SDS PODCAST EPISODE 963: REINFORCEMENT LEARNING FOR AGENTS, WITH AMAZON AGI LABS' ANTJE BARTH

Jon Krohn:	00:00	An AI agent that's reliable 60% of the time for nearly all real world use cases is 0% useful. Welcome to the Super Data Science Podcast. I'm your host, Jon Krohn. In today's outstanding episode, I'm joined by Antje Barth, a multi-time bestselling O'Reilly author, an instructor on Gen AI with over 400,000 students and a member of the technical staff in Amazon's prestigious A GI Labs, where they're focused on building reliable AI agents that feel like a digital coworker, not just a tool. Learn how in today's great episode, this episode of Super Data Science is made possible by Dell Intel Fabi and Cisco ,Antje welcome to the SuperDataScience Podcast. It's a treat to have you on. Where are you calling in from today?
Antje Barth:	00:46	Thanks so much for having me. I am calling in from the Amazon a GI lab in San Francisco.
Jon Krohn:	00:51	Very, very nice. Amazon, A GI Labs is the focus of our episode and all the cool things that you guys are doing there. Our research is fascinating on what you're up to at the Amazon A GI labs. I can't wait to dig into it over the course of this episode.
Antje Barth:	01:05	Let's do it.
Jon Krohn:	01:06	Yeah. So you're a member of the technical staff there at Amazon, A GI Labs, which is a very cool role. You're also a multi-time O'Reilly book author. That's not really going to be the focus of this episode, but the point is you've got deep technical chops, you're a really well-known individual. I guess maybe from those kinds of things like making those O'Reilly books, you're also a developer relations lead at Amazon, A GI labs, right?
Antje Barth:	01:27	Right, yes.
Jon Krohn:	01:29	Nice. And so from there you drive the vision strategy execution for how developers engage with Amazon's next

generation AI products. And the latest exciting AI product you're promoting is something called NOVA Act. So tell us about NOVA Act.

Antje Barth: 01:44 Yes, so Nova Act is a service that we just recently launched. We had a research preview going on since March last year and are super excited that this past December at our AWS Reinvent Conference, we launched this as a GI Service. NOVA Act helps you to build UI automation tasks at scale very reliably and helps you to really kind of start prototyping and putting it in production really fast so you can get started, which I love as a developer really fast in a playground experience and then iterate on it, debug it, and once you're ready, push it onto the AWS side and run it there reliably and safely at scale.

Jon Krohn: 02:29 Really cool. And it's free to get started, right?

Antje Barth: 02:31 Absolutely. So one of the key things, again, I'm excited as a developer, is we want to make it really easy for you out there to get started, right? In this industry, we know the speed to delivery. Speed to shipping is kind of the mode. So you don't want to spend too much time spending up the right environments and infrastructure and integrations. You really just want to validate your ideas. A lot of startups, you just want to go build the ideas you have, validate them really quick, iterate on them so you can get started really quick in our playground experience to especially do that. And then you iterate there and then you can move into the next step. For example, if you want to customize more in IDE environments. So we really want to keep also the surface area really where you're doing your day-to-day jobs as a developer,

Jon Krohn: 03:20 Really interesting. Would you be able to walk us through the typical user journey through a NOVA Act experience? Obviously I'll have a link to Nova Act in the show notes for

this episode so people can go there and describe to them what it would be like to us to experience it as we go there for the first time and we're playing around in the playground all the way through to deploying.

Antje Barth: 03:42 Absolutely. So you can go to nova.amazon.com/act and then there's the playground experience. There's also links to all the dev tooling we are offering ID extensions, SDK downloads, but really kind the first step is you go into the playground free of charge. You don't need any AWS account or anything. So you just go in there, you provide a website. For example, you're going to, let's say a booking website or a specific event, maybe sign up site conferences on the starting soon here in the Bay Area and everywhere. So you just put in the website and then in natural language you can decide and put in the actions to take, let's say I want to sign up to an upcoming meetup. So maybe I put in the Luma website and I going to say, Hey, search for a specific meetup. Maybe I want to join an AI performance meetup.

04:36 I look for that. And then you can also put the actions in to fill out, click the RSVP, click the join it and have it fill out the form for you. And on the same playground you see an embedded ui, the browser environment. So you don't have to set that up any manual way so you can see it right there. So you can observe how act, how the agent is going to that website. It's performing your actions. You see also the reasoning traces. So what it is doing, which is exciting, especially important for developers to be able to debug, to troubleshoot, to really see what's happening there. And you can tweak it if it's not getting it at the first time. So you can optimize your instructions, see it. And then once this is performing well to what you want to achieve, you can then download the script on the background.

	05:27	It's writing a Python script that captures all of those steps. So you have the ready script, you are using natural language, it converts it to the code for you. And then we have IDE extensions for example, or just an SDK. You get that into your preferred ID of choice. You import that script and then you're basically back in your coding world as a developer. So you can customize it, you can tweak it in there. And also the IDE extension has this embedded life preview. So a lot of times when we're building those automation workflows, we have a separate window popping up that shows you the browser, and we got a lot of developer feedback that that's just a little bit too much. We want to stay in the flow when we're building something. So we included that in your IDE, so you can really stay in there, have a unified window with all the troubleshooting, the debugging, the traces, so you can keep really close eye develop customize, and from the IDE, you can then if you want to also connect to your AWS account if you have one and for example, deployed there to run it in production.
Jon Krohn:	06:30	So that sounds like a really easy way for somebody to be experimenting with developing an agent because you can go to nova.amazon.com/act and then be able to use your natural language to describe what you'd like your agent to do, watch it, do that task and then get the Python code, use it in whatever environment you're comfortable writing your code in. And then that allows you to easily scale up whatever you're doing. But it sounded like you mentioned also being able to productionize on AWS with this solution.
Antje Barth:	06:59	So a core kind of motivation for us, talking to a lot of customers, talking to developers out there, we've seen so many flashy demos. You go to any meetup, especially here in the Bay Area, but in other parts of the world in a similar way, and everyone has a flashy demo and all the kind of fun stuff that agents can do. But what we

observed and the feedback we've received as well from customers is that on average, those agents work maybe 60% of the time. And to be honest, an agent that is reliable, 60% is 0% useful, especially if you want to productionize this, you need to really have reliability that you can trust those workflows and that agent to complete that one task. So this was the core motivation for us, really the P zero to make sure NOVA Act can reliably 90% and more deliver on those workflow executions.

Jon Krohn: 07:55 Awesome. So how do you do that? How do you get that kind of confidence? How do you go from a 60% reliability to better than that? What kinds of tooling do you have in NOVA Act to ensure that?

Antje Barth: 08:07 Yeah, so I want to talk a little bit how we train overact, which is exciting, at least I find it very exciting. So in the past when we trained AI models and things, it's a lot about imitation learning. You collect data and you show it that data. Now as we're moving from the chat-based conversational models that we all know and work really well into this space where we're building agents that need to take an action that doesn't work anymore, right? Because the agent is not just predicting the next word, next token anymore, the agent needs to predict the next action to take. So what we are doing is we're building out those reinforcement learning based web gyms as we're calling them. We have actually one in the playground, you can actually play and observe one. And those web gyms are replicas of very typical ui, so maybe a form filling ui, a shopping workflow, et cetera.

09:07 And we are letting the agent train in those web gyms. So imagine hundreds of those gyms like typical UI design elements there, typical tasks to do, and then give the agent thousands of tasks and they basically self play in there. So this is similar in the past, how AI learned to play chess, how it learned to play go, it's really kind of a

trial and error approach. So the agent goes in there, tries to do that form filling, it might fail a couple of times, but it self-corrects, it does it again, and it learns a better way to achieve the task. So this way the agent understands to reason through this UI and understands how to accomplish the task and helps it also to generalize well, right? Because UI change. So that's super important. If you're building that agent, yes, you have a specific site, maybe you're in those gyms, you're training it on, but then again you wanted to generalize if the website changes, if the checkout button moves, if the sign-in button moves somewhere else, if it's using different icons, because a lot of UI are really designed for us humans to navigate.

10:15 And for us, this is a simple task. If you think about maybe you go and write an email and depending on which email program you're using, which application, sometimes the button says draft or new create or it's just a little pen icon to create a new email. For us humans, we learn that. We understand that it's not a hard task for us, but if you're sending an agent to that environment, to that ui, the agent needs to be able to generalize and understand and reason in a similar way, even if some buttons and some tool says draft, the other one says create. So this is really exciting. So you're training the models and nobody act is trained like this on those web gyms to then really kind of be able to generalize and navigate those webpage even in real life as they're changing the structure and then they look

Jon Krohn: 11:04 Cool. So a big problem that occurs to me that I think a lot of people have with using agents and maybe the kind of training that you've been describing, it helps increase the reliability in a lot of common scenarios. But how can we evaluate that to be sure? Is there anything within NOVA Act that allows us to evaluate the performance of whatever ag agentic workflow we're automating?

Antje Barth:	11:26	Absolutely. So obviously you can check the leaderboards. We've done the public benchmarks on work arena, on real bench. That is important to just be able to understand the capabilities of the solutions. But even more important and what we focus a lot on is really working closely with our customers because we want to deliver agents that work on real use cases with real company tasks and for real users. So we're not focusing on this kind of the flashy demo side. We're really kind of working closely in collaboration with the companies, with the customers we have and then making sure the evaluations are running on their specific tasks. And this is where we achieve those over 90%, really kind of on those early customer enterprise workflows where we're really kind of focusing down what needs to be done and evaluating and making sure we are really delivering this reliability there.
Jon Krohn:	12:21	Love it. Tell us a bit more about work arena and real bench specifically. I have heard of those benchmarks, but I don't know too much about them. Myself and I bet they're new to a lot of our listeners,
Antje Barth:	12:31	So you can check it out. The real bench for example, super exciting. They have in a similar way how those agents work. They have those web gyms, those replicas of websites. So you basically can submit your agent there and then it's navigating a similar environment where there's a bunch of tasks across a different set of those gym environments to solve. And this is kind of this new benchmark, especially for those UI automation agents to see how they perform on those unseen new environments.
Jon Krohn:	13:01	We'll have a link to both of those in the show notes for sure. So a related question to the evals is there's lots of other things that you need to worry about when an agent goes from being in a playground to going to production,

security governance, access controls, change management, how do you handle those kinds of things,

Antje Barth: 13:17 Right? And this is where we really leverage the AWS integration, right, running as an AWS service. So we fully use, we're building on the foundation security that AWS delivers all the security, all the data is running in your account, so you have the full control. And we're making sure that is really running in a very reliable, in a safe and controlled environment that you as a customer can control. So all the goodness that AWS delivers in terms of reliability and security, that comes as well. And the important thing is really have this seamless journey. So yes, there is a playground that helps you to get started and evaluate ideas quickly, but we also wanted to make sure, because we know as a developer, you need to be able to debug, you need to have traces, you need to see what is happening, if any of those steps fail, why did they fail across this journey, across those different surfaces, whether you on the playground, whether you are in the IDE, and then obviously when you're running it on AWS, you have that observability, you have those traces, it's really exciting.

14:28 So for example, once you deploy this workflow onto AWS, and even if you're running it in the IDE, we are creating very detailed step views. So you see for every run you're making, so basically you're doing an act call, what we're calling it. So you're triggering this workflow, and then within that workflow there's a couple of steps, right? Look for the search button, insert this, and then grab maybe, I don't know, the cheapest coffee machine I can find somewhere, whatever your task might be or fill out this form. And for every step you see the reasoning traces so you can understand why the model took a specific action. You see also the next action prediction. Let's for example, click on those coordinates, put this in. And this is super important because you can really debug and troubleshoot

as you're developing how the agent behaves, where it might get tripped over something and then correct that. And then as you're running that in production on AWS, you have the similar use. So you can, for example, run the same workflow if you're thinking like enterprise cases form filling that needs to be done maybe a thousand times in parallel, you can schedule those runs and for each run you have the detailed view. So in any way, you have the audibility, you can look into the locks there, integration with CloudWatch and all the other tooling to really kind give you the full control and also the visibility into what's happening.

Jon Krohn: 15:51 I like that. And now I'm starting to understand the model here. I think as well, the business model from Amazon's perspective, which is a lot of organizations will have open source initiatives, but even when an enterprise has an open source initiative, they're looking for some kind of goodwill or some kind of benefit downstream. And maybe that's something is that when people want to scale up their agents on AWS Nova Act is free, gets them going, but then when it's time to scale up, it's convenient that you have AWS available to do that if you so choose to use AWS to do that scaling up.

Antje Barth: 16:30 And I would add to that, it's also giving customers, giving developers choice. Some developers really enjoy building it themselves and using the open source tooling and just do a very customized approach. But I think as you're scaling this out, you really kind of want to consider how much of this work do I want to do myself? And that might be very well very important cases where you want to have that customization ability and that has always as well and a W S's approach in this given customers choice in the building blocks, whether they're using open source, whether they're using more individual services and capabilities. For example, on the AWS side, there is this trans agents SDK that you can use to build agents and

you can use any kind you of tooling of models there gives you full flexibility. But then again, there might be customers that say, well, I don't want to go this route.

17:25 I don't want to spend that much time customizing. I'm really looking for a solution that has the best parts integrated and I don't have to worry about. Sometimes we refer to this as kind of a Frankenstein AI agent where you grab a model, you have to develop, look for the orchestrator, and then with the UI actions an actuator that actually taking those actions in the browser, in the ui, and you have to bring those together. Our approach to the Amazon AGI I lab is to train this all together. So basically we're training the brain and the body in one to give really kind of optimize it and give you that reliability. So again, there's plenty of solutions you can achieve this with NOVA Act. We're giving you this fully integrated way to hopefully take a lot of this heavy undifferentiated building away and give you a faster time to value with this.

Jon Krohn: 18:22 Yeah, there's a lot of thought to be put into what's the right balance to allow people to be doing things independently or using pre-made tools between customizability versus speed and reliability. And we dug up something, a project that you've done where you used, and you can correct me if I'm wrong on any of this, but our research indicates that you used Amazon Q and VS code to develop an app called Summarize Me. And that Summarize Me app produces an NCHE avatar animated video summaries of your meetings. And so you argue that developer environment agents act like a personal tutor sitting next to you while you work and that they help you, they help developers stay in flow, which is always a nice place to be. So drawing on your experience building that Summarize Me app as well as all of your NOVA Act experiences with Amazon customers, given the subjective

nature of developer flow, what do you think is the ideal balance of agency and oversight?

Antje Barth: 19:20 Yes, I want to also bring in another example here in a bit. But yes, correctly, I built this thing as kind of, I worked as a developer advocate for many years and with ai, many of us probably we're thinking what tasks can I actually automate? Every one of us has those really kind of boring, tedious tasks we have to do during the week. It might be filling out an expense report at the end of the week because we were traveling somewhere for business and we're really dreading this exercise, but we have to do it or something similar or researching the web for AI news that's coming out. I don't know, we cannot even keep track of what's happening throughout the day. So I think a lot of us have those little site projects where we're trying to automate some tedious parts of our daily jobs.

20:08 And this was also kind of a motivation for me when I put this project together just out of fun, really can I actually using some avatar technology and meeting summarizations and stuff to kind of automate parts of my job where I'm like, I'm in so many meetings throughout the day, but then again with AI summaries, you're getting then now a million of those AI summaries. It's also sometimes too much. So I was trying to think how can I streamline all of this? And to your point about the agency and the control of things, I have another example. If you're looking at AI assisted coding, for example, I think many of us are using those tools now. And if I just look back a year ago when I kind of worked with the tools or even one and a half years ago, we started out very early in the days with tap completion.

21:03 So we were writing the code pretty much, yeah, we did a tap complete. Then in the next phase with those AI coding agents assistance becoming more powerful, we then had them to maybe scan our repo and then also co-develop

that. We're saying, Hey, can you create this capability for me within this larger project, within this larger application? But for example, I still kind of reviewed the code once I got it back, I was like, okay, what did you do there? So I went line by line through that. And to be honest, fast forward 2026, probably most of us are using some sort of AI tool. I cannot even keep track, right? There is so many software agents that are helping you build your applications, build the projects. My job shifted to less reviewing, but really kind of just the supervisor. So I cannot even keep track of reading every line.

21:58 So to your point with the agency and the control, I think it's really about building trust with a tool. And I think we went through that period over the last one and a half, two years where we started to work with AI assistant coding, and now we have this trust, but with trust also comes we need to verify. So a lot of focus, what I'm doing is when I'm coding like this, I want to make sure that I have really solid unit tests and verifications in place. But as we shifted it from this vibe coding to more specter and development, we are still interacting with AI in the same way we're telling it what to do mostly in natural language. We're writing some specs now, and then we have the testing framework. And with this all together, we have developed a decent amount of trust that I think we are giving more and more agency. So this is an example from the coding world, but I think a similar pattern will arise with those UI automations right now, of course we want to have for specific things, a human in the loop that may be approved, that checkout workflow or just takes care of some of the important decisions that are part of this task. But ultimately as we're building up the trust and we have specific verifications in place, I think this agency will slide a little bit too, to give it more.

Jon Krohn: 23:22 I think that makes perfect sense. I do think that is where we're headed. You mentioned at the beginning of your

most recent response that you had another example other than summarize me and now would be a good time to go into that.

Antje Barth: 23:34 Yeah, that's kind of how I code. This is what I want to get in. It's shifting how we're still using natural language talking to all those tools, but we're giving it a little bit more agency.

Jon Krohn: 23:45 Right, right. Gotcha, gotcha, gotcha. So we have a quote from you from a blog. We'll have a link to that in the show notes where you point to over a thousand generative AI applications already built or in development at Amazon. So from those, do you have some kinds of takeaways for our listeners from those from a thousand plus internal deployments of gen AI where agents can actually deliver value and where they still struggle?

Antje Barth: 24:12 Yes. So it's really exciting being part of Amazon also, we are working and collaborating with a lot of the internal teams here and seeing the different use cases coming together. And one of the very exciting ones is from Amazon, Leo, you might know Leo previously they were known as Project Cooper. They're working on those low orbit satellite networks to bring connectivity into the underserved regions. So very exciting work that they're doing. And they've been looking at NOVA Act as well how they can integrate it. And they built some amazing automations that help them do testing QA frameworks across different from the web to Android s apps as well. And then also build additional tooling on top of that where for example, they can now product managers in the team can use Novak to from a FMA design, from a wireframe design, create test cases even before those get implemented. So there's a lot of exciting work that's happening. And yeah, I'm also curious out there a lot of you startups, developers, data scientists that are listening how you think UI automation can come together and

which processes you can automate. But yeah, certainly there I think so much opportunity space and we're really, really just at the very beginning of what's possible

Jon Krohn: 25:43 For sure. Are there some kinds of general things you've seen that work or don't work? Maybe the UI example is a really good place to go because it seems like NOVA Act specifically is designed for browser-based tasks.

Antje Barth: 25:55 We started with a browser because it's the easiest way to give us the closest to this universal action space. If you look at how we humans work, a lot of our work is actually happening in the browser where we're working across different applications. We have the different tabs open. So this gives us a very great first point to get to this universal space. But we're also looking beyond the browser. So again, we have great research teams here. I really enjoy talking to a lot of our cognitive scientists who are also looking into those problems space. And one of the main use cases really is to your point, is the QA testing. We see a lot of things happening there for a couple of various reasons I think, but really it helps you really quickly to automate some of those. As you develop new applications, you're pushing out an update and you want to make sure that the experience delivers.

26:52 For example, one of our customers PGA tour, they're using NOVA Act to make sure the website correctly works, especially before any golf tournaments where there's a lot of excitement and people coming to the website. So they're making sure that, for example, the weather information wit everything works, all the buttons, everything in there. So a lot of this QA testing on websites, but also, for example, another one herz uses NOVA Act. They're doing testing, automated testing for a lot of their core rental workflows, which help them to just increase the shipping velocity five x. So it's really kind of, and sometimes internally we use this playful term that

we're calling those norm core agents, which really automates some of the most more boring stuff. It might not be those fleshy cool things that we sometimes come up with at hackathons or stuff. And really kind what are kind of the core workflows that seem a little bit more boring, seem a little bit more tedious, but you can actually automate pretty well. And we see a lot of interest as a first use case in this automated QA testing, but we can imagine a ton of more use cases out there.

Jon Krohn: 28:07 I was going to eventually ask you about this core stuff because we did have some research on it. What does it mean, what does it mean core? Where does that come from?

Antje Barth: 28:16 So this is, I think it was coined in the fashion industry, kind of core looks, which is a little bit more a simple white and jeans something. And normcore for us is really kind of those agents focus on more kind the traditional, the boring use cases you would call them maybe sometimes exactly that form that I need to fill out a thousand times that application that I need to do. Maybe you need to apply for something, you need to navigate websites to get a new driver license renewal or whatever it might be. So you have those tasks that you're dreading as I mentioned before, or that there's just some work that you need to do. And if we can automate those use cases for you, those tasks in a very reliable way, I think that would be amazing to just give you all that time back to focus on the more higher level, a more creative task we're all excited about.

Jon Krohn: 29:14 Alright, core. Yeah, I was thinking maybe it's kind of like the opposite of hardcore and when you were talking about core, you were talking about QA engineering, and something that's interesting with this shift to a agentic is that traditionally QA engineers were focused on finding bugs, really, but now I feel like a lot of it is probably

related to auditing reasoning traces that LM spit out. I dunno if you have any thoughts on that. This is kind of a tangential question.

Antje Barth: 29:41 I think there's a lot of things that will change how we're doing our jobs. We learned this in the software engineering world that they were more talking to AI than we're actually writing the lines of code sometimes. And yeah, I think similar things for QA engineers that they don't have to write those brittle scripts anymore. If you're looking like how you achieve those tasks in the past you had to write those very kind of rule-based scripts like selenium scripts or something. If you wanted to automate those processes, RPA where you said, okay, here's the button, click that, do that here and here, and whenever some element changed, you had to rewrite those. So hopefully we can again simplify this for those QA engineers and you can use much more QA through natural language creating those scripts for you. And then also with those agents, they're much more resilient to adapt to those changes. So you don't have to spend that much time in keeping those scripts up to date. And hopefully QA engineers can focus on the real things. And that might include yes, looking at different type of traces, for example, reasoning traces, et cetera. But hopefully this gives them also a little bit more focus on what's exciting in their job, figuring out what's going on rather than just spending hours and hours and hours in writing those scripts and updating them.

Jon Krohn: 31:05 That makes a lot of sense. Going back a little bit, going back a few minutes at least, you're giving us an example, a real world example of how Amazon NOVA Act can be useful for the PGA tour. And we dug up another example here, which I thought was kind of interesting, which is with one password. And so one password highlights NOVA Act handling complex sign-in patterns, which is a high stake surface area, especially for an application that

you're trusting with all of your passwords. So how has NOVA Act dealt with secure interaction with authentication flows? So agents don't become the weakest link in identity and access,

Antje Barth: 31:41 Right? And security is really kind of one of them, if not the most critical part as well, right? Reliability, security to be able to build a trust. So for Nova Act, we're really kind of building on the security foundations of AWS. I briefly mentioned this before. So you can be assured that all of those executions are running in your AWS account, they are protected, you have full control over them. And also like we give you back control. So for example, if you are encountering a password that you need to fill in, you can give back to the user. So basically the user would then be in control and you would, for example, with a playwright integration, you can then prompt locally in the terminal to hey, get the password from the user. It's never being sent over the network, but then the agent can fill it in.

32:31 So a lot of thinking really goes around that. And then one password, obviously given they are the password manager, one of the leading ones. So they build their framework around that as well to make sure, for example, the password manager can intelligently navigate the site and then they have their part how they deal in a very safe and secure way with the passwords. But that is definitely an area where you want to make sure you're not just sending anything over network, et cetera, and really want to make sure you have a very trusted and secure way to deal with those in a similar way with captures less sensitive maybe. But also this is another case where we would give a user the control back to solve those and we will probably see a lot of development in that area as well, what's happening, but very critical and very important to make sure you're not sending any sensitive data.

Jon Krohn:	33:24	Definitely another critical area with agents. We did talk about this earlier when you were talking about the reinforcement learning gyms that teach agents how to behave, but related to that, something that, and it also ties into the QA engineering thing we were talking about more recently, which is that even though QA engineering is easier in many ways than before, thanks to agents and to LLMs, one of the things that's tough and potentially becoming tougher is evaluating agents because they're so stochastic, you don't, in a traditional QA role, you could say the results should be this as a result of this test. And when we run an automation, this thing should output as the results. But with agents backed by lms, you're getting maybe slightly different responses can make that part of the QA engineering job trickier, I'd think. So how do you choose the right success criteria so that the agent doesn't learn shortcuts that pass the test but aren't really doing what you intended
Antje Barth:	34:26	For that? It's really important on a couple of different layers I think, to tackle this. So for one, you want to make sure that the agents can generalize well. So even if things are changing, they're not tripping over. So as you're building out those web gyms, you want to make sure they're really high fidelity gyms and the agent learns a lot of things. Like for example, it sounds simple, but one webpage is light mode. Either one is in dark mode that is already kind of a totally, even if it's the same UI environment, it's a totally different experience maybe for an agent. So there's a ton of variations that you can, even with the same task ui, just mix it up a little to make sure the agents can handle all of those. And then also to your point, how do I evaluate the success?
	35:14	You really want to focus on the end result, right? Because there might be a couple of different ways to achieve it and you can look into it, maybe one took a little bit longer than the other one, but ultimately what you're interested

in is a, it's a successful task completion. And then there's a couple of ways to verify, right? Let's say you have an agent and you're asking it, Hey, I want to schedule a one-on-one with my colleague and the agent needs to figure out which calendar is to open, where to look like for free spots. And then the end state would be okay, you can confirm that the meeting is actually scheduled in that calendar, so it's going to be somewhere in a database entry if you're doing shopping, checkout workflow or something and that sort. So you have a very verifiable end state to do that. It gets a little bit trickier when there's not an easy way to verify that, right? But right now a lot of those tasks, there isn't end state where you can check and then give the agent a little bit of freedom for the generalization part, which way they go to figure this out.

Jon Krohn: 36:21 I like that answer. It makes a lot of sense. Thank you for making something that seems so complex, easy to understand with a relatively simple solution actually. I appreciate that. So let's look ahead a bit now to what's next for agents as we kind of reach near the end of the episode. Let's talk about where this is going. Obviously it's a fast moving space, very difficult to predict what's going to happen, but in a recent podcast appearance, you describe AI evolving from a helper into a kind of peer in the coding experience. And we talked about that earlier in the episode together already in presentations you've expressed the idea that the atomic unit of all digital interactions will be an agent call and virtually every customer experience we know will be reinvented with ai. So how will this changing environment not only change how software is built, which we've talked a lot about in this episode, but also how AI engineers, data scientists, software developers, how they see themselves and how businesses value their efforts.

Antje Barth: 37:18 So to start, where I see the space evolving, which I'm really, really bullish about, is we want to build, especially

here at the MS NAG lab, we want to build useful ai, useful and practical ai. So this is really our core focus here. And what we think this might look like and will look like is that the agents become much more of a digital teammate, a digital coworker of yours, and we spend a lot of time here in our research teams to think about how do humans actually learn. So the future of agents as we see it, is definitely a multi-agent, a very collaborative environment interacting with other humans. Imagine you have this digital coworker of yours, how do they learn how to do the tasks, everyday tasks? And my colleague, cognitive scientist, Dr. Danielle Persik, she actually talks a lot about that in her podcast Making a Mind.

38:19 She really looks into those aspects, how humans learn things, how humans collaborate on a task together. And then we're working together on how can we translate that to building those agents and how can we teach those agents to learn a human, to collaborate, to ask for help to being humble. So this isn't super exciting space, and if you think about what this can bring us, right? It's like we might be able to just select the agents and say, Hey, can you help me with this task? And the agent maybe checks back a couple of times, but then also it learns from those interactions like humans would do and then can do those tasks more independent in the future. So I think a lot of things will go in this direction working with agents together to help us multiply our own productivity, which is exciting. So I think for everyone, whether you're a developer out there, whether you and I engineer, I'm super excited about this space. I think how we interact with technology will change and it has, looking in the past over all the decades and the technology innovations we always had adapt, we will still have continued to do that, but I think there's a lot of positive in there. Again, doing it obviously very responsible and safe and reliable, but getting to this stage where we can all use AI to then

augment our own capabilities. So I think this is what I'm excited about, at least for the future.

Jon Krohn: 39:47 Wow, it's a really cool mission. Is that kind of in the namesake of Amazon, a GI labs, obviously artificial general intelligence is this idea of an algorithm or system that can replicate all the aspects of human intelligence. So it kind of sounds like that aligns with this idea that you have someone that feels more like a coworker working alongside you that could be helping you out on any kind of task.

Antje Barth: 40:11 Yes, and again, the focus is really kind of what we call the useful ai. So we want to make sure everyone gets value out of working with ai, useful and practical for day-to-day tasks. And this will hopefully reinvent how we as knowledge workers work, how our work looks like, and hopefully loading off those really kind of tasks we dread throughout the week and we can focus on more the creative and higher level tasks of our jobs.

Jon Krohn: 40:43 Cool. And one last technical question I think here before I start getting to wrap up my questions, which is this is tying together a few different things that you've said in different places into one kind of forward looking question where on where you see the underlying models and the architectures going behind autonomous systems. For the last couple of years, two active areas of discussion have been about one, how much to rely on large language models, like really big ones versus smaller specialized lms. And then another big area of discussion two is how much to push the boundaries of model context or whether it even makes sense to have million token contexts considering retrieval systems and tools are continually improving to fill in the gaps. So you recently mentioned in a presentation that Alexa plus relies on hundreds of specialized expert systems orchestrated together and that an internal agent in AWS manages over

6,000 tools using retrieval instead of stuffing everything into a context window. So yeah, tell us a bit about that. How much context is too much? Where do you see that going and where do you see this idea of lots of specialized models versus big monolith LLMs with trillions of parameters that can do everything?

Antje Barth: 42:04 I think there will be a diverse environment where all of the different kinds will have a place if we're thinking like AI at the edge, that will most likely be kind of smaller models that just giving the constraints there. I think for a lot of tasks, the models will be the larger models that have the reasoning capabilities in all kinds of different shapes and sizes. And going back to this a thousand specialists and experts. So my thinking is you have your AI within maybe your company, but then your company does business with another company. So we saw this last year a lot of hype around the protocols, how can I connect my models to tools and then how can I connect one agent to the other agent inter and communication. So I think it's pretty much still an open area for research. What's the best approach here to build this?

42:55 And there are a lot of components I think still being built, but my understanding and my thinking of how the future looks there is we're basically entering a stage where we'll have what I call every single interaction will be an agent AI call. And then if you think about this, not just within your team or your company, how you do business with the other companies. So there will be a lot of, I think, level and layers up there where the agents communicate with each other and figuring out the right technology, the right protocols in that space. And that will be super exciting. So I think yeah, there will be a lot of place for all the different kinds of things we're developing, which is exciting and I'm really curious to see how the world looks like maybe even a couple of years from now.

Jon Krohn: 43:44 Great answer. It makes me feel silly about the question in the first place because it's kind of like, which way are we going? And you're like, obviously this is a complex ecosystem where it's going to involve lots of different kinds of solutions for different scenarios. Makes perfect sense. Well Ange, this has been an absolute treat for me. I'm sure it has been for our audience as well. You're an outstanding speaker on all these topics. It's such a joy to listen to you. And for all our video viewers, you'll be able to see, of course, that Antje was smiling this entire interview. That probably even comes through in the audio in the way it sounds. You just seem like such a happy person talking through all this stuff. So for people who want to be able to follow you after this episode and get more of your happy, insightful thoughts, where should they do that?

Antje Barth: 44:27 Follow me on LinkedIn. There's definitely a lot of content I'm sharing. Follow me on X and yeah, you will see me hopefully also in person if you're in the Bay Area. I'm planning on going to a lot of the conferences this year is popping into meetups. So yeah, hopefully we have a chance to meet in person and I'm super interested in learning what you all out there are building.

Jon Krohn: 44:49 Perfect. Yeah, so we'll have links to all your social media in the show notes. And then my very, very last question for you, which I actually usually ask second last, but for some reason it just felt right to ask you the followers one first. And so yes. So my final question today is do you have a book recommendation for us?

Antje Barth: 45:08 I do. So it's actually on my desk right now as well. So my former co-author from the O'Reilly books, Chris correctly just released a huge, I think it's a thousand pages book on AI systems performance engineering. And this is definitely an area as we're developing systems performance becomes so important and how to optimize

performance throughout the whole stack, really kind of from the GPU level to the application level. And that's on my reading list. I haven't managed a thousand pages yet, but hopefully over the next couple of months I will achieve that.

Jon Krohn: 45:42 Well, maybe we can have Chris Ragley on the air to discuss this. He's someone that's been on my radar for about a decade, but he's never been a guest on the show, so maybe we'll have to get Chris to talk about that new tome that he's put together. Cool. Thanks Antje. And hopefully we'll have you on air again soon. We'd love to check in and see how everything's going over there at HEI labs.

Antje Barth: 46:05 We'd love to. Yeah. Thanks so much for having me, Jon.

Jon Krohn: 46:11 Well that was a terrific episode with Anier Barr to in it Anier covered NOVA Act. Amazon's new service for building reliable AI agents available free to prototype now. Check it out. We've got a link for you in the show notes there. She went into detail on how Nova Act achieves over 90% reliability by training on reinforcement learning, web gyms where agents self play through thousands of relevant tasks, making them norm core agents that focus on boring but high value tasks like QA testing and form filling rather than flashy demos. She talked about how the future of agents involves multi-agent collaboration, where AI becomes a digital coworker that learns from interactions the way humans do and how Amazon has over a thousand gen AI applications built or in development internally providing real world validation for all of these approaches. Alright, as always, you can get all the show notes including the transcript for this episode, the video recording, any materials mentioned on the show, URLs for Ajay's social media profiles, as well as my own at superdatascience.com/963.

47:12 And yeah, thanks to everyone on the SuperDataScience podcast team, our podcast manager, Sonja Brajovic, media editor, Mario Pombo, our partnerships team Natalie Ziajski, our researcher, Serg Masís writer, Dr. Zara Karschay, and our founder Kirill Eremenko. Thanks to all of them for producing another great episode for us today for them to create this free podcast for you. We're deeply grateful to you and to our sponsors. You can support this show by checking out our sponsors links, which are in the show notes. And if you ever want to sponsor the show yourself, head to Jon c crone.com/podcast to find out how. Otherwise, support us by sharing, reviewing, subscribing, but most importantly, just keep on tuning in. I'm so grateful to have you listening and I hope I can continue to make episodes you love for years and years to come. Until next time, keep on rocking it out there and I'm looking forward to enjoying another round of the SuperDataScience Podcast with you very soon.