An Interview with

ANITA K. JONES

OH 476

Conducted by Jeffrey R. Yost

on

24 June 2015

Computer Security History Project

Charlottesville, Virginia

Charles Babbage Institute
Center for the History of Information Technology
University of Minnesota, Minneapolis

Anita K. Jones Interview

24 June 2015

Oral History 476

Abstract

Computer security pioneer Antia K. Jones briefly discusses her undergraduate education at Rice University and work at IBM. The bulk of the interview concentrates on her graduate education at Carnegie Mellon University and her career as a computer scientist. This includes the discussion of capabilities (a mechanism for implementing naming and security), the HYDRA Kernel, the Take-Grant model, various research collaborations, serving as a faculty member at CMU, co-founding and helping to lead Tartan Laboratories, chairing the Computer Science Department at University of Virginia, serving as the Director of DDR&E, gender and computer science, and digital humanities.

Professor William A. Wulf also participates in this interview.

This material is based upon work supported by the National Science Foundation under Grant No. 1116862, "Building an Infrastructure for Computer Security History."

Yost: My name is Jeffrey Yost from the Charles Babbage Institute at the University of Minnesota and I'm here this morning on June 24, 2015 in Charlottesville, Virginia, with Anita K. Jones. This is an interview for CBI's NSF-funded project entitled "Building an Infrastructure for Computer Security History." Anita, can I just begin with a few biographical questions; can you tell me when and where you were born?

Jones: I was born in Fort Worth, Texas, on March 10, 1942.

Yost: And did you grow up there as well?

Jones: From six months old all the way through college, I lived in Houston, Texas.

Yost: Can you tell me about your parents?

Jones: My father was a petroleum engineer and he was extremely bright. He was a large influence in my life. My mother finished high school but didn't go to college. She was a ballet dancer and was a homemaker while I was growing up.

Yost: As you were growing up, were there any other people heavily influential in your life?

Jones: Probably not. I went through public school, and didn't think much about what I was going to do. As I said, my father was a major influence and what I remember him saying was do something; just do something. It doesn't matter what you do, but just do something.

Yost: Were there particular subjects in school that you had a special interest in or affinity for?

Jones: I really liked school. When I was six years old I ran away from home on my tricycle because I wanted to go to elementary school and wasn't allowed to at that point. They found me across the street from the elementary school. I always loved school and I liked most subjects. I particularly liked mathematics, especially liked geometry and trigonometry. At the same time I went through about three years when I sewed all my own clothes and I particularly enjoyed that. That's actually an engineering construction task, when you get down to it.

Yost: What went into your decision for you to choose to go to Rice?

Jones: My father said I could go anywhere I wanted to as long as it was in Houston. So Rice was the only place I applied. Luckily, I got in. I had a life-forming experience. I graduated in January so I was in what was called in Texas a mid-term class. So I had from January to September to do something, so I went to work for an insurance company and I retyped — there were no computers — I retyped renewal insurance policies. This is one of the most boring jobs in the entire world and it was highly motivating to get an education so I could do something else.

Yost: When you started at Rice did you know what you wanted to major in?

Jones: I majored in mathematics just because I always particularly liked that in high school, as I said before. But I stuck to it; I actually got a double major in math and English just because I had taken a lot of English courses. But mathematics at Rice, at that time, was highly theoretical. I mean, we didn't deal in numbers, I almost never saw a number, and for me it was difficult. I was about a B student but I was tenacious and stuck to it; I didn't switch to another major. By the time I left, I didn't feel I'd — because I worked so hard on the math — that I didn't feel like I'd gotten a liberal education so I decided that I would take a year and a half and I went to the University of Texas next, and I got a master's degree in English literature. Basically, the degree was just an excuse so I could just read. So I majored in English literature, Victorian literature actually, and minored in art history. So my objective was to finish my education.

Yost: When you were at Rice, you finished in 1964, is that correct?

Jones: Correct.

Yost: Any exposure to computers? Had they acquired a computer facility by that time?

Jones: Exposure; extremely little. At that time they had an IBM 1604 and I did take an evening FORTRAN course, which was about six lectures long. At the time, there was something extremely interesting happening, and that was John Iliffe was building a machine there at Rice called the Rice-1. I recall seeing it, but I had no involvement with it and I wasn't working with students who had involvement with it. I knew some of them, and we met them back at CMU later, but I did take one course. There was no computer science degree. However there was a faculty named John Robinson. He invented the quite important notion of resolution theory and he taught only graduate courses, and so I took one of his courses. I don't remember why, I just took it. It had three modules of instruction. First he taught us ALGOL, then he taught us LISP, and then he taught us resolution theory. I was in way, way over my head. That is a great deal to try to pack into one semester, especially for a student who had no background in computing. But I found it very interesting especially.

Later I moved to Austin to go to the University of Texas for a liberal arts degree. My experience with English literature reinforced that there was no way that I was going to be able to make a living with English. I'm not a poetry interpreter, or such. So, I resolved that after I finished at UT, I'd become a programmer. I think I took another language course at UT, but I don't remember what it would've been because I'd already taken FORTRAN. But I didn't do any programming while I was at the University of Texas.

Yost: So in 1966 you complete your degree —

Jones: I did a thesis on Thomas Hardy.

Yost: And you became a programmer for IBM?

Jones: Yes, I moved to Bethesda, Maryland. During the first month of employment, IBM taught me programming, assembly programming. They taught me two things; one is the assembly language for the 790, and IBM was in the later stages of developing the IBM System/360 software, so they taught 360. When I finished, it was clear the 790 was the past and the 360 was the future, so I said I'd take any job as long as it was on the 360. IBM gave me a job there in Bethesda. It was a quite exciting job because it related to what I did later at CMU. I never knew who the customer was; the customer was some classified agency. The project objective was to take two 360-65 machines and make a multi-processor out of them – that is, the two machines collectively executed the same operating system. The job of our little team, which was run by a man named Bernie Witte, was to make the operating system (OS) manage the resources of two machines when it had been built assuming it was executed by only one machine. The challenge was to assure that one machine did not "step on" the other. I recall that I was assigned to learn about some function in the OS, and the documentation didn't match the assembly code at all; so the documentation didn't document what the code was that I was looking at. However, I learned a huge amount from the people on that project, especially a man named John Sapp, S-A-P-P, who I have lost all contact with. This was a challenging first job and I really came to like operating system software because it's the software that manages the hardware resources and poses all sorts of interesting problems. By the way, just for the history, the solution of how you made the operating system software execute well on two machines, instead of one was to ensure that when one machine was performing a function, that it was not possible for the other machine to execute code that would disrupt the actions of the first. The operating system was so difficult for this team to understand — a set of very good people — that the solution was to let the operating system only run sequentially. So if any piece of the operating system was running on one of the two machines, it was locked out of running on the other. The other machine could execute user code, but not the operating system code, not any part of it. So that tells you that we could not even take the operating system and divide it into two, or five pieces, that we knew wouldn't trample on each other when they were executing. That was a very telling decision.

Yost: Was this project under the IBM Federal Systems Division?

Jones: Yes.

Yost: Was IBM the target when you first got out of graduate school to get a programming job, or did you look around?

Jones: I don't remember.

Yost: Do you recall if you took the programmer aptitude test that IBM developed?

Jones: Yes. [Laughs.] It's funny that you ask about that because I was in Austin when I took that test, and I am a competitor. So, I thought it's been so long since I took any tests, I'll just go find some similar tests — these were timed tests — and I'll do them to practice ahead, so that the wheels in my head get used to doing tests again. And I recall that when I was taking those tests, there were pencil marks on the early pages of the test questions — you had the questions on one sheet and you were filling out another sheet — but there were pencil marks. But, as I went through the test the pencil marks got fewer and fewer, and finally there were no pencil marks for the pages that I was working on. I put a few in on my pages. I don't know whether I interviewed with other companies. I do remember telling my mother I was going to try to find a job in Houston and she said 'Houston, you've lived in Houston all your life, you ought to go somewhere else.' That hurt my feelings, but opened my eyes, and I ended up in Bethesda.

Yost: Both with the initial training you went through and then once you were placed, roughly what was the gender mix and was it a conducive in environment at IBM for women?

Jones: First of all, I thought IBM was an enlightened company because they had this open door policy. So if you didn't like what your manager said to you, you could go to your manager's manager, and all the way up the line. There were very few women who were in technical positions; and there were no other women working on the 360-65 project. Our group numbers about eight or nine people. I don't recall getting a security clearance, but the machines were in SCIFs [Sensitive Compartmented Information Facilities], lead-shielded, because of whomever the customer was. But I was quite comfortable working with this team; I got all kinds of support and mentorship from the men around me; again particularly, John Sapp. But, also from Bernie and others as well, and I just thoroughly enjoyed my first real job.

Yost: So working on this project that was classified, did that get you thinking about computer security?

Jones: No, I was wholly focused on operating systems and how you managed the hardware resources running processes, running the disk silo, the interrupt structure, and things like that. I was at IBM for several years and I decided that I would really rather have a job where I was the one telling me what to do. And so I thought maybe I would like to get a Ph.D. I'd think about it one year, and then the next year I'd think about it again, and so finally I told IBM I was going to move to Cambridge. I planned to work full time, but I was going to take some courses in computer science at MIT and the reason was I wanted to find out whether I wanted to do a Ph.D. or not. I figured if I went there I'd either like it or I'd flunk everything and have to find something else to do. IBM says 'oh, wait a minute; don't quit; we can find you a job.' And they found me a job in 545 Tech Square, which was the same building that the MIT computer science department was in at the time. So I moved to Cambridge. I would take courses one at a time; and I think I was only there a year, so I could take two semester long courses. The one I remember most was Marvin Minsky's course in artificial intelligence. Another person and I decided to build an artificially intelligent piece of software, which I later showed to Allen Newell when I arrived at CMU; he asked me to study with him. I had gotten very interested in AI. Minsky was a very challenging, interesting lecturer. But, operating systems were still my first priority topic, so I did not elect to work with Newell.

Yost: What year was that, that you were taking courses at MIT?

Jones: It would be the year before I went to Carnegie Mellon.

Yost: 1968?

Jones: No, that was [pause]; let's see, my notes say that I went in 1969, but it could've been 1968 when I went to Carnegie Mellon.

Yost: I meant at MIT.

Jones: Oh, MIT. Probably 1968.

Yost: And then what led you to choose to go to Carnegie Mellon?

Jones: I knew then that I wanted to get a Ph.D. in computer science, but I was from Texas and I found the New England culture extremely unfriendly. And so I was not going to stay at MIT; I didn't even apply at MIT. I applied to a number of other places but a couple of people from Rice that I knew; for example Peter Freeman, actually, at CMU. And so in the end, I went there.

Yost: In starting there, did you have an idea what area of computer science you wanted to focus on?

Jones: Absolutely. I had decided I really liked operating systems, and that was what I was going to do. And actually, I did take my project from Minsky's course and showed it to Allen Newell. I told him that I was going to focus on operating systems, that's what I really, really care about. I'd always liked system software and so I would've traversed a very different kind of professional career if I'd started working with Allen, focused on complex information processing, as they called it then.

Yost: Can you tell me about coursework you took and what stands out to you?

Jones: Let's see; it's all very vague. I found it interesting; I think it was the second semester I was there that Nico Habermann, who became my advisor — Nico was the only student, I think, that Edsger Dijkstra graduated — and Nico was coming at that point just for a visit and he was going to teach an operating system course. I was slated, in the course of things, in the curriculum, to take linear algebra. I remember going in to Alan Perlis and saying I'm not going to take linear algebra because I absolutely, positively have got to take this course; this is the only time Habermann is going to teach it. So Perlis said okay. [Laughs.] Likely, I never took linear algebra, but I had an undergraduate degree in math, so no matter.

Wulf: He was a very flexible guy.

Jones: Perlis liked to see passion about the disciplines. So I took the standard mix of courses and things I enjoyed most. The theory was the most difficult for me, even though I'd done a mathematics undergraduate degree, and it just more solidified that I liked systems. So that's where I ended up.

Yost: Having taken a course with Marvin Minsky and Nico Habermann, and interacting with Allen Newell, did you have a sense for what giants they were intellectually in computer science at the time?

Jones: I didn't. I mean, they were all extremely intelligent and talent and at the same time really nice, nice people. I didn't get to know Minsky; but one of the privileges I had is that at one point, CMU was doing the DARPA renewal. DARPA funded computer science departments by giving umbrella grants and so the universities would propose a set of projects, and Newell must've drawn the short straw because he was responsible for drawing up the proposal. I was basically his sous chef and did a great deal of work on that, which was very educational because we focused on all the different research activities being conducted and envisioned at CMU and documented them in the proposal. We reasoned through technical arguments and so it was just a real privilege to be that close to Newell when he was doing that.

Wulf: Just to elaborate, these grants were block grants. They were justified by the set of projects we were going to do, but there was no guarantee that we would wind up doing those things. It really was very flexible money in large amounts, multi-millions of dollars over fairly long periods of time, like five years or something; I don't remember exactly. So a lot of money, flexibility, long period; and that was just the way ARPA did things.

Yost: And this, of course, was out of the IPTO office?

Wulf: Yes.

Jones: And so, my graduate fellowships, which I had every year, were funded by DARPA. I was one of a legion of people, all across the country, that DARPA grants in computer science and electrical engineering supported, and it is that set of people that really catapulted both the software and the electronics industry because you had a bunch of people that were probably all smart to begin with but they had the freedom to only focus on learning their discipline when they were in school. For example, there was a bunch of people who later went to Intel, you can just look down the list at who were major designers of this chip, or that device or whatever, and they track back to having been funded by DARPA, so one topic we'll talk about later, science and technology policy in the United States, and the investment — and it was mostly by DARPA — NSF at that time did not have CISE, that came much later. DARPA basically put in the underpinnings, the people and infrastructure, which was the bedrock on which the U.S.

information industry was built. At CMU and at other places; DARPA also block funded Stanford, MIT [pause]

Yost: Utah and graphics research?

Jones: At that point? I don't know when that would've started; might've been later, but a number of universities.

Wulf: You've been saying DARPA but at the time it was ARPA. It got renamed DARPA . . .

Yost: In 1970, I believe.

Wulf: Yes.

Jones: When I finally went to the Pentagon to run the science and technology program, and the DARPA director reported to me, we had a joke which is they just saved the stationery because Congress would put a "D" on the letterhead, and then they'd take it off, then they'd put it back; so DARPA just saved the stationery.

Yost: In the 1970s, am I correct in thinking that DARPA IPTO became a bit more mission oriented for clear defense applications and moved away from truly open-ended long term flexible block grants for computer science research?

Jones: I think DARPA never left the notion of open-ended projects. I mean, they funded speech for decades; they funded autonomous robotics forever; but I think they moved away from the block grant the way Bill just described it, to the universities to funding specific projects, but the projects were often sequentially funded — project after funded project, related to the same fundamental challenges.

Yost: How did you come to work on the dissertation topic that you ultimately chose?

Jones: I don't remember; I don't remember where it came from. I think it came from me, and I don't think it came from Nico because while he was a great advisor, this was not one of the subjects that he had really attended to. My thesis dealt with capabilities, a computing mechanism to implement both naming and security. The notion of capabilities had been invented but what existed very *ad hoc*, and I thought not clearly defined; and I latched onto that. Early on, I did not think so much in terms of security, I thought in terms of naming. A capability is effectively a token that gives you access to some entity, which could be a page of memory, a file, or another process — it could be any entity of use in the system. I've always believed that it is important to solve fundamental problems and still, in my view, one of the most valuable aspects of capabilities is that they are, in effect, a name for something because they give you access to it. So in a capability-based system the active entity — let's call it a process — a process has a set of capabilities that define what all entities that the process can name. If it doesn't have a capability for an entity, in effect, that entity does not exist for the process. It can't reference it because it doesn't have a capability. And so the capabilities of a process define that process' world; what it can touch, what it can write, what it can read, whether it can perform maybe more sophisticated operations on. And that entity could be hardware, for example a disk. So, when the process executing file system code actually writes on the disk tracks, the process has a capability for the disk asserting that it can perform write operations. Now if a user process that wants to access a file entity and is permitted to access that file, then it has a capability for the file and it has to proffer its capability down to the file system software, which has the capability to actually read and write the disk, and that's how the system brings in the contents of the file and makes it available to the user process that would like to use that information. And so, again, still in my mind, the security is powerful but naming things is even more fundamental. Bill alluded to representing things in higher level languages. Well one of the things you have to have to represent things in software is a mechanism for naming things. Capabilities are such a mechanism. What I did in my thesis is I defined what I still think is extremely elegant, rigorous formal description of capabilities that had not been done before.

Wulf: Can I jump in here?

Jones: Yes.

Wulf: I think of capabilities as an abstraction because we all know how important abstraction is in structuring programs. For example, use the word "file". There might be a capability whose implementation involved accessing the file, but the user performed the operations provided by that system, and whether that system is a process, or a subroutine, it provided its users with some operations that you could perform. So it was an abstraction mechanism as much as anything, I think.

Jones: So actually, in this formal definition — I'll give a short summary. There are three levels. Initially there is a "type type", a capability that gives the right to create new types of entities. Instantiations of "type type" yield "types" for example "disks", "files", "processes" and the other sorts of entities needed for the operating system. But further [under operating system control] the OS on request will instantiate new types for a user such as a "book", "IRS 1040 form", "a definitive description of a hairdresser's coiffure", essentially any type the user might dream up. At the last and third level, software holding the capability for a specific type can create instances of that type. It is the same for the OS as for the user. And when you define the type, the particular type, you define a set of operations that are allowed on it. Down on the low level hardware, it's things like read/write, but for more higher abstractions of type, it might be draw, because the representation of the type of entity involved is a graphic something-or-other. And so you have the ability to either read or extract something out of that. So there are three levels: there's the genesis description where the genesis of everything is a type type entity and initially only that type type entity exists; and then you build up an entire universe by instantiating types; whoever instantiates that type actually implements examples of it. And so there's the type type, you create a type, then you instantiate types, and that's the entire universe. It allows you to describe it quite formally and succinctly.

Yost: Was there any prior work that influenced you in any way? Were you aware of the work at Cal-Berkeley on the time-sharing system with capabilities?

Jones: Is that Bob Fabry's work?

Yost: Yes, and Butler Lampson.

Jones: Yes, I was. Yes, I knew them both and they were focused on implementation. If you look at the thesis, it is much more abstract and it's trying to capture the essence of that is. And also I knew slightly, Maurice Wilkes and Roger Needham, who were working on the [Cambridge] CAP system at roughly at the same time. So it was an idea that was in multiple places, which is often the case in research, and people were working on it in different ways, and you did read other peoples' papers when they became available.

Wulf: There was a whole background of program correctness and structuring programs, which was mainly going on in the programming languages area and the programming systems area, but I think strongly influenced the capability model.

Jones: The capability model and its refinement by multiple people paralleled the development of object-oriented thinking. It is related to object-oriented research, there's this whole abstraction of an object-oriented world that I guess people usually trace back to SIMULA, right?

Wulf: Yes.

Jones: To Dahl and Nygaard. Simula requires an entity to have a type or a class, and that's very similar. My thesis added the notion of the original "type type" that could be used to derive typed, and a notion of "amplification" – and this takes us back to security – which defines how the software that defines a type gets access to the representation (the bits) when a user asks that a manipulation of that representation be accomplished. There were places in the world — and I'll come to those later — where this had not taken hold at all; and there were places where this was viewed as a useful organizing principle, and the capabilities aligned extremely well with that. In fact, it helped you define what a type object-oriented system was.

Now's probably a good time to discuss approach; a researcher typically will pick a problem and approach it different ways. If I look back, that really is what I did with capabilities. I still think it is a fundamental and powerful abstraction, which I think as Bill said yesterday, capabilities have not been put in service as well and in as many places as they could. But if you look at what I did with capabilities, first of all I did the thesis. And then —I wanted to get some more experience in actually implementing systems. Bill was starting the Hydra project and so I volunteered to work on that. It was just a handful of students and Bill, and we were in the very early stages and thinking what should this system look like? I thought well, this should be a capability-based system, so I wrote about a three-page description of what the fundamentals of the Hydra system should look like. At that time I had never talked to Bill about what I was doing in my thesis because

Nico was my advisor so I talked to Nico about that. So I walked into Bill's office with roughly three pages and I said this is . . .

Wulf: In pencil.

[Laughter.]

Jones: . . . right. Well, we hadn't gotten a laser printer at that point, I don't think. So yes, pencil. Anyway, I said this is what I think Hydra should be, and I gave it to Bill. He read it overnight and I think we had a meeting the next day; and he said — the next day — this is going to be the basis for the Hydra operating system. I was quite focused on completing the thesis I was not one of the implementers of Hydra, I didn't write any of the code, as I recall. There was some really early stuff, but nothing of substance.

Wulf: She's to blame for the whole thing. [Laughs.]

Jones: I'm proud of that, very. But the Hydra project did a rigorous implementation in its own context, in the C-dot underlying hardware context, and explored the ideas. I did two things later with this. After I had defended my thesis I was in the copying room copying the thesis — you have to give different people copies — and this guy walked in and he said, 'My name is Richard Lipton.' He had been in my class but I had never met him, that is because as far as I know, he never came to class. Just a brilliant guy who didn't come to class. He said, 'I'm really interested in what you did; I think we can do an analysis of that more theoretically.' Off he went off to Princeton or Yale — anyway — and so we started collaborating and did a series of papers. Two other people joined in; one was Larry Snyder and the other was David Dobkin. We did a sequence of papers and developed what we called the *take-grant model* because it was strictly a model of security by that point. On question that could be asked in the model was if you have a known set of capabilities, what other capabilities can you take? And what capabilities can you grant to other active entities? And so we developed and explored that model. This was another excursion in exploring this notion of capabilities. I will point out that capabilities are essentially the only way I know of spanning distributed systems. A distributed system has multiple computers, but more importantly, multiple operating systems and they each define their own world. If you want to give the right to access something in one environment out to entities in a wholly different — across the network — operating system, how do you do it? Well, you pass a capability. So it became; I didn't work in that area – but it became very important in distributed systems. For example, one of my students, Mike Jones, worked for Rick Rashid when he was doing the distributed kernel that he did when he was at CMU and basically they used capabilities in a rigorous way or controlled way to move access permissions across the network between computers. We did the thesis, we had Hydra, we did the take-grant model stuff, and the last thing that I did was actually some years later [was] explored the notion of capabilities and access control in the language expressions, and wrote one paper, jointly, with Barbara Liskov. She was at MIT and I was still at CMU, and she remains a good friend. We looked at capabilities in the context of languages and wrote a paper exploring that. So, all in all, I considered capabilities from multiple different vantage points in trying to explore the idea because I do think it's quite fundamentally critical.

Yost: Did you see any limitations to the capability model for computer security?

Jones: One of the perceived limitations — and I don't know whether it's actually real — is what is the cost to implement? The cost was very high in Hydra because we did not have tag-store. After I finished my thesis, Intel had a project called the 401, I think, where George Cox and maybe Justin [Rattner] did an implementation of a capability-based naming and protection scheme. But it is widely believed that a dominant issue there was just cost, what was the cost to resolve a name? And that was a heavyweight in terms of implementation execution cost mechanism. I think that will be overcome at some point; there are these great ideas that come back and come back. For example, Bill worked on C.mmp, which was a crossbar switch. Well, crossbar switches had been reinvented, and reinvented, and reinvented because it's such a good idea. As Bill taught me long ago, anytime there's a factor of 10 differences in execution speed, you go try something new. Capabilities, I think, are the same way, that they get reinvented and reinvented. And as far as I know, this is the only notion of doing security where you can actually in a controlled way, pass a right to access something in one element of a network system into active entities in other elements, and then allow them to use it and actually come back to touch something in the first node, or operating system, or whatever you want to call it of the other. And so it's just a fundamental idea that's repeatedly used.

Yost: I know Peter Neumann is involved in several projects currently with capability systems and securities.

Jones: I'd like to go look at that; I wasn't aware of that so I'll look into that.

Yost: As you begin working with Richard Lipton and David Dobkin and Barbara Liskov, are you increasingly becoming part of, and interacting with the computer security research community? And are there some early workshops or conferences? I know that in, I believe it's 1977, the workshop for foundations of secure computation is held, but maybe a few years prior to that?

Jones: Let's see; there were two overlapping communities in security. There were the people in operating system, which was mainly where I was because I liked the implementation and the algorithms that make things work in practice, not just in theory. And the key sequence of conferences were the Symposium on Operating System Principles, and that was an ACM conference that was just year by year, and if you actually look at the program committees you'll see a lot of the same people that chair the program committee for one of them. And there was almost always a section of security papers in there. The other community was the theory community that Dick Lipton was plugged into; people like Mike Harrison. Harrison and some colleagues did a very important paper in the middle 1970s — I can find the title but I can't recall it now, I think Rousseau was another author — and they started with some very simple security system models and proceeded to prove that literally any question about them was undecidable.

Now, to somebody who actually implements systems you know, this is very disheartening because that says you can't "know". But in fact, you can implement these systems and they actually work. One of the key papers that Lipton and I authored was on the take-grant model. In linear time you can decide a security question; you can define the notion of "safeness" or "liveness" of systems, and then – in the take-grant model – decide whether a specific system is "safe" or "live". In linear time you can prove whether or not something can happen. And so this was highly gratifying because so much of theory community wrestles with these very difficult problems that are so open that you can't prove much about them. I was always thoroughly pleased that with the take-grant model we can not only prove something, but we could do it in linear time. So those are the two communities, the operating system community that was implementing secure systems, and the theory community that was trying to wrestle with the fundamental questions. You ask the question differently when the model that you're asking about is different. The take-grant model became one of those and I think was a contribution because some things were decidable.

Yost: Within the operating system community, can you characterize the place of the Air Force-funded research that leads the Anderson committee, and funds research at MITRE, and the work of Bell-LaPadula?

Jones: Bell-LaPadula defined a well-known security model, and that would've been in the early 1970s, maybe going back even earlier. But that is one of the models that if something ever rose to a higher level — the model characterized what the government uses for security classification — and once you raise the security level of something, it never comes down because there was no clear way to theoretically model that. I mean, you can do it in practice; you can have people review something and say yesterday it was Top Secret, today it's Unclassified. But in the theoretical models you couldn't do that. So those models, including Dorothy Denning's lattice model, they were very elegant but everything went up; Clark Weissman's watermark; everything just gravitated toward system high and for an implementer — I'm more a pragmatist when it comes to computer systems than I am a theorist — so it felt constraining. And so all these different research findings were swirling around, but I would argue that the theoretical models did not inform advances in implementation very much. The pragmatic advances marched on, for example, the detection of intrusion.

Wulf: I'm not sure.

Jones: I'm not sure either, but those were pragmatic and it was just a way of doing it. We attribute to Butler Lampson the notion of the access control matrix. Implement it one way and you have the capability mechanism. Implement it the other and the resulting mechanism is called access control lists, for example, as used in the Unix file system. It depends on where, let me say one sentence of explanation; it depends on where you store the permissions to access. So in a classic file system access control, you store the permissions with the file itself; that's a classic UNIX implementation. Anybody in this group can read this file but not write it. In the capability representation of access control, you store the statement of who can access what with the accessor, not the things to be accessed. That's why with capabilities, only if an active entity can access something via its capability for it can it even name that thing out there. Whereas in the opposite access control — access control list, that's what we call it — access control list implementations, security control information is stored with the entity and so to find out whether a process or thread can access the entity you already have to know the name of it to check the access control information. So these were duals of each other. Note that the access control list can be quite succinctly represented. The UNIX file access control system is an example. And so those weren't really informed so much by Bell-LaPadula, and by the lattice model, or by the all undecidable stuff. I mean, you can't build an operating system where it is not possible to decide whether this process can access this entity or not. You have to have a way of not only coming to a decision, and preferably, very efficiently coming to a decision. And so all that was going on in parallel and I think all these research activities informed the others, but the theoretical models did not lead, except perhaps for capabilities, to new implementation models, to the best of my knowledge. Is that right?

Wulf: I think that's right.

Yost: Taking a step back for a moment, when your dissertation directly led to the Hydra project, was there a sense that you were making a significant contribution to computer security with the dissertation?

Jones: Let's ask Bill. [Laughs.]

Wulf: Sure influenced me, I'll tell you that! [Laughs.] I think I'm disappointed that it hasn't broadened out and become more popular. It is the right way to do things, in my humble opinion.

Jones: I made a fundamental error, for the record. Classically, after you finish your dissertation you write a paper. I did not do that and so there is no canonical paper out there that describes capabilities the way that I quite elegantly characterized them. I looked at it and said you know, this is so straightforward, so clean, so simple. I made a fundamental error; I should have written a paper. It would be useful to have it in the archive at this point and so I think capabilities are not so well understood as they might otherwise have been. But people like Lipton picked it up. Barbara essentially worked with me on it and so it did influence other people but there's not a canonical paper on it.

Yost: I don't believe there's an earlier dissertation that makes a major contribution to computer security. I mean, Dorothy's did, but it came in1975 two years after yours in 1973.

Jones: I hadn't thought about it that way but that would be a nice thing to believe.

Yost: And I don't believe any of the graduate students that were involved with Multics worked on or wrote dissertations on security. Roger Schell's MIT dissertation, was on a different topic.

Jones: You mention Multics — when I finished my Ph.D. I hadn't thought much about what I was going to do next. Just going with the flow I would be a professor, you know, like my professors. Among others, I had an offer from MIT, but remember I had been to Boston and transplanted this girl out of Texas where everybody is real friendly. I remember I was lost in downtown Boston and I wanted to ask a lady on the street where Filene's was, a huge department store. Her reaction when I approached her was she grabbed her purse and ran away. I thought, I don't want to live in such a culture. So I turned down the faculty offer at MIT and stayed at CMU even though many people counseled me that you should not stay at the same place because you're imbued with that culture and you should go into another culture.

Yost: And generally only top people have the opportunity, since there is a bias against hiring one's own.

Jones: Well thank you, but it's for your own development. I think it was Corbató who said if you're not going to come, why don't you come and spend the summer with us? And so I accepted that. I rented a house just off Inman Square so it was almost walking distance to MIT. And I just spent the summer there, mainly with the Multics people, Jerry Saltzer, and Schroeder, and some other people. It was a great learning experience for me because it was a complete clash of cultures. I was out of a culture that believed in the object-oriented model, and it was completely reinforced because as I said earlier that was so aligned with capabilities, and they had just come off of doing Multics. Multics was well-known; I think the book might even have been out. This would've been the summer of 1973, I think.

Yost: So Organick's book?

Jones: *The* classic book on Multics, whoever did that.

Yost: Elliott Organick's book published in 1972.

Jones: That sounds right. And so I went there and I essentially did battle every day because they didn't believe in this object-oriented stuff; I mean, their protection model was based on concentric spheres. Or sorry, they're not three-dimensional, they're two-dimensional circles; concentric circles, for which there was not a good — at least as far as I knew — a good theoretical model. It did not provide an abstraction on which you could build higher level things. It was all read and write memory, very flat, and quite unforgiving for some kinds of relationships that you might want to build. There's a very nice paper by Mike Schroeder, among other people, about problems you'd like to solve, like "mutually suspicious subsystems". You cannot begin to solve that with concentric spheres because if they're mutually suspicious — sorry, circles — one has to be more interior and therefore have more access than the other. Whereas with capabilities there's no such structure and you can just give each the capabilities that it needs; and one system may have a capability for a type of object that is actually defined by the other, but to actually manipulate that, the one who has the object has to pass it to the other which defines that type of object, to actually manipulate its representation and then it would be handed back. And so capabilities gave you a mechanism to address such problems. I'd never been challenged quite the way I was at MIT. Sometimes I felt a little wire brushed, but it was a very good learning experience and it sort of complemented the fact that I wasn't extracting myself out of the professional culture I had grown up in by leaving. So it was a real gift that MIT allowed me to come and spend a summer with the systems guys because there were some very bright, good thinking people there and it was a pleasure interacting with them.

Yost: And do you recall what year that was?

Jones: It was the summer I finished my Ph.D. so I would've been 1973 or latest, 1974.

Yost: And in 1977, a major workshop was put together that was the basis for foundations of secure computation. You were a co-organizer along with Richard DeMillo, David Dobkin, and Richard Lipton and you were one of the four editors of the resulting book. Can you tell me about the organization of that event and your sense of what it achieved in putting the work out?

Jones: First of all, it stems from that meeting with Dick Lipton in the copy room at CMU because he and I, and then later all four of us did a set of papers that explored different aspects of the take-grant model that was based on my thesis work, but we extended the model and the analysis. As I said, there was a set of, a community of, people with a theoretical bent who had been working on different models. Dorothy had the lattice model, we had the take-grant model, and there were different people; and there were — just looking down the list of names in the book — what a great list of people! Gosh. Let's see, so it was at a time where there were a lot of models and security was a crucial topic for the implementation community, which was not particularly represented at this conference. I notice that Bob Fabry was there, and Jerry Popek was there; they are implementers. There were people at this time thinking about how you incorporated security into languages and databases; interest had grown, and so there was an element of security in databases that was discussed here. It was a conference that brought together that community at a time when there was just a rich set of ideas and alternative approaches being explored. So we decided to do the work actually to create a book that had the papers in it, that captured that because it was such a crucial time in the development of security.

Yost: In looking at the book, it looks like it would've been a really rich text for emerging computer security courses. Do you know if it was used as a course text?

Jones: I did not use it as a course text and I don't know if others did; it is a set of papers and so for lower level courses you want a coherent, gradually developed set of materials. But for a graduate course, where you want to just read the state of the art and practice it would have made a good source of readings. I don't know of a specific person who used it that way, but at the time it would've probably been an interesting thing to contemplate.

Yost: You mentioned the field of secure databases, that's an area you've published in. Can you talk about that a bit more?

Jones: I have really not done that much that I recollect sitting here at this time, in that area. What was done there is frequently associated with an object-oriented abstract basis because people think of the sets of records as being of a type of object, and so there is natural affinity, but this is not an area that I've done a lot of work in.

Yost: You mentioned the ACM Symposium on Operating System Principles as really important in the early days, a conference that brought computer security researchers together. If I remember correctly in the 1970s or late 1970s, Steve Walker organized some meetings I think in part funded by NBS. Did you attend any of those on computer security?

Jones: I don't recollect that, no.

Wulf: I don't even recollect the name.

Jones: Before we leave SOSP, just again, for the record, one of the things quite unrelated to security or operating systems that happened at the Austin Symposium on Operating System Principles was out of the hundred-and-some-odd — 110 or 120 people who were there — about 13 of us were women and we decided to have dinner together. And so we all sat at a table and a half, and men would keep coming over and ask can we sit with you-all? We'd say no and it was at this meeting that we created SYSTERS, and the SYS was for systems because all of us were systems researchers. Anita Borg was at that conference and she volunteered to manage the mailing list that consisted of women in the research community. It grew and grew and was an important way for the women – who were greatly outnumbered by men in computer science – to stay in touch. We recently had a reunion of the SYSTERS people and I went and found the names of every single woman who was at that dinner, which I can give you if we want to put that into the record and I think we should.

[The founders of SYSTERS and their professional location at the time they were in Austin (given in parentheses) and their professional location at the time of the reunion are:

Anita Borg – (DEC SRC) Founding Systers' Keeper – deceased
Stella Atkins – (SFU) Professor Emerita, Simon Fraser University
Miche Baker-Harvey – (DEC) Member of the Technical Staff, Google
Carla Ellis – (Duke) Professor Emerita, Duke University
Joan Francioni – (MTU) Professor, Winona State University
Susan Gerhart – (MCC) Retired
Anita Jones – (CMU) University Professor Emerita, University of Virginia
Rivka Ladin – (MIT) Consultant, DKS Consulting
Barbara Liskov – (MIT) Institute Professor, MIT
Sherri Menees Nichols – (CMU) Community volunteer in education and politics
Susan Owicki – (DEC SRC) Counselor, Stanford University
Liuba Shrira – (MIT) Professor, Brandeis University
Karen Sollins – (MIT) Principal Research Scientist, MIT]

Yost: Yes, that would be great.

Jones: It took a lot of calling around to find that because nobody remembered everybody. Anita Borg was there, and she was working for a company. Almost all the rest of us were working for universities. Anita was working for DEC, I think, at the time; and she volunteered to maintain the mailing list. That's how she became her SYSTERS keeper, which is how she would sign her email. She maintained the mailing list and this grew to include all the women doing any kind of research and then some of the people out in industry doing advanced development in information technology of any kind. And SYSTERS was a valuable lifeline because we were in such a minority; it was a very nice mechanism to remind yourself that you're not the only woman in the milieu because of course, at SOSP, we were a small minority.

Wulf: That was the genesis of what became the Anita Borg Institute.

Jones: Bill, by the way, was for many years a director of that institute.

Wulf: I was a founding director.

Jones: Founding director. Maybe we should add a few sentences back into your transcription. [Laughter.] We should, you know.

Yost: As a new area of research that you're working on it, did you get a sense that computer security was an area any more or less open to women researchers than the broader field of computer science?

Jones: Computer science is well, a science, but also an engineering discipline and in my experience I cannot say I was never discriminated against, but almost never. It's not something that I thought about more than once or twice a year when something untoward happened. If you said sensible things, people would say, 'Hmm, she said a sensible thing; let's keep talking to her about whatever we're talking to her about.' And so I had lots of people, essentially all men, who were mentors and who helped me, and the venues where it was much more closed was, for example, when we started Tartan, you talk to the lawyers, you talk to the bankers, and they wouldn't even look you – a woman – in the eye. That was quite foreign to me because it was very open at CMU and I was always treated as a serious person. People agonize today over tenure decisions and I never thought much about it until one day, the chair, who was Joe Traub at the time said, 'You know, you really have to go up for tenure pretty soon.' I said, 'Really?' And it was not a big event, it was in the normal scheme of things. Maybe it was a big decision for the faculty but I did not view the tenure decision as a sword of Damocles over my head; the tenure decision had no relation to whether I was a man or a woman – at least in my view, and as far as I could tell the CMU faculty had the same view.

Wulf: Within the last two or three months — I tried keeping a copy of this but apparently I didn't — the AAAS [American Association for the Advancement of Science] weekly publication, *Science*, had an article on representation of women in various technical field and I was embarrassed as all get-out. In some biologically oriented fields, it is more than parity; 50 percent or more. The two worst, in terms of percentage representation, were engineering and computer related things that I found incredibly embarrassing. It was down in the 20 percent range — I don't remember exactly — but more than a factor of two away from the best examples.

Yost: There's also been a major decline since a mid-1980s. I believe women peaked at 38 percent of majors in computer science in the early to mid-1980s.

Jones: Enrollment though has been rollercoastering. Now, of course, it's sky high.

Yost: But in terms of percentages of women majors have decreased substantially; by more than half that, so it became less than 20 percent in recent years I think. We actually did a book; the CBI director did an edited volume called *Gender Codes*. I can give you a copy.

Wulf: Oh yes, we'd both be very interested.

Yost: I'll send one out. Can you tell me about what you began teaching when you became a faculty member at CMU?

Jones: Operating systems!

Yost: So that was continuous throughout [pause]

Jones: Yes, I recall teaching software engineering and I don't know that I taught cybersecurity; I probably did teach cybersecurity at CMU, and I certainly taught it at UVa; data representations; the things that had to do with actually implementing systems that did things. So, software engineering because you need all that analysis and process; data representation, because you have to represent things; I think I once taught compilers but Bill is the expert on compilers, not me. But operating systems was just a key focus from the beginning. I must say I learned a great deal from Nico Habermann on that front, as well.

Yost: I really didn't ask you a question about describing him as a mentor. Can you talk a bit about him as your mentor?

Jones: First of all, Nico was a fine human being. He was an advisor. I mean, in some cases your thesis advisor actually formulates a lot of the ideas and writes the thesis. That was not the case for me. Capabilities was my subject, it was not his subject. But he always asked probing questions. He was always supremely supportive, and his family just embraced me so I still am, to some extent, friends with his wife Marta. He has a son and three daughters. I saw Fritz in Seattle a few years ago and I haven't seen the others because I haven't gone back to Pittsburgh. The Habermanns were from the Netherlands and so they celebrate Christmas I think on December 5 and at night, Sinterklaas comes and leaves gifts. They had a tradition that with each gift you give you also write a poem. I was included in the holiday activities and felt like a member of the family. They were very warm, all of them; Marta was wonderful, as well; just warm and welcoming.

I do remember one incident [when] Nico went on sabbatical to Europe and he had four students (Tim Titlebaum, Larry Snyder, myself, and one other) who hadn't finished their dissertations but should or might finish in time to graduate in the spring. Nico asked us to send him a telegram jointly and the last thing in the telegram was to be a "one bit" indicating "yes/no, I will finish in time to graduate in May". We were to put a "one" bit for yes and a "zero" bit for no. We did that; mine was a "one". However, the telegraph people must have thought that the ones and zeros was just garbage at the end so they dropped that off. Nico was very angry that we hadn't acceded to his simple request. He needed the information and he didn't receive it. [Laughs.]

Another recollection is that CMU was a warm and wonderful place; just a hotbed of students talking about ideas. So, in my class you would find in the acknowledgements thanking people for their help in the front of a person's dissertations. You will find a bunch of us that acknowledge the contributions of the "brownie plate group". There was an ever changing group of about a half dozen people who would go to lunch at the student center and somebody, I think usually Larry Snyder, would buy brownies and they came on a stiff paper plate. Somebody or other would be talking about some idea, the brownie was now gone, the brownie plate was flipped over, and someone drew a diagram or wrote an equation to amplify on their idea. I think that for a while Larry kept in his bottom drawer a bunch of brownie plates with ideas scratched all over them. In my dissertation and in others there is a thank you acknowledgement to the brownie plate group for all the discussion that contributed to education.

Yost: Great story. Did you have any doctoral students that focused on computer security in the period from 1973 to 1981, when you were on the faculty at CMU?

Jones: When I was first a faculty member I took responsibility for the software operating system for a new parallel machine, a CM*. This is Bell and Newell notation, you know that? C for computer, M for memory, and * for many memories. There were 50, I think, roughly, computers and memory units.

Wulf: It was an architecture in which you could grow the number of processors and memory units, whereas for the C-dot machine, it was 16 and 16.

Jones: Right. Except that the cost for accessing memory from a specific computer was dependent upon the distance from a computer to memory. A memory access cost one-time unit; if the memory was physically packaged with that computer. It was three time units if the memory was in the same . . .

Wulf: Cluster.

Jones: Cluster, yes. But it was nine time units if the word being accessed was in another cluster so the cost of a memory access went up by a factor of three each time. This uneven access time constituted a huge hurdle. The software had to manage where things were stored and it really mattered how many hops – i.e. the distance between the processor and the memory. Processors were 16-bit machines, which forces a very constraining address base; it's so small! We expended a great deal of effort on algorithms that managed where data and code were placed in memory.

Most of my students were doing operating systems theses, and that's what I mainly remember. I have not reviewed; don't even know if I have a list of dissertation titles but nothing comes to mind. But I was still focused on operating systems at that point.

Yost: Before we talk about Tartan, are there topics we haven't discussed in the Carnegie Mellon years that you'd like to discuss?

Jones: Why, yes. I made what is now very clearly a transition to working in the national security sector. I had a mentor — I didn't even think of people in terms of mentors — but there was a man named Keith Uncapher who was indeed an important mentor in my life. Keith was at RAND and spun out Information Sciences Institute. For some reason — it never occurred to me to wonder why — Keith would call me every couple of months and say 'how's it going?' Keith had been head of whatever the international computer society was; he created ISI; he acted as though DARPA was an organization for which he was personally responsible to ensure that good program officers went there. He put huge amounts of time and his own personal professional capital in getting people to go to DARPA as program managers. Anyway, there may have been many people that Keith called up from time to time as he did with me; he happened to call me one day when I said, 'I feel like I'm in a rut.' I was probably at the tail end of Cm* and the operating system development for that experimental machine; I was teaching and everything was fine; CMU seemed happy with me as far as I could tell; yet I said, 'I feel like I'm just down in a rut doing the same things.' Keith said, 'Let me think about this.' And then he called me back the next day and his reply was, 'You should join the Air Force.' This was about 1978, 1979. I was about 35. 'Keith, I grew up a girl in Texas, and the one thing I had never thought about was going into the military.' But that's not what Keith meant. What he meant is that I should join the Air Force Scientific Advisory Board (AFSAB), of which he was a member. I said, 'Sure, that could be interesting.' And maybe he told me what they did and really sold me on it, but he arranged an invitation for me to join. So Gene Covert, who was in aerospace at MIT and was chair at the time, called me up and asked if I wanted to be a member. I said, 'Yes, I'd like to do that.' Well, I didn't realize what a new world would open for me. First of all, this scientific advisory board numbered about 35 people — some retired military, almost no academics, military corporation-type people who do work for solving Air Force problems. Everybody, 100 percent of the people were 15-20 years or more older than I was; the closest person was 15 years older. What the Air Force Scientific Advisory Board, as the other advisory boards do for this element of Department of Defense and others, is somehow they decide on a problem and then they form a task force to make recommendations about to solve, or at least reduce, that problem. Well, I had walked through a door into a completely different world. I mean, I was stunned by the gravity and scope of the problems that they dealt with, and the import to society and to national security. And so joining the AFSAB caused a huge change in my life in two ways: one is I proceeded to get more and more involved in the national security community, in their problems, and working with the people who were trying to solve those problems; and second I became much more interested in science and technology policy, and gradually that became more important to me than computer science research. So I joined the scientific advisory board and I just sort of sat at the feet of these people and listened and learned how they solved these problems. In addition, I used to sit behind Allen Newell at theses defenses and I would try to guess what he would ask because I wanted to figure out how he thought about problems. And so the magic in a graduate student education that you are one-on-one with people, and this — I never talked to him about this, he never knew this — but it was just clear that he and other people including Bill Wulf, Raj Reddy, and Herb Simon, thought so clearly about things. And they had a way of organizing things so that the questions they came out with were — it was the right question or it clarified something or they'd make a statement such as "this is not the right way to look at that problem, this is the way to look at it". Especially Allen, I tried to second guess him so I could try to figure out how he was thinking about things. I did that same thing with the Air Force Scientific Advisory Board. It's a completely different venue; my first task force was on a problem of the inability to reproduce electronic circuit faults observed in the theater. I think it was F-4s. This problem really matters. Electronics in Air Force fighters were packaged in boxes and the boxes were inserted and removed from slots in the aircraft. So if you had a fault in one of them, e.g. a pilot said it failed, or the diagnostics said it didn't work, the AF would pull it out and try to reproduce it on the flight line because they had test equipment. They couldn't reproduce the fault; so they would ship the electronics box to the U.S. depot to be repaired. At the depot, they too could not reproduce the error. They'd ship the box back to the flight line in Europe, or wherever, unchanged. And so this was a huge problem and there were aircraft that couldn't fly because they didn't have all of their black boxes. Again, this was a major issue for Air Force mission readiness, because what they do is fly. So that was my first taste of a problem and I just learned immense amounts and I realized that the world was a lot broader than computer science, operating systems, and security, and such and essentially for the rest of my career, including now, part of my life has always been involved in looking at science and technology issues that are important to the nation, often national security, but may be important for research and development that would lead to industry advancement that would strengthen the economy. So I tried to work on those kinds of problems, whatever was important at the time. The benefit is that you get to sit around a table with some of the best minds in the world and try to solve those problems. It is one of the most rewarding things I've ever done and Keith opened the door to that part of my future.

Yost: He was a great supporter of the Babbage Institute and I got to know him.

Jones: Was he? He supported things that he thought helped the discipline, the industry, the country, so I'm not surprised at all.

Yost: I imagine Willis Ware was one of the central senior figures on this board during that time?

Jones: I think he was on it.

Yost: At one point he was the Chief Scientist for the Air Force.

Jones: Oh, that's right.

Yost: I'm wondering did computer security in the 1970s, was that ever an issue of the Air Force Scientific Advisory Board?

Jones: That was not a topic that I recall. Software was a big topic because the litany was that of ten projects that are late, nine out of ten are late because of software. And at that level, they didn't focus on security; just getting something to work was the problem of the day.

Yost: We've gone over 90 minutes, do you want to take a break?

Jones: Yes, why don't we do that.

[BREAK IN RECORDING]

Yost: So in 1980, you and Bill decided to take leave of your academic posts at CMU and start a company called Tartan Laboratories. Can you tell me about that decision and the thought process you went through?

Jones: As Bill told you yesterday, this was a time in computer science where there were many entrepreneurial people starting companies based on this idea or that idea. Tartan, by the way we're proud of the name, it wasn't a made-up techno name.

Wulf: The colors were a tartan.

Jones: From Andrew Carnegie. Anyway, the company was based on Bill's research. He had an optimizing compiler and a set of techniques, a multitude of techniques for optimization that was better than anything else in the world. We actually split from CMU roughly in the first year. One of the things we also observed was that a number of people started companies and stayed at the university, being paid full time. We were concerned about that for the health of especially the top several computer science departments because that sort of behavior basically sapped their strength. We did not think that was a good idea and so what happened was Bill left, I guess, in 1980 and I stayed on for one year, but then I left. We did not think that the company should ride on the university because it just was not the right thing to do. I don't know whether Bill talked to you about getting venture capital, but he was the prime mover in doing that.

Wulf: This was an era when lots of people were spinning out mostly venture capital-based; and so I don't remember where we got a list of names, but we had a list of venture capital companies and contacts. We trooped around — particularly around the West coast — presenting this idea to lots of different venture capitalists. At the time, Ivan Sutherland was on the faculty at CMU and Ivan, of course, had started Evans and Sutherland. One day Ivan came to me and said, you know, I hear you're looking for venture capital; why didn't you talk to me about it? I just didn't think about it, frankly. So he took us to meet the venture capitalist who had funded Evans and Sutherland. His name was . . .

Jones: Teddy Walkowicz.

Wulf: Teddy Walkowicz, right.

Jones: Advanced Technology Ventures.

Wulf: Yes, and he was in New York City. So we went there and made a presentation to Teddy and walked out the door with a million-dollar check in our pocket. [Laughs.] Going back on the airplane with a million-dollar check in my pocket was just unbelievable. Anyway, Teddy was a very insightful guy and maybe I'm biased on that because he funded us, but anyway, that's what we started from. We may have incorporated before that, I don't remember, but that was the real start of Tartan.

Jones: Digression — when we edit Bill's tape, this is not the only time he did the presentation and walked out with a million dollars. Bill and a few other people, in what year?

Wulf: I don't remember.

Jones: In the 1990s sometime. Anyway, it doesn't matter, we'll get the date. He said look, computing for the humanities just got to have immense impact. And so Bill created this notion of you take a set of fellows, humanists — history, English, whatever — and what you do is team them with computer scientists, and you give them the computational resources that would be routinely available 10 years from now. You let them do whatever they want to do on humanistic research with that. And that became the Institute for Advanced Technology in the Humanities. And there are now a number of organizations like that. But, there were none at this point and so Bill made a presentation to IBM and they also plunked down a million dollars.

[Laughter.]

Bill was the first director of this center. However you elected a set of fellows, you elected a set of fellows, right?

Wulf: Yes.

Jones: Several of them did really landmark work. One that sticks in my mind is Jerry McGann; he is in English and he is expert on Dante Rossetti, who was both an artist and a poet. With the computational help, particularly by Bill and Alan Batson, who was also involved in this, Jerry built the first non-sequential book. It is the first hyperlinked book where you could compare the pictures that were painted with the then-current version of poetry on a related subject. Apparently Rossetti revised and revised; he revised both his artworks and his poetry, and so Jerry's "edition" is hyperlinked so you can see the development of it. Amazing. In 2006, I conducted a summit on digital tools for the humanities; I'll just give you a copy of our report. At that point, only six percent of humanists were really seriously using computational tools.

Wulf: The observation was, just think about the enormous impact that computers have had on science and they fundamentally changed the nature of the discipline. We do scientific research differently now than before; we ask questions that we couldn't ask before. My feeling was that we could do the same thing for the humanities. It's not panned out quite like I had hoped.

Jones: Yet. Historians should build simulations based on their theories, and then they should ask about predictive things. This is counter culture for them now, historian look to the past. Economists actually do prediction; I mean, they build elaborate, especially financial simulations. And anyway, IATH was a landmark in humanities and Bill was really the prime mover. Again, IBM was so impressed they wrote him a check. But that's his story, not my story. [Laughs.]

Wulf: After just a two hour or so presentation, and after this presentation they said, a million bucks. [Laughs.]

Yost: That's terrific.

Jones: You can have that.

Wulf: I haven't seen that in a long time.

Jones: So we started with a small group of people, and they were really there to work with Bill; people like Guy Steele and Erhart Ploedereder, and a number of Bill's students who had worked with developing the technology. It was just a very exciting time. Tartan was a great team.

Yost: So in 1981, you joined Tartan as vice president and secretary. Can you talk about your role with the company?

Jones: I was just a mid-level manager, basically, doing whatever needed to be done because it was a small team and you just did what needed to be done. For some number of years, I was responsible for quality control. For me, that was an interesting experience in software development because there is an inherent tension between the developers and the testers. The testers force freezing of the code and the developers say I want to fix that little thing. And the testers find problems and it irritates the developers. So the challenge was how do you make the developers think the testers are their best friends? As I mentioned yesterday, during this time Bill built some innovative software for taking the code and generating the regression tests that you needed to test input/output conditions. So it was intellectually an exciting time.

Yost: Do either of you recall some of the earliest clients?

Jones: That's the first sale [pointing to a framed paper on the wall], that little piece of paper; and who bought the first?

Wulf: I don't remember.

Jones: Chuck Geschke.

Wulf: Oh, okay. Chuck was one of my students.

Jones: And Chuck was one of the founders of Adobe. Bill talked about meeting Steve Jobs. Later we found a sweet spot with Texas Instruments, especially the signal processors. Tartan optimization, as I said yesterday, halved the screen performance-wise. I mean, every percent you can get out is value to the customer.

Wulf: It's enabling.

Jones: It's enabling, absolutely. And the contract with Intel, specified that compiler generated microcode had to be five, 10 percent, whatever it was, better than hand coded, which they gave us. And we beat it hands down, very early on.

Yost: So in the seven years with Tartan, were you continuing to spend at least a bit of time staying active in the computer security research community, as well as the policy area that you'd become involved with? Or was it all-consuming?

Jones: It was mostly all-consuming. I didn't do any research, you know, just solving problems at Tartan that maybe you could describe as advanced development, but I was doing no research. I did continue to be involved with the Air Force Scientific Advisory Board because it just took me into a new world that I found most interesting.

Wulf: When did you switch over to the DSB?

Jones: After I "aged out" of the Air Force Board. After some number of years, like seven or eight, you have to leave and so told one of my colleagues, Bob Everett, who was then president of MITRE, and by then I might've been one of his trustees, I can't remember. Anyway, he arranged for me to join the Defense Science Board. Each of the services has a very senior advisory board. I was on the Air Force one, and then there is one that works for the Secretary and the Chairman of the Joint Chiefs, it's called the Defense Science Board. I went on that in the, I think, early 1990s or very late 1980s. Except for when I had to leave for government service, I have been on the DSB and am now a fellow of it, and am currently very active. We have several studies going.

Yost: Everett was one of the brilliant minds, along with Jay Forrester, on Whirlwind.

Jones: Yes, that's right. We still keep in contact with him. He's retired and he lives in Mashpee, a little town on Cape Cod. He, by the way — it's probably in the MITRE archives — but I have seen him give historic talks back to the SAGE days and he has a lot of pictures. Again, it's probably in the MITRE archives because he was head of MITRE, but there are some real gems there. And if you haven't interviewed him, Babbage should.

Yost: I'm not sure, but I don't think we have an oral history with him.

Jones: Why don't you consider that? We exchange Christmas cards with him and Ann, but I haven't seen him in many years and I don't know what he's doing or how he's doing.

Wulf: He's making trouble, somehow.

Jones: Hope so.

Yost: So in the early 1980s, the Department of Defense launches a National Security Center housed at NSA to develop a standards and certification infrastructure, resulting in TCSEC, or The Orange Book, in 1983 and revised in 1985. Were you following these developments and what was your opinion of them?

Jones: Yes, I was following those. In fact, I still have some of those books back in my office, I saved those. But I was not involved in the development and by then I knew quite a bit about Department of Defense acquisition, and I was skeptical of how much penetration those standards could get. I mean, it was expensive to get something certified and they got some penetration, but not really a lot. Fundamentally, the majority of security software developers targeted commercial markets, so they were not highly motivated.

Wulf: I think of those efforts as a convocation of the kind of traditional multilayer systems. Is that wrong?

Jones: No, I don't think so. But in particular, I think TCSEC codified standards for auditing: logs provided a description of history; what do the log entries have to look like – i.e. what information must they capture? What kind of analysis did you do on them? I think that was helpful. This was a time when there was a great deal of research activity in intrusion detection and basically to little end because there's just not enough signal in the noise to detect very much. Intrusion's very hard to detect when software did not have much discipline of development or discipline of structure. In many cases it's hard to detect an anomaly and there was no; there was essentially, to my fundamental knowledge — maybe there's some work there I don't know about — but there was very little theory behind it, it's seat of your pants. But I think it was useful to codify the best of what was known about software development and particularly about auditing and what analysis can be done after the fact.

Yost: Do you think it's fair to say that economics of computer security weren't considered carefully enough with regard to the incentives for industry?

Jones: Then, and now, people are not willing to pay for security. So I wouldn't use the word economics, it's market. Today, the government will not pay for security. So, for example, we have yet another cyberattack task force for the Defense Science Board, and one of the things — and this has been tossed around for two decades — one of the topics of discussion is even though DoD buys a huge amount of off the shelf software; I mean, they are a big buyer of Windows – yet they are not a driver of the market. They can't dictate anything. If the DoD – or the government as a whole – would say we would pay you a premium for some well-defined improvement in security, they could have some effect. In fact, many of us think that would be very attractive to the vendors because they don't want their software to break. They don't want to have a reputation for having bad software and proof of that is that over the last five years, Microsoft has started using some more disciplined software development methodology for building software, updating software, and there are less bugs; there are less security flaws in it. I can't testify to that but some people that I think do know about that assert that Microsoft has made substantial improvement in the quality of their software development.

Yost: We interviewed Steve Lipner about that, the Security Development Life-cycle.

Jones: He works for Microsoft; so you can't put a lot of faith in that, but serious people independent of Microsoft will tell you that that is the case.

Yost: I've heard that from others, as well.

Jones: It's just evidence that they would want to. The government may not be a dominant customer, but first of all the DoD, and second of all the government as a whole, if it would pay even a tiny premium for security; and second, if you could define what would be a modest improvement, and be able to test that you really got that improvement, then in fact I think that the government could incentivize industry to improve security. And I think that was true back in the 1970s, 1980s, 1990s, but they haven't gotten their act together.

Again, I'm focused very much on science and technology problems. One of the things that did in fact work extraordinarily well in software development was the CMU Software Engineering Institute Capability Maturity Model (CMM). They defined five levels and, at each level, a set of questions without answers. And so you want to know whether you're at level three, you get some smart people in software development and say, are this company's answers a reasonable body of answers? And if so, you're at level three. Well this was useful, and I would contrast it to The Orange Book, which laid out specifications of what the software had to do. Capability Maturity talks about software development and what good software should do, but merely by asking questions about the process, and every company answers them in different ways. The CMM made a huge difference for one reason; well, for two reasons. Somebody — and I need to talk to Larry Druffle to find out who engineered this — in the Department of Defense put a condition into all RFPs for software that said that the software had to be at some level of CMM maturity – like level three -- and they did this starting in the 1980s. I was on the board of a company and so I watched in a single case what effect that had. If you think about it from a corporation's point of view, at that point, software was responsible for almost every project being late, whether it was a combined hardware /software project like Fighter Aircraft, or just a piece of database software. CEOs would be asked to invest money into their software development process and they would say what will I get for that? And the software developers would spin stories and try to explain to the best of their knowledge, but their story was not convincing to somebody who wasn't a software developer, so the CEOs would say you cannot tell me what I'm going to get so I'm not going to invest. Now DoD put in a source selection criteria. Literally immediately, every CEO in the defense industry said, 'Whoa, our evaluation on a contract' — and this was contracts across the board, that's why it was genius to do this — 'says we have to be level three.' What does that mean? And, well, here's two pages and it's a set of questions; we have to answer these in a way that somebody thinks is reasonable. For example, there was one defense company that hired me; the CEO said what should we do because we have to be level three? So I went and spent some time and wrote a report and I said you should invest in this, and this, and this. He said okay, I will. I was pretty shocked because I hadn't done much consulting and I said, 'Whoa! That's impressive.' But in fact, that's when I realized what had happened. Because of a source selection criterion, the marketeers in a company, the management in a company were going to pay attention to this. All of a sudden, there was a way of defining ratcheting up of their software developing processes because they would be judged by outside evaluators whether they were level three. And the genius of it was it was not like the prescriptive Orange Book because first of all, you could take the questions and evaluate yourself. That cost almost nothing. You could hire a set of people who had done this for the government before —

you wanted to make sure you would pass, right? — and then you would ask the government to get their evaluators to do it, and I don't know who paid for that latter, but all of a sudden management and CEOs who were skeptical of the loosey goosey software people had a basis on which to make investment in their development process, and software was a good portion of what they delivered to DoD. And so the SEI CMM was hugely successful, and it was because — again, I think it goes to two reasons — one, it was put in as a source selection criteria just across the board. So, management and sales paid attention. Now I've got to figure out how to do this for security, and I know what I want to do, and I've got a fair shot at doing [it]. But second, it was not a set of specifications that were foreign to the company – specifications about what their process should be. That would've failed. It was just to say how do you ensure integrity of testing on whatever the questions were? And so those two things made it highly successful. In contrast, The Orange Book and what's it called now?

Yost: The Common Criteria?

Jones: The Orange Book and Common Criteria are prescriptive; they tell you what has to be, so these two approaches are night and day different. One works, and the other is much harder to get across and somebody's got to pay for it; and it's a foreign set of criteria and maybe the company has a better idea about what they do to earn a living. One has been shown to work and the other had limited success. There was some success, at least at the C level — I don't know how many systems were judged to be at A, B, C levels — at least some were.

Yost: The initial systems that were widely purchased by the companies were IBM's RACF and SKK's ACF2 [interrupted]

Jones: HP had something. I thought they were particularly attentive to the Common Criteria.

Yost: Perhaps later. I think the first three were RACF, a startup company that came up with ACF2, and then Top Secret. ACF2 and Top Secret became part of Computer Associates. RACF was at the C1 and in later iterations certified to C2 I believe.
So can you tell me about the decision to return to the academic world and to leave Tartan?

Jones: We had been at Tartan for six, seven years, and had some success by what we'd done, and we have always gravitated back to academia. I think it just felt like it was time.

Wulf: Yes.

Jones: And so we disengaged.

Wulf: And the company was mature enough that our presence wasn't critical anymore. I think we were still contributing but the success or failure of the company was not going to depend on the fact of whether or not we were there, whereas in the first few years that really was critical.

Jones: The company succeeding; we would not do anything to harm the people because those were our friends and our colleagues, and we'd all been in the ship together. It was a reasonable thing to do, so we just basically stood down for six months and decompressed, did some consulting, and thought about what we were going to do next. I think the big question was do we go back to CMU or do we go someplace else? Certainly for me, I really like entrepreneurial things and once you get bitten by that bug, it persists and so going to a place that was not one of the top places in the world and doing "entrepreneuring it to be better", was attractive. So, that's why we went to a place – the University of Virginia – which was not comparable to CMU. We could've gone back to CMU, but going to Virginia was more entrepreneurial and we felt — at least I felt — like I could make more of a contribution somewhere else than going back. And I was more interested in the science and technology policy, than the research *per se* by that time.

Yost: In coming to UVA as chair of the computer science department in 1988, what was your view of the department and opportunities to enhance and improve the department?

Jones: The department was very small. I think it was ranked in the teens somewhere; it was vastly smaller than the large universities. Both Bill and I thought that there was some really good people and some really good work being done, and it wasn't visible from the outside, mainly because it was so small. Another attribute was that the faculty were desperate for leadership. I mean, to the point that as Bill told you, first they asked him to be chair and he said, 'I'm going to NSF.' So they then said, 'Anita, would you be chair?' The conditions on my doing that are direct personal control over discretionary budget and control over space. Those are the two resources that matter in a university. And a bunch of promises from the dean, some of which came to pass and some of which didn't, for growing the department. Bill and I went, although I guess it fell more to my watch, to deliberately change the culture. So I went in and I said, 'Okay, we need to have graduate students on every committee with faculty to make decisions like on space.' Oh, no, the faculty had difficulty with that. For example, with the space committee, most of the space decisions are where the students sit, so I wrote down a page of guidelines and then we elected three leaders of the students, and for the next years — at least under my watch — students made decisions on where students sit, and I would just check that my guidelines were followed, [e.g.] you can't put all one faculty member's students in the same place. They have to be mixed up by discipline; they have to be mixed up by level because the older students could mentor the younger – guidelines like that. The one — I caved on -- was the budget committee. Faculty said, 'You absolutely can't put a student on the budget committee. Okay, well we'll put them on all the other committees.' We created a very collegial environment in the department; we tried to emulate CMU; it's a very collegial — or it was at that time. When department grow larger they evolve different cultures compared to small organizations. But we tried to do everything we could to make the Virginia department like a small CMU department. When we got there —

Wulf: When she said, 'It got larger' she meant CMU got larger.

Jones: CMU got larger. And that by and large worked at Virginia because it was already a set of people that liked each other. There weren't a lot of friction or wars going on. And remember I said I owned all space so faculty couldn't set up fiefdoms because I owned the space and I could take it from one use and put it to another use. And at that time, NSF had competitions in CISE infrastructure grants. UVa CS had competed in the last two competitions and they had not succeeded. It was crucial that we got money to really beef up the computation facilities, and so I led the next proposal. We went in jointly with EE, which had never been done before, and we won the next round because joint with EE we could make a stronger story for what we could achieve than going it alone. That was just another example of when you don't pay attention to department boundaries, you do what makes sense for what you want to achieve. That was just more culture adaptation. Not all the faculty was supportive of teaming with EE, but I was PI on the grant proposal and I did not know that we could win without their involvement, and I thought we had a good chance with it. We did win an award; it was great fun and I think everybody prospered, the department prospered, and I thoroughly enjoyed it. A lot of academics regard it as a chore to be chair but I don't look at it that way. I think if you're in the leadership position, you can actually vector things, you can have influence that you can't have without being in that position, and you can behind the scenes make things happen that make the department a better place. So we can have great faculty and I think everybody prospered.

Wulf: I can't remember whether I said this yesterday or not so stop me if I did, but when we came to UVa, the liberal arts school was considered to be very good, one of the top ones in the nation. Engineering and, to some extent, science was viewed as this other kind of dirty thing that the all-important humanists looked down their nose at. I think starting with Anita's work in CS, that view has just dramatically changed. Engineering is becoming one of the better schools in the country. We just went through a selection for a new dean of engineering; every one of the applicants was just spectacular and from a major university engineering department. The one we selected actually is a member of the National Academy of Engineering. We just announced a new provost to the university, who comes to us having been dean of engineering at Duke..

Jones: Tom Katsouleas.

Wulf: Twenty years ago the notion of the provost being an engineer was absolutely inconceivable, and it's because the engineering school has become so much better and so much more visible. She's to blame.

Jones: After I was chair, one thing that the computer science department did that was quite absolutely extraordinary, is it started a new degree — of course, the computer science degree in the school of engineering is a bachelor of science degree — a bachelor of *arts* in computer science given in the college. We started that because that was the right thing to do, and the department did it essentially with its own resources. We got nothing from the school of engineering and next to nothing from the college, and it is a very popular major at this point. We just did it because that was the right thing to do. Still, it's the computer science department over at the school of engineering that runs that discipline in the college.

Wulf: It complements this — I don't remember whether we gave it a name before but — the Institute for Advanced Technology in the Humanities, which essentially was a program in the college, in the liberal arts school, housed in the library, actually.

Jones: Which you started. You and Alan Batson did the heavy lifting from the computer science point of view . . .

Wulf: Absolutely.

Jones: So things over in the college of liberal arts evolved — it's actually arts and science but it's dominated by arts and humanities things — all of a sudden here are the engineers creating a computer science degree which is extremely popular with the students. Here comes the Institute [IATH], but it's over there and it's run by Wulf and Alan Batson was a prime player. Jeffrey is going to interview Alan; you should talk to him about IATH. It would've been nothing without their insight of where computing would be, helping the humanists craft how they were going to do things; not *what* they did, because it's their own field, but how they were going to use computation and what made computational sense, and what did not.

Wulf: Let me just give you an example. One of the founding members of IATH from the college was a guy by the name of Ed Ayers. Ed was a historian focused on the Civil War, and he built this archive of the Civil War.

Jones: It's called the Valley of the Shadow.

Wulf: That's right, I had forgotten that. And, you know, that just changed the way that a set of historians did their research because all of a sudden, there was this place where you could go and wander around virtually, and you could make incremental additions to it, and amplify things that were important, and that was the users could do that. Ed, by the way, now is president of . . .

Jones: University of Richmond.

Wulf: University of Richmond, yes, and I don't know whether he's continuing in this field or not, but we just had people like that at IATH.

Jones: So computer science, in effect, through some individual actions, strongly affected liberal arts. And of course, there were lots of people that were self-driven that did things over there in the computational realm.

Yost: Did most faculty in the liberal arts embrace this or was there some meaningful resistance?

Wulf: I don't know whether there was any resistance but it certainly is a subset of people that vigorously embraced it. The story of doing history scholarship for a few hundred years, and there's a lot of inertia there in how you do that, which you ought to know, right? [Laughs.]

Jones: So it wasn't resistance so much as I don't want to be part of that because I'm doing other things.

Wulf: And I'm doing these other things the way that all of my colleagues are doing it and I'm evaluated by my colleagues.

Jones: So when we did this workshop with the blue cover report on digital humanities — that's just something that I got money from NSF, we did it here at Virginia. By 2006, there were a bunch of places — Bill was off at NAE so I was just doing this solo — of organizations in different universities like IATH but they had come along. And so we had invited half a dozen of their directors, a number of which served on my organizing committee. And the humanists there reported that roughly six percent of humanists really adopt information technology in a cutting edge way. I mean, everybody used text editors, but that's not what we mean by that, but really use serious computation in their scholarship. Still in 2006, it was a very small percent, much smaller that I expected by that time, yet it was such a small percent. You talk to historians and suggest that they build a simulation with all their premises in it and then perform hypothetical what-ifs.

You run smack into a brick wall of their culture because they don't do that, and they certainly don't do prediction, and so it's going to take decades for them to figure out how to formulate that so that they can make use of that technology. They will, but I don't know if that will happen in my lifetime.

Wulf: And there is probably a growing number of humanists that use computation for scholarship.

Yost: Digital humanities has grown steadily as a field, I think there's more and more faculty.

Wulf: Yay!
[Laughter.]

Yost: I was very aware of University of Virginia as being a leader in that area and I didn't realize that it was [interrupted]

Wulf: Oh, yes; I'm at fault.

Yost: . . . your contribution.

Jones: When the leaders of IATH became humanists, they did not have the key insight on which IATH was started. I mean, a key premise was that what you have to give a humanist is not what is cost effective computation right now, you have to give them what will be cost effective, and that cost, you want 10 times whatever computation they would have on their desk today, and some software developers, because they need to develop software and graphic user interfaces. So for example, Jerry McGann, the historian that built the hyperlink edition, he won a McArthur Award, essentially I think for that work. And where he did he apply that award money? He hired software helpers; or did, last time I talked to him.

Yost: Can you tell me about the opportunity, and the context for the opportunity, for you to become the Director of the Defense Research and Engineering [DDR&E] for the DoD?

Jones: Let's see; as I said, led by Keith Uncapher, I had gotten much more interested in larger science and technology issues for the nation, and I think of it as service to the nation. Both Bill and I are strong believers that you need to give back. I wasn't going to join the military; so serving pro bono on science based advisory organizations is my way of giving back. Since then, I always have been doing some kind of *pro bono* advisory work, always trying to be where the interesting problems are. We can talk about some of those things later. I had no interest or thought about going into the government. By that time I was a trustee of the MITRE Corporation. I had spent many years being on the MIT Lincoln Labs Advisory Board, I had rotated off that by 1993. I was a member of the Defense Advisory Board. 1989-1991when I was on the advisory board for Lincoln Labs, so I was off that but I was a trustee of MITRE. I was on the board of a for-profit company that did a lot of work for the government, primarily the national security community, called Science Applications International [SAIC], and it turns out that Bill Perry was on the board and I think John Deutch was on the board of SAIC as well. Either Bill or John was also a trustee of MITRE. When Bill Clinton was elected, Bill Perry went in as Deputy Secretary, and John Deutch went in as his direct report, Under Secretary for Acquisition. And one day, they gave me a call and they said we'd like you to come and be the Director of Defense Research and Engineering, which is in the office of the Secretary, not in one of the three military departments, and reports to the Under Secretary. The DDR&E is the most senior person in the Office of the Secretary [of Defense] who is responsible for strictly science and technology.

Wulf: Jeff, if you're a general officer, you get a flag for you and you'll notice this [indicating a miniature flag of office that Anita has in her home office] has four stars. So she's the military rank equivalent of a four star general, and this is her flag.

Jones: The position has the flag. There are a lot of positions where the person who goes in it has a rank; and there are civilians who go in and they have a rank. One of the things that served me extremely well at the Pentagon, going in as a woman — I wasn't the only woman, there were other women — although John Deutch will tell you that I was very worried about that. Everybody, especially the military, everybody knows everybody else's rank and four stars is as high as it gets, except in time of war so it facilitated doing some things. Anyway, so they asked me [interrupted] …

Wulf: Everybody would stand up in the room when she came in.

Jones: [Laughing.] If they had three stars or less, yes. DoD is a place where people attend to protocol and people are very respectful of rank. Anyway, I had no intent of doing anything like going into government. Another one of my friends, actually, Caz Zraket, who is at that point was the president of MITRE, said, 'Do you want me to send your resume in for a political appointment?' I replied, 'No, no, no, no; there's nothing, absolutely nothing they could offer me that I would want to do.' And then Bill and John called me up for the DDR&E position. Never in the world would I think that they would ask me to do that, but the reason I did it was because of Bill Perry and John Deutch. I revere and respect both of them. Smart, talented, dedicated to the country and what they said is, 'We are going to defense and we need you to come with us.' So it's like those offers you can't refuse; I had to do it because they had asked. So I went into the job. Now, I will say that it is the best job in the government if you want to make a difference with science and technology; there is no other job that is better because you have many organizations that report to you. What I had responsibility for budget-wise was science and technology in OSD only. The money in the three military departments — I'll tell you a story later — but those are in the departments and I was not in a department. But for example, the whole DARPA budget was part of my budget.

Yost: So like the Naval Research Lab [NRL]?

Jones: That was in the Navy budget, but each of the three military departments had research offices and I was very strongly responsible for protecting them if there was a way that I could do that, and challenging them to do better. The Navy was quite special because they had the ONR, the Office of Naval Research; but also, it was the home of the best research laboratory in the entire country – the Naval Research Laboratory. There was a survey done by somebody in the mid-1990s, where what they looked at physical sciences (excluding biology) and they asked the question, when someone filed a technical patent, what articles did they cite? Among the top 10 organizations cited, there was, as I recall, only one government research lab and that was the Naval Research Lab. None of the DOE labs were on that list. Lincoln Labs was on that list so it was one of the DoD FFRDCs [Federally Funded Research and Development Center]. So, in my portfolio were some of the best performing research organizations in the country.

I was responsible for the FFRDCs because they fell under the technical umbrella. That became problematical because Senator Ted Stephens was trying to kill them all. At that time, Paul Kaminski had replaced John Deutch as the Under Secretary, and both Paul and I had a lot of scars from defending the DoD FFRDCs. We did so because the FFRDCs are very valuable organizations in the context of how the DoD does science and technology, and we paid in blood to protect them and we didn't allow them to be killed. Had we not done that, they would not exist today, I firmly believe. So that's some of the kinds of things you're responsible for in that job; it's not that you invent this widget; you want to ensure that the performers can do the best possible job that they can.

Yost: Was there anything you pushed for change with DARPA?

Jones: In my office we mainly worked on strategy, developing new capabilities for the military. DARPA would have multiple programs in any given strategic area. DARPA, ahead of other agencies in DoD is responsible for assuring that there are no technical surprises for the military. So, I would work with the DARPA directors on strategic thrusts. Also, I'm not a micro manager, and there was only one time that I directed a change in a specific program. It was in the area of MEMS, Micro Electrical Mechanical Systems, and I told the DARPA director, who was Gary Denman at the time, to double the budget. The then-current budget was $30 million. I had reasons for doing that and it turns out that all of a sudden I was the best friend of the program director who ran that, whose name was Ken Gabriel, who is particularly talented — he's currently the director of Draper Laboratories — and that's the only time I directed a budget change in a limited project. We can talk about what are constraints as such, but DARPA is a most-effective funding organization for science and technology. It is mission-driven; it can act very rapidly; it cancels programs in a way that a curiosity-driven organization like NSF does not. DARPA has a number of restrictions on it; it does very little basic research, even though university people think it does. Most of the funds for university research are not in the 6.1research account, it actually comes from later accounts, and program officers at DARPA think more in terms of potential military capability than do the university researchers. And DARPA was running well under both directors that were there while I was there. So my issues were more linking to the Warfighters and making sure we were solving problems that were pertinent to the Warfighters. And I worked as closely with the Services. The DoD leaders/managers of science and technology met routinely in something called the breakfast club, which I inherited — actually, those meetings were a great idea. So every Tuesday at 7:00 we'd have a breakfast meeting in my office. The DARPA director attended but most of the other attendees were from the services because the bulk of the science and technology money is in the Services. Perhaps a third of it was in the office of the Secretary of Defense and a quarter of that was actually in my office. The other money — outside of OSD — was/is actually crucial because it was invested by the military departments directly against — they were a bit more bureaucratic — but directly against the mission and the problems that their leaders had, and that was important.

Academics look at DARPA and they see the academics run through as rotators, but there are typically as many military officers from the four services who do a rotation into DARPA. Their presence, and their many projects, are focused on military needs. Most of the science and technology funds are in the military department and each department has organizations to address science and technology. There's no marine office of marine research; that is covered by ONR, the Office of Naval Research. One of my objectives in working with the departments was to strengthen the quality of their technical people. There is a core of senior executive services people who are very senior civilians all across the military departments; some are leading technologist in the science and technology program. But they're also elsewhere, and they are critical to the way the department works. And I actually started getting them together, making sure they knew each other, and to highlight how important they were, at least in OSDs view. OSD is the Office of the Secretary of Defense.

And then – in addition to DARPA – I had some initiatives also, that I fund directly out of the DDR&E office. One initiative was in supercomputing. The objective of the OSD supercomputing program was to put very high performance computing in each of the three military departments. I sought to ensure they were closely linked and doing computation on behalf of the science and technology program. But for example, the Navy's machine was with the naval oceanographer and so was doing production research for the Navy – in this case marine weather prediction. The Navy pays for the oceanography and weather prediction in fuel that's saved by steaming around storms and staying out of places where the ships would founder. So supercomputing was one initiative. I created the Multidisciplinary University Research Initiative, the MURI program, which still exists today, and the funds at that point came out of my budget but I would fund projects selected and managed by the services, the ONRs and such. Those funds augmented their budget. I would approve the topics but then they would just compete and manage the projects. DARPA got annoyed because they couldn't control the topics, so in the end, I just routed the funds to the Services. Okay, you don't want to play by my baseball rules; and that's just fine. The Services have done a very good job over the years making wise investments and getting good value from the MURI projects. These were fairly large grants for OXRs [the offices of research of the three Services] and so it was a useful addition to Service portfolios. Let's see; I did a lot with the Warfighters, but that's less what I think Babbage would be interested in.

One of the key issues that the **DDR&E**, which was a hallowed acronym at that time, was responsible for was oversight of the entire DoD S&T budget, and that's crisply defined as all money is in some accounts, all accounts are numbered; and 6.1, 6.2, and 6.3 are basic research, development, advanced development, something like that. Anyway, the military departments and OSD, which include DARPA, have money in all three accounts. The problem was that, if you remember Clinton in the first term balanced the budget. That balanced budget was achieved by doing two things: increasing tax on the wealthy, and a reduction — which had started long before Clinton's election — a reduction of 40 percent of the DoD. So I arrived at the Pentagon in the last several years of 10 years of 40 percent decline of the DoD. Now, it is not too difficult to deal with a 10 percent budget reduction, but a 40 percent budget reduction requires making very serious changes.

Wulf: Changing what you're doing.

Jones: And, of course, the Services had not reduced core structure, which means reduce the size of core right off the bat. So, at that time, they had literally cut way deep into development of new programs. They were cutting core structure, but they weren't buying new systems because there was no budget for it. I was there four-and-a-half years and a new budget is prepared every year. In two of the years, one service and one other year two services proposed to cut their science and technology budget by 30 percent. Now, I didn't have any trouble protecting the OSD budget, particularly with Bill Perry as SecDef. I'm going to say nobody made inroads, but the budget was not harmed. But the military departments were in great need, genuinely, of new materiel and by-and-large the military leaders are operators, okay? They are not out of the science and technology community and so they would come and they would say, 'Anita, we're going to cut our S&T budget by 30 percent this year but we'll put it back next year.'

Wulf: Baloney.

Jones: Well, even if they could — and the reversal of the budget cuts would not have happened — even if they could, you have graduate students who are being funded and all of a sudden you're going to reduce by 30 percent? I mean, if you applied the reduction across all OXR projects, consider how that would affect research support for graduate students, you're going to cut that by 30 percent! And all of a sudden in one year oh, we can't fund you this year but we'll fund you next year. S&T projects typically take many years, and they must have stability of funding. Actually stability of funding is more important than the level of funding. It is better for it to go down somewhat, but remain stable. Two things, then I'll end this story; first of all, I told you that in the Pentagon everybody respects rank. Early on I took some actions that my mentors told me were wise: it's useful for people to see that a manager is decisive, and takes hard actions. It makes people less inclined to mess with you, to challenge you; and so I thought this is good advice so I did some things. Actually, there was a cartoon that was circulated after my first year, and it was two guys outside the door — it says DDR&E on it — the men are obviously waiting to go in and see the DDR&E, and one guy says to the other, 'If she jiggles her earrings, you're dead.'

[Laughter.]

I thought this was great, but that's just a joke. Yet ….. The other thing is the budget process in the Department of Defense is quite disciplined and well defined. The services prepare their budget and they do not submit it to OSD until August, but they have to turn it in in August. The secretary turns the budget into the president in January. In that intervening time, it is possible for certain office-holders to raise issues, sort of "by right" based on their office, and as part of the process. And the DDR&E office had the right to raise an issue, not just with its own (that is inside OSD) budget, which as I say, nobody really messed with. So the first year it was the Air Force that submitted a Service budget that dramatically reduced their S&T. I considered it, and for me, this was a resignation level issue. And so, I was going to fight back with everything I had because, I mean, first of all, you're talking about over a billion dollar disruption; I don't know what the amount was, but the S&T budget lines were being reduced by one-third. And again, the budgets were coming down so you could tolerate a few percent cut, which was acceptable in the overall budget circumstances. But a 30 percent reduction would be very disruptive to the Air Force scientific program so it just was the wrong thing to do, and it was compromising their future. Coming out of this arid time when they weren't building things, the first thing they're going to need is developed technology that they could put in the systems when they started building them again. S&T advancement was crucial. So I passionately believed that the AF was making a mistake in their own terms. So on budget issues, the way they're resolved is you are sitting at a table and at the end of the table is either the Secretary or the Deputy, and the Vice Chairman of the Joint Chiefs, and they're going to decide. Now, this in the scheme of the DoD, I understand we're talking probably about 300 or 400 million dollars, but this is a small issue compared to the overall DoD issue. We had a total of 15 minutes. I was sitting across from the Chief of the Air Force and the Secretary of the Air Force. Because of protocol, I made my argument first and what I chose to do is to give all the reasons why the Air Force was desperate to get money for other things and the fact that they really needed those funds. And then I spent about the last minute explaining why S&T was absolutely the wrong place to get it. The next speaker was the Chief, who is a fine military officer named Ron Fogleman, and the first thing out of his mouth was the phrase *mea culpa*. I had won. And so yes, there were some small reductions but not the 30 percent reduction. Part of the DDR&E job is not doing the science and technology; it's protecting the budget so the brilliant people could do it. I would also testify often with Congress, particularly to the Armed Services Committees in the House and the Senate. Members were quite supportive of science and technology, and sometimes they would put back budget that I (well, really S&T) had lost inside the building. So the Services really played "hard ball", but it is a disciplined process and there are group decision processes. I lost some battles but I never lost a 30 percent cut to any military department, even though they tried to make such cuts several times. I think I would have resigned if they had succeeded because that was, for me, that was what the stakes were.

There were some interesting things from a computer science point of view that occurred during my tenure. Do you know SEMATECH? Okay, the history was the U.S. had gone below 30 percent of the market in microelectronics and the industry came to the government; industry joined together and created SEMATECH. The government put roughly $100 million a year into early non-competitive research via SEMATECH. It was during my political appointment — Bill Perry's administration that SEMATECH, led by Craig Barrett, who's been the CEO of Intel and really the leader of the microelectronics industry at that point, said, Thank you very much, we liked your money but we can take it from here." And so surprisingly, the people that became scared to death were actually the university researchers in microelectronics at Berkeley, at CalTech, at some other places who got a fair amount of funding both through DARPA and SEMATECH. They were quite concerned that SEMATECH would fund only very late stage work – i.e. not research. And that's not what they wanted to do; that's not what they should be doing. Paul Kaminski, my boss, and I, made a deal with Craig and the SEMATECH people. Who was head of SEMATECH at the time? Bill Spencer. He was instrumental in making it a really advanced development type of place where non-competitive technology could jointly be developed.

So we created a program in DARPA, where DARPA would fund one dollar for every two dollars, as I recall, that SEMATECH or the Semiconductor Research Associates, or whatever it was, would fund. And the deal was that for the large general research areas, the academics would choose and I would make sure that that happened. Because DARPA's money went in, it was conditional on that and that was — I'd forgotten all about this — the compromise that we struck to protect the academic researchers who were really deeply concerned that SEMATECH would attempt to channel them into late stage development work. I remember going to them and saying, 'Look, this is the deal so tell me what topics; tell me two or three sort of general umbrella topics you want to pursue.' Those choice will be a DARPA program, so DARPA program officers will monitor what was done, so it would be reasonably overseen. And they picked I think it was three topics, and I don't remember now what they were, but two of them were ones that SEMATECH wanted to do as well so they just needed a little bit of assurance that there would be government oversight and enough government money in [it] that the government could ensure that their equities would be protected. So this is the kind of thing that the DDR&E did, among other things. It was part of protecting the research and advanced development enterprise in the U.S. The S&T enterprise broadly serves to advance the U.S. economy as well as serving national security. That's what that job was all about. Only a portion related to computer science; it was radar, it was missiles, it was propulsion, it was submarines, it was firefighting in submarines, it was hydraulic systems in helicopters — once the hydraulics is shot out, how long can the helicopter fly? When does it fall out of the air? So I learned there's no job that's better because you're involved with all these challenges, and you're involved with the smartest people on the planet working on them. And so I learned a huge amount both from the military and from the scientists and engineers funded by the DoD S&T program. I had a modest staff, but with experts; in some cases, first rate people and they would educate me on whatever the topic was. People ask was it a fun place to be? It was not a fun place to be, I mean, you have budgets that size [and] people play dirty tricks. They try to go around you and they want their objectives to be served rather than your objectives. But it was hugely rewarding, immensely educating, and I really felt I was giving back. Bill and I both believe that when you've been given opportunities —

Wulf: You can see why we have this enduring interest in this interface between science and public policy. We've been there fighting the fight.

Yost: Extremely important fight.

Wulf: You see the strengths of both sides. I mean, it's not as though there are good guys and bad guys, there are good guys and good guys. And what you need is balance.

Jones: But sometimes you try to kill something that's stupid and even logical arguments didn't work sometimes. That part was not good.

Yost: I realize computer science is a relatively small piece and computer security or cybersecurity an even much smaller piece, but in those years from 1993 to 1997, were there any developments with regard to infrastructure funding levels, initiatives with computer security that you recall?

Jones: I'll make a couple of observations. First of all, I did not get involved in the DARPA Information Processing Technologies Office. I don't like micromanagement and I had a bunch of interests; there was an exercise I was involved in, which is not quite security. Duane Adams, who was the then-deputy, put together an advisory group to explore what research DARPA should fund in software development. And basically the group did not come up with new ideas. It was very disappointing. But this exercise was not focused on security. There were programs in security; research was progressing. I do not remember any sort of great breakthroughs. This would have been at the time when RSA and private key encryption was really maturing, so I guess you would call that certainly a breakthrough. The work had been done earlier, starting with the Diffie-Hellman. But RSA really made public/private key use real; that was great. There was an important thing that happened that was not research that Bill was deeply in involved in, and that is that the Department of Defense. This is a time when the NSA is trying to maintain control of cryptography and they had developed BLACKER. You remember BLACKER? And they were working hand in glove with NIST who was cooperating, doing standards specifications, but NIST was cooperating pretty closely with NSA, at least as viewed from the outside. And NSA wanted control and there were some people who proposed that NSA to provide crypto chips to maintain that kind of control. I thought that was a very bad mistake. So I think it was Emmett Paige, the Assistant Secretary most concerned with such topics who proposed that the National Academies study the issue. I don't know who were the progenitors of the National Academy's cryptography study — but the cryptography study was done while I was at the Pentagon, and you were a member of it.

Wulf: Yes.

Jones: There were many issues swirling around in cryptography. In particular, the NSA was saying we know things about breaking it, and there was all the angst around export control, which was overly strict, and not moving with the technology in terms of thresholds. And industry was very concerned because they had to start deploying cryptography and they wanted to implement it in hardware. They did not want a foreign chip provided by the government, thank you very much. But they wanted to build a single hardware system and sell it internationally as well as nationally, because international sales were now a large market for the microelectronics and the computer manufacturers. These issues were crucial to market decisions and so DoD, who was not prime consumer — and by the way, NSA to this day has to certify all computer chips that go on satellites and schedules are delayed until they jolly well say it's ready to fly — so command and control, which was run by General Emmett Paige at that time, a good friend of ours, funded the academies to do just a classic academy study. Once a funder has agreed with the terms of work, that the funder of that study is in no way involved. So DoD was not involved but they were the customer, and they were the ones that paid for it but they were not the main audience. The main audience was industry, government, NSA, and the intelligence community in particular. And Bill was a member of that study. We have a copy of it somewhere. It is a very important study because a subset of the members of this study were cleared — and I think that includes Bill, but maybe I'm not cleared to know [laughs] — and they saw anything that was relevant to the questions being asked, which means NSA could not do what members of the intelligence community sometimes do, which is to say, "if you knew what we knew then you would be of a different opinion, but we can't tell you because you are not cleared to know." Well, this group was cleared to know everything relevant to the set of questions. Literally, the first sentence in the report is we're going to give you a set of recommendations and the committee is witting of all that is known by the U.S. government on these issues. And this just cleared the air. All of a sudden, BLACKER was not going to be what it was before — I'm not sure of the timing — and industry built their own cryptography chips. There were export controls on what could be sold outside the country, parameterized by key length. Industry still had some degree uncertainty about what can be broken and what can't be broken because that information remained very close hold. But it became clear in this time frame, that keys even for one-way, single key encryption, not public/private, which is vastly more expensive – if the key length is long enough, it's mighty hard to break. Exactly when it's broken, and how much time, there were some open contests at about this time where some messages were broken in a matter of a few months by supercomputers and really smart people. And then AES [Advanced Encryption Standard] had come along, another single key encryption system and with longer keys it.   I walk around believing that it is very hard to break. So people operationally knew what kind of security they could buy. They didn't necessarily know everything that NSA might know; I don't know everything NSA knows, I never did. But this cryptography study, it did what the academies when they are at their finest are doing. They take an issue, they frame it, they tell you what the issues are, and they make a recommendation of what the nation should do. And such a study set the nation along a reasonable path and by the way, the people who cared about this report were Emmet Paige, and John Deutch, and Bill Perry, and myself at the OSD level, and it was the right report at the right time. We went on to think about other things because this just cleared the national air.

Wulf: I don't know how to say this in a really constructive way, but let me just blunder through it. Anita talked to you about encryption and how hard it is to break keys; and then she used the word security. I think there is frequently a confusion or a mixing up of the notion of breaking codes and security. They're not exactly the same thing. Encryption is a technique that you can use for ensuring the security of certain things, but it's not exactly the same thing as the problem that capabilities are addressing. They're both addressing the security issues, but they're quite different from each other.

Jones: But the way you take a capability, you take it out of one protected environment and send it out into the real world, it's an encrypted thing. They're complementary.

Wulf: Yes.

Jones: It's the implementation. I've got to say this because it's so huge. In another area there was something that we did in DARPA during my tenure that has made a huge difference, and that is we built the first Predators aircraft -- drones And by the way, we did it without any cooperation of the Air Force, because the Air Force at that time had a culture that insisted that pilots flew inside planes. So DARPA is really at its best in building prototypes with partners. This was when Larry Lynn was director.. But once a prototype is build, one asks how to going to evaluate its military worth or non-worth. So we sent a few Predators — the Air Force didn't want them — we sent them to General Cash in Bosnia, which was the hottest theater at the time, and his evaluation was these are really useful. They were just used for surveillance and reconnaissance. Indeed, they were so useful that when we asked for them back to upgrade them, for example, to print a lat/long on every frame that was recorded because that hadn't been done initially, as I recall. He said, 'No, I'm the commander in the theater, these are useful to me, I'm keeping them.' So DARPA just built some more and what happened is that very soon some enlightened Air Force people said, 'Look, the air domain is our domain and these things are useful. The ground forces are saying so. If we don't adopt these, then the ground forces are going to come into our domain and that was worse than changing the culture.' So six, nine months later, particularly with some leadership from General John Jumper, the Air Force, at least portions of the Air Force said we're going to adopt drones. Not many years later, the Air Force and I think still today, they train more people to be pilots of unmanned aircraft — they use the word "man" and so I can't fight that, it's the structure — than they do pilots who fly in the cockpits of the sum total of all the planes that they have. Now, I will point out that they train those people to be pilots whereas the Army does not; they just teach people how to use the controls, they do not train them as pilots. So within the military culture differences remain. Predators were a major influence and it really was an example of doing something dramatic in the science and technology program. These were not the first drones to be built, I mean there were drones in Viet Nam. If you think about it, a missile is a drone. It happens to move very fast and you don't use it for surveillance, normally. Drones were not new but it was sort of the right prototyping at the right time and has made a huge difference.

Wulf: They say it's game changing.

Jones: Absolutely. I mean, there are actions being taken, especially against the terrorist elements in the Middle East, where no U.S. forces are put at risk and so it has saved huge numbers of lives. Huge numbers of lives.

Yost: In 1997, you returned to UVA and you're made a University Professor. Looking at your publications in the succeeding decade they're increasingly policy focused. Can you tell me about both your research and publication interests, and work, as well as your teaching, the courses that you taught when you returned?

Jones: I came back and I taught sources like Date Representation, and then Cybersecurity. I have an enduring interest in security. One of the pleasures of teaching a course is that first of all you review all the fundamentals, and then you also learn what's new and so teaching a course is a great way to keep yourself up to date. We should still be doing that.

Wulf: Yes we should.

Jones: But yes, I really was interested in science and technology policy. Typically, it involves much more that developing new ideas; it is necessary to consider how to protect and advance the cadre of smart people, the infrastructure and instrumentation and such, so that the U.S. can be a leader. I strongly believe that you have to protect it. So in 1999, I was appointed by the President to the National Science Board, which is the Board of Directors for the National Science Foundation, so it was quite dovetailed into thinking about S&T policy. It's that really where my interests were. I started teaching science and technology policy at that point because as a university professor, one of things you can do is teach anywhere in the University you want. For example, Bill could teach his Responsible Citizenship in a Technological Democracy course to freshman over in the college, and so I taught the Science and Technology Policy courses in a different department, Science, Technology and Society, where the history of science work is done. UVA is unusual in having a department in that field in the School of Engineering. I started doing a number of things outside so I was not doing classic computer science research, I was much more focused on policy.

Yost: A question that I asked Bill. You too have an amazing list of awards and honors. Are there ones that you want to talk specifically about, ones that have special meaning to you?

Jones: The awards that really matter are the ones that come from your peers because they are saying, 'Oh, you did a good job.' And so I particularly care about the Lovelace Award that I got from the Association of Women in Computing. And I particularly care about the Arthur M. Beuche Award, which is given by National Academies basically for contributions to the nation in science and technology policy. And likewise the award from the AAAS, the Abelson Award, which I received in 2009, which is likewise for contributions to the AAAS, the American Association for the Advancement of Science, a science and technology group headquartered in Washington. I think those are the three that come to mind because they are really peer awards.

Yost: Before we conclude, are there any topics I haven't brought up that you'd like to discuss?

Jones: I've done a number of things since then. They're not in security, but they're in science and technology. Let me just quickly run through them and tell you what they are and why I did them. I was a founding trustee of something called Science Foundation Ireland [SFI]. The Prime Minister in Ireland decided that if they invested in scientific research, and particularly engineering and advanced technology, they would draw industry to their shores. So Ireland created a SFI to do that and they operated similar to the National Science Foundation, all research funding awards are merit-based. Irish universities didn't like this very much. They just wanted thank-you-very-much block grants, like some of the European research organizations give. But SFI used Request For Proposals in the classic U.S. National Science Foundation mode, and it was highly effective. So I went to Ireland from time to time; SFI had enough funding to make a huge difference and not only for people in Ireland. The question you might ask is can investment in science and technology matter to the economy of a country? The answer is yes, and the demonstration of that is Ireland. Unfortunately, I had to leave and the reason I had to leave was because the lawyers both at the National Science Foundation — I was still on the board there — and with the Defense Science Board — I was back on the board there — said that a person who was not paid but whose expenses was paid by a foreign government was a "foreign agent". And you cannot be both a foreign agent and what's called a "special government employee". Both members of the National Science Board and the members of the Defense Science Board are special government employees. And, so I had to resign. Now they have reinterpreted the rules and that would not be necessary. I mean, it's not as though Ireland had paid me for my time, they merely paid travel expenses.

Yost: And the Science Foundation of Arizona?

I was a founding trustee of the Science Foundation Arizona. This was similar in that it was a state funding science and technology. Governor Janet Napolitano created Science Foundation Arizona [SFAz], and they modeled it on Science Foundation Ireland. They hired Bill Harris, the man who had been the director of Science Foundation Ireland, to come to Arizona and invest Arizona taxpayer's money in research and development projects. Specifically, SFAz funded work in the bio area and the information technology area; merit reviewed, in fact reviewed by people outside the state; but to be conducted by organizations inside the state. There are three research-capable universities in Arizona. And I thought it was very enlightened for a state to make such investment. And so when I was asked to serve as a founding trustee, I did and I'm still a trustee. We do not, any longer with change in administrations, have state government funding, and SFAz is moving more to K-12 work, but it's an example of a nonprofit that is really investing in fundamental activities like K-12 education, and science, and engineering research that will make a difference in the economy. And there's actually evidence that Science Foundation Arizona is making a difference.

Wulf: Bill Harris was on the staff of the Academies when I was there. I don't know whether you knew him before?

Jones: I did not. I met him somewhere along the way before SFI.

Wulf: Anyway, so I knew Bill and thought a great deal of him, so when Anita told me about this I said, 'Yeah!'

Jones: I went back to the University in 1997 and I didn't retire until 2010, but I did some outside work for pay; like I remained on the board of Science Applications

International, then we went public and we split into two companies, I continued on the board of one of the two companies that did not take that name — it's called Leidos — because I believe that the government-serving industry, that that industry serves national security and the well-being of the nation. And so I do that as part of give-back. I went back on the Defense Science Board; they made me a fellow, which is very nice recognition also, I guess. I continue to be very active with *pro bono* advising, so I continue to do that. I was asked to join the MIT Corporation; it's a group of about 100 people who are technically the governing board of MIT, which is a private university, not a state university. I believe that the best of the research-capable universities are one of the crown jewels of the country, crucial to the economy, and the economy is crucial to both national security and the happiness of the citizens. So, I serve on organizations like the MIT Corporation. A year after I joined the Corporation they asked me to serve on the Corporation's executive committee, which is a dozen people, including the president, who are the trustees, if you will, of the university and the counterpart of the University of Virginia's Board of Visitors, which is politically appointed. The MIT executive committee is really extraordinarily good, and through this time, even though the economic downturn, when the economic bubble burst, the university was well managed by the executive committee, the president, and her executive team. So working on behalf of MIT is just another kind of giving back.

Wulf: A little bit of context. Massachusetts is unusual among the states in that they have a corporation that is a state institution so IBM or General Motors, those are all federal corporations. A small number of states, a few states, have state corporations; a law creating state corporations. MIT is one of these Massachusetts-only corporations.

Jones: But it is not funded by the state, just chartered to exist.

Wulf: Right.

Jones: I was on the board of the Draper Corporation. Draper was Doc Draper's instrument lab . . .

Wulf: At MIT.

Jones: . . . at MIT, yes, and spun out. Draper is a not-for-profit located in Technology Square, right where MIT has a lot of space. And it's really like the FFRDCs but it is not an FFRDC; Federally Funded Research and Development Corporation. Lincoln Labs, which is part of MIT, is, and so for example, I was one of the people on the executive committee that knew Lincoln extremely well, because I was responsible for them [the FFRDCs] when I was in the Pentagon, but before that I had been on the Lincoln Labs advisory committee for a number of years. That takes me to computer science; about eight years ago, a set of us were sort of concerned about the vibrancy and maybe a little lack thereof, in computer science research in general. So we created something called the Computing Community Consortium [CCC]. Are you familiar with that?

Yost: No I'm not.

Jones: We wrote a sole-source proposal to NSF to fund this, and what it does is effectively try to strengthen the computer science community. We needed to have the ability to account for the money that we received, so we made CCC a subsidiary of Computing Research Association because they did have the ability to account, they were an operating organization, and they care about the vibrancy of the computing community as well. CCC does a number of things [and] one thing it does is to support *ad hoc*, small conferences seeking new visions in computer science, any field related to computer science; so routinely there may be eight in a year. These are quite small, *ad hoc*; you just put together the right people. CCC is governed by a council of about 20 people and we review the proposals. It is peer review so the proposers get a boatload of written stuff and CCC make decisions within weeks, not months. So CCC is an agile, fast-acting organization. CCC started something called the Science and Technology Policy Institute, which is run once a year, that is open to sort of mid-level faculty. They come to Washington and get a day or two introduction to science and technology policy and to dealing with Washington institutions, NSF all the way up through Congress. Just sort of year in and year out CCC does that for 30, 40 people. That's just building infrastructure; a researcher has to be able to say who she is and what she does in order to get funded. She needs to be able to communicate with Congress. The field needs to have people coming in as rotators to DARPA, to NIH, to DOE Labs, or DOE corporate. And this sort of opens the aperture. This is the kind of thing that I needed when I first talked to Keith Uncapher about feeling like I'm kinda down in a rut. When you're midlevel career, sometimes you have a hankering to view the world more broadly and this Policy Institute provides for that specifically for computer science. Another thing CCC did was react to the downturn in government funding, the bursting of the economic bubble, and sudden reductions in university hiring of new faculty – across the board in science and engineering, but particularly in computer science. So you had PhD students graduating – something like 1800 Ph.D.s in computer science graduate each year – and most of them have aspirations rightly or wrongly, of being clones of their advisors because that's what advisors train them to want to do, which is another problem I'll come back to. So we said my gosh, all these people have to get a job and there are no faculty jobs, and that's where most people go. There actually were very few post-docs in CS at that point; this would be 2000; whenever the bubble burst came about.

Yost: I imagine industrial research, like IBM Research and HP Research, positions at such places declined substantially, too with the bubble burst.

Jones: I don't know that but I assume it did. I mean, the economy was compressed and so people were hunkering down in general, but particularly state funding of universities was really declining.

Yost: The University of Minnesota was hurt badly.

Wulf: Really a very negative time.

Jones: So the problem was, remember, CCC was concerned about vibrancy and the health of the computing research enterprise. Well, this pipeline of students coming out is the lifeblood and suddenly they don't have a place to go. So within a matter of a couple of months, CCC wrote a proposal to NSF, again sole source, saying we want to fund graduating PhDs and we're going to call it Computing Innovation Fellows. CCC wants to fund them just for two years, just to get past this economic downturn. This is an economic program so we will fund a few in industry, just a handful; we're going to fund in the university research enterprise, so these will effectively be post-doc fellows, but we're going to put this as an economic program so we will fund no more than two people coming out of any institution or going to any institution. Now, if it was strictly on merit, the top 10 universities would've just taken off the table some huge percentage of these fellowships because their people would out-compete the rest. Again, we're interested in the enterprise, all the universities were feeling this pain and this was an economic program. And so the Ph.D. candidates who would propose being a Computing Innovation Fellow for their next position could propose I think one, two, or three advisors; had to be from a different place than they were coming from, and the advisor had to write a statement. And so when we evaluated each proposal, we evaluated each pair [PhD and proposed advisor] so each Ph.D. candidate actually got up to three evaluations, and what was remarkable is the evaluation paired with this advisor might be much higher than the evaluation paired with that advisor, because this advisor was not "with it" in terms of advising a Ph.D. student and what one candidate advisor proposed was not as good as what another advisor proposed. Peter Lee, who was at that time at CMU led this crash effort; he was also a CCC council member, but a bunch of us were involved. I served on the selection committee. NSF funded the activity and they made their decision, I think, in a matter of months – very rapidly – so I think we started planning in the spring, and we completed selections and people were funded for the next September. It is unheard of to get anything funded by government this rapidly; I mean, DARPA is slow in comparison and NSF is . . .

Wulf: Glacial.

Jones: Glacial, as is DOE, and NIH. Anyway, creating Computing Innovation Fellows is the kind of thing CCC did and does. The first program we had 60 post-docs the first year, then we had 40 fellows. We funded them, by the way, at roughly 100k apiece, 75k for their stipend, I think 10k might've gone to the university, and the remainder went to them for travel and research expenses. That number is important because the average wage for somebody who's a post-doc in the biological sciences, who's been in a post-doc for five or six years is only about $40,000, and so we are also sending a message there.

Wulf: That's a real problem, by the way. Biological sciences have huge numbers of post-docs . . .

Jones: With no place to go.

Wulf: . . . and hired to be slave labor. A post-doc is supposed to be a training experience; this is how you become a faculty member and this is what you do. In biological science it is not.

Jones: So we had this program, 60 post-doc fellows selected the first year, 40 in the second year, 20 in the third year, and then we shut down the program. The reason is the economy was coming back. CCC did not want to be in the business of funding graduate students or post-docs; it was dealing with a people/infrastructure crisis that our community had, and to my knowledge, essentially no other group did anything similar to this. Well, based on this, I got all fired up on the issue of post-docs and I saw what Bill saw over in the biological sciences. And then looking at the numbers from the Taulbee Report, which is a very useful survey that the CRA [Computer Research Association] does, [I] noticed that the number of post-docs had doubled three times over in less than a decade. The increases dwarfed the Innovation Fellow program, so it was not responsible for this considerable growth of post-docs in computer science. And I personally knew of some pockets where post-docs were in effect long-lived labor just to run laboratories. These people were not being trained, they weren't being educated on where they could go other than academic enterprises and such. So I thought CCC should address this situation in computer science. With less than unanimous support from the council, we used some of the remaining innovation fellows money to do a couple things. One was to do an evaluation of how the fellows program worked, how it compared to other post-doc experiences for computer science people. In addition also funded an activity to figure out what are the best practices for particularly university in computer science and engineering supporting post-docs. So this is a subject I'm now passionate on. I've written an article in *CACM*. And Erwin Gianchandani, who at that point was the executive director of CCC and I just wrote over a few months a manual, which is posted on the CRA website describing best practices for supporting CS post-docs. We just took the best of what was known, borrowing heavily from the National Academies report on this subject.

Wulf: Focusing on the educational aspect of being a post-doc . . .

Jones: And career preparation.

Wulf: . . . and career preparation, okay, not just running a lab.

Jones: Advising and having a development plan. For example, in one university I know, there is a department that hires post-docs to teach. The department assigns teaching of courses at a half time level. The department assigns the post-doc some mentor or other that may or may not be related to what the post-doc do, and that's it: go off and teach. And that's the extent of that university's support. This is a very wrong way to develop the most critical resource of the research enterprise – the people. So CCC ran an RFP, three university consortiums were selected. One proposal was led by Arizona State University, and Science Foundation Arizona was involved so all of a sudden I couldn't be involved in the selection. They booted me out of the room, which was the right thing to do. Another awardee was University of Washington and the third was a New York City consortium, which also includes Cornell (which is starting a New York City campus). The three awardees are being funded for three years to consider and define the best practices for supporting CS post-docs. I am hopeful that they will all end up saying we need to provide selected supports. But, in addition the institution at large needs to support post-docs, helping with career advice, having career advisory programs. If you want to know how industry works it's not just computer science; so there are economies of scale to be had in providing that support. The awardees finished year one in about April.

I find issues that I think are important and I choose to sink my time into them, and sometimes more often than not, it's easier to do within the context of an organization; Science Foundation of Ireland, Science Foundation of Arizona, the National Science Board, the Defense Science Board and the Computing Community Consortium.

Yost: You are a trustee of InQTel, can you talk a bit about that?

Jones: Yes. — I'll talk about that in a minute — but CCC is one and this post-doc issue is one I've sort of taken on personally because the people infrastructure that underpins a research enterprise is the most valuable resource you've got and they must be treated well. You ought to challenge people to be the best they can be and our post-doc system does not always do that, but it can. And the reason why it's particularly important is if you look all across science and engineering — put biology aside because it's terribly broken; in some cases it involves exploitation, I mean, it's disgusting – but across the rest of the science and engineering enterprise, because we are now so experimental, the post-docs are now a critical part of the research enterprise and their numbers are increasing in most research universities. They've grown at the University of Washington. There are now two post-docs for every faculty member in science and engineering at MIT; they outnumber the faculty. They have grown hugely, again, where you have experimental laboratories and are doing prototyping, particularly that empowered by information technology. So I find things that I think are important and I just go out and work on them; whatever appeals to me.

So another — I'll end with this — another thing I'm doing right now that I find just most interesting, is I'm a trustee of something called InQTel. InQTel is funded by the intelligence community. It was started by the CIA but it is now funded by a number of different agencies, including DHS. InQTel is an impedance match between small, startup companies and the government. Small companies, I like to say, can't even find the front door of the intelligence community. They do not have the accounting infrastructure to take a government contract, so what InQTel does is it does strategic investing. InQTel finds small companies that have technology, particularly in a prototype widget, or object, or device, or service that might be of use by some of its customers inside the government. InQTel has extensive dialog with those customers so they can tell InQTel what they need and what problems they have or challenges they have. And InQTel tries to find things that would relate to solving their problems and invest in a way that it can end up with some equity. Some people describe InQTel as a venture capitalist, it's much closer to being a strategic investor. For example, friends started and ran an Intel investment capital venture and it's a way for the corporation, to put its fingers on the pulse of technology development. And that is what InQTel does; it finds these small companies and contracts with them, often to make a slight adaptation to what they have, whether it's intelligence analysis software, covert communication, or covert power sources – for example. InQTel has been quite successful, and one of the ways you can tell it's successful is the other intelligence agencies outside the CIA are now piggybacking on it, and putting money into it, but most importantly piloting the technologies that InQTel finds to serve whatever their mission is. So remember my base interest is science and technology that advances the nation. Well, bridging this chasm between small company technology and the government with its dreadful and ponderous acquisition processes is an important contribution. InQTel is maturing so that what it provides to its government customers is no longer technology of single-shot companies, but it's putting multiple offerings together and authoring architectures or frameworks that allow an agency to bring multiple technologies together or to tackle a problem that is not solved by the technology of a single new company. And interestingly enough, the agencies are saying that these architectures and frameworks may be even more valuable than connecting to a single company. InQTel staff is informed by the collection of investments in the companies and the architectures they design are likewise informed. Even when InQTel evaluates a new technology and does not invest, its staff is informed by the how a noticeable number of people are choosing to work as the technology evolves.

Wulf: You may have already said this, sorry, but I have this distinct memory of when InQTel was first founded and the quick description of it was that it was venture capital entity that was funded by the CIA, and that just seemed totally incongruous and people made a lot of fun of it; but in point of fact, I think it's worked out extremely well.

Jones: That is right. Congress scrutinizes InQTel and on any given day, some member of Congress could say that taxpayer money should not be used that way, I'm going to shut this baby down. InQTel has a set of trustees, people who have national respect; so when it started the trustees included people like Bill Perry, and Norm Augustine, Paul Kaminski, Jim Barksdale Ted Schlein, Jamie Miscik and Buzzy Krongard. Trustees have no interest in the stock equity so that the trustees have no financial interest in any of these companies succeeding. And indeed for InQTel, return on investment is simply a byproduct and never considered when the trustees do an evaluation of the CEO or how the firm is achieving its mission. What matters if what has been adopted by the agencies; what technologies they are experimenting with; what are they saying about the activities of InQTel? InQTel has gained some revenue because they had stock of a company and it went public and they sold it. But they just reinvest that in their activities and so it is working in service of the government's interest.

Wulf: Multiplicatively.

Jones: Multiplicatively, yes. Anyway, as I have said, I find issues that I think are important, but particularly I like to be associated with and work in situations where people are nurturing the infrastructure that underpins research and development in some way or other, in this country. That's what I do and I have a great time doing it, and I work with some of the brightest, most fun people on the planet.

Yost: All of that sounds like really incredibly important . . .

Jones: I believe it is.

Yost: . . . and useful work for our nation. Thank you so much, this has been fascinating.

Jones: It's great to be able to talk about yourself! It's really been fun.