Artificial Intelligence and Augmented Intelligence for Automated Investigations for Scientific Discovery

AI3SD Interview with Dr Louise Dennis
27/11/2020
Online Interview

Michelle Pauli
Michelle Pauli Ltd

03/11/2021

Humans-of-AI3SD:Interview-13

AI3SD Interview with Dr Louise Dennis
Humans-of-AI3SD:Interview-13
03/11/2021
DOI: 10.5258/SOTON/P0043
Published by University of Southampton

Network: Artificial Intelligence and Augmented Intelligence for Automated Investigations for Scientific Discovery

This Network+ is EPSRC Funded under Grant No: EP/S000356/1

Principal Investigator: *Professor Jeremy Frey*
Co-Investigator: *Professor Mahesan Niranjan*
Network+ Coordinator: *Dr Samantha Kanza*

Contents

1 Interview Details

Title	AI3SD Interview with Dr Louise Dennis
Interviewer	MP: Michelle Pauli - MichellePauli Ltd
Interviewee	LD: - Unviersity of Manchester
Interview Location	Online Interview
Dates	27/11/2020

2 Biography

Figure 1: Dr Louise Dennis

Dr Louise Dennis: 'Computer science is a very young discipline, but we have managed to kill people with our systems'

Louise Dennis is a Senior Lecturer at the University of Manchester where she is part of the Autonomy and Verification group. She is a member of the IEEE Global Initiative for Ethical Considerations in the Design of Autonomous Systems and is currently co-investigator on two EPSRC Hubs for Robotics in Extreme and Challenging Environments: Future AI and Robotics for Space (FAIR-SPACE) and Robotics and AI for Nuclear (RAIN). Her research focuses on the development and verification of autonomous systems with interests in rational agent programming languages, and architectures for autonomous systems, with a particular emphasis on ethical machine reasoning and creating verifiable systems.

In this Humans of AI3SD interview she discusses the ethics of autonomous systems, the difficulty of grasping objects in space, and the value of being open about your unknowns.

3 Interview

MP: What's been your path to where you are today?

LD: I'm a Senior Lecturer at the University of Manchester in the Computer Science Department. My undergraduate degree was in mathematics and philosophy and from that I became interested in mathematical logic, which is at the intersection of those two disciplines. That got me into artificial intelligence and what we now call symbolic artificial intelligence. I did a Master's in knowledge-based systems, which was a kind of synonym for artificial intelligence at the time, with a focus on implementing logical reasoning type methods in computational systems. And then I did a PhD in automatic reasoning, a branch of artificial intelligence. It's about getting computers to prove mathematical theorems and the major application of that is in software verification, so verifying that computer programs do what they're supposed to do.

That got me into the area of program verification and now I combine program verification and artificial intelligence so I do verification of autonomous and artificial intelligence systems. I'm looking at systems that use artificial intelligence technologies, but I'm also using automated reasoning to prove things about those systems. Primarily, if you're looking at autonomous robotic systems you're talking about safety properties but there's a whole lot of other properties we've got quite interested in recently in this area of artificial intelligence and ethics. What are the ethical properties that we might want an artificial intelligence program to have and how can we specify those mathematically and then verify that the program actually adheres to those properties?

MP: What are you currently working on?

LD: I'm working on three grants at the moment. The Industrial Strategy Challenge Fund supports four hubs on robotics in extreme environments, such as nuclear, space and offshore technologies. I'm associated with RAIN, which is Robotics and Artificial Intelligence for Nuclear and FAIR-SPACE which is Future Artificial Intelligence and Robotics for Space.

The other grant I'm associated with is the verifiability node of UKRI's Trustworthy Autonomous Systems programme.

The idea behind the node is to look at issues related to how you verify autonomous systems programs. This is both programs that use the symbolic AI techniques, and programs using what most people associate with the term AI now, which is machine learning, deep neural network-based techniques. How can these be fitted into frameworks that allow you to be thinking about verification and formal specification for what you're doing all the way through the software development process? The goal is to make it not just something that happens at the end when you have a program, but that at every stage you're tracking what your requirements are and that everything is still meeting those requirements and you're doing this in a formal mathematical way and you can cope with the fact that you might be using quite different tools for different parts of the system and how you can fit all of that together. And what happens when you're not just looking at the system as a piece of software, but understanding that this also interacts with users and that has implications for things like safety and ethics as well? So, how do you actually kind of bring the users into your understanding of what you're validating about systems?

MP: What are the consequences of not verifying - why does this research matter?

LD: In the worst case, people die. Computer science is a very young discipline, but we have managed to kill people with our systems. There was a software control component in a medical imaging system that was in charge of determining x-ray doses and a bug in that meant that a number of people were exposed to high doses of radiation as a result, some of whom then developed cancer. There was an instance in the military where a missile targeting system went wrong and overshot and actually hit a US Army base rather than the target it was aimed at.

It brings in concepts of how much people tolerate risk. We know that people have quite a low tolerance of risk in aerospace so aerospace has used software verification techniques for a long time in its computational systems. It has rigorous processes for trying to ensure safety. Whereas, people have quite a high tolerance of risk with cars. When we move into autonomous systems, the aerospace industry is expecting to have to use very high degrees of assurance for its autonomous unmanned air vehicles but the automotive industry is not even convinced it needs these kinds of techniques. So it's not as straightforward as saying, people will die, therefore, we must do this. You have to talk about how much society is prepared to accept the risks that there will be some deaths compared to what they perceive as the benefits they're getting from this technology and, therefore, how embedded these sorts of techniques will become in your development processes for these technologies.

MP: Do we expect higher standards of autonomous vehicles in that sense than we do of humans?

LD: When you say, "Well, I've got two human beings on the road in front of me, how am I going to decide which I swerve to avoid," people obviously don't think long and hard in that situation, but as a programmer we have the option to decide in advance – or at least decide how the decision should be made. But do we even want to have some kind of algorithm making that decision? You can't start saying, "Is this person worth more than this person?" Should the car even have this information? Do we really want cars making judgments on our value as a human being based on our features – should the car be able to look up your criminal record to decide whether it chooses to run you over ahead of someone else? We feel the obvious answer there is no but there's a question of where do we start drawing the line? Lots of people, for instance, feel children should take precedence over adults.

There has to be a conversation between manufacturers and regulators and people who develop assurance technologies to decide where and how and whether we want assurance to work with these systems. Some of that's a political decision. Whether people actually decide that the cost of implementing those high standards is worth it needs to happen at the political level. If you're in a situation where a crash is inevitable or there's only bad choices, which of the bad choices should the system take? And again, that gets us into the ethics realm. And you have to have some way of letting people actually see what ethics have been put in the system and, at some level, sign it off, whether that's government, or user groups, or individuals in their home, depending on the system you're talking about. And then, A, you have to get into the question of how you engineer the system, and, B, you have a user interface problem because nobody wants to answer a 40-page questionnaire before they get in their car. How do you enable users to specify their choices without ending up in the internet cookies situation?

MP: And what about space?

LD: The FAIR-Space project is about robotics in general and one use case is tackling the big issue of space debris and the challenge of getting a spacecraft with some kind of arm to grasp some other object and do something with it. The further into orbit you go, the more difficult it becomes for a human operator to control these things. Autonomous grasping could help. But, in space, the physics is quite counterintuitive in lots of ways because, on the earth, you're fixed in position. So if you grasp something and pull, you don't move. Whereas, if you're in space and you grasp something and then you start pulling, you both move. So it's a different problem and one that's hard to test. You have to do a lot of simulation before you spend millions of pounds putting a robotic arm in space and trying to grasp something. The project is also looking at how to give vehicles such as Mars Rovers more autonomy – can we improve how they detect the environment around them and how they react to changes in the surface they're moving on? Finally, the project is researching how to give astronaut spacesuits additional functionality through robotics.

MP: What's surprised you with these projects?

LD: It's less a surprise than a conclusion I feel I'm reaching for but I've not quite got there yet - leading to more questions. It's to do with the role of verification and assurance. If you're dealing with something that exists in the real world, then, unless you're God and have a kind of complete description of how the real world works, which we don't, you can't offer guarantees. When you're doing verification, you're always working on some simplification of how the world works. So, even if you've done a verification, it still could be wrong. What sort of guarantees can you get out of verification that is beyond what you get from simply building your code in a responsible way? How is the verification illuminating the practice of the underlying system or helping you find errors in the underlying system? Is there some way that we can make verification more deeply a part of the programmers' process so that all the way through it is helping you to produce stuff that's correct? How do we embed what we do into these artifacts and processes in a way that we don't seem like hostile outsiders who are there to say you can't do things?

MP: What advice would you give to other early career researchers based on what you've learned or discovered?

LD: It's important to look for what people don't know. Scientists are often quite bad at saying, "And the things we don't know are..." We're encouraged to write papers which say, "This is what we've done," not, "This is what we've not done," or "This is what's left to be done." What you really need to be doing as an independent researcher is figuring out what's not known. Seeing that can be quite difficult so it's worth asking "Oh, what is it that you don't know? What are the questions here, what is it that's not working because..." When you publish, try to be open about what your unknowns are, what the things you don't think you've quite got right yet are. In the intersection between science and engineering, part of what you're doing is trying to work out how the world works and build things that do things. Reading other people's papers, ask yourself, "How much testing do I actually think they've done? Do I think this will work on larger datasets or more complicated problems. Can I work that out?"

It's easy to be discouraged and think that people have solved all the problems in one area but they might not have done. They might have solved a very simple version of the problem and that's what they published but the system won't work on the more complicated problem. Look for the gaps, and be aware that people aren't dishonest about the gaps but they don't publicise them either so you have to read between the lines to figure out where they are.