Official Transcript of Proceedings

NUCLEAR REGULATORY COMMISSION

Title:	Advisory Committee on Reactor Safeguards Plant Operations Subcommittee
Docket Number:	(not applicable)
Location:	Rockville, Maryland
Date:	Friday, March 26, 2004
Work Order No.:	NRC-1387 Pages 1-299

NEAL R. GROSS AND CO., INC.
Court Reporters and Transcribers
1323 Rhode Island Avenue, N.W.
Washington, D.C. 20005
(202) 234-4433

UNITED STATES OF AMERICA

NUCLEAR REGULATORY COMMISSION

+ + + + +

ADVISORY COMMITTEE ON REACTOR SAFEGUARDS (ACRS)

+ + + + +

PLANT OPERATIONS SUBCOMMITTEE MEETING

DIGITAL INSTRUMENTATION AND CONTROL

+ + + + +

FRIDAY,

MARCH 26, 2004

+ + + + +

ROCKVILLE, MARYLAND

+ + + + +

The subcommittee met at the Nuclear Regulatory Commission, Two White Flint North, Room T2B3, 11545 Rockville Pike, at 8:30 a.m., John D. Sieber, Chairman, presiding.

COMMITTEE MEMBERS:

JOHN D. SIEBER, Chairman

GEORGE E. APOSTOLAKIS, Member

MARIO V. BONACA, Member

F. PETER FORD, Member

THOMAS S. KRESS, Member

STEPHEN L. ROSEN, Member

ALSO PRESENT:

SERGIO B. GUARRO, Consultant

MARVIN D. SYKES, Cognizant Staff Engineer

JAMES D. WHITE, Consultant

NRC STAFF:

STEVEN ARNDT, RES

MICHELE EVANS, RES

TEKIA GUN, RES

JIAN HONG, NRR EEIB

DEAN OVERLAND, RES

ROMAN SHAFFER, RES

DOUG TIFFT, RES

MIKE WATERMAN, NRR

PETER R. WILSON, RES

I-N-D-E-X

P-R-O-C-E-E-D-I-N-G-S

(8:30 a.m.)

CHAIRMAN SIEBER: The meeting will now come to order. This is a joint meeting of the Plant Operations and Reliability and PRA Subcommittees.

I'm Jack Sieber, Chairman of the Plant Operations Subcommittee. And with us also is George Apostolakis, who is Chairman of the Reliability and PRA Subcommittee.

ACRS members in attendance are Mario Bonaca, Stephen Rosen, Tom Kress, and Peter Ford. And we also have two of our consultants present, Sergio Guarro and Jim White. Marvin Sykes of the ACRS staff is the Designated Federal Official for this meeting.

The purpose of this meeting is to discuss digital instrumentation and control research activities, including the development of digital system reliability models. We will hear presentations from representatives of the Office of Nuclear Regulatory Research, the University of Virginia, and the University of Maryland.

The subcommittees will gather information, analyze relevant issues and facts, and formulate proposed positions and actions, as appropriate, for deliberation by the full committee.

The rules for participation in today's meeting have been announced as part of the notice of this meeting previously published in the Federal Register on March 8, 2004.

A transcript of the meeting is being kept and will be made available as stated in the Federal Register notice. Therefore, we request that speakers identify themselves and speak -- move to a microphone and speak directly into the microphone with sufficient clarity and volume so that they may be readily heard.

We have received no written comments or requests for time to make oral statements from members of the public regarding today's meeting.

We will now proceed with the meeting, and I call on Steve Arndt of the Office of Nuclear Regulatory Research to begin. Steve?

MR. ARNDT: Thank you. I'd like to introduce my Division Director. He may have a couple of introductory remarks.

MR. MAYFIELD: Good morning. I'm Mike Mayfield, Director of the Division of Engineering Technology, and this work is sponsored out of my division. We want to thank the committee -- subcommittees for the opportunity to come and discuss this. We have tried unsuccessfully a couple of times to schedule onto your calendar, and events kept overtaking us, so we appreciate the opportunity to come brief you on this important work.

We think we've put together a pretty comprehensive story to present to you today, and we look forward to feedback and the opportunity to interact with the committee.

With that, Steve?

CHAIRMAN SIEBER: Thank you.

MR. ARNDT: Thank you. We've put together a pretty aggressive schedule. You have in front of you -- but I just want to highlight what we're going to try and accomplish today.

The first presentation, which I will give, is an overview of the research program, a discussion of the state of the art -- actually, the state of the practice is probably better terminology -- in this area, and review of several of our research programs.

Following that, the University of Virginia and the University of Maryland will highlight two of our larger programs specifically. I will then come back to the microphone to discuss future plans in the area, and then we'll have the adjournment.

So the idea basically is to give you a comprehensive overview of the program, highlighting the two particular programs that the committee has been interested in in recent years.

As I mentioned, the overview will give you a context of where this -- of where the reliability program fits into the overall I&C program, and also discuss some of the issues we have with the particular state of the art in this area.

As requested by the committee, we will have conclusions, review of the I&C program, boundary conditions and drivers, why are we going down this particular path at this particular time, review of digital system reliability modeling, current methods, and then discussion of the research programs.

Our research program is designed to answer the questions that we think we're going to get as an agency in digital system risk assessment. The drivers, as I will discuss later, have to do with getting ready for the reviews that the licensees are likely to submit.

So as much as we'd like to do exotic, fun research, we also have to temper that with, do we have enough information of the methods that are most likely going to be submitted to be able to make reasonable judgments.

Research includes model development, data collection and analysis, and guidance development. What we're trying to do is put together a tool package for our licensing brethren, so that they can do their jobs more efficiently and realistically.

We're working on development tools not only to understand the methodology but also to assess the methodology as a check tool. And some of those are in the demonstration phase right now, and we're trying to work with both our contractors and other researchers in the area to stay abreast of the state of the art.

The particular issues are to develop the kinds of guidance we need. We need to be able to assess whether or not there is enough information and enough experience in the application of these methods in the domain we're interested in to make some judgments.

We currently think that the models are sufficiently mature to do that. Now, are they great? Maybe not. But the threshold here is, are they mature enough that we can make judgments as to whether or not they are sufficient for the application they're going to be looking at?

We have ongoing future work -- we'll talk about that later in the day -- associated with integration into PRA models and audit calculations, and things like that. There are a lot of different issues that we continue to have and we continue to work to, especially including the data issues and the coordination with our international colleagues. So that's one of the issues that we continue to strive to improve on.

The next few slides are going to be an overview of the I&C research program as a whole to give you a context of where the reliability program fits. As you know, the current program plan was embodied in SECY-01-0155, published in August '01. It will come to an end -- the planning horizon for that plan -- at the end of this fiscal year.

So we're in the process right now of developing a new research program plan, which will describe our successes, the things we haven't gotten to for resource or commitment issues, and then talk about what we're going to do in the future. We'll probably have some interactions with the committee late summer or early fall on that issue.

The research plan was developed in answer to the National Academy of Sciences' National Research Council study calling for a more systematic and integrated research program in this area. It was reviewed and endorsed by the ACRS and the Commission.

It has five basic program areas. We'll get to those in a minute. The reliability program is one of the five program areas within the research program. Our goal is basically to improve the staff's analytic capabilities and their fundamental knowledge.

To do any kind of reasonable assessment you need both a fundamental knowledge of how the systems work and how they fail and what problems you can get yourself into, and the analytical capabilities, the tools, the models, the procedures, to be able to use that knowledge in a review process. And that's our basic goal -- to get those two pieces and provide them to our regulatory brethren.

MEMBER KRESS: Your 10 minutes are up.

MR. ARNDT: Okay.

MEMBER KRESS: Is this research in cooperation with any of the industry? Is EPRI or NEI involved at all?

MR. ARNDT: We've done some cooperative work with EPRI. That is always a challenge, to try and find efforts that mesh well and also don't have a conflict of interest in various other areas. As with all of the other research programs, we meet with EPRI on a fairly regular basis, with industry brethren on an occasional basis, to talk about what's going on, what we can do.

We currently are doing some work I believe in the wireless program collaboratively with the industry, but none of the reliability programs are currently collaborative in a strict sense. We're using work in the industry.

MEMBER APOSTOLAKIS: Yes. Can you go back to 6?

MR. ARNDT: Yes.

MEMBER APOSTOLAKIS: On what basis have you decided that the current analysis methods are sufficiently mature?

MR. ARNDT: The basis -- well, we'll talk about that later in the presentation. But the basis is that they're being used in other industries for safety-critical decisionmaking.

There has been -- define "successful" as you like -- successful applications of these methodologies for safety decisionmaking in industries that are sufficiently similar to the kinds of decisions and the kinds of systems that we have to be practical for -- in implementation.

MEMBER APOSTOLAKIS: And these industries are?

MR. ARNDT: The transportation industry, for example, the rail industry --

MEMBER APOSTOLAKIS: Is NASA using any of these?

MR. ARNDT: NASA is using many of these methods. The aerospace industry -- not all of the industries are using the same methods. All of them are as comfortable with the methods as others.

MEMBER KRESS: When you say "methods," are there more than one?

MR. ARNDT: Yes.

MEMBER KRESS: To say the fault injection process?

MR. ARNDT: Well, there's a number of methods, and you can dice them up any of a number of ways. One would be a fully integrated system modeling type method versus modeling systems that are not fully integrated, like software separate from hardware --

MEMBER KRESS: Yes.

MR. ARNDT: -- things like that. You can dice and buy the kinds of particular analytical method to use, petri nets, dynamic fault trees, dynamic flow graphs. You can dice them by whether they're primarily data-driven or system model driven. You can dice them in a lot of different ways.

But the point is that some subset of the models have been successfully used in a regulatory sense, which is the basic piece of information that drives the conclusion that it -- we are capable of doing -- writing regulatory guidance.

Now, whether or not we can write regulatory guidance that would be effective in this industry is something that remains to be seen.

MEMBER APOSTOLAKIS: But when we say "analysis methods," maybe we can make a distinction between methods that search for faults in the software and methods that attempt to quantify the reliability or probability of failure. And you're referring to both sets?

MR. ARNDT: I'm referring to both sets.

MEMBER APOSTOLAKIS: Because a number of years back the staff, when they were writing the standard review plan I think, they told us they talked to Boeing, and Boeing told them to forget about all of these markers, and just test the thing. And, in fact, there is a regulatory guide that --

MR. ARNDT: Yes.

MEMBER APOSTOLAKIS: -- or someplace where it says the staff, at this time, does not place any confidence in --

MR. ARNDT: Yes, that is --

MEMBER APOSTOLAKIS: -- on that.

MR. ARNDT: That is the current regulatory position.

MEMBER APOSTOLAKIS: So since then things have changed.

MR. ARNDT: Since then, the progress of technology, both in the ability to model how the system fails, and the ability to quantify that, has progressed.

MEMBER APOSTOLAKIS: Okay. Okay. We'll see later --

MR. ARNDT: Okay.

MEMBER FORD: Steve, I've got a general question.

MR. ARNDT: Okay.

MEMBER FORD: Some time ago you mentioned to me that you were involved in SCSIM development.

MR. ARNDT: Yes.

MEMBER FORD: Is that with respect to quality?

MR. ARNDT: No.

MEMBER FORD: Are you using it in this program?

MR. ARNDT: We're not. That happens to be just one of my personal sidelines.

MEMBER FORD: Oh, okay.

MR. ARNDT: Examples of meeting those goals have to do with developing analytical models, updating guidance like the reg guide that you recently saw from us, and doing technical support of other regulatory programs, be it software quality, instrument work, systems and review work, etcetera.

The four aspects -- the five aspects of the program -- I will go through quickly the four that are not reliability programs, just to give you a context. One of them is systems aspects of digital systems, environmental stressors, PMI/RFI environmental qualifications, those kinds of issues, requirement specifications, operating systems. These are things that have generic application to a large group of systems or component-level type issues.

Software quality assurance issues, requirement specifications, the issue of how do you test requirements, how do you test failures like that, how do you look at engineering -- specific engineering criteria -- the work at Maryland touches on this program as well as the reliability program.

Emerging technologies and their applications -- this is a proactive part of our program where we're looking at specific technologies that either are becoming or already have become major issues in the balance of plant applications and may become safety issues in the future.

So advanced instrumentation, smart sensors, wireless communications, large programs for security, as you might imagine. And we also have a program that continuously reviews technology to determine what we should fold into this program. Things like application-specific ICs and things like that will probably get folded into the next update of the plan this year.

Advanced reactor I&C infrastructure -- as you have heard from many briefings on advanced reactors, one of the parts is the reapplication reviews. The other part is the infrastructure development. I&C has a piece of that. We're looking at various different issues. We have a lessons learned document looking at what we can learn from the other plants.

One of the recommendations of the National Academy's study was to do more, learn more from what has happened in the industry, other places than the United States. And I will point out that one part of the advanced reactor program is the development of risk assessment for plant applications -- specifically, developing issues to, one, support I&C in the risk framework for advanced reactors, as well as look at specific applications to new technology that's going to be developed for advanced reactors and how that will impact our other work in the reliability program.

Risk assessment of digital systems -- this is the program we're going to talk about today. There are four basic areas, and they kind of, over the last four years since we wrote the plan, have kind of diverged a little bit.

But the basic areas are looking at data sets and understanding what's available, how we can use it, how we can bound things, not only for specific applications of developing failure rates, but also what does the data tell us? Is it confirming our assumptions? Is it giving some information on what's more important and what's less important? Those kinds of issues.

MEMBER APOSTOLAKIS: You'll address this later?

MR. ARNDT: Yes. We'll talk about this later.

MEMBER APOSTOLAKIS: Good.

MR. ARNDT: But one of the big issues is: will there ever be enough data to really do reliability predictions? Well, that's a debatable issue, but there will always be some data. And we can use that data to do these other things as well.

MEMBER APOSTOLAKIS: These are failure data from other industries, I suppose.

MR. ARNDT: Well, both -- very limited from the nuclear industry and from other industries.

MEMBER APOSTOLAKIS: Okay.

MR. WHITE: Steve, this is James White. One of the -- a couple of things that we found in the National Academy's study was we had a lot of -- people seemed to have a lot of difficulty finding this reliability data, that vendors who had worked in other industries were a little reluctant to share that data. I'd be interested in how much progress you think we've made since the National Academy's study.

And the second question, before I forget it, is that we found that it -- it seemed that the nuclear industry was talking to itself a lot when it was wrestling with the software reliability problem. And I'd be interested -- and maybe you're going to cover it in your presentation -- how we are really putting out work that is: a) published with peer review, and b) how we are becoming part of the community, so that we are not alone in the work.

Thank you.

MR. ARNDT: I will attempt to answer those questions as part of my presentation in the presentation to the contractors. If I don't, please remind me again, because one of the big issues in this, as you say, is it's a very difficult problem. It's a problem we've been wrestling with as a community in the software business and the digital system business for some time.

Nuclear is a very small piece of it. It's a very specialized small piece of it, in addition to that. So tying in, both consciously and through our contractors and through collaborative work, is a conscious effort we have made to try and improve that over the last four or five years. And we've been I think reasonably successful in that area. Obviously, we can do more, and we're working to do more, both in the nuclear area as a whole and the other industries and other efforts.

The two areas here -- digital failure assessment methods and digital reliability assessment methods -- this really gets to, do we understand the systems? Do we understand how the systems fail? Do we understand the failure modes? Can we model them properly?

And this basically has to do -- once we know that, and we take that and put it into a methodology, that will get us actual quantitative numbers that we can then use in regulatory space.

And then the last part, of course, is guidance, be it reg guides or review guidance or checklists, or whatever, for assisting NRR staff in their ability to review this work.

Just to give you a --

MEMBER APOSTOLAKIS: Well, let me understand this a little better. What actions, regulatory actions, do you foresee NRR will face in the next couple of years?

MR. ARNDT: Okay. We're going to talk about this a little bit more. But to give you the five-second version, a lot of the plants are upgrading their systems, both small individual pieces and some plants -- I think the number now is four that have already told us they're going to do complete control room upgrades. And we suspect that there's going to be a lot more than that.

As well as -- that's basically large-scale reviews that are going to hit all of the different areas, including software and other things, as well as there are several plants that as part of that review would like to risk-inform at least parts of their application, particularly the defense-in-depth and diversity requirements.

So we have both the issue of specific areas that are going to want to use risk information that we need to find methods to assess and information to validate, as well as the overall process that we would like to improve, make more quantitative, more realistic, more consistent.

MEMBER APOSTOLAKIS: So do you foresee that we may have a regulatory guide like we have now for risk-informed ISI and --

MR. ARNDT: That's under discussion.

MEMBER APOSTOLAKIS: Okay.

MR. ARNDT: We haven't -- we haven't discussed it enough with NRR for me to comment on it.

MEMBER APOSTOLAKIS: That's fine.

MR. ARNDT: It's something that we're looking at.

MEMBER APOSTOLAKIS: Yes.

MR. ARNDT: Just to give you a quick perspective, the budget for the I&C section, all of the stuff I've just talked about, is about

$3.8 million in ISDE. Of that, about one FTE and $1 million is devoted to the reliability program.

This gives you a quick perspective on the kind of resources we're spending on this kind of --

MEMBER APOSTOLAKIS: So reliability program means Virginia and Maryland?

MR. ARNDT: No. It means everything we're going to talk about today -- Virginia, Maryland, the BNL work.

MEMBER APOSTOLAKIS: Okay.

MR. ARNDT: Some of our in-house work.

MEMBER APOSTOLAKIS: Okay.

MR. ARNDT: Okay. Program external drive -- we've talked about this a little bit. National Academy of Sciences' National Research Council recommendations -- Jim was on that committee.

One of the many issues that they raised was this whole issue of software reliability and digital systems reliability, and we should be more proactive in that. We'll talk about it a little bit more.

I mentioned the DOE I&C and human machine interface working group recommendations. This was a group of people that was convened by DOE a little less than two years ago to specifically look at what are the things that advanced reactors aim for? Basically, the NERI/NEPO kinds of issues. What is going to come? Why is it going to be an issue?

They had a subgroup on regulatory issues. The biggest recommendation out of that subgroup was you've got to be able to risk-inform the applications. Outside the mainstream you can't do that, particularly since the advanced reactors reviewed are hopefully going to be more risk-informed.

There was a workshop in Halden in December of 2002 that also looked at this from an international standpoint. There were recommendations out of that that basically said we need to do more than -- there is not self-consistency within the international community, and that we need to develop these issues.

And I'll talk to this last one. In particular, the draft EPRI report on diversity and defense-in-depth -- that's what I mentioned a few minutes ago. The diversity and defense-in-depth requirements were written when we rewrote Chapter 7 of the standard review plan, because at the time the information available on software common mode failure and those kinds of issues was very sparse. The requirement, in the opinion of many in the industry, is unnecessarily restrictive.

EPRI has developed a draft topical report that they tell us they will submit in I think it's August of this year for review.

MEMBER KRESS: Is that diversity you're talking about having a separate analog system?

MR. ARNDT: Yes.

MEMBER KRESS: Okay.

MEMBER APOSTOLAKIS: Can we get a copy of this EPRI report?

MR. ARNDT: Is it publicly available, do you know?

MEMBER APOSTOLAKIS: Do you have a copy?

MR. ARNDT: Yes, I have a copy.

MEMBER APOSTOLAKIS: Then we should have a copy.

MR. ARNDT: It was given to us for a courtesy review.

MEMBER APOSTOLAKIS: Yes. Well, not the public, I don't think. If you have a copy, we should have a copy. And we will treat it appropriately.

MEMBER ROSEN: A follow-up to Tom's question, you said on this diversity and defense-in-depth it was -- it meant an analog system backing up a digital. Is that what I heard you say, or could it mean a different digital system backing up?

MR. ARNDT: It can be a different digital system.

MEMBER ROSEN: Either one.

MR. ARNDT: Yes.

MEMBER ROSEN: Okay. Now, while I've got your attention, let me just ask what your thumbnail sketch is of what you mean by "risk-informing these requirements." I could guess, but I'd rather hear what you think.

MR. ARNDT: The draft that's on the table basically uses a methodology that we'll talk about a little bit more in -- later in the presentation to come up with a criteria based on .174 risk criteria that basically says, "This is good enough from a risk standpoint."

The current requirement asks you to go through and do a very detailed review of what can happen if a system fails due to a common mode failure software. This is an alternate method to do that analysis that basically uses risk-informed criteria as the decision point as opposed to a deterministic analysis of, if it fails, it's not a problem.

MEMBER KRESS: So it takes into account the consequences of failure, not just the fact of failure.

MR. ARNDT: Yes.

MEMBER KRESS: Is that what you're saying?

MR. ARNDT: Yes.

MEMBER KRESS: It takes into account the frequency also.

MR. ARNDT: It derives a frequency of failure of the system, of the software --

MEMBER ROSEN: And then assesses the consequences and comes up with a risk.

MR. ARNDT: Right.

MEMBER ROSEN: As opposed to just saying, "Deterministically, show me that everything that failed -- that can fail, will fail, and what the effects are."

MR. ARNDT: Well, it's a somewhat unusual thing, because it requires certain specific assumptions on how the system failed and what you can credit and what you can't credit. But basically that's correct. It says, "These are the basic assumptions you have to make, do a deterministic analysis and come up with, will it meet the threshold or not?"

MEMBER ROSEN: Okay. Thank you.

MEMBER APOSTOLAKIS: Was the DOE report really a driver, though, Steve?

MR. ARNDT: It wasn't a driver so much as a confirmation that -- the people who design things and look at these kind of things are going the same direction.

MEMBER KRESS: Was the National Academy report useful to you?

MR. ARNDT: It was, more so in some areas than others. Of course, it's somewhat dated now, but it highlighted some --

MEMBER KRESS: It was '93, wasn't it, when it --

MEMBER APOSTOLAKIS: No.

MR. ARNDT: No, no, it was --

MEMBER APOSTOLAKIS: '99? 2000?

MR. ARNDT: I've got it right here.

MEMBER KRESS: Well, '93 is when it started.

MR. ARNDT: Yes. But it was published in '97.

MEMBER KRESS: Okay.

MR. ARNDT: The final recommendations were hashed out relatively late in the process, if I remember correctly.

MEMBER KRESS: The reason I ask is, you know, I sometimes wonder whether ACRS recommendations are useful to you. That thing got started as an ACRS initiative.

MR. ARNDT: Yes, I know. Yes, they are, particularly since the committee has a broader perspective on these things than sometimes we do.

MEMBER APOSTOLAKIS: Are you asking him whether the ACRS is useful?

MEMBER KRESS: Well, I was --

MEMBER APOSTOLAKIS: Did you expect him to say no?

(Laughter.)

MEMBER KRESS: Actually, no, I didn't. But actually, I was wondering --

MEMBER APOSTOLAKIS: Steve is an honest guy, but this is pushing too far.

MEMBER KRESS: I was wondering in that specific case whether it was good advice to them.

MR. ARNDT: A quick review of the -- what the National Academy said and what the NRC's PRA policy says. This was in your package that we sent you, so I won't go over it in detail. But the basic thrust was we need to be able to assess software failures in a reliability sense.

We need to be able to develop failure probabilities, particularly including COTS software, or COTS hardware for that matter. We need to be able to understand and analyze the systems, and we should be working with whoever is appropriate to develop the capabilities and expertise to be able to do this kind of thing.

MEMBER KRESS: Well, that sounds like an ACRS letter.

MR. ARNDT: Well, you can thank Jim and his colleagues for that.

MEMBER APOSTOLAKIS: And by the way, that letter the committee wrote, when was it, 10 years ago?

MEMBER KRESS: '91.

MEMBER APOSTOLAKIS: Yes. It was one of the most obscure letters --

MEMBER KRESS: '93.

MEMBER APOSTOLAKIS: -- ever to come out of --

MEMBER KRESS: Yes, I know it was --

MEMBER APOSTOLAKIS: -- this committee.

MEMBER KRESS: I know. Sort of wandered around. That puts it in real concise terms.

MEMBER APOSTOLAKIS: Yes.

MR. ARNDT: Just a reminder that the PRA policy asks the staff to increase the use of PRA. The operative word here -- to the extent supported by the state-of-the-art methods and data.

The real issue is, as we've pointed out, the last time we looked at this in the '97 timeframe when we updated the SRP, we didn't think that it was appropriate. Now we're looking at it again, and we think it may be appropriate.

MEMBER KRESS: We were wondering what your interpretation is of what's meant by state-of-the-art methods. It can be interpreted several ways.

MR. ARNDT: Yes. My personal opinion is state of the art was a poor choice of words when we helped -- when we wrote that. I actually helped write that particular part of the document. What it really should mean is state of the practice.

MEMBER KRESS: That's what we thought.

MR. ARNDT: Can you practically do this with the domain that you're interested in, with the kinds of information that is necessary to make a decision?

A quick review of the kinds of things we're trying to attack -- this is actually from a paper that Nathan and I wrote about a year and a half ago. The kinds of things we need to be able to do this work is an understanding of the state of the data, what is it -- what are the limitations, what are we going to have to work around, understanding -- a deep fundamental understanding of how the systems fail, what kinds of effects are important, is communication issues important, is timing issues important, software important, strengths and limitations of these models, what are they going to tell you, what are they not going to tell you.

MEMBER APOSTOLAKIS: Has this been done?

MR. ARNDT: Part of our research in several of the programs we're going to talk about gets at this particular issue.

MEMBER APOSTOLAKIS: So there is a review of the available models, so there will be a review of available --

MR. ARNDT: Actually, almost all of our projects have this as part of their program.

MEMBER APOSTOLAKIS: So we're going to hear about it today?

MR. ARNDT: We're going to hear about some of it today.

MEMBER APOSTOLAKIS: Okay.

MR. ARNDT: There was a short discussion of this in the first report that University of Virginia put out. There's going to be a much more extensive discussion in the report of BNL. Our future work is also going to reassess these issues.

MEMBER APOSTOLAKIS: Because that has been a major problem with the human reliability models.

MR. ARNDT: That's correct.

MEMBER APOSTOLAKIS: There was never a critical review of other people's work, and, you know, trying to build on the good parts of different models.

MR. ARNDT: Right.

MEMBER APOSTOLAKIS: Each guy develops his own or her own. Okay.

MR. ARNDT: The whole issue of how do you incorporate a model into the PRAs, not only PRA as a whole but the actual PRAs that are being used -- the practical applications that are being used. And there are some significant limitations because of the structure of the current PRAs that are out there.

And then, understanding what your acceptance criteria is, not only for the actual number and the uncertainty associated with that number, but also, if you will, PRA quality or the model quality. How good does it have to be? What kind of assumptions are acceptable? What are not acceptable?

What we're trying to accomplish is to improve the review process by providing additional information, guidance, and tools. To accomplish this, we're going to basically develop the understanding, improve the guidance, and develop tools that can assess the system, inform the reviews and/or provide audit calculation type capability.

I'll try and skip through the next three or four slides pretty quickly. It's basically just the structure of what we're trying to accomplish and how we're trying to accomplish it, how the programs fit into what I just said.

The kinds of products we're going to have -- we'll basically develop a tool box that can develop guidance as to what is acceptable and what's not by quantitative measures to better inform the reviews. At this point, we do not envision going entirely to a quantitative review, like 2,200 degrees for fuel mount.

What we want to do is make the reviews that are currently very qualitative more quantitative to increase their realism and their repeatability, and perhaps demonstrate alternative methods to meet the safety goals, like third party audits and things like that.

These are the research projects that we have in this program. These are diverse integrated digital systems modeling, which you'll hear more about

-- at the University of Maryland, the software metrics project, which you'll hear more about.

And we have the BNL project on digital system risks. This project is basically going at it from the PRA standpoint backwards. These two projects are basically going from the failure kind of methods and the -- how you model how the system has failed toward the PRA. So it's a different perspective on the same problem.

And we have several other programs that I'll go over briefly, basically some additional database issues and some additional efforts, including the work that Halden is doing in this area.

MEMBER ROSEN: And we're going to hear about the BNL project, too?

MR. ARNDT: Right now.

DR. GUARRO: Excuse me, Steve. On Chart 20, you say digital system failure mechanisms. Can you clarify the scope of that? In other words, when you -- the term "failure mechanism" extends to what?

MR. ARNDT: It extends to how the system fails. Basically, is it failing because of random failures of the hardware? Is it failing because of software encountering situations it was not designed for? Is it failing because of data communication issues? Is it failing -- basically, how does it fail, and why does it fail? And what design and implementation issues or contexts --

DR. GUARRO: So you include the design side as well.

MR. ARNDT: Yes.

DR. GUARRO: Thank you.

MR. ARNDT: Quickly, the way we're trying to accomplish what I just talked about in these particular programs -- the University of Virginia is integrating -- is looking at integrated digital systems modeling projects. They're going to develop assessment methods that can be used by the staff for independent assessment -- $4 billion for that matter -- to understand the models and come up with other numbers on whether or not they function properly.

And they are also developing information on failure modes in reliability that can be used in the regulatory guidance to form our guidance development.

MEMBER APOSTOLAKIS: So how is this different than from what BNL is doing, digital system risk?

MR. ARNDT: I'll tell you in a minute.

MEMBER APOSTOLAKIS: Okay.

MR. ARNDT: They are basically developing methods from the -- how does the system --

MEMBER APOSTOLAKIS: They? They?

MR. ARNDT: Virginia.

MEMBER APOSTOLAKIS: Yes.

MR. ARNDT: How does the system fail? How can we model those failures? What are the critical issues associated with it? And developing a methodology that we can use to evaluate it.

MEMBER APOSTOLAKIS: Okay.

MR. ARNDT: And they're using the information they gained through that process to form our reviews.

Maryland's software metrics project is developing methods to assess -- help us independently assess software quality, basically developing a method using software metrics that is readily available. Metrics are developed as part of the design process and testing process -- that can help us independently assess the system. That will also --

MEMBER APOSTOLAKIS: Wait a minute now. So you will have two methods for reliability assessment -- Maryland and Virginia?

MR. ARNDT: Yes.

MEMBER APOSTOLAKIS: Two separate methods.

MR. ARNDT: Two separate methods.

MEMBER KRESS: Yes. I'm having trouble figuring out how this University of Maryland work led to reliability. I sort of envisioned you ended up with a software quality index of some sort, based on the processes it went over.

MR. ARNDT: Well, we'll talk about this in detail this afternoon. But the issue is: you will end up with an understanding of how the particular metrics of software quality affect the overall quality of the system, and also whether or not those are good predictors of its reliability.

MEMBER KRESS: But you have to have another way to measure the reliability in order to make that assessment?

MR. ARNDT: Yes.

MEMBER KRESS: Yes.

MR. ARNDT: You have to test the system to validate --

MEMBER KRESS: Okay.

MR. ARNDT: -- the methodology.

MEMBER KRESS: Okay. This --

MR. ARNDT: One of the things we're doing is testing it by doing that to determine whether or not it is --

MEMBER KRESS: So this is like the six or seven parameter input. It ends up with quality and reliability, and you're going to try to --

MR. ARNDT: You've got to validate it.

MEMBER KRESS: -- get some sort of correlation between the two or --

MR. ARNDT: Well, it's not a correlation.

MEMBER KRESS: Not a correlation, but some --

MR. ARNDT: It's a model --

MEMBER KRESS: It's a --

MR. ARNDT: -- that basically says this kind of information will give you a good prediction of how well it will behave in the future, because --

MEMBER KRESS: You expect that to be a qualitative thing rather than quantitative?

MR. ARNDT: It will be a quantitative system. It probably -- we will probably not get to the point that says, "If it meets this number, it's okay." It's not going to be that kind of quantitative, but it will be a number that we would be able to use to inform the process.

MEMBER APOSTOLAKIS: So, and BNL is also going to develop a risk model?

MR. ARNDT: The BNL project is focused on --

MEMBER APOSTOLAKIS: Yes.

MR. ARNDT: -- these kinds of things. They're looking at helping us write the regulatory guidance. They're doing a detailed review of the current methods, as you mentioned. They're looking at the database issues, and they're looking at how do you take these kinds of models -- these two models were assessments of the systems.

This is specifically looking at taking that and other data and putting them into the PRA context.

MEMBER ROSEN: Are you going to give us some more detail about that?

MR. ARNDT: Yes.

MEMBER ROSEN: So we have some sort of flavor of what's being thought about?

MR. ARNDT: Yes, sir.

MEMBER APOSTOLAKIS: Well, it's interesting that you are developing two reliability models. Why?

MR. ARNDT: The big issue is we don't know what the licensee is going to submit to us. There is a lot of different methods out there currently, which we'll talk about in a minute.

Some of them are completely integrated systems. Some of them are not completely integrated systems. As you know, there's a large debate as to whether or not that is reasonable and how to do different things like that.

The bottom line is we need to understand how these issues affect the system, so we can make an assessment. So we're going at it in several different ways, so we can gain enough information to be able to write guidance, what is acceptable, what is not acceptable, what the limitations are of various methods, and look at improving our regulatory process in various specific ways.

MEMBER APOSTOLAKIS: So the thing that will ultimately really be the final product is this digital system PRA model.

MR. ARNDT: There will be several things. The guidance will be --

MEMBER APOSTOLAKIS: Yes. Yes.

MR. ARNDT: -- an issue.

MEMBER APOSTOLAKIS: In terms of numbers.

MR. ARNDT: In terms of numbers, we hope to have, either through this work or other work, a tool that we can basically run like we run Sapphire now, to give us a check on whether or not the number that the licensee is giving us makes sense or not.

MEMBER APOSTOLAKIS: And work at Maryland and Virginia and possibly other places provides input?

MR. ARNDT: That's correct.

MEMBER APOSTOLAKIS: Okay. That's a very interesting approach.

MEMBER KRESS: Yes. Let me tell you what my initial view of this was, and you tell me where I'm wrong. The current way we look at software quality is by evaluating the process mostly.

MR. ARNDT: Mostly.

MEMBER KRESS: Rather than the product.

MR. ARNDT: Correct.

MEMBER KRESS: Now, I viewed the University of Maryland work as looking at that process and trying to maybe rank the parts of it as to their effect on quality in some way, but not yet looking at the product. And I viewed the University of Virginia work as focusing on the product and actually trying to figure out a way to take the product and get some measure of its reliability. And then you have a way to maybe connect the two, and is that --

MR. ARNDT: Yes.

MEMBER KRESS: -- is that a pretty good

view of what you're doing?

MR. ARNDT: It's an appropriate view. It's not a comprehensive --

MEMBER KRESS: It's not comprehensive.

MR. ARNDT: -- approach, not inaccurate.

The process we're trying to do is to take various pieces and both improve the current process, which is mostly process and development based, and to develop a new process that is primarily product based, so that we can review the systems more effectively.

MEMBER APOSTOLAKIS: So there may be a combination at the end.

MR. ARNDT: Absolutely.

MEMBER APOSTOLAKIS: Yes.

MR. ARNDT: And in many cases it will be driven by what the licensees give us.

MEMBER ROSEN: Steve, I think that's my cue for jumping in here. I'm a little bit surprised by that attitude -- that it will be controlled by what the licensees give us. We don't know what the applicants are going to send to us to review.

I mean, those kinds of statements you made are a little bit surprising, because I think there's another way to go at this, which would be to define through this research what the licensees or applicants need to give you.

MEMBER KRESS: Well, a lot of that will come out of your guidance, I think, yes.

MR. ARNDT: And maybe the tone in which I said it was not appropriate. But the research has several slants on it. One, of course, is exactly what you said. We develop an understanding of all of the different commonly used methods, so we can assess what is provided.

The other issue is we need to make a decision, both in terms of a number if we're going to use a number, and also on what is acceptable in terms of modeling. If we make a determination that certain models are simply not sufficiently accurate, sufficiently reliable, whatever, based on our research, then we draw a threshold there.

So, yes, you're right. A large part of our research is to define what is acceptable, what the validity of the models are, if you will.

You'll hear later this afternoon about a lot of the programs, particularly in Maryland and Virginia. It's not just the model, but it's also validating the system. We're using actual nuclear instrumentation and control systems to validate it. Does it work? Is it acceptable?

MEMBER ROSEN: I mean, to simplify this discussion, it seems to me that one could say to an applicant, "You can design software any way you like, and to have it do anything you'd like it to do in the powerplant. But you must analyze it after you're done with that and submit that analysis this way," because that's the way we evaluate the -- what your products are.

And that would then allow the applicants and the vendors to say, "Okay. Ultimately, we're going to have to pass this test, so our software may have to be -- and the way we design it -- may have to facilitate that."

MEMBER KRESS: Yes. Quite often the reg guides serve that purpose -- talking about developing reg guides.

MR. ARNDT: The reg guide -- a reg guide can serve that function, but not as strongly as you just put it.

MEMBER KRESS: It's one way to --

MR. ARNDT: It highlights an acceptable method. In some cases it becomes a de facto requirement because of the way we --

MEMBER ROSEN: Because it's too hard to do otherwise. To support the -- a review by the staff of some alternate method that maybe somebody thinks is better, they say -- you can rightfully say, "Well, you can do anything you want to do not to comply with this reg guide, but it will take us longer." And that's rational.

So this becomes, de facto, the way they do business. But as long as that de facto was is a good way that's well supported by research and your knowledge, I don't see there's anything really wrong with that. And I would -- I would think that it's a better posture to be in, saying that's where we're headed, than saying, "Well, we'll have to deal with anything they send us."

MR. ARNDT: Well, yes, and that has, in point of fact, been done in several industries. And I think Dr. Johnson will mention that in his talk, because he has done work in --

MEMBER ROSEN: Well, it's the way the agency does business now. I mean, you can't just send us anything. We have, you know, regulatory guides.

MEMBER BONACA: But, yes, in general, however, vendors also propose ways in which you should be testing. I mean --

MR. ARNDT: Yes.

MEMBER BONACA: -- they propose -- or they will propose, you know, concepts that should be used for testing. And so you are -- you are trying to understand acceptability --

MR. ARNDT: We're trying to understand, based on the things that have been proposed or been talked about -- like the EPRI guidance -- what is acceptable and what is not acceptable. And the current methods that are being used, both in the United States and other places in the nuclear business, are not as sophisticated, shall we say, as some of the research we're doing.

And we also have the issue that the current structure is basically qualitative. And if we want to change that --

MEMBER ROSEN: And we do.

MR. ARNDT: Yes.

MEMBER ROSEN: And we must, I think.

MR. ARNDT: Well, we then need to demonstrate that not doing it the other way is not sufficient.

MEMBER ROSEN: We need to demonstrate that?

MR. ARNDT: Well, we have a backfit rule that we can't --

MEMBER ROSEN: Oh, well, for existing

plants maybe that's so.

MR. ARNDT: Yes.

MEMBER KRESS: Is it premature for you to -- I guess it is -- to start thinking about what your acceptance criteria are? I can see we're going to have -- you know, look at specific digital I&C systems related to safety functions probably, and you're going to look at the defense-in-depth aspects of it.

And then you're going to quantify the reliability and see what its contribution is to the actual risk of various sequences. I don't know what the -- you know, I don't know how to say -- when you're focusing on some specific SSC --

MR. ARNDT: Yes.

MEMBER KRESS: -- what an acceptance criteria might be. I mean, are you giving some thought to that?

MEMBER APOSTOLAKIS: Why couldn't it be 1.174?

MEMBER KRESS: Well, that's for the whole -- I don't know how you parse .174 into various sequences and various components.

MR. ARNDT: You don't.

MEMBER APOSTOLAKIS: No. You just --

MEMBER KRESS: I know. But what we're

doing is you're going to -- you're going to have before you an I&C system for the safety function, and we'll say, "Is it acceptable or not?" And I don't know how you parse that into 1.174.

MEMBER APOSTOLAKIS: But we don't parse anything out.

MEMBER KRESS: I know. But that's what they're going to be faced with -- the decision. Is that acceptable or not?

MR. ARNDT: Yes. And, really, the more difficult issue, although that will be a difficult issue, is the licensee may come to us with an analysis based on whatever methodology and say, "The answer is X, and that meets the .174 threshold," or acceptance criteria.

The real issue we're going to have is: is the analysis quality sufficient?

MEMBER KRESS: What's the uncertainty in that --

MR. ARNDT: What is the uncertainty? What is the -- do we believe the answer based on the methodology that they use?

MEMBER ROSEN: And that's my exact point.

MR. ARNDT: And that's exactly correct. That is the --

MEMBER ROSEN: That's my exact point. You shouldn't get into that box. You should have your own way of analyzing the software which you impose.

MR. ARNDT: Right.

MEMBER ROSEN: So you can analyze it any way you like for your own purposes. But when you come in here for regulatory approval, you must analyze it this way. This is the way we understand it. We get a delta CDF from that. We can compare to 1.174, and make a judgment as to whether that's accurate -- acceptable within our --

MR. ARNDT: And one way to write the reg guide is --

MEMBER APOSTOLAKIS: Now, you know, Steve, yesterday we had a meeting on another subject, but we were told that EPRI has started a project on uncertainties in general with particular focus on model uncertainty. We were also told that the staff here -- Mary Druin I think is involved in that -- has a parallel effort, and now they will start talking to each other.

I believe you should at least be aware of what they are doing and maybe give them some input, because in my opinion you will have a serious model uncertainty issue here --

MR. ARNDT: Yes.

MEMBER APOSTOLAKIS: -- and all these questions from Steve and Tom, you will have to address it --

MR. ARNDT: Yes.

MEMBER APOSTOLAKIS: -- you know, the issue of acceptability. So if the industry is doing something on it, the staff itself is doing something on it, you should be a participant and maybe by giving them some of your problems you will help them as well to do a better job. But you should also be aware of what they are doing.

Right now they are looking at the major model uncertainties in Level 1 PRA --

MR. ARNDT: Right.

MEMBER APOSTOLAKIS: -- like the RCB or seal LOCA.

MR. ARNDT: Right.

MEMBER APOSTOLAKIS: And so on, and human reliability. Yours is closer to human reliability. I suspect you're going to have model uncertainty that's pretty significant here.

MR. ARNDT: Yes. And --

MEMBER APOSTOLAKIS: So were you aware of these efforts?

MR. ARNDT: I am aware of the effort. I have not been an active participant in it.

MEMBER APOSTOLAKIS: Well, take it as a first piece of advice from the subcommittee.

(Laughter.)

You should be aware of what they're doing.

MR. ARNDT: Oh, absolutely.

MEMBER APOSTOLAKIS: And they should be aware of your problems.

MR. ARNDT: Absolutely. And one of the challenges in this work is, of course, we have various stakeholders within the agency. We have our PRA group, we have NRR's PRA group, we have our I&C group, we have the regulatory PRA group, we have the various stakeholders outside the agency, including EPRI and their --

MEMBER APOSTOLAKIS: That's model uncertainty right there.

(Laughter.)

MR. ARNDT: Okay.

MEMBER APOSTOLAKIS: Okay, great.

MR. ARNDT: At the risk of trying to --

MEMBER BONACA: I just -- this is for information for me. I mean, I am not an I&C person, and I -- before you made a statement regarding the fact that some of the applications are -- maybe I misunderstood, but limited or simple or -- now --

MR. ARNDT: Many of the models that are being used --

MEMBER BONACA: Okay.

MR. ARNDT: -- particularly in the nuclear area --

MEMBER BONACA: Yes.

MR. ARNDT: -- where this has gone a little bit further down the path like in some of the foreign countries, are more simplistic than the ones that we are going to talk about today -- was the statement I made.

MEMBER BONACA: What's the limitation? I mean, why are they so simplistic? I mean, it seems to me that, you know, we live in a world where there is so much application of digital systems right now with tremendous sophistication. I mean, what is limiting? I'm trying to understand the limitations you are talking about, the simplistic portion.

MR. ARNDT: The limitations are mostly driven by compulsive -- the model you want to use, the data you have available to populate that, either failure data in a more generic sense or actual faults and testing of the faults, and things like that, the amount of information you have about the proprietary systems, and those kinds of things, because, as Dr. Johnson will talk about a little bit -- and we'll talk about a little bit elsewhere -- one of the challenges in any kind of models like this is getting sufficient information to populate them appropriately.

You have a lot of different computational problems associated with it, which we are to a point now I think it's not a major problem anymore, because there have been some new methods developed, but not everyone has embraced those, things like states-based proliferation and things like that.

So there's a lot of specific modeling challenges associated with this, and there are much simpler kinds of methodologies, like software fault trees and things like that, that don't deal with some of these issues.

MEMBER BONACA: Okay.

MR. ARNDT: And it's a judgment call. Is it sufficient? Is it a sufficiently accurate model for the application you're trying to do? Can you decouple software failures from the hardware context?

MEMBER ROSEN: Well --

MEMBER BONACA: Okay.

MEMBER ROSEN: -- I think the question of sufficiency is one of risk. I mean, it depends upon what the risk introduced is.

MR. ARNDT: That's correct. And there have been some proposals that basically say for certain kinds of systems you need to demonstrate risk to a certain level. One way of writing the criteria, as has been proposed, is basically to say, for a certain kind of system you have to have a sufficient demonstration of the risk as lower than -- choose a number -- 10^{-4} failures per demand with a reasonable uncertainty, and develop a criteria based on that kind of statement.

That's what was done in part at the size we'll be analysis that they did. They basically set a criteria that they didn't want the system to have a --

MEMBER BONACA: Okay.

MR. ARNDT: -- failure on demand worse than a particular thing --

MEMBER BONACA: So when you use the word "simplistic," really you are talking about simplistic approaches to evaluating the reliability of the systems and determining faults. Okay. Because, I mean, I was thinking about systems themselves and the sophistication that they may have.

MR. ARNDT: Yes.

MEMBER BONACA: And you focus, of course, on the -- okay, I understand. Did we get a written report from BNL?

MR. ARNDT: No.

MEMBER BONACA: No.

MR. ARNDT: That's still in draft form.

MEMBER BONACA: Okay.

MR. ARNDT: When it's available, we will forward it to you.

Quickly, the other programs are focused on providing the traditional information in these areas. We'll talk about them very briefly. We're running a little late on this.

We've talked about a lot of this, but let me go through this quickly. The modeling issues that we're facing -- the state of the practice now -- have to do with issues of what kind of failure modes do you include, how do you know you have all of the failure modes, have you done a failure mode effects analysis, and it has what kind of systems, the level of detail of the models, both the software and the hardware, is processor level sufficient, do you need to go lower than that.

The big issue, of course, is: can you

treat hardware and software independently or not? To a certain extent, that's a bit of a red herring, because you -- you always have to treat software, to some extent, dependent on hardware because software doesn't exist in isolation of what the system is running on. But can you separate it from an analysis standpoint?

And software diversity issues, of course, is a big issue. How diverse really is this software? How do you ensure diversity and things like that? The whole issue of the number of possible stakes and space proliferation. Although some of the more sophisticated stratified testing has dealt with this, there's not as much need to anymore.

The requirements -- what is the ability to predict? How do you demonstrate that the analysis is really predicting the real failure? And what kind of validation studies are necessary? And things like that. And is it at least supportive or at least consistent with what data is available?

MR. WHITE: I'm sorry to interrupt you. But on software diversity, as you will remember, on the National Academy panel we spent months wrestling with that. Where is your program on the issue of maybe having to write the requirements in a different way to assure that you get software diversity?

You know, the argument that Nancy Levenson was putting forth is, if you and I sit down with the same requirements and write software, we're going to make the same mistakes regardless if you use one language and I use another language.

And we just didn't have time to -- to wrestle that particular concern to ground. Are you going to address that today, or could you just give me a quick summary of where you are?

MR. ARNDT: We're not planning on addressing that particular issue today. But as you point out, that is an issue. There has been several actual studies done in the last few years specifically looking at that particular issue. Are you going to use different languages and different databases, and things like that? And the real solution that has been proposed that I am aware of is basically enforced diversity basically.

You don't just put two people in a room and tell them to go use different methodologies. You force them to use a different methodology. And that, I believe, is the state of the practice for that particular issue.

As we've talked about, there are various methods available that have been used -- are being used. I'll talk a little bit more about what the current state of the practice is. But the real issue is, as we've talked about, is setting an acceptance criteria for both the modeling fidelity and the system reliability. That's the real challenge.

MEMBER APOSTOLAKIS: I guess Dr. Guarro is the originator of the dynamic flow graph methodology, and I have worked on it, too, so he and I will say nothing when it comes to this.

(Laughter.)

MEMBER KRESS: That would be unusual.

MEMBER APOSTOLAKIS: Huh?

MEMBER KRESS: That will be unusual.

(Laughter.)

MR. ARNDT: A lot of the methods, particularly the dynamic flow graph methodology, are very powerful and effective in doing this kind of analysis. Again, the challenge we have is setting a threshold. What is acceptable?

The context we have --

MEMBER APOSTOLAKIS: When you comment on DFM, I have to reply the way the French team replied -- non salons il voltre repons. Nobody seems to know French.

MEMBER KRESS: Oh, yes. I knew what you meant.

MEMBER ROSEN: I just don't understand it with a Greek accent.

(Laughter.)

MEMBER APOSTOLAKIS: But that's how the team said it.

(Laughter.)

MR. ARNDT: The background or context is what's -- where we currently are. Most of the trial methods that you see in nuclear space are using methodologies that more theoreticians would have serious problems with. The biggest particular issue is treating software failures, in a modeling sense, independent from hardware failures. That is a significant problem.

Some methods are even not that sophisticated. They use very simplistic bounding analysis. That is to say, demonstrating that the particular failure mode of a particular component is no worse than its analog colleague without dealing with issues associated with timing issues and communications issues, and common mode issues, and things like that.

Where we set the threshold in this area is one of the reasons we're doing -- investigating various methods to understand the advantages and disadvantages.

MEMBER APOSTOLAKIS: But it's not always the acceptability, though, Steve, isn't it? I mean, if you first satisfy yourself that maybe by using two or three methods you have identified the important failure modes, without any attempt at quantifying, that will be a major achievement.

MR. ARNDT: Yes.

MEMBER APOSTOLAKIS: Then you go to the next level, which brings up risk acceptability, and so on, where things are a little shakier there.

MR. ARNDT: Right.

MEMBER APOSTOLAKIS: So maybe the separation should be always in our minds that certain methods do a really job at identifying certain failure modes, but there is another method that does a better job for other failure modes.

And I think that's where a lot of the work out in the literature is. And another thing that's happening in the -- and I've seen it in other places -- oh, we have to use a model for reliability of software, and somebody I know is using this model. So, and it was published in the proceeding, so this really must be good. Let's use it.

And you are actually evaluating critically, I hope, the underlying assumptions for each model, not just because somebody used it.

MR. ARNDT: Right.

MEMBER APOSTOLAKIS: Okay.

MR. ARNDT: What are the advantages -- what are the inherent limitations of the modeling technique?

MEMBER APOSTOLAKIS: Right.

MR. ARNDT: What are acceptable assumptions? What are unacceptable assumptions? Those --

MEMBER APOSTOLAKIS: Yes.

MR. ARNDT: Those kinds of issues. For example, as Steve mentioned, we can set a particular methodology, or we can set a set of issues that have to be addressed in whatever methodology that's been put forth. We're currently going down the second path, although we can certainly look at the first as an alternative.

But the particular issue, particularly when you start dealing with things like -- that are not state of the practice models, is is it at that threshold where it is dealing with the assumptions that it -- that it's making in a way that makes sense and can be useful?

I'll go back to the first part of your comment having to do with there are really two issues dealing with failure modes and understanding them better and understanding the more reliability and failure-type issues as opposed to the PRA issues. That is specifically what we're trying to -- we're trying to both go down the path of risk-informing, but also trying to make the current methodology a little more realistic.

MEMBER APOSTOLAKIS: And I suspect you will make the methodologies more quantifiable. I suspect you will make much more progress on the failure mode analysis than the quantification, which will -- probably will be challenged more by the reviewers than by us, of course, but --

DR. GUARRO: Steve, do you have any activity, either ongoing or planned, to try to determine whether in the context of the nuclear industry this assumption of separating software from hardware is a good one or bad? Because -- and I'm asking this because some of the more spectacular failures that have occurred in the aerospace industry have occurred because the software was simply the messenger of the sign error.

So is that declared out of scope or --

MR. ARNDT: No, it is not. That is a specific area that we are looking at. You'll hear Dr. Johnson this afternoon -- this morning talking about his methodology which, of course, doesn't make that assumption. It looks at it in an integrated fashion. But also in our review of the methodology it was done by BNL, which we're going to talk about in a second, as well as future work.

We're going to look at that -- those specific kinds of assumptions. Can you make those assumptions? If you make those assumptions, is there any way to mitigate those assumptions? How you look at something else that will catch some of those issues.

What is the threshold, in essence, for an acceptable model? And this is obviously one of the big issues.

DR. GUARRO: Okay, thanks.

MEMBER ROSEN: Steve, I'm getting a little troubled by one sense I'm getting, and maybe you can help me understand it better.

MR. ARNDT: Okay.

MEMBER ROSEN: The sense is that we're

going to analyze this very hard problem and figure out how to deal with it, and then overlay that understanding with the risk approach. And it seems to me that the risk approach itself has the power to make your first problem easier. Let me explain.

If risk -- if you use the risk approach integrated with the underlying assumptions, underlying work you're doing in the static failure modes and effect, you can say the risk approach brings in the question of consequences. And if the consequences of a failure of a particular set of software is very limited, then you're almost done with the problem before you have to get -- you don't have to solve it from a first principle aspect.

If you can say, well, the worst that can happen, for example, is it will trip main feedwater, well, tripping main feedwater happens now, and it's -- you know, the plant will scram, and that's a relatively benign event.

MR. ARNDT: Right.

MEMBER ROSEN: I mean, so you can use the risk modeling --

MR. ARNDT: Yes.

MEMBER ROSEN: -- to make your first problem easier.

MR. ARNDT: You can do it backwards, basically.

MEMBER ROSEN: Yes.

MR. ARNDT: And that's actually the fundamental concept behind most of the bounding methods. They look at, if it fails anyway, is it going to be --

MEMBER ROSEN: What kind of failure can it make?

MR. ARNDT: What kind of failure can it make? And will it be any worse than X? Analog equivalent or the issue associated with it -- it won't drive you to Part 1 under release or whatever.

MEMBER APOSTOLAKIS: Which is a fault tree type analysis. You start with the consequence, and you are asking yourself, now, how can the system, in combination with the software, can take me there?

MR. ARNDT: Right.

MEMBER APOSTOLAKIS: Right?

MEMBER ROSEN: Well, I'm not sure exactly that's what I meant. I was looking at thinking about the software's function, saying if the worst that this software can do, regardless if it just locks up, it doesn't do anything, or it sends a signal, the worse it can do -- the only wire it's got is to the main feed pump circuit.

MR. ARNDT: Right.

MEMBER ROSEN: Then, the worst that can happen is I can -- my main feed pumps can go to full speed, or they can go to zero speed I guess. There aren't any other options, are there? And so -- and both of those are okay, I mean, from the standpoint of consequences.

MR. WHITE: Well, it's an interesting perspective. The problem is that the software that would, first of all, cause a failure of the main feedwater pump, or indicate a failure of the main feedwater pump, might also cause a failure in another piece of software where the consequences would be more important. That makes it a little more difficult to --

MEMBER ROSEN: Well, I understand that that may be the case in some software. But in -- for the particular software you're looking at has the feature that it can only affect what the main feed pumps do or don't do. Then you have a much simpler problem.

MR. ARNDT: Yes. And at the risk of being difficult, that's one of the reasons why we're trying to evaluate different kinds of methodologies for their acceptability for the particular application. If it's an isolated system, it doesn't have any significant impact on other systems, if you can model the software in such a way that it doesn't have the kind of issues that Jim brought up, then you can use a less sophisticated model.

MEMBER ROSEN: That's my only point.

CHAIRMAN SIEBER: Well, but the software really isn't written that way.

MR. ARNDT: In most cases that's correct.

CHAIRMAN SIEBER: For example, you may have a software module that acts like a controller. Okay? And then sitting someplace else is the contents of the scaling manual that says, "Here's proportional band, here's rate, here's reset," etcetera. And that same model is used in 500 different applications, the same piece of software.

So you really can't say that if the -- if you have a software failure some device quits doing its thing. It may be that every device in the plant quits doing its thing. It would --

MEMBER KRESS: At the same time?

CHAIRMAN SIEBER: Yes, because it's the same model.

MEMBER KRESS: Same input to each one of

them.

CHAIRMAN SIEBER: Yes. And so if you crash the model, or it has some kind of a hang-up loop that's not available to do anything else. So to me I think the problem is pretty complex for systems that are designed that way.

Now, there are other systems that are independent. And, you know, for the sake of diversity they have separate trains with separate models using different algorithms, and so forth. And we've seen some examples within the last two years of -- some of us -- of that kind of methodology. And maybe you can comment on that.

MEMBER KRESS: I don't think looking at it in a backwards way like that helps you a lot, because you already their subsystems, that if they fail you're in trouble, like the control systems, the scram systems. If things don't work right, you've got a problem.

MEMBER ROSEN: Well, you're talking about the solid-state protection system, for instance, in a Westinghouse plant. You can't --

MEMBER KRESS: So if there's -- so if there is a number of systems like that that you already know, you're going to need this information.

MEMBER ROSEN: Sure. I'm not saying that you're not going to need this. I'm just saying there's a class of problems where it might get simpler, and you should think about those, too.

MR. ARNDT: Yes, absolutely.

The next part of the presentation is on the BNL research. We're running a little late, so I'll go through this reasonably quickly. The BNL research was designed to basically look at the issues from a more PRA standpoint as opposed to a digital failure standpoint. Of course, they dealt with those issues as well.

And they looked at strengths and weaknesses of current models. They looked at what was necessary to develop guidance in this area, suggestions for improving the integration methods, database failure type issues.

The reports that they're going to have will include basically this information: the review of the current models, list of issues associated with probability failure, and some of the things we've talked about already. Some of those were new issues. Some validated what we already knew.

The draft interim review guidance that we're going to use --

MEMBER APOSTOLAKIS: So is BNL going also to present, or this is it?

MR. ARNDT: This is it.

MEMBER APOSTOLAKIS: Okay. When will we get the draft report? You said they are preparing a draft report.

MR. ARNDT: It should be available fairly soon.

MEMBER APOSTOLAKIS: You have to step to the microphone and tell us who you are.

MR. OVERLAND: Dean Overland, Risk Assessment Group in Research. The draft report will be available -- I believe it should be available this month, this upcoming month.

MEMBER ROSEN: From the PRA standpoint, PRA Committee, that's what we want to see. That's how we would --

MEMBER APOSTOLAKIS: Maybe we can have another subcommittee meeting in the future to talk about the risk aspects.

MR. ARNDT: Well, depending upon how aggressive we are on the guidance, we may want to come talk to you about that specifically anyway.

MEMBER APOSTOLAKIS: Well, you said that this year you will develop a plan for the next several years.

MR. ARNDT: Yes.

MEMBER APOSTOLAKIS: Like to have our input.

MR. ARNDT: Yes.

MEMBER APOSTOLAKIS: So I guess in the next several months we will have to write a letter. Is that correct?

MR. ARNDT: It probably won't be several months, but probably late summer by the time we discuss it and get input from various --

MEMBER APOSTOLAKIS: So you will come to us in the fall some time?

MR. ARNDT: Probably, yes.

MEMBER ROSEN: Well, George, don't you think it would be better for -- once they get the draft report, for them to review it internally rather than just send it to us at the same time in parallel? I don't think there's that --

MEMBER APOSTOLAKIS: I don't understand what --

MEMBER ROSEN: Well, I would rather hear from the staff about what they think about the BNL report rather than being sent the BNL report and --

MEMBER APOSTOLAKIS: Well, let's get it

first.

MEMBER ROSEN: -- alone and --

MEMBER APOSTOLAKIS: Oh, you are proposing another subcommittee meeting?

MEMBER ROSEN: I'm proposing, yes, a subcommittee meeting in which the staff and BNL come together and say, "Here's the report we got three, four months ago."

MEMBER APOSTOLAKIS: Because that's the ultimate problem, actually. You're right.

MEMBER ROSEN: Right. And then -- and here is -- at which point, you know, we get staff's view as well, and then we write the letter.

MEMBER APOSTOLAKIS: So when do you think that can be --

MR. ARNDT: Well, we're mixing apples and oranges here. There is three issues that were talked about. One is the BNL report specifically. That will be available next month, and then what we're going to do with it we'll figure out shortly thereafter.

The other issue is any guidance document that we may develop, that will be a little bit longer timeframe. The third thing is the staff plan for the overall digital I&C program.

MEMBER APOSTOLAKIS: Right.

MR. ARNDT: Which will be late summer.

MEMBER APOSTOLAKIS: Right.

MR. ARNDT: So we could combine these, we could do them independently, whatever you guys think is most appropriately -- appropriate.

MEMBER APOSTOLAKIS: Well, certainly from past experience, I assume you would like to come and brief us on what you are doing on the guidance --

MR. ARNDT: Yes.

MEMBER APOSTOLAKIS: -- before you finish the guidance.

MR. ARNDT: Yes.

MEMBER APOSTOLAKIS: Get some ideas back and forth, and so on. So that is one of the most critical meetings we're supposed to -- we are going to have.

MR. ARNDT: Right.

MEMBER APOSTOLAKIS: Why don't we leave it up to you and our staff to arrange? Because the time is short, actually. We can't have too many subcommittee meetings. But we will have to judge --

MR. ARNDT: Okay.

MEMBER APOSTOLAKIS: My inclination is -- would not be to do all three in one subcommittee meeting.

MR. ARNDT: Okay. I agree. It's, one, too much material, and they are different aspects of the issue.

MEMBER APOSTOLAKIS: Yes.

MR. ARNDT: Okay. One of the areas was the interim guidance. Basically, it identifies particular needs in the review and makes information -- makes use of some of the information that they generated when they did an evaluation of one of the generic platforms.

As you all know, or should remember, there are three generically approved digital platforms. These are most likely going to be the basis for most of the safety grade upgrades in the plants in the future. Brookhaven used one of those generic platforms in its work.

MEMBER APOSTOLAKIS: Now, I'm a little curious, because I wasn't involved in the approval. How did the NRC approve those platforms? I mean, was it -- did they do any of this, the stuff that you presented to us the last hour and a half?

MR. ARNDT: They used the current version of the standard review plan, which, as we discussed --

MEMBER APOSTOLAKIS: Okay.

MR. ARNDT: -- is primarily qualitative.

MEMBER APOSTOLAKIS: Process-oriented.

MR. ARNDT: Process-oriented, yes.

MEMBER APOSTOLAKIS: Now --

MR. ARNDT: Now, they're going to do another plant-specific review when the plants use the generic platforms for plant-specific application.

MEMBER APOSTOLAKIS: Well, that would be more limited, then, because you have already approved the platform. It's like approving AP1000, the design, and then somebody actually builds it. You don't start from scratch, right?

MR. ARNDT: No, you don't start from scratch, but I would -- I would caution to say limited. It's going to be a fairly extensive review.

MEMBER APOSTOLAKIS: Okay.

MR. ARNDT: And we're hoping to have some of these tools available to at least inform those reviews.

MEMBER APOSTOLAKIS: Processes.

MR. ARNDT: As part of BNL's work to develop the guidance, they did some quantitative assessments. They looked at analysis. They looked at the initiating events, particularly the differences in initiating events from a traditional --

MEMBER APOSTOLAKIS: This is too exciting, Steve. You're really giving us stuff that is really very interesting, but we're not going to talk about it. So why don't you skip it?

MR. ARNDT: Okay.

MEMBER APOSTOLAKIS: I mean, you are talking about new initiating events. I'm dying to see what they've done. And you say, "No, no, no, you're not going to see it." So keep going, then.

CHAIRMAN SIEBER: Well, before we go too far, we would like to take a break this morning. When is a good place for you to stop to allow us to take that break?

MR. ARNDT: This is probably as good a time as any.

CHAIRMAN SIEBER: That's what I was thinking.

(Laughter.)

Why don't we take a break until quarter after 10:00.

(Whereupon, the proceedings in the foregoing matter went off the record at 9:55 a.m. and went back on the record at 10:15 a.m.)

CHAIRMAN SIEBER: Okay. Let us return to session.

MR. ARNDT: Thank you, Mr. Chairman.

When we left, I had just started a brief description of the BNL work. As we're running a little late, I will try and work through that fairly quickly.

As part of their review, they looked at both state-of-the-art issues and modeling issues. Some of the issues that they looked at in the development of the guidance we talked about. They also looked at software failure issues, both the whole issue of whether or not probabilistic modeling is appropriate, as we have discussed previously, for software failures independent of hardware.

The various kinds of models were looked at, as well as the common cause failure issues for software. They looked at hardware failures, particularly the issues associated at what level of component failures needs to be modeled in an appropriate model, as well as the issues associated with failure data for hardware systems, common cause hardware failures, particularly things like communication buses and things like that that can have potential issues, software-hardware interactions, which are a particular issue, and then the integration of the digital systems within existing PRAs.

One of the challenges of this methodology is there is a lot of fairly sophisticated methodology, some of which are easy to integrate into static PRAs and some of which are not very easy to integrate into static PRAs. And --

MEMBER ROSEN: The question here is: if a plant -- an existing plant with an existing PRA chooses to make a safety-related system improvement using digital software --

MR. ARNDT: Right.

MEMBER ROSEN: -- how does one then incorporate that into the model to answer the question as to what happens to the CDF --

MR. ARNDT: Right.

MEMBER ROSEN: -- to the whole plant? That's the question I have.

MR. ARNDT: That is -- the primary issue in the bullet referred to as integration of -- into the existing models.

MEMBER ROSEN: Okay. So you're going to -- somebody is going to answer that question for me. I'm not smart enough to answer it. I just want the world to answer it.

MR. ARNDT: That is one of the issues, and there are methods that have been proposed. For example, dynamic fault trees and Markov models can be integrated into static PRA or the whole PRA can be turned into a dynamic fault tree and then integrated.

Those are not easy things to do, but they are theoretically possible. Obviously, there are other methodologies that can be developed. You can use them as an input to a particular failure rate that then goes into this an initiating event, upfront module, ahead of the initiating event.

MEMBER ROSEN: Right. Events

MR. ARNDT: Events.

MEMBER ROSEN: Not much one.

MR. ARNDT: Multiple events. There are several different methodologies that have been proposed and have been worked on. NASA, for example, has done a lot of work on dynamic fault trees for these kinds of issues. So there is examples in the literature on how to do this.

MEMBER ROSEN: Dealing with the issue of the fault tree is what fails first, and then assess how the system reacts to it, or the system has an upset of some kind --

MR. ARNDT: Right.

MEMBER ROSEN: -- and the fault failure -- software fails during the upset.

MR. ARNDT: Right.

MEMBER ROSEN: Or -- so, I mean, both of those issues.

MR. ARNDT: Right. Both of those issues, and the issue of, in the particular model that you're using, is issues particularly common with failure is the model you're going to use capturing all of the common mode failures of a software-driven system.

MEMBER ROSEN: The most challenging piece of it seems to me to be that if the software system fails first, it would initiate the transient, and you're relying on the same software system to mitigate the occurrence that it just initiated.

MR. ARNDT: That's right. And on top of that, one of my personal pet peeves is there have been failures in which not only are you counting on it to mitigate it, but also the failure prevents you from doing other things that might mitigate it, like, for example, it locks out the manual action, things like that, which is both difficult to model but potentially very significant from a consequence standpoint.

MEMBER ROSEN: Okay. I just checked to make sure the scope of what you're addressing is something like what I hope you're addressing. I think I got the answer yes.

MR. ARNDT: It looked at a variety of methodologies -- for example, the fault tree analysis for AP6000, the INEL study, the work that Barry is doing in fault injection methodologies that uses Markov models.

They looked at some of the other guidance that is out there and that has been proposed. The Bayesian belief network, which is a methodology that some of you are familiar with that is very useful for combining qualitative and quantitative data to provide information to basically make a decision.

We're also investigating this on a separate project, both from a reliability standpoint but more importantly for improving the review process. As we get more quantitative information, how do we integrate that into our current qualitative programs?

And as part of their work, they did a failure modes and effects analysis for one of the generically approved platforms, to understand how this can be done and what the appropriate level of modeling should be.

The did the traditional top-down step-by-step approach, identified the potential dependencies, and generated the questions about the particular design that you would have to answer to do an effective analysis.

And that's one of the big issues, because when -- if you're going to do more detailed modeling, you're going to need more information, or in some cases different information than you would ask if you're going to do a process-based analysis.

Some of the insights they got when they did that, in order to capture the information you basically have to do what you would do in any probabilistic model. You have to have a very detailed understanding of how the system fails, which we discussed that previously.

And you have to have a generic method for evaluating various kinds of issues, such as communication between redundant channels as an example. You have to figure out how you're going to do that and have an agreed-upon method to do that.

Another part of the review -- we asked them to go and look at the databases that are available, both within the nuclear industry and in other industries. One of the things they did was they looked at the LER work. There is a large number of failures in the LER database, and many of them are digital systems or software-based systems.

One of the biggest issues with that

database, not only for this application but for other applications, is the amount of information you have. And that's one of the biggest challenges in the digital failure databases is frequently, one, the people who have the failure data may not populate it into the database.

But also, they may not have it, because the solution was a card failed, we pulled it out, we put a new card in. And exactly what failed, how it failed, and what the root cause of that was may not exist, or may not be populated in the database. So that's one of the significant challenges.

MEMBER ROSEN: Now, there are plants that are repairing cards.

MR. ARNDT: That's correct.

MEMBER ROSEN: And those people know what failed on the cards. And they can then tell you or give you access to data which would let you know what that failure did.

MR. ARNDT: That's correct.

MEMBER ROSEN: If you know that this electrolytic capacitor, for example, failed on the card, because it was replaced and the card worked --

MR. ARNDT: Right.

MEMBER ROSEN: -- then you know a lot more than -- and so one of the threads you might try to pull is, where are places that are repairing digital cards? Because they will have data that will be useful to you, and may be willing to share it.

MR. ARNDT: Yes. And the biggest challenge in all of this is going out and pulling all of those threads, or finding other people who have pulled them before and building them on, as George mentioned, what people have done, what information is available.

One of the reasons we asked BNL to do this was to get a better understanding of not only what is and is not available but what people are doing with it. They reviewed the MIL handbook data, PRISM data. They looked at other sources that could be pursued, other industries and government agencies, manufacturers, and remanufacturers in the case of cards.

MEMBER APOSTOLAKIS: NUREG 6734 is what? Is it the data?

MR. ARNDT: Yes.

MEMBER APOSTOLAKIS: Can we get a copy of that?

MR. ARNDT: I think so. I'd have to go and --

MEMBER APOSTOLAKIS: Well --

MEMBER ROSEN: One minute. I don't think -- maybe we're not communicating yet.

MR. ARNDT: Okay.

MEMBER ROSEN: You said remanufacturers. Sure, but I was talking about utilities, maintenance staffs, I&C maintenance staffs, that are repairing their own cards.

MR. ARNDT: Yes.

MEMBER ROSEN: Those people will be a great source of data.

MR. ARNDT: Yes.

MEMBER ROSEN: So I just wanted to make sure you understood what I meant.

MR. ARNDT: Okay. Yes, I understood. I was remembering different -- yes, sir.

DR. GUARRO: Just curious -- what were you looking into in the review of 217?

MR. ARNDT: I'm going to have to defer that question to one of our contractors who is in the audience.

MR. CHUN: This is Lewis Chun, Brookhaven Lab. Mainly we got hold of the 217 and see what information is there, and see how people use the method provided there in their analysis. Basically, it's part -- part stress method they use, and then the PRISM database is kind of a replacement database that -- because 217, my understanding, was discontinued, and the PRISM is kind of like replacement, which keep updating the data in the database.

DR. GUARRO: Well, yes. The direction of my question was two ways. In 217, there is nothing that is software-specific. I was wondering how that will apply to the --

MR. CHUN: Right. It's lumped -- if you look --

DR. GUARRO: Also, yes, it is true that it has not been updated since 1991. So it's very old data, in any case. PRISM is an evolution, but it's also now no longer a government-endorsed database. So it -- the usefulness of it is sometimes questioned.

MR. CHUN: We look at it as just another source of data, and it's somewhat like a continuation of 217. But the method there is still similar to that of 217, so in terms of software failure I think it is embedded in the failure events that they use in estimating the failure rates.

DR. GUARRO: Okay. Thanks.

MR. CHUN: How adequate that is, you know, is questionable.

MEMBER APOSTOLAKIS: What kind of events are these? I'm not familiar with 217. Is it aerospace, or what?

DR. GUARRO: 217 is an electronic component failure rate database.

MEMBER APOSTOLAKIS: Who developed it?

DR. GUARRO: The Department of Defense.

MEMBER APOSTOLAKIS: Okay.

MR. ARNDT: It was across --

DR. GUARRO: It was one of the MIL Standards that was discontinued in the acquisition reform era in the '90s.

MR. ARNDT: The primary idea is not only looking at what's available, but what are the underlying assumptions in the databases that are available. So understanding what's in there, both what you can use and what you can't use.

As Professor Guarro said, that particular one is not particularly useful for the regulations.

We looked at both significant major type issues, like the Airbus crash and the Therac and other large issues, but also looked at what information we've been able to derive so far in various studies. There have been some limited studies over the last five or six years, but look at information from available sources.

This particular analysis looked at all of the different LERs in this timeframe and tried to comb out what failures were digital system failures and try to attribute some level of consequence associated with them, to give us a perspective in dealing with --

MEMBER APOSTOLAKIS: So these reactor trips were spurious reactor trips I hope.

MR. ARNDT: Yes. So --

CHAIRMAN SIEBER: Well, I wonder about that data, since there aren't very many digital systems in existing powerplants right now. That seems very high.

MR. ARNDT: Well, yes, that -- that is correct. The issue you have to understand is, because of the level of detail of the information here, that particular study was done in such a way to be inclusive. So if, for example, the LER discussed potential application or potential root cause, and if any of the root causes included a digital system, then it was included as a potential failure.

So how do I put this to give you a perspective? The idea of this particular study was to try and scope the issue. Are there cases where we're getting spurious drips or initiating events associated with digital systems?

So the input filter on this particular study was not if this system didn't fail, would the event have happened? It was, was there a digital system involved in the initiating event? So it was a broader --

CHAIRMAN SIEBER: Whether it failed or not.

MR. ARNDT: Well, it had to have had an impact on the failure. But it didn't have to be the single initiating event was the failure of the digital system.

CHAIRMAN SIEBER: Let's say a pressure transducer failed and it failed high, which would initiate a reactor trip. Would you call that a digital system?

MR. ARNDT: If it was --

CHAIRMAN SIEBER: If the signal processing was digital?

MR. ARNDT: In this study, yes. Understand, this was a very generalized scoping-type study, but it did -- the biggest issue is that the LERs contain digital failures, and you can get some information out of it, is the point you should take away from this particular example.

There is information -- one of the big issues is it's difficult at the level of detail of LERs to make that distinction, because that -- the situation you just described may have been the event, but the description in the LER might have been the digital feedwater system failed.

The reason it failed -- it may have been because the pressure transducer associated failed, but we just didn't have that level of information in the LER.

CHAIRMAN SIEBER: All right.

MR. WHITE: I think the Chairman has made a very interesting point, and maybe you're mining this data already. But one of the questions that leaps to mind is, out of all the LERs you looked at how many of those systems -- how many of those plants had digital systems that could have contributed, so we could get to the -- I think the point the Chairman was making.

Does it look like 20 percent of all the digital systems that could cause failures have been causing failures? Or 25 percent? Or five percent? And I didn't know if you intended to look at that a little -- have you already looked at it? And if not, are you planning to?

MR. ARNDT: One of the studies that we did was looking at, of these kind of failures, what systems are failing, both slicing it -- associated with the kinds of plants, the kinds of systems, and things like that.

This data is a little bit old now, because things are changing more rapidly now, so the usefulness of this particular analysis is becoming less and less effective, because these are mostly older systems, many of which are starting to be replaced now with newer digital systems.

But yes, that particular slicing of what was failing, why -- what kinds of systems were failing, were they safety systems or non-safety systems, were they feedwater systems or other systems, was done in this cut.

MEMBER ROSEN: Could you tell me a little bit about Therac-25, 1985 to '87? Is that what -- what is -- I mean, the first bullet I don't understand at all.

MR. ARNDT: The Therac system is a very well-known digital system failure. It's not a real-time system. It was a therapeutic irradiation system in Canada. It is very well-known, one, because it killed people, but, two, because it was a classic example of a lot of the problems that hopefully have been solved by now in terms of software requirements and not understanding software-hardware interaction, and issues like that.

It was basically a software -- a set of software that ran the therapeutic irradiation device --

MEMBER ROSEN: Okay.

MR. ARNDT: -- that irradiated the patients. And because of the way the software was written, and particularly the way the software revisions were done, had some inherent flaws in the software. And as part of their revisions, they put more and more safety functions into the software and took them out of the hardware interlocks. And this is a classic example of all of the bad things you can do in software design, and it killed people. So that's one of the more --

MEMBER ROSEN: Okay.

MR. ARNDT: -- significant events.

MEMBER APOSTOLAKIS: Did you discuss the last bullet while I was out?

MR. ARNDT: Yes.

MEMBER APOSTOLAKIS: Poor timing.

MR. ARNDT: Again, we looked at --

MEMBER APOSTOLAKIS: But wouldn't that

argue against treating the software as a separate entity with its own failure rate? I mean, you say it was divided, so --

MR. ARNDT: Yes, it would.

MEMBER APOSTOLAKIS: It would.

MR. ARNDT: As we mentioned while you were out, the study was limited, and the amount of information you could obtain from it.

MEMBER APOSTOLAKIS: Okay.

MR. ARNDT: But yes.

CHAIRMAN SIEBER: It's not clear to me that what you would get out of LERs you could even tell whether it was hardware-software or human interface.

MR. ARNDT: Again, as we discussed, what we could tell gave us that -- it was a limited study, one, because of the timing --

CHAIRMAN SIEBER: Okay.

MR. ARNDT: -- of the dates we looked at, as well as the amount of information you can get.

CHAIRMAN SIEBER: Okay.

MR. ARNDT: This is basically just a discussion, again, of what kind of things we were looking at for the larger databases. The point in particular is that these databases make certain assumptions about the data that's in them.

And one of the biggest issues with using these data, or other databases, is you really have to understand the assumptions associated with it, because they're making estimations and they're making assumptions based on a particular model in mind in most cases, be it a reliability growth model or a straight amount of failures per time in service, or whatever.

And one of the biggest challenges in gathering and combining data is understanding these issues and being able to deal with them.

MEMBER APOSTOLAKIS: How extensive is the review that something like ISO 9000 gets? I mean, is it something that has been really reviewed by competent people so I can -- we should take it seriously? Or is it something that some committee somewhere developed?

I mean, certification to estimate software mean-time to failure -- wow. That assumes that there is such a thing as a mean-time to failure.

MR. ARNDT: Yes.

MEMBER APOSTOLAKIS: There is such a thing as a reliability growth model. Has anybody questioned those things? Have they convinced themselves that, yes, this is a reasonable thing to do?

MR. ARNDT: The short answer to that question is yes.

MEMBER APOSTOLAKIS: Well --

MR. ARNDT: The longer answer, and probably more appropriate answer, is that the people who are using the particular model -- in the case of the ISO 9000 software model or the capability maturity model for -- or whatever model they're using, are by and large people who have a similar application background, are doing it for a particular reason. And they have convinced themselves that for the particular application that they're using it's acceptable.

As you'll recall when we briefed a couple of months ago about the validation and verification program, there is a lot of different verification and validation programs out there. The one that the NRC endorses for real-time systems is the IEEE 1012, which is -- with the various levels, which we basically say for a real-time system has to be at the highest level.

But there's a lot of other people out there that do this work at different levels using different methods, and for the particular application that they are working with they are -- they have convinced themselves, either by standards committees or by regulation in a particular domain, that they're comfortable with this particular model.

MEMBER APOSTOLAKIS: Okay.

DR. GUARRO: In the review of major software-induced or related failures, have you looked at the ones that have occurred in the space systems? Recently -- there was a string of recent -- recent ones, in '98/'99 timeframe.

MR. ARNDT: I don't believe that was part of our review.

MEMBER BONACA: Was it in the United States, Sergio?

DR. GUARRO: Delta -- the Delta 3, first flight; the Titan 4, 820 flight; and then a couple of spacecraft failures. And they -- some of those were -- as was mentioned before, you know, the software was the messenger of a serious design problem. A couple of those were actually errors in entering parameters. So there is quite a bit of interesting material there to look at.

MEMBER KRESS: Are we going to hear a discussion on the concept of mean-time failure for software? Because I was under the impression that that's predicated on the basis of random failures, and a question I would have is: how do we attribute random failures to software? And --

MR. ARNDT: I wasn't planning on going into a detailed discussion of that particular issue, beyond the issue that to do that kind of analysis you have to make the assumption, one, that that makes sense, and that you can come up with a failure rate, if you will, for software. And there is argument in the field associated with whether or not that makes any sense.

It basically comes down to the fact: can you model software in that way? From a theoretical standpoint, it's pretty obvious that software doesn't have a failure rate. But the real issue is: can you model it that way in a meaningful way, and treat it separately in a fault tree analysis or some other analysis?

MEMBER APOSTOLAKIS: Won't later presentations address this?

MR. ARNDT: It will address it to some extent, yes.

MEMBER APOSTOLAKIS: Yes. So maybe we can -- yes.

MR. ARNDT: Again, the assumptions in the actual data is a particular issue. For example, we talked about earlier fault tolerant systems, which is an important understanding of how it fails or doesn't fail or gets -- it is -- you're not going to see that in most databases, because it either fails or it doesn't fail, and that's what in the system.

Redundant channels, the same kinds of issues. In a lot of failure databases you cannot extract that level of information. It's one of the reasons that looking at -- sorry.

MEMBER ROSEN: I wanted to -- go ahead, finish your thought. But I wanted to ask another question about the prior slide -- 44.

MR. ARNDT: One of the things that we're trying to evaluate is whether or not it makes sense to have a real-time nuclear-specific database that addresses the specific issues we have -- whether that is a meaningful, cost-effective, rational thing to do.

MEMBER ROSEN: This first bullet under the -- on this slide, the failure rates were estimated by dividing the number of reported failures by the total operating time, it can give you a lower bound, but it's surely not, you know --

MR. ARNDT: Right.

MEMBER ROSEN: -- there's lots of other failures that --

MR. ARNDT: Right.

MEMBER ROSEN: -- that happen that are just simply not in the database. So if you treat that as a lower bound that's okay. But otherwise, you're making a mistake.

MR. ARNDT: Yes. And the point here is you have to understand these underlying assumptions to be able to utilize the data.

The tentative findings from their review basically are things that we've talked about before. Quantitative methods for assessing software failure is something that works, that we need to be able to deal with this.

One particular methodology that they looked at was a Markov modeling at the processor level, and the idea was: is that an acceptable standard to put -- draw your line at? Is that good enough?

Looking at the fact that, of course, probably that level of detailed analysis of failure modes to be able to support the analysis from a PRA standpoint -- it goes back to the concept that just having a failure rate doesn't necessarily make the model work. You have to understand -- you have to have the deterministic analysis of how it fails, and things like that, to be able to support it.

And, of course, data is needed to really understand this, to be able to model these kinds of issues.

MEMBER APOSTOLAKIS: So there is a conclusion, then, that the concept of a failure rate is meaningful here.

MR. ARNDT: It can be meaningful.

MEMBER APOSTOLAKIS: Well, I guess that's a major issue. Sometime we have to discuss this. I don't know whether it's today or some other day.

MR. ARNDT: If we have not discussed it appropriately by the end of today, the we'll revisit it.

MEMBER KRESS: Is that failure rate driven by the rate at which the input carries the software into some error mode? And it's really the rate at which the software -- the input --

MR. ARNDT: It can be looked at in a number of ways. That's one way of looking at it. The likelihood that given the operational parameters that it's --

MEMBER KRESS: But you will enter into a combination of inputs that --

MR. ARNDT: Right.

MEMBER KRESS: -- exercises some part of

the software that has an error in it.

MR. ARNDT: Right. And that's why I mentioned earlier software failure probability, in and of itself, is something of a misnomer, as George has pointed out many times, because software has to run on something. I mean, it can't independently do that.

The issue is: can you --

MEMBER KRESS: It seems like a real stretch to consider that as a random failure.

MR. ARNDT: Right. And can you model it -- one of the big issues is: can you model that independently of its software -- hardware interactions? It means you have hardware failures, you have software failures in that -- the operational condition on the hardware has exercised a software failure, and then you have the interactions between hardware failures and software failures, which, depending upon what model you use, can be modeled separately or can't be modeled separately.

And one of the things we found, particularly in Dr. Johnson's work, is that a lot of their bad failures are exactly that. It's the interaction between hardware and software failures.

Let me quickly go through some of the other work we're doing. We have two other database efforts. One is the international effort to develop a software -- develop a database of -- and this is actually a typo. It should be computer -- software-driven computer system failures. It's not just software failures. It's all failures in systems driven by software in the nuclear industry.

And this is what I mentioned a minute or two ago. We're currently evaluating whether it makes sense to have a nuclear domain specific database with all of the kinds of information that you need to be able to make rational judgments.

MEMBER KRESS: Is this being done under the --

MR. ARNDT: This is being done under the auspices of NEA.

MEMBER KRESS: NEA. Okay.

MR. ARNDT: It's a CSNI project.

MEMBER KRESS: Yes.

MEMBER APOSTOLAKIS: So you are a member of that?

MR. ARNDT: I am actually the Chairman of it.

MEMBER APOSTOLAKIS: The Chairman.

MR. ARNDT: And this is actually Computer Systems Important to Safety. This is the abbreviation.

We also started an in-house effort to do this. Depending upon where we go in the future -- and that will be decided this year when we redo our research plan -- my guess is we're probably going to fold this either into the Brookhaven effort or the COMPSIS effort. But we have an in-house effort to look specifically at the data.

There are several other efforts going on. The committee is very aware of the Halden research program. That's a collaborative NEA program that looks at a whole bunch of different issues -- human reliability, human factors, fuels, materials. They also have a piece in digital system safety.

And in the last two or three years they have expanded their digital system safety research program extensively. One piece of that is a reliability program, and they are particularly looking at risk assessment of COTS systems and how do you deal with the fact that it's a black box and you can't get at the information, and things like that, what kind of models can be used.

Human system interface issues dealing with software and these things in an integrated fashion -- they, of course, have done a lot of work in human reliability and human factors. So that's a natural fit for them.

And the last of their major programs is the Bayesian belief network to help integrate systems, and that's the one we're dealing with.

MR. WHITE: Excuse me, Steve. This is an example of one of the concerns we had in the National Academy study, and you may have -- you may have alleviated a concern. But the concern that the panel had is that a lot of really interesting work and work that could be very influential in what you do was not subjected to peer review, and it was back to the old concern of the nuclear industry just talking among itself.

So have you made much progress in that area of getting more open review of the Halden work?

MR. ARNDT: That, as you mentioned, has been an open issue throughout the work. We are trying a lot -- not only in this work, but all of the work in the I&C area, to do more of that. And Carol and Barry will mention that in their presentations, and I'm trying to get out there more and our other researchers are.

In the case of the Halden work, in particular -- this is a challenge because of the proprietary nature of their reports. However, they have made a specific conscious decision at the last -- I guess two years ago management meeting to do more peer reviewed literature work.

They've made progress. Is it as much as I would like? No. But they have made progress in doing that work. They are publishing certainly a lot more in peer reviewed proceedings, in the journals not as much as I would like, but they are making progress. At least they're doing much more in peer reviewed proceedings to both put out the work they're doing and also get feedback on the work they're doing.

MR. SYKES: I have a question.

MR. ARNDT: Yes, sir.

MR. SYKES: Before you showed us that eight percent of LER contained digital I&C failures, and nine percent of our PS. And that was for a period of time '94 to '98. Do you have a sense that there is a trend of decreasing failures in digital systems?

MR. ARNDT: In that study, we tried to look at that particular issue. And we actually -- one of the things we tried to look at was how recent was the system implemented.

MR. SYKES: Okay.

MR. ARNDT: We didn't get -- we weren't

able to get a statistically significant interpretation one way or the other.

MR. SYKES: Okay.

MR. ARNDT: The anecdotal data from reading the LERs was that that was the case. When a new system was introduced, the failures were high for a period, and then they started reducing. But we didn't have a statistically significant amount of information to make that determination.

The issue, of course, is more complicated than that, of course, because analog systems tend to have a much longer lifetime in the plant. The systems that we actually were studying in that time period are already starting to be ripped out and replaced with newer digital systems.

So it is comforting to know that as we get more experience with these systems that their failure appear to be being reduced. The mitigating issue is that their lifetime in the plant tends to be much shorter than previous systems.

This is just a quick other effort we're --we've been asked to -- the Committee for Safety of Nuclear Installations is becoming more interested in this area, and they're talking about having a new working group in this area or more international work in this area.

The NRC is also holding discussions about starting an international program in this area, similar to camp or something like that. So we're continuing to work both externally in other industries but also internationally within the nuclear industry.

This is just a quick summary of what I said. It's basically a reiteration of the -- what I hoped would be the conclusions.

We have various programs in this area. We're looking at various aspects -- data, guidance, failure methods, and reliability. We're working on the development, and you're going to hear more about this from Barry and Carol.

The U.S. industry is moving in this direction to say ahead of that. This, of course, is an open debate. We believe that the methodology is such that we can make assessments that are sufficiently mature. Hopefully, by the end of the day you will have more information to agree or disagree with that.

There are significant strengths and weaknesses of the current methodology -- what's being used there, as well as the issues that we're proposing. Our future work is going to be looking at the actual integration type issues, as well as development of testing methodologies.

One of the biggest issues, as I think was discussed earlier, is because this has not been used extensively in the nuclear domain, we need to validate the models as we develop them, at least as well as we can based on the available data.

Additional data, additional coordination is something that we need to continue to do.

MEMBER ROSEN: Are you thinking about ultimately having a pilot with somebody to -- who has an existing plant and PRA and might be willing to at least try to put into a research version of the model -- obviously, not the model they're using for plant, but, you know, put a Rev model out there for research and try to do some digital systems stuff in it?

MR. ARNDT: One of -- you'll hear later today what we're doing in our validation work is actually using real nuclear applications in our validations. We did a study on the Calvert Cliffs feedwater system, the actual system that they're using. We're looking at some -- several other programs to test the methods using nuclear-specific applications.

The second half of your question is the

actual application of that to a regulatory structure, and we're not that far down that path.

MEMBER ROSEN: Well, I'm not thinking so much about a regulatory structure. I was thinking about once you get some ways to integrate digital system reliability into PRAs that you think are doable --

MR. ARNDT: Right, yes.

MEMBER ROSEN: -- to find someone who is willing to work with you --

MR. ARNDT: Oh, yes.

MEMBER ROSEN: -- in the existing industry to do it.

MR. ARNDT: Absolutely.

MEMBER ROSEN: Try it, to see what it does to the event trees and the fault trees, to see how hard it is to incorporate it into models in a coherent way, to see what it does to the CDF, depending upon what kind of input parameters you use, to see how it -- it works in terms of if you need to update -- you know, all of the operational --

MR. ARNDT: Right.

MEMBER ROSEN: -- and implementation issues.

MR. ARNDT: Operational issues.

MEMBER ROSEN: In other words, take -- don't just do the research from an academic point of view. Take it out beyond that to an actually -- if we were to do this, this is the way it would behave in the field.

MR. ARNDT: Right. And later in the day when I talk about future projects, that's one of the future projects we have specifically is to do some pilots with particular models in particular PRAs, either ones that we have for other regulatory reasons or doing it ourselves with the information that we currently have, or with --

MEMBER ROSEN: I don't know if the SPAR models are a good enough platform for this.

MR. ARNDT: No. We have access to actual plant PRAs --

MEMBER ROSEN: Right.

MR. ARNDT: -- in some cases. And we would use those.

MEMBER ROSEN: But maybe you could do it in-house or -- but I would never try to use a plant PRA without talking to the plant's PRA people.

MR. ARNDT: That would not be the preferable method, no. That --

MEMBER ROSEN: I mean, you can do it.

MR. ARNDT: Yes, that is one of our --

MEMBER ROSEN: The part of this that I'm aiming at is not just the doability, but the confidence-building measures --

MR. ARNDT: Right, exactly.

MEMBER ROSEN: -- in the practitioner community, in the nuclear PRA domestic practitioner community.

MR. ARNDT: Exactly.

Okay. I think that's all I'm going to say for this particular minute. I'm going to turn it over to Barry. What I will -- what I'd like to -- next up on our agenda is Professor Barry Johnson from the University of Virginia. As I discussed earlier, he is leading work in digital systems modeling using the fault injection method. I will let him provide some additional input on his background and get right into the program.

MEMBER APOSTOLAKIS: Either you sit down or we'll have to put a mobile microphone on you if you want to stand up. Do you prefer to stand up?

DR. JOHNSON: I can do either one. I'd like to stand if it's okay, but I -- I don't have to. I'll sit. That's not a problem.

MEMBER APOSTOLAKIS: We're getting you a

microphone.

DR. JOHNSON: Not a problem.

CHAIRMAN SIEBER: I think it might be over there. You have to turn it on, too. There is a --

DR. JOHNSON: Test, test.

CHAIRMAN SIEBER: You just swallow the microphone, and that will do it.

(Laughter.)

DR. JOHNSON: Is this okay?

CHAIRMAN SIEBER: Yes.

DR. JOHNSON: Okay. Well, I would like to preface my talk with a couple of things. One is just to thank you for the opportunity to be here. I enjoy talking and interacting with groups of this sort. I find I learn more perhaps from you than you learn from me, and I certainly appreciate the opportunity to be here.

The second thing is Steve had asked us to give a little bit of our background as a way of a very brief introduction. I started my career in the aviation industry. I worked for Harris Corporation where I designed flight control systems, and, in fact, did the safety assessment for several flight control systems during that part of my career.

And, in fact, it was the genesis for many of the ideas that I've been pursuing in the academic environment for the last 20 years. I joined the University of Virginia in 1984 and have continued research in this area since then, and have come up through the ranks at the university -- 1989, was promoted to Associate Professor, and in '94 to a full Professor. So I've been there since that time.

The third comment that I'll make, just as a way of introduction, is I apologize -- I am suffering from a fairly severe cold that has worked its way from my sinuses through my chest. I'm on the tail end of it, but my voice will crack, and so forth, during the course of the conversations, and I apologize for that. I'll try to make sure that I speak as clearly as physically possible at this point.

My contact information is on the first chart. Again, please feel free to contact me if you have any questions or comments as we go forward from today.

Several things we'd like to cover in the outline that I think Steve had indicated you preferred for these types of presentations. I'll start with some conclusions. The important thing to note about that is that they really are conclusions for this talk. This research is ongoing. We find we almost generate more questions than we do answers sometimes, and I think that's a good thing. But, you know, the conclusions will focus on the talk.

I'd like to talk a little bit about the objectives of the program that we have, a little bit of background, some of what we see as the challenges, our methodology that we've been working on and that we've applied in several cases, along with the process that that involves. And we have -- we have shown this in some real applications. I'll talk more specifically about them.

They are predominantly transportation applications, but there is a lot of similarity between advanced training control systems and some of the reactor control systems that are in place.

We've used this in Los Angeles, we've used this in Copenhagen. We're currently using it in New York. We're using it in Illinois. We're using it in Pittsburgh. We have several projects that have gone from cradle to grave with the methodology, at least as the methodology existed at the time that we went through that process.

And as a result of that, we've gotten some real hard critical review of it from the safety assessment organization TUV in Germany, as well as some independent consultants that were hired by Los Angeles for the Metro green line transit system. So there's a lot of good information I think that has come out of that, and then we'll summarize and move forward.

I'm one of those people that believes in looking at the integrated hardware-software system. I'm a firm believer in that. I don't think that precludes things that you might do in software alone, or things you might do on the hardware alone.

But as Steve has pointed out several times, ultimately the software becomes a collection of bits that get loaded into memory and they get executed by hardware. And the interactions between those two things influence a lot of what happens, and that's where I focused my work is on those interactions.

So we have been looking at techniques that can be applied to integrated hardware-software systems, real software running in some cases on real hardware, or real software running on a model of the hardware. And those are two things that will be prevalent in the discussions that we'll talk through.

I mentioned the process has been applied. TUV is one of the organizations that spent an incredible amount of time actually looking at this, and I'll talk about that in a little bit more detail later on. But, you know, we developed at UVA the analysis that was done for that Copenhagen system.

There was a set of documents created for each step of that analysis, and the experts at TUV reviewed and critiqued each one of those. And, in fact, they were iteratively developed over the course of a couple of years of that critique.

We've talked about -- this has come up a couple of times so far today about assumptions. In both the Los Angeles application and the Copenhagen application, we have an entire document devoted to assumptions. And every assumption is documented, every assumption is discussed, and every assumption is evaluated as to the consequences of that assumption either holding or not holding. It's an important part of the process.

Currently, we're looking at new ways of modeling. For example, the issue of COTS has been mentioned. One of the things that we are looking at and that you'll see a little bit later on is, how do I take an application-specific integrated circuit that unbeknownst to me may have a hardware-software system inside it that's executing certain implementation of the protocol that that particular ASIC has provided.

And I may not even know the internals of that. I only know the interface of that. Is there any way that I can model at the interface the things that could happen if something goes wrong inside the chip? And that work is actually being funded by Electricite de France, which is the electric utility EDF in France. And so that work is something that's currently ongoing, as well as additional things that are involving the new statistical models and other types of things.

And that really covers some of the COTS work, but the point of this bullet really is that we've, over the years, developed a lot of tools. And one of the things we learned very quickly is nobody really wants to use university tools.

Universities don't have very good ways of supporting those tools. Students come and they go. We have a built-in turnover of our workforce every, you know, two to five years, depending upon whether you're dealing with Ph.D. or master's students. So it's difficult to produce tools that are up to quality and have the support necessary to be used in industry.

So what we've been doing in the last few years is trying to take the techniques that we've developed and integrate them into commercial tool sets. Simics is one that we've been using recently. We've used the Mentorgraphics tool set, the Cadence tool set, a lot of the system-level design, and, in fact, complete design capabilities in those tool sets. We've tried to integrate our tools into that, and we've had some success in that.

And then, lastly, we have started to look not just a rail applications but at nuclear applications. Calvert Cliffs is the most recent, but we also have an objective to be able to look at -- in fact, I'd love to be able to look at one of the three systems that have been generically approved, and to get that into the lab and to be able to do some modeling and simulation and experiments with that. But Calvert Cliffs' digital feedwater control system is the one that we've looked at today.

What are the objectives? There are several. And, again, just to focus on a couple of them, we've been looking at safety assessment, and, again, for digital systems. And to me -- I'll show you what I mean by "digital system" in a moment. It's not just hardware and software.

It actually involves -- certainly most of the -- well, a lot of the elements are hardware, but real systems involve mechanical components and they involve sensors and they involve other things that you have to worry about as well, even though we focused mostly on the controller parts, which are, you know, processors and memories and other types of things with software running on those.

MEMBER KRESS: When you say -- excuse me. When you say "safety assessment," what exactly do you mean by that?

DR. JOHNSON: Essentially, what we mean by that is the -- a process by which I can look at the safety of the system -- and it involves both quantitative and qualitative issues. But most of our work has been driven by an attempt to quantify the safety, to be able to put a probability of occurrence, you know, on an unsafe event.

MEMBER KRESS: Okay.

DR. JOHNSON: And the use of the execution of this system. That's what we focused on. And by a methodology, it's a -- just a sequence of steps that we go through to try to get to --

MEMBER KRESS: I understand what you mean now.

DR. JOHNSON: Okay. Modeling simulation and experimental techniques -- one of the points I want to make here is that, you know, sometimes I'm accused of focusing too much on the quantitative side of things, and I believe you need both. I think there are process things you need to do. I think there are, you know, things that you need to have in place that allow you to -- you know, to have a successful development enterprise.

But I also believe that there are quantitative things that are important. And, in fact, in the systems we've done in the past, I've learned more by just the process of trying to get to a number than perhaps I learned from the number itself. So I do think both qualitative and quantitative things have to be a part of it, and we've tried -- even though a lot of our work focuses on the quantitative, we've tried very hard to not lose sight of that fact.

I've mentioned tools. We've created a bunch of them, and I'll show you some of those over the course of the presentation. But we are trying to use COTS tools and design systems where possible, because ultimately -- and you'll see this at the end of the presentation -- but ultimately the best way for a real safety assessment to be done is for it to start the day you start developing a system, and for it to be an integral part of the design of that system with certain products at various points that can be reviewed as evidence of what was done at that particular state in the evolution -- or step in the evolution of the system.

So that's important, and then demonstrating it. And, you know, again, we've worked with Boeing quite extensively. We've worked with the rail industry extensively -- NASA, nuclear obviously, and some others. Medical is another area that we've been pretty heavily involved in.

MR. WHITE: Excuse me, Barry.

DR. JOHNSON: Yes.

MR. WHITE: I think this is really exciting work. One question that I have is: what size of system have you been able to analyze to date? In other words, do you think you'll ever get to the point where you can actually do a complete reactor I&C system? Or do you think you're probably going to be down in the system level or subsystem level?

DR. JOHNSON: There are -- I'll give you a couple of examples. The Los Angeles system that we analyzed was a -- what's called an interlocking, and it -- if you've ever ridden the Washington Metro, we have a station that you stop at, and you can have trains on both sides.

Approaching that there are points for

trains to cross over track. And then when you leave they can cross over, and that interlocking consisted of six boxes. Each box had two processors in it. Each processor was executing approximately 100,000 lines of code. The boxes were all interconnected with a network, and it was -- in that case it was an optical network, but it was a serial optical communications path.

And then they also interconnected with sensors that were placed along the track, and then they had an interface with another communications box that was a wireless network that allowed you to communicate to trains and other points that were remote from that system. So that -- that's a rough illustration of complexity that was looked at.

In the Copenhagen system, it was actually much more complicated than that. The number of lines of code in that particular system was just a little bit less than a million lines of code that were involved in that. There were on board each of the cars -- were 20 processors, and you have two main pieces of the system.

There's what's called an automatic train operation system that actually controls in a driverless fashion all of the starting, stopping, acceleration, velocity control, opening of the doors, and so forth.

And then you have something that's called the automatic train protection system, which is somewhat analogous to the reactor protection system that overlooks all of the system, measures certain things, and makes decisions on whether something has gone awry, and then shuts the system down if something has, by using emergency breaking if something has gone wrong.

So those are the -- that hopefully gives you a little bit of feel for the type of complexity that we've been looking at. Now, I guess the last example -- the Calvert Cliffs system is a commercial off-the-shelf digital control system, distributed control system.

It's an Intel -- actually an AMD version of the Intel 486 processor. It's running, you know, Windows. 3.1 is one of the -- is the operating system that's running at least portions of it. So that's -- and then the application is running on top of that.

CHAIRMAN SIEBER: And what's the scope of control for that system?

DR. JOHNSON: For Calvert Cliffs?

CHAIRMAN SIEBER: Yes.

DR. JOHNSON: I'll show you a diagram of that a little bit later on. I mean, it's essentially controlling the level of water in a tank, and controlling the valves.

CHAIRMAN SIEBER: Is this feedwater?

DR. JOHNSON: It is feedwater, yes, sir.

CHAIRMAN SIEBER: Okay.

DR. JOHNSON: And I -- yes, I have to state right up front I'm not an expert on nuclear systems in terms of the applications, and so forth. I'm a hardware-software guy. I'm an electronics guy.

CHAIRMAN SIEBER: I'm familiar with that system.

DR. JOHNSON: Okay.

CHAIRMAN SIEBER: I'm disappointed that it's using Windows.

DR. JOHNSON: It's Windows 3.1, which actually was -- that was one of the difficulties in doing that. I mean, finding a copy of Windows 3.1 or anything associated with it is a --

MEMBER KRESS: You can have mine.

(Laughter.)

MEMBER ROSEN: With all of its software problems. It keeps telling me I've done an illegal operation.

CHAIRMAN SIEBER: That's the least of your problems.

(Laughter.)

DR. JOHNSON: So those are the objectives. Just a couple of points that I want to make with this slide. One is that, you know, these systems are incredibly complicated, and that's one of the things that makes it so difficult. The area that we focus on at UVA is really what's inside the dotted line.

I say that simply to point out that, you know, that obviously there are human beings involved in these systems. There are, you know, complex mechanical and civil infrastructures that are involved in these systems. We don't focus on those activities. What we focus on is really the sensors and actuators that make up the control system, analog hardware that's interfacing to those, digital hardware that's interfacing.

But predominantly -- and importantly from our standpoint -- is the hardware-software system that is executing the control algorithms and other things that are being used to make the system happen.

MEMBER ROSEN: Now, when you say you focus on the -- I think you said dotted line, is what that you said, or --

DR. JOHNSON: The two dotted lines.

MEMBER ROSEN: -- was it dashed lines? Which is where --

DR. JOHNSON: Yes, the two dotted lines.

MEMBER ROSEN: -- you're focusing?

DR. JOHNSON: I apologize. We focus on what's inside the big box.

MEMBER ROSEN: Okay.

DR. JOHNSON: And now --

MEMBER ROSEN: We had an earlier question -- Jack did -- about the data that Steve was presenting about digital system failures in nuclear plants between the years 1994 and 1998, and whether it was really on the inner box or whether it was within the outer box. And I think we -- we concluded that a lot of the failures in that database were outside the inner box but inside the outer box. In other words, they were in sensors or things like that.

DR. JOHNSON: I'm not surprised. I didn't know that, actually, but I'm not surprised based on what we've seen in some of the other -- the other cases.

MEMBER ROSEN: Those databases basically just say, "Bang. Everything inside those -- that dashed box, the outer box, is a digital failure." And

I'm not sure about what, but maybe.

DR. JOHNSON: I mean, the thing that complicates this a little bit even more is -- I mean, this is an oversimplification of it, because sometimes sensors nowadays have embedded processors in them, and a lot of things are going on in there from a hardware-software standpoint. But these are the types of systems that we focus on.

The other point I wanted to make with this is just the -- you know, the concept of interfaces. And, you know, when I first started my career one of the things that I did a little bit in was hardware testing. And one of the things that you commonly found was that, you know, you could have a piece of hardware, and it would pass every test you could expose it to.

But then when you put it in a system, and it had to interface to other things, it started failing. And it was because of that interface, and the interaction between those components, that created events that you hadn't really anticipated in your test process.

And we've actually found that in the hardware-software interface as well. The things that happen in the hardware that exercise features of the software -- that maybe you hadn't completely tested or you hadn't envisioned being exercised in that way, and vice versa. And you end up with some interesting things happening there that you might not have anticipated.

So the interfaces are critical, and I think that's --

MEMBER ROSEN: Right. And I think your point is it's more complicated even than what you're showing here.

DR. JOHNSON: Absolutely.

MEMBER ROSEN: And one example is an actuator that goes to a new position under control of the software, sends a signal back to the software, saying, "I have reached the position you sent me to." And now that's -- so you have another feedback loop inside from the actuator circuit back.

DR. JOHNSON: That's exactly right. In fact, if you look at -- for example, one of the systems we've looked at are the turbine control system. And they actually use what are called, you know, flux summing actuators.

They actually have multiple drives going into them, and then they have an electromagnetic summation process that occurs there. And then there's feedback from the actuator that goes back into the digital controller, and you compensate, you know, based on, you know, where you think you're driving it versus where it is. And you've got a lot of interactions that are going on there.

It is incredibly complicated, and I don't want anybody to -- you know, to misunderstand our work in the sense that, you know, we're not claiming to have solved all the problems. And, you know, we're focusing on the things we feel like we can get a handle on and that we can make contributions to. But it is a complicated system. Even the simple systems are complicated systems.

MEMBER ROSEN: Well, I think the important point here is that you've made a contribution just by drawing this chart to me. But I think no one should go away with the understanding that that's the picture. And we do need to have -- when we do the real stuff, we need to have the whole picture, not just a model of the whole picture, which is what this is.

DR. JOHNSON: Yes. I understand.

MEMBER ROSEN: Because it's as you say, and as I say, somewhat more complicated than this. And those complications can affect the outcome.

DR. JOHNSON: Absolutely.

MEMBER ROSEN: And the risk

DR. JOHNSON: And, again, this is all in background. The other thing that's important that I think is sometimes forgotten in some of these systems is that most of them are what I would call real-time systems, meaning that they have timing requirements.

And the timing requirements show up in several different ways. They show up in some time that I must be able to read inputs, calculate outputs, and deliver them. You know, I come from an aviation background where you're doing this process 180 times a second. And if you don't get the right answer in the right amount of time, you might as well not get the right answer.

So there are some stringent timing requirements typically. The other place that the timing comes into effect is when an event occurs -- and, again, we look at what happens when a fault or some event occurs, and how does your system respond to that.

And typically you have some requirement on how quickly you have to be able to respond. You have to be able to identify the problem, remediate it, or mediate it somehow, and reconfigure the system and keep it running, or shut it down, or do something.

And, for example, again, in the aviation industry, you have about 500 milliseconds typically to do a lot of that. And if you don't, then the dynamics of the aircraft are such that you start noticing problems and can get catastrophic results if they're not taken care of quickly.

So timing is an issue. And, again, this is another reason that we look at the integrated hardware-softwares, because that integration is -- it can have a big impact on timing, and that's I think an important issue.

I've have students, for example, write programs for real-time systems, and then, you know, after we talk about them they go back and write them again. And you can -- you can change performance of those by an order of magnitude, just based on how you do things in writing your software. So it's an important issue.

So real time -- complex systems, real-time requirements -- again, this is an oversimplification of it, but I've got just a couple of points here I want to make with this. And Steve actually has made several of these.

But the first point -- and this doesn't

attempt to show everything that's involved in the evolution of a system or the design and operation of a system -- but a couple of points that I want to make. I mean, there are things that happen in the operation.

Once the system is out there and it's running, there are things that happen that are due to, you know, components just failing. I mean, hardware just dies sometimes. Operators make mistakes. They enter parameters incorrectly or they make wrong decisions. Or external disturbances -- I mean, the biggest problem you have in an airplane is lightning.

You know, if you solve the lightning problem, you've solved a lot of your other problems typically in terms of external disturbances. So there are a lot of things that happen.

In the development there are a lot of things that happen, and this is just a subset of them. But, you know, you can misunderstand requirements or make mistakes in creating those requirements or have incompleteness in those requirements.

One of my colleagues likes to refer to the completeness problem, and that's the way he sums up the whole issue. How do I know that things are complete in terms of understanding whether I've got sufficient requirements and sufficient testing and other types of things?

I can make implementation mistakes, and these things can lead to problems in either the hardware or the software or both. You know, for example, Intel will acknowledge that there are 79 design defects in the Pentium processor, 39 of which they've chosen to fix. You know, some 40 that they've chosen to ignore, because they occur so infrequently that they are normally not an issue, and we use it successfully every day. But there are design defects in that process.

So you can have design defects here. You can have random defects, randomly occurring failures there. You can have design defects here. These can interact with one another. I've actually seen examples of systems where a bug in the software did some things to the hardware, activating things that should not have been activated simultaneously, and actually burned out a portion of the system, creating, in effect, a hardware fault due to the occurrence of a bug that was in the software.

So those things can interact with one another. You can also have corruptions in your data structures that make up your system, and these --

again, you can have all of these things leading ultimately to what I call a failure, which some people call a malfunction, but it's fundamentally just a -- you know, a non-performance or an incorrect performance of something that the system is supposed to do.

MEMBER APOSTOLAKIS: But it seems to me that you are making now a very strong case for looking at software as part of the system and not in isolation.

DR. JOHNSON: I am.

MEMBER APOSTOLAKIS: So --

DR. JOHNSON: I am. I believe that very strongly. I believe that very strongly.

MEMBER APOSTOLAKIS: No apologies required. You are doing a great job. But now I come back to my earlier question about ISO 9000, where they talk about mean time to failure. And I'm wondering what that means now in this context, especially in light of your last statement, that the software triggered something in the hardware, and then came back and, you know, there was an interaction there.

DR. JOHNSON: Right.

MEMBER APOSTOLAKIS: So how -- I mean, what does it mean to talk about mean time to failure of the software? I can understand maybe talking about the mean time to failure of the whole thing you have there. That might be a concept that would be acceptable.

DR. JOHNSON: Right. And that's --

MEMBER APOSTOLAKIS: But you are making a very strong case for, you know, looking at the whole thing as an integrated whole, which this agency has been doing for nuclear powerplants now for 30 years.

DR. JOHNSON: Right.

MEMBER APOSTOLAKIS: Right?

DR. JOHNSON: And I do believe that. I mean, I --

MEMBER APOSTOLAKIS: I'm sure you are not lying to us, yes.

(Laughter.)

DR. JOHNSON: I do indeed believe that.

MEMBER APOSTOLAKIS: I just wanted to point that out, because this is a question that at least is in my mind.

MEMBER ROSEN: Yes. George, I see that exactly as a mean time to failure for this system is when you hit that box on the bottom.

MEMBER APOSTOLAKIS: Yes.

MEMBER ROSEN: You don't think about what happens before that.

MEMBER APOSTOLAKIS: Yes.

MEMBER ROSEN: But by analogy, also, what I think of is a latent defect in a plant system. For instance, someone makes a maintenance error in setting up a motor-operated valve. And -- or, let's say, tightens the packing too much and he's redoing the packing.

So, actually, when the valve gets a signal to stroke it won't, because it's one bang. That's a latent defect. Now that is exactly analogous, in my view, to the things on the left side. Someone had to put a requirement -- there's a mistake in the requirements.

Maybe there's a file structure that transfers into the processor that the processor wants to have 100 fields filled up, and the processor has 120 in it. So when it tries to transfer the data it transfers 100, and it can only transfer 100, it leaves -- it drops 20 bits, and you don't know what -- which 20 it's going to drop. So it's, you know, that kind of is a latent defect from a -- coming in from the outside.

And so I see those latent defects in a powerplant, the one I described of the valve with the packing, like the one I described in the file. Those are very analogous in my view.

MEMBER APOSTOLAKIS: I was intrigued by what you said about Intel. They decided to leave design faults because they figured that those would be triggered under very rare circumstances?

DR. JOHNSON: Yes. Those are --

MEMBER APOSTOLAKIS: What is rare in their view? Do you know?

DR. JOHNSON: You know, probably the most famous example of one that was found by the general public and created an uproar within Intel was actually found by a professor at Lynchburg College who was doing some fairly complicated modeling simulation applications and he started using -- he started noticing that from his two different Intel processors that were running the same software he was getting different results from the floating point calculations, and they were out in very, very, you know, far out digits, you know, to the right of the decimal point.

And he started -- that was important to him, and he started asking and inquiring, and it uncovered a flaw that -- or a design defect that was in the floating point unit of the Pentium processor.

MEMBER APOSTOLAKIS: So Intel didn't know that?

DR. JOHNSON: Intel did not know that. They had not uncovered it with, you know, all of their testing and simulations and all the things that have been done. And, obviously, they had sold millions of Pentium processors that, you know, had that in it, but it was just this particular person was doing something that was exercising the hardware in a specific way that no one else had really done, or either hadn't noticed. And there are a lot of examples of --

CHAIRMAN SIEBER: In these instances, you know, the regular commercial user would never run into it because there is no software that's commercially available that uses every feature of a Pentium chip.

DR. GUARRO: I can give you an example of a software defect that exists now in Excel. If you go into the beta function and you put a very low value, the alpha parameter you get a 95th percentile higher than the 99 percentile. And it has been there for a long time, to my knowledge, and I don't think anybody worries about it.

MEMBER APOSTOLAKIS: Does Microsoft know this?

DR. GUARRO: I don't know.

CHAIRMAN SIEBER: The question is: does Microsoft care?

DR. JOHNSON: Right.

(Laughter.)

MR. ARNDT: Well, that really goes back to the issue we were discussing earlier. In a lot of cases these are conscious decisions based on the applications that you're doing.

DR. JOHNSON: That's right.

MR. ARNDT: And in the application that they're interested in it's not an issue for safety or for performance or whatever.

MEMBER APOSTOLAKIS: Yes. But the problem that I'm having, though, with that is that it's not so much that, you know, some academic someplace was doing work and found a strange thing that's very rare. It shakes my confidence in the whole enterprise.

I mean, if we use this now in safety critical applications, I don't know -- I'm kind of scared, because, you know -- you know the famous saying there are things that we know we don't know, and things we don't know that we don't know. It's the latter part that really scares me.

MR. ARNDT: Right. It's a knowledge uncertainty issue.

MEMBER APOSTOLAKIS: Yes, that's right.

CHAIRMAN SIEBER: Well, I think that you would have to be carefully considered if you found failure rates for digital systems that were significantly different than the failure rates you get out of analog systems. And then the regulation of it would be easy. You would just write a rule that says don't use digital systems.

MEMBER APOSTOLAKIS: But the point is we're talking about the rare application that this professor was doing. But suppose you are in the middle of a severe accident. That's a rare thing, and now you are relying on some software --

CHAIRMAN SIEBER: Well, I can give you an example of a mistake I made years ago that took a year and a half to reflect itself, which was a -- basically a routine that directed to a series of tables that had to be solved, and where it went in those tables depended on parameters of -- in the powerplant. And it took a year and a half before it ever got to the combination that took it to a bad table, you know. Once it got there, it never came out.

MEMBER ROSEN: What I'm concerned about in your story is that Intel doesn't know what people are going to use the chip for.

CHAIRMAN SIEBER: That's right.

MEMBER ROSEN: And that someday someone may use it for something that invokes the chip in one of those areas that they didn't fix.

DR. JOHNSON: I will not mention the name of the company, but I have a -- there is -- in fact, I had reason to review a supplier agreement not too long ago from a company that makes integrated circuits, and so forth.

And one of the things that I was surprised to find in there was a statement that the -- you know, the customer buying that component is warranting that they will not use the integrated circuit in aviation, nuclear, military, or -- there's a long list of applications where, again, as part of the supplier agreement it was -- you were signing up to not using it. So you were, you know, limiting your field of use. And, in fact, if you chose to use it in those arenas, you were accepting liability for anything that might happen there.

MEMBER ROSEN: Is that common now?

DR. JOHNSON: I don't know how common it is. I -- you know, honestly, I've only had occasion to review a small number of these supplier agreements, so I don't know the answer to that.

MEMBER ROSEN: From a regulatory standpoint, I don't know that we would be too thrilled with having -- with licensees that agreed to that sort of thing.

CHAIRMAN SIEBER: I'm not sure you can get some chip-makers to pay for the cost of a severe accident at a powerplant.

DR. JOHNSON: Okay. The other thing in the way of background that I wanted to just have one slide on is this concept of coverage, because it's really at the heart and soul of what I do. Because, I mean, there -- as you'll see a little bit later on, I mean, there are a couple of issues or questions that you can ask yourself. One is, you know, rate of occurrence of some of these problems.

I know we've already talked multiple times about difficulty of being able to assess what that rate is for hardware-software systems, and I -- you know, it is a very, very difficult thing. And that's not what we focus on in our work. What we focus on is: what if something does happen?

You know, when something occurs, then how capable is the system of responding to that something? Whether it's a hardware-software defect or some other element of the system. And we use this coverage estimation concept as part of the work that's done there.

Coverage can be broken up into several pieces. We normally lump it into one -- you know, one probability, which is a conditional probability. Given that a fault has occurred, what's the probability that your system is going to correctly detect, locate, isolate, recover from that?

And that's this concept of coverage that we talk about, and, in fact, talk about in all of our papers is, you know, what -- given that something occurs, am I going to handle it correctly or incorrectly? And that's really what this concept of coverage is all about.

MEMBER APOSTOLAKIS: Well, let me -- when you say "fault occurs" -- let's go to the previous slide if we can. Now, where in your ovals can that fault occur?

DR. JOHNSON: It can actually occur -- the fault itself can occur anywhere, actually, in these ovals. I mean, it -- you can have a design fault that's in there from day one as a result of something that you've done in the development process. You can have a random problem occur.

On the next chart --

MEMBER APOSTOLAKIS: But these are not necessarily faults. I mean, if you -- if the operator does something wrong, would you call that fault?

DR. JOHNSON: In the world I come from, which is the fault tolerance or dependability community, that would all be considered a fault. Fault is defined as a physical imperfection or defect or flaw in anything -- hardware, software, whatever it may be -- and it's a defect.

MEMBER APOSTOLAKIS: But an external disturbance might be --

DR. JOHNSON: Power loss. Loss of power through --

MEMBER APOSTOLAKIS: Yes, it doesn't have to be a fault. I mean, it can be some external condition which had not been anticipated by the designer, for example. That wouldn't -- would that be a fault?

DR. JOHNSON: It would be -- the external disturbance would be the cause of the fault.

MEMBER APOSTOLAKIS: So the fault was not anticipating it.

DR. JOHNSON: Right. The fault -- for example, I have a lightning strike. The lightning strike induces hundreds of thousands of amps into a conductor, and, you know, because I haven't properly designed for that I get an open or a short or something that occurs as a result of that.

And now I've got a hardware fault that's in the system due to an external disturbance that occurred. Similarly, with an operator -- an operator -- you know, for example, the -- one example was mentioned here of the space application, where the conversion parameters were entered incorrectly.

And to some extent that's a data structure problem. Someone had to enter parameters that were used in that conversion, and they form a database that the hardware and software use, and that's a latent, you know, defect that's in that system that when -- when attempted to use you'll have a consequence resulting from that.

MEMBER APOSTOLAKIS: So fault can be anywhere, including the hardware.

DR. JOHNSON: Absolutely.

MEMBER APOSTOLAKIS: Whoa. Okay.

DR. JOHNSON: Absolutely.

MEMBER APOSTOLAKIS: It's pretty ambitious, though, isn't it?

DR. JOHNSON: We have to focus on what we can.

MEMBER APOSTOLAKIS: Absolutely.

DR. JOHNSON: You know, again, it's a complicated --

MEMBER APOSTOLAKIS: Now I understand what you mean.

DR. JOHNSON: Now, I even hesitated --

MR. WHITE: I'm sorry. Can I ask you a question about the fault coverage?

DR. JOHNSON: Absolutely.

MR. WHITE: You know, the other possibility here is that the fault occurs, the software -- the system never knows that the fault occurs, so there is no fault detection. But still there are no consequences.

DR. JOHNSON: That's right. And we actually -- the community calls those no response faults.

MR. WHITE: Ah, thank you.

MEMBER APOSTOLAKIS: Or they are latent, right?

MEMBER ROSEN: No. Latent fault was one that would -- if you get into the wrong circumstance, like to close, it doesn't close because the packing is too tight.

MEMBER APOSTOLAKIS: Well, but these --

MEMBER ROSEN: But as long as it's open and not -- the system runs fine.

MEMBER APOSTOLAKIS: But these kinds of faults have the same property. They will be there. You don't know they're there until some circumstances will make the fault -- identify the fault.

DR. JOHNSON: And it's -- I mean, part of what's important there is that the way you exercise the system influences that. And, in fact, that's one of the things we found. We've actually -- in some of the systems we've done, we've found bugs in the software that were there from the beginning of time in terms of the system. And they were not discovered until we actually exercised the system in such a way that they were needed -- you know, that the software function that they were in was needed.

MEMBER ROSEN: But isn't that the responsibility of the owner, to give -- it has a certain system -- to test it in all its operating modes so that -- so that even though you only use one of the nine modes typically, if you ever switch to one of the other eight modes, you get all kinds of strange things happening that you didn't anticipate. It's your fault for not having tried that during the setup.

DR. JOHNSON: And the real difficult

problem there is something I mentioned a few minutes ago -- is there is the completeness problem. It is knowing that you have exercised things in enough of the ways that they will be encountering in the real application to be able to state with any confidence that you've covered those types of scenarios.

MEMBER BONACA: But then you -- I mean, can you be sure that you've covered everything?

MEMBER ROSEN: You can't.

DR. JOHNSON: That's a big issue with digital systems.

MEMBER APOSTOLAKIS: In other words, the problem is where it says fault occurs.

DR. JOHNSON: That's right.

MEMBER APOSTOLAKIS: You have to have an envelope of faults.

DR. JOHNSON: And, in fact, the way we approach that is that, you know, because -- I'll give you some examples. But, you know, the envelope that you're referring to is critical, and there are several schools of thought there if you look around the community that does this type of work.

One is that you should do things randomly here, that you should randomly inject faults into the system, choosing random times, locations, you know, characteristics, and other types of things.

There are others that -- in fact, we developed a technique that we referred to as malicious faults. We actually derived -- in fact, this -- we have a patent on this, where there's a technique for taking, at the highest level in the system, the algorithm that is going to be executed by that hardware-software system, and then devising or creating from that what we call malicious faults, which are things that could go wrong in the execution of that algorithm.

And if they go wrong and are not mitigated, they will cause an unsafe action, and then we inject those into the system and determine what the system does in response to those.

So there are multiple schools of thought there, and actually, you know, what we've seen over the years is that you really -- there's a lot -- you really have to do both. You have to do some of the malicious types of things. You have to do random things. You have to do other things as well.

MEMBER ROSEN: If it matters.

DR. JOHNSON: If it matters.

MEMBER ROSEN: I mean, if the coverage -- if a fault is -- if a fault's consequences are important --

DR. JOHNSON: Right.

MEMBER ROSEN: -- you have high risk.

MEMBER APOSTOLAKIS: But you don't know that.

DR. JOHNSON: You don't know that.

MEMBER APOSTOLAKIS: You don't know that.

DR. JOHNSON: Sometimes it is difficult to know that.

CHAIRMAN SIEBER: But putting random faults in almost assures that you aren't comprehensive.

DR. JOHNSON: Right.

MEMBER BONACA: That what?

DR. JOHNSON: That's right.

MEMBER BONACA: That you're not.

CHAIRMAN SIEBER: That you're not.

DR. JOHNSON: It gives you some confidence, but it doesn't give you ultimate -- it doesn't give you complete assurance.

CHAIRMAN SIEBER: Well, it depends on how long you let it run. And this fault injection --

MEMBER BONACA: Either process is -- without the process you don't know.

MEMBER KRESS: I'm kind of interested in

the details here. How do you inject a fault?

DR. JOHNSON: There are several -- in fact, I'll address that actually a little bit later on. There are several ways that you do it. In the -- we are -- we look at both simulation-based approaches and physical experiment-based approaches. And if you are talking about a physical experiment, there are, you know, a couple of ways that you can do it typically.

You can instrument your system, so that you can actually get access to points where you can, you know, control corruptions. You can insert them, you can control the time that they're there, and things of that sort. You can get access to the software in the memory of the processor, and, you know, cause things to change in terms of the software structure, and so forth.

In the simulation environment, you actually have a lot more control over what you can do, because you have access to things that you don't have access to in the physical system. And, again, you have similar types of approaches, though, where you can actually instrument your simulation to allow you to go in and create problems, and then determine how the system responds to those problems.

MEMBER KRESS: Is that like inserting a virus?

DR. JOHNSON: Well, it's -- certainly you can look at it partly that way, certainly. It's -- but if you look at the techniques for fault injection, there are hardware-based techniques, there are software-based techniques, there are simulation-based techniques, and then there are hybrid, which is combinations of those three.

MEMBER KRESS: Well, how normally are these faults detected in the system? Are there detections -- are you talking about a self-correcting system there?

DR. JOHNSON: These are -- the systems that we deal with are systems that have built-in mechanisms for, you know, detecting and managing faults that occur. And, you know, they may have reconfiguration capabilities. They may have shutdown capabilities.

They may have -- you may have one system that's overseeing another system, and you are injecting faults into this system and seeing if the other system actually detects that. So there are a number of different architectures of systems that we've looked at over the years.

MEMBER KRESS: How do they recognize a fault? I'm getting right down to the basics.

DR. JOHNSON: Sure. I mean, you have -- there are lots of different techniques for doing that. I mean, we -- you know, for example, sometimes you use your redundancy as a way of detecting a fault, so you have voting --

MEMBER KRESS: Okay. You're getting voting, and then --

DR. JOHNSON: Loading and comparison.

MEMBER KRESS: I'd like to see how that would work.

DR. JOHNSON: Like, you know, for example, in the Boeing 777 aircraft they have a triplicated architecture. But what they've done is they've actually taken, you know, three different versions of the processor and three different versions of the software running on each of -- you know, so they had nine different processors.

So that they have all of the versions of the software running on all of the versions of the processor, and then they have a voting architecture that is used to try to detect disagreements that show up between those different processors. So there are a lot of -- a litany of ways that that's done.

MEMBER KRESS: And I presume there's ways to locate the fault within --

DR. JOHNSON: Yes.

MEMBER KRESS: -- a software --

DR. JOHNSON: Ways to locate faults within systems.

MEMBER KRESS: Within systems.

DR. JOHNSON: A lot of times, you know, the fault location techniques don't focus on whether it's a hardware problem or a software problem. They're simply focusing -- in fact, they typically focus at what we call the information level, because for you to be able to detect things it has to somehow corrupt a piece of information in your system.

And you have to be able to either detect that because it differs from what was expected or it differs from a replica of it.

MEMBER APOSTOLAKIS: Now, the word "fault" appears -- you know, fault occurs, and then you have four boxes. Are we talking about one fault? Because you told us earlier that "fault occurs" means some environmental condition. It's not necessarily a fault. I mean, it's -- something happened.

DR. JOHNSON: Well, it could be --

MEMBER APOSTOLAKIS: And then there may be a fault --

DR. JOHNSON: It's not just environmental.

MEMBER APOSTOLAKIS: There may be a fault somewhere inside the system --

DR. JOHNSON: Absolutely.

MEMBER APOSTOLAKIS: -- which is uncovered by that.

DR. JOHNSON: Absolutely.

MEMBER APOSTOLAKIS: So we are not talking about a single fault.

DR. JOHNSON: Not necessarily. I'll show you another diagram that --

MEMBER ROSEN: Now, that has -- that fact has enormous consequences for PRA modeling of digital systems.

CHAIRMAN SIEBER: Yes. But this chart here is showing a process -- detection, location, isolation, and recovery. And so you're moving to the right through that process for a single fault. Now you may have multiple faults. A single fault may generate other faults.

MEMBER APOSTOLAKIS: Or it may not even be a fault.

CHAIRMAN SIEBER: Well, an example of that is --

MEMBER APOSTOLAKIS: The input may be some abnormal condition. That's not a fault.

CHAIRMAN SIEBER: The tuning of a process loop where the equations -- the algorithms that you use don't take into account the harmonics of the system. Okay? And so now you've got some valve that's gone from full open to full closed, back and forth, that's caused basically by the mechanical features of the sensor and the actuator, and the computer is just doing what it was told to do. That's all, however, because it can trip the plant.

MEMBER BONACA: Actually, the -- it was mentioned before on the system failures. The Delta 3 failure was of that nature. But essentially a control system was overreacted, and it ran the actuator too hard and too long. The actuator lost the hydraulic oil, and eventually failed. So it was a very complex and drawn-out situation. You know, if the flight had been shorter, it would have not -- you would have gotten away with it, but so --

CHAIRMAN SIEBER: But the interesting thing about those kinds of faults is that if you don't run the physical plant, you can't test for them.

MEMBER BONACA: Right.

CHAIRMAN SIEBER: You know, I think there are some -- some mathematical ways to do it. On the other hand, it's not particularly reliable because you don't know the dynamic parameters with enough certainty to be able to model everything. But that is clearly kind of a fault that can occur that can trip the plant or cause some unsafe actuation.

MEMBER KRESS: This discussion brings to mind -- in your fault injection technique, are you injecting one fault at a time? Or can you inject multiple faults when --

DR. JOHNSON: There are two things to keep in mind. When we do the fault injection experiments, in the case of simulation-based experiments, we have, you know, the real ones and zeroes that correspond to the software. We're executing that real software. So if there are any defects or faults in the software, we're exercising those faults.

And then when we do the injection, we can actually inject, you know, one or hundreds. I mean, we can inject as many as we want to inject, simultaneously or at different times, or for different durations. So we can emulate permanence and transience and things of that sort.

MEMBER APOSTOLAKIS: First of all, I'm dying to go Slide 9, because there is a probability --

(Laughter.)

-- have failure rate. But before I die, what are the C's? C_DC_L is the fractions or --

DR. JOHNSON: These are probabilities of -- for example, probability of detection.

MEMBER APOSTOLAKIS: How do you know that?

DR. JOHNSON: Well, this is what we focus on estimating.

MEMBER APOSTOLAKIS: Estimating.

DR. JOHNSON: This is what we use the fault injection techniques to try to estimate.

MEMBER APOSTOLAKIS: Now, in your fault coverage, as it was discussed a few minutes ago, the real issue -- and you acknowledge that -- is really how do you come with an envelope of faults so that you build up your confidence that what you are doing is very meaningful.

I was wondering -- could one use some of the analytical tools that, for example, are used in PRAs, like fault trees or some other method, to go back to your previous slide and -- and develop a model for the hardware-software environment, and maybe that can help you to be a little smarter when you select the faults. Has that been tried?

DR. JOHNSON: That's exactly right. In

fact, we have -- one of our publications -- and, in fact, the patent that we have is based on something we call malicious faultless generation.

Essentially, what we do is we take the hardware-software execution and we -- we generate essentially a fault tree, but it's a time varying fault tree in the sense that at every step of the execution you have a fault tree of all of the things that could go wrong at that point in the execution that could lead to an incorrect result from that execution.

And those things -- not only that, but those things would lead to an unsafe output being delivered to the system, and we call those malicious faults. And we actually, you know, developed some algorithms that allow you to do that automatically, find some of those, and then you can inject those.

Now, one of the difficulties we have is that since those are not, you know, randomly-occurring things necessarily, when you start trying to integrate those into a probabilistic model, you have some things that are truly randomly selected. You have some things that are not.

And one of the issues that we've been, you know, struggling with quite honestly in looking at ways to handle is how do I -- how do I integrate some things that are, you know, non-probabilistic with things that are probabilistic and come up with a probabilistic answer.

MEMBER APOSTOLAKIS: But it seems to me that this is a general method that can be used in connection with any method that has been developed to identify failure paths.

DR. JOHNSON: I think so.

MEMBER APOSTOLAKIS: Right?

DR. JOHNSON: My hope would be that it could.

MEMBER APOSTOLAKIS: Yes. Another critical question here is: when you find a fault, don't you fix it, so that $C_D C_L$, $C_I C_R$, are not constant.

DR. JOHNSON: That's correct.

MEMBER APOSTOLAKIS: Which is a crucial observation.

DR. JOHNSON: That's correct. I mean, and if you think about it from an experimentation standpoint, let's say you do find -- let's say that -- you know, let's say that fault location is done by software. And let's say that you run this set of experiments, and you find a bug in your fault location software. I mean, obviously you're going to fix it.

MEMBER APOSTOLAKIS: Unless you are Intel.

(Laughter.)

DR. JOHNSON: But you're going to fix it. So now the question is, you know -- I mean, that obviously changes your system, and the question is: now what do you do? And, you know, those are --

MEMBER APOSTOLAKIS: I mean, any probabilistic calculation now is really up in the air in my mind.

DR. JOHNSON: Now, what we've done in the systems that we've done when we've found bugs, you know, we've fixed them, and then we've -- we've, you know, generated another set of experiments. We've repeated the process.

The danger you have there is that that can be an iteration that can go on forever, and you've got to know when to stop.

CHAIRMAN SIEBER: It sounds like lifetime employment.

MR. ARNDT: Well, the saving grace there is the decision point in most cases is: is it this good or better? Or do you fix it? You're driving the system to be --

MEMBER APOSTOLAKIS: Well, but the -- I mean, I'm sure Dr. Johnson will come to it, but I've seen other models in the past where, for example, they say, "Okay. It's not as detailed, but maybe if you consider the fault boxes as one box." And so there is a probability P of something going wrong.

Then, I ran it many times. I find five errors. I fix them. Well, P is not constant anymore. You can't use the binomial distribution. You can't use any of that. And yet people go ahead and use it. And, I mean, I would rather gain confidence by doing this with a reasonable envelope than try to force a probabilistic model that probably doesn't mean much.

But now let's go to Slide 9; there is a failure rate.

CHAIRMAN SIEBER: No. Let me take a little timeout here for a second. You're about a third of the way through your presentation, if I count the slides.

DR. JOHNSON: And I'm about three-fourths of the way through my time probably, and halfway through --

CHAIRMAN SIEBER: Your time will run out in 16 minutes. On the other hand, it seems to me at this point when we move to Slide 9 you're getting into a lot of detail where a break would not be appropriate. Is that true?

DR. JOHNSON: Well, we could certainly --

CHAIRMAN SIEBER: This would be the slide to break for lunch on, would it not? Or would it?

DR. JOHNSON: It would certainly be an appropriate place to break, and George is left hanging.

MEMBER APOSTOLAKIS: Yes.

CHAIRMAN SIEBER: I did that on purpose.

MEMBER APOSTOLAKIS: But let me raise another thing, though. I mean, we -- obviously, Barry will need more than 16 minutes to cover all of this.

CHAIRMAN SIEBER: Yes.

MEMBER APOSTOLAKIS: Then we have a Maryland presentation scheduled for 1:15. Carol, can you stay later?

MS. SMIDTS: Yes, that's fine.

MEMBER APOSTOLAKIS: The members will stay, too?

MEMBER KRESS: We'll always stay.

CHAIRMAN SIEBER: They told me you wanted to work overtime today.

MEMBER APOSTOLAKIS: I'm going to have to get out by 4:30 or so, 5:00 at the latest.

CHAIRMAN SIEBER: We will finish by 5:00.

MR. ARNDT: Yes, I think we're running

something like a half hour late now.

MEMBER APOSTOLAKIS: But we're going to lose Dr. Guarro at 3:30, because it's his birthday today. And we are so cruel we don't -- can you move your birthday?

(Laughter.)

DR. GUARRO: In 20 years --

(Laughter.)

CHAIRMAN SIEBER: Yes, Steve.

MR. ARNDT: I think we're running about a half hour late, maybe slightly more.

MEMBER APOSTOLAKIS: more.

MR. ARNDT: If we continue to try not to get any later, I think we're going to be okay. But --

MEMBER ROSEN: As you can see, there's some interest in the material.

MR. ARNDT: Yes, absolutely.

MEMBER APOSTOLAKIS: But I would like, though -- I mean, I would hate the idea that Sergio doesn't hear anything from Maryland. So somehow --

CHAIRMAN SIEBER: Yes, let's shoot for finishing by 3:30. But I still think now is a good time to break for lunch. Forty-five minutes I think would be sufficient. That will get us started again at quarter to 1:00.

(Whereupon, at 11:59 a.m., the proceedings in the foregoing matter went off the record for a lunch break.)

CHAIRMAN SIEBER: Since people have to travel and so forth, I suggest we continue on without a break until we're done.

MR. BONACA: Do you want to skip most of figure number 9?

CHAIRMAN SIEBER: Yes, let's move to 9.

MR. JOHNSON: I guess one question to start here is how much time would you target for me to try to get through the slides? Do you have a stop point that you want to shoot for?

CHAIRMAN SIEBER: I think that our crowd is going to dissipate around 3:30.

MR. APOSTOLAKIS: We have one more presentation.

CHAIRMAN SIEBER: Yes, we have one more presentation. And it is scheduled for --

MR. ARNDT: An hour and 15 minutes.

CHAIRMAN SIEBER: Yes.

MR. ARNDT: I have a short presentation after that.

CHAIRMAN SIEBER: Which is what, 20 minutes?

MR. ARNDT: It could be a little less.

CHAIRMAN SIEBER: Okay. If you could finish by no later than 2:00.

MR. JOHNSON: 1:45.

CHAIRMAN SIEBER: 1:45, that's perfect.

MR. JOHNSON: I'll try to move through as quickly as possible. The point of this slide is, it's a very, very simple model. You know, most of the models that we deal with are considerably more complicated than that, but the point that I wanted to make with this model is that what we look at in these systems is something we call safety, and we have a definition for that that I'll show you on the next slide. But more specifically, we look at something that's known as the steady-state safety.

We know that in these systems that there's some rate at which these problems occur. We just don't know how to estimate that rate, nor do we know how to partition it necessarily between hardware and software, so we are looking at systems that are collections of things, and we know there's a rate. We don't know what it is. We're assuming that there's a coverage that's associated with that, based on the definitions that we've had on the previous chart. And what we're focusing on is how do we estimate probability of being in one of these two states, either the operational state or the failed safe state.

Now more specifically, because we don't know these rates, and don't know how to estimate those, at least I don't, we are looking at something that's known as the steady-state safety. In fact, we have several papers that I'd be happy to make available to you that show solutions of these various types of Markov chains for time varying failure rates, time varying coverage factors and so forth. And one of the things that's intriguing about them is that if you look at this property steady-state safety, you know, as you start out, if you assume you start in the operational state and you transition over time, the probability of being in one of those two states, either operational or fail safe is something that will decay to a constant value, and a constant value is what we call the steady-state safety.

And you can show for at least all the architectures that we've looked at to-date, that if you look at that steady-state safety, it'll approach a value that's dependent exclusively on the coverage and not on the rate of occurrence of these events, but on your ability to handle them when these events occur. And so most of our work has focused on this examination of so-called steady-state safety, and we'll see this show up a little bit later on.

MR. APOSTOLAKIS: So that's independent of LAMDA?

MR. JOHNSON: That's independent of LAMDA.

MR. APOSTOLAKIS: But you have assumed that LAMDA is constant.

MR. JOHNSON: We've assumed not that it's constant. In fact, we have in one of our publications the time varying failure rate LAMDA, and you do have to make some assumptions about how it varies with time. I mean, you cannot have arbitrary variation with time, but you can have time variation. We've actually looked at several fairly well-known failure rate functions that do have some time varying properties to them, and we can still show a bounding box essentially for the steady-state safety where you can put a bound on it, depending upon exclusively this coverage factor.

MR. APOSTOLAKIS: And C is assumed to be constant.

MR. JOHNSON: C in this case is assumed to be constant. We've also looked at the time varying coverage, and again if you make certain assumptions about coverage as it varies over time, you can show some of the same properties with this bounding. In fact, I'd love to have some other folks look at some of these papers and make comments on them.

So again, steady-state safety is what we focus on, since we've shown for certain architectures, it depends on coverage. We focus on how we estimate that coverage, and that's really the focus of the research that we do.

Now the challenge, obviously - we've already talked about this a lot, but I did want to make a couple of comments on this, is that software and hardware are not independent entities. The software executes on a hardware platform. One of the interesting things that we found in some of the systems that we've looked at is that a lot of your software oftentimes is developed to handle problems that occur in your hardware, your fault detection, fault management.

In fact, we did a survey and looked at both aviation and railroad. We didn't look at nuclear in that case, but we found that in the systems that we looked at, that 80 percent of the software typically was for fault management purposes, or the management of events that would occur in the life of the system. And one of the interesting things in some of the systems we've done, we've actually found that there are -- we've actually found bugs in software, and the software regimes were there for the detection and management of faults that occurred in hardware, but they were not -- those defects in the software were not made visible until you actually exercised it in the way that it was designed to be exercised in the fields.

So this introduces some interesting things where you could have now a fault in the software that would never be a problem if you didn't get a fault in the hardware, or you can have a fault to the hardware that wouldn't be a problem if you didn't have a fault in the software. But the existence of both there can be difficult. So this lack of independence, again, is a focal point.

Now a little bit about our methodology. There's a fairly complicated or fairly extensive document that describes all of this, but I guess what I tried to do is to put some of this into a fairly straightforward diagram. I mean again, the types of things we focus on, steady-state safety is the primary one that we look at. We do analytical models. We've actually looked at Markov, and Petri Nets, and fault trees, and dynamic fault trees, and all the ones that we've been able to find out there in terms of analytical models. And we've actually done models based on all of them to-date. And from those analytical models, again for steady-state safety, coverage is the key parameter. And then the question, obviously, is how do I estimate that coverage?

And we've looked across a spectrum of possibilities there focusing -- really most of our work focuses on the three blocks to the right here, where we create physical prototypes, and we do experimentation on those prototypes. We create simulation models and do experimentation with the simulation models. And by experimentation, I mean fault injection.

We have statistical models that allow us to look at the data that we derived from these experiments, and we use that information to estimate predominantly this coverage parameter. That's the primary parameter that we're focused on.

MR. GUARRO: I'm just wondering, for those situations that we were discussing or envisioning before in which essentially you may have a design problem in the software, how would the coverage question be posed, because it seems to me that if you have a design issue, you normally would inspect coverage with that. In other words, if you're encountering a situation that has not been clearly anticipated, unless you have some catch-all type of provisions that are devised in a way that -- hopes to catch unforeseen things. I'm just curious, is there something that addresses that in your approach?

MR. JOHNSON: Typically, in most of the systems that we look at, the way that they attempt to address those types of design defects in the design is through diversity, so you have a system that is controlling the turbine, and then you have a system that's overseeing that control, and they are diverse systems. And you're attempting to overcome some of that. A design flaw that could occur in the system that's doing the control by a diverse implementation that hopefully doesn't have those.

MR. GUARRO: Okay. I understand, but in terms of your attempt to estimate the degree of coverage that you have, how do you address those type of -- I guess you're trying to develop a C condition or probability of coverage.

MR. JOHNSON: You're trying to encapsulate within that coverage both the design and the randomly occurring faults that can occur. And you're attempting to do that by really two mechanisms. One is the type - in fact, we talked about it earlier - the envelope of faults that you inject, you're attempting to address certain types of design faults in that injection process. Now the difficulty there is how do I model those, how do I represent those? And we don't have a solution to that yet, although it's something we are working on.

The other thing that you're doing though is in this experimentation that you're doing, you're exercising the real system, so the design faults that are in the software and the design faults that are in the hardware, you're exercising those as part of the experimentation. And your objective is to try to uncover some of these design faults, and use the information on the number of those that you're uncovering as a means of estimating the probability that there may be design faults remaining in that system.

And in the systems we've addressed so far, we have, indeed, uncovered design faults by the experimentation. I'll show you some of those systems in a moment.

MR. APOSTOLAKIS: But there is a fundamental assumption behind it though, that these faults that remain and the faults that you are encountering are exchangeable in some way, and that may not be the case. Maybe there is a single design, there is a design that affects something under certain conditions or under different accident conditions you have something else. This is a pretty strong assumption.

MR. JOHNSON: What do you mean by "exchangeable"?

MR. APOSTOLAKIS: They come from the same population, and that the order of appearance does not -- in other words, there are four design faults there. If I capture two of them, then I can say something about the remaining two because they are essentially from the same process.

MR. JOHNSON: Yes.

MR. APOSTOLAKIS: But that may not be the case.

MR. JOHNSON: That's an excellent point. That is an assumption that's being made.

MR. APOSTOLAKIS: A pretty strong assumption.

MR. JOHNSON: It is. And it's one we would like to figure out ways to be able to overcome that.

MR. APOSTOLAKIS: I know it's an extremely difficult problem. And come to think of it, I mean even for the hardware, we don't have any acceptable models for design.

MR. JOHNSON: No. And in fact, if you can think about how hardware is designed today, I mean hardware design and software design have become almost indistinguishable, because the way you design hardware is you write a software program that describes the functionalities that you want, and then you use another incredibly complicated piece of software that automatically synthesizes an implementation of that hardware. So your hardware and software design processes have become almost indistinguishable, and the types of problems you create in software, you have the potential to also create similar types of problems in hardware.

MR. APOSTOLAKIS: That's right.

MR. JOHNSON: It's an interesting paradigm that has come about as a result of the way that we design systems.

MR. APOSTOLAKIS: But I think this is really something that bothers me about - not your work only, but in general - attempts to quantify the probability. This assumption that all the design faults are exchangeable is an extremely strong assumption, and that implies the assumption that you can have a failure rate or a constant probability of failure, so it's not quite valid. Now what to do, I don't know. I don't think anybody knows, but we have to acknowledge that we're making some pretty strong assumptions.

MR. JOHNSON: I agree with you, and I think it is -- I agree with two things that you said, two that pop out.

MR. APOSTOLAKIS: Yes.

MR. JOHNSON: One is that it is a strong assumption, and obviously, an assumption that's being made not only in my work, but around the world.

MR. APOSTOLAKIS: No, I know.

MR. JOHNSON: And the second thing is that it's -- I'm not aware of anybody that has a solution to that.

MR. APOSTOLAKIS: I am not either.

MR. JOHNSON: And, in fact, I think someone would be quite famous once they find such a solution. It's a hard problem.

MR. APOSTOLAKIS: It is a very hard problem.

MR. JOHNSON: A very hard problem.

The other point - this is a little bit of a cluttered slide. I apologize for that, but I guess the point that I want to make with this is a couple of things. One is that systems are made up of lots of different things, basic circuit elements, basic logic. You know, typically when you look at systems, you look at them from an architecture all the way down to detail circuit level. And there are several things that are important here. I mean, the whole concept of defense in depth normally is that you're going to put protection mechanisms in at different levels of the system, different layers of the system so that if one thing misses a problem, another thing has the ability to catch that problem. And some of those are not part of the electronic system. They may be mechanical things that you've done, or containment buildings that you've put in place and other types of things, but there are layers of protection.

Those layers are subject to design faults because you may have made mistakes in the creation of those protection mechanisms in each of these layers, as well as the function, basic functions. You also have things that just happen, and the random events that occur in the failure of hardware and so forth, and the point of this diagram -- I mean, really when you're talking about designing systems, your objective is to eliminate a problem, either a design fault or a randomly occurring fault that can somehow make it all the way through your protection mechanisms and create a failure.

The other point of this chart is that from an analysis standpoint, those are the ones you'd love to be able to find because again, that would be a tremendous insight into the system.

Now the other point on the right-hand side of this is that the modeling that we've done - and I'll start at the lower levels because this is important, because when you're looking at a lot of systems nowadays, you don't have access to this information. Part of it's because the way we design hardware nowadays, and even software. You know, you can specify a digital filter in Netlab and synthesize C codes that will implement that, so you may not know a lot about some of these lower levels. So one of the aspects of our research is that what we're trying to do is we are trying to look at these lower levels, so hopefully people in industry don't have to. So we're trying to use the results of the analysis that we do at low levels of hardware and software to try to characterize the elements at higher levels of abstraction, so that that characterization can then be used by people, and not necessarily require them to go down to these lower levels of detail.

The way we've approached the analysis of these systems is that we've done all of this. We've done the higher levels, and we've done the lower levels, and we've used information from these lower levels - in the hardware world they call this back annotation, where you try to extract information from those lower levels and better characterize your model at the higher levels. And what that's leading to, hopefully, in the work that we're doing is a couple of things.

Actually, let me hold that thought for a second because it's not the next chart. It's the one after that, that points that out. But first, in this –– particularly these couple of blocks, what we've done is to develop modeling schemes that allow this integrated model. So, for example, we look not only at the actual code levels of modeling these systems, but we look at higher levels, as well, so we can create data flow representations of the algorithms that you're going to run on your computer, and have a way of interfacing that high level description of your system to a high level description of a hardware element. So that, for example, imagine that as you're simulating this, you have a function that needs to use hardware resources in order to execute, and you don't necessarily know the details of the hardware, but you can characterize potentially the timing and the other fetch and executive processes that you have to go to function, so that there are two points with this.

One is that, we do have some models that we've created that are actual bits representing the code running on gate level models of the processors, but we also look at higher levels where you have much more abstract representations of your software and algorithms running on much more abstract representations of your hardware. So the concept of integrated hardware/software modeling is intended to span all of those levels of that diagram I had on the previous chart. So that's part of what the integrated modeling is all about.

And then the second thing, the point that I was making earlier is trying to characterize these things. You know, we talked about hardware synthesis a second ago. I mean, for example, suppose that a synthesis program creates an application specific to integrated circuit, and as part of that application specific integrated circuit, it synthesized a little processor and a little memory, and put bits or state information in that memory that now caused that little processor to, in effect, be embedded in that ASCIC. Now is that ASCIC now a hardware element, or is it now a hardware/software element? And my argument that it effectively has become a hardware/software element because it has programmable features associated with it, even though it's one-time programmable. The synthesis routines created a program effectively that defines the function of that piece of hardware.

So what we're attempting to do, and again we don't have time to go into all of the details, but the attempt here is what I call interface modeling. The idea is to be able to model using state machines, the interface between this device and the outside world, and be able to characterize the things that that can do at the interface when something goes wrong internally, independent of whether it's hardware only or whether it's a mixture of hardware and software.

This we can very quickly -- several times we've talked about peer review. And in my world, the key publications that we go to are the IEEE Transactions on Reliability or the IEEE Transactions in general from someone in an electrical engineering department, transactions on computers and transactions on reliability. I've listed just some of the key publications over the last few years that look at some of the modeling techniques, some of the parameter estimation techniques, statistical models and others, so that's just for reference more than anything else.

MR. APOSTOLAKIS: The "IEEE Transactions on Software Engineering" --

MR. JOHNSON: "IEEE Transactions on Software Engineering", I personally have not published there, but that's one of the major publications, as well. And there are conferences and so forth, as well.

MR. APOSTOLAKIS: Have you ever had a reviewer say you're working with rates that's unacceptable, reject. Since there is so much controversy out there, do you occasionally get the guy who just rejects it outright because you dare talk about the failure rate?

MR. JOHNSON: Occasionally, you will get a reviewer that just rejects it outright and doesn't tell you why, but I've not had that particular --

MR. ROSEN: It's Tuesday, and I feel like rejecting it.

CHAIRMAN SIEBER: That's the way I would read it.

MR. JOHNSON: Those are just, again, some examples of some of the work that we've done. The other thing - again, Steve had indicated that we should put in some things about who's looking at this, and the peer review and other things as more of the examples.

As I mentioned earlier, the theoretical foundation for the work that we do was really created when I was at Harris, and I applied the very preliminary ideas to a flight control system that I was working on as part of the team at Harris. Also, one of the products of our modeling and our research is modeling and simulation tools. ADEPT which stands for Advanced Design Environment Prototype Tool, was the first place that we implemented these ideas in a tool set. This was actually funded by NSF and DARPA and NASA, and so that's -- ADEPT was integrated into the metrographics tool set. That was the basis that we used for the ADEPT tool set.

ROBUST was the second in our line of tools that was funded by the U.S. Air Force, and we've been -- again, I'm only talking about my particular piece of the center today, but my work has been funded since 1984 continuously by all of the organizations that are listed there. And my students that have come out, and the papers that have been peer reviewed, the patent and so forth, and I've been fortunate enough to be named an IEEE Fellow for the contributions that I've made in this area, so other people have looked at it. And I think that was really the point of this.

Applications - I've mentioned several of these, but I wanted to give you a few more specifics. The Los Angeles Metro Green Line was a transit application that we've talked about. We developed a model that had the actual software, the real ones and zeros executing on a model of the hardware. We created results for more than 10 billion experiments, using some techniques that we developed for not just running experiments, but helping you avoid running experiments that were not going to teach you anything. And also running experiments that were meaningful, so there were some 10 billion experiments that were created.

We actually uncovered three software design faults in the system. This was a system, this was a software system that had been in the field for almost ten years at 150 different installations, and it was software that was developed using all the right processes, and all the things that were -- and this was an ISO certified house and everything else, and we found three bugs in that software. And that software was updated and revised and so forth as a result of that.

The California Public Utility Commission hired an outside consulting firm to review all of the documents that we created as part of the analysis and to sign-off on those.

Copenhagen was very similar, a more complicated system. We did both simulation modeling, as well as physical experimentation, including some gate level things of a modern 32-bit processor. We actually uncovered one software design fault in that particular case, and all of this was actually approved by TUV in Germany.

We're currently doing Calvert Cliffs, New York City, CSX, a mag lev system in Pittsburgh, and Illinois Department of Transportation system as well, so those are currently ongoing activities.

MR. WHITE: Excuse me, Barry. I would assume that all these systems are very reliable normally, so I think an interesting point to be made here is that you've been able to work with systems where the reliability is already pretty significant, as we would hope would be the case in a nuclear situation.

MR. JOHNSON: Absolutely.

MR. WHITE: But one of the issues is how

do you do an analysis for a very reliable -- a system design to be very reliable, and it would seem to me that you're on track, but if you had any numbers for reliability for any of these systems, it would be interesting. I don't know if you'd be able to divulge any of that or not.

MR. JOHNSON: These are -- I can't -- the only thing I can tell you there is that the requirements, what they were shooting for is almost identical to what the aviation industry and others have been promoting over the years; that they're looking at 10 to the minus - anywhere from 10 to the minus 7, to 10 to the minus 9 probability of unsafe event occurring over a life, over a period of time. So those are the types of numbers that they're targeting.

I think in all of these cases, again we produce numbers, but I think in all of these cases it really was looked upon as the final number coming out of the analysis was not as important as the analysis itself, and what was demonstrated or learned as a result of doing that analysis.

MR. APOSTOLAKIS: Which is what people say about risk assessment.

MR. ROSEN: That's what he said all along.

MR. JOHNSON: Absolutely.

MR. ROSEN: But 10 to the minus 7 or 9 are so low that one has to wonder what do you think about uncertainty for that?

MR. JOHNSON: Well, actually, I think that's a huge issue. It's a research issue that needs to addressed, is an uncertainty assessment - because you've got uncertainty in the models, you've got inaccuracies and uncertainty in the models and parameters, and the estimation and so forth.

We've actually done -- some of the statistical models are based on some uncertainty principles that are used so that you can at least get an understanding of how much confidence you might have in some of the estimates you're getting out of the model.

MR. ROSEN: But if you're saying those kind of very low numbers, you need to be saying something like we think it's between 10 to the minus 7 and minus 9, and probably at least an order of magnitude one way or the other, whatever. But you fixed a number in there. The real number is probably within an order of magnitude either way, but you have to say some -- give some decision maker some feel for how sure you are of the result.

MR. JOHNSON: I mean you really have to really understand that issue. I mean, I've been amazed at some results that I've seen published where the accuracy of the computers that they were running these models on wasn't as accurate as the results that they were presenting. You really have to understand those issues, and it is an important problem.

MR. APOSTOLAKIS: Was the process for developing all of these systems controlled, or was it as controlled as the nuclear?

MR. JOHNSON: Very, very heavily. Very heavily controlled. I mean, they have a very, very rigid process for developing requirements and reviewing those requirements, and developing specifications, and the whole process of -- for both the hardware and the software, very rigid.

MR. APOSTOLAKIS: And these systems have been tested before you did your analysis?

MR. JOHNSON: Yes.

MR. APOSTOLAKIS: And they still haven't found design faults.

MR. JOHNSON: Yes. And one of the reasons, and most of these cases, the design faults were the scenario that I had illustrated earlier, where it was never a problem until you had a fault occur somewhere else in the system. And the combination of the two became now visible. And the reality is that in the field they just had never encountered those situations, nor did they encounter them in the testing process.

MR. APOSTOLAKIS: Somewhere else in the system was hardware?

MR. JOHNSON: Yes, in this case it was. These were -- in fact, in three of the four that I mentioned here, they were bugs in the software that were only revealed when a certain type of fault in the hardware occurred. Now these software routines were software routines designed to manage the occurrence of faults that could occur both in hardware and software.

MR. ROSEN: Doesn't that say that you can test it until you're blue in the face, but that as the system ages and the hardware ages and some of the stuff begins to -- you begin to see some premature failures of something on the cards, that that failure of something on the cards then creates a circumstance in which you'll see a software fault.

MR. JOHNSON: Certainly. This data is pointing exactly to that.

MR. ROSEN: But that has operation notifications.

MR. JOHNSON: It does.

MR. ROSEN: One of them is that maybe a strategy to avoid that is to trade-out, as my colleague Dr. Kress says, trade-out cards on a planned cycle as the system matures. Now that has some of its own problems because you can introduce premature failures in the new cards, but at least you're renewing the system rather than just letting all the cards age in time.

MR. JOHNSON: I think there are some - and I haven't looked at it any, but I think there are some operational issues that can be addressed perhaps more effectively as a result of some of the things we're learning from the research that's being done. I agree with that.

What I wanted to do in just a few remaining slides is just show you a couple of quick things about the Calvert Cliffs system. I won't go into a lot of detail, but again, since we are concerned with nuclear applications here, I just wanted to make sure that you knew we are in the process of working on this one. And actually, have pretty much finished it.

One of the things that we're going to be preparing as a report for this year is what I'm calling a Lessons Learned Report, where we're going to essentially talk about some of the things that we learned as a result of applying this to the digital feed on our control system. But this is the system, and again I'm not an expert on nuclear, but essentially it's controlling the water level and the flow of water in and out of the tank, steam generation.

MR. ROSEN: Just for interest, it's called a steam generator.

MR. JOHNSON: Oh, okay. It's an important part of the process, right?

MR. ROSEN: Oh, yes, it's pretty important.

MR. APOSTOLAKIS: A minor detail.

MR. JOHNSON: Yes. The control system was completely replicated in our lab at UVA, and you'll see a photograph of this. We have two controllers - not a very good photo - PID controllers. We have an experiment control station, which is where we were essentially simulating - I hate to use the word "simulating" in this case, because you're not really simulating the plant, but we're emulating it. Essentially what we were doing is applying from this control station a set of test sequences that were --

there were 23 of them that they used to test the system in their own lab at Baltimore Gas and Electric before they would actually put anything into the field. So it was a sequence of inputs and expected outputs that were being driven on this system, and then our experimentation - the physical part of the experimentation all occurred in this system, so it's a complete replica of what's in the plant, except that obviously we don't have a power plant. We're emulating that.

We did develop - this happens to be a dynamic fault tree. And again, I won't go into the specifics of it, but this is a dynamic fault tree of the digital feedwater control system, where we represented several things. The key feature of the dynamic fault tree is the ability to represent reconfiguration and coverage-related matters, so we have –– and this is described in our documents. I'd be happy to make those documents available to you.

If you look at just the controller portion of that, there's an equivalent Markov model that you could derive. Again, this is documented as well, but it's very simple because you have two units. You can have both of them working. You can one working, you can have a repair occur during that operation period, and you can have an unsafe or safe failure of that system.

DR. KRESS: Would you call MU a fraction of that that went from two to one and made it back before an unsafe condition --

MR. JOHNSON: It's a repair rate but it's exactly that concept, where you have both units working. One of them fails and shut downs, and some time later you'll have it either automatically or physically repaired and brought back on-line, so you're going to have both of them up and running.

DR. KRESS: Before an unsafe condition --

MR. JOHNSON: Before an unsafe conditions could occur.

MR. APOSTOLAKIS: But again, there's an assumption here --

MR. JOHNSON: It's a fraction.

DR. KRESS: I think it's a fraction.

MR. APOSTOLAKIS: All these things rest on the assumption that they are constant, C is constant --

DR. KRESS: Yes.

MR. APOSTOLAKIS: Otherwise, a Markov model would be --

MR. JOHNSON: Again, you can consider time variations and so forth, but this particular case obviously is based on constants.

MR. APOSTOLAKIS: So this is a Markov model for what?

MR. JOHNSON: This is a Markov model for the two controllers, just the master and the backup controller. It's actually a subset of the fault tree that I showed you earlier.

MR. APOSTOLAKIS: Yes.

DR. KRESS: These are independent redundant controllers.

MR. JOHNSON: Yes. Now if you look at the solutions, I'm going to show both the MTTUF -- let me make a couple of points about this. You know, we don't -- again, because of the -- you know, we don't have a good way of estimating that LAMDA. We don't focus on this piece of it, and this is really -- it's meantime to first unsafe failure, so it's the occurrence of the first unsafe failure.

The steady-state safety, though, depends on a couple of the coverage parameters. One that's gained by having these two units compare amongst themselves, and the other that's gained by diagnostics that are running on each of the two units.

MR. APOSTOLAKIS: But there is a reason

for that, and the reason is that if you go back to your transition diagram, you're assuming the same LAMDA for both controllers, and they're failing independently.

MR. JOHNSON: Yes.

MR. APOSTOLAKIS: There is absolutely no coupling because you see from --

MR. JOHNSON: They are assumed to be independent.

MR. APOSTOLAKIS: Yes.

MR. JOHNSON: That's exactly right.

MR. APOSTOLAKIS: That's why LAMDA cancels that. Now is that a reasonable thing to do? I don't know.

DR. KRESS: Of course, you don't have a MU in there either, so I presume you're --

MR. JOHNSON: Yes. The mean time to the first unsafe failure --

DR. KRESS: The first unsafe --

MR. APOSTOLAKIS: Because MU takes you from one to 200.

MR. JOHNSON: That's right.

DR. KRESS: No, no.

MR. JOHNSON: Now we have -- that's one of the things -- I mean, the paper that I've referenced here actually looks at more complicated architectures and some of the more complicated models, so there are -- you do have the ability to look at some of these more complicated issues, but it's not addressed in this particular model.

DR. KRESS: You have to -- to get this mean time you start out with determining what the failure frequency is.

MR. JOHNSON: That's right.

DR. KRESS: And they've gone over that.

MR. JOHNSON: That's right. And that's why we don't -- again, I show this for reference only. We have not focused on this metric because of the difficulties with estimating this LAMDA.

DR. KRESS: But you do get the frequency.

MR. JOHNSON: You'd like to have that, certainly.

DR. KRESS: You know, this is equivalent to the frequency.

MR. JOHNSON: Yes.

MR. APOSTOLAKIS: Now I don't know, Sergio and Jim, you have seen more data than I have. Is the assumption that the controllers fail independently a reasonable one, and that they both have the same LAMDA, or could be there some common cause failure?

MR. WHITE: Well, there certainly could be some common cause failure, and I'm sure Barry is aware of that. And I guess it would be interesting to see, and maybe you've addressed it in your paper, what happens if you assume certain degrees of dependence. So how does that affect your results?

MR. JOHNSON: We have looked at that.

DR. KRESS: You could actually have two controllers out at different LAMDA.

MR. JOHNSON: Absolutely.

DR. KRESS: That's a complicated --

MR. JOHNSON: The results get more complicated.

MR. APOSTOLAKIS: But then the result would not be independent of LAMDA, which is your objective.

MR. JOHNSON: This would still be independent of LAMDA.

MR. APOSTOLAKIS: Would it be?

MR. JOHNSON: Yes. Even if the LAMDAs were different you can -- in fact, we've done a generic model where you've got differing LAMDAs --

MR. APOSTOLAKIS: So what this is, this is the probability of being in the unsafe state?

MR. JOHNSON: This is the steady-state

solution of the safety expression. Safety is the probability of being in either the operational or the fail-safe state.

MR. APOSTOLAKIS: So this is safety.

MR. JOHNSON: This is the steady-state solution to that particular probability. This is --

MR. APOSTOLAKIS: Either in one, or two, or what? I mean, if we go to the previous diagram, which state is that?

MR. JOHNSON: It's one or two or the FS state.

MR. APOSTOLAKIS: One of the three.

MR. JOHNSON: One of the three. That's right. And, in fact, if you look -- if you ignore repair of a system, you know, if you look as time goes towards infinity, what you're going to find is that the probability of being in one of those three states is going to approach a constant value. It'll approach a limit, and that limit is what the steady-state safety is.

MR. ROSEN: And that's dependent mostly on the coverage. Is that right?

MR. JOHNSON: Yes.

MR. APOSTOLAKIS: Only on the coverage.

MR. JOHNSON: Only on the coverage.

MR. ROSEN: Only on the coverage.

MR. WHITE: Okay. Now I have to ask a question I was hoping I wouldn't. The limit as time goes to infinity, that always catches my interest, and that kind of analysis is very useful if you're just trying to get passed a problem and you can do that simplification. I would presume that the times we're talking about are really long compared to other things you're worried about, or not - if they're really short compared to -- so my question is, how limiting an assumption is that if you're trying to estimate a failure rate of a digital system?

MR. JOHNSON: Well, if you're looking at the -- again, if you think about the safety expression that we would find early on, and the safety function, what you can show is that this limit is a worst case. I mean, it is because your safety function will decay from -- you know, if you think of safety as the probability of being either operational or fail-safe, it will actually decay from starting point of one to this bound.

MR. WHITE: Okay. I don't want to take up the Subcommittee's time, and I'm sure you've thought about this, but there are some cases where that may not be the limiting case. A limiting case – and I may be wrong. It may be that if the system continues to operate the way you would expect it to, that is worse than if it were to fail immediately, if it's going to fail later on. And I just don't know. It hurts my head to think about it, so I didn't know if you'd gone through that kind of reasoning in any --

MR. JOHNSON: We have. We've thought about that. We don't have any results to show on that, but we actually have looked at that quite a bit. There are other things that you can do. When you do start to look at some of the time variations of the parameters and other types of things, you can still find bounds, but they're not -- the bound is not necessarily identical to the steady-state solution. There are some things that show up like that, depending upon repair issues and other types of things, but you can still find a bound that's dependent upon the coverage factors.

DR. KRESS: Isn't the PRA likely to use the upper expression?

MR. JOHNSON: I'm sorry?

DR. KRESS: For use in a PRA, wouldn't you just stick with the upper expression?

MR. JOHNSON: I guess my --

MR. APOSTOLAKIS: Why isn't one minus this the probability of interest to us?

MR. ARNDT: That's the most important parameter.

MR. APOSTOLAKIS: But the failure be independent of the rate of challenges? I have difficulty understanding that.

MR. JOHNSON: Isn't the probability of failure upon demand? Really, I mean, one minus this would be the probability failure on demand. But again, it's a bound. It's not the actual probability.

MR. APOSTOLAKIS: But this is a continuous controlling -- we're controlling the feedwater level continuously, so I should have a failure rate at any time, shouldn't I?

MR. JOHNSON: If it's a digital system, you may not. Well, what's continuous and what's --

MR. APOSTOLAKIS: What does it demand then?

MR. WHITE: In the digital world, that also hurts my head.

MR. APOSTOLAKIS: No, but the model itself has a rate of challenges LAMDA, which then disappears.

MR. JOHNSON: That's right. Right. And again, it disappears because you're looking at the probability and its limit. If you look at the probabilities at any instant of time, they depend on LAMDA. But if you look at the limit as time gets large, then it will decay. And essentially, your variables that depend upon LAMDA are eliminated from the expressions, because again it's a bound. I mean, if you think -- the simplest example is where you have -- safety might be your coverage plus a term that is exponentially dependent upon time. And as time gets large, the exponential term disappears, and that term goes to zero. And what you're finding in the safety function is that there are some terms that disappear, and there are some terms that remain.

In the architectures we've looked at today, the terms that remain are dependent upon the coverage and nothing else.

MR. APOSTOLAKIS: Let's go back to 22 for a second. The rate at which I visit the FU state from two or from one depends on LAMDA. Right? LAMDA is there. The rate at which I go into FU depends on LAMDA. Then the steady-state probability of being at FU is independent of LAMDA. That's interesting. I guess you have carried out the calculations.

MR. JOHNSON: I'll be happy to show them to you.

MR. APOSTOLAKIS: Yes, I'd like to see

that.

MR. ROSEN: Remind me again, what's cease of S?

MR. JOHNSON: Cease of S is a coverage factor but it is specifically the coverage that's provided by diagnostics that are running on a single processor unit. We use the term Simplex, so you have two ways of detecting problems in the system. One is you have comparisons that you're making, and then you have others that are diagnostics that are being run in real time to try to assess the health of the system. That's cease of S.

MR. APOSTOLAKIS: There's another --

MR. ROSEN: So if you go to your resulting S of SS expression, explain to me what S of SS is that's equal to coverage.

MR. JOHNSON: No. S of SS is the probability of --

MR. ROSEN: Well, no. Let's go across the equation. It's equal to the coverage times one minus the Simplex.

MR. JOHNSON: Right.

MR. ROSEN: Times the Simplex squared.

MR. JOHNSON: What this is really showing you is that you have a couple of ways of handling problems. Okay? If you detect something by your comparison mechanisms, and you do not detect it by your Simplex or diagnostic mechanisms, then you'll still fail in a safe manner.

MR. ROSEN: Okay. That's the term that represents --

MR. JOHNSON: If you detect it in both processors using their detection mechanisms, you'll also fail in the safe manner. The case where you will not fail in the safe manner is where you have a problem that is undetected by the unit that's bad, and it's undetected by the comparison mechanisms. So what this is showing you, and again, it's a simple case, but you've got two contributors to the probability of being safe. You detect the problem with your comparisons, and you don't detect it with your diagnostics, or you detect it with both units detecting it via diagnostics.

MR. ROSEN: In your diagnostics.

MR. JOHNSON: And those are the conditions that lead to a safe failure.

MR. ROSEN: I'm sure that will be very helpful after I think about it.

MR. APOSTOLAKIS: Let me give you an interpretation of this. Let's go back to the diagram.

The diagram is critical here. The states FS and FU are what are called in Markov analysis absorbing states.

MR. JOHNSON: They are indeed.

MR. APOSTOLAKIS: Once you enter, you cannot get out. If you enter one, you can always get out through -- right?

MR. ROSEN: Right.

MR. APOSTOLAKIS: So the probability is one that if you wait long enough, you will end up in one of the absorbing states. Right?

MR. ROSEN: Right.

MR. APOSTOLAKIS: Because you can never get out. The probability is one. I think what the expression that Barry showed us is, is it splits the probability of one between FS and FU, and it says this fraction of time you will be in FS.

MR. JOHNSON: That's right.

MR. APOSTOLAKIS: And then one minus that is the fraction of time you will be in FU.

MR. ROSEN: Yes.

MR. JOHNSON: That's right.

MR. APOSTOLAKIS: But it is not what you call a safe thing. I am safe as long as I am in one or two, not in FS. FS is a spurious failure.

MR. JOHNSON: No, FS is a safe failure in the sense that --

MR. APOSTOLAKIS: Yes, it's a safe failure, but it's a failure.

MR. JOHNSON: For example, in the case of --

MR. APOSTOLAKIS: What I want is, I want to be in one and two.

MR. JOHNSON: Oh, absolutely. Absolutely.

MR. APOSTOLAKIS: And that probability you don't have.

MR. JOHNSON: No.

MR. ROSEN: But that's an operational view.

MR. APOSTOLAKIS: But that's what I want.

MR. ROSEN: No, no. I think the safety view is what we -- you could think about it in both spaces. Think about it in safety space. All we really care about is that this thing be safe. Then you don't care about LAMDA, because you're satisfied if you're in one, two, or FS. Even if you fail, the system is shutdown, the main feedwater pumps trip, the reactor goes into shutdown and you're safe.

DR. KRESS: You can learn something about one and two by the frequencies of these failures.

MR. APOSTOLAKIS: But then I think it's the fraction of time that of the given failures, you will be in FS.

MR. JOHNSON: Oh, yes. That's exactly right.

MR. APOSTOLAKIS: So it's a conditional thing.

MR. JOHNSON: It is.

MR. APOSTOLAKIS: It's conditional on knowing LAMDA.

MR. JOHNSON: Coverage by definition is conditional.

MR. APOSTOLAKIS: Yes, it's conditional.

MR. JOHNSON: It is conditional.

MR. APOSTOLAKIS: So you cannot take this and put it directly in a PRA, because it's conditional on the coverage.

MR. JOHNSON: It is, indeed, conditional.

MR. APOSTOLAKIS: Even the coverage. I know that eventually I will be either in FS or in FU, and what that is telling us is the fraction of times you will be in FS is the expression I'm giving you.

DR. KRESS: Yes. And then it's going to go on further --

MR. APOSTOLAKIS: And it makes sense.

DR. KRESS: It's going on further to tell us how you can use fault injection to get these coverages.

MR. ROSEN: But he hasn't explained yet how to do it in PRA space.

MR. APOSTOLAKIS: Because he's not a PRA man.

(Simultaneous speech.)

CHAIRMAN SIEBER: Why don't we continue on.

MR. JOHNSON: Okay. The fault injections that we've applied to this digital feedwater control system, we've actually done two different approaches. And I'll show you the software-based approach in a second, a simulation-based approach. We've done both software and simulation, software being where, you know, again as the system is executing we're able to insert corruptions into the system in the physical prototype. And then the simulation-based is obviously a simulation.

We actually have a scheme that we've developed that uses interrupts in the operating system to do this injection during the execution, so you can think of as the system is running along, you have a brief interrupt that then is your saboteur is the term that's used in the literature quite often, where you then can go in and do various types of corruptions based on the models that you've got, and then allow the system to continue from that point.

DR. KRESS: Have you got that automated so you don't have to sit there and type something in?

MR. JOHNSON: Yes, we do. We have -- we didn't for a long time, and so my automation was undergraduate students that --

(Laughter.)

MR. JOHNSON: We have automated much of that now. And then the simulation-based part of it is we've actually migrated this into a COTS tool that's called Simics to allow us to do some simulations. I'll show you a little bit of that in a moment. In fact, this is the Simics.

The main point that I wanted to show with this is that where we talked about the entire system and our goal is to be able to model the plant or to get -- we're not going to model the plant, but to get a model of the plant, and have a model of the plant that can be interacting with our model of the system that's controlling the plant. So that, for example, what we've done so far is for the GE turbine controller, we've got a very simple model of the plant. We have the -- it happens to be an Intel 386 processor with operating system and application code running on that simulation model with interfaces then between this part of the simulation and the digital feedwater control system in the physical prototype. This is all done in physical hardware and software implemented in the lab. This is done by a very simple input/output relationship.

In the GE gas turbine controller, this is a completely simulated hardware/software simulation, and then this is a simulated plant model. The objective is to be able to have these fault injection experiments done in an environment where you're actually interacting with a model of the plant that that's going to be interacting with in the real world. And again, this framework allows you to do that, and actually we've done it. And that's built on a COTS tool call settings.

Now I guess the last slide, just one comment. I actually debated on whether to put this in here, but I put it in here for the following reason - because I do think that the ideal place to do a lot of the things that we've done and are doing is in the design process. I mean, if you go look at some of the things that are done, there are a lot of people that design the system and then step back and say how do I make it reliable, or how do I make it safe. And I think that's not the right way to do it. I think the design for safety ought to be an integral part of the design process, and the assessments that we do should really be an integral part of the design process. So that from the very beginning of the system, the simulation environment, which is what I'm calling the virtual prototype - actually, this is taken from the program we did with DARVA where some of these techniques were developed. The program is called Rapid Prototyping of Application-Specific Signal Processors, but the intent was to have a virtual prototype that as you go from start to finish in the hardware/software design process, including integrating and testing, that would all be done in a simulation environment prior to building anything. And all of the fault simulations, simulation-based and so forth that is done in what we developed can be a part of that process.

The point of this chart is not that we're there, but the point of the chart is that this is where we would like to go from the standpoint of some of the product of the research so that you've got the ability to integrate some of these things into the design process, and you can evolve this assessment as you're evolving the design.

MR. ROSEN: It may interest you to know, and I'm sure you do, that this Committee and me personally, are very supportive and insistent even on the use of PRA in the design process.

MR. JOHNSON: Yes.

MR. ROSEN: Not as an afterthought to evaluate how good did the design come out, but as a first principal thing. First you set down the system definitions and functions, and then you do the PRA first, the first PRA. Then you do a little more detail, a little detailed design, then you rev up your PRA until -- and use your PRA to say, you know, instead of having three of those things there, we really need a more full tolerant kind of thing. Here we need a separate system, more diverse, and I can change these split fractions and get a better answer here. And basically going out like this using your PRA tools until you get to the final design. What you're suggesting now for the software aspects of this is exactly the same thing.

MR. JOHNSON: Exactly.

MR. ROSEN: And I applaud that.

MR. JOHNSON: Exactly. I believe that

from the bottom of my heart, that's the right way to do it.

DR. KRESS: Now in his iterative process, the ideal is to get the risk down to a level that you accept, and not only to get the risk down to sort of minimize the uncertainties, and to spread the risk out in design in defense-in-depth over a variety of things.

MR. JOHNSON: Yes.

DR. KRESS: What would be your equivalent to these objectives for the software and hardware?

MR. JOHNSON: You know, I think to -- I mean, to some extent I think it's very similar, because I think there are -- you know, as you begin to create a system from a functional standpoint, there are going to be functions even in your software that are going to be more critical than others, because they are going to be -- you know, a good example of that is in the case -- in the GE system that we were working on they use a voting technique. So they go out and they sample a bunch of inputs, and then they come together in each of the units, then uses a software that does a vote across these multiple inputs that they've collected. And if you think about it, you can envision that that voting process may very well be more critical than some of the other processes that are out there. And if you could understand that from the beginning, that might determine where you put that routine in terms of mapping it to a specific processor. It might determine the level of scrutiny that you apply to that routine in terms of the test and evaluation, and other things that you might do. So I think it's very similar. It's just at a different level.

DR. KRESS: The objective would be to end up with the hardware/software, the end that meets the functional requirements at a high reliability level.

MR. JOHNSON: Yes.

DR. KRESS: Something like that.

MR. JOHNSON: I guess I should point out, this is -- I mentioned the DARPA project, which is where some of these concepts were initiated, but they also were further evolved with a project that I did with Boeing, so the objective -- Boeing's objective was exactly what I was describing in terms of the development of aircraft.

I am finally at the end. I appreciate your patience, and I appreciate the dialogue and the interaction. As I expected, I learned a lot. I hope you got some information that will be of value to you.

This repeats what we started with, working on the safety assessment process, integrated hardware/software. We've done it multiple times, not trying to imply that it's in any way finalized, but we've at least had some real experiences with it. And we're continuing to work both in terms of the models, as well as the tools that we're evolving. So again, thank you. Appreciate your time.

CHAIRMAN SIEBER: Thank you. And I guess, Steve, we're ready now --

MR. APOSTOLAKIS: The chairman insists on no breaks.

CHAIRMAN SIEBER: Yes.

MR. APOSTOLAKIS: We're doing so well.

CHAIRMAN SIEBER: Pardon?

MR. APOSTOLAKIS: We're doing so well.

CHAIRMAN SIEBER: I think we should not take a break.

MR. APOSTOLAKIS: Okay.

CHAIRMAN SIEBER: Because people have to leave, and I want them to get as much of the presentations as they can.

DR. KRESS: You're the chairman.

CHAIRMAN SIEBER: If you have an emergency arising though you may attend to it.

DR. KRESS: You have to raise your hand.

MR. ROSEN: The criteria will be you should not make medical history here.

CHAIRMAN SIEBER: Or any kind of history.

MR. ARNDT: Okay. As I mentioned earlier, another one of our programs is with the University of Maryland. The principal investigator of that project is Professor Carol Smidts. She'll make some self-introduction, as well.

MS. SMIDTS: My name is Carol Smidts. I'm an Associate Professor at the University of Maryland in the Center for Reliability Engineering, the Department of Mechanical Engineering. I graduated from the University of Brussels with a Ph.D. in Engineering Physics. My research interests are in probabalistic risk assessment and software reliability modeling.

The work I will present this afternoon is essentially geared towards using software engineering to predict software quality or reliability. So what my presentation wants to introduce a method that we have devolved to bring software engineering measures to actually estimates of reliability, and we have piloted this method on small applications which we'll talk about in the presentation.

The results of the method at this point are promising, and the method itself is in that paralleling the review process for software, which is the current process which is used by the NRC staff, so we believe that then it should be straightforward to implement.

Work is currently ongoing. We're performing work on the actual nuclear application or we're planning to do that, to actually validate the method on the larger scale application of high reliability, and specific to the nuclear field. So this is to reiterate.

When we started the project, the project was geared essentially towards review, and helping the review process to provide somewhat of a systematic framework. Basically, the software developer who comes to the NRC staff and is trying to get the license approved for their system has to go through a process of software development which is characterized in the branch technical position 14, and this is geared at developing plans, developing things such as software maintenance plans, some development plan, and also products, requirements, design, code, test results, and things of that nature.

Now the reviewer at NRC then has to look at this information, and from there infer whether or not the application should be accepted, so there is no quantitative measurement going on in that process really. Plus, at this point, the measurements that the licensee wishes to provide can do anything, as we discussed this morning.

So project history - we actually inherited this project from Lawrence Livermore National Laboratory. We started in 1996, and what Lawrence Livermore did is actually identify the first set of measures, software engineering measures that they believed were relevant to reliability. We then performed an expert opinion study to try to rank these measurements, and we performed a small scale validation study in 2001. And we're currently enlarged in the large scale validation study.

So Steve trapped me into doing this slide, and I'm still wondering why I did it. So here what I'm showing is what I understand to be the contributions that we believe we can at this point assess if we were to use the reliability estimates we're producing. So if you look at this event sequence diagram representation, what we're trying to show is what we believe are all the contributors of software, or contributors related to software, so at the very top of the diagram, what we do have --

MR. APOSTOLAKIS: You've got a pointer there. Can you use it?

MS. SMIDTS: Sure. So what we do have here is whether or not the support platform functions correctly. So this is actually the work that Barry concentrates on, which is to look at support platform degradation. And this is cases where the support platform actually functions correctly. And then what we do have is that the software gets inputs from sensors or humans and things like that, and this input needs to be characterized. We call it operational profiles sometimes. It's actually really the definition of the input. The software executes, and there's a delay of execution of the software depending on the input which we, at this point, do not characterize. No environment measurements actually look at that.

Then there is an assessment whether or not the behavior that is specified in the requirements is actually implemented, and that is what we're looking at - whether the behavior leads to a safe condition. And some of our measures actually look at that. And then let's assume that indeed the requirements are followed, well, it is also possible that the output still doesn't match what is the output required by the next component in the process, and we do not explicitly look at this. So I think this explains a little bit the context of what we do.

So here is our work. We look at software engineering measures, and try to create subsets of measures which can then be related to reliability. We may be able to create only one such subset, or we may be able to create several such subsets. If that's the case, it will be possible in the future to imagine that we could use another uncertainty framework to actually create better estimates for the reliability, so for these blocks here.

MR. ROSEN: George, you should be thrilled at this point.

MR. APOSTOLAKIS: I can't control myself.

MR. ROSEN: Besides that.

MR. ARNDT: Well, what's important to recognize is this is a method to help us quantify, to make the software review process more quantifiable. And it also has the opportunity to tell us something about the reliability.

MS. SMIDTS: Okay. So what is the idea behind this research? In other words, why would we want to look at software engineering measures, and how can we possibly relate them to software reliability?

Well, basically software reliability is determined by the characteristics of the product, so the software, and the characteristics of the operational environment which I call the input.

Now the product characteristics are actually determined by characteristics of the project, such as the type of application, and characteristics of the development environment, such as the skill level of the people involved in the development, or such as the schedule pressures and things like that.

Now these characteristics are actually measured by software engineering measures which apply to all of these elements. So in essence, software engineering measures are actually determining software reliability.

MR. APOSTOLAKIS: Wait. I don't understand the last bullet. How does that follow from the --

MS. SMIDTS: So what I said is that basically the reliability of the product is determined by the product itself, how it is, actually the functions in the product, the logic in the product and so forth and so on. And it's also determined by the development in the operational environment, how that product is executed. So the features of the product are influenced not immediately but indirectly by the project characteristics and the development characteristics.

MR. APOSTOLAKIS: Software engineering measures, what are these?

MS. SMIDTS: Those can be many types of things, such as, for instance, the logic complexity of a module, the number of lines of code in a module. It could be things like the number of requirements. The fact that requirements are traceable to the system, the software requirements are traceable to the system, and so forth and so on, there is a very large number of such measures.

MR. APOSTOLAKIS: But this is an assumption really. I mean, why is the number of lines determining the software reliability? It depends on how you wrote it.

MR. ARNDT: Right. There's a whole body of research associated with what things, how many errors or how many problems you have in software, and then you come out with the size of the code, the complexity of the code, the amount of times you change the code, all sorts of different kinds of issues. Many of those are measured for one reason or another.

We're trying to understand the quality of the code, or we're trying to improve the efficiency of the code, trying to do a quality process in the development of the software or whatever. There have been measures that are used for various reasons.

MR. APOSTOLAKIS: So what doesn't follow though is that they determine the reliability. They influence the reliability. They are indirect measures of the reliability, but they do not determine the reliability.

CHAIRMAN SIEBER: No, they don't. But you have to interpret some of that too. For example, it sort of follows to me anyway, the more lines of code, the more chances for mistakes.

MS. SMIDTS: Right.

MR. ROSEN: Well, it's not linear.

MR. APOSTOLAKIS: No, because I may have a million reviews.

CHAIRMAN SIEBER: Yes.

MS. SMIDTS: So that's why you would want then to combine that other measurement, which tells you how many measurements, I mean, how many --

MR. APOSTOLAKIS: What matters is the whole process.

MS. SMIDTS: Right. So you would want to

get all these measurements together, and together they actually should give you a pretty good --

MR. APOSTOLAKIS: That's the whole point of controlling the process.

MS. SMIDTS: Right. Right.

CHAIRMAN SIEBER: But if you set out to minimize the lines of code, you may be simplifying the algorithms to the point where you don't get very good answers, and so there are a lot of conflicting kinds of things here. I think the best thing to do is hire the smartest person you can to do the programming.

MS. SMIDTS: If they are too smart, then the code is really difficult to maintain.

CHAIRMAN SIEBER: Yes, I know about that. Smart, not tricky.

MS. SMIDTS: So the idea here was to postulate the existence of subsets of measures that could help us determine reliability. Now since we don't know what those subsets are, and we don't know what are the models that need the subsets to reliability, what we wanted to do was to be able to rank these subsets since NRC staff would actually get several measurements, and they would have to determine whether these sets of measures are actually going to product good estimates of reliability. Do they help us identify what the reliability of the product is, or is the set of measurements insufficient?

So basically, we can't rank these sets. So the idea is that we decided to start by ranking individual measures with respect to reliability, and hoping that by obtaining these rankings, we would be able to actually build sets that would lead us to top reliability prediction systems, that's how we call them.

Now to rank the measures, and I should --

MR. GUARRO: Carol, I'm sorry. You're saying you set out to rank individual measures?

MS. SMIDTS: Yes.

MR. GUARRO: Okay. Well, you probably guess what the next comment is; which is, it seems that there would be very strong combinatorial effects, in the sense combination actually, not combinatorial, combination effects so that depending on environment and situations, the relative ranking of measures could change from one situation to another. Have you thought about that?

MS. SMIDTS: Well, that's what I would have thought too, but as you will see, the results of the experts don't really seem to indicate that. I mean, we were trying to take experts which were coming from very diverse backgrounds to try to cover the most generic of cases. We took people from the telecommunications industry, from financial industry, from aerospace and nuclear, and so forth and so on, to try to cover all the possible aspects.

MR. GUARRO: Well, I'll give you an example in which you could force the number of lines of code down to the point where you generate logical errors. And I know of situations in which at least you could run into that type of problem when you're -- for reasons of efficiency. There's a certain type of processor that don't accept a lot of lines of code.

MS. SMIDTS: Okay.

MR. GUARRO: And again, I'm conditioned by this bit system experience where we still use because of space qualification issues. We use processors that are, in terms of technology, they're 25 years ago.

CHAIRMAN SIEBER: Z-80s.

MR. GUARRO: Right. Also, in certain languages are line of code intensive and certain languages are not.

MS. SMIDTS: Right. So if were to look at the lines of code that these people created, probably you would see that other measurements would show that, such as - I'm thinking about cyclimatic complexity, which show that the logic has become very complicated, so you would have --

MR. GUARRO: Yes, but I mean that's the point.

MS. SMIDTS: Right.

MR. GUARRO: You're in an environment which now, you know, if you're in a free environment where the lines of code are just left completed unconstrained, well, then probably yes, there you will find that the more lines of codes that people want to right, the more errors they may produce. But in an environment in which the lines of codes are constrained, now that factor, that particular metric is not free to influence the reliability as it in others. So something else flips ahead of it. So what I'm saying, if you're ranking one-by-one independently, you might not see these combined effects.

MS. SMIDTS: You wouldn't. You wouldn't see the combined effects. Yes, because we are ranking them one-by-one. Right. So another -- well, one of the things we could do in the future is to try actually to rank them by several factors, but we haven't done that.

MR. GUARRO: Well, I guess one way of

saying that is that the issue perhaps looked at different sets of environments in which the combinations of factors act in different ways.

MS. SMIDTS: Differently, yes.

MR. ARNDT: And one of the reasons we're trying to do pilot studies in various applications, particularly the nuclear domain dependent application for that very reason. We want to be able to valid the methodology in the kind of environment that we're interested in.

MS. SMIDTS: Okay. So this part of the slide that actually shows the criteria that we selected for the ranking of the measures, so one of the criteria is actually the relevance to reliability of the measurement. The other criteria try to assess the internal validity of the measure, so what we have here is, for instance, how costful the measurement is, what is the benefit of having this measurement to the organization. Has this measure been validated extensively by the scientific community, has there been much experience, industrial experience with this measure. Here, what is the level of credibility of the measure; in other words, does it actually assess the goal of the measure, and finally if this measure is repeatable or not. In other words, if performed repeatedly by different individuals, do we get the same measurement.

MR. APOSTOLAKIS: These overlap a lot, don't they? I mean, the degree of credibility depends on everything else, and the validation - V depends on R. The more people use it, the more validated it is, isn't it?

MS. SMIDTS: Right, but this the validation really by the scientific community. Here we would look at the industrial experience, so you may have a lot of validation from us, from the scientific community and nobody is ever using this measure.

Repeatability actually is really in the way the measure is being defined. Now we find that some measures like lines of code have a very low degree of repeatability, but they're used throughout industry largely.

MR. APOSTOLAKIS: Repeatability. What is repeatability again?

MS. SMIDTS: Repeatability is the fact that you can make a measurement - if you and I make the same measurement, do we get the same result. And a lot of the --

MR. APOSTOLAKIS: You said something about the number of lines of code.

MS. SMIDTS: Right.

MR. APOSTOLAKIS: What kind of measurement would that be, just the number of lines?

MS. SMIDTS: Right.

MR. APOSTOLAKIS: Do you disagree?

MS. SMIDTS: Yes.

MR. APOSTOLAKIS: Why would we disagree?

MS. SMIDTS: Because there are multiple definitions of the line of code to start with, so actually one of the problems that is in this field, my experience with this field now is that most of the measures are very readily defined. Repeatability is the real problem.

So this is the ranking process that we actually followed. So the first step in our work was to actually narrow down the set of measures that Lawrence Livermore had identified to 30 measures. Actually, in the set that Lawrence Livermore had prepared, there were things that actually were not measures, but techniques, things that were models and not measures either, so we narrowed that down, and then restricted to a set of Perti because we were gearing up for expert opinion elicitation. And we thought our experts would not be able to rank more than 30 measures. So step two was expert identification.

Now why did we perform this? We tried to look in the literature to see whether or not there would be some data that would allow us to rank the measurements on our own. And actually, it's very difficult to find any relevant data. The data may be there, but it's usually proprietary and companies will not share it, so the only way we thought we could actually approach this problem is by expert opinion elicitation.

So we identified a set of 10 experts, and I'll show you the name of the experts in the next slide. We defined the criteria specifically, and identified levels for the criteria. So for instance, for the experience criteria we had five levels, from a case where there was absolutely no experience with the measure, to cases where hundreds of companies had used the measure.

So in the next slide, we also design a questionnaire which we sent to the experts. The experts sent us their ranking back, and then we held a workshop to actually look at, and the experts actually explained their ranking. We also had interviews with the experts after the workshop to follow-up on some of their results.

We had left intentionally a door open in the sense that we allowed experts to identify what we call missing measures, so if they believed we had missed some important measurements, they actually indicated that.

The next step, we aggregated the opinions of those experts using utility fields, and this framework, the utility field framework has a number of parameters, such as the weights of each of the ranking criteria and so forth and so on. We performed a sensitivity analysis to see whether or not within bounds that we thought were acceptable or reasonable, whether the rankings would be actually modified. Then we analyzed the results.

So here are the experts. They were selected out of a set of 30 initial candidates, and we see the backgrounds of those experts. It's industry, academia mix, some have actual experience both in the industry and the academia. And here are the areas in which they actually -- the domains in which they work. All experts have knowledge with critical systems, with actually software reliability, and software measurement.

Here is the set of measures which we considered initially, and here are the results of the ranking. Actually, the experts provided rates for each of the measures in different phases of the development life-cycle. So a rate of zero essentially means that the measure is worthless, and a rate of one means that the measure is actually excellent.

As you see in the requirements phase, we have very little measures available because some of the measures become defined only in the later phases of the life-cycle, by testing all the measures available.

MR. APOSTOLAKIS: Are these -- you had what, 12 experts?

MS. SMIDTS: Ten.

MR. APOSTOLAKIS: Ten. So if I look at completeness requirements, you say .41 - don't tell me all 10 said .41. So how did you come up with .41? What is the dispersion?

MS. SMIDTS: The dispersion - do you remember, Ming, what is the dispersion, because I don't remember. How much dispersion --

MR. APOSTOLAKIS: You have to come to the microphone if you want to speak, and say who you are.

CHAIRMAN SIEBER: For the record.

MR. LI: I'm Ming Li. I'm the post doc researcher for Dr. Smidts. And this research actually is my Ph.D. topic. I have been working on it for over six years. And I found the number of repeats of that measure should be around four to six out of ten experts.

MS. SMIDTS: No, that's not the question. The question was, was there a lot of dispersion in the rating of the experts.

MR. LI: Oh, okay. Fine. Well, since we ranked using the latter scale, and if converting to zero to one, I would say 30 percent around. We didn't calculate that rigorously and have the statistics, but I will say it's around from -- let's say from letter D to letter B, something like that.

MR. APOSTOLAKIS: Let me understand. What exactly did you ask the expert to give you regarding completeness?

MR. LI: Well, for each measure - do you want to continue?

MS. SMIDTS: Yes. So for each measure, we asked them to tell us for each of the ranking criteria what was the level of that particular ranking criterion.

MR. APOSTOLAKIS: On a scale of what?

MS. SMIDTS: So the scales are -- they go from letter -- let's say there are five levels, so from letter A to E.

MR. APOSTOLAKIS: And then you converted it to a number.

MS. SMIDTS: Right. So what we did is actually, because we weren't sure that the conversion would not change the numbers which, of course, it does. It changes the number, but we wanted to verify whether or not the ratings remained correlated, so we performed a sensitivity analysis later on that.

MR. APOSTOLAKIS: So in terms of the letters then, for a particular one, what did they give you? Did you have a situation where somebody gave an A, somebody gave an E, another guy gave a C - it was all over the map?

MS. SMIDTS: We had cases like that.

MR. APOSTOLAKIS: So what does that tell you?

MS. SMIDTS: That there was, in that case, indetermination between the different cases. But most of the cases were not like that.

MR. APOSTOLAKIS: They were like what?

MS. SMIDTS: One letter grade probably.

MR. APOSTOLAKIS: From all ten of them?

MS. SMIDTS: From all ten of them. No, I mean maybe most of them were A, and then some gave B.

So I --

MR. APOSTOLAKIS: Remarkable.

MS. SMIDTS: I didn't study the -- I don't have in mind the actual variations of the experts. But it wasn't outrageous, like you would assume that -- I mean, it wasn't like you had a person gave E, and then everybody else -- and then one gave D, and two gave C, and then one gave A. It wasn't that bad.

MR. APOSTOLAKIS: And then you converted the letter scale to a numerical scale using what?

MS. SMIDTS: We actually did that using different curves. And what we did is we actually performed a sensitivity analysis on the different -- we varied the curves, the transformation.

MR. APOSTOLAKIS: Now when you use additive, you really have to make sure that the measures are independent. I mean, there is an implication of remarkable accuracy when you say .15. And it seems to me that some sort of statement of uncertainty would be required there.

MS. SMIDTS: Okay.

MR. GUARRO: Now when you're saying you used these curves, you adopted actually one curve for all the measures, or depending which measure you were dealing with, you used a different curve?

MS. SMIDTS: So we used one curve for all the measures, and then we looked at the rating at that point. Then we used another curve for all the measures, and what we were trying to do is to see whether or not the ratings were correlated for these different curves. And we did that for all these different curves. And then since we were looking at an aggregation framework which was additive, so we had different weights for the different criteria, and we varied the weights. So these are the sensitivity analysis schemes we looked at, those different weights.

MR. APOSTOLAKIS: So these are weights that you show there?

MS. SMIDTS: Right. Here.

MR. APOSTOLAKIS: Yes, and these are your weights.

MS. SMIDTS: This is the first -- these are my weights.

MR. APOSTOLAKIS: These are your weights, not the experts'.

MS. SMIDTS: They're not the experts', no.

MR. APOSTOLAKIS: Why didn't you ask the experts to also tell you relative importance --

MS. SMIDTS: I asked them to give me 30

measures, and for these 30 measures, I had how many criteria - seven criteria, and I had four phases of the life-cycle. So I didn't ask them the weight.

MR. APOSTOLAKIS: So what is it that we learn from this, Carol? Can you tell us what the conclusion from all this is?

MS. SMIDTS: Yes. I mean, my conclusion is the actual measurements which are important are relevant, and the others which are not.

MR. APOSTOLAKIS: So which are they?

MS. SMIDTS: Okay. So these are for the different phases of the life-cycle, the best indicators of reliability. Now some of them are obvious, of course, like failure rate. Now here we have code defect Ansically, which is surprising, but the experts considered that there is a lot of experience with this measure, and this measure actually measures the flaws in the code, the defects, so it is relevant to reliability.

MR. APOSTOLAKIS: It measures the defects?

MS. SMIDTS: Yes.

MR. APOSTOLAKIS: So you know how many there are?

MS. SMIDTS: That's what the measure gives you.

MR. APOSTOLAKIS: Then what do you do, you say --

MS. SMIDTS: That's the defects found.

MR. APOSTOLAKIS: Oh, these are defects found. And why is the --

MS. SMIDTS: Because actually they normalize it to the lines of code. But the measure itself, you have to understand, the measure is not on the number of defects per line of code. It's also the location of the defects found, the nature of the defect, the type of the defect, so it's all the information that was relevant to that defect, and identified in inspection.

MR. APOSTOLAKIS: So again, there is an implication as I was saying earlier to Barry, that these defects that you found are exchangeable with the ones you have not found. And that's a pretty strong assumption.

MS. SMIDTS: Well, found is defect found. If you have several inspectors that inspect at the same time, you have some models, and I haven't done that. I haven't pushed the research to that point yet, but if you have multiple inspectors inspecting, you can actually calculate through some statistical models to recapture models. You can calculate the number of defects remaining.

MR. APOSTOLAKIS: No. That's where I disagree.

MS. SMIDTS: You do disagree.

MR. APOSTOLAKIS: They all assume that these things -- I suggest we --

MS. SMIDTS: Homogeneous.

MR. APOSTOLAKIS: Yes. Not exchangeable.

MS. SMIDTS: Homogeneous, yes, in that sense.

MR. APOSTOLAKIS: Yes.

MS. SMIDTS: Yes. The only thing it gives you then is a first order estimate of what the number of defects remaining may be.

MR. APOSTOLAKIS: Sure. I mean, if I find lots of defects, I form an opinion about the process.

MS. SMIDTS: Right.

MR. APOSTOLAKIS: Right. I say, you know, these guys really didn't know what they were doing.

MS. SMIDTS: Right.

MR. APOSTOLAKIS: But presumably, you never do that in a strictly controlled process, I hope.

MS. SMIDTS: Well, that's what you believe.

MR. APOSTOLAKIS: Yes, well, I can only believe what I believe.

MR. WHITE: Excuse me, Carol. The coverage factor rate says third most important in the testing phase. Is that correct?

MS. SMIDTS: Right. In the sense that that measurement you can get at this point only during the testing, because what you do is you actually inject flaws and you measure whether or not it can recover from the flaw. So it becomes important on event, because it becomes available on the event.

MR. WHITE: Okay. What is fault number days?

MS. SMIDTS: Fault number days I think is actually the number of days that the fault remained in the application. Is that correct?

MR. WHITE: But how do you note that under the requirements? Say under requirements column, I see fault number days as rank number 5.

MS. SMIDTS: Right.

MR. WHITE: What does that mean?

MS. SMIDTS: So it would be, let's assume we start the development process, and then how much time did it take for us to detect the critical fault.

MR. ARNDT: After it was put into the

process.

MR. WHITE: Yes. Thank you.

MS. SMIDTS: Yes.

MR. GUARRO: Can you elaborate on the definition of the design defect in implementation versus design fault in the design phase?

MS. SMIDTS: Yes. So design defect density is actually in the same type of code defect density. It's the same type of measure. Design defect density is actually assessed with respect to the number of lines of design, so this would be with respect to - let's assume you have a design document, and you actually measure the number of lines of design. And you would actually then calculate --

MR. GUARRO: Okay. But essentially, defect for you is any variation from requirements?

MS. SMIDTS: Or it could be problems in the requirements.

MR. GUARRO: Okay.

MS. SMIDTS: Inconsistent, incorrect, ambiguous, anything.

MR. GUARRO: Okay. I understand that. So now in the design column, what is a fault, and how is it different from a defect?

MS. SMIDTS: Okay. So a defect, if I

remember correctly, is something that you identify by inspection.

MR. GUARRO: I understand the difference between defect and fault in the execution, so to speak. Fault is as executed, defect is just there. It may not be called upon. Am I interpreting it correct?

MS. SMIDTS: Yes.

MR. GUARRO: In other words, a defect is a latent fault, but is not an active fault.

MS. SMIDTS: Right.

MR. GUARRO: So I'm trying to understand what fault means in the design column, because in the design phase you will not know if something is being executed or not, so it's really a defect, isn't it?

MS. SMIDTS: Yes. However, you may have let's say a simulation at the design level which would allow you to infer that you have actually some kind of a failure, so if you're --

MR. GUARRO: Yes. But you're not using a reoperational profile --

MS. SMIDTS: No, we're not.

MR. GUARRO: -- so it's really speculation whether that is a defect or a fault. You see what I'm driving at? I don't --

MS. SMIDTS: However, you have -- I mean, you're right because you're not in the real environment. But in the sense that you know whether or not this would create a failure, it is your assessment, of course. Yes.

MR. GUARRO: Because you're using a postulated profile.

MS. SMIDTS: Right.

MR. GUARRO: And a postulated code itself, because you're still in the design.

MS. SMIDTS: Right.

MR. GUARRO: Okay.

MS. SMIDTS: So you have the -- yes.

MR. GUARRO: I'm just trying to understand the definition.

MS. SMIDTS: No problem. So here are the missing measures that were identified by the experts. Actually, the missing measures identified were the first four ones. And actually when we started, we were not considering OO projects or OO software, object-oriented software, because at that point there was little experience with object-oriented for safety-critical systems, so the experts recommended that we add a category of measurements which would capture object-oriented programming. So this is actually what we did here, we added those measurements for that.

The first one that they recommended was the coverage -- I mean, one of the first ones was the coverage factor for fault architectures. Then that of a full function point for real time systems. They believe that full function point was more relevant to real time systems then function point, which is another measure that we have.

CHAIRMAN SIEBER: Which number of children, fourth from the bottom?

MS. SMIDTS: The number of children I think is when you have a parent class, and the number of derived classes from that parent class. Okay. So this is the result of our sensitivity analysis, and what we -- I think we looked at 100 and something sensitivity analysis variations. And of those, you see that most of the variations are actually with a correlation coefficient, which is superior to .9, which is very encouraging in those results.

Okay. So now the hardware we are trying to actually validate our method, so we performed a validation on small scale studies. So this is the method which we applied. The first part, of course, is the selection of the application. We took an application which was a small control system, which was real time in that sense. It was pertinent to what the NRC types of applications are.

In the next step, we were looking at which measures to actually select for the validation. And we considered a limited number of measures due to the small scale of the validation study. We took measures which were highly ranked measures, which were ranked medium, and measures which were ranked low to see whether or not we could see actually whether the predictions were actually following that trend.

MR. ARNDT: This was also because we wanted to gain information on whether or not the licensee comes in with a ranking, be it high, medium, or low, or what amount of credibility they assigned to that.

MS. SMIDTS: Right. So in the third step, we performed the reliability assessment. So what happens is that we split our research team in two components. One component actually was performing measurements, and trying based on those measurements to predict reliability. And another part that the team considered to be a team that knew what the ideal behavior should be, so they actually had what we call the Oracle, the perfect behavior, or assumed perfect behavior.

Now in the fourth step, what we did is we actually tried to construct those reliability prediction systems. In other words, we tried to bridge what we knew the measures for reliability. In step five, we performed measurements and analysis. And in step six, our results of peer review.

So here are the small scale systems that we considered. The first -- so this is personal access control system to enter in a building. The first system was devolved by industry. It was devolved following the Capability Maturity Model, and that particular company at the time was rated at level 4, and they were asked to perform this development at level 4. This was actually -- we used the system in another study that was sponsored by NSA, so the code was devolved in C++, and the reliability of that application is .92 per demand, around .92 per demand. So it's not a very high reliability system. It's a low-medium reliability.

DR. KRESS: When it was unreliable, refuse access to somebody that should have been --

MS. SMIDTS: Let in.

DR. KRESS: Or let somebody in they shouldn't have.

MS. SMIDTS: Right. Right. So then since

we wanted to see whether or not we could use what we had devolved on a more reliable application, we actually asked West Virginia University to develop another version of this same system, again in C++. And here, the reliability is much higher with this system. It's .999 per demand. This work was sponsored by NASA. So the measures which we are using in the validation were these, two high-ranked, two medium, two low.

SPEAKER: Before you leave the slide, should I draw any kind of conclusion from the fact that you had a CMM level 4 that was reliability of .92 per demand? And if so, what would that conclusion be?

MS. SMIDTS: The conclusion would be that you cannot trust that you cannot trust a CMM level to tell you what is the reliability of the application. And now if you want to probe further, I can tell you that this is because there are no real measurements which are required by CMM. It's a process without actual final measurement.

MR. ARNDT: There have been several studies related to the CMM process and its ability to predict the quality in the software. And there's been a lot of controversy associated with it, obviously because it's an important issue, particularly in military software development. And many of those studies have shown various issues, in my opinion the most important of which are for classes of software. You get a good prediction, you narrow it down to single codes with reliability or the validity of that becomes more difficult as you might think. Also, as the code size shrinks, like sometimes in say the critical real time systems, the validity of CMM as a predictor of quality goes --

MS. SMIDTS: Yes.

MR. ARNDT: That's one of the many reasons that SEI, Software Engineering Institution, has looked at individual code, individual measures for individuals or small teams, as opposed to whole companies, which are more applicable to smaller codes.

MR. GUARRO: Carol, this may be a silly question, but why mean time to failure was included as something to test? I mean, it's essentially a parameter that defines reliability so, of course, it will be highly correlated with reliability.

MS. SMIDTS: Yes, you're right. So actually what we did is that in the second study, PACS-2, we took it off.

MR. GUARRO: Okay.

MR. APOSTOLAKIS: It's not the true mean

time to failure, is it?

MS. SMIDTS: It's not the true.

MR. APOSTOLAKIS: So somebody estimates it, so it makes sense.

MR. GUARRO: But it's essentially the only measure that you have out of your system that tells you what the reliability may be.

MS. SMIDTS: Yes. So I've been going back and forth because, I mean, for the first study what we did is that the team that was performing the measurement calculated the mean time to failure. And the team which had the Oracle, calculated the failure rate. Now they're not the same perception of the system, yes.

MR. GUARRO: Yes. Okay.

MS. SMIDTS: Okay. So this is the environment that we used to perform the reliability assessment, so using this Oracle. So what we do is we start actually from the requirements, and the team develops, analyzes the requirements and devolves a finite fake machine that represents the behavior of that system.

Then if you put that in some test generation tool, such as the Test Master Tool, well, you can automatically generate test cases, and those test cases are run automatically by a test execution tool which is not WNRunner, but WinRunner, so that allows us to run a very large number of tests automatically actually. And the failure/success is captured also automatically.

MR. APOSTOLAKIS: So this is your approach, these boxes?

MS. SMIDTS: This is -- so what we -- what I was saying is what we did is we split out team in two parts. One part was measuring and was trying to assess reliability, and the other part was supposedly the Oracle. And that team defined this, so this represents the Oracle and the testing using this perfect image of what the system should be. So then what we do is we try to compare the results.

MR. APOSTOLAKIS: Did you at any time actually look at the process that the NRC has blessed for the development of software? This is your approach. Right?

MS. SMIDTS: Right.

MR. APOSTOLAKIS: Did you look at that?

MS. SMIDTS: I looked at the process. I read the documents which are related to that. No, this is just a process to assess what the reliability, the true supposedly reliability of the application is.

MR. APOSTOLAKIS: But I thought what you were trying to do was to ultimately go - maybe I was wrong - go through the process that the NRC staff has established and say based on whatever I have learned, if you really follow this process you end up with a reliability of such and such.

MS. SMIDTS: Right.

MR. APOSTOLAKIS: Isn't that what you --

MR. ARNDT: Not quite. What the idea is, is to -- the process that the licensees need to follow is laid out. What's laid out is how we're going to review their process.

MR. APOSTOLAKIS: Yes, I agree with you.

MR. ARNDT: What we're trying to do is inform our review of their process by adding a quantity of measures.

MR. APOSTOLAKIS: Yes, but they will follow the process that you have in your SRP.

MR. ARNDT: Right.

MR. APOSTOLAKIS: So at some point you could take these insights, apply them to that process.

MR. ARNDT: Right. What we're trying to do is update the process, our review process so that we look at things that are the most important to final system reliability. And this is designed to find out if there are measures that are going to help us do that.

MR. APOSTOLAKIS: Now are there any measures -- I mean, from Professor Johnson's presentation, we learned that in some of the errors that he caught, there were hardware/software interactions. Are any of the 30 measures addressing that?

MS. SMIDTS: That was the coverage factor, actually. So that one actually --

MR. APOSTOLAKIS: The coverage factor is the same as his coverage factor, and that's the only one.

MS. SMIDTS: There are others -- let's say there are others that probably look at it indirectly, such as, if you look at requirement traceability, what requirements traceability does is look at whether the software requirements are traceable throughout the development of software. But also, if the software requirements are traceable upstream to the system.

MR. APOSTOLAKIS: Yes. I have a couple of examples in my mind of actual failures, and I'm wondering how this approach relates to that. There was a case that I read some time ago where the pilot in a fighter plane commanded the software to raise the landing gear while the plane was on the ground, and it went down. And then, of course, they realized that the software should have an interlock of some sort that said if you're on the ground, don't do that.

CHAIRMAN SIEBER: Or get a new pilot would be good too.

MR. APOSTOLAKIS: Now that is a requirements problem, is it not?

MS. SMIDTS: Yes.

MR. APOSTOLAKIS: Would your approach find anything like that?

MS. SMIDTS: Well, normally in the requirements they should actually define what are the range of correct inputs in different situations.

MR. APOSTOLAKIS: Right. But in this case there was an incorrect situation, I guess.

MS. SMIDTS: Right. So all --

MR. APOSTOLAKIS: So would you find that?

MS. SMIDTS: Well, in the case you're not in the range of correct input, you should have specified behavior for inputs that are not within that range. If such are not defined, there is a problem in the requirements.

MR. APOSTOLAKIS: And I know there is a problem.

MS. SMIDTS: Okay.

MR. APOSTOLAKIS: The question is whether your method would find that problem.

MS. SMIDTS: Right, because you would have --

MR. APOSTOLAKIS: How would you find it?

MS. SMIDTS: You would have normally a measurement that would tell you that the requirements are incomplete, because that range of parameters outside the correct range is not considered.

MR. APOSTOLAKIS: Which measure of the therapy would do that?

MS. SMIDTS: Well, requirements completeness, for instance.

MR. APOSTOLAKIS: Requirements completeness. Yes, it's easy to talk about requirements completeness but somebody has to actually evaluate it.

MS. SMIDTS: Right.

DR. KRESS: You have to have a complete set of requirements.

MR. APOSTOLAKIS: Right. And I think that's what part of the problem is, isn't it? That you need somebody with an imagination, in this case maybe it doesn't take much imagination but it does, that would say you need this. So I don't know whether any formal methods, or any of this, or even what Barry is doing, whether it would find something. I don't know.

MS. SMIDTS: So actually, to go back to one of the first slides I have is that actually you have to create those input conditions depending on the sequence in which you are, you need to assess what is the set of input conditions that that software is going to seek.

MR. APOSTOLAKIS: And I agree with you. And it seems to me this is the real issue we're facing in the nuclear industry. Right? The software may get some inputs that command you to do something that is inappropriate for that particular context. And that, it seems to me, is more a matter of technical knowledge on the part of the designer than anything else.

MR. ROSEN: You got hold of a very good point I think, George. Let's take some real operational circumstances, for example. Let's take a case, a plant I know where three trains of central cooling water should never all be out of service at once.

MR. APOSTOLAKIS: Exactly.

MR. ROSEN: And let's say this plant was digitally controlled, and one could give a command to the software to take out a train of central cooling water. And then one could go to the next train, to B Train. Let's say you did it to A, and then go to B and do the same, and they would accept the second command too. But when you went to the third train --

MR. APOSTOLAKIS: Should refuse.

MR. ROSEN: -- it would refuse to take out the train, so that's the third train, because the other two are out. You have to put one back before you can this one out. Now that should be a requirement in the requirement software.

MR. APOSTOLAKIS: And my question is whether the methods we've been discussing here from Virginia and Maryland, if the designer had made the mistake and allowed all three to be out, would any of these match this? Again, this is not -- don't take me wrong. I'm not actually criticizing you. I'm addressing what I think is the real issue in nuclear safety.

MR. ROSEN: I understand, but I don't think you can ascribe that to the software. I think the software --

MR. APOSTOLAKIS: It's a design, the

design of the software.

MR. ROSEN: The designer of the plant has to work with the designer of the software to say amongst all the thousands of other things he wants the software to do --

MR. APOSTOLAKIS: I want you to do that one.

MR. ROSEN: -- I want for the central cooling water never to be take out three trains at once.

MR. APOSTOLAKIS: That's right.

MR. ROSEN: And this will not allow an operator to do one, two, three, or the software to make a fault in which it automatically takes out all three. If it tries to do that, or even succeeds to do that, there's a fault error message. There's an error message pops up immediately, and the software takes another algorithm and puts one of the trains back in service, or something like that.

MS. SMIDTS: So that actually should be specified in the system requirements.

MR. APOSTOLAKIS: I agree. What should have been done is clear. Whether you catch it is the issue.

DR. KRESS: It's just like the PRA

completeness issue.

MS. SMIDTS: Right.

DR. KRESS: If you're incomplete, you're not ever going to find it until something happens and you say oh, I should have had that in my PRA too.

MS. SMIDTS: Right.

DR. KRESS: This is the same way. You will never find it with any of these messages, and you can't hope to.

CHAIRMAN SIEBER: No, you can't ask software to --

DR. KRESS: You can't ask it to do that.

MR. GUARRO: That is true, but the question is, for example, if there are ways of analyzing the interactions between hardware and software that help identifying situations in which key requirements have not been identified.

MS. SMIDTS: And the answer is that in any reliability assessment we do, be it based on measures or anything, one of the primary issue is to characterize the input space, because once you characterize the input space, you will be able to trigger conditions that may not be represented in your software model.

MR. GUARRO: Exactly. I think that's the

key point, whether there are ways to generate the sets of input conditions in a way that essentially probes the design of the integrated system. It's the same type of story of the example we were discussing before, the one that I'm familiar with because it comes from this basis and environment. That was not a software problem specifically, because if you hotwired their own parameter value into an analog controller, it would have caused exactly the same failure.

MR. ROSEN: Damage the valve are you talking about?

MR. GUARRO: Well, the overreactive launch vehicle control system that ran the system out of hydraulic fluid. And that's a particularly tricky one, but there are things of that nature that if you have some orderly way of verifying the requirements and looking at the spatial requirements, I think it can help you think in the right direction. I don't think that there is any particular silver bullet that automatically says okay, here are your missing key requirements. Unless you look at the hardware and software together, you're not even triggered to think in that direction, so I think that's the key thing too.

DR. KRESS: I was wondering if the University of Virginia processed moving from the left to the right, iterating with a simulated system would uncover something like that. You would ask -- at every point along the line you would ask your system if the plant has some sort of unacceptable failure, what conditions would make it lead to that. That might be one of the things you have to pick up.

MR. APOSTOLAKIS: Still though, if you went there with a mindset that when I command it to raise the landing gear it has to do it, without ever thinking that if I'm on the ground I shouldn't allow it, then you probably convince yourself, even with this approach that it's okay.

DR. KRESS: I think you --

MR. APOSTOLAKIS: It comes down to technical --

MR. ARNDT: The basis you're going to have to have, as we discussed earlier in the day, a detailed understanding of what you're trying to accomplish in the system.

MR. ROSEN: And knowledge of the system itself, whether it's an airplane that wants to crash itself on the ground, or a Delta rocket with a hydraulic control system, or in a central cooling water system that can't have all three systems out at once. And nuclear, aerospace and airplane requires an initial definition of the system requirements by the engineer, not of the software but of the engineer of the system.

MR. APOSTOLAKIS: Exactly.

MR. ROSEN: And then once those requirements are set down, then it becomes the job of the software engineers to accurately translate them. But absent having the system requirements from the engineers of the system, the software process is doomed to start with.

MR. APOSTOLAKIS: I'd like to come back to the various measures that you have evaluated. If we all agree that this is really a major, if not the major problem with software requirement specification, are we creating a false sense of security by looking at things like number of lines, density of faults. That's where the action is. Shouldn't we be focusing on this issue? Like, for example, I don't think you have a project on formal methods.

MR. ARNDT: We have a small project that's part of the --

MR. APOSTOLAKIS: Now these guys claim that they check for internal consistency. Now again, if internal consistency means I want to raise the landing gear and I always do it, then that doesn't help.

MS. SMIDTS: No, it doesn't.

MR. APOSTOLAKIS: It doesn't help me either.

MS. SMIDTS: No.

MR. APOSTOLAKIS: Why should I care about what you do, Carol?

MS. SMIDTS: Well, you should because I look at the combination of the input conditions. I force you to actually look at the input conditions, because you cannot create a reliability estimate if you don't define the input conditions.

MR. APOSTOLAKIS: Absolutely agree with that.

MS. SMIDTS: I cannot --

MR. APOSTOLAKIS: But which of your measures deals with that?

MS. SMIDTS: The measures themselves, the 30 that are there don't.

MR. APOSTOLAKIS: Are doomed.

MS. SMIDTS: So I have to add measures to my set to actually get that.

MR. APOSTOLAKIS: Well, now you're

talking. I would really love to see those measures.

MR. WHITE: But what about your measure of review inspections and walk-throughs? So my question is, and I think what we found on the panel was, if these reviews inspections and walk-throughs are done by the equivalent of plant engineers, that's good.

MS. SMIDTS: Yes.

MR. WHITE: But if it's done by a bunch of software engineers, then you're going to get into the same problem because you're going to miss these other -- so which of these did you mean in reviews inspections and walk-throughs?

MS. SMIDTS: Is that the one in the requirements phase?

MR. WHITE: That's one of the pre-selected 30 measures.

MS. SMIDTS: Okay. Those are done at different phases of the life-cycle, typically by different groups of individuals. So if you're early in the life-cycle requirements phase, you will have plant engineers in that group. You will have user representatives in that group.

MR. ROSEN: That's the key, that the user representatives get on board I think the day the contract is signed for the new system. The very first person after the project manager who is assigned, the project manager picks up the phone and calls the equivalent of a plant engineer and says put yourself on an airplane and be here at 8:00 Monday morning. We're starting the design of the new whatever. Airplane, space system - because it's his input that's crucial for almost everything you do from that point on.

MR. APOSTOLAKIS: I would suggest –– maybe you're already thinking about it, Steve, that you have somebody, a group or whatever, think about this issue of requirements. What is it that we can learn from the existing literature on faults that have been found, and what can be done about it? I agree with Sergio and Tom, that it's an issue of completeness and our brains cannot handle issue of completeness in a sense that we can prove that something is complete. But as Sergio says, there might be ways that can guide, that would enhance the probability that you will identify something in the process. It seems to me that's so important that it certain –– that doesn't mean you can do this at the expense of this or something else, but it's such an important thing that it seems to me by itself should be a task.

I've looked at a number of these things, and it's not really the fault of the software, it's the system. The guy who designed the whole thing that either didn't foresee something, or didn't know enough, or whatever.

MR. ROSEN: Had never flown an airplane like that or something like that.

MR. APOSTOLAKIS: Yes.

MR. ROSEN: But the minute you put someone on the team who has flown an airplane like that and his life depended on it, he will tell you his life's anecdotes in very brief time, and you'll make sure you don't make those mistakes at least again.

CHAIRMAN SIEBER: I don't want to interrupt but there are 13 slides in 10 minutes.

MR. APOSTOLAKIS: Ten minutes, 35 minutes.

MS. SMIDTS: Okay.

CHAIRMAN SIEBER: No, we're going to let Steve also talk.

MR. APOSTOLAKIS: Steve can talk after 3:30. I'm here.

CHAIRMAN SIEBER: I will encourage -- you may even want to pick out the best of your slides --

MR. APOSTOLAKIS: The best of the best.

MS. SMIDTS: Okay. So --

MR. APOSTOLAKIS: And the most legible.

MS. SMIDTS: Okay. So one of the things I wanted to say though is that I know you insist a lot on requirements, but do not forget that there are a lot of implementation errors also. So this is -- I have 10 minutes? Okay.

CHAIRMAN SIEBER: You're skipping the interesting part.

MS. SMIDTS: Okay.

MR. APOSTOLAKIS: When challenged, she responds.

MS. SMIDTS: So here are the results -- so these are -- so what I skipped is actually the building of the prediction system from the different measures. So what you can see is actually for PACS-1 on the left-hand side, you have the values which are obtained for the different measurements, so this is by the measurement team. And here you have the predicted probability of success by each of those measurements.

We're using in this box here - what you do have is the actual correct evaluated probability of success of the system. So here is just the relative error for the different predictions.

MR. ARNDT: Predicted relative error.

MS. SMIDTS: Right. Predict relative error? No, the actual relative error for each of these. So here is the original rankings of the experts, and here is actually the rankings that we obtained based on validation. So some of the results we get is that as you see the high-ranked measures produce the best estimates, medium-ranked measures product not as good estimate. And, of course, low-ranked measures produce actually relatively bad estimates.

So the same thing -- so this is another thing I wanted to show you, is that the method that we use for validation is actually reviewed by these four experts. These people were pretty familiar with our research earlier because they had participated in the expert opinion elicitation. They didn't flag any major significant problems with the --

MR. APOSTOLAKIS: Where is Michael Lyu now?

MS. SMIDTS: He's in Hong Kong, University of Hong Kong. Okay. So here the study carried out for the second application, so here is the reliability estimation, and here again is the rankings obtained based on expert opinion. And here again, the rankings based on the validation. Here the predictions from the different measures, I mean reliability prediction sets.

MR. APOSTOLAKIS: What is number one?

MS. SMIDTS: We took it out. We took the mean time to failure out from this study.

DR. KRESS: Now if you had, I say looked at the dispersion of the predictions, would that have changed your opinion?

MS. SMIDTS: From the experts you mean.

DR. KRESS: Yes. You might have had a bigger dispersion for some of these than others, and it might change your opinion of which ones --

MS. SMIDTS: Are actually --

DR. KRESS: Right.

MS. SMIDTS: Yes, that's a possibility. I'll consider that definitely. Okay. So here are some of the publications that relate directly to this work. The expert opinion study was actually published in Transactions and Software Engineering, and here is some other publication. What we use actually, the predictions to reduce the amount of testing. This is some other things that can be used for it, that can serve as some prior estimates. And we can reduce the amount of tests that needs to be performed on an application.

So our current research is to look at an actual system for the nuclear industry, and we have picked the STAR system, which is used at Oconee. And this is the Safety STAR System, reactor protection system.

We've also extended the number of measurements we're going to look at so this is now a total of 12 out of the 30 measures. And we're going to consider the different phases of the life-cycle requirements design coding and testing, and see what those different phases tell us about the reliability, and what we can extract from that.

MR. ROSEN: What did you say, it was done at Oconee?

MS. SMIDTS: The STAR system. It's a digital system used for the reactor protection system.

MR. ROSEN: Okay. It's a new digital system for Oconee.

MS. SMIDTS: Right. So we'll continue working on the improvements for those reliability prediction systems. And one, of course, of the major problems is getting defects, and what to do about them.

So as a summary, the summary just repeats in the same way that Barry had, we have the summary slide repeat the conclusion slide that was the second slide of our presentation. So we worked on a method to use software engineering measures for predicting reliability. The results of the method so far as promising.

We think the method can be used in the current review method, and the work is going on on a nuclear application, large OCS.

CHAIRMAN SIEBER: Are there any questions that anybody would have?

MR. WHITE: That haven't been asked already.

CHAIRMAN SIEBER: Right.

MR. GUARRO: What is the time frame for carrying out your next validation?

MS. SMIDTS: I think that we have two years. Is that correct? Yes. We started in December, so we just started actually.

CHAIRMAN SIEBER: Well, thank you, Professor Smidts.

MS. SMIDTS: Thank you.

CHAIRMAN SIEBER: That was a very good presentation, and we appreciate your coming here.

MS. SMIDTS: Thank you.

CHAIRMAN SIEBER: Steve, I think you have -- in fact, you finished early by two minutes.

MR. APOSTOLAKIS: Steve, you are repeating yourself here. You're describing ongoing problems. We've done that.

MR. ARNDT: Then I'll work through it very quickly.

MR. APOSTOLAKIS: Why don't you go through the slides that you like.

MR. ARNDT: I will go through the slides I like.

MR. OVERLAND: I've been in and out randomly, and every time I come in you're giving him a hard time.

CHAIRMAN SIEBER: You should have been here the whole time. Let's let him make his presentation here.

MR. ARNDT: Okay. What I want to do is talk a little bit about future things, particularly things I haven't talked about before. Some of these things we have talked about before, and I'll just give them 20 seconds of time. I'd also like to talk about some things that we're planning on doing, and based on our input from this and other inputs we may revise that.

Continuing new research is planned, with basically trying to investigate different aspects of the assessment process. If you recall from my introductory comments, we're trying to do several things. We're trying to improve the current process, we're trying to make it more qualitative - I'm sorry, more quantitative. I'll get it. And we're trying to work toward the ability to do real risk assessment of this area.

We're trying to provide tools and guidance to NRR so they can do better assessments. And we're trying to coordinate this both internally, both between the PRA groups and I&C groups, as well as in various international and national groups in the nuclear area.

As Barry mentioned, this new work is going to be on one of the three generically approved platforms that actually work on COTS software, and continue to develop this as a potential independent assessment methodology.

As Carol mentioned, she's starting to work on a large-scale application, full life-cycle so we can actually look at all the life-cycle areas, both to assess what's most important in the review, and also to give us some more quantitative measures.

DR. KRESS: You've done this one time expert opinion ranking.

MR. ARNDT: That's correct.

DR. KRESS: So we've got that.

MR. ARNDT: Yes.

DR. KRESS: And one thing you might want to think about is the dispersion, but do you plan some sort of update as you accumulate data as you go through these things?

MR. ARNDT: Part of the process of all this is research program planning both for what programs are we going to do, what we're going to try to accomplish in those programs and things like that. And that's part of the research planning you'll hear about in a couple of months. But also, it's continually reassessing both the methodologies and new methodologies as they become available.

One of the biggest challenges in this area is not only is the technology changing, but the ability to assess things is changing. So you'll notice that in the BNL work, in Barry's work, he did an assessment, in BNL's work - they did an assessment. And talk about some of the future work, we're also going to probably do an assessment. The idea is to update that issue.

In the case of Carol's ranking, you'll notice that the file cases basically validated the experts' opinions so I don't think that's necessary to the point, but the larger-scale study demonstrates the things that give us better predictions of reliability in nuclear-specific applications not working out the same way, then we'll probably do an update.

DR. KRESS: Yes. Well, it's quite a bit more lines of code. You might expect some --

MR. ARNDT: Right. It's a different domain, although there are similarities. I mean, it's a real time system, it's a no-go kind of system and kind of things, but it's different and we would expect some differences.

MR. APOSTOLAKIS: I thought this morning you told us that BNL will think about methods for including software in the PRA.

MR. ARNDT: Yes.

MR. APOSTOLAKIS: The first bullet here seems to say that they have already decided to use a Markov model?

MR. ARNDT: It says one of the things they're looking at, development of a process, Markov model, one of the three platforms to identify the splitting analysis need to support individual features. What we're talking about doing is having them do that analysis at that level.

MR. APOSTOLAKIS: I must say I'm a little cool to the whole idea. Markov model means you have transition rates, and to get anything useful out of it, you have to assume they are constant, and that justifies that. So I would expect them to start with that and think about whether it's appropriate to use a Markov model or something else.

MR. ARNDT: Okay. That's why we're discussing future plans with you. That's the whole point.

MR. APOSTOLAKIS: No, I'm not questioning this, but I'm a little surprised because the impression I got in the morning was that they would essentially have free-hand to look at what's available and try to put things together. And now this says oh, no, no, no, they have already decided to use a Markov model.

MR. ARNDT: Continue review of the database, particularly in conjunction with other database work, and look at some of the quantitative methods for assessing software reliability in conjunction with the other software.

This work I want to highlight, even though I know the Committee is not overly thrilled with Halden's work in the past, one of the areas that they specialize in is the Baysesian Belief Network in combining qualitative and quantitative data, where they've also done extensive work in formal methods. That's one of the areas that they are probably going to present in the May meeting which I will be attending, and we will assess whether or not we want to include that in this continuing work with them. They're one of the leaders in the European nuclear community for formal methods.

MR. GUARRO: Steve, with respect to the database review, I would just suggest that the horizon is kept wide so that you look at some of these egregious type of examples of failures that have been pretty catastrophic, and those are not very many. And you look at them from the point of view of kind of a case study to see what needs to be learned from them. It's not a matter of how many happened and how many trials.

MR. ARNDT: Right.

MR. GUARRO: It's just a matter of what really happened.

MR. ARNDT: Right. And I didn't mention it when I was talking about, but that's one of the specific goals of the international nuclear database, is not so much to come up with reliability data, but it's to understand what the failures are telling us, both from a specific individual failure analysis, and a trending kind of statement, the COMPSIS International Database Program is looking at that for smaller events where we actually had nuclear-specific data.

MR. ROSEN: What I would have wished you had said in response to Sergio's comment was that you would look at the known failures, the most egregious examples.

MR. ARNDT: Yes.

MR. ROSEN: And that you would derive from them the generic implications to the nuclear program from that.

MR. ARNDT: Yes. Absolutely. And that's one of the things that you'll see in the BNL report. But we need to do that more.

We plan on having a new project that's going to look at specifically looking at what kind of models work best in current generation PRAs. The project is specifically looking at the risk importance issues, what is most important in putting a system into a model, and what are the practicality issues associated with trying to put a Markov model, a dynamic fault treaty, or the various issues. This project is specifically designed for that analysis.

MR. APOSTOLAKIS: Now we just said a few minutes ago that the completeness of the requirements is extremely important, so it seems to me it should be -- the next you talk to us, I would suggest that there is a separate bullet for that.

MR. ARNDT: Okay.

MR. APOSTOLAKIS: It's really so important. And if you look at all these events that have occurred, you will see that there was a problem with the requirements.

MR. ROSEN: And I think ultimately you go to what are the regulatory requirements for developing digital software. And some place in those regulatory requirements there should be an embodiment of the principle that the user is embedded in the process from a very early point and continues throughout.

MR. ARNDT: There is a specific requirement, specific regulatory review guidance on requirements. I just don't remember the exact phraseology and level of detail.

MR. ROSEN: For that specific requirement, for the user input from very early-on and continuing throughout the life-cycle of the development?

MR. ARNDT: The requirements, and who needs to specify them, and how they need to be followed and things like that. I don't remember the level of detail.

MR. ROSEN: Well, I think I just told you what I would want to see. It's the result of listening to this discussion, but also a career, a lifetime in doing, not software but doing design work, knowing how systems work, and knowing how to get to a good answer.

MR. ARNDT: Another effort that's going to be ongoing is the review of the draft EPRI report, which proposes a risk-informed approach to a particular software issue; that is the defense-in-depth requirement, diversity requirements. So that is going to be one of our efforts in the near future.

And as I mentioned, we don't know this to be the case, but it could be the first step in the industry's push to use risk-informed ideas in digital system submittals.

There is a little bit of work that's going to be ongoing in the reactor program, particularly trying to develop information to support the risk-informed regulatory approaches that Mary is working on, and also to try and understand better the kinds of issues in software that can have potential issues in pre-application and application. Some of these you raised when you looked at the ACR-700 software.

You asked Carol when her next program should be complete. That's going to be in early FY06 for some of the work that we talked about, that's been published already. We're going to have an additional report on the small-scale validation. It's going to be published this year so that that work could be folded into any regulatory guidance document we develop.

As I mentioned, Barry's new program which hopefully will be one of the generic platforms should be completed in late `05. The first products of the new research program should be ready in `05. The database work is ongoing, and the guidance review depending upon what response we get from industry and various other things should be completed in `05.

MR. APOSTOLAKIS: Well, you said you are developing a plan, a research plan.

MR. ARNDT: Yes. We're updating our research plan basically.

MR. APOSTOLAKIS: So that plan will have new tasks or projects and so on, because from what you're presenting, you're pretty busy already well into 2006.

MR. ARNDT: In this area, yes. It will

not just be this program, but it will be all the rest of the I&C programs, the emerging technology and other programs, as well.

CHAIRMAN SIEBER: There's some pretty basic stuff still in the basic program.

MR. ARNDT: Yes. And one of the areas is systems aspects, things like operating systems and design reviews and things like that, which we've touched on as it affects these kinds of things from this presentation.

MR. APOSTOLAKIS: We heard years ago when we were reviewing the SRP that the Canadians when they licensed - which one was it, Pickering? No, another one. Darlington. They used a mixture of formal methods and testing, and all that stuff. Are you familiar with all that?

MR. ARNDT: Yes. We've looked at that, as well as several other countries' reviews, like the review that was done for Sizewell, and for Choose-B,and some of the ABWR work and things like that. And that's actually part of a product that's going to be published here in a month or two on Lessons Learned from evolutionary reactors.

MR. APOSTOLAKIS: One interesting thing that is related to what we were saying earlier from the Canadians, is that they didn't really use formal methods, but they borrowed what they thought was appropriate. And one of the things they borrowed was some tables where the requirements are specified in a formal language, and maybe that helps. What Sergio said earlier, you know, it enhances their ability to catch problems with the requirements if you do that. Because, as you know, if you use a formal language, then there is no two ways about it. I mean, either you're precise or you're not.

MR. ARNDT: It helps, like a lot of other things.

MR. APOSTOLAKIS: Yes.

MR. ARNDT: And least once you've done your system work, you're software requirements are very tight. We still don't have as much of the system issues solved, but it doesn't certainly more formalize the software requirements.

CHAIRMAN SIEBER: Okay.

MR. ARNDT: And this is just a quick some of the things we're doing to keep up with current work. We've talked about this, a major meeting this year - this is the joint IA, EA, NEA Maryland project to look at validation and verification. There's going to be a meeting in Instanbul. COMPSIS where work through our contractors to stay in touch with other industries and things like that, and of course standard professional things that we try and do.

CHAIRMAN SIEBER: Very good.

MR. ARNDT: And again, we're continuing our work. We're continuing future work to look at different aspects. The goal is always to provide tools and guidance to NRR. Okay. Mr. Chairman.

CHAIRMAN SIEBER: Yes.

MR. APOSTOLAKIS: You stood up and we're finished.

CHAIRMAN SIEBER: We have one of our guests that has to leave. I thought it would be nice to say goodbye.

MR. APOSTOLAKIS: I was wondering where there was a correlation.

CHAIRMAN SIEBER: And our Designated Federal Official stood up too. I'm not sure what that means.

MR. SYKES: You're still in control.

CHAIRMAN SIEBER: Well, I think that's what I want to do. What we will do with the information that we received today, which is very good presentations all down the line is, I will make a report to the Full Committee in April. And what I would like to do now is just go around the room and take a little bit of time for you to give me your opinion of what you've heard today, and suggestions as to what should be in my April report. And, George, why don't you start.

MR. APOSTOLAKIS: Well, I'm still not sure that we're addressing the real issues that are important to us, granulating nuclear power. Perhaps if we had seen some actual failures that involved software and then seen some methods, how these methods, for example, that were presented were consistent or would have found these things in advance, I would feel much better.

CHAIRMAN SIEBER: Yes.

MR. APOSTOLAKIS: At this time, I'm still not sure we're on the right path, so I am willing to be convinced, but I'm not sure that we're really focusing on what's really important to this agency.

CHAIRMAN SIEBER: Okay.

MR. APOSTOLAKIS: Okay.

CHAIRMAN SIEBER: Just as a comment to that, I've asked these same questions on other occasions, and there is no audit data in the U.S. Nuclear Industry on digital I&C because there aren't very many systems. The systems that are there are really sub-systems, and relatively rudimentary systems. There is some European data, but the major data really comes from other industries. For example, the steel mill accident where they dumped a ladle of steel in the middle of the floor. That was a digital I&C problem, as I understand it, and that information wasn't presented. It is available, but right now other than aerospace and commercial aviation, and some process industries, like chemicals and petroleum, there isn't a lot of data out there.

I'm not sure where you go though when you assess what it is this agency should do to assure the integrity of the software systems, lacking that kind of data. And you may want to speak to that.

MR. ARNDT: Well, there are several issues, and most of them were brought up during the course of the meeting. One is that that whole issue is what do we need to do to assess the safety and to ensure the safety of the digital systems as they're implemented. One thing that we could do, as Dr. Rosen mentioned, is to set a particular high threshold requirement that if you're going to do it, you need to do it this way.

CHAIRMAN SIEBER: Yes, I sort of agree with that.

MR. ARNDT: And that's fine. Another thing we can do is to update our review methods and technology to try and better handle, so when presented with analysis we can make a more informed decision as to whether or not it's acceptable or not, going at the same issue from a slightly different perspective.

In trying to develop tools and methods for our colleagues at NRR to do their current assessments, that's what we're currently trying to do. The issue I think that Professor Apostolakis is getting to is, are the things that we are doing either in the reviews that NRR is doing, or the research that we are doing really attacking issues that are going to make a significant difference in the likelihood of a problem. And that's a tough thing to get at.

CHAIRMAN SIEBER: Yes.

MR. ARNDT: And I think we are doing that. We may not have articulated it as well as we would like. There are certainly things that we can do more in that area.

MR. APOSTOLAKIS: Going back to the issue of data - you know, there have been some really spectacular failures, like the Ariane failure and so on. And there are some minor, like the one I mentioned with the fighter plane and so on. It would be very enlightening, I think, to look at those and maybe categorize them in some way, maybe say that this thing will never happen in a nuclear plant, but this other thing might. And then get a basis from which we will start focusing on what's important, a combination of the failure experience and theory. We seem to be jumping into things like, you know, the density of faults. I mean, why is that important? On what basis is that important, because somebody used it?

This is where I get lost. Why are we doing certain things, and what's the basis for those, and how relevant are they to nuclear reactor regulation?

MR. ARNDT: I think it's important that you bring that up because it provides us a background on future interactions.

MR. APOSTOLAKIS: Absolutely.

MR. ARNDT: Things that we need to try and do to inform the committee better.

MR. APOSTOLAKIS: I want you to succeed, Steve. I really want you to succeed. Don't think I -- but I have to give you my honest opinion now.

MR. ARNDT: That's why we're here.

MR. APOSTOLAKIS: I'm not sure we're fitting the right places.

MR. ARNDT: That's why we're here.

CHAIRMAN SIEBER: Jim.

MR. WHITE: Thank you. I'll be brief. Steve, you opened the meeting saying you'd like to be prepared to answer the questions that you expect to be asked about how NRC will do risk assessment. It would help me if you could give us a little scorecard, what are the questions you expect to be addressed, and then lay out your programs to show how you are, and the progress you would expect. Just kind of put it all into context for us.

One thing that we learned on the National Academy Study is we've got a lot to learn from the software engineering practitioner community, and I'm glad to see that you are really trying to get engaged with those folks.

The other thing that we - and that's a really big positive. The other thing we did learn, however, and I know this is controversial, but it's the design of safety assessment rigor in those industries seemed to pale in comparison to what we're expected to do in the nuclear industry. And it seems to me that you're going to have to forge new -- you're going to have to blaze new territory to make that happen, and so good luck on all that.

There is the issue of data, and it looks like you're trying to go out and get the data. And that's going to be a continuing challenge.

With respect to the software community, I think you saw today how hard it is for some of us to understand what they're trying to tell us. We know that they're trying to tell us something that's really important, and we're trying to grasp what it is. And it's not always so obvious to us.

I see that you're beginning to, and maybe you always have, pay attention to IEC standards, which is one thing that we'd recommended. And I'm just about done.

I think it's really excellent that Barry Johnson with your funding is looking at large systems with very high reliability, because trying to assess the probability of failure of a very high reliability system is difficult, as you know better than I do, so I'm glad that you're doing that. And it seems to me that one of the big questions is going to be uncertainty, and how do we handle uncertainty. And I think we have some models from our PRA on the thermal hydraulic-type accidents, how we might do that. That concludes my comments. Thank you.

CHAIRMAN SIEBER: Thank you. I presume

you'll provide us with something in writing?

MR. WHITE: Yes.

CHAIRMAN SIEBER: Thank you.

MR. WHITE: When do you need that?

CHAIRMAN SIEBER: Actually, it would have been handy yesterday. Tom.

DR. KRESS: I was glad to hear Jim mention the word uncertainty, because as everybody knows, it's my hobbyhorse, so I want to agree with that comment. I also want to say, I thought today's presentations were superb. It was much better -- a lot better than we're used to, and wanting to thank the speakers and everybody.

I think this research has some very bold proactive elements that are badly needed, and I'm really glad to see something like this being done. And I applaud the effort. It looks like the program is well-conceived, and the various parts of it actually fit together nicely or complimentary, and each one of them appear to me to be needed for this.

That said, I have following other thoughts. Like George, I think more is needed to justify the use of the Markov model. Now I'm not as skeptical that it can't be used, as George appears to be, but I think -- I haven't seen the real justification for it yet. It wasn't presented to us today, and I think something needs -- you're going to get asked this question over and over, and I think something needs to be done about that.

I think you need some early thinking on what your acceptance criteria are going to be for when you actually get ready to stick a PRA model of digital systems in. When is it good enough, and what are the acceptance criteria? And these need to be ready to think about the uncertainty and the reliability numbers you get, and the defense-in-depth issues. And 0174 may have some in there, but I'm not sure.

I'm glad to hear that the fault injection method can use injection of multiple faults simultaneously. I hope to see more of that, because I think that might be important.

I was also very glad to hear you are seeking some international programs in this area, and I really urge you to continue that. And you might even work on trying to get the industry involved, through EPRI or NEI.

I share George's thought that we might want to think about how to approach the business of the initial requirements. On the PRA completeness issue, you just have to think about it, and think about it, and get enough people, experts to look at it and see if you've covered everything. And maybe you haven't, and maybe you have, and you're never going to know until something happens that you didn't think about. But perhaps if we give it some more thought, it might be helpful.

In the University of Maryland expert opinions, I still think you need to look at the dispersion and factor that into your ranking some way. And I think you need to think about how to update the rankings as you go along, as you get new information. So that's all I have.

CHAIRMAN SIEBER: Okay. Thanks. Steve.

MR. ROSEN: Well, I learned a lot today, and I thought the presentations were very good. Of course, it was easy for me to learn a lot because I didn't know very much to start, but I thought the presentations were very interesting, very useful.

With regard to the University of Virginia's programs, the one on developing an integrated digital system assessment method that the staff can use is, think crucial, along the lines of your comment, Steve, that if you're going to do it, you need to do it this way. It's a very valuable thing for the industry to have the staff's idea in front of them, how this process should be done. They may do it another way, but they'll certainly check the way they do it against your methods, because they don't want to be surprised when they come in here.

With regard to the modeling, risk modeling, I think it's very, very good idea to push the research to figure out what the most effective method is for including digital system modeling in PRAs. It's always been a worry of mine, but I didn't really face it directly in my career because there wasn't that much digital stuff in the plant. We just assumed the failure of the reactor protection system as an initiating event. Its frequency was tiny, but it was there, and then we tried to figure out what the most effective method is. The consequences are very large, the frequency was tiny, but it wasn't very instructive to do that. We need something much better. I'm glad to see that you're focusing on that. We'll be very interested in the results of how one does that.

I'm also glad to see that at Maryland, I guess it is - maybe no, I'm not sure - maybe you can help me with this, but that the first products of the new research will be pilot models integrated into current plant PRAs. Is that Maryland or Virginia?

MR. ARNDT: That's going to be the new project we're starting this year.

MR. ROSEN: That's the new project. At Maryland or at Virginia? Don't know yet.

MR. ARNDT: We haven't decided yet.

MR. ROSEN: Okay. That's why it's so unclear. But whoever does it, how one does it will be of great interest to me, and I'll be thinking about it, having been a practitioner or manager of practitioners at one point in my career. Could we really do it, could we back-fit it to an existing plant? These plants now, the ones I'm familiar with are 20 years old, let's say, sure to be relicensed, sure to have digital systems incorporated in before the end of their operating terms. And so the people who I know will be faced with the problem of integrating into the PRA model, these new systems, and doing it in a way that preserves the integrity of the existing model and results. And so I'll be very interested in how that's done.

So those are my comments. I thought, as I said, I learned a lot and I'm hopeful for the future.

CHAIRMAN SIEBER: Okay. Thank you. I agree that the presentations today were excellent, and

I thank the speakers for coming in and making those presentations, and informing us as to what they're doing.

I'd sort of like to step back just for a second and look at the overall scheme of what it is we're trying to do. Really what you're preparing to do is to write SERs that will approve the use of digital I&C systems in nuclear power plants. And if you're going to do that on a risk-basis, which I think is the way to do it, then you have to decide what your goal, your safety goal is, and what methods either the staff will use, or the applicant will use in order to establish whether or not they meet that goal. And I think that that has to be pretty prescriptive in order to do that, and I would see that as part of regulatory guidance of one sort or another. And that's a project that you ought to be actively engaged in finding that.

Now what kind of systems are proposed is irrelevant, except to the extent that different system architectures have an influence on how risky the system really is. And so you won't be dictating to vendors what the system functional requirements will be, or what is architecture, either software or hardware design should be. On the other hand, you're setting up a performance standard that they ought to meet. And if you do it in a consistent way, I think it's fair across the board, and there is a real basis then to write an SER that says basically there's no substantial risk to the public when these systems are employed.

I think that I would put some additional direction into developing that framework. How is it that we're going to approve the systems? And you've already done it with three systems, and I'm not exactly sure how you do that.

DR. KRESS: Engineering judgment.

CHAIRMAN SIEBER: Well, it goes beyond that. You know, I bought a computer -- I buy a computer about every 18 months for some reason or other, because they turn obsolete like you wouldn't believe. They're either too small or what have you, and so let's say combustion engineering comes out with a digital I&C system. That becomes obsolete pretty fast. And if you're still using 8086s and 486s, and Windows 3.1, I think there's a problem there.

You know, it's like your thermal hydraulics programs, they're relegated to operate on some ancient main frame that it becomes difficult to continue to operate some of these design and analytical codes because you've got to maintain some old, decrepit, antique of a machine for which it was approved. And so there has to be a way to be flexible enough to allow the manufacturers to be able to change processors and some of the architecture inside the machine. Every time you change processors, you're changing the instruction set, because there is an instruction set that goes with a Pentium IV or what have you. And it makes a difference as to what chip you have as to how the operating system performs, so it seems to me that there's an area that needs some attention too. How do you accommodate people's desire to upgrade systems and still establish the fact that that SER applies, or do you have to start from scratch every time somebody wants to change a chip.

DR. KRESS: 5059.

CHAIRMAN SIEBER: Right. 1.174.

MR. ROSEN: Well, that's an extraordinarily good point. We've got two factors operating, and they're going in opposite directions. The life-cycle of computers is going down, and the life-cycle of plants is going up.

MR. APOSTOLAKIS: Somewhere they meet.

MR. ROSEN: In third space maybe.

CHAIRMAN SIEBER: When they're going like this I don't think they meet. That's one of the problems. But in any event, those are some of the thoughts that I had when I was preparing for this, and hoping would be answered. And I'm still hoping. Okay. But I think that that's -- if I were doing it, that's where I would put a little more emphasis, is to figure out what I'm going to do with the applications when they come in.

And so with that, anybody else have any comments or any comments from our guests? Well if not, then I would take this time to adjourn the meeting. Thank you all very much.

(Whereupon, the proceedings in the above-entitled matter went off the record at 3:52 p.m.)