifacts is the *amount, variety and newness of the knowledge embedded in the artifact*. It is this embedded knowledge that I call the epistemic complexity of an artifact.

3. Complexity in normal design

One might expect that there is a direct correlation between systemic and epistemic complexities. Specifically, if an artifact has many components that interact with one another in nontrivial ways and produce behavior that is surprising or obscure, one might expect that such an artifact also encodes a rich body of knowledge. But let us keep in mind that an artifact is epistemically complex not simply because of the amount of knowledge it embeds but the *kinds* of knowledge and the *ways* in which old knowledge combines in the production of the artifact and the new knowledge it generates.

Consider, as an example, the situation Vincenti called *normal design*.[8] As he stated it, in normal design

> The engineer … knows at the outset how the device in question works, what are its customary features and that, if properly designed along such lines, it has good likelihood of accomplishing the desired task.[9]

1962.

7 Walter G. Vincenti, *What Engineers Know and How They Know It.* Baltimore, MD: The Johns Hopkins University Press 1992.

8 *Ibid.*

9 *Ibid.*, p. 7.

In normal design, then the overall composition of the artifact is known *a priori*. Brown and Chandrasekaran called this "routine design", and described, in the context of artificial intelligence application, the design of an air cylinder – a piston and rod arrangement which, by moving backward and forward against a spring within a tube creates a to-and-fro movement of some other connected device: air cylinders have a well defined hierarchical form. Starting with this "generic" form, a specific air cylinder may be designed by filling in the details so as to meet specific parametric requirements.[10] In the language of cognitive science, normal design entails the designer summoning up from his personal knowledge system a well-defined *schema* representing the artifact in some stereotypical form, and then *instantiating* this schema to meet specific requirements.[11]

In normal design very little *significant* new knowledge may be produced; old knowledge is used in more or less the same way as in the past. There is little anticipation of surprise. The systemic complexity of the artifact produced by normal design may be considerable but the epistemic complexity will be quite low.

4. The causal connection between systemic and epistemic complexities: An example

A *direct* causal connection between systemic and epistemic complexities can arise in some acts of design and invention. An example is the development of the computer operating system called *Multics* in the 1960s.

In general, operating systems – software that manages computational resources, supports application software and controls the proper functioning of the computer as it goes about its multifarious tasks – is one of the most systemically complex artifacts in the realm of software. Thus when an operating system is conceived and designed to *be significantly original* its systemic complexity directly causes epistemic complexity.

Multics was designed and built at MIT in collaboration with Bell Laboratories and General Electric in the mid-late 1960s as a time-sharing operating system, for the General Electric GE645 mainframe computer.[12] (Later the GE645 and Multics became Honeywell products.) In its mature state Multics consisted of some 1500 modules for a total of approximately one million lines of machine

10 David C. Brown and Balakrishnan Chandrasekaran, "Knowledge and Control for a Mechanical Design Expert System", in: *Computer*, 19, 7, 1986, pp. 92-100.

11 Michael A. Arbib and Mary B. Hesse, *The Construction of Reality*. Cambridge: Cambridge University Press 1986; Roy C. D'Andrade, *The Development of Cognitive Anthropology*. Cambridge: Cambridge University Press 1995; George Mandler, *Cognitive Psychology: An Essay in Cognitive Science*. Hillsdale, NJ: Lawrence Erlbaum Associates 1985.

12 Elliot I. Organick, *The Multics System: An Examination of Its Structure*. Cambridge (Mass.): The MIT Press 1972.

instructions.[13] Its structure was a direct outcome of its overall objective: to create a general computer utility analogous to electric power and telephone utilities which would run continuously and reliably and provide a comprehensive range of services to a population of users interacting with it through remote terminal access. Multics, thus, was conceived as a *technological system*.[14] The particular capabilities that Multics possessed, in response to this overall objective included: (a) time-sharing facilities; (b) an elaborate information storage system that would protect individual user's programs and data from unauthorized access; (c) a sophisticated programming environment for users, including support for several programming languages, inter-user communication facilities (a forerunner of the email); (d) maintenance and monitoring facilities; (e) features to enhance the management of the system's users; and (f) flexibility that would allow the system to absorb new technologies and changes in user expectations.

Clearly, systemic complexity was built into the requirements that Multics would have to satisfy. And though it was not the first time-sharing system to be built – it was anteceded by another system built in MIT called CTSS (Compatible Time-Sharing System, built between 1960 and 1963) and the Cambridge Multiple Access System developed in Cambridge University, England (completed in 1968)[15] – it was the first experiment in creating a comprehensive computer utility. Multics entailed anything but normal design. It had to be *invented* not just designed.

And because it was invented, the systemic complexity inherent in its requirements gave rise to the epistemic complexity of Multics as an artifact.

In fact, its *phylogeny* (that is, its evolutionary lineage) gives us a good sense of this epistemic complexity. It drew upon (a) CTSS; (b) two alternative schemes, invented elsewhere in the early 1960s for implementing "virtual memory", the illusion of unlimited memory capacity;[16] (c) the technique of "multiprogramming" invented almost contemporaneously, whereby several user programs simultaneously share the computer's memory, and the computer's central processor is passed around amongst them so as to keep the processor always busy;[17] and (d) schemes developed in the early-to-mid 1960s for protecting a user's program and data from unauthorized access by other user programs.

13 Fernando J. Corbato, Jerome H. Saltzer and Charles T. Clingen, "Multics – The First Seven Years", in: Peter Freeman (Ed.), *Software System Principles*. Chicago: SRA 1975, pp. 556-577.

14 Thomas P. Hughes, "The Evolution of Large Technological Systems", in: Wiebe E. Bijker, Thomas P. Hughes and Trevor J. Pinch (Eds.), *The Social Construction of Technological Systems*. Cambridge (Mass.): The MIT Press 1987, pp. 51-82.

15 Maurice V. Wilkes, *Time Sharing Computer Systems*. London: Macdonald and Janes/ New York: American Elsevier 1975.

16 Peter J. Denning, "Virtual Memory", in: *Computing Surveys* 2, 3, 1970, pp. 153-190.

17 Jack B. Dennis, "Segmentation and the Design of Multiprogrammed Computer Systems", in: *Journal of the ACM* 12 4, 1965, pp. 589-602.

The point is, the designers of Multics did not just draw upon these earlier inventions; they combined, expanded on, and generalized them and in the process created a significantly original product. Furthermore, the development of the Multics system entailed a major experiment in the use of high-level programming languages to write a very large piece of software.[18] It also entailed the application of a design method in which beginning with an initial crude and incomplete system, one used it and observed its behavior, and based on the observed problems the designers simplified, redesigned and refined the system.[19]

Thus, the Multics project both absorbed much prior knowledge and produced significant new knowledge. The artifact itself *embodied* this new knowledge – in the form of what cognitive scientists would call *procedural knowledge*.[20] The situation was rather similar to that of the Britannia Bridge, a wrought-iron tubular railway bridge that crossed the Menai Straits in Wales, built by Robert Stephenson and his associates in the 1840s: here too, the very design and construction of a bridge faced with certain specific requirements produced valuable knowledge about the behavior and properties of wrought-iron structures.[21] The Multics system affords a marvelous case study of an artifact in which systemic complexity is inherent in the desired functional requirements of the artifact, which in turn engendered a rich phylogeny of old knowledge that entered into the invention/design process and generated new knowledge. It is a case study in how systemic complexity gives rise to epistemic complexity.

5. A case of decreasing systemic complexity but increasing epistemic complexity

As a case study in which an artifact has a decrease in systemic complexity but an attendant increase in epistemic complexity, consider another historical episode from computer science. This example also addresses another question: does the evolution of artifacts inevitably entail the emergence of progressively greater systemic complexity?

In fact, there is a general viewpoint that technological evolution carries with it a growth in systemic complexity; that is, artifacts evolve from the simple to

18 Corbato, “PL/1 as a Tool for System Programming”, in: *Datamation* 5, 1969, pp. 68-76.

19 Corbato and Clingen, “A Managerial View of the Multics System Development”, in: Peter Wegner (Ed.), *Research Directions in Software Technology*. Cambridge (Mass.): The MIT Press 1979, pp. 139-158.

20 Dasgupta, *Technology and Creativity*, *op cit*., p. 37.

21 Nathan Rosenberg and Walter G. Vincenti, *The Britannia Bridge: The Generation and Diffusion of Technological Knowledge.* Cambridge (Mass.): The MIT Press 1978; Dasgupta, “Testing the Hypothesis Law of Design: The Case of the Britannia Bridge”, in: *Research in Engineering Design* 6, 1, 1994, pp. 38-57.

the complex, from the less to the more complex. Here, parallels have been drawn between the natural and the artificial since biological organisms are considered to have evolved in complexity.[22] (There is, however, an alternative view of the relationship between complexity and evolution in the natural world[23]).

My case study pertains to the development of the "reduced instruction set computer" (RISC) between 1980 and 1985.

From a functional perspective, a computer presents a certain "façade" to those who are to be its users. This functional façade is usually called a computer's *architecture*.[24] Very briefly (and in somewhat simplified terms) a computer's architecture describes precisely those features of the computer that must be known for a programmer to write an executable program for that machine; it constitutes the lowest-level view of the computer that a programmer can interact with. Examples of architectural features are the details of the computer's instruction set, the syntax and semantics of the instructions, and the types of data that the computer can process.

In general, a computer's architecture expresses one of the basic characteristics of systemic complexity: its various components are mutually dependent; they interact with one another.[25] More interestingly, by the end of the 1970s, the pattern of evolution of computer architectures evidenced a distinct tendency towards *increased* systemic complexity: if one examined a particular genealogical line of computers made by specific manufacturers, one would find that the sizes of the instruction set, the syntax of the instructions, and range of data types, the various modes of referencing instructions and data in memory, etc., had all increased markedly in any manufacturer-defined "genus" of computers.

In the early 1980s, computer scientists at the IBM Thomas J. Watson Research Center, the University of California, Berkeley, and Stanford University independently initiated a movement to *reverse* this trend toward increasing systemic complexity. There were sound empirical and technological reasons for this movement. And based on these arguments, these designers proposed the idea of the "reduced instruction set computer" or RISC – the idea of designing computers with *simplified* architectures wherein all architectural features were greatly reduced in variety, numbers and mutual interactions. The RISC movement represented the notion that evolution in the artificial sciences can proceed towards *decreased* systemic complexity.

However, while the first RISCs that were designed and built were *systemically* simple (relative to their ancestors or their conventional counterparts), the inven-

22 John T. Bonner, *The Evolution of Complexity by Natural Selection*. Princeton, NJ: Princeton University Press 1988.

23 Daniel W. McShea, "Complexity in Evolution: A Skeptical Assessment", in: *Philosophica* 59, 1, 1997, pp. 79-112.

24 Dasgupta, *Computer Architecture: A Modern Synthesis, Volume 1: Foundations*. New York: John Wiley 1989.

25 Dasgupta, *Computer Architecture*, *op. cit.*, pp. 108-109.

tion of the RISC concept and the translation of that concept into actual computers were far from being *epistemically* simple. Much historical knowledge was brought to bear by the original inventors in arriving at the RISC concept. And in transforming concept into reality, significantly new knowledge was generated in the realms of computer systems design.[26] The first RISCs were, thus, systemically simple (compared to their predecessors) but such simplicity was gained at the "cost" of considerable epistemic complexity.

6. Epistemic complexity as a marker of the artificer's creativity

What I've tried to demonstrate is that systemic and epistemic complexities are not necessarily coupled. However, our examples also suggest that epistemic complexity is related to the *originality* of artifacts and therefore to the artificer's *creativity*.

An artifact may be systemically complex, but if it is not original, it will be epistemically simple. The products of normal design may exemplify this situation. For example a civil engineer who designs an elaborate flyover system connecting several busy freeways is very definitely creating a systemically complex artifact – both structurally and functionally. But if that system is a product of normal design, it will not be original; no unusual prior knowledge enters into its design or construction, and no new knowledge is produced by it. Epistemically it will be simple.

On the other hand an artifact that is original *will* be epistemically complex, whether they are systemically complex or not. The Multics and RISC systems mentioned earlier exemplify this situation. As another example consider the Italian engineer-architect Pier Luigi Nervi who, in 1936, designed and built aircraft hangars for the Italian Air Force.[27] There were "several traditional solutions" to build such structures: designing aircraft hangars could be seen as exercises in normal design. But Nervi eschewed the normal path. Instead he created an "organism" which transmitted the loads to the supports and columns at the sides and thus provided a large, uninterrupted volume of space for the aircrafts. The huge, dome-like vault was composed of a curved, intersecting network of ribs: this was old knowledge invented eight hundred years earlier by the master masons who built the Gothic cathedrals – but adapted to a radically different type of structure. The sublimity of medieval houses of worship was transposed to the most plebian of buildings – with arresting aesthetic effect. Here was a structure that was

26 David A. Patterson, "Reduced Instruction Set Computers", in: *Communications of the ACM* 28, 1, 1985, pp. 8-21; Manoli G. H. Katevenis, *Reduced Instruction Set Computers for VLSI*. Cambridge (Mass.): The MIT Press 1985.

27 Pier L. Nervi, *Aesthetics and Technology in Building*. Cambridge (Mass.): Harvard University Press 1966.

epistemically complex because it deployed old knowledge in a wholly surprising context. Epistemic complexity is, then, a marker of the artificer's creativity.

7. Descriptors of epistemic complexity

Notice I use the word "marker" above, not "measure". Can epistemic complexity be *measured* at all? For that matter, can *systemic* complexity be measured? In fact, in the latter realm there is no single set of universally accepted measures, each different domain of systemic complexity is perhaps adapted to its own metrics. John Tyler Bonner in his discussion of the evolution of (systemic) complexity of organisms drew upon such measures as body size, diversity of cell types within an organism, diversity of organisms within a community.[28] In the realm of artifacts similar quantitative criteria have been proposed. A well known example is the number of transistors on a integrated circuit chip; the systemic complexity of a software system has been described by the number of lines of instructions, the number of modules comprising the system, the average size of modules, and so on.[29] The study of the (systemic) complexity of algorithms is an important branch of computer science, wherein complexity is measured by the (average or maximum) number of operations of a certain type that the algorithm performs to solve certain classes of problems.[30]

The situation for epistemic complexity is more problematic: it appears to be far less amenable to quantification than its systemic counterpart. One might claim to measure epistemic complexity by simply counting the number of significant and distinct items of knowledge such as facts, concepts, hypotheses, etc., that entered into the invention of an artifact. But such a count would serve as the crudest of measures, not least because what constitutes an "item" of knowledge can be ambiguous. A single "fact" may itself be of limited use in the design or invention process: its significance may only be in its relationship with other items of knowledge – in other words, it may well be an entire *schema* (mentioned earlier) or what cognitive scientists and artificial intelligence researchers call a *semantic network* (that is, a linked network of knowledge and beliefs that show the relationships between the components) that is the significant "item" of knowledge.[31]

For example, in an earlier study, in discussing the invention of the first "superalloy" for gas turbine blades I was able to identify some twenty three significant

28 Bonner, *Ibid*.

29 L.A. Belady and M.M. Lehman, "Characteristics of Large Systems", in: Peter Wegner (Ed.), *Research Directions in Software Technology*. Cambridge (Mass.): The MIT Press 1979, pp. 106-138.

30 Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, *The Design and Analysis of Computer Algorithms*. Reading, MA: Addison-Wesley 1974.

31 Robert M. Harnish, *Minds, Brains, Computers*. Malden, MA: Blackwell 2002.

items of knowledge that appeared to have participated in the invention process.[32] Most of these constituted "old" knowledge which the metallurgists drew upon; the remaining were generated in the process of invention. But such a count conveys nothing of the intricacy of the interactions of these knowledge items, nor the *manner in which they participated in the act of invention*; nor, for that matter, why and how they were invoked at all. The only adequate descriptor of epistemic complexity would be a *description of the ontogenetic process of an invention itself* or some plausible *representation* of this process.

For instance, I attempted to characterize the epistemic complexity of the Britannia Bridge, designed by Robert Stephenson and William Fairbairn in the 1840s by describing a network of cognitive and physical actions involving reasoning, hypotheses construction, experimentation, and model building which Stephenson and his associates engaged in the design process.[33] This description consisted of an interacting web of previously established goals, facts about various bridge forms, general heuristic rules pertaining to engineering design, general problem solving strategies, as well as new facts, new goals, and new hypotheses produced in the course of the design process. Epistemic complexity is, ultimately, a qualitative characteristic: it is not, in general, measurable.

8. Conclusion

In this paper, I have argued that artifacts are characterized by two kinds of complexity. Of these, "systemic complexity" is not unique to artifacts: natural systems manifest it also. The other type of complexity which I have called "epistemic complexity" is uniquely characteristic of human-made systems – artifacts. It is not unique to technological products; "non-useful" things manifest it also. Paintings, sculptures, novels, poems and plays, symphonies, fugues and *ragas* are all infused with epistemic complexity, especially in the intricate ways their creators summon up the past and integrate it into their works.

Understanding systemic complexity tells us *what* the nature of an artifact is. Understanding epistemic complexity tells us *how* that artifact assumed the form it did. Most significantly, in my view, the epistemic complexity of an artifact, useful or otherwise, provides a trace of the artificer's creativity. In this sense it is a *richer* attribute of artifacts than systemic complexity, for it contributes to a depth of understanding of the artifact which analysis of systemic complexity cannot.

If we understand the *sciences of the artificial* as those disciplines that seek to understand artifacts, both in their completed states and the process by which they

32 Dasgupta, *Technology and Creativity*, *op. cit*., pp. 69-74, 152-156.

33 Dasgupta, "Testing the Hypothesis Law of Design: The Case of the Britannia Bridge", *op. cit.*

come into existence, it seems to me that a theory of epistemic complexity has an important place in such sciences.

The Institute of Cognitive Science
The University of Louisiana at Lafayette
PO Box 43772
Lafayette, Louisiana 70504-3772
USA
subrata@louisiana.edu