

Automation and the Future of Structural Engineering

By Eytan Solomon, P.E., LEED AP

In July 2021, I discussed the “hot” topic of automation and the future for structural engineers with two of the industry’s leading experts in digital design: **Rob Otani** is a Senior Principal and the Chief Technology Officer at Thornton Tomasetti, and **Zak Kostura** is an Associate Principal and the Americas Region leader of Advanced Digital Engineering at Arup. Below are highlights from our discussion.

There are two things that we call “code.” First, building codes, meaning the rules for structural design – including the jurisdictional codes and model codes (like the *International Building Code* [IBC]) and code-referenced standards (like the American Institute of Steel Construction’s AISC 360). And then computer code, meaning the software programs we use for structural analysis and design. A frequent question is – what is stopping ALL of the written building code from becoming written computer code?

Otani: Nothing is stopping us. It’s just hard to implement. It’s hard to create an app that is going to redirect you to all those other things... A lot of engineering is actually not just designing a beam. It is deciding what beam to design, what material to use, etc.

Kostura: Yeah, I would say that it may be shocking to a structural engineer, but I think building code is typically more forgiving than computer code. A lot of discretion and judgment goes into engineering that you only appreciate when you try to automate it. One of the things that I really had to learn a lot about when I got into coding software is the concept of user-experience design. You really have to plan what you’re doing and the group of users you are trying to help. In our line of work, there is a fundamental question about who uses the software. There’s one line of thinking that you can make software for engineers to assist in completing the design. Alternatively, the end-user could be someone else, such as the architect or owner with less understanding of the internal process. So it’s important to think about the user we are building that software for. It defines what the software will do.

Otani: So, tools like RAM, SAP 2000, or ETABS already have code-based automation in their systems. By the way, they didn’t always. And I would argue that is what makes them reliable tools for structural analysis and design.

Is it only a matter of time before all design is automated?

Kostura: I don’t think so.

Otani: I don’t think it is going to happen in our lifetimes.

Kostura: Is Apple ever going to be done designing the iPhone? No, because design is something that is

self-perpetuating. If you simplify the process of designing, you give yourself the opportunity to iterate that design more and focus on things you would not otherwise be able to. Plus, there are advances in culture and technology.

When you are talking about designing a design process, it sounds like systems engineering.

Kostura: At Arup, before someone gets to the point where they automate anything on a project, in theory, they are supposed to put a workflow together that documents the design process. The workflow diagram allows you to understand the linear parts of your process versus the iterative. It helps you understand the dependencies between the two. For example, if you look at three commercial high-rise buildings, you might say that no one piece of software will *design all three of these entirely*. But when you map the workflow for these three projects, you see that certain steps are common. Those common steps can be a tool or a set of tools. The more you can channel that and think about it in terms of process, the more clarity you have about how far you go with the scope of any software.

Can people be taught to think that way before they have really mastered the job?

Otani: Well, there are a couple of questions there. The end-user is not going to see a lot of that. I think Zak was referring to the person who is mapping this out. I think you know the machine learning apps that we have created over the years. We need independent little physics-based checks along the way because, you know, engineers hate a black box. They love Excel because all the formulas are very clear. The engineers can write their own little checks along the way. What Zak was talking about, in general terms, is called robotic process automation. They use that in the automotive industry, where a very clear road map is needed to identify when and what to check along the way because the engineers just by nature don’t trust anything.

Kostura: Digital practitioners talk about user journeys. What’s the

“A lot of engineering is actually not just designing a beam. It is deciding what beam to design, what material to use, etc.”

Rob Otani



process that the user goes through? What is the user's goal, and what is the process to accomplish that goal? It is amazing to me how many project engineers are unable to sit down and articulate the process. It isn't a technique we prioritize, so most engineers are not getting better at it.

When the author was in engineering school, in the early aughts, there was a requirement to take one coding class. Should aspiring structural engineering students be taking more coding classes?

Otani: I would say yes. I mean, everyone is not going to be a consultant software developer, but you kind of need to know what's possible. Like to Zak's point, if you recognize something that is a real pain point, and you have a little bit of software development knowledge, you can piece things together to automate the process. So I think it's awareness.

Kostura: We are in a time-based business, right? And we know that our profit margins on conventional design will not get bigger in the future. They are probably going to continue to get smaller, so I think we must equip as many engineers as possible with the ability to readily identify repetitive tasks and perform them more efficiently. That has been the role of technology for 40 or 50 years, and I think to Rob's point, a new way is needed with competence and literacy in design thinking and programming. And every organization is going to face a dilemma around what you task an Engineer to know and what you leave in the hands of another professional digital practitioner. A lot of us in engineering are experimenting with Cloud services now, right? You can run up an AWS [Amazon Web Services] instance and run your FE [Finite Element] model there, which is faster. And if you were to use Microsoft, you might choose Azure services over AWS. You might get a price break today, but next year AWS might be cheaper. And if you do the analysis in the middle of the night, it is even cheaper than during the day. You can save money by analyzing when computing resources are less in demand, like electricity. Cloud economics is an example of a field of expertise that will ultimately come into our analytically-heavy industry. Can you expect the engineers to increasingly take that on? It seems like most organizations are ultimately going to come to a point where they have to make a clear decision about what digital skills should be left in the hands of people who practice structural engineering every day or in the hands of another field of conventional building design. They will also need to decide how they cover the other areas of digital knowledge, like cloud economics and many others.

Otani: Yes, but take that one step further. Recently, I did something with a technology-in-architecture practice group. I showed a graph of the software I had when I first started: Risa 2D and SAP 90. That's it – those two things. And by the way, there were only three computers in the entire office. So today, there is probably at least 10 times that amount of software, right? So, the engineer coming out of college needs to know so much more than I knew. And now, the practicing engineer needs to have significantly more

training every year to keep current. This has affected the "shelf life" of our project managers: their shelf life used to be their entire careers. Like, someone who was 50 in 1990 could tell someone who was 21 exactly how to do the job – with tracing paper, the green (AISC) book, whatever it was. And today, the senior folks who know their stuff inside and out have a hard time being a mentor to the engineer who is churning away (with all the new software). And I'm not even talking about Revit, just the engineering tools.

However, most would assume that must be and has been the case for decades in things like aerospace where you clearly have needed more advanced computing for the analysis, and yet they had their gray-haired engineers.

Kostura: The difference is that nobody is expected to make a profit on the first prototype in aerospace and automotive. In our industry, that is precisely what we are expected to do, so we have a lot less rigor than they do.

Otani: Aerospace has ridiculous QA/QC. There are checks upon checks upon checks. For example, you put a new pillow in an airplane, and it has to be verified.

Kostura: You know, it's another delineation between us and aerospace and automotive that you do not have this awkward transfer of risk and responsibility midway through the project, right? From design to construction. You have a team that is incentivized to collaborate and work together. If someone misses something and the model's not correct initially, it is everybody's problem, not just one person's problem. And the way we have carved our industry up makes it harder to undo. So that is a big issue for all of us.

Do you think that will lead to more design-build?

Otani: Yes. I think it will. I think it is going to go in two directions. I think contractors will start to get more involved in design because they can have more influence during that time. For example, *do not use that facade material because we cannot get it for two years*. As well as the engineers going further and producing shop drawings. And having the contractor we know who will build it fill in the rest because we have been staring at this project for two years, right?

Kostura: We are all looking for a way to hedge our profits, right? Because the conventional things we do are getting harder to make money doing. So, what else can you do? What added services can you offer? There are many great ways to extend what you're already doing.

Otani: In the startup world, they talk about making the process vertical. ■

The author would like to thank both Rob and Zak for their insights into these critical topics. There is much food for thought as structural engineers consider how the profession integrates with new technologies and innovations.

Eytan Solomon is a Senior Associate at Silman and a member of STRUCTURE's Editorial Board. (solomon@silman.com)



We know that our profit margins on conventional design will not get bigger in the future.

Zak Kostura

