

# Design Notes Episode 06 - Molly Wright Steenson

Google Design Podcast Transcript

Published February 13, 2018

---

Liam Spradlin: Design Notes is a show from Google Design about creative work and what it teaches us. Each episode, we'll talk with people from unique creative fields to discover what inspires and unites us in our practice.

Aaron Lammer: Hello and welcome to Design Notes. I am your special guest host Aaron Lammer. I taped this conversation at the Google Design SPAN conference this fall in Pittsburgh. I sat down with Molly Steenson who is a designer and writer. She has a new book out, it's called "Architectural Intelligence".

Like most of her work it stands at the intersection of architecture, design, and artificial intelligence. Uh, she focuses on figures like Christopher Alexander, an architect who coined the idea of a pattern language.

Molly W Steenson: It's a solution to a problem that you could apply again and again.

Aaron: She makes a strong case that the dominate metaphor for artificial intelligence is not humanoid robots, but architecture itself.

Molly: Architecture is about building worlds. And AI, in computation, is also about building worlds.

Aaron: She teaches at Carnegie Mellon University's School of Design, and is an expert on pneumatic tubes.

Uh, welcome. Molly Wright Steenson.

Molly: Hello.

Aaron: I really enjoyed your talk yesterday.

Molly: Thanks.

Aaron: Ah, which I found out at the end of the talk, is based on a book that you have coming out called...

Molly: "Architectural Intelligence: How Designers and Architects Created the Digital Landscape".

Aaron: So... the characters, in this story, are people that I was familiar with like, cause, buildings at MIT Media Lab are named after them and I've heard of them, but I, I admit that I didn't really know their history at all. So how did you get involved with this story? What was your first brush with it?

Molly: My first brush with it came out of my dissertation in my master's degree research, but maybe in some ways it came out of a question some twenty years ago. My first day of work at Netscape in 1996, Hugh Dubberly, who is a very well known designer, and at the time the creative director of the Netscape website, suggested to us that we use Christopher Alexander's patterns in redesigning the Netscape website. And I was like "Wow that's weird, why are we thinking of an architect when we're looking at redesigning the Netscape website?" Uh... I had done some research when I was in undergrad in architectural history and really enjoyed it. And that question stuck in my head. And when we saw the rise of things like information architecture, it was really kind of interesting to see again, you know, digital people turning to architectural metaphors. It's a long story about how I, you know, there's another ten years in there.

Aaron: (laughs) The lost years.

Molly: But, um, (laughs) Yeah the lost years. And then I actually went to do a master's and PhD in architectural history in order to start addressing that architectural question: What is the architecture in contemporary digital and the contemporary internet? And I found out that there's a ton, there's a lot of cross over.

So I researched Christopher Alexander, and-

Aaron: Do you have a- do you have, for someone who has no idea who Christopher Alexander is, how do you describe who Christopher Alexander is?

Molly: (laughs) Christopher Alexander is an architect who wrote a book called "A Pattern Language" with his colleagues at Berkeley.

What a pattern is, what he says a pattern is, is, it's a way to- it's a solution to a problem that you could apply again and again and it would work in all of those circumstances. Right? The patterns are all in a hierarchy, they go

from everything from huge to like, nation state, all the way down to minutia, you know like how far, how big a balcony should be or set-ups for bedrooms, or light on two sides of a room, or... um, things like that. So there are 253 patterns.

And this idea gets picked up by a bunch of, um, software engineers, in the late 80s, and interaction designers, so-

Aaron: Is it fair to say like, for those listening at home, that CSS is a kind of an example of a pa- or uh, derived from a pattern language, a series of rules that control visuals.

Molly: Yeah! That and even-

Aaron: In a [hierarchical] manner.

Molly: And even in this sense, you know, patterns, these ideas that these programmers pick up ...

Aaron: Yeah.

Molly: Eventually... um, so people like Alan Cooper, who's the founder of Cooper the design firm; uh, and Kent Beck, the founder, one of the founders of Agile programming and Extreme programming; and Ward Cunningham who created the Wiki; all of these are directly inspired by Christopher Alexander's ideas about design.

Aaron: And if you'd like, combine all the descendants of those things you've covered like half of the modern internet.

Molly: Oh my gosh, totally! And, if you look on Amazon right now, and look up 'design patterns in software', you'll see something like twelve hundred different books. I'm not exaggerating.

Aaron: I believe it.

Molly: So you know, when you look at, um, Google and Material Design, those ideas come from Christopher Alexander.

Aaron: Yeah, yeah.

Molly: But there's something funny about him. He's the architect, that all programmers seem to know, in fact they might only know of him and no other architects, architect architects largely can't stand him.

Aaron: (laughs)

Molly: So, (laughs) So there's-

Aaron: What're their, what's their primary beef?

Molly: Well, he moralizes and he doesn't like form, they think he's a crappy designer,

Aaron: Hmm.

Molly: they find him preachy, but he's too important for programmers, engineers, and designers of all kinds.

Aaron: How did all these guys find- if this work is not seminal to an architect, how does a programmer find it in 1982?

Molly: Funny right? Alan Cooper told me that he found notes on "The Synthesis of Form", which is Alexander's first book, in his junior high school library. And Kent Beck couldn't afford the books by Christopher Alexander, so he'd go to the college bookstore in Oregon, and read them, you know, page by page and then go back, put it on the shelf, and return another time and read a little further.

Aaron: This isn't like a thing that someone found and then spread throughout the programming community, this is like an example of convergent evolution where multiple people who were going to be influential on the web, independently found the same thing and started using it to create original works.

Molly: At the outset, these programmers pick it up in the 80s, a group forms called the gang of four, and they publish something called "Design Patterns in Software", and that comes out in 1994. And, now it's- it's a really, um, central area of research, there're conferences devoted to "Design Patterns in Software" all over the place. And so, um, that gets picked up by the web. Everybody seems to love Christopher Alexander, and uh, yeah. And then the architects can't stand him.

Aaron: (chuckles)

Molly: It's amazing. So that's- he's one figure that I look at in the book, and it's- with that- with that angle of architects not liking him, I was at a conference and, um, talking to a couple of friends who are both architecture professors, and one of them said, "You can't- you cannot write seriously about him. You must not take him seriously". And I was talking to another woman and she said, "Make him the bad guy". And when I wrote my dissertation, I was really kind of snarky about him. And in subsequent revisions of, you know, the dissertation, and then draft of chapters, I think I rewrote the Alexander chapter five or six times- I have no snark left. He's too important. He absolutely shaped what we do on the internet. You can't do anything online without in some way, being touched by what he came up with. And that's pretty fascinating.

Aaron: One of the- the big points of your talk that was very striking to me is: we have this kind of AI is everywhere hype cycle right now, whether you want to talk about self-driving cars, uh, machine learning... we have this idea that AI is something that, like, just jumped out of the closet, and is either gonna kill us or save us, but it didn't exist ten years ago.

Molly: Yep.

Aaron: And these quotes, that these people have, from the 1950s and 1960s, actually less alarmist, they're more sort of like oh of course this is gonna happen.

Molly: Yeah.

Aaron: Like, and not abstract like I'm sorta spitballing something that could become anything, pretty specific ideas about AI that really prefigure like a lot of what we talk about in AI now.

Molly: Yep.

Aaron: So how did they start thinking about this stuff?

Molly: This is what they were doing way back when. So 1955 is when John McCarthy coins the term "artificial intelligence", and in 1956, he writes to a number of his friends at various institutions and gathers them for the summer at Dartmouth, to figure out what the platform is gonna be for research in artificial intelligence. And some of the things that it includes are: neural networks, game playing-

Aaron: Someone sets the-

Molly: Learning, yeah-

Aaron: Someone sets the bar that it's something that would require intelligence for a human to execute.

Molly: That's exactly it.

Aaron: As a blanket term.

Molly: Yep. It's, uh, I think John McCarthy's definition was, um, "it would require intelligence if done by a human, by man."

Aaron: Looking back on this time, as a historian, how different was the way that AI was talked about when there weren't computers? Like I think when people think of AI now they think "oh it's a thing happening in a computer, I don't understand cause I don't understand how computers work". How is AI being thought of in periods before the microchip? Before home computing?

Molly: I mean, it is imagined by computers, so, um, early machine learning, 1952, Arthur Samuel, he taught a computer to play checkers, and it learned from its mistakes and its wins and became a better checker player than the human player. But you know, people like Herb Simon and Alan Newell, you know, we're sitting here in Pittsburgh, having this conversation-

Aaron: Sure.

Molly: And they are some of the most illustrious Carnegie Mellon faculty we've ever had. Simon and Newell believed in about 1957, that it would be possible to simulate the human mind with a computer. Within a couple years, like early 60s, they'd have this licked.

Aaron: This close. (laughs)

Molly: (laughs) but, but it begs a really interesting question.

Aaron: Yeah.

Molly: Um, how we conceive of intelligence and how we model it. You know, there's another person, uh, Douglas Engelbart, who is the inventor of the

mouse, among other things. But in 1961 he wrote a proposal for something called augmented human intellect and his first example in this proposal is an augmented architect. This is 1961. What does Douglas Engelbart know about architecture? From what I can tell, you know, probably not a ton. But I think that architecture is about building worlds, and AI and computation is also about building worlds.

And so I think that this, this intersection of architecture and AI is almost very natural.

Aaron: When I go through, uh, the list of the speakers here. I just- uh, I talked to Madeline Gannon about this robot arm that, um, she made. Almost every speaker here's work at least touches on these ideas.

Molly: Yeah.

Aaron: What did your research about the 1950s and 60s lead you to think about AI right here, right now in Pittsburgh alive in 2017.

Molly: One of the architects that I write about is a guy named Cedric Price; and he turned design problems kind of upside down.

Aaron: Mm-hmm (affirmative)

Molly: So that they could be addressed in different ways. And he was very interested in rethinking the role of how a user or, or human interacts with a building or what even a building is. He didn't wanna lock it down in space and time. Uh, and he wanted to blow open the role of the designer and the, the computer as well.

But to the point of Mimus, Madeline Gannon's robotic arm-

Aaron: Yeah.

Molly: Uh, this is the, the robotic arm that was, um, th-that was kind of behind an enclosure at the London Design museum.

Aaron: Yeah, displayed in kind of a zoo manor.

Molly: Yeah! And you have to, you- you try to capture its attention.

Aaron: Yeah.

Molly: Her attention. And as she's swooping around you try and do something that's gonna make her notice you and interact with you. And then she'll get bored and go away from you.

Aaron: Yeah.

Molly: Toward someone else, if- if you get boring. Well this idea is something that Gordon Pask started playing with in 1953 with something he invented called the musicolour mach-machine. So you'd have to interact with it and play music and it would be a mobile, sort of mobile. And the parts would move, and if you weren't interesting enough it would get bored.

Cedric Price picks up this project with his Generator project. Which was instead of cubes, this was never built, but instead of 12 foot by 12 foot cubes and, um, walkways and ramps just sort of a recombinable center.

Aaron: Best way I can think of it is when I was a kid, in the Bay area, there was an attraction called "The Wooz". And The Wooz was a hedge maze-

Molly: Whoa!

Aaron: That was changed at various junctures so that you would never have the same Wooz. But The Wooz is basically like a giant Cedric Price sculpture.

Molly: That's amazing.

Aaron: The goal for Price wasn't to lose children in The Wooz it was to have a building that was constantly renewable.

Molly: Changing. And changing according to people's whims. And he realized that maybe people would not want to have their building move around and, and so he worked with a group of- with a couple, um, John and Julia Fraser who are programmers and architects to come up with a set of programs. And so they proposed putting microcontrollers on all of Generator's pieces.

Let me point out, this is- this project took place from 1976 to 1979. So we're talking about some crazy stuff here right? We're talking about stuff that was roughly 40 years ago. But the best program, and this is where we connect to Madeline Gannon, was the Boredom program, because if it got bored of its users and it hadn't been recombined enough it would come up with its own layouts, its own plans, its own menus, and then hand them off to the crane operator, Wally Prince, who would then take



the mobile crane and move them around. So this is a building that you- that can get bored with you. How cool is that?

Aaron: Was any of this stuff actually built or-

Molly: No, No.

Aaron: Are these all- all theoretical?

Molly: It was never built. It wasn't theoretical, um-

Aaron: Right.

Molly: It- it was a real project with a real client and I have pored over hundreds of drawings of the thing.

Aaron: It doesn't seem like computer chips were at- in a place that could have supported-

Molly: Right?

Aaron: Anything like- I mean. This vision would have been hard to execute in the 1990s.

Molly: Yes. Exactly. But they kept trying. You know? That idea is still a prescient idea. It would still be a weird and awesome thing to interact with this set of cubes and stuff that would get bored with you.

Aaron: It's a little like- the like modern idea of the Hackathon.

Molly: Mm-hmm (affirmative)

Aaron: But it's less- they seem less oriented around like "we're just making stuff". Like the ideas and the crosspollination of the ideas seems as important as executing those ideas. Which I guess is something you have to be comfortable with if you're ahead of your time.

Molly: Yeah.

Aaron: And you're trying to make a house that reconfigures itself in the 1960s.

Molly: Yeah.

Aaron: You've gotta be a little... willing to compromise on physical reality.

Molly: Nicholas Negroponte has written a number of books. One of the books that he wrote is called "The Architecture Machine". This is the book that he wrote in 1970 and dedicated to the first machine that can appreciate the gesture. So this is- this is what he says in "The Architecture Machine", he says, "My view of the distant future of architecture machines, they won't help us design, instead we will live in them. The fantasies of an intelligent and responsive physical environment are too easily limited by the gap between the technology of making things and the science of understanding them. I strongly believe that it is very important to play with these ideas scientifically and explore applications of machine intelligence that totter between being unimaginably oppressive and unbelievably exciting." 1970.

Aaron: Wow.

Molly: And that's- that's the crux of this stuff, right?

Aaron: So what do- what does someone who's like obsessed with AI now think when they realize that Nicholas Neg-Negroponte said something like that in 1970?

Molly: You know, what I hope is whether I'm talking to students or, you know, whether it's somebody reading my book, what I hope is that people realize that these ideas have long histories and it's not just 22 year olds working on them but actually if you consider it it's really some 70 years of computing and some of these ideas, you know, are- are centuries old we just have the technology.

I think you're less prone to make stupid mistakes if you understand what has happened previously. One example I think of is, it seems like one of the low hanging fruits for people working in design has been trying to work on conversational user interfaces, or chatbots. People have learned that they're really hard to get right, but you know what? Joseph Weizenbaum, who created Eliza, could have told you that in 1965.

These are hard problems. I guess, you know, when I consider conversations about the singularity or when I consider things like, you know, Nick Bostrom's argument about super-intelligence, I also think that quite often the toilet flushes on the automatic toilet when I don't want it to and then I wonder if the toilets won't flush the way we want them to when they're automated. Is AI really gonna be taking over the world and leaving us all jobless in the next five years and I'm not so certain.

Aaron: When I listen to you talk about this, I looked out over this sea of young people, um, who wanna do this kind of stuff. What was so striking to me was these thinkers were thinking 40-50 years in the future. They weren't thinking about a product they could make tomorrow. I mean, there does seem to be a utopian level while they were trying to make it tomorrow, but they were setting the bar way out ahead of themselves. And there's something about ideas that are generated when you've got that 50 year stare as opposed to ideas that, like, you could maybe get funded as a start up right now.

Molly: Right now, yeah. You model intelligence with whatever you have at hand. You know, I assume 2,000 years ago people had artificial intelligence, they had novel ways of communicating at great distances, long before we had telephones and telegraphs. Right? Here's what I wonder. Are we modeling the same thing, using the tools at hand? And maybe these ideas are actually very old.

Aaron: It does not surprise me that this book and this line of thinking comes out of the history of architecture, because buildings are the things that we have to think about in the 100 year span. We do not-

Molly: Mm-hmm (affirmative)

Aaron: think about cell phone design in the 100 year span.

Molly: No.

Aaron: We do not think about- really we don't think about almost anything in the 100 year span.

Molly: No.

Aaron: And- except potentially like being able to live on the Earth and what buildings will look like because we have a pretty good guide that buildings last over 100 years.

Molly: Yeah.

Aaron: If they're properly built.

Molly: My house of the future is 120 years old.

Aaron: Do you think that was- those gu- why this started with architecture? Was because of that length that it has?

Molly: I think that's part of it, and I think metaphors.

Aaron: Mm-hmm (affirmative)

Molly: You know, we all know what a building is and we all have an idea of what a city or a town is and we have our own feelings of what it is to move through those- those spaces. Those metaphors were powerful ways to think of structuring the early chaos of the world wide web in the 1990s when the web was a much uglier place than it is today.

I also think the longevity is nice and I wish there would be more thought given to the ramifications of desi- design decisions and funding decisions of start ups and technologies but the other thing about architecture is it gives you an interface. Right? All technologies need an interface for people or even programmers to understand them. And architecture was one of the first places where those interfaces developed.

Aaron: How can people who want to read this book, um, find it?

Molly: You can go to my website, which is [girlwonder.com](http://girlwonder.com). And you can find it on Amazon. You can also find it- it's MIT Press' website as well.

Aaron: Google can be used to locate this book.

Molly: This is right.

Aaron: Um, well thank you very much Molly. It's been great conversation.

Molly: Thank you so much Aaron, it's been great.

Aaron: That was Design Notes, a very special episode, taped by me: your guest host Aaron Lammer at the, uh, SPAN conference this fall in Pittsburgh, which was put on by Google Design. Uh, thank you very much to the regular host of this show, Liam Spradlin, this is what he had to say about why he makes this show in the first place.

Liam: So, I'm a big believer that talking to people from other creative disciplines really feeds into my own creativity. So the ulterior motive is really that I'm just interested from hearing from people working on different stuff.

Aaron:

You can find out everything you need to know about "Design Notes" by going to [design.google/podcast](https://design.google/podcast). And also subscribe to their newsletter at [design.google/subscribe](https://design.google/subscribe).

Thanks to Liam Spradlin, Amber Bravo, and Hailey Peek. They all helped make SPAN 2017 happen. We taped 4 incredible interviews at it that will be put online. I hope you check them all out and subscribe to "Design Notes". Thanks for listening.