

Chicago IL
1:24:14
m4v

Russ Taylor

An interview conducted by
Peter Asaro

September 15 2014

Peter Asaro: So we usually start off by asking where you were born, where you grew up and went to school.

Russ Taylor: Okay, well, I was born in Hampton, Virginia, but I really grew up near Valley Forge, Pennsylvania. We moved there when I was six. And then I went to university at Johns Hopkins University in Baltimore. I was there from '66 to '70. And then I went to Stanford for my Ph.D. in computer science, and I was at Stanford, oh, from 1970 through 1976.

Peter Asaro: And what did you study as an undergraduate?

Russ Taylor: Well, it was a non-departmental engineering major. I was interested in computing and there wasn't really a computer science major at the time at Hopkins, so what I did it was a mixture really of the electrical engineering and operations research.

Peter Asaro: And what made you decide to go into computer science for graduate school?

Russ Taylor: Well, I just liked it. The summer after high school I'd taken a programming course at the University of Pennsylvania and I'd really enjoyed that, and I enjoyed programming. My undergraduate –

<break in recording>

Russ Taylor: – man named Manny Belmar, and I actually got a chance to – it was sort of a part-time job but also working writing mathematical optimization codes for Dr. Belmar, and I really enjoyed that quite a lot and so that made me interested. I had summer jobs as well at Burroughs Corporation for a couple of years, and then my family while I was in college moved to Bethesda, Maryland, and then I had a summer job at Bell Com, which was a subsidiary of Bell Labs, and I enjoyed that quite a lot, too, and so I sort of drifted into it.

Peter Asaro: And who did you work with at Stanford?

Russ Taylor: Okay. I was at the Stanford artificial intelligence lab essentially all the time I was there. My thesis advisor was a man named Jerry Feldman, and Jerry was quite interested in artificial intelligence. But I really interacted quite a lot with other faculty there. My other two readers of my thesis were Vint Cerf who was on the faculty at Stanford at the time, and Tom Binford as well was a machine vision person. But one of the nice things about Stanford AI lab is the faculty and the staff were really sort of thrown together, and so I also interacted a lot with John McCarthy who was the lab director, Les Earnest who was the executive director, oh, gee;

other faculty as well and obviously many grad students who were out there at the time, and we were all sort of working on robotics.

Peter Asaro: And what was John McCarthy like to work with?

Russ Taylor: Oh, he was great. I mean he – I never worked directly for him, but he was like a Tom Sawyer. He would often have these ideas. He was a true visionary and actually a great engineer as well as a scientist, and he'd get people kind of interested to do different things. And we also just interacted – you know, he'd not infrequently go out to dinner with the students and whoever was around on a dinner expedition, so that was what we did. Other people who were early days of robotics who were out there included, oh, gee, Victor Scheinman, Bernie Roth who I never really interacted with much. He stayed mostly on campus, but his students were up there and I overlapped with him. I believe Victor had been one of his students. Lou Paul, we overlapped. Lou was actually on my thesis committee. He graduated in '72. Oh, gee, other people, other students who were doing robotics, Ken Salisbury was one, Bruce Shimano, they were more in sort of the mechanical engineering and control aspects. Oh, gee, other students. Hans Moravec was quite active in sort of mobile robots and visions. Bruce Baumgart actually was doing similar things. Others go into my mind and out of it. There were a few. Bob Bolles was doing computer vision at the time, and there were a few others who I'm sure I will remember in the course of reminiscing with you.

Peter Asaro: <laughs> And so what was the first kind of robotics project that you worked on?

Russ Taylor: Well, I was interested – it was the development really of a programming environment for robots, and we developed a language. We had a research project that was funded by the NSF, research applied to national needs. There had earlier been DARPA – or actually at the time was called ARPA funding for robotics, but then the last couple of years I was a grad student there was this research applied to national needs program, and there our focus was on developing programming environments for robots and techniques for combining things like vision and various sorts of sensing with manipulation to do primarily robot assembly tasks we were looking at. And a couple of the other early days pioneers, from IBM there was Peter Will was on the advisory board for that grant, and partly because of Peter's involvement – eventually I went to work at IBM and Peter hired me, but there was also a researcher named David Grossman who came out on basically a sabbatical from IBM and spent a year while I was working on my thesis, and I interacted a fair amount with him as well.

Peter Asaro: And was this your thesis project?

Russ Taylor: Well, my thesis was related to that. We developed the language. It was called – as I said, originally it was called HAL but then some people, I believe it was from Draper Lab, objected that they had a language called HAL so we called it AL with an apostrophe, and then it was pointed out to us that that apostrophe would create a bibliographical nightmare, so we dropped it and we just called it AL. And so that was the language that we used. I used this sort of an object that I would generate these programs in in AL, but my thesis was called “The Synthesis of Manipulator Control Programs from Task Level Specifications,” and one of the things about a robot is a robot itself isn’t super accurate. It has much better precision, incremental accuracy than it does absolute accuracy in space. And so you can do more precise tasks than if you just simply program a robot to make the motions. And so the idea of my thesis was that could I automatically write those programs to incorporate both kinematic analysis of the task with you could think of it as tolerance propagation and have programs that would rely on sensing explicitly to do things. So I did various pieces of this as chapters in the thesis.

One was, for example, if I know a part is sitting on the table that constrains its configuration space to lie on a plane, and if you push it into a nest you get more constraints, and some of that information you may know a priori are the parts on the conveyor belt or whatever, and you can use those constraints to constrain the position of the robot. And also as you do assembly, once you’ve assembled two parts you know that they have a certain relationship. But then as well if you think about sensing error and robot execution error and alignment error these are result propagate, and so another thing that I did is I basically used fairly simple linearization to propagate these uncertainties through the task and through a kinematic model of the task, and then a program would do things like it would select an order for doing steps based on how you could best constrain uncertainties so you had the next step was more or less guaranteed to work and also would figure out where and when it needed to add some explicit sensing tasks. One of the driving applications was a little metal box assembly that I’d built a little box in actually Victor Scheinman’s machine shop and I was using that as a model. It was in a way a slightly simplified version of a Model T Ford water pump assembly that had been done earlier and was used as a driving example in Lou Paul’s thesis, and some of the assembly and manipulation primitives that Lou used I also used in this box assembly and would develop basically programs. One was sort of you’d do this little spiral. You’d put an aligning pin through a hole where you had a cover plate on the box and a hole, and so you put this aligning pin in there, and once you put two aligning pins you knew the cover was in the right place. But to get the pins in you could either use vision or you’d know where they’d go, or if you failed you could then do a little spiral search, and based on the uncertainties you could predict some limits on how long that would take, and so we were trying to sort of develop an efficient assembly program. So then what I had was there was sort of a set of template programs with sort of pieces that you could either edit in or out of that program based on this kinematic and precision analysis that you’d done, and so I would kind of edit them all in together, and that was basically my thesis. I do remember Vint Cerf who is not really a robotics guy, he was the internet guy, but was a very good reader on my thesis, and I remember being very relieved when he read one chapter and said, “That chapter was okay for a thesis,” <laughs> and then of course there was more, but that was a great relief so that

at least the least robotics person on the committee thought it was a good computer science <laughs> thesis. So that was at least part of what we did.

Peter Asaro: So did you actually have a robot then that could assemble this box?

Russ Taylor: Yeah, oh, yeah. I mean we actually – the programs that were created would do the assembly. Mostly we were using a generation of the Stanford electric arm that Vic Scheinman developed. There were two. There was a gold arm and a blue one, which was a little bit bigger and just slightly newer, but these had a couple rotary joints, a sliding joint and a wrist and a gripper, and the gripper had little contacts in them so you could – and there was a primitive, actually, that Lou developed and I used and later wrote some other ways of abstracting it called centering grasp, where you'd close the fingers and when this one got in contact you'd start moving together like that.

Peter Asaro: So what did you start working on after graduate school?

Russ Taylor: Okay, well, I got hired – well, I also had a summer in the army, which kind of slowed my thesis down, but I got hired into the robotics research group at IBM Research, which was managed by Peter Will, so Peter Will was the manager but we had, oh, just an amazing group. There was Peter. Dave Grossman was back. There was Larry Lieberman who was more a vision person. He'd been Regina Bajcsy's student. A little I believe after I joined there was a guy named Mark Lavin. There were a couple of extraordinary engineers. There was a guy named Andy Bannerman who was a control engineer and a guy named Hugo Panissidi, Pat Panissidi, who developed the hardware. Pat was an amazing engineer. He had like 50, 60 patents, I believe. The IBM robots that the group were developing were hydraulic because you could get a lot more sort of strength to weight ratio, and Pat had just a fiendishly clever hydraulic linear motor that was sort of a technical artifact. So we developed that. There were some early manufacture – and I was mostly working on programming paradigms and programming methods for these robots. There was a language that someone else had developed – I believe it was called Emily – that compiled into what you could think of as an assembly language called ML that moved these robots and did sensing and so forth. And the implementation of that was a mess, so the first thing I did is I redid a compiler for that language, and sort of roughly at the same time two other things were happening. There was a production problem up at the mainframe computers in Poughkeepsie where the machines at the time had circuit boards that were about that big and there were nine of them on a big frame, and there were several of these frames made up the computer. And the back panel wiring was – there were often wiring errors because it was a semi-automated process. And the way you would find these wiring errors is you would run diagnostics, and it was very, very tedious and laborious.

So what our boss, Peter Will, volunteered for and actually a number of us implemented was we built a giant robot that had two arms that would reach some feet into this frame and do basically electrical contact testing, but the frames were not all that precisely assembled, and these arms were a little bit unprecise, and so you had to use all of these calibration, registration and error correction techniques to reliably hit these pins on top of which, by the way, if you did the wrong thing it would damage this very expensive piece of equipment and the computer. So in the middle of winter I was – the hardware was built in the lab, but the programming, a lot of that we had to do up in Poughkeepsie, and so every morning at like eight in the morning Dave Grossman and I would drive up the Taconic Parkway, which was very icy at that time of year to the plant in Poughkeepsie where we would work on debugging these programs and getting them to work. And that was actually very successful. I remember when we first thought it was well enough debugged that we could test some of the code on one of the gates, one of the machines, one of the computers, we found some number, I think it was about 17 wiring errors. And after that the production people didn't want to let us have the robot back for any time to continue debugging the programs. <laughs> Eventually we got it back, and there were some number of those. I really can't remember how many. If you can get a hold of Peter Will or Dave Grossman they would be good people to talk to who may have a more precise number. But I was told that the availability of these machines significantly increased the number of mainframe computers that IBM could sell that year because – well, they could sell as many as they could make, and this was one of the great limiting steps in manufacturing. So we did that and kind of concurrently with that activity and then continuing after we were also – I was heavily involved with developing a new programming language for robots, and we called that language AML. And one of the impressions that I had gotten from the work that I'd done at Stanford, this language AL there were control structures and you could say that things would happen in parallel, but we had put a very large amount of effort into very elegant ways of describing robot motions.

What I'd already come to realize, though, is that the elegant description of robot motions is only a small part of what you need to make a practical robot program for working in a manufacturing environment. And so I wanted to develop a language that was more like a really terrifically good interactive programming language in which the robot motions were more embedded as parameters to subroutine calls, things we called C-subbers. And so this language AML was – well, first it was interactive. We were putting it on a very small computer or at least what would now be considered a small computer, a series one computer, and I was writing the interpreter in assembly language. But the language itself looked somewhat like APL, which was an interactive mathematical programming language that IBM had introduced in the sixties, actually, and then we built up subroutine libraries on top of that. And it was actually quite a successful language for its time. Then what happened is IBM decided to start a product group, and they started that in Boca Raton, Florida. And so my manager, Peter Will, was asked to go down and had sort of a research arm, but I think really to be down there to provide technical input, and he brought down – and there was going to be a small research arm, and so he brought me and Pat Panissidi with him to Florida. I think it was officially a transfer, but it was a transfer

with a return ticket with the understanding that I could come back to research after a couple years. And this was a really good way – you can't just transfer technology. It works much better if you move people and then move them back, and so that's what I was doing.

When I got down there for the product group they wanted me not to work in research but to be the product architect for these robots. And so I basically wrote the product architecture for these IBM robots that were commercialized. They had sort of a preliminary sort of beta version called the RS1, and then the product, which I think was actually announced after I went back to IBM Research was called the 7565. It was a very sophisticated robot. It had this programming language, AML. One of the things, all of the IBM robots, even the earlier ones that research had is the fingers of the graspers had force sensors built into them, and you could do actually quite sophisticated things with those. Mostly we still used them for basically contact sensing with forces. So you'd grab something. You'd know when you were holding it. You'd know when something collided or you could find corners of things. So after a couple years I really realized that I'd transferred most of what I was able to do and knowledge and I moved back to IBM Research in Yorktown Heights, where I rejoined the research group. Peter at some point left IBM from Boca, and I believe, yeah, at that point he went and joined Schlumberger-Doll, oil exploration company or oil consulting. But we kind of kept in touch. Dave Grossman I believe at that point had become the manager of the research group at Watson. So when we're back at Watson, again, a lot of what we were doing was research to support the product group but also to support manufacturing groups inside IBM. There was a great deal of synergy there because if you're going to make a robot for manufacturing you want to – you need a lot of experience dealing with manufacturing applications, and a lot of the application set had to do with, again, inspection, but similar to this – I'll get to that in a moment.

But in addition we did some other things as well. One was assembling print chain, basically taking the very slugs that made up the type characters in a printer and basically assembling them into sort of a kit. And there were like 80 subroutines in this AML language of which only a few had to do with moving the robot. Most had to do with things like error correction and so forth. Sometime around there, and it may have even started sooner like before I even moved, but the next generation of IBM technology was – there were enormous manufacturing problems. You had these 1,000-pin chips or some huge number – I think it was about 1,000 – that would plug into these sockets in these ceramic substrates that would shrink a little bit unpredictably, and these ceramic substrates had wires and wires and wires in them, and they're very tiny wires. And, again, we wound up doing a lot of applications in which we would do – and these would plug into big motherboards. And one of the big problems were how do you test these electrically? And there were some very, very sophisticated electrical tests that the other part of research could do, and our job as it were were to find ways of getting the probes in the right place without destroying wires that were a couple thousandths of an inch across or smaller. This whole thing – the internal technology, one of the buzzwords was the Clark board, which I believe referred to these boards which had – I think there were like six or nine of these ceramic carriers in them. Anyhow, the whole thing became Camp Clark Board.

One of the things that was a great strength of IBM Research was that it was sort of like every now and then the survival of the company would be at stake, and you call everyone into the auditorium and say, “What can you do to save us?” And it was sort of like the Manhattan Project, but it was actually very productive, both obviously to save the company but as well even for research. It kept you focused on real problems. And you would see often something that we didn’t know how to do, and that helped set a research agenda that you would have ready for the next time. Other things were happening around that time. We were continuing to develop software that would support the product group in Boca Raton. The ability to control multiple robot arms at once in sort of a coordinated fashion was one thing. The other thing that was happening sometime around then – actually, it was more like 1982 – we began to think about a new generation of this AML programming language, and I was involved in that. Two other research staff members, Lee Nackman and Mark Lavin really were more the leads, and especially Lee, where basically the language was made to be object oriented and really an enormously sophisticated programming language. I’ve always felt that IBM should’ve gotten behind it and pushed it as a product because it basically had all of the functional capabilities of Java but was sort of more interactively friendly.

Peter Asaro: What was that called?

Russ Taylor: It was called AML/X. But so that was evolving. One of the things as well, I got interested, again, how do you specify the real-time architecture of these sensor-based systems and kind of came up with now an idea of a hierarchy of behaviors, although my behaviors were not what Rod Brooks called a behavior, but the idea was that you had some more or less continuous control primitives, some motion or something you’re doing, and then you’re concurrently monitoring sensors and then transitioning to different control modes based on detected sensory events, and so you build up this task graph of these behaviors. And one of the things that was really a simple example we used to describe that was exactly the centering grasp that Lou had used, but you can do much more complicated things with them. And so kind of concurrently with all of what was going on here I became a manager at some point, had a research group. I believe it was called Robot Systems and Technology, although I think that was the original name of the group before I became the manager. And my boss had become a second-line manager, and then the whole organization was expanding because IBM had created a manufacturing research department that we were part of. Eventually I became a second-line manager where I had four groups reporting to me. There was a robot systems group, which was mainly all about software and programming, a robot technology group that was managed by Ralph Hollis. He was sort of the first person I had recruited when I was starting to be a first-line manager and then became a manager when I became a second-line.

There was a robot vision group that Mark Lavin managed and something called integrated automation systems that was managed by a guy named Bela Musits. And there was an intermediate stop where the robotics and then the vision group was separately – the whole

management hierarchy is probably not important for this, but eventually I became the second-line manager of these groups, and at some point it became called automation technology where the computer vision people weren't there, and what I found was I was spending all of my time or it seemed like a great part of my time doing management; budgets, personnel, strategic plans, presentations to hire management, justification of our program and things like that, and I got tired of doing that. And one of the great things about IBM Research is there was this tradition of sabbaticals either between university or internal sabbaticals, and so I said, "Look, let me step out of this second-line management stuff for a while and go out with a- a very small group of either couple postdocs or a postdoc and a couple of research staff members and do-do a focused project to get to the point of- of a complete system that does something exciting and useful." And I really discussed two possibilities with my bosses. One was to take scanning tunneling microscopes, which are very precise and you can see an atom with them and basically make very intelligent robots that had extraordinary precision, and that would've been useful in various IBM manufacturing processes and inspection processes, having sort of atomic scale scanning microscopes. And the other was to take a project that was going to die without someone to work on it and turn that into a complete system, and that was a system to make a robot to do surgery. Within my second-line group there had been a semi-bootleg effort that was done with the permission and understanding of management to build a robot to help with orthopedic surgery.

IBM had been approached by two surgeons, Bill Barger and Howard Paul. Bill worked on humans and Paul's patients were the family pets of movie stars and millionaires. And they were on the faculty at UC Davis, and their research was in the area of custom hip implants. And they did a study in which they realized they were designing these implants and building the implant to a tolerance of five one-thousandths of an inch, and the manual process for making the hole was extremely crude and you would leave gaps sometimes on the order of two millimeters. It's sort of amazing that manual surgery works at all, but it does. And so they had tried various robot manufacturers to see if someone could help them machine the cavity to accept the implant. So they'd come to IBM. After a few false starts, and there are various stories there about what happened, it wound up in my second-line department. There were a couple of engineers who were really interested in it and I was, too. And so that was bouncing around, but then the engineer who was most interested in that was changing jobs. He was going to go be one of the technical staff of the director of research. And we'd been sponsoring some work at Davis and had had a Davis graduate student spending time in our lab and so forth. But the project was going to die for the lack of someone really to work on it because we weren't going to continue to give money to something that didn't have IBM people working on it because we're not the NSF. But I said, "Well, gee, you know, I could take that and in probably about a year uh... I could get something that would be good enough to work on Dr. Paul's patients," the dogs. And my bosses, I was really surprised they said this, but they said, "Well, you seem a little bit more interested in that. Why don't you- uh... why don't you go do that?" And so I did.

I formed a very small group. There were a couple postdocs that I got. One was a guy named Peter Kazanzides, and Peter and I then really in about a year developed the prototype of

what became known as ROBODoc, which was this system for preparing the bone for hip and knee implants. The other project I was working with another postdoc named Yong-Yil Kim, and we were developing a system that combined surgical navigation and a passive manipulation aid to help do craniofacial osteotomies. It was one of the – there I was working with a surgeon at NYU Medical Center called Court Cutting. That's a great name for a surgeon, <laughs> actually. Dr. Cutting was an extremely good programmer and computer scientist, actually, in his own right. But in any case those were the two projects. In about a year Peter and I had a system that for various reasons including some concerns about animal rights advocates IBM wanted us to have no formal relationship with that effort after a certain period of time. And so we literally turned over on the last possible day before the shareholder's meeting we completed the qualification task on a cadaver dog bone and turned the system over to UC Davis. And then over the next few months the director of research realized that this could be a great test case to see could you commercialize something that is completely outside of IBM's business interest, and so they started a company, which was called Integrated Surgical Systems, to make this orthopedic surgical robot.

In the meantime I had realized that there was a much larger opportunity in medical robotics for an information company, and so I started a research group within IBM Research, which we called the computer-assisted surgery group, which developed a lot of the early technology for surgical robots. One of the things, again, we were doing was we were continuing to support this start-up company and doing research on various things that would be useful for their application. The other thing that we were doing, and I continued to do some work on this craniofacial osteotomy thing, one of the other things we did in partnership with Johns Hopkins University is developed a robot for minimally invasive surgery, endoscopic surgery. I'd been asked to give a guest talk at a surgical meeting for a society called SAGES, which was at that time a very small society, but these are the endoscopic surgeons – and of course now they're huge – and realized that this was an area where a robot could actually help, and so could we develop robots that would initially manipulate endoscopes but also could very, very precisely manipulate surgical tools under either image guidance like under x-rays or video or under various sorts of guidance or under sort of manual control. So we developed a system we called the LARS with Johns Hopkins. The LARS was an interesting system. Probably the thing that was most obvious about it is if you stick a tool into a patient you really want the robot motion to pivot about where the tool goes into the patient rather than do this. You can always program a robot to make those kind of motions, but it requires the part up here to move rather fast, and that didn't seem such a good idea for a surgical robot. So we built a robot with a natural mechanical isocenter, and it came to be known as a remote center of motion. A very high proportion of the surgical robots today that are used for these kind of procedures have that feature built in, and there are several ways of doing it. The one we developed that we used was one that is probably the most common. So we did that. The other thing that happened, there were a couple other potential commercial partnerships that we explored then. One was for a surgical navigation system with another company that was nearly the size of IBM, and that one kind of fell through

when my bosses or the grand boss got cold feet. For a number of reasons I began to wonder if inside an information company was the best place to do this sort of research.

Peter Asaro: So prior to the work on the surgical robots was there interest in IBM to do other kinds of non-manufacturing applications?

Russ Taylor: Oh, yeah. Well, IBM did a bunch of things including – people don't realize this but was one of the early developers of magnetic resonance imaging among other things. Another product that was marketed by the same division that had the surgical robot was an electrocardiograph machine, EKG machine. So they had these but they were never all that successful. It was almost like – the culture of an information company is different, which in a way is too bad because if you think about it – and what's actually the vision has continued to drive what I do in my current center, is really information centric. You're dealing with partnerships between people and technology and information to change a process. So for medical robots the idea is you start with everything you know about the patient. A lot of that is in the form of medical images, but eventually other things that you want to fuse in, the clinical record, lab results, all sorts of things, and you build up a representation of the patient. You call it a model. It might look like a computer graphic model. It usually has that element, but it's a data structure that lets you do the rest of the process, which is you can diagnose what's wrong with the patient, you can plan an intervention, but then if you can take all of that information into the operating room or intervention suite and can register all of that to relate all of that information to the physical reality of the patient then you can use appropriate technology to do what you plan to do and verify that you did it. And so for me that's a control loop closed by information.

The other thing – and that's a remarkably similar thing that you might see in computer-integrated manufacturing or computer-assisted manufacturing. The other thing is that control loop. That process really occurs in many time scales from an entire patient treatment cycle all the way down to every second in the operating room. The other thing that was I think in the vision from the beginning is the observation that hospitals in many ways should be run more like factories, not that you treat people as mechanical objects but realize it's a process. One of the keys in manufacturing and especially if you have information-driven machines like robots involved, is you're much more consistent in what you do, and you have all of this information you're using to control the process. Well, in IBM we saved all that information. So you had a wheel problem, say, out on a disc head line or something. First thing you do is you go mine that data. The word data mining I never heard until much later, but that's what we were doing. And either that would show you what you were doing wrong or it would tell you where now to go do some other tests, which often a robot or something would help you do to help diagnose and improve the process. We call this process learning. So what you would really like to do is in medical robots you have the same potential. You can treat each patient better. You have this information loops that close that can give each patient an optimized intervention with fewer side effects, greater safety and all of that, but then you can use all of that information, which lets you

do more precise and more consistent execution and given you what amounts to a flight data recorder for what went on. So you know what you did. You were consistent. Eventually you know the outcome. The vision is you ought to be able to combine all of those and take advantage of it to improve your treatment plan and your treatment process.

So that sort of was always the driving vision. Every time I would be at Johns Hopkins someone would pull me aside and say, “Why don’t you move here?” And eventually I realized that if you want to do that it’s easier if you’re in the same institution as your customers. In IBM for the first part my customers were the manufacturing engineers I worked with. Now they were these surgeons. So in ’95 I moved to Hopkins. And medical robotics at that time was considered highly exotic. The ROBODoc Company was out there and there were a couple other startups. One of them was Intuitive Surgical just getting started, but it was still highly exotic. And this may be a good time to mention sometime after I left IBM, IBM effectively sold my patent portfolio to Intuitive. They licensed all of that technology, and so some of those patents from the IBM systems were I believe used in the da Vinci robots among others. In any case, so here we are at Johns Hopkins with this nascent field. And, actually, one of the first people I told even before I moved to Hopkins I was on some panels at NSF. I’d been on advisory boards and the like, and I happen to mention to them that I was moving and they said, “Why don’t you apply for one of these ERCs? Uh... the proposal is due in a month and a half.” I said, “That’s not feasible,” but a year later we did apply for an engineering research center where it was a two-year process of hell, but we put together – it was referred to as the Dream Team Proposal. It was led by Hopkins. I was the director. But we also had Carnegie Mellon. Takeo Kanade was the first CMUPI, MIT and Brigham and Women’s Hospital. And so we got something over \$30 million in seed money from the NSF to pursue that strategy of computer-integrated interventions of which a key part were robotics, but it really is the whole thing. And while we were spending out the seed money – we spent a total of about \$65 million. NSF wants you to keep track of all of the spending while you do this, and so we did that and there were many papers. It really is I think what validated the field of medical robotics as a research area. Many of the people in these big medical robotics tracks are the students or the grand students or somehow influenced by what came out of that center. And, of course, the other real driver has been the commercial success of Intuitive, especially.

Peter Asaro: And what year was that that you received the grant?

Russ Taylor: Well, I moved in ’95 and we got funded in ’97. It was really ’96 when most of the proposal writing and early ’97 that most of the proposal writing process and multiple site reviews, and there was a preliminary – it was huge. It was basically 18 months of hell, but we got it. <laughs> And then like all of these ERCs it gets seed money in a critical mass of people, and that is also what gave us the critical mass to build an amazing research program at Hopkins as well as other places, or we do things now other than just medical robots.

Peter Asaro: So who were some of the roboticists that you were able to bring to Johns Hopkins?

Russ Taylor: Okay, well, let's see. First person recruited specif – well, to start out with there was me and Louis Whitcomb who joined Hopkins the same year I did, and Greg Chirikjian had already been there, but then recruited Greg Hager who was a computer vision person. I tried to hire him when I was at IBM, but he had wanted to go to academia, and so he eventually <laughs> came back to work with me. So we hired Greg who eventually became the research director for the center. Allison Okamura was the second hire. In both cases each year you got an IROS or an ICRA and say, “Gee, that’s the hire of the year,” and we would smile. So Allison who does haptics and that sort of thing was hired as an assistant professor. A couple years ago for various reasons she moved to Stanford, but that simply expanded our diaspora. Other people, there’s some research faculty especially in this area. One of them is Peter Kazanzides who was a software and systems engineer who had been my postdoc who developed ROBODoc. There was another faculty member, Euleen Yartakita, Noah Cowan. Oh, gee, Marin Kobilarov is another recent person who’s joined us, Ralph Etienne-Cummings. They’re all told – at Hopkins there are like 18 robotics faculty of whom 10 are resident in the lab, but it was people who we would work with. At CMU initially it was Takeo, Branko Jaramaz, Tony DiGioia who’s a surgeon, Cam Riviere who does microsurgery work. MIT, originally we were working with Erik, although he’s now sort of moved on off into administrator land. <laughs> Ron Kikinis and a number of more image guidance people at Brigham. So those are some of the people.

Peter Asaro: What was the robotics at Johns Hopkins like before you arrived and <inaudible>?

Russ Taylor: Well, Greg Chirikjian was working a little bit on sort of snakes and modular systems and a bunch of theoretical things. I arrived the same year that Louis Whitcomb arrived, and I think Louis got there maybe a few weeks or maybe a month or so ahead. And Louis’ main focus is undersea robots, but as well he got quite interested in design – he’s a terrific design guy and had recruited a – and I actually helped him recruit a postdoc, a guy named Dan Stoianovici who was actually hired by urology and –

<break in recording>

Russ Taylor: – founders. And so we were developing between us technology for applications in which you poke a needle into a robot under very accurate guidance of imaging, of CT, MRI, x-rays.

Peter Asaro: Who are some of the graduate students that you’ve trained who are continuing to do work in robotics?

Russ Taylor: Oh, I'm terrible at remembering names. Okay, I can mention a few. There was a guy named Andy Bostek who went to work for Medtronic very early on. Actually, he was a Vanderbilt student who was a postdoc with me at IBM. And then at Hopkins a guy named Steve Schreiner. I'm trying to think. Okay, there was Ming Li who is at NIH now. There was also a guy named Ankur Kapoor who was working on this very precise robot stuff. Most recently there's a robotics student, Marcin Balaki and a student, Kevin Olds who – Marcin was working on a system for ophthalmic surgery, for retinal surgery. We do a lot of co-advising, so who's my student, who's someone else's is often a little vague. Oh, gee. Why am I so bad at remembering grad students? It's somewhat embarrassing. There were also a number of students who did sort of more imaging or image-based modeling things. There was a student, Jianhua Yao who is, again, now at NIH, and he did early work on 2-D, 3-D registration. I had a student, Rajesh Kumar who worked on some robot things who worked for various places. He was back at Hopkins for a while, but I'm not sure where he is now. He's somewhere in California. In the early days these students, you'd sort of work together. One of the things, we had a project funded by NIST with IBM and the ROBODoc Company to develop methods for what are called revision surgery where you've earlier had a failed hip implant and now you have to clean everything out and make a slightly bigger hole for basically a replacement implant. It's much more demanding than primary surgery, and we did various pieces of that.

One of the key things there is how do you actually find the bone, and so we were using various techniques with x-rays to do a process that's called registration, so those were some of the early, early work. There have been a number of other students who were working on things like statistical modeling of anatomy. If I have a CAT scan of you I can use that for planning, but if I don't have the CAT scan I usually know you're a human. I usually know your sex. I might know a few other facts and maybe have a couple x-rays. So can I put all of that information together to make a very good guess as to what your bone looks like and then I could use that for planning? And Yao, that's what he was doing with his thesis. There was later another student named Ofri Sadowsky. So I've had a whole line of students doing those sorts of things. Postdocs, probably the one who's been the most influential in robotics or medical robotics is Nabil Simaan. Nabil's advisor in Israel in Technion was a guy named Moshe Shoham who has actually started a company called Mazor Robotics. But Nabil I hired as a postdoc actually to do some things that Moshe and I had been talking about as part of a project that we were trying to get funded, but the idea is continuum robots. Can you replace bearings, pulleys and pivots, which is you try to make the mechanism smaller and smaller and get harder and harder with bending metal, and we had two ideas. One was what are now called concentric tube robots and the other is to use – which is what Nabil spent his time implementing are parallel. If I have two pieces of bending metal and I pull on one and push on another I'll get bending. With three pieces of bending metal I can get bending in two directions. And so we built a prototype using this that was smaller in diameter than a da Vinci surgical robot but had more degrees of freedom. You had three concentric tubes in a four-millimeter form factor and through each tube was a wire so that you had two stages, and then you had a little gripper that would be actuated.

And so these continuum robots of both sorts have become increasingly common. I should say that at Hopkins the graduate student who did really the most with the concentric tube robot idea was a guy named Bob Webster who was Allison Okamura's – Allison was his primary advisor. I was on his advising board, and I can't remember who the other advisor was. But Bob is now also a faculty member at Vanderbilt University, and a major portion of his research is these concentric tube robots, which were a big part of his Ph.D. thesis, and he's done some amazing things with them. Nabil was a faculty member at Columbia. After he finished at Hopkins as postdoc he then went to Columbia and then moved maybe two or three years ago to Vanderbilt, so he's there now, too, so they have actually quite a strong mechanical engineering/medical robotics group there as well. And so they're now graduating students, and so we have grand students and eventually great-grand students and so forth. So that's some of the names, at least.

Peter Asaro: Okay. What do you see as the biggest challenges facing medical robotics going forward?

Russ Taylor: <sighs> The biggest challenge. Well, there are obviously discrete technology challenges largely focused around biocompatible miniaturization, making things smaller, more dexterous, more capable for sensing, but in a way I don't actually see that as the major challenge. I think there are obvious also barriers to commercialization or even human trials. I mean you have much harder issues to do with safety and sterility and institutional review boards and the FDA, so the total cycle to get to a human is much longer than, say, you would have for a manufacturing robot to get to the plant floor or a construction robot to get to the construction site or whatever, and that's a challenge.

I think the biggest challenge, though, actually comes from the fact that even more than other forms of robotics I think medical robotics you've got to think of it in terms of systems. To do anything useful you practically have to be able to do everything, and you have these other constraints as well impacting that. And so that means that you have to train almost all of your students to have an appreciation of a complete system and even the parts where it's not their narrow research specialty. So there are mechanical engineering students who are doing amazing work developing excruciatingly sensitive miniaturized sensors to build into surgical tools, but they need to be able to work with and understand both the clinicians and the software guys, and so you have to go big teams and you also need a reusable infrastructure, and a lot of that is software. I mean maybe this is the computer science person in me, but a modular software architecture, system architecture that allows you to quickly move results – you've got, say, develop something for a da Vinci robot and now you want to try it on a microsurgery robot or vice versa, something I think very important. And the development of a shared research infrastructure that enables that I think has always been one of the barriers and something that we've put a lot of our effort at Hopkins into and are now fortunately beginning to see a real research community beginning to emerge for these sort of things. So some of this I think would

be common to other fields of robotics as well. The medical ones you have on top of that all of the, as I said, FDA and things like that. So that's probably the main thing.

Also, there's a great deal of anatomical variability in humans, and what you need is ways of capturing that but still being able to synthesize what's common between each person, and so statistical modeling of things like anatomy and using this flight data recorder to analyze what motions are needed are common. Of course, some anatomical variability is I think a very interesting research challenge that we have. Other things have to do with outcomes measures. In the end this process control loop, which I think is going to be enabling actually for fundamental changes in medicine. You need to be able to measure outcomes better than we can. To relate clinical outcomes that are often vague and often very long term to technically measurable outcomes and then develop reasonable metrics for that kind of thing, and so that's another research challenge that people are looking at. At Hopkins one of the key people looking at that is in fact Greg Hager.

Peter Asaro: So given that you've done so much work on programming languages for robots I'm curious what your thoughts are on the ROS open-source robotics system.

Russ Taylor: I think it's very good. It's not a cure-all for everything. The shared infrastructure that we talked about talks to ROS, uses ROS, has open interfaces to ROS for the things ROS is good for. There's another layer of things in medical robotics that are a little bit closer to real-time interactive capabilities that you can use the ROS stacks for those but it's not always super convenient, especially for some of the things we do, and so we've kind of built another open-source environment that interoperates with ROS that does those things well. But it's not the panacea. It really isn't. But I think it is nevertheless a very, very good element in trying to make it easier for people to share results. So I'm very supportive of it and we use it kind of ubiquitously in a lot of our research. There are other things it's not so convenient for, and for those we've got other solutions, but I think all of these open-source packages, one of the crucial things is to pay attention to how they can interoperate, and so I think that's generally something that we need to do in robotics.

Peter Asaro: And do you see that there's still space for some kind of universal robotic operating system of the next generation sort, or are the applications –

Russ Taylor: Well, I honestly don't know. For instance, one of the things that – the universal robot operating system, no. Or maybe there is, but I think it's not the crucial thing, although open interfaces are. And I might have very different hardware and ways I might want to do the real-time control of my system, but I want to be able to then interact with sensing in various ways, and so I need probably a menu of open interfaces and ways of doing that. To the extent they can interoperate that's I think very good. I think there are other things. We need better

safety architectures for robots. More and more robots are being called on to work cooperatively with people, and that's an area I think you're going to see more and more attention to. A lot of what you see out there they're very largely aimed at keeping the robot from running amuck, which is something that good engineering requires you to do. Some of them you can also have like collision detectors, although if a student has programmed a robot to do that and it gets to about here and hits you it will do some damage, so I don't know if that's the only solution. I think there's a lot needed on a research front, especially watching what happens with telemanipulation, being able to interpret that in the context of a task both for machine learning and for custom adapting the behavior of the robot to whatever it is you want the robot to be doing, so these are some of the things.

Peter Asaro: During your time at IBM was there ever any discussion about trying to create consumer robots or personal robots?

Russ Taylor: Was there ever any consideration for pers – that's a broad question. The main focus we had was more on systems for manufacturing, for maybe some service and inspection applications, but the household robot probably less so, if that's what you're thinking.

Peter Asaro: Or things that would interact more with people directly like from the surgical standpoint.

Russ Taylor: We've thought about that a fair amount. I mean as a practical matter you normally would worry about that, but on plant floors there are a lot more people than most folks who have never been in a factory would imagine, and so there were some consideration of that. But I wouldn't say that was a major focus. One of the things I was quite interested personally as a research thing is using this sort of hand over hand control idea and very precise motion ideas for things like technicians trying to for instance wire up custom chips or assemblies before you automated them. But, no, I wouldn't say at least that I'm remembering that consumer robots were a big focus back then. They're expensive. I mean we often said, well, you should be able to build a robot for less money than it costs you to build a Mercedes or sometimes if we're ambitious for less money than it costs to build a Volkswagen, but you'd have to get the volumes way up there.

Peter Asaro: So is there anything that we haven't covered that you'd still like to talk about?

Russ Taylor: Oh, I'm sure there must be, but I'm not remembering. I'm not sure quite what era you're after. Certainly my career for the last few years at IBM and through the time at Johns Hopkins has been primarily medical robotics, although I'm personally quite interested in some of these other things, and I think we're going to see more synergies closing loops. One of the

projects that Peter Kazanzides, who was the postdoc with me who developed the ROBODoc prototype and then went to the company, came back a few years ago to Johns Hopkins, and he's a research professor now. And one of his projects I call the surgery on satellites project. He's using a lot of these same ideas to explore how can you do satellite repair or satellite refueling in earth orbit when you can't get to them with something like a space shuttle, and the problem there is you have seven seconds of time delay, and so some of these notions of virtual fixtures are pertinent for that, so I think you're going to begin to see more crosstalk from the medical stuff back out to other applications, but I'm not sure. I've been mostly rattling on here. I guess that's what you want me to do.

Peter Asaro: <laughs> It's a lot of material. We always wrap up with a question about what would be your advice to young people who are interested in a career in robotics?

Russ Taylor: I think – well, there are a few pieces. One is pay attention to the relationship between the science and the application. The best robotics research usually is driven with at least one real application in mind or application area, and if you can identify as a researcher – well, first, if you just want to work in the field as an engineer what you need is very broad skills and a fairly broad education, as it's a systems discipline. You need to know some mechanics and electronics, some software, and be really good in at least one and maybe a couple of those. But then if you want to be a researcher I think one of the other things to understand is be able to say, "Here's something useful I might want to do with a robot and here's what I don't know that would enable me to solve that problem," and then be able to clearly articulate that bridge and identify how you will measure progress toward that goal. A lot of what I would consider unmotivated or not so good robotics research tends to focus on some academic question because other academics have been posing it, and you can sometimes even ride that to tenure, but in the end you won't have that much impact, where at least you should be able to articulate if I solve the question I'm trying to solve first how will I know I've succeeded, and second, why is it important I'll succeed and who cares if I succeed? So they're often referred to as the Heilmeyer Questions, but Ralph Gomory who was the director of research when I was at IBM used to ask a very similar set of questions, and I think that's important in any field but maybe especially robotics.

Peter Asaro: Great. Thank you very much.

Russ Taylor: Well, thank you.