

## **DISRUPTIVE: BIOINSPIRED ROBOTICS Pt.1**

Host Terrence McNally interviews Radhika Nagpal. Podcast published July 27, 2015.

### **McNally:**

Hello, welcome to DISRUPTIVE the podcast from Harvard's Wyss Institute of Biologically Inspired Engineering. I'm your host, Terrence McNally.

The mission of the Wyss Institute is to: Transform healthcare, industry, and the environment by emulating the way nature builds.

Our bodies — and all living systems — accomplish tasks far more sophisticated and dynamic than anything yet designed by humans. By emulating nature's principles for self-organizing and self-regulating, Wyss researchers develop innovative engineering solutions for healthcare, energy, architecture, robotics, and manufacturing.

They focus on technology development and its translation into products and therapies that will have an impact on the world in which we live. At the Wyss, folks are not interested in making incremental improvements to existing materials and devices, but in shifting paradigms. In this episode of DISRUPTIVE, we will explore: BIOINSPIRED ROBOTICS.

Many of the most advanced robots in use today are still far less sophisticated than ants that "self-organize" to build an ant hill, or termites that work together to build impressive, massive mounds in Africa.

From insects in your backyard, to creatures in the sea, to what you see in the mirror, engineers and scientists at Wyss are drawing inspiration to design a whole new class of smart robotic devices.

We're going to explore this exciting territory in a three-part episode of DISRUPTIVE, featuring three members of the Wyss faculty, CONOR WALSH, ROBERT WOOD, and RADHIKA NAGPAL.

### **[01:30] Nagpal's Bio**

Today's guest, Radhika Nagpal is the Fred Kavli Professor of Computer Science at the Harvard John A. Paulson School of Engineering and Applied Sciences (SEAS) and Bioinspired Robotics Platform Co-Leader and a Core Faculty member at the Wyss Institute. Naming her one of the ten scientists and engineers who "made a difference" in 2014, NATURE Magazine wrote that her "self-organizing, swarm robotics are today's state of the art in collective artificial intelligence."

Radhika is developing programming paradigms that enable new types of autonomous robotic systems to mimic the collective behaviors of living creatures to meet real-world challenges. Inspired by social insects and multicellular systems,

she's developing sensor networks that monitor the environment, and robots that collectively construct or self-assemble complex structures without human supervision. Her recent work includes the Termes robots for collective construction of 3D structures, and the Kilobot thousand-robot swarm.

Welcome, Radhika Nagpal to DISRUPTIVE...In order for listeners to get a sense of you as a person, beyond your work and ideas, Radhika, can you take us back and tell us a bit about your path?

[02:37]

### **Nagpal's Path**

#### **Nagpal:**

I grew up in a small town in India called Ambrutha, and I think my life's goal was to leave a small town and enter a big one. I think a lot of my path has been focused not really with a plan but just enjoying the opportunities that came along. One of my first great opportunities was to come to MIT as an undergraduate. I experienced for the first time computer science and fell in love with the subject. I fell in love with programming.

Of course I fell in love with a lot of subjects so it was kind of hard to choose. But maybe ironically the subject I disliked the most was biology. So I proceeded through life thinking I would be a hardcore computer scientist. I didn't like hardware or mechanical engineering and I didn't like biology, and now I do bio-inspired robotics. So I think a lot of times paths don't quite go the way you start them.

#### **McNally:**

I've interviewed people for years and in almost all cases people end up achieving something of significance, it's been this accident, this twist, this turn that ended up leading them there.

#### **Nagpal:**

Absolutely. I think it makes you more appreciative of all the accidents and "oopses." Things that people label sometimes as failures actually turn out to be the turning points that lead to future successes. It's good to keep in mind sometimes when things aren't going well that maybe this is the learning point that will make something really amazing happen.

When I was starting graduate school I was really interested in networking and computers and distributed computers. And at the time there was a new project called Amorphous Computing, and the idea was that one day we would make computers the way biology makes cells. You would just make so many cells and together they would accomplish amazing things. But the cells themselves would seem really limited and small and replicable compared to, say, the whole system which had millions of cells or million and a thousand of cells, but you would never babysit or worry about individual cells.

So somehow you would get this amazing collective power from a system that was really sort of disorganized in a way, with many individual parts that seemed very simple and unreliable, but as a group would have this power. And that was a powerful idea. There were a lot of fields that were born around the same time: sensor networks; synthetic biology actually was also an outcome of that same project area. A bunch of us got excited about this idea that biology crossed computer science or physics crossed computer science.

At that point I started reading about biology, and it's funny how later in life you read these things and they're so much more exciting than I remember them being. I read about the development of the fruit fly, how cells cooperate to make these exquisite structures. Then later on I realized that, of course, ants cooperate to make exquisite structures, and all these examples where each individual seemed so limited but what they can do together is this incredible complex global phenomenon.

That got me very excited about what kind of programming goes into that system, if you thought of those systems as having behavior, as running algorithms, what algorithm makes that possible? And I think that's really the path that took me first to thinking about the algorithms, and then eventually to building the robots.

[05:55]

### **From Computation to Building**

#### **McNally:**

It is interesting the twists and turns. At what point in there does it shift for you from computer science, algorithms, computations, to envisioning creating the organisms or engines, as you will, that will put that stuff to work?

#### **Nagpal:**

That was an interesting path too, because for me that path started after or during the time that I was a junior faculty member. I hadn't really thought that I would start out and have a lab and do work. I had friends who built hardware and I thought I'd collaborate with them. I had friends who looked at programming cells in the beginnings of synthetic biology and I thought I'd collaborate with them.

But I think after a while you realize that there's a very deep connection between the body and the mind, and it's really hard to separate those two. When you separate those two apart from each other, you lose a very deep connection and miss the fact that sometimes we think with our hands or with our senses, we don't think with our brain. And that idea that your body can do part of the computation for you is very powerful. There are a lot of people who understand this, but our understanding has to translate into engineering.

So at that time I started to realize that I really wanted to build. I really wanted to have tangible examples so that I could make the computation simpler by choosing the right mechanical structures, in the way that ants have or termites have or birds

have. With any of the biological systems, there's a lot of computation that's happening because of the way their body is shaped.

So I wanted to learn more about that, which sort of took me more into mechanical engineering. Robotics is a very interdisciplinary field. You cross all the way from looking at how this is accomplished in biology and whether it's the brain or the body or the colony that's actually achieving the right behavior. Or at the other end, which is how do we replicate or build those systems with the kind of principles that we have in mechanical engineering, where we can't use tissues or cells but we can use compliance and we can use ideas like simple sensing feedback loops. So I think that's what really moved me past the idea that computation was the only piece that was interesting in a system, it's really the whole system.

[08:11]

### **Computation Beyond the Brain**

#### **McNally:**

It sounds like what you ended up saying was, computation takes place throughout the system or throughout the organism, and suddenly you can't just tinker with what might be the brain piece...

#### **Nagpal:**

Exactly. So if you think of computation happening in the body...For example, maybe the way I walk is made better by the fact that my feet are shaped a certain way, so I can walk up slopes or my body automatically balances to certain things. Then my body is doing a computation that's helping with the task. If you go all the way to the highest level, a group can do things that an individual can't do; or a group can figure out and sample the environment in a way the individual can't, and so now the brain is really the colony, no individual in the colony actually has the full picture. The full picture is distributed over these many, many minds.

So at every level there's a kind of computation happening, and it's interesting to understand how we take advantage of each of those pieces. In my case, I am really interested also in that top level, like how do you get the most bang for your buck from numbers? And as a result be able to reduce the capabilities of the individual to the point where they're really like ants. So that your investment in the individual is small and your investment in making many of them is large, and it's really that distributed group that's going to be the one that solves the problem. It's not the individual who has the capabilities that are so amazing.

[09:40]

### **A Whole Life is Bigger than Tenure**

#### **McNally:**

We're going to return to that, but before we do, I want to talk a little more about you and your approach. In July 2013 you published an essay length blog post on the site of *Scientific American*: the title was *The Awesomest Seven Year Post Doc, or How I Learned to Stop Worrying and Love the Tenure Track Faculty Life*. Now to

honor what you talk about in there, which is your commitment to balance or, as you put it, to a “whole life,” we’d be remiss if we didn’t talk about that, if all we talked about was work. So can you share with listeners a sense of what you wrote there, and how you evolved your approach?

**Nagpal:**

Absolutely. I think that everybody’s career has different paths, but very early on I figured out that as a woman, people really had a very rigid view of what my path could be. A rigid view of how things were done and the fact that as a woman I wouldn’t be able to do them. And that was really very difficult for me to accept. On the other hand, not accepting it meant I got a little used to rebelling. So by the time I became a junior faculty member I was pretty used to being a rebellious woman.

I found that often there is this notion that you have to work amazing hours; you have to sacrifice everything in your life, and that’s what’s going to mean you’re dedicated. And that’s a really difficult way to start your life, to think that everything else is supposed to be “less than,” and if everyone else in your life isn’t ready to be less than, then somehow you’re a failure at your chosen path. I think that message is promoted in way too many ways. Of course, I started in 2004 when Larry Summers gave his famous speech, so for me that was up right in front when I started.

My article really was about, “If you don’t believe that, then what do you believe?” What can you believe that makes you go through life? And I actually believe that the best scientists are the ones who have whole lives, who aren’t being a scientist to the exclusion of being a good citizen, being a mindful person, being a thoughtful person. The truth is, if you’re running all the time you definitely don’t have time for thought.

So how do you create that system where first you believe that’s true, you believe that by creating balance in your life you’re actually creating better science. Then the question is, how do you practically do it? And it took me a lot of trial and error on the practice part. I think I believed it somehow in the beginning, but it took a lot of work to figure out how to practically work in a system that was not encouraging that behavior at all.

[12:27]

And so at different points during my career I talked to people about my strategies and I wasn’t very shy about it. I didn’t advertise it openly, but a lot of people I know knew everything in that article. And they were often shocked. They said, “I don’t think people know this can be done...” “I don’t believe you can do this...” It’s like, “Are you sure, would you ever say this out loud?”

This went on for a while, and then I got really upset. I thought this is ridiculous, how can people go with these false views of life? I guess my article was kind of an attempt to come clean. I didn’t want people to think that I was willing to be that person. Even with tenure, I’m not willing to be that person. So I wanted to come

clean and I wanted to put it out there that I don't believe that a system that encourages people to give it all up in the name of science is a good system.

**McNally:**

I remember the line that struck me the most in there. You're writing to the public and you say in that article, "If a department or a university decides that I'm not giving that percent of my life to them that works for them, well I guess I'll go somewhere else, and if a marriage partner felt that I wasn't giving..." It seems like you're really willing to say that the whole may not fit with people and you've got to find people that it fits with.

[13:55]

**Response to Her Scientific American Essay**

**Nagpal:**

It's interesting. You'd be surprised, I got so much email afterwards. And to each person, there was a different part of the article that spoke to them. People reflecting back to me what I said has been hugely beneficial to me. It's made me much stronger and much more committed to thinking openly about these issues and also understanding the issues from other perspectives.

So that is one example. I think that what I was trying to get is that all of us fear failure. Right? We fear failure in a big way. And in society we set up certain things to be all or nothing. Marriage is one of them. Divorce is like, "Oh my God, you know, marriage is supposed to be for forever," and tenure is supposed to be for forever. And I thought, "Wow, this is really setting you up, because now you look at that and the fear that something that's going to be for forever doesn't happen to you - and you're devastated."

So how do you attack that? Well, first is to realize nothing is forever. I've decided that is actually the stupid thing. You can quit a job and you can quit a marriage. And to me, the possibility of quitting a marriage was very important. I saw a lot of women in my family suffer because divorce wasn't allowed. So as far as I was concerned, divorce is one of the best things that comes with it. But the flip side is, the other person can divorce you too.

So I figured, I can leave Harvard and Harvard can choose not to tenure me, and that is an acceptable choice versus being an indentured slave. On the other hand, I realize that having kids is the one decision that doesn't come with that, so once you've had kids, there is no quitting.

**McNally:**

That is forever...

**Nagpal:**

That is forever. So I think actually that's the decision people should really be thinking hard about, and that's the one that should invoke fear. Because other ones... I think it's good to remember that there's lots of paths through life and most

of us don't know what our alternate paths would have looked like. We don't know that we would have been unhappier somewhere else. So if you don't expect everything to be forever, it reduces the bar - I'm not failing forever.

I think what I was trying to convey, is that yes, I do fear failure, but that's just how it is, there is the possibility I fail and there's no way around that. So let's just go with it.

[16:15]

### **Success and Failure on Her Own Terms**

#### **McNally:**

But it sounds like you wanted to set the terms of what failure for you was. Failure for you was failing to demand at least the possibility of a whole life.

#### **Nagpal:**

I agree. A sign that said, "If I succeed, it will be on my own terms," was pasted on my desk for most of the seven years so that every day I could read it and remember it.

Of course, towards the end I realized that I still want success, I still want to get all the things I want. But I've seen a lot of people live really great lives after what would seem like really traumatic events. So I personally believe in, I don't know if restoration is the right word or recovery or rejuvenation, but people can recover from amazingly difficult things, and tenure is just not one of them. It's not, it's really not. I have a cousin who lost a child. I mean, can you imagine? That chokes me up. It happened some twenty years ago and it still chokes me up because I got to watch part of it. Compared to that, I'm sorry, you've got to really scare me. So I do believe that my terms are better than what other people were promoting.

[17:35]

### **Does Pursuing a Whole Life Lead to Better Science?**

#### **McNally:**

It would be stupidly hypothetical for me to ask if you believe you would be as successful in robotics if you had not taken certain actions toward living that whole life years ago. But can you at least talk about how you think, for you, the two sides, the strong approach to work and the strong approach to life, have fed each other, or not?

#### **Nagpal:**

I think there are lots of ways to look at that. First it's really hard to dissect your own life. I know certain things about myself. When I get unhappy, I quit. So that's not a good way to succeed generally, that if you get pushed to the corner, you quit. So a lot of my strategy is around keeping a big distance from the quitting point. So thinking well about myself and living a full life where there are many things in my life, many things in which I am making progress, that is one way of keeping myself away from quitting. Not all of your self-esteem is invested in one thing.

I think this works within research too. We try to do different projects. We teach, we do research, and we mentor our students. You always try to remember that there are many ways in which your life is valuable. Therefore, when one thing doesn't go the way you planned, rather than get too hung up, you try to fix it, you try to understand it, you try to learn from it. You don't sit there and label it as a failure and cause yourself to go down some downward spiral. Because, as my adviser so wisely said, "We spend way too much time labeling everything a success or failure, and if we just stop doing that all the time, we would actually be happier people."

So part of it, I think is that the ability to have many ways in which I felt my life was valuable has made me more risk-tolerant in research. I can take more risks. I know that I'm doing something because I'm passionate about it and I'm excited about it and it's risky and it may not work.

I do have an obligation to protect my students, so that's where it becomes a little harder to take risks. But personally I am okay with who I am, and therefore I can take that risk. So I do think that in a way I've done a lot of risky projects that maybe other people are not so comfortable with. And in so many cases it's paid off, and so I find it hard to go back and think that wasn't the right thing to do.

I've always worked a little bit in biology, which is not easy if you're a computer scientist, because how does a computer scientist evaluate your impact in biology? It's also not easy for the biologist, because you only spend a tiny amount of time in biology. So people say, "Oh that's too risky to do." But every time I've worked with biologists on biological questions, it's fed back into how I think about robotics.

In more direct questions like - How do cells solve problems? How do insects solve problems? How does that give us new algorithms? That's definitely true. But also less direct questions like - How do you think about an experiment? How do you think about an experimental question in biology and computer science? What is enough proof that something is happening a particular way?

You can rewrite programs in computer science, but when you run a robot, there are a lot of other things happening, and it's interacting with the world and it's sensing. You don't know all the things that are happening. So you're going to run experiments to try and understand why things are happening a certain way, and in that sense, we're basically replicating exactly what the biologist taught me.

There are so many ways I feel the different things I do feed into each other, and it's very hard to separate them out, but I think also that cross-disciplinary stuff helps in many ways.

[21:17]

### **Learning from Non-Scientists**

#### **Nagpal:**

Another example is that my kids will come and ask things about what I do, and I

will go to the Cambridge Science Festival and explain to people what I do. As I'm doing that, the things that people say back to me give me so many ideas. I see so many ways in which they view my work that's different from the way an academic would view my work or another professor or another graduate student. They really look at it differently, and sometimes those conversations have led me to work on different topics. So I like to have a diverse experience in life, I think it makes for more interesting science.

[21:50]

### **Harvard Women in Computer Science**

#### **McNally:**

You made very clear in that article how protective you are of your time and your commitments. Are you still functioning as an adviser to Harvard WICS (Women in Computer Science) and what does that mean to you?

#### **Nagpal:**

That is one of the highlights of my last many years. The Women in Computer Science group started with a very small number. Actually it started and restarted many different times but never took off. Then about four years back or so it started with a small group, and today there's about three hundred members on the mailing list. We talk about articles. They host a whole conference where students from the New England area come to Harvard for a day of events and Hackathons. It's an amazing experience.

I have to say there are a lot of times I feel very depressed about the situation for women in academia and the sexism that seems to persist, and it's really the young people who keep me going. So I tell the Harvard WICS, it's not me who is their adviser, they are actually my advisers, because they're telling me how things are going to change. They're committed to changing, and with that group I feel much more optimistic about the future. I feel that these women will go out and demand that people meet their terms. They're not shy about it. They understand a little bit better that everybody has pluses and minuses, and it's not okay to target them with only minuses because they're female, but it's also not okay to assume that you know everything because you're in computer science.

I love the fact that they really deeply think about the articles that come out and the things that happen and they give me a chance to talk to a lot more people about how something affected them or made them feel.

I used to feel very alone. When some event might happen, most of the men around me don't want to talk about it or don't really want to even bring it up, and so you read something and your heart just shrinks and there's nowhere to go. Now, in an interesting way I can talk to the Harvard Women in Computer Science. I think at the beginning I felt very alone. So a lot of the Harvard women are my advisers.

[24:05]

### **Integrating Biology and Computer Science**

**McNally:**

We're going to shift into the work now. This is a question about how you integrate the biology and the computer science. I've read that a couple of your team on the Termes Project, Harvard graduate student Kirstin Peterson and staff scientist Justin Worfel, actually spent time in Namibia observing and studying termite mounds. Have you actually gone to any of that type of biological observation?

**Nagpal:**

Yes, I got to go to Namibia too. I think part of this was a deep envy that I had been feeling over a long time, over the fact that biologists get to go to all these amazing places and I only get to read papers. So the Namibia trip was our first example.

Actually termites are a classic system where people have studied decentralized cooperation or implicit cooperation. The idea is that with termites, there's definitely no supervisor who's telling them what to do, there isn't even some sort of hierarchy. Instead there's sort of these chance encounters where information is propagating, but there's also just encounters with the environment.

Imagine if people together were building a puzzle. You don't necessarily need to talk, you could sort of see what other people are doing. You can see the puzzle pieces that have been placed in the past, and you can sort of infer from past actions, from the environment what past actions have happened. So people believe very strongly that termites are one of the key examples where this is true.

[25:30]

**Distributed Systems**

**McNally:**

Can you talk a bit about three big terms that show up in a lot of your work - : self organization; self assembly; distributed systems. What do those terms mean to a listener or a layman? And examples? And the role they play in your work?

**Nagpal:**

I think that the distributed systems is a good place to start, and it's often really easy to understand these systems if you forget about termites and just think about a group of people. You have a group of five people doing something, it's really easy to get together and for everybody to be in the same place and discuss things. Maybe as you get a larger group you can have cell phones or something. But now imagine a hundred people trying to do something together. It's not really possible for me to know what the other ninety-nine are doing, and they've distributed over space, so they're in different locations; they're seeing different things; they're moving at different rates. And the flow of information cannot be that I know every millisecond every millimeter that every one of the hundred people moved.

Instead, if you go to the other extreme, maybe I don't really need to know what the individuals are doing. I can just do my bit and if I know that everybody else does

their bit, then we kind of understand that all of these bits will add up together in a good way; that they won't add up together in a bad way.

The field of distributed computing is really about algorithms that function that way. You could think of computers connected over wires, that's a very traditional system, but the Internet as you can imagine is a huge distributed system. In the future self-driving cars on your roads will be an incredibly different distributed system. So in all of these systems each individual can only integrate information that's close to it, but you want the system as a whole to achieve something correct.

[27:16]

### **Self-Organization**

#### **Nagpal:**

In self-organization, it's kind of the same idea, except the assumption is that the individuals are very much more myopic. So I really don't know... I don't know if there's a hundred other computers or a thousand other computers in my network. I don't know if we're connected in a tree or connected in a flat network. I know very little about the system, and I can operate on very little information and still the system can achieve what it needs to achieve.

So if you think about a distributed system and you say, okay make each individual in the distributed system simpler and make more of them, then you're sort of moving towards the field of studying self-organization.

[27:55]

### **Self-Assembly**

#### **McNally:**

...and self-assembly?

#### **Nagpal:**

Self-assembly is a very specific kind of self-organization. It's also something that isn't usually studied in distributed computing, which is "How do you form structures?" In physics, self-assembly is - How do snowflakes form? That's a good classic example of a really beautiful self-assembly. And the assumption is not that there's a water molecule who knows exactly what shape they want to form and they collect and talk to the other molecules and tell them exactly where to stand until you have the perfect shape. Instead somehow there's this process where a structure is able to form but the individual rules are actually quite simple; they might be rules of just adhesion and repulsion, and somehow the adhesion and repulsion or the specific way molecules fit together means that the assembly proceeds in a particular way.

As a distributed system, you could say you just have this bag of stuff and you shake it and now suddenly, even though each individual part only can interact with things near it, you're supposed to get the end structure that you want. So the field of engineered self-assembly is basically - How do we make the molecules or the

robots or the cells that self-assemble the thing we want, as opposed to the thing that they're naturally going to do?

I'm interested in self-assembly because at one level you could think of the future of creating programmable materials the way we have 3D-printers. Imagine materials that could self-assemble into one structure and then you melt them and self-assemble them into another structure. So there's a lot of emphasis on this idea that we could create these kind of active materials that could change shape.

And at the other extreme, if you think of robots putting together a house, that's also a form of assembly. The parts of the house are not putting themselves together, they're not smart beams, but actually the very act is exactly the same. So any kind of assembly process where many individuals are assembling together is essentially a larger version of self-assembly.

So I found that that's one of the interesting areas... You have to ask if you're going to make a structure that has something very specific at the end... If you want to make a fruit fly, how much of the picture of the fruit fly needs to be in an individual cell? That's essentially the question. What's the program that goes in the cell that self-assembles a fruit fly? And I think there are so many people who are fascinated with this question from within biology to physics to synthetic biology to computer science, that there's just so many ways to think about the problem and I find it really exciting.

[30:30]

### **TERMES – Robotic Construction Crew**

#### **McNally:**

In February of 2014 your research group announced TERMES, the creation of a robotic construction crew, capable of assembling blocks into 3D structures without any human intervention; no foreman, no central brain. What is the story behind this? What inspired you? And then a little bit of the path... What's a little bit of that story of how you got to that moment?

#### **Nagpal:**

I actually started working on this project I want to say around 2004, and it was really my co-author, the first author of the paper, Justin Werfel, who came to me with this idea. His idea was just in simulation, just in a computer model, in mathematics, what would be the program that a termite would run or an ant would run to build a structure that had certain qualities? You know, if I wanted to build a wall or I wanted to build a particular shape, what would the individual need to know? And how would it happen if they were ten individuals or twenty individuals? Would it matter? Could it just go faster if you put twenty and slower if you put ten, but you didn't have to change at all the rules by which anything was being built?

He was really inspired by all of these papers on termites and wasps and ants and the construction rules that biologists were uncovering. So he got me really excited about the project, and I have to admit after that I read a whole bunch of papers

because of him. And we started working on it, and we were really surprised. We realized that there were simple ways about thinking about the formation of structures so that you could actually take a really complicated structure idea and write in computer science what we call a compiler. A compiler is taking one description and transforming it into another, and it has a mapping between one description and the other.

So we could write a compiler where you could essentially draw a shape and out would pop these rules that individual simulated agents could run around and build these structures. And this was super exciting. Justin Werfel's PhD thesis was about this topic.

[32:30]

The thing though was there were so many ways in which we were living in this completely abstract world. When I talked to roboticists about this, they just shook their heads and they were like, "Yeah...we ain't never gonna build that."

I had a really great friend, James McLurkin, who won numerous awards. Actually some of his robots are in several museums. He was building robots, and he was one of the first people I knew who had a swarm of robots. He would show up with this suitcase, open it up and his fifty robots would come out. It was incredible, nobody had that.

He taught me new ways of thinking about robots. He taught me about what things are easy and hard, what might seem easy in math but would not be easy to build. And I think over time Justin and I got more and more excited about this idea that we could meld those two ideas. If we thought harder about a robot that really could be built, then we could design algorithms that operated in that model. We didn't have to have algorithms that were not so well correlated with reality. That actually led us in this direction, and we actually built a couple of Lego Mindstorm robots. My start into robotics was using Lego Mindstorms.

Then Kirsten joined our group. Kirsten independently in Denmark had been thinking about the same ideas with her Master's thesis. How would you build robots that could move blocks around and climb over them? How would you make three-dimensional structures?

So just to give you a sense of what's challenging: robots that climb are a cutting area of research; robots that manipulate are a cutting area of research. Now you're asking a robot to pick up something and climb over the structure previously built while carrying this thing and then place it well. You're basically taking the worst of climbing and manipulation and sticking it together. So the question was, is that actually going to make it easier or harder?

[34:30]

Your first impression might be, that's just taking the worst of both worlds and putting them together. But the flip side is that a lot of times we build structures, we work with tools, right, we work with tools that fit our hand. We don't work with tools that only a chimp can use, and chimps use tools that are appropriate for

them. Birds use tools that are appropriate for them. So customizing both the material you're manipulating and the robot that's manipulating it turns out to be a very key thing that can be exploited. And the simplicity that termites have very special mouth parts that help them manipulate mud in really well ways, they're not good for manipulating other things.

So we built our robots to be really good at manipulating particular kinds of blocks and we changed our algorithms to think about how we would compensate for the fact that robots can't build things that will fall over halfway through or they can't lock themselves into a hole, or they're not going to be able to communicate with other robots that are behind a wall or far away, so can they do it without any communication whatsoever.

We melded all of these different ideas together. Justin's really deep specialty is mathematics and algorithm development and thinking about decentralized systems, and Kirsten is an amazing builder, and I'm just fundamentally interested in implicit coordination and networking. And so the three of us together were just this team where we constantly went through different ideas and it was a terrific time.

So the paper came out in Feb 2014 and we really started the project four years earlier. It was four years marathon getting to that point. It was definitely not something that emerged overnight, and there were points in the middle, the two year point where I think all of us just thought we were doomed. We were just going to stop at the halfway point, declare victory and say, "Okay, we learned but this is harder than what we thought, and this is it."

[36:25]

### **Success Can Be Learning from Failure**

I think that's also important, that you have successes and failures. Even our final paper is both to me a success and a failure. We thought we would be able to do a much bigger system and we thought we would be able to do it for longer, but we still have a lot to learn about how to make these robots robust. So, for example, there are lots of error correction that happens in the system, and we thought about low-level error correction and high-level error correction. But if a robot actually fails on the structure, you need a tow truck.

It's interesting... you look at termites and they're able to drag the dead termites out of the way, they just build over them. They have a lot of ways in which they can compensate for the world. We built these super specialized robots, thinking about how are we going to have robotic systems where many robots work together where some of them are going to fail? How do we deal with those failures? Which failures require creating essentially an immune system where basically there's a second layer of robots looking for failures and clearing them up? I think all of us are really excited about thinking about this, because I think up till now we just gave up on that part. Like, "Okay, that's it, we can't solve this," or "there will be another robot that will come and take it away," and, "that sounds like a horrible and difficult one, but we're just going to ignore it."

But I actually think that in that process was we were learning how to build more robust robots. I think that was one lesson. The second is, the more robots you have the more failures you will have, even if they're low probability, so how do you deal with rare failures, and whether you need an immune system.

I think the other thing we learned is that humans tend to build in very rigid ways. We build out of bricks and we build out of two by fours. Insects build out of mud and sticks and much more compliant structures. As a result, when they're in an environment that doesn't fit their model, they don't need to level the ground in order to get something done. Termites will build totally functioning mounds that lean against a tree, and, as far as we know, they're just taking advantage of the tree. They're like, "Oh, this can be part of the wall, we'll just put it right in this corner." Bees do that.

[38:40]

So if we wanted to have robots that were actually in a disaster area or building levees or put in an environment that's unfriendly, we'd really want a system that's more like the insects and less like people, where we need to cut everything to the specification, have a blueprint, clear the ground. Maybe robots that manipulate sandbags is better than ones that manipulate bricks or square objects, so manipulating softer materials might be really interesting.

I think all of the ways in which we didn't achieve our goals, although we were pretty excited with what we did achieve, are actually the starts of really new investigations. Like, well, why did this happen? Why didn't we think of this before? And if we did think this way, how is it going to change our whole thinking about collective construction or self-assembly? And I think that to me is what keeps me going.

[39:35]

### **Self-Assembling Construction Robots - Current Status, Next Steps**

#### **McNally:**

And what stage is that work in? How far have you gotten along producing hardware versus just the theoretical paper?

#### **Nagpal:**

The paper in Feb 2014 was really about having a combined hardware system. What took us the four years was building the full hardware system. We're not working on that particular version of the hardware system any more, but with a former post doc of mine, Neils Napp, who is now a professor himself, we actually started working on robots that deposit foam. We're trying to look at robots that would actually manipulate a material that's much softer and fuzzier.

We're also looking at robots that could build structures where you didn't know the end goal ahead of time. An example would be, let's say you put a robot in a situation and it needs to reach some light or it needs to reach some particular goal. But there's a bunch of rubble in front of it and it can't get over it. Can it build itself

out of the situation? You don't know exactly what you need to build, but you do know that you need to build some sort of ramp that's basically going to get you there. Obviously since you can't climb over these rocks, there are things that you can and cannot climb over.

So we looked at how we would algorithmically solve that, and I think we have a lot of really interesting answers. That project is still sort of halfway through, and I'm thinking about how that one is going to morph and I think Neils is also thinking about how it's going to change. Self-assembling in the face of real uncertainty is sort of the direction in which we're both trying to build hardware.

...also I think, where the mathematics gets much more complicated... We're not as good thinking about spatial problems when they're not in a nice gridded world or in a nice rectangular world. That's very comfortable for us in geometry, but thinking about arbitrary functions in the environment is a little bit harder. So there are a lot of technical reasons why we just as of yet don't have as good ideas about how to do it or even how the insects do it.

[41:39]

### **The Role of Biology**

This is also part of the reason the Namibia project is really interesting to me. We're continuing that project, and we're working very closely with biologists to understand how are the insects thinking about building when they don't have a very specific goal in mind, but they have a function in mind. They're building for shelter or they're building to get out of something or they're building to close something, and they're really just responding directly to the environment to take decisions. I think the more we understand techniques from insects, which are easier to study than say beavers – we have read papers on beavers as well – is that maybe we'll get more ideas and more ways of thinking about the hardware as well.

[42:19]

### **Real World Applications**

#### **McNally:**

We're going to switch now to the swarm, but If anyone's asking what's the practical value of this sort of thing... you referred to it a little bit - Could we build a system of robots that could deal with natural disaster areas or areas of danger to humans and they could go in and sandbag a levee or do something with something that's happened in an explosion, that sort of thing, right? That's sort of the picture of where that could go.

#### **Nagpal:**

Absolutely, that's part of it, but also if you think you want to build underwater, or...

#### **McNally:**

...in space...

**Nagpal:**

...in space, or you want to build in the desert. Where are the places where we could basically not send people, not have to worry about their lives, instead send a whole bunch of cheap robots and not worry about the cheap robots either. If you send a super expensive robot that's doing a terrible job, then you might as well send the people in, but if you're sending a lot of robots that are doing a reasonable job, then they're machines, they're not alive, and so that's a really big difference. I think there are many environments where robots can build small structures to help.

Also, just sort of an aside - we do need to get better at building robots that do more than observe. That's part of the goal. A robot that just walks around and sees things but can't take any action is still a really pale choice. So how do we get to that point? And I think construction is just one of those examples. What you really want to build is robots that manipulate.

[43:53]

**The Thousand Robot Swarm**

**McNally:**

So let's talk about the other big thing in the last year ...In August 2014 your group unveiled your thousand robot swarm, small simple robots collectively arrange themselves into shapes. Again, no central brain, like ants or termites. What is that story?

**Nagpal:**

Swarm robotics is actually a field that has a long history, where long is like a few decades. The idea is that, like we were talking about self-organization... The idea is that, rather than having the one perfect robot - which anyways is not necessarily very useful because it would be slow - having many, many, many simple robots, and using the swarm to achieve things in the way an ant colony achieves things.

If you think about a bee colony... An individual bee is pretty small and puny, and yet a colony of ten thousand honeybees can forage over areas 2-6 kilometers range. So imagine you have this little group of robots that could actually monitor and look through an environment that large, and you weren't worried that a few of them got stepped on or some of them fell out of the air or a few things happened, because you have ten thousand of them. So you're manufacturing more robots quickly. If you thought of your bee colony as a robot colony, it's an amazingly powerful system.

So the question is, how do we get there? The swarm robotics field has really been looking at, first of all, how would we design colonies of robots that would do something that was useful for us? Monitoring is a great example, or foraging, so going and finding things and bringing them back to a particular location. Environmental clean up is another example.

All of these are examples where having many simple robots could be extremely useful, but it's been very challenging to get there. I think in part because we don't

really understand what is a sufficient robot and also we don't understand what things we're going to face that are going to be different with a hundred robots than say ten robots or different about a thousand than a hundred robots.

So simultaneous to all this, there's been some really great projects and we've learned a lot of lessons from them. No way do I want to say that our project somehow was the first in that sense. But our goal was really to try and push it above. There were a lot of groups that got stuck at making a hundred robots and the question is, are a hundred robots like a thousand robots, and the answer is definitely no. I can conclusively say it's not true.

And so the question was, what's going to allow us to get to a thousand robots? ...where we really experience our robots the way one thinks about an ant colony, that there's so many of them that you're not going to name them, you're not going to touch individual ones, you're not going to remember the ID number. If one of them moves, you're going to lose track of it. That kind of number, how do we get there?

And so I worked together with a really talented person, Mike Rubenstein. Mike is actually starting a new faculty position at Northwestern. He will definitely continue some more crazy, crazy ideas, so his lab's going to be a very exciting place to watch.

Mike felt that, given the way technology was moving, we really were at the right point where we could imagine not getting stuck at a hundred, and thinking about what would a thousand robots look like and what things would a thousand robots not have, that we were giving to a hundred robots, but wouldn't be willing to do a thousand times. And he started with that mentality.

[47:14]

## **Kilobots**

### **Nagpal:**

So we created... It's called the Kilobot Robot, and the idea is that you would buy robots in the kilos, so that's the kilos, a thousand robots. You buy a kilo of robots the way you buy a kilo of bytes. And each robot is a robot. You can program it, it has a controller, you can tell it what to do, it can talk to its neighbors over a short wireless range. It can communicate very complicated things with its neighbors. It can give a whole program to its neighbors if it wants. It can move in a controlled way.

Basically we thought about how to make all of those aspects cheaper, what kinds of technology we could use, how to mass manufacture the robots, how to let them sleep in low power so you would never have to touch them, and all of these factors. And the result is, I have to say... This is one of the projects where I feel we achieved more than we had set out to, which when it happens it's such a fabulous feeling.

In the process of really trying to make a robot that we could design a thousand times without making people in the lab super unhappy, we created a robot that was really easy to manufacture. So midway through our project - thanks to the Wyss Institute - we were able to license our robot to a company, K-Team. They now produce this robot and at this point there are about ten other groups that have hundreds of robots that are the same robot as us, so now we have a community of people with these robots. These are not robots that go on the shelf, these are robots that are being programmed to study collective behavior in so many different ways that I had never imagined.

And with the Wyss Institute we've been able to create a software system and a programming environment, and now there are biologists using our robots, which I never imagined would be the case, but they're very excited about them. There's actually a group, Roderick Gross's group in the UK, that has a thousand robots of their own. So now there's another group that has a swarm as large as ours, and I never imagined that this would be possible.

[49:14]

That actually was one of the earlier and growing outcomes of this has been that creating these platforms, these robotic swarms means that more of us can experiment with them and more of us can quickly move the science forward. What is different about algorithms on a thousand robots? What are the kinds of failure modes or emergent patterns that are going to happen? You set off a chain reaction, what do those chain reactions look like? In a hundred robots - first of all people run very few experiments with a hundred robots. In twenty robots, you don't see chain reactions; you don't see rare events happen. In a thousand robots you see every rare event happen, because with a thousand robots for a long amount of time, you're just playing a probability game. When we started programming these robots, we learned very quickly how incorrect our assumptions were, our mathematical models were, and now we're starting to really understand more and more about how to better program these systems to be really resilient.

[50:15]

### **Swarms Forming Into Shapes**

#### **McNally:**

Just so people can get a picture...What you did when I first heard about this story on the news, what you did with those thousand robots...?

#### **Nagpal:**

We wanted to demonstrate that you could program a thousand robots which are all super unreliable, not precise, can only talk to their neighbors... You could program them to do something that was really specific and predictable, and much larger phenomenon than any individual robot could experience itself. So we chose self-assembly and shape formation because that's one that you can really show that somebody can pick an arbitrary thing and I have a compiler that will put the rules and the robots will actually form it.

So we chose particular shapes. We chose the starfish and, in particular, the starfish is actually a model, it's a dream for people who study self-assembly, because it's an organism that has incredible powers of regeneration, so if you broke part of it, it should repair.

We can't actually do that yet so we tried to at least create that shape; the letter K, because it's arbitrary; a wrench, thinking that in the future these would be programmable materials that would actually produce something physical at the end.

So those are all fairly arbitrary shapes, and what we wanted to show was that we could actually program a thousand robots to do something predictable, reliably, even though each individual had a very limited... Each individual robot can actually only see about twenty robots nearby it, and the whole system, you can imagine from end to end is many, many... the information from one end to another is really far.

**McNally:**

You can go the Wyss Institute website and see a video of this... These 1,024, I believe it was, little bitty robots, each the size of a quarter, each vibrating on the surface, it took them how long to form into the shape of a star?

**Nagpal:**

Long. About eleven hours.

**McNally:**

About eleven hours...But they did it.

**Nagpal:**

But they did it, and I think that we learned something very specific. First of all, eleven hours of running robots is not a common thing to do, so the way we made the system work is we realized that we had to create systems that would look for errors and find errors. As I was talking about, this robot immune system idea, there were so many ways in which one failure from one robot can actually cause a cascade, so we thought a lot about how do we have robots that monitor each other's behavior? Your neighbors can tell you, for example, I don't think you're quite behaving correctly, maybe you should reboot yourself, and a robot could respond by actually doing that. We had to program algorithms like that, to make the thousand robots work for eleven hours.

One of the reasons why they're slow is because the vibration motors are very cheap and really very reliable, in the sense that they work out of the box really well, but they are also very slow. So you're giving up speed for other benefits. In real robots, another reason why they're slow is that robots spend a lot of time taking decisions. So if I have to integrate a lot of sensory information before I can take an action, then by definition I'm wasting time in order to take an action.

So one of the things that we are trying to learn is how can we have the robots make simpler decisions or quicker decisions. We're thinking about different ways in which we can do the self-assembly so that the whole process dramatically reduces in size. I think we have a couple of ideas on how we can - in the same system with the same speed robots - actually make self-assembly dramatically faster by just thinking about the process differently.

[53:40]

### **What Big Question Are You Asking Now?**

#### **McNally:**

Let me just ask you one final question. What's the question or the vision that's turning you on the most right now?

#### **Nagpal:**

That's a tough one. Since I have a whole life, you can imagine I have many visions. I just find it amazing that there's so many systems where the collective has so much more power than the individual. I think human systems are like that too. And for me one of the visions is really understanding that process. What is it about groups that makes them really strong and really powerful, and, in some sense, what is it about groups that make them not that way?

I think there are a lot of people who would love to understand that, not just for ant colonies but also for human colonies. What makes a research team, an institute function in an amazing collective where all of us come together and the result is supremely more powerful than any of us could have produced? That kind of power would be excellent to understand.

#### **McNally:**

You've been listening to DISRUPTIVE: BIOINSPIRED ROBOTICS. I'm Terrence McNally and my guest today has been RADHIKA NAGPAL.

We invite you to listen to the other two segments of this episode – with ROBERT WOOD and CONOR WALSH. We'll talk with Walsh about his work with soft and wearable robots with practical applications from healthcare to the military. Wood's projects include tiny flying robo-bees and self-assembling robots inspired by origami and pop-up books.

You can find both those podcasts at iTunes or SoundCloud.

You can also find them at the Wyss site – Wyss is spelled “W-Y-S-S” and the site is [wyss.harvard.edu](http://wyss.harvard.edu) – [w-y-s-s-harvard.edu](http://w-y-s-s-harvard.edu) You can also learn more there about the innovative work of the Institute. There's an extensive library of articles and videos.

You can also sign up at Wyss, iTunes or SoundCloud to have DISRUPTIVE podcasts delivered to you monthly.

My thanks to Seth Kroll and Mary Tolikas of the Wyss Institute and to JC Swiatek in production, and to you, our listeners. This is our second episode of DISRUPTIVE

WYSS  INSTITUTE  
for Biologically Inspired Engineering

and if you like what you hear, let us know and feel free to share far and wide. I look forward to being with you again soon.