Research over the last few decades has suggested the possibility that computer-based systems may transcend their roles as media and tools for design, and actually adopt the role of designer. This prompted the topic for this debate: “Will computers be able to design as well as human designers in the foreseeable future?”Arguing to the affirmative is Ganapathy Mahalingam of North Dakota State University. I offer the argument to the negative. These arguments are presented below, so that you may weigh them and form your own opinions.

Yes:
Computer-based Systems May Already Be Able to Design As Well As Human Designers
Ganapathy Mahalingam

The question “Will computers be able to design as well as human designers in the foreseeable future?” already acknowledges that computers are able to design in the present, but the issue is whether they can design as well as human designers.

Design encompasses diverse activities that are integrated in complex ways. Debates such as these often depend on particular definitions of design for their resolution.

To resolve the issue of whether computers, or more appropriately, computer-based systems, can design, an Architectural Turing Test should be used. Take this scenario in an architect’s office. The principal walks into the studio and gives a staff architect the task of generating the initial spatial form of a proscenium-type auditorium. The architect is given general requirements such as the seating capacity, area per seat, the performance type, and acoustical parameters such as reverberation time. The principal walks away. The architect sets to work. After 12 hours, she has drawn a perspective drawing of the initial spatial form of the auditorium, after resolving issues such as volume, seating areas, sight lines, seating slopes and sound reflection panels. Has she designed the auditorium? If she has, then a computer-based system that does what she has done is also designing. This is the basis of an Architectural Turing Test. Given the same input, if a computer-based system produces the same output as a human designer, and the human is considered to be designing, then the computer-based system can be said to be designing as well.

There currently exists a computer-based design system (Mahalingam, 1996, 1997, 1998) that can perform better than an architect does in the task of designing the initial spatial form of proscenium-type auditoriums. The design system...
takes programmatic, functional and acoustical parameters, and generates the initial spatial form of the auditorium. In the design system non-spatial information is converted to spatial form by an algorithm. This can be said to be the essence of computer-based architectural design. The initial spatial form of the proscenium-type auditorium is generated instantly from the inputs making it a superior design system. This system proves that certain non-trivial design tasks can be automated, therefore, other non-trivial design tasks can also be automated if they are computationally articulated.

Alonzo Church and Alan Turing independently postulated formal descriptions of computation that led to what is known as the Church-Turing thesis. The thesis basically suggests that all problems that can be solved by a sentient being (note that this can extend beyond humans) are reducible to a set of algorithms. Any problem not solvable by a Turing machine is also not solvable by human thought. This is the premise and goal of the “algorithmic auditorium” project (Mahalingam, 1998) to develop computable processes that design different types of auditoriums. If the process of designing a proscenium-type auditorium is a result of cognitive activity, it is also computable. The major task in achieving this is mapping the cognitive activity onto computable processes.

If you believe the Church-Turing thesis, then architectural design, which is a cognitive activity, can be done by computer-based systems. With their inherent speed of operations, erasable memory (curiously a key factor), access to large volumes of data, and data processing capabilities using electronic logic (which has its own characteristics), computer-based systems will perform better than human designers will in the foreseeable future, if not already in many tasks.

Note: The Church-Turing thesis emerged around 1936, when Alonzo Church postulated lambda calculus and Alan Turing postulated the universal Turing machine. A thought-provoking discussion of many versions of the Church-Turing thesis can be found in Godel, Escher, Bach: An Eternal Golden Braid (Hofstadter, 1979, pp. 559-86).

Design is a complex combination of activities, involving both generation and evaluation of potential designs. In order for a computer to design well, therefore, it must be able to handle both generation and evaluation well, or else must handle one so perfectly as to compensate for the other. Yet there are severe obstacles to computerizing either generation or evaluation of designs.

Creating a scheme by which a computer can generate good designs is difficult—more difficult than coming up with good designs. This is because procedural knowledge (learned activities) and declarative knowledge (knowledge of facts) are separate in humans (Best, 1989, pp. 6-9); we often know how to do something (procedurally) without being able to describe it (declaratively). It has long been known that our awareness of our own cognitive actions is limited; Maier (1931), for instance, described how subjects in problem-solving experiments were frequently unaware of the reasoning they used. Verbal protocols and drawings can help fill the gaps, but they are still prone to omissions; while such records rarely provide false clues, they can not be relied upon to be complete (Ericsson and Simon, 1980). Every attempt to codify human design behavior increases our understanding of it, but such attempts are unlikely to ever match the complexity, versatility, or robustness of the processes used by a skilled designer.

Even if we succeed in creating a system that could replicate the design style of a particular designer, we’d still fall short of the design abilities of an architect. It is one thing to codify and follow Wright’s rules of composition; it is another thing entirely to be Wright, and invent a new style of architectural expression. This is one of the characteristics we admire most in great architects: the ability to innovate, to develop new types of solutions in response to new technological, contextual, and cultural situations.

The barrier to evaluation is even more severe. Some types of evaluation—e.g., cost, thermal, structural, or lighting analyses—are performed faster and better by computers. But a large part of design is art, rather than numeric analysis. Does the design look good? How does it make us feel? The ability to answer these questions is based on the ability to respond to the design emotionally and culturally. Any algorithmic attempt to model these human responses will necessarily be inferior: any time the model does not match the human response, it is by definition the model which is incorrect.

Automated design remains a goal of AI research. Architectural researchers have borrowed many AI approaches and adapted them to their own needs: rule-based systems, case-based reasoning, genetic algorithms, etc. Some preliminary results have been useful and informative, but even these perform best in situations where standardization is greatest, and the need for innovation and artistic sensitivity is minimized.
situations like stair design, window design, design of a particular style of house, etc.

Many researchers, therefore, focus on systems where a human designer selects designs generated by a computer, where a human guides the computer’s generation of designs, or where the human generates and helps evaluate designs represented in a computer. Such systems recognize the inherent difficulties trying to get a machine to replicate the creativity and artistic judgment of a skilled architectural designer. As publisher Elbert Hubbard wrote, “One machine can do the work of fifty ordinary men. No machine can do the work of one extraordinary man” (Hubbard, 1903, p. 26). Though the statement was made long ago, it is still accurate today, and it will likely remain accurate far into the future.

**Mahalingam’s Rebuttal**

The argument that it is difficult to create a scheme by which computers can generate good designs because procedural and declarative knowledge are separate in humans has a curious ring to it. When you realize that computation is the processing (procedural knowledge) of stored programs and data (declarative knowledge), this separation should be precisely what makes the computation of design possible.

Designs are created by a sequence of explicit decisions. Design behavior is a larger rubric that surrounds design decision-making. Protocol analyses of design behavior will lead to computational articulation if they focus on design decisions. Psychologists distinguish between affective and cognitive activity. We can articulate design, the cognitive activity, but not design, the affective activity, unless the affective leads to cognitive activity. We may treasure the realm of the affective, but we take out liability insurance only for our cognitive acts as designers. To be unaware of our cognitive actions is a contradiction in terms.

The auditorium design system discussed earlier is based on performance standards and their translation into spatial form. Because of this, the system generates “good” designs. It was possible to create this system, because it was possible to computationally articulate the decision making in auditorium design. Here lies the promise. No machine can do the work of an extraordinary man only if we cannot articulate the work of the extraordinary man.

**My Rebuttal**

The major task in making the design process computable is indeed mapping cognitive activity to computable processes (or coming up with other computable processes that achieve similar results). This can be done in situations where topology, style, and other factors are relatively constrained. Creating an architectural solution in such situations may qualify as design, but humans are capable of much more.

Humans can design in situations where the parti, construction methods, and functional program are not entirely fixed. They consider aesthetic and symbolic questions. They consider what shape rooms should be and how they should fit together, whether more ornate materials are worth the cost, whether to relax functional demands for the sake of maintaining fenestration rhythm, etc. Architects follow sets of mental algorithms when generating these possibilities and deciding whether to implement them, but identifying and codifying these algorithms is a daunting task—much more difficult than codifying design for simpler, more specialized scenarios. Then there is the matter of innovation. How do we algorithmically describe how to come up with ideas that haven’t been done before?

Because of the difficulties of codifying these algorithms, a computer system will not pass an “Architectural Turing Test” any time soon.

**Final Comments**

Ganapathy Mahalingam presents a thought-provoking argument about what it means for a computer to design as well as a human. He further argues that this can be achieved because human cognitive activity can be expressed algorithmically. Conversely, I argue that the design process cannot be completely codified. It is left for the readers to reflect on these arguments and form their own opinions on the subject.

“Binary Oppositions” is intended as a recurring column, a quarterly forum for the thoughtful debate of controversial issues in the area of Computer-Aided Design in architecture. It is my hope that the debate presented above will foster further thought and discussion on the issue. Feedback and ideas for future columns are actively encouraged. If you have comments or ideas, or if you would like to volunteer for future debates, please contact me via e-mail at sven@umich.edu.

**References**


Hubbard, Elbert. (1903). The Philistine: A periodical of protest, 18 (1).


**Scott Johnson** is a Candidate in the Doctoral Program in Architecture, Taubman College of Architecture and Urban Planning, University of Michigan.

**Ganapathy Mahalingam, Ph.D.** is an Assistant Professor at the Department of Architecture & Landscape Architecture, North Dakota State University.