

An Interview with

PAUL MORRIS

OH 467

Conducted by Jeffrey R. Yost

on

20 April 2011

National Science Foundation

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

Paul Morris Interview

20 April 2011

Oral History 467

Abstract

With support from the National Science Foundation (Grant No. 0811988, “Designing and Using FastLane: Distilling Lessons for Cyberinfrastructures”) CBI researchers Jeffrey Yost and Thomas Misa conducted oral history interviews with 70 NSF staff members as well as numerous additional interviews during 29 university site visits. An overview of the project is available at <www.cbi.umn.edu/oh/fastlane/> and a complete set of 643 publicly available interviews is at <dx.doi.org/10.13020/D6RG6B>. Here on the CBI oral history database is a selection of notable NSF staff including Joseph F. Burt, Jean Feldman, C. Suzanne Iacono, Constance McLindon, Carolyn L. Miller, Paul Morris, Andrea T. Norris, Erika Rissi, Craig Robinson, Mary F. Santonastasso, Rich Schneider, Frank P. Scioli, Beverly Sherman, George Strawn, and Frederic J. Wendling. Topics common to many of the interviews include the design and development of the NSF’s FastLane computer system, interactions with users, e-government initiatives, grants management practices, peer review, and NSF policies and practices. These interviews span a wide range of NSF staff, from program officers to senior managers.

Paul Morris was a visiting staff member who developed an innovative compliance checking software tool, which has been widely adopted across the agency.

Yost: My name is Jeffrey Yost from the University of Minnesota and I'm here this afternoon at NSF on April 20, 2011 with Paul Morris. Can you begin by telling me when you came to NSF, and what directorate and what program?

Morris: Yes, okay. Well I'm here as a federal temporary employee. I arrived two years ago, and that was to cover for another program officer, who was on a detail. But not to the astronomy division, where I am now; it was actually to the atmosphere and geospace sciences division in Geo Directorate, which is on the seventh floor. So I had one year down in one directorate, and I found myself moving to astronomy, where I am now, to manage the Arecibo Radio Telescope. So I'm a program officer concentrating on the management, operations and oversight of a large NSF facility, and that happens to be Arecibo Telescope, in my case.

Yost: Now was e-jacket being used for declines when you came on?

Morris: Yes, it was. And as a new program officer as part of your introduction, you do get some basic training on parts of e-jacket but pretty much, I just had to shadow other program officers for the first few weeks or months that I was here to get up to speed on it. So while there may well have organized, formal training, but I didn't do it that way. I seem to have just sort of picked it up, as is the case for lots of program officers, I'm sure.

Yost: Have you had interaction with the team in DIS that developed and works on different iterations of e-jacket?

Morris: I have because when I first arrived as a program officer I realized that certain of the things that the tool did or didn't do that I would've liked to have seen in a back office function. So I ended up developing my own sort of add-ons and additional features, which required me to interact with -- at least in the very early days -- let DIS know that I was developing my own tools. That was through courtesy, first and foremost. One thing that I did write was a web robot that interacted with e-jacket. E-jacket is typically a manual tool that we use interactively but I realized that some of the ways that I could make it better is to alter some of the instruction sequences. If I need to process 1,000 proposals I didn't have to go through all the 1,000 manually with a mouse. The web robot can extract out bits of the proposals for me automatically. Obviously, that had a small impact, potentially, on the loading of the e-jacket. So I had to just contact DIS and let them know that I was providing an automated front end to their tool. So yes, that established quite a good relationship with everybody downstairs.

Yost: And were they supportive of this [development effort]?

Morris: Remarkably so, actually. I was sort of expecting them to; because, of course, we have to sort of agree. If you've ever seen the login screen to e-jacket, there's plenty of quite scary information of how you should not be developing certain tools to drive it in ways that it was never intended to. And of course, that's exactly what I was doing. I was providing a web robot. But what I deliberately did was slow down the web robot interaction with EJ to the point where it was invisible to their loading. The web server,

the data base servers that they use couldn't even tell when I was doing it. So I slowed down each page impression, each page interaction, to maybe once a second. And so it was indistinguishable [from] a dedicated program officer clicking away for a few hours at a time. And once they realized there was absolutely no loading issue for me building this web robot, they were perfectly happy to let me run it. And it's been quite useful. If we have got a long list of proposals, e-jacket, as you know, can be slow at times and therefore, it's much better to let the robots go to work. It can only go at the pace of e-jacket, but at least you're not the one waiting for the page to load. You can just do other work, exactly. So, sometimes bringing up a project description, or a project summary, or budget page in e-jacket at certain times of the day can be a little sluggish. So it's crazy for me to be sitting there with my arms crossed waiting for that page to load when the robot can do all that work for me.

Yost: You've certainly gone into it in depth; but can you go through all the functionality of the tool?

Morris: I wrote it in a language called [AutoIt], which is a Windows scripting language approved for use here at NSF, and it's used quite extensively throughout. Many companies use this Auto It language as an alternative to Visual BASIC. So it's a language that's free; it's open source; and it's designed specifically to automate Windows functions. And of course, that's exactly what I wanted to do. I needed some way of hooking into the Internet Explorer -- the browser -- I needed some way of controlling the browser automatically via a scripting tool. AutoIt did the job extremely well. Typically, the way I

wrote the tool is that you give the tool a list of proposals -- it could be one, it could be 10, it could be 100 -- and it automatically then logs in to the e-jacket system, it extracts out all the relevant parts of the proposal as separate PDFs. It converts those PDFs to text; and once those PDFs are in text form, we can either search on them or use regular expressions. There's another programming tool called Regular Expressions that will allow us to do pattern matching within a text document. So the PDFs themselves, as you know, Jeff, are typically the currency we use here. It is actually what proposals are stored in, but it's not a very useful format. Typically, program officers are using Adobe Reader to do searching within a PDF. To take that PDF, convert to xml or text, and use that text version of the PDF to do any matching you want done; I mean you can hunt down PI names; you can check for conflicts; you can check the budget pages to make sure that the amounts on the cover page match the amounts elsewhere in the proposal; all stuff that was being done manually. I realized that could be done in an automated fashion. Even things like checking that the fonts are the right size; checking that the margins are correct in the PDF; all stuff that's perfectly doable with a ruler and program officer, but it's not very efficient to do it that way. And the GPG, the guidelines that we ask PIs to conform to when they submit a proposal is quite extensive. And I coded it as 150 separate rules into my engine. And those rules are checks. To [do] them manually can take up to an hour per proposal, and that was typically the case that we would have program assistants helping us to check the proposals. They came in and it could easily take an hour per proposal. My tool does it in under a minute so it's much faster and, of course, it works 24 hours a day.

Yost: Then is there a report that's generated?

Morris: Yes. I can obviously show you these reports. I know we're being recorded now. I mean if you wanted me to show you an example and as you are a PI, we can even compliance check one of your own proposals and you can sort of see the output. What they get back in return is an e-mail message. They submit compliance checks via an e-mail and they e-mail a proposal at nsf.gov, which is an internal e-mail address. My program interacts with Outlook; it takes the e-mail, strips out the proposal numbers in that e-mail, then it goes to work on that list of proposal numbers. And then in return, the program officer gets an e-mail. In the body of that e-mail you get a list of warnings that major compliance warnings are embedded in the body of the text, but they also get this little Excel spreadsheet with all of the gory detail about how compliant or not that proposal was, as a spreadsheet that is color coded so they can see instantly at a glance where the proposal has deviated slightly from the rules and then they can make the decision whether the PI is contacted to request revisions, or whether it's returned without review. I mean it's a sliding scale. If it's a small issue, then almost certainly a revision is sufficient but if it's a glaring error; if they've got more pages in the project description than they should have done; many divisions and directorates here at NSF are pretty tough about that sort of thing and I can sort of understand why. If you've got large solicitations with thousands of proposals coming in, and with a success rate of no more than 10 or 20 percent, you have to be tough. And so my compliance checker allows large solicitations; it can process up to 1,000 proposals a day. Do you know how many proposals we get specifically within NSF a year?

Yost: Is it around 40,000?

Morris: Yeah, I kept quoting 40,000. I think it's gone up. Obviously, it's gone up.

Yost: Evidently it went up in the years there was...

Morris: During the stimulus. But even since then, Jeff, we've seen a steady increase in the number of proposals coming in and obviously, the number of staff here in the building are not going up. So I think it's over 50,000 a year now. So if this automated checker can do 1,000 a day you can certainly see how it's keeping up with the load. Now, it doesn't replace a pair of eyes; it certainly doesn't replace a manual check; but it definitely does the triaging. It does a lot of the checks that a computer can do, freeing up the program officer and the program assistants to do the more important stuff such as read the project descriptions, read the project summaries, assess the proposal for intellectual merits and broader impacts. So all the stuff that the computer could never do is now; and the staff have more time to do the clever stuff, and I do all the automated stuff, the automated checking, which is done pretty fast and lightens the load considerably. So, yeah, but a thousand a day. And when the program officer or program assistant gets those thousand proposals back, they can be merged into a master spreadsheet, then you can filter out the warnings, you can then make the decisions on whether you want to just ask for corrections or whether it's such a howling error that they don't deserve to go forward for a full review. So those are the steps, anyway. I think

maybe 20 to 30 percent of proposals are being checked by my tool here within NSF.

Purely, I'm understanding, that it's just a prototype tool. It's not a requirement that people use it. It's actually hosted on Research.gov. I'm sure you're familiar with the Research.gov? Well, one of the sections within Research.gov that we see as program officers is a Research.gov labs environment; a bit like Google Labs where new tools and ideas are being tested and prototyped. So program officers get the option to; I mean again, I can show you on the screen later or now. Do you want to see it now or later?

Yost: Let's go through the interview.

Morris: Yes. Since program officers get a very nice web interface that they can use to submit their proposals for checking, using my software. DIS is running it in a production environment now, so they took my code and are running it on servers downstairs. In fact, we're now load balancing. We have two literally standard desktops running the tool. So we haven't used any complicated server environment; it's literally two old PCs: one running odd proposals, one running even proposal numbers. And those two boxes together give us this thousand a day throughput. As I say it's keeping up quite happily with the load. People don't get the compliance checks back straightway; typically, it might take a few minutes, or maybe tens of minutes to get proposals coming back. I wrote a queue that allows multiple program officers to all share the limited resource so if you submit a thousand proposals and then somebody else submits one, they're not stuck in the queue waiting for the thousand to go through from one of their colleagues. So the queuing theory I wrote to be fair and allow every program officer to have a fair crack at

the resource that's available on the servers. Typically, program officers or program assistant would submit a job of a few hundred proposals overnight, as the last thing that they do before they leave. Come back in the morning and they're all there in their inbox, ready to review.

Yost: How long did it take to program? What size is the program?

Morris: It's about 8,000 lines long, now; or 10,000 lines long. It's always a difficult answer because either you say it was no problem, it took me no time at all, which trivializes it slightly. Or you say I was up every evening working on it, which makes people wonder whether you're a very good programmer. So those are the extremes. (Laughs.) So, to answer your question, it's not my day job. My day job is to manage the facilities here in astronomy. So whenever I had half an hour to spare, or half an hour spare over lunch, I would write the tool. When I'm on an airplane or something; so it probably took longer than it should have done because I was never doing it full time. But 10,000 lines of code, it probably took a few months to get up and running, really. But not a few months of absolute time; 10 months of my occasional half hour of free time.

Yost: When did you complete it?

Morris: Because it's a prototype, it never felt completed. Features are being added all the time and particularly because different solicitations here at NSF have different rules. We don't seem to have one rule book. We have the GPG, as you know, which provides the

vanilla; the sort of what we expect of all program solicitations. But then again, different divisions have their own separate rules. So my program's constantly evolved to improve more and more solicitation specific checks. So it's never stood still and so every week or so I'll have a different version that I'll release. Now that it's in a production environment, those releases are much more controlled now. They have to be because it's on a production box. The releases might come out only once a month. But typically every week or so I might make a small change to the tool; and especially because my e-mail address was embedded in the replies and I used to say if anybody thinks what this checker is giving you is complete rubbish, let me know. So I got that constant feedback from Program Officers saying ah, hang on, you didn't quite do that right. So there were some tweaks to the code. But the working version was up and working within two months. But now it's a year old; I've been working on it for a whole year and it's changed remarkably from the early days and I've added more and more checks to it as it matures.

Yost: Thinking back to when you first conceived of developing it, was the goal to share it or was it something more just lightening your work load?

Morris: Well first of all, I did it because somebody in my division downstairs over coffee once with the other program officers, they all said oh, it's impossible to do stuff. I said that's a fair enough challenge. It's been talked about; it's not the first time somebody thought if only we could write some tools to do this. As far as I can tell, DIS has dabbled in it; odd information systems people have looked at this. But typically the problem is that the capability in-house isn't necessarily there to do this type of coding. It would

typically be outsourced to a software firm and it can be very expensive. And of course as your criteria, as your requirements change, the cost tends to go up, doesn't it? And the time it takes to get fixes down is pretty slow. And by the time they've priced in a service level agreement and getting Booz Allen [consulting firm] to write a code like this is a multi-million dollar operation. And you end up, typically, not getting what you wanted in the first place. Because I'm a Program Officer anyway, I started from the bottom; I coded from the bottom up. I actually wrote it because I had a burning need for some of this. And then I told some friends I was doing it; a few of my colleagues in my section started using it, and then everybody in my division started using it; and then other divisions. I got phone calls from people I'd never met saying I hear you've got a tool, could we try it out on our solicitation that's coming up at the end of the week? We've 2,000 proposals coming in this afternoon and it could take three weeks for them -- four weeks more, maybe -- to compliance check them all. Could we try your tool? By word of mouth, it got used and finally, DIS phoned up and said we've heard such great things, Paul, could we host it on our servers to allow everybody in the foundation to use it? I never intended; it was never designed to be used; I wrote it purely for my own use and it has grown in a fairly organic way. I don't think I would've ever written it the way I did if I had any idea how many people were going to use it. But it's proved extremely robust. It runs 24 hours a day, and it's running on free software; and actually, for the first six months, it was running on a box in a broom cupboard 'round the corner. We didn't even have it hosted on a proper IT infrastructure; we just had a four-year-old desktop, an old Windows desktop that I was running the code on. And it was literally, Jeff; you go around the corner by the elevators, there's a broom cupboard; and in that broom cupboard there was a computer

that just sat there for six months churning away. So you don't have to spend millions of dollars. I do like that open source model, and I do like a model where you can change the code within seconds. DIS can't operate that way; by their very nature they can't just be dabbling away with code. They have to provide up times and all of that resilience. I never had that issue. I was able to write prototype code; tell people that it's free, you don't have to use it but if it's any use to you, by all means. So people were very happy to be using it; they weren't expecting it to be shrink wrapped with a 1-800 number. (Laughs.) So it turned out to be quite an enjoyable activity because I got to meet the other people around the division and was happily solving problems that they thought were impossible to solve and making their lives easier in the process.

Yost: Can you speak a little bit about your educational background and where you picked up the programming skills?

Morris: Yes. Must be a sign of a misspent childhood, musn't it? I'm obviously not American; I'm originally from the United Kingdom. We moved to the U.S. five years ago and my wife actually works here at the National Science Foundation. We're both academics; we both did our Ph.D.'s together at Oxford University; whereas I left academia after five years of being a Fellow at the university to be a telecoms consultant. So I learned all of my software skills really from my placement industry. So I had a fabulous immersion in the world of wireless communications, and software development was part of what I did; so I picked up a lot of tools. I think my Ph.D. helped a little bit, as well. I mean I did a very numerical satellite remote sensing Ph.D. where I was familiar

with large data sets working with a lot of different languages. And then in the industry, just took that to the next level, really. So I've never seen myself as a software; I'd never sell myself as a software programmer. I suspect any bored teenager could do what I do in their bedroom. So the trick is not to advertise the fact that you write software, but I find it quite easy to do, and very rewarding. So that's how; I ended up; if you wait around for somebody else to do it, it will never happen. So I just sat down. The actual story is I was stuck in San Francisco at a conference, so that's why I started writing this code. There was a big snowstorm in DC two Christmases ago. I was stuck at a conference, not able to get home, and that's when I started committing. I wrote the majority of what I needed to do, stuck in my hotel room, because I thought this was more interesting than watching the TV. So it was a challenge in software; an algorithmic challenge to write what I did and it just started from there.

Yost: Is there a mechanism you've tried to get feedback from people?

Morris: Yes, and it's only as good as the feedback I get; that's absolutely true, Jeff. I think I've almost reached the point where I've done as much as I can as an individual, now. It really needs more developers and I think that's where we're going now. We're engaging some other software consultants; DIS [is] employing other software consultants to now look at my code. I've donated it to DIS so they can use it as they see fit. But what was the question?

Yost: Has there been any kind of systematic way you've tried to get feedback?

Morris: Okay. Other than I mentioned every time you get a reply back from the automated checker, my name, my e-mail address is embedded in the e-mail. So I've encouraged people to send me any [comments] good, bad, or indifferent about your experience. I had a lot of feedback that way. People stop me in the corridor, and it was quite a surprise to see that the way I had written it turned out to be useful for other people. It could be really hard to please everybody with this type of software because each division operates with a totally different set of rules. I don't know if you've noticed, but the consistency around NSF for what; for example; can I pick something at random?

Yost: Sure.

Morris: Like 'et.al.' in the references. So, if you write a proposal, you've done the same; I'm sure you have to list references you've used; and your biographical sketch, you have to list your publications. Now if you use the word et.al., and it's perfectly understandable why you might do that, some divisions will return that proposal without review if they see et.al.; they will be incredibly strict about that. Other divisions will perfectly happily let that type of thing slide through. So the GPG says you can't do it, but that's why it's been quite hard for me because I've written the [software] rules. And so what works for me may seem either overly harsh or not harsh enough for other divisions. So I've sort of struck a balance. I've sort of tweaked and tuned my tool to be most use[ful] to most divisions. And if anybody has a very specific requirement then I write specific code that handles only a particular program element or a division. If there are division-specific

requirements, my checker reads the cover sheet, finds out which division the proposal comes from, and therefore might operate in a slightly different prescription. And the feedback's been good. Typically, you only hear if it's broken, don't you? You never hear if people are enjoying using the codes. And what's nice is to see that the complaints have fallen away in the last month or so as I improve it, basically. People [are] writing back and saying it's just given me some complete rubbish here, Paul. And I typically feel terrible when that happens; I typically then fix it; and it's a constant process because as you know, the rules change all the time. The templates -- FastLane, E-jacket -- they're not fixed quantities at all. There's changes often made to the budget sheet, or the data management plans that are now a requirement. There's many more examples where that came from. It's not a static process, by any means.

Yost: Has there been any fundamental changes to FastLane or e-jacket that have made it necessary to [interrupted]

Morris: To break it, you mean?

Yost: To break it or just have to do significant modification?

Morris: No, thankfully. But there have been, historically. If we go back five years; proposals that were submitted five years ago do break my checker because the cover sheets and the PDF templates have all changed, but not in the last two years. I think E-jacket's about to have a major revision in June; I think I've seen e-mail. So a lot is going

to change in June. Most of it's cosmetic, as far as I can tell, but we have a non-production version of E-jacket running that we've tested with my tool and it doesn't break the tool either. So the small cosmetic changes to E-jacket that are coming out in June; also, the code's pretty robust and it will never crash. It will typically just report a warning and then carry on. But there will be times in the future when a new rule comes in that doesn't exist in my list, and then I will have to do some proper code changes.

Yost: You mentioned that you handed it to DIS and donated it, and that they've looked at hiring contractors. Does that mean that you're pretty much done working with it?

Morris: No, it's a tricky balance, isn't it? As far as I can tell, DIS is not just going to convert it to Java. The tool is written in AutoIt. If you go home tonight and have a look at Auto IT on the web, you'll find that it's a very high level scripting language [for Windows]. Yes, you can port it to Java if you want, and typically a lot of their systems run on Java. But that doesn't achieve very much because it's pretty robust anyway. And so it's been harder for me because I can't tinker with it anymore, you know. For example, there's a change I'd like to make to the software, and I can't do it until the June release of the production systems. So it's a real anathema to me; it's really not how I like to operate. It's slightly killing the goose that lays the golden egg. When you have an idea and you have somebody like me who's happy to get their hands dirty with code; and the next thing you know, you can't make changes; you have to put a change request in; that has to be tested; do regression testing; and it takes two months to get a one-line code change through the system. So it slightly takes the fun out of it, that's for sure. But I can certainly

understand why they have to do that, as well. So meaning my involvement, I certainly have slowed down my involvement because I no longer can make rapid and responsive changes to the code. And I can see both sides of the argument, that's for sure.

Yost: You mentioned that you touched base with DIS before you started. Did you also do that with the policy office?

Morris: Narrowly, yes.

Yost: . . . up the ladder?

Morris: Those are all good questions, aren't they? Policy office eventually got involved. Not as early as perhaps I should because there are plenty of implications. Well even the unions might want to know what, why we're automating away lots of the work functions of some of the staff. There's certainly that to consider. Policy, again, have many opinions about this because they're the ones setting the rules. And I only check a subset of the GPG rules and the question is; there's no priority list. I mean I'm picking the ones I think are relevant but maybe there are ones that; they don't rank order their compliance checks. We don't know if the page length, the project description is more critical than font size. And so those are decisions that are hard to make, and typically I was allowed to do whatever I wanted, as long as I could, at a prototype seems to be the conclusion. To answer your question, I just about got away with anything if I say this is a prototype and I'm not offering it as an NSF service. So that was just how I got away with it.

Yost: And I think you mentioned something along the lines of 30 percent of proposals?

Morris: It varies. I think it's about 30 percent of proposals are coming through the checker but still, mostly, by word of mouth. It's still very hard to even find it if you don't know about the tool. It's still not particularly well publicized within NSF; people barely read the news sections on our intranet. So if you haven't heard about it, it's easy to not know it's there. But typically, a lot of people have tried it out.

Yost: How quickly did it get to that point?

Morris: Pretty fast, actually. Faster than I would have liked almost, because I was perfectly happy with having just my division using it. It was quite nice when I had maybe 40, 50 people relying on me. But when I have hundreds and hundreds of program officers; and, of course, it's not just writing code. It's got to be pretty reliable. There's no point if it crashes on a Friday afternoon, it's got to run through the weekend. I don't want to be checking on it or restarting the queue so making it reliable was the big deal. And that's probably what I've done in recent months is not necessarily add more features and services, just making it more robust. So it's been quite hard to have that many customers out there but it's been very educational to build operational software. Really, it's my first exposure to writing code that has to run all the time and not produce rubbish.

Yost: Have you given any talks or workshops to present it?

Morris: Most divisions have invited me round to talk about it, and given presentations to the program staff. And I've got some nice slides that I prepared that explain why I did it; show people examples during the demo. It basically gives them demos while I'm doing my presentation. And mostly people are very [inaudible]; this is great; I mean the idea doesn't solve all our problems. And some people understandably will still be skeptical. There are plenty of people, I'm sure, in the building that will still prefer to do it themselves; would still prefer to manually check the proposals. And that's fine in the smaller divisions where you might have only a small number of proposals. But in engineering and biology, which constantly are the two directorates that have the most proposals, they're the ones most interested in embracing new technology.

Yost: Would it be possible for you to send me the slides?

Morris: Sure. It's a tool and there's nothing proprietary absolutely in those slides. And yes, we have it in PARS. I keep emphasizing I'm trying not to oversell what I've done. But I think it started off with an interesting idea and it's proved to be useful but I think we're now at the point now where it would be great to get more people engaged. Even here at NSF [we should] get some more programmers on board; see if we can do even more stuff with it. I think that the bottom line is we may be approaching the limits of what the computer can do given that when the proposals are in the format that they are -- which is PDFs. If we had much more form-fillable FastLane system where your

biographical sketch, you would answer in your Ph.D. supervisor in one box, and you'd enter in your other bits and pieces in separate boxes, then it would be much easier to search. But when people submitting biographical sketches as a PDF and do these in just about any format they have to choose. I don't know if you've seen that, but they're all suggestions of how to submit your biographical sketch to NSF but typically not everybody follows those rules and the next thing you know, somebody's listed all their scuba diving certifications, or not listing their publications properly. So I think we're reaching the point of maybe how accurate the results can be from an automated checker when all it's got to go on is PDFs. And the PDFs themselves sometimes are scanned PDFs, so they're not even searchable. For example, letters of endorsement from your university may well be a signed letter of endorsement and they would have been bitmap scanned, and if it's bitmapped you can't search it, it can't be converted to text. The only option I have is to OCR that document and then search the optical character recognized document for keywords. And so all of that seems very, very not the right way to manage data. [Inaudible] the IRS; the Internal Revenue Service; they don't let you submit your tax return in the form of a PDF, do they? You're expected to fit in the boxes. And I think NSF really should be going that way as well if it absolutely wants rigorous compliance it really needs to change the way that we ask PIs to submit their proposals.

Yost: Do you have the sense of the mix of support staff versus program officers that are using it?

Morris: That's a good question. I would have thought the majority of people using it are the program assistants, the program staff. But again, that's so direct[ly] division-dependent. Some program officers insist on compliance checking their own proposals by hand themselves, which is an expensive use of federal time, is my honest opinion -- don't have to quote me on that -- but it seems like a crazy thing to do to get a relatively highly paid research scientist to get their ruler out and measure the fonts, or measure the margins, or the number of lines per inch on a document. That is something that the program assistants should do. And again, back to my earlier point that I only see that happening in the smaller programs, or the smaller divisions where a program officer might only have a hundred proposals a year to work through and you simply can't apply that level of scrutiny if you've got a thousand or two thousand or three thousand proposals in your 'My Work' in your inbox. So that's the situation wherein I agree it's very division-dependent for how people use the tool. Some, for example, here in astronomy, you don't actually have that many program staff. I don't think the balance of program staff to program officers here at NSF is a fixed quantity. You may know more than me about this. It seems very odd that work functions of staff here at NSF are not consistent across divisions or directorates. They're very stovepiped, very individual fiefdoms. And so sometimes I deal with program officers in some divisions with the compliance checking but the vast majority of cases it's program staff that are the people running, turning the handle on my code and then getting the results back, turning them into a master spreadsheet, then going in to the program officer and sitting down together and reviewing the results that way.

Yost: And I'll conclude with a couple types of questions that I'm just asking of all program officers that are just kind of broadly evaluative. Are there ways that FastLane and especially E-jacket could be used more effectively to advance the research enterprise and if so, how?

Morris: Good question. I've already touched on the fact that I'd much prefer a better way of submitting data. I think uploading PDFs or Word documents in the way that we allow PIs to do it now is not optimal. And I'll give you an example why it isn't because in recent months we've realized that certain people, when they upload their PDF, the PDF is redistilled within the FastLane system and sometimes that strips out the searchability of the PDF. So we understand the problem, and it's an Adobe problem. A PI may develop or write their project description with an Adobe tool and then they upload that to FastLane. And the Adobe Distiller that we use corrupts that PDF. It only happens in a very small number of cases but it doesn't seem ideal to have that situation arise where we're passing information around and losing contextual information within the document itself, just in the simple process of delivering, uploading through FastLane. So that's the first thing we should probably do is clean up the way that people submit information. Secondly, not have us as program officers check the compliance of a proposal; get it done more upstream, perhaps as the very last step of a submission so the PIs submit their proposal and hit 'send' and then within five minutes they'll get essentially my compliance checker results back in their inboxes. "The automated checker found the following mistakes. We may not be right but I would strongly suggest you look at the things that the checker thought were noncompliant." That would save us having to do a lot of the checking later

on in the process. To be honest, it should be done; the SRO should be done; marked more at the beginning of the process. There are many examples on the web of ways that you can't submit a form when all the boxes are not filled in. I mean again, the IRS analogy is that they don't let you submit a tax return unless everything's they're happy with. You don't then get an e-mail a month later from some program assistant saying we didn't really like the way that you submitted one extra page in your project description, or the equivalent. So more compliance checking done right at the beginning of the process and more form fillable parts to the FastLane process would certainly streamline things enormously here.

Yost: How are noncompliant proposals handled generally?

Morris: There is no general answer to that for the reasons I already alluded [to]. It's very much a discretion of the division, the program, or the program officer. Sometimes, for example, my program will check the number of lines per inch and sometimes it will provide warnings that there's more than six lines per inch. But it might be only slightly more than six lines per inch and the decision rests on the program officer's discretion.

Yost: It might only be 14 pages.

Morris: That's right. It's very hard. We do allow complete discretion, I think, at the program officer level and that is all fine until such time as PIs submit the same proposals to two divisions, or two directorates. That happens quite a bit, because here in astronomy,

a PI might submit a proposal to astronomy but the same proposal could go to somewhere else in physics. And they may not understand why astronomy rejected their proposal because of a particular compliance issue, but physics has already accepted it as is. So we need some more consistency. Perhaps it's been very hard to do that because everybody's grown up and each division has acted so independently over the years. GPG is just for guidance only, as far as I can tell. It's not the bible; it's not the absolute rulebook. Because when you write code -- and I'll use this lines per inch example -- because, of course, if it's over by just 1 percent then my checker gives a warning. But when we give rules in the GPG we don't say plus or minus. There's no tolerance of those rules; it's just you shall not have more than six lines per inch in your proposal. Well what happens if it's 6.1 lines per inch? There is no answer to that. So you have to apply a little bit of discretion.

Yost: Was the interactive panel system in place by the time that you arrived at NSF?

Morris: Yes it was.

Yost: Can you evaluate how that's worked?

Morris: I don't do that many panels. I do a couple, two or three panels a year, generally.

Yost: Do you a lot of ad hoc?

Morris: Mostly ad hocs, and it was all ad hocs in my other division. Actually, here in astronomy, it's almost always panels but I've been [inaudible] from a facility project, which had infrastructure; a telescope project; and therefore, we don't have the same panel [process]. But the people behind you and me in these other offices spend a lot of time in panels. My only exposure to the panel system is that it seems. Have other people have mention to you that the fonts get slightly corrupted when you're submitting a review?

Yost: When they paste Word [documents].

Morris: Pasting and it must be an [inaudible] in ASCII. I don't know quite what the reasons are but we're using a system that doesn't support apostrophes and extended ASCII codes, do we? That just seems crazy that we can't fix that because it makes the reviews look a bit low tech. It's a [inaudible] thing but a little bit of basic formatting helps a lot and if you strip all of that out and the next thing you know you've a page full of question marks. And it doesn't look very professional. That could easily be fixed. And the other thing is I think review comments keep being lost if somebody doesn't tick a box right at the beginning when they log in to the panel system, there's some box they have to check. If they don't do that they can easily lose all of their panel summary results or their review results. Again, there are a couple areas where a page may be not quite as user friendly as they could be, but it's obviously fantastic that it exists at all. I don't know how they coped without an interactive panel system. It's absolutely dependent; and we know that because whenever we've had power cuts here at NSF, pretty much it just; even if you might have lighting in the building, if we don't actually have access to the interactive panel system

the panels can just basically give up completely. So it's absolutely crucial that we have it, and I imagine a few extra tweaks would make all the difference.

Yost: Looking back at the history of FastLane and E-jacket, are there key lessons that can be discerned in the design development of cyber infrastructures and if so, what are those?

Morris: Can I have the question one more time; it's quite a big question, isn't it?

Yost: Looking at the history of how FastLane and especially [how] E-jacket developed, are there key insights or lessons to how best to develop and modify cyber systems?

Morris: I'll directly answer that question with just two comments. I've never heard the expression 'good enough for government' before I came here to the U.S. Is that an expression you're familiar with?

Yost: Oh yes.

Morris: Okay. So that's my first answer; that it may be not the most cutting edge tool sometimes and if Google and eBay ran their business like we do they wouldn't have a business [chance?], I think. So we huff and puff about how complicated it all is to run our interactive panel system, or compliance checks on documents, or submit proposals, but we're talking about tens of thousands; 54,000 proposals a year. I suspect people who are

in industry faced with huge computational tasks, data basing tasks, and GUI user interface, none of this would; I'm sure they would laugh in the offices of eBay or Amazon or whoever, or Google or whatever; at what we consider to complex computational back office functions.

Yost: Then again, their resources are (pause)

Morris: I think there's some very good people here within the Foundation. The DIS typically they've got so many tools, so many legacy tools that they have to keep going that I feel like it's; that we're not innovating as much as we should be. And you're right that we're not throwing as much money; we say that we're not throwing as much money, but some of the complex parts are being outsourced to Booz Allen, and I'm sure those are the same designs and substantive designs of the IT company. It's sheer genius at times; there's some amazing stuff going on. And other times you think okay, it does feel a bit dated and we could be doing more. And that's why I ended up writing a few tools of my own. I wrote a search tool, which I thought was more useful than the search tools that we have here; building on open source search engine, which lots of other companies use. I thought we should apply it here, as well. And I tell you, I don't want to be too mean at all, because we get the job done. But there are a few old systems that definitely are showing their age slightly now and the product life cycle here is so long. I mean 10 years of the same type of tool seems a very long; it's a very long shelf life, isn't it? In the world of software.

Yost: Very much so. And finally, are there any topics I haven't covered, questions I haven't asked that you think would be useful to understanding FastLane and E-jacket?

Morris: The only other thing I developed, which I think has its uses, is a sort of conflict checking tool. So for example, we need to be sure that we don't introduce conflicts into our panels or our ad hoc reviews. And up until recently, the only way that we were really policing that was by manually reading the biographical sketches of the PI and making sure that a reviewer's name doesn't appear in that biographical sketch. All I did was just automate that process a bit so that we could flush out those conflicts more easily, because it's quite hard to do manually. And if you've got hundreds of potential reviews and maybe a thousand proposals; very hard to manually make sure that you're not asking the PI's [doctoral] supervisor to write a review of the proposal. So I did write some software that has its uses, limited uses, but improving it.

Yost: Is that also something that you share?

Morris: It's another one of these tools that's available [on] the Research.gov lab site. And lots of people; I find that actually more useful; it's obviously never going to be 100 percent accurate, mostly because it's still relying on accurate information being provided in the biographical sketch. And that's why it's really important that we don't allow our PIs to write the papers with et.al. because of course, you could easily be asking a reviewer who was a co-author on that et.al. paper to review for that best friend. Although

sometimes it's gray, the paid program officer knows the community. But sometimes [inaudible] best friend in the world, program office says we don't know every linkage.

Yost: It's been one case the name has a middle initial and the other [interrupted]

Morris: There's all of that, yeah; does all that checking. It checks my name as Paul E.

Morris; Morris, E. Paul. So it takes a more thorough check. It obviously struggles with double-barreled names and it's the double-barreled names not being used in some cases. But it typically will throw out a few false positives but it rarely does any false negatives; it typically catches the majority of conflicts and it uses this Regular Expression, which is a tool I used a lot when I was younger. And Regular Expressions have been around since the '80s and they're an incredibly powerful way of detecting patterns in text. And that's what we're trying to do is match a surname, or a first name, or some permutation of first name somewhere in documents. So I find the conflicts stuff keeps us all out of jail. When I wrote this tool I realized it was useful because I ran it on some awards that I've made personally and found conflicts that I wouldn't have been happy with. So it keeps us all honest. We're only humans. It's very easy to miss conflicts so the checker is a very helpful way of at least controlling them, or demonstrating that we're applying due diligence to the peer review process and the integrity of that peer review process.

Yost: And finally, can we do a quick demo and, if it's all right with you, just leave the recorder on in case you want to describe anything about what we're seeing.

Morris: Of course. Just trust in the screen [inaudible] objective. You can probably pull your chair around slightly. I mean are you a PI or an M?

Yost: First, I've been a PI before; I'm co-PI on this project.

Morris: Is it a pending award or is it declined? Or have you got something awarded, do you see what I mean?

Yost: Well, the project, the FastLane project, I'm a co-PI and that's a current project. There's one that's pending that I'm also co-PI on.

Morris: Do you know the number. I mean I can just find it for you.

Yost: No I don't. The PIs name is Misa, M-I-S-A.

Morris: So if I do a search on; have you seen the E-jacket interface? So as a PI; PI last name?

Yost: M-I-S-A

Morris: Yeah. Begins with first name?

Yost: Thomas.

Morris: Is it Tom or Thomas do you think?

Yost: Thomas.

Morris: Let's see if we've got any active awards. There, okay. Is that his?

Yost: Yes.

Morris: Okay. So it's pending. So there's no harm in me taking that number. I mean, I won't drill down into; I'll show you how my tool. Do you want me to show you how the tool works in its basic form; all charges submitted to the checker. Let me just submit it to the checker, which is how a program officer would do. So you open up an e-mail, send it to NSF proposal check, which is proposalsnsf.gov. I'm going to do 'compliance', just put that in the subject line and then just paste a seven digit number in. So that's it. Or, if you have a spreadsheet, you could paste a thousand of them in. So it doesn't matter. So should we just do one now then?

Yost: Sure.

Morris: So what I'll do, if you want to see what happens, I can login to the live server that's running. So we're going to ask [inaudible] checker; this is the actual production box. Okay. So I've run a remote desktop into a server down in DIS. Now you can see

here, it's counting down to the next check so I'm going to submit this; should I just submit that job?

Yost: Sure.

Morris: And what will happen if I just do that. I'm okay with this.

Yost: You've either used . . . (laughs)

Morris: It doesn't get any easier than that. And I'll show you the web interface; but for now, the web check every two minutes, it polls the inbox of the compliance checker and so in a minute's time it will kick into action and process the; I mean, quite interesting to see. This is one that's pending. Have you had any . . . let's see . . . Program Officer is Carl Landwehr?

Yost: Correct.

Morris: Okay, pending grant. So it's in proposal admin. Now you see my e-mail has just arrived; you see it's arrived in the inbox of the server and in 45 seconds it will start to check. Can you see that alright? I could speed it up but you can get the flavor now that people are regularly sending in jobs. Typically, at the end of the day we get a big dump after they go home, in fact. I mentioned that there's people go home at night, they'll submit a few hundred. Typically, we have hundreds, if not thousands a day that come

through. So in 16 seconds it'll kick off and if you wanted to do a conflict check on your proposal you would just type 'conflict check' instead of 'compliance check' in the subject line. So it reads the e-mails and there we go, it's off. So the first thing it does is open E-jacket automatically; what I just did manually. And it now goes to your proposal and I mentioned that I slowed it down, so each part of the award is now being extracted automatically, and then it'll click on the next part; it'll do the cover page, it'll do the project summary, the project description. You get an idea of the speed that it's running at is the speed of the code, it's the speed of E-jacket. I can't go any faster than the speed E-jacket serves up pages. But you can see that now, it's on the biographical sketch of Thomas; and then it'll move through to a cover [inaudible] does all the extractions in. So I deliberately slowed down the process so getting the proposals out to EJ can take a minute or so but the actual compliance check itself is only tens of seconds. So do you want to sit and wait for that to go through to give you an idea; then I'll show you; it is worth waiting for another minute to see the final result. Do you have to get somewhere else?

Yost: I made my next one for 3:10, so . . .

Morris: What time is it?

Yost: It's a couple minutes to, so (pause)

Morris: Okay, alright now this will all be done in a minute. But at this stage you can see that this is the web robot that I described that does nothing more fancy than just emulates

all the manual reading extraction stuff that we would have done anyway as program officers. And once it's extracted everything out; now it's bringing your facilities, equipment and other resources; these are all the sections; in this particular proposal you didn't ask for any facilities or other resources.

Yost: No, it's free, generally.

Morris: Okay. You don't need any labs or mass spectrometers, do you?

Yost: No.

Morris: And then 'references cited' is coming up and that typically means we're coming up close to the end of the process. And of course the major difference here in the future is I won't necessarily need E-jacket in the future, I'll be able to have direct data base access to the PDFs themselves. I wrote this originally without any cooperation of DIS. I thought I would end up developing the tool myself but it turns out that DIS can offer me direct access to these proposals without having to go through E-jacket, which saves all of this scraping; all of this web scraping. And a data management plan; did you submit one in this case?

Yost: I think it was before it became a (pause)

Morris: It threw up an error saying 'Generation Failed' so the PDFs; there is no data management plan. But you did produce your PI information and I think that is; now it goes through your awards, as well. We keep a record of every proposal that the PI and the co-PIs have submitted; and as yours has just come up there, so that's the last bit [inaudible] now, you see this is the compliance check. So it opens up a spreadsheet, a template, and it's processing different rules and each rule is on a separate line so it just spins around, populating the spreadsheet with particular relevant information for the program officer, stuff that they would; again, this is a template that they were filling in manually. And now you see it went pink; and now it's just going to embed the sections of the PDF in the far right hand column, and there's actually a few pink. Pink is bad in this situation, so now it's done. So e-mail now will all come back into my inbox. Do you have to go now?

Yost: No, I have about two or three minutes.

Morris: Well let's just see what it; let's wait for it to send it back. It's now up to my Outlook to send it out.

Yost: Thank you so much.

Morris: Okay. So let's see what it's found warning-wise. . . . major warning; a human subjects warning . . . excessive person months -- that's you -- so the GPG rule is that you have to provide an explanation for more than two months of time and here it says you;

the [inaudible] person man-months on page one is 3.6; is 3.6 person-months on year two and year three. Warning; that's how we use in the publication list, in the bio one. So the first biographical sketch, which is the PI, possible missing graduate or thesis advisor information in bio number one. Possible missing graduate or thesis advisor information, bio two. So maybe in your bio sketch it's possible it's not clear whether your graduate or your thesis advisor's information is there. Warning: Thomas J. Misa, professor, PI mention in text, possible previous NSF support. He hasn't listed the previous NSF support clearly enough. And here, again, I don't know if you remember from the proposal, but you haven't clearly described the NSF support.

Yost: Wow.

Morris: Warning: data manager [inaudible]. And if you want more gory details, it's all there. So that's your spreadsheet. [inaudible]

Yost: Yeah, that's terrific.

Morris: Just a flavor of what we can do. But I'll find out who decides about how I automate this and send them on to you. The trial.

Yost: Here's a brochure on our institute.

Morris: He was a very famous scientist, Charles Babbage, wasn't he?

Yost: Yes, he invented a mechanical computer.

Morris: Yes. He wasn't very American, was he?

Yost: No, and our resources really don't focus on him; we just use the name.

Morris: It's a lovely name. And the Babbage engine or whatever it; they keep redrawing; making those [inaudible]; an amazing whatever it is . . . making differencing engines.

Yost: The Science Museum of London; the difference engine has been built and then recently in Mountain View, California, the Computer History Museum's done the same.

Morris: Well I'm [inaudible] to read that; that's fantastic.

Yost: I have a permission form. We're both the leading historical research center on computers in the world, we're also the leading archive, so we collect paper records and now digital records of use for historical research. And so when we do a research project, we like to add the data to the archives.

Morris: Cool. These are signatures here, also.

Yost: These are initial, and if you're willing to allow it to be used. This is the broader information.

Morris: I don't need to do that.

Yost: Just sign.

Morris: Where do you need to be for your next meeting?

Yost: South or north, but 68507 36th Floor.

Morris: That would be elevator seven, in this building, still.

Yost: Yes, I do have a few in Stafford, here.

Morris: Great. Well that was interesting, anyway.

Yost: Yes, very helpful. I first started on my trip in February, interviewing people and they started talking about this tool. So I got an interview with the creator.

Morris: Well, I didn't do anything anybody else couldn't have done, it's just I decided to do [it] because people said it was impossible of a challenge and I don't think; I could be [inaudible], and I'm hoping as more people get involved [inaudible].