# Computing and Computation

Paul S. Rosenbloom

*Department of Computer Science Viterbi School of Engineering University of Southern California*
*Los Angeles, CA 90089*
*Corresponding author: rosenbloom@usc.edu*

**In this essay we claim that computing is the fourth great scientific domain, on par with the physical, life, and social sciences.**

## 1. INTRODUCTION

Over the past few years I have been engaged in an effort to understand computing as a scientific domain [1–4]. In the process I have gradually become convinced that computing amounts to a great scientific domain, on par with the physical, life, and social sciences. In brief, a great scientific domain concerns the understanding and shaping of the interactions among a coherent, distinctive and extensive body of structures and processes. Exploring the consequences of this way of thinking about scientific domains, in conjunction with the conclusion that computing is the fourth such domain, has led in a variety of directions, many with implications for computing and the other scientific domains. This article explores three implications of particular relevance to computing and computation:

(i) building on the notion that great scientific domains are about structures and processes to define computation in terms of information transformation;

(ii) leveraging the combination of understanding and shaping at the heart of great scientific domains to see computing's inherent intertwining of science and engineering as a strength rather than a weakness, and as a model for the future of the other domains; and

(iii) subsuming mathematics within computing.

The first topic is the least controversial, but also the most directly relevant to this symposium's focus on the nature of computation. The latter two topics are more likely to be controversial because of how they extrapolate from lessons in computing to conjectures about other fields. The hope is to at least initiate useful conversations on these topics if not to provide final answers.

## 2. COMPUTATION AS INFORMATION TRANSFORMATION

If a great scientific domain operates on a coherent, distinctive and extensive body of structures and processes, and computing is to be such a domain, a key question becomes what are its structures and processes. The need to answer this question has led to the adoption of a working definition of computation in terms of information transformation. There is nothing terribly surprising in this definition, as it is at the essence of many previous attempts to define the field, going all of the way back to the earliest days of information processing. It does, however, appear to differ in two ways from the definition proposed by Denning [5] for this symposium, that (for the discrete case) computation consists of controlled transitions among a sequence of representations. The first difference concerns the nature of the structures, and in particular whether it is more appropriate to think of them as information or representation. The second difference concerns the nature of the processes, and whether they are best considered as transformations or as some form of sequence control.

The distinction between information and representation can be subtle given the range of meanings of each term, and the resulting complexity of overlap between them. Both terms combine narrow technical definitions with broader ways in which they are used in practice. The technical definition of information comes from information theory, where it is structure (bit patterns) that resolves uncertainty. For example, a single bit is sufficient to resolve whether an unbiased coin comes up heads or tails. The technical definition of representation instead originated in philosophy, where it is structure that refers to something else: the referent. For example, when I mention a coin, I may be referring to a specific coin held in my right hand.

At this level these two terms are similar yet distinct. Most structures with referents embody information and vice versa. However, it is possible to imagine information without representation. Consider an informational structure created by a learning program with the sole purpose of yielding more accurate output choices given input features. Such a structure will have procedural semantics, with a meaning that can be determined implicitly by the procedures that use it. But it need not have declarative semantics, where the meaning is tied to an explicit referent. An analyst may occasionally be able

to hypothesize appropriate referents for dynamically created structures, but there is no guarantee, and the computation proceeds whether or not there is such an analyst. In the reverse direction, it is hard to imagine representation without information. Information would, at a minimum, appear to be required to enable identifying from among all possible objects the particular one intended as the referent of the representation. The validity of such an asymmetry would suggest that representational structures comprise a proper subset of informational structures.

Although the technical definitions of representation and information did not originate in the context of computation, both concepts are clearly relevant to it. More than this is required though for either of them to play a role in defining categorical bounds around computation (as opposed to merely specifying a prototype, or central tendency, for computation): computing ought not be able to exist in its absence. For representation, our example above should be sufficient to disqualify it. The learning program is clearly engaged in computation while employing referent-free structures in critical roles. It is possible to cope with this counterexample by enlarging the notion of representation to include all structures with semantics, whether procedural or declarative. However, this would deny the necessity of referents in representation-and thus in computing-and would appear to change the meaning of the term to something essentially indistinguishable from information.

Is information essential for computing? Suppose we were to chop wood instead of logic. This would involve a transformation, but of wood rather than information, and the result would not seem anything like computation. However, if decisions-either by people or machines-are based on either the number of wood chunks or the sizes of the chunks then the wood would embody information, and the process of chopping it would amount to an information transformation. While such an argument is far from water tight, it at least provides an intuition that information is necessary, and thus justifies for the present its use as part of a working definition of computation.

One of the arguments in favor of defining computation in terms of representation is that representation plays such a central role in the human use of computers, whether humans are programming them, understanding them, proving them correct, or interacting with them [6]. For this reason, prototypical computations are indeed representational, with referents and all. My sense though is that this is driven more by the necessity of coherent communication between the computer and the person than by anything inherent to computation per se. If a human and a computer are to have common ground for interaction it helps if they both use structures that mean the same thing. Declarative semantics is ideal for this. Computation without this human-interaction constraint is a different matter though. While it may still involve representation, perhaps in support of common ground across multiple computations or across multiple aspects of a single computation, this does not appear to be essential for computation itself to exist.

Another argument in favor of representation, at least in contrast with information, is that representation is a clearer and less ambiguous term. The term information certainly has a wide range of meanings. The technical definition we have been working with so far provides one example. However, information also has a broader everyday meaning that covers essentially anything that conveys content. This latter usage becomes difficult to distinguish from representation. The definitional space of representation is narrower as long as you stay away from procedural semantics, but with it representation unfortunately becomes just as vague. What we are left with is a pair of terms that are essentially equivalent in their most generic senses, but where the technical definition of information seems to be a more accurate specification of what is minimally necessary for computation while the technical definition of representation may be a better characterization of most human experiences with computation. I have opted for information in my working definition because a minimal-necessity criterion, if valid, seems like a more fundamental criterion; and because misuse of a prototypical definition as categorical could lead to the exclusion of work from the field that really does belong in it.

With respect to processes, the difference is the use of transformation rather than sequence control. Much as information and representation are two variations on a single structural idea, transformation and sequence control are two variations on a single process idea, and moreover one that goes back to the earliest days of information processing. Information processing requires the selection and application of operations that transform information. The term transformation emphasizes the latter aspect, but also implicitly includes the former. The phrase sequence control emphasizes the former, but I would assume also implicitly includes the latter. Thus, this appears to be more an issue of emphasis in terminology than a substantive disagreement. An ideal term or phrase might conceivably include both aspects; for example, something like controlled transformation might do. However, this phrase raises additional questions, such as whether a random transformation of information would be computation. While randomness may provide a degenerate case, sciences should not necessarily define even degenerate cases as outside of their scope, so I lean towards retaining the simpler term, transformation.

## 3. INTERTWINING UNDERSTANDING AND SHAPING

The focus of a great scientific domain is its subject matter, as defined by its structures and processes. For the physical sciences, this means such things as matter, energy and force; for the life sciences, living organisms and their associated processes, such as metabolism, development, reproduction, and evolution; for the social sciences, people and their non-biological processes, such as thought and communication; and for the computing sciences, information and its transformation.

The people who devote their lives to working with these domains are part of the social domain, and thus not part of their domain of study itself-unless of course they are either social scientists or life scientists studying human bodies-however, they do interact with their chosen domain, yielding a flow of influence from the domain to the person, from the person to the domain, or bidirectionally. Understanding amounts to a flow of influence from a domain to a person. The notion captures the essence of what science is about-learning about the world from the world-while glossing over any a priori distinctions in science about which domains may be considered sciences, which methods may be considered scientific, or which people are doing the understanding. Shaping involves the reverse flow of influence, from a person to a domain. It captures the essence of engineering-using what has been learned about the world to alter it in useful ways-but bears the same relationship to it as does understanding to science.

Defining a great scientific domain in terms of a combination of understanding and shaping is far from the norm. Science after all is normally understood to just focus on a methodologically restricted sense of understanding. But the centrality of this combination emerged directly out of my experience as a computer scientist working across the breadth of the field. As a science of the artificial, computing largely seeks to understand phenomena that it itself creates [7]. While some phenomena studied by computing are naturally occurring, for the most part computing studies the human made. The relative dearth of naturally occurring phenomena in computing, along with the resulting difficulty in distinguishing where shaping leaves off and understanding begins, is often viewed as an embarrassment, leaving it unclear to some whether computing is a science under the standard view.

To more clearly articulate the breadth and depth of computing's science base, academics continue to work hard at separating out understanding from shaping. But what if the more fundamental problem instead turned out to be that we have been looking at this issue backwards all of this time? In other words, what if the inherent intertwining of understanding and shaping within computing were actually a strength rather than a weakness? Furthermore, what if this meant that computing is not a problem child within the sciences, but a model for the future of the other sciences? Such a case can in fact be made based on a combination of 1) the increasing brittleness of the traditional distinction between the natural and the artificial and 2) the pragmatic utility of intertwining understanding and shaping.

Is there a fundamental distinction between natural and artificial? For two reasons, the answer increasingly looks to be no. First, the distinction seems to originate in a tradition that god created both nature and people, but with people occupying a special position outside of, and in a dominating position over, nature. Within this tradition, everything god created is natural, along with anything else engendered by processes in nature, whereas anything created by humans is somehow outside

of nature, and thus artificial. If, however, people are merely one more fragment of nature, then their products are as natural as anything else. Second, although it has historically been easy to distinguish human products from natural products, this has become-and will likely continue to become-more and more difficult as our understanding of nature continues to improve and we are increasingly able to shape it at its most fundamental levels. Consider food flavorings. Both natural and artificial flavors may consist of identical molecules, with only their sources differing. Similarly, plants first evolved without human intervention, and then under general pressure from human selection, and now via pointed genetic modifications. Are these plants really becoming more artificial? They are still made out of the same chemical and biological ingredients as the original "natural" plants. When doctors influence stem cells to become organ cells, are the new cells natural or artificial? The body can't tell the difference. And nanotechnology now gives us the ability to shape both the living and physical worlds at the molecular level. The future seems likely to look more and more like this, where we will need to understand and shape an environment in which human and non-human effects are increasingly difficult to distinguish. Thus, even in these traditional domains, the distinction between natural and artificial appears to be heading towards the intellectual scrap heap, ineluctably leading to the same form of inherent intertwining between understanding and shaping across the traditional sciences that we have seen in computing since its inception.

In computing, this intertwining of understanding and shaping has actually been one of its greatest strengths rather than a weakness for which we should feel apologetic. It is a key factor in computing's astonishingly rapid development. The life and social sciences in particular have long suffered from their limited ability to shape their domains in conjunction with their understanding of them. As our ability to create and manipulate living and thinking systems continues to improve, the life and social sciences will have an increasing opportunity, and in fact an imperative, to embrace the intertwining of understanding and shaping that has so long been a major feature of computing. While people have long shaped the physical domain, even there our ability to manipulate it at its most fundamental levels is making a giant leap with the advent of nanotechnology.

We may have to wait until scientists from these other domains fully appreciate both the inevitability and the power of intertwining understanding and shaping in their own work and domains before we can hope to see a broader acceptance of what computing has been both confronting and leveraging since its inception. But this may not be too far in the future, as intertwining increasingly becomes the norm across the sciences. In the meantime, it may make sense to start moving away from a top-level division of human intellectual activities based on science versus engineering, and towards one founded on the four great scientific domains-the physical, life, social and computing sciences. Individual efforts, and perhaps even particular subdisciplines, may be distinguished by how

much they focus on understanding versus shaping, but that is second order. Overall, the fully intertwined combination of understanding and shaping is the heart of science, and facilitates its rapid progress. With such a perspective, most of traditional engineering would naturally be merged with the physical sciences, medicine with the life sciences, and the professions of law, education and business with the social sciences.

Computing, as an intertwined great scientific domain, includes, not only computer science, but also computational science, computer engineering, software engineering, information technology, computational science, information science, information systems, information theory, and informatics. Given the centrality of information within this domain, one question that can be asked is whether the domain would more appropriately be called the information sciences. My answer, however, would be no, because information is merely the structural component of computing. By itself, information-as with all structure-is passive. Any domain focused too much on passive structures and too little on their interactions with active processes lacks the dynamic richness at the heart of all great scientific domains. There can, for example, be no significant role for experimentation in passive domains, leaving analytical methods as the only recourse. Information by itself, without the transformations central to making computing active, would be such a passive domain. The name computing sciences, emphasizing as it does the active nature of the domain, thus seems more appropriate than information sciences. Still, either way, the label is only of secondary importance at best. What really matters is the domain itself, its equality of status with the three preexisting great scientific domains, and its potential as a role model for these other domains as they increasingly intertwine understanding and shaping.

## 4. SUBSUMING MATHEMATICS

Although a passive domain can be of undoubted intellectual and pragmatic importance, it does not possess the additional richness yielded by active processes, and thus, according to the definition here, cannot on its own amount to a great scientific domain. Such a domain can, however, form an important component of a more comprehensive domain that does fully embrace the interactions among structures and processes. The humanities, for example, with its concentration on human-created structures-such as books, paintings, and statues-that yield insight into the human condition, appears to be a passive domain that cannot on its own therefore meet the criteria for being a great scientific domain. But, given its dedication to studying humanity, it could fit naturally as a key constituent of the social sciences, even if its passivity means that its analytical methods will differ from, and likely be weaker than, the experimental methods more common in the rest of the social sciences.

What about mathematics? It clearly possesses the rigor of the most stringent sciences and plays a central role as a tool in all of the sciences, yet it never fit as a discipline within the physical, life or social sciences. Nor has it seemed to many quite like a scientific domain all on its own. There has in fact been a long-standing ambivalence over whether mathematics should be considered a science. One possible explanation for this awkward status is that mathematics is largely a passive science of the artificial. It is artificial because its structures-expressions, equations, theorems, proofs, etc.-are human made. Some have argued that mathematical expressions are reflections of abstract but unobservable truths of the universe, so that what mathematicians study is no more human made than is the subject matter of the physical, life and social sciences. Such an argument, based essentially on the reality (but unobservability) of Platonic ideals, can in fact be made for anything traditionally considered artificial, and conceivably could thus be marshaled as another general argument against the distinction between natural and artificial. However, the important point-that there is doubt about mathematics as a science because its structures are not generally observable in the world without human shaping, and that this is thus akin to the concerns some people have about computing as a science-is independent of whether the subject matter is considered artificial versus natural-but-unobservable.

The structures at the heart of mathematics are informational, just as are those in computing, and in fact representational. However, while there is process in mathematics that is concerned with the transformation of this information, its study has not been of central concern within the field. Mathematics can be used to represent processes in other domains, such as models of the dynamics of physical or social systems, but the represented processes and associated experiments on them are parts of these other domains rather than parts of mathematics. The inherently mathematical processes, such as calculation and proof, are computational, as noticed early on by Turing. But these processes could traditionally only be performed by people, making experimentation with them difficult. Whether or not for this reason, mathematics remained mainly analytical rather than experimental, and focused little of its attention on the nature of these processes.

With both mathematics and computing focused on information and its transformation, there is little to distinguish between them except that mathematics has principally limited itself to a region of this overall scientific domain that is concerned with the analysis of (passive) informational (or representational) structures. With the advent of computers, the study of information transformation became more feasible, including the extensive use of experimental methods. Computers are also more adept at processing non-representational information than are people. Computing has thus been able to expand to cover the full range of interactions between informational structures and processes.

This suggests that computing and mathematics should ultimately be merged into a single domain. In many universities, computing actually grew out of mathematics, but then had

to separate itself from its erstwhile host in order to work more freely outside of the narrowing constraints of mainstream mathematics. In principle, if mathematics had been more open to the full extent of information transformation-including the complete range of understanding and shaping activities implicated-along with the methods appropriate for its study, computing could have remained a part of mathematics. In such a case this domain might have been called the mathematical sciences. My undergraduate major actually went by this name, having predated the existence of an undergraduate computer science major at Stanford. In toto, the major was composed of mathematics, computer science, operations research and statistics. However, mathematics in general has remained focused on its more limited niche of structure analysis while computing took on the interactions among structures and processes. For this de facto reason, and for the related but more principled reason that great scientific domains are, at their essence, about the dynamics of interaction among structures and processes rather than just about structures, computing is a more appropriate label for this domain.

Subsuming mathematics within computing in this fashion should enable a rationalization of the study of information transformation, while also finally laying to rest the long-term ambivalence concerning whether mathematics is a science. According to the arguments here, it isn't a great scientific domain on its own, but it is a key analytical component of a domain that is: computing. Its passivity is eliminated as an issue by its becoming a theoretical facet of a fully active domain, while its artificiality is handled by the earlier arguments about artificiality in computing. Potential worries about computing being again constrained by the more limited methodology in mathematics could be offset if all of the computing disciplines mentioned near the end of the previous section are also welcomed as full members of the computing sciences.

## 5. CONCLUSION

Reconceptualizing computing as a great scientific domain has many potential implications, particularly as great scientific domains are defined here. Three of these implications have been briefly explored in this article, concerning the definition of computation, the intertwining of understanding and shaping, and the relationship of computing to mathematics. Other potential implications of interest arise from combining this notion with a relational architecture for the sciences that is being developed and investigated in the context of computing [1–3]. This pairing clarifies how, beyond just providing tools for use by the other scientific domains, computing acts as a full and equal partner with them in several symmetric relationships. It also aids in understanding both the disciplines and the disciplinary structure within computing, while providing insight into how it might be possible to rethink the focus and boundaries of academic computing.

## ABOUT THE AUTHOR

Paul S. Rosenbloom is a Professor of Computer Science at the University of Southern California and a Project Leader at USC's Institute for Creative Technologies. He received a BS degree in Mathematical Sciences (with distinction) from Stanford University and MS and PhD degrees in Computer Science from Carnegie Mellon University. Before coming to USC he was a Research Computer Scientist at Carnegie Mellon University and an Assistant Professor of Computer Science and Psychology at Stanford University. At USC he spent twenty years at the Information Sciences Institute, including ten years leading new directions activities and a stint as Deputy Director. Prof. Rosenbloom's research has historically focused on the creation of cognitive architectures that model the human mind while supporting the construction of artificially intelligent agents, but impelled by his experiences with new directions he has more recently also begun exploring the nature and place of computing as a scientific discipline.

## ACKNOWLEDGEMENTS

## REFERENCES

[1] Rosenbloom, P. S. (2004) A new framework for computer science and engineering. *IEEE Computer*, **37**, 23 – 28.

[2] Rosenbloom, P. S. (2009) The great scientific domains and society: A metascience perspective from the domain of computing. *The International Journal of Science in Society*, **1**, 133–144.

[3] Rosenbloom, P. S. (Forthcoming) *On Computing: The Fourth Great Scientific Domain*. MIT Press, Cambridge, MA.

[4] Denning, P. J. and Rosenbloom, P. S. (2009) Computing: the fourth great domain of science. *Commun. ACM*, **52**, 27–29.

[5] Denning, P. J. Ubiquity symposium 'what is computation?': Opening statement. *Ubiquity*, **2010**.

[6] Denning, P. J. Personal communication.

[7] Simon, H. (1969) *The Sciences of the Artificial*. MIT Press, Cambridge, MA.