Official Transcript of Proceedings

NUCLEAR REGULATORY COMMISSION

Title:	Advisory Committee on Reactor Safeguards Thermal-Hydraulic Phenomena Subcommittee
Docket Number:	(not applicable)
Location:	Rockville, Maryland
Date:	Wednesday, November 19, 2003
Work Order No.:	NRC-1197 Pages 1-348

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)

MEETING OF THE SUBCOMMITTEE ON THERMAL-HYDRAULIC PHENOMENA

+ + + + +

WEDNESDAY,

NOVEMBER 19, 2003

+ + + + +

The meeting was convened in Room T-2B3 of Two White Flint North, 11545 Rockville Pike, Rockville, Maryland, at 8:30 a.m., Dr. Graham B. Wallis, Chairman, presiding.

MEMBERS PRESENT:

GRAHAM B. WALLIS Chairman

F. PETER FORD ACRS Member

THOMAS S. KRESS ACRS Member

VICTOR H. RANSOM ACRS Member

JOHN D. SIEBER ACRS Member

ACRS STAFF PRESENT:

RALPH CARUSO Staff

SANJOY BANERJEE ACRS Consultant

ALSO PRESENT:

Joseph Staudenmeier	RES
John Mahaffy	Penn State
Ken Jones	APT
Christopher Murray	RES
Jack Rosenthal	AEOD/ROAB
Steve Bajorek	RES
Joe Kelly	RES
Ralph Landry	NRR/DSSA/SRXB
Chester Gingrich	RES/DSARE/SMSAB
Weidong Wang	RES/DSARE/SMSAB
William Krotiuk	RES/DSARE/SMSAB
Phil Reed	RES/DSARE/RPERWMD
David Ebert	ADSTM, Inc.
Birol Aktas	ISL, Inc.
Shandai Lu	NRC/NRR/SXRB
Yue Guan	ADSTM, Inc.

I-N-D-E-X

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

8:33 a.m

CHAIRMAN WILLIS: The meeting will now come to order.

This is a meeting of the Advisory Committee on Reactor Safeguards Subcommittee on Thermal-Hydraulic Phenomena.

I am Graham Wallis, Chairman of the Subcommittee.

The Subcommittee members in attendance are Tom Kress, Victor Ransom, Jack Sieber and Peter Ford.

Dr. Sanjoy Banerjee is attending as a consultant to the Committee.

The purpose of this meeting is to hear presentations from the NRC staff and its supporting contractors about the development and use of use of the TRACE thermal-hydraulic computer code. This Subcommittee will gather information, analyze relevant issues and facts, and formulate proposed positions and actions as appropriate for deliberation by the full Committee.

Ralph Caruso is the designated federal official for this meeting.

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 November 13, 2003.

A transcript of the meeting is being kept, and will be made available as stated in the *Federal Register* notice.

It is requested the speakers first identify themselves and speak with sufficient clarity and volume so that they can be readily heard.

We have received no requests from any member of the public pertaining to make an oral presentation.

I have a few preliminary remarks before we start.

The NRC has a long history developing computer codes to analyze the behavior of nuclear reactors. As part of his regulatory mission to evaluate and assess to computer code to be used by industry to demonstrate the safe operation of nuclear power plants, the NRC has developed several of its code. And the ACRS has been very supportive of the NRC having its own codes.

About ten years ago during the review of several advanced reactor designs and with the burgeoning availability of advanced computer at greatly reduced cost, the staff decided to consolidate its support for a number of separate computer codes into one code that could perform multiple functions.

They also decided to take advantage of increases in computer power to restructure the codes to reduce maintenance and improve the ability to include new information or modeling techniques as they developed.

We understand that the result of this effort is now known as the TRAC RELAP Advance Computational Engine, otherwise known as TRACE or TRAC-E.

As part of its oversight function, the ACRS reviews analytical codes developed by both the staff and the industry. This Subcommittee meeting today is the first in a series of meetings to acquaint the ACRS with the progress that the staff has made with the development of TRACE. During these meetings we expect that the staff will provide the members with information about the technical foundations of the code, its implementation, assessment against experimental data and its application to regulatory issues.

At the end of this process, we envision a meeting with the full Committee and appropriation of a letter providing our advice to the Commission on the program. The Committee will also enter interim letters if it decides that an issue raises a question that requires more immediate action.

On a personal note, I've been on this Committee for six years, and this effort has been going on while I've been on this Committee. And I would very much like to be able to report a major success story, either today or in the near future. And if there are things that are preventing this being a major success story, we would like to know them now.

With that, I will now proceed with the meeting. And I call upon Dr. Jack Rosenthal of the Office of Nuclear Regulatory Research to begin.

DR. ROSENTHAL: I'm Jack Rosenthal. I'm the branch chief of the Safety Margins and Systems Analysis Branch of the Office of Nuclear Regulatory Research. And I was asked to make a few introductory remarks, and then my staff advised me not to say too much because they have a lot to say in the next two days. They thought that would be better. But I just wanted to say just a couple of words on process and couple of words on product.

Process wise, it must be at least a year ago we had one of these one day marathon meetings on all things calculational and experimental. And at that point, Dr. Wallis advised us he wanted to get more involved and in greater depth.

Over the spring we did bring in some of our contractors to discuss the experimental program, and it was piecemeal. And then in reality, we begged off on briefing the Committee while we were busy working on the code. But now we're ready to move forward with briefing the Committee. I think you'll find this meeting of substance.

And then what I'd like to do is when you identify related areas that you want to hear more about; a specific numerical treatment, an expanded explanation of some experiment, we'll keep a list going. And then over the course of the winter we'll schedule other meetings and attempt to go into those areas that you want. So we'll be able to get through the presentation.

But we look forward to working with you on the code substance wise.

NRR used the RELAP5 that RES developed to do the audit calculations of AP1000. We have used TRACE to do other audit calculations of AP1000. We are developing capability for TRAC to do ES-BWR audit calculations. So the code is a prominent role in independent analysis to guide the decisions.

We use the code to resolve or to work on GSI 188, which involves hydrodynamic blow down of a steam generator. And we used at least the PARCS aspect coupled with RELAP in this case to work on an issue called -- on boron dilution issue, GSI 185.

So we really are using these products to address safety issues directly and to perform audit calculations with other work presented to us. So it really is important to us.

I think that we've come a milestone now. We consider the consolidation effort completed. And now we can spend our efforts at making the code better. That is something that we're all anxious to do and it's starting to happen, so it's quite exciting time.

With that, let me turn the mike over to Joe.

CHAIRMAN WILLIS: Can I ask you something before Joe starts?

DR. ROSENTHAL: This is what he warned me about. I'm sorry.

CHAIRMAN WILLIS: Maybe I should be asking just the group in general.

This is a design task, making something. Usually one starts out with a sort of specification.

Say I want this thing I'm creating to do A, B, D, F, G. And you have pretty clear statements that it's got to be ten times as fast as previous codes, do away with certain problems with all codes which are listed, and so on and so on.

And you also have measures of success, which says, you know, it's got to be running on this kind of computer at this speed or something, some sort of thing it's got to do.

And then when you've designed it and built it, you check has it done all the things we set out to do? Is it, in fact, ahead of specs? Is it doing better than we thought it would do or is it way behind? You've got some kind of measure of how well it's doing.

Is that the sort of thing we're going to hear from Joe?

DR. ROSENTHAL: Joe has some of that in his presentations.

CHAIRMAN WILLIS: Well, I'll be looking for it. I hope it's there. Thank you.

MR. STAUDENMEIER: Okay. I'm Joe Staudenmeier. I'm head of the Codes and Models Development Group in Jack's branch. I'm going to try to give you a brief overview of other things we're doing and then mostly concentrate on TRAC code development.

The objectives I hope to accomplish today is to provide an overview of our activities going in the branch in terms of thermal-hydraulicing system code development and get feedback from the Committee on topics that they would like to see more details on in the future.

As you know, this is the first in a series of meetings that we're going to have discussing TRAC. And today is more of an overview level type meeting. I think we have quite a bit of technical detail, but it'll show you enough to get a feel for everything going on. And we're in the process of setting up future meetings where we go into a lot more detail on topics of your interest.

CHAIRMAN WILLIS: Now you're not just going to talk about development? You're going to talk about what it can do, aren't you?

MR. STAUDENMEIER: Yes.

CHAIRMAN WILLIS: As it has been developed, I understand?

MR. STAUDENMEIER: Yes. We're going to talk about what it can do.

Okay. First, give you an overview.

Even though we're developing TRAC, RELAP5 has disappeared off the map. Right now it's in a stable maintenance mode with some limited development going on. It's main priorities in development are fixing outstanding bugs. There's never any shortage of outstanding bugs in these codes. And we're also putting in level tracking into RELAP5. That was a big request from our CAMP members, which is our international code applications and maintenance program. So that's being implemented into the code also.

CHAIRMAN WILLIS: These bugs are bugs that have been around for 25 years or something?

MR. STAUDENMEIER: I think most of them are recent. I mean, you fix a bug in one place, new ones pop up. I don't think we have any that are 25 years old on the list.

CHAIRMAN WILLIS: Well, the code is pretty old.

MR. STAUDENMEIER: I think RELAP5, well Vic knows better than I. I don't know, maybe late '70s or early '80s.

DR. RANDOM: 1975 it started out.

MR. STAUDENMEIER: Yes. The current, I guess, renovation of it RELAP5 mod 3 was early '90s.

CHAIRMAN WILLIS: So these are bugs in the genes, something.

When you say a "bug," is this a bug that gives you the wrong answer, the numerical answer, or is it just a computational glitch?

MR. STAUDENMEIER: Sometimes it gives you wrong numerical answer. It could be numerical instability, lots of oscillations or it could be something that makes the code outright die and stop running.

CHAIRMAN WILLIS: Okay. I understand the latter. How do you know the former, the wrong data is wrong?

MR. STAUDENMEIER: Well, generally it's people notice these bugs in assessment calculations where they're comparing to experimental data and they see something, wrong behavior of the code compared to the data trends or something like that, or heat transfer coefficients off by a factor, something that they know is wrong.

CHAIRMAN WILLIS: So it's wrong against data?

MR. STAUDENMEIER: Against data or numerically. I mean, it could be lots of oscillations where it should be smooth behavior, something like that.

CHAIRMAN WILLIS: Okay. Okay.

MR. STAUDENMEIER: Okay. Future development for RELAP. RELAP isn't going away in the near future. I think it's going to be around at least for the next four or five years. It's going to be maintained in stable maintenance mode, anyway, fixing bugs and finishing level tracking. And there'll probably be some, after level tracking is finished, if there may be some next feature that CAMP members want in the code that happens in the next few years, that may go in.

CHAIRMAN WILLIS: Well level tracking is sort of, to me, going from liquid to gas or some vapor. But actually you have another kind of level tracking where you go from, say, a bubbly pool to a drop suspension. Is it fine between regimes rather than a real level?

MR. STAUDENMEIER: That's the level tracking we're talking about from --

CHAIRMAN WILLIS: Between regimes?

MR. STAUDENMEIER: -- liquid continuous regime to vapor continuous regime.

CHAIRMAN WILLIS: Something like that? Okay.

MR. STAUDENMEIER: Yes. From like bubbly flow to mist or something like that.

CHAIRMAN WILLIS: But it could be the end of an annular film or something like that, too?

MR. STAUDENMEIER: It could be. Our level tracking isn't that smart yet to know where the end of an annular film is, but that could be enhanced in the future.

MR. BANERJEE: Are you going to tell us more about any of this or just give us a list?

MR. STAUDENMEIER: For RELAP5, I'm not going to talk anymore about RELAP5.

MR. BANERJEE: Do those problems also come up with TRACE?

MR. STAUDENMEIER: TRACE, I would say, has many of the same type of problems that RELAP 5 does.

MR. BANERJEE: You'll discuss it there?

MR. STAUDENMEIER: Sure. We'll be talking about problems with TRACE. And you could probably look through the problems and there are essentially classic problems that come up in two phase flow's codes.

DR. RANDOM: One thing in the past year on PTS, was saw some deficiencies identified, you know, in RELAP5 but no plan to correct those? Is that still the situation?

DR. ROSENTHAL: Yes. Right there as this meeting is going on there is a pressurized thermal shock review meeting, and Ivan Catton is discussing all these issues with the staff. And some of those issues will be -- I think we're addressing the issues.

DR. RANDOM: Did you say Ivan Catton?

DR. ROSENTHAL: Yes. He's --

DR. RANDOM: A consultant or --

DR. ROSENTHAL: As a consultant to us.

Now I'm talking too much. I apologize, Jeff.

We have a peer review ongoing of the pressurized thermal shock program. It's a multi-day review in greater depth than we could do it with ACRS. And he's addressing those issues with them. And is being provided with the ACRS transcript, so he knows the concerns that you raised.

And even yesterday we agreed to rerun actually fracture mechanics calculations with the different heat transfer coefficient to see what the sensitivity was.

But if we could -- PTS is going on on the tenth floor of the building, I guess what I'm saying.

CHAIRMAN WILLIS: So I guess the answer is you haven't fixed the problems, but you're working on them?

MR. STAUDENMEIER: Okay. I guess ideally we would like by the end of maybe the next four or five years, that TRAC would run RELAP decks so well that everyone would want to use TRAC instead of RELAP5, and we have a smooth program for people to move over that. But we'll assess a need for maintaining RELAP5 beyond that time or when we get to that time.

Current NRC use of RELAP5, AP1000. It's been used for some preliminary calculations for ACR-700, I believe, PTS and operating plant calculations.

CHAIRMAN WILLIS: Are you going to show us that TRAC is better than RELAP5?

MR. STAUDENMEIER: We're going to show you that it's at least as good as it later on, in a later presentation. And not for everything yet, but I'd say a lot of applications. There's some still some bugs remaining to be worked out.

MR. BANERJEE: Well, ACR-700, the dormitory is completely different in many features. How are you using that for that?

MR. STAUDENMEIER: Well, the reason RELAP5 is being used for some preliminary calculations, the

Koreans through our CAMP program have contributed a RELAP5 component called Can-Chan, that's supposed to be a CANDU channel that did some small modification to I think the horizontal flow regime map for rod bundles. I don't know how good it is. It wasn't done with a detailed development process I don't think. I think it was some sort of a first modifications you would think about doing. But ultimately, I think for CANDU analysis there's probably going to have to be a new horizontal flow map and rod bundles developed, and other components that are different in CANDU than in typical pressurized or boiling water reactors. So that'll be being looked at over the next couple of years to prepare for when CANDU comes in for design certification that there will be a program looking at what code development and what test data has to be taken to get in a CANDU analysis capability into the codes.

MR. BANERJEE: Will you use RELAP5 or TRACE?

MR. STAUDENMEIER: TRACE will ultimately be used, and essentially the deck is being developed in RELAP5. We'll use the translation capability that's being developed to run RELAP5 decks in TRACE to run that deck in the future in TRACE. But right now since RELAP5 has that Can-Chan component, it's going to be used for some preliminary calculations, is my understanding. I haven't been involved in the CANDU effort, but I think that's what's going on.

Okay. PARCS is our 3-D reactor kinetics code that we have coupled to both RELAP and TRACE.

Some recent developments in PARCS is we now have burnup capability to burn out a fuel cycle. Currently 2.5 is our current version, it's under active development. Current priorities is to complete the documentation.

Users manual is in pretty good shape. We're working on a theory in programmers manual for PARCS. Right now it's coupled through PVM, a parallel virtual machine, some software developed at Oak Ridge for loosely coupled calculations.

We found that we're taking runtime hits in that as the processors have gotten faster, we're spending more time in communications. So it's gotten to the point where 20 to 30 percent of the CPU time is being used in essentially the overhead of using PVM in PARCS. So we're going to be replacing that with some tighter coupling into TRACE. Some direct subroutine calls there is a library call.

Recently we've improved the runtime by

almost a factor of two within the last month or two, other than the PVM coupling time.

We're doing assessment with PARCS and TRACE coupled together against Peach Bottom turbine trip data. In the future, we'll probably do Peach Bottom stability data. Right now we're doing Ringhals plant stability data, that was measured as part of an international standard problem.

And right now, I guess the main thing we've seen with these coupled calculations for BWRs, at least that there's no real good way to do reduced channel mapping like reduce the number of neutronics channels to thermal-hydraulic channels. We've been running with 30 thermal-hydraulics channels but found that when you do this mapping, it gets kind of arbitrary, and you start mixing thermal-hydraulics channels together that aren't really behaving alike because they're close together neutronically. So we're moving to a one-on-one channel mapping. Within the next few months we're going to be running Ringhals with all 648 channels.

CHAIRMAN WILLIS: Well, that's an improvement, isn't it?

MR. STAUDENMEIER: Yes. Well, right now, I mean it's not really going to cost us much in runtime because PARCS takes 10, you know 20 times as much time for time steps as TRACE does with 30 channels in the code. So, I mean we could go to 300 and be even.

CHAIRMAN WILLIS: So the thermal-hydraulics is more efficient than the kinetics?

MR. STAUDENMEIER: It was surprising to me, too. And that's why we're also looking at PARCS speedup efforts to see if there's more performance we can ring out of PARCS also.

DR. RANDOM: We passed in a one day your point kinetics. So, you know, the neutronics part has always been a trivial part.

MR. STAUDENMEIER: Right.

DR. RANDOM: So now it's grown to quite a significant burden in the computational sense.

MR. STAUDENMEIER: Yes.

DR. RANDOM: What would you say, like half and half or more?

MR. STAUDENMEIER: Well, PARCS is about ten times as much now with our current noding. I would have expected about half and half with our current noding based on some past reading kinetics calculations I've seen with RAMONA and also some TRAC-B coupled 3-D kinetics. But I'm not sure where the performance programs are in PARCS or why it's taking so long per time step. But we're starting to look in to that, do some performance profiling of the code and speeding it up. And we're also looking at parallel processing within PARCS itself.

CHAIRMAN WILLIS: How does this compare with the commercial codes or the codes used by industry? And PARCS is an NRC development, isn't it?

MR. STAUDENMEIER: I know there's a Studsvik product called the S3K where I think they have a very fast code. They model every channel in the code hydraulically. And I think they do things like lot ejections in under a minute, and things like that.

CHAIRMAN WILLIS: But Studsvik, that's a Swedish government --

MR. STAUDENMEIER: This is Studsvik of America that developed these codes. Studsvik is still centered in Sweden, but their fuel management and transient analysis software is developed at Studsvik of America. They have an office in Boston and Idaho Falls, and in Gaithersburg now because Studsvik and Scanpower merged together and Scanpower has an office in Gaithersburg.

DR. RANDOM: What do they use for the thermal-hydraulic part?

MR. STAUDENMEIER: Well, they're moving to RAMONA because Scanpower had the RAMONA code and they're looking at integrating the Studsvik's neutronics with RAMONA thermal-hydraulics. But they developed some thermal-hydraulics in-house, which is essentially equilibrium modeling.

CHAIRMAN WILLIS: So how do you compare with something like what Westinghouse or GE uses?

MR. STAUDENMEIER: I'm not familiar with Westinghouse's 3-D kinetics. I don't think they have highly cut -- I don't think they have their 3-D coupled to a sophisticated thermal-hydraulics code because PWRs you don't really need that for a lot ejection.

CHAIRMAN WILLIS: I think GE's working on it.

MR. STAUDENMEIER: GE has 3-D kinetics in TRACK G. I don't know the runtime performance of their code with 3-D kinetics. I've seen calculations, but I don't --

CHAIRMAN WILLIS: They use for a benchmark. And yet they saw we can do thus-and-so in this time, you can compare with what you can do.

MR. STAUDENMEIER: Yes. You could ask the NRR that in a proprietary meeting or a closed meeting, since they have the code in-house they can run something for you.

CHAIRMAN WILLIS: Oh, yes. It's a good measure of your performance. I mean, if you can really say I have something which does more quicker than industry, then that's an achievement.

MR. CARUSO: A question about, you're going away from PVM. Are you going to -- one of the ideas behind going to TRAC was to be more modular, to allow you to plug new technologies into the code. Will going away from PVM compromise the modularity at all?

MR. STAUDENMEIER: No. Actually, what there will be, inside the code there will be essentially a 3-D kinetic solver interface, call interface that should be general enough that you could plug in any 3-D kinetic solver underneath that, or that's what our design goal is on that anyway. And I don't see any reason why that can't be done. And it would be either linking to a PARCS library or using a PARCS module in the code and Fortran 9 determines if you had a program --

DR. RANDOM: This coupling between the neutronics and thermal-hydraulics an implicit type of coupling or --

MR. STAUDENMEIER: Yes. Not right now,

it's not implicit coupling. And that's something that will have to be looked at, especially for BWR calculations where you get strong coupling between the voids and the kinetics.

I guess another thing is PARCS limits the time step size, too. TRAC can take bigger time steps than PARCS, but it gets to outrun PARCS right now.

John Mahaffy wants to say something on it.

DR. MAHAFFY: Yes, this is John Mahaffy.

I am aware that there is a research project at Purdue in which they are doing an implicit coupling between PARCS and TRAC. They really haven't put out any kind of concrete results from that. I don't believe it's even funded by the NRC. I think that's a side project as part of somebody's thesis. But that work is in progress.

MR. STAUDENMEIER: Yes. When things settle down, that's definitely something we're interested in, especially getting implicit coupling between the voids and the kinetic solution for PWR calculations.

CHAIRMAN WILLIS: Right.

MR. STAUDENMEIER: Okay. The future development for PARCS, Oak Ridge is developing their TRITON cross section generation suite, which in the past it's been called SCALE and something else before that. And the new thing about TRITON is they now are going to have a lattice physics solver so that they can actually have better development of cross sections for reactive fuel bundles.

So PARCS is going to be able to hook into that in the future. Right now it can read HELIOS cross section files, which was a Scanpower code, lattice physics code and we should be able to connect it up to CASMO, which is the Studsvik lattice physics code and we'll look at that in the future.

DR. RANDOM: HELIOS is that database? Or HELIOS also happens to be the name of a server at Purdue.

MR. STAUDENMEIER: Oh, yes, I don't know about that.

DR. RANDOM: But it doesn't the data file stored there. That's not the meaning of this, right?

MR. STAUDENMEIER: Probably not. No. HELIOS is the name of a code, a Scanpower code as part of their fuel management package.

Okay. Finally on to TRACE. TRACE stands for track RELAP advanced computational engine. And it's the current focus and future focus of NRC thermal-hydraulic system safety code development.

The goals of a modernization project were to develop a modern maintainable and extensible code that had all the capabilities of our previous codes RAMONA, RELAP5, TRAC-PWR, TRAC-BWR.

CHAIRMAN WILLIS: Is there something incompatible about saying I'm going to make a modern version of an antique? I mean are there some different characteristics now that codes could have in a modern world that the old codes never had? It's not just a question of encompassing their capabilities? There are some things that modern codes could do that they could never do before?

MR. STAUDENMEIER: That's right. And --

CHAIRMAN WILLIS: Have you found out what those are and detailed those and set those as a part of your objectives?

MR. STAUDENMEIER: I mean, TRACE will do many things that those codes never could. I guess, another code you could add to that list of things it he COBRA series of codes, COBRA-TF, COBRA-TRAC or COBRA-NC has you'll see in a later side --

CHAIRMAN WILLIS: If you had a customer here, I would say that the goals of a project ought to be to meet some needs of a customer. And you have the list of what the customer wants. And I don't see that here at all.

MR. STAUDENMEIER: Well, I think the minimum list is to have the capabilities of all these codes.

CHAIRMAN WILLIS: You mean that they were meeting the needs --

MR. STAUDENMEIER: And then go beyond that from there. But that's the base level of needs that we have, is to --

CHAIRMAN WILLIS: But if there was some sort of -- if you could sort of look at the needs of the agency and say in the future we're going to need to do this, this and this, was want to run a Monte Carlo thing with 10,000 different code runs instead of just the five or ten we could do in the past; then you're going to have to develop that capability.

MR. STAUDENMEIER: Right.

CHAIRMAN WILLIS: But I don't see that.

Did anyone go through that sort of intellectual -- not even that, intellectual. I mean, at the hearing --

MR. STAUDENMEIER: At the start of the project there were a list of views or capabilities written down.

CHAIRMAN WILLIS: That you try to aim for?

MR. STAUDENMEIER: A lot of them were make the code more robust, make it faster. There were some additional capabilities listed like dissolve gases in the liquid to look at gas coming out of solution one on accumulator depressureizes. I don't know the complete list. But this code will be well suited. It's well suited to do everything the past codes can and it also gives a platform or a development platform that's easy to add future needs.

As you know, this is mainly a reactive agency. We react to things that happen outside and we can't --

CHAIRMAN WILLIS: There's now risk-informed regulation. In the old days when you had it the other way, we just had to make one calculation. It's deterministic.

To do risk-informed, you may want to do a whole spectrum of stuff in order to say, well, you've got to make some probability assessments and statistics and so on. It's a different world.

MR. STAUDENMEIER: Oh, yes.

CHAIRMAN WILLIS: So we need something more.

MR. STAUDENMEIER: And I think you'll see we're developing things to do those type of sensitivity studies you're talking about.

CHAIRMAN WILLIS: Yes. So it may mean orders of magnitude improvement and capability. But it may mean carrying the uncertainties along while you do the calculation and actually not just giving a result, but giving uncertainties where they're unsolved.

MR. STAUDENMEIER: And that is a goal that eventually. We don't have an uncertainty analysis methodology building yet, but that's something we're going to be doing in the future, we have planned.

CHAIRMAN WILLIS: You see what I'm getting at? It's going to be a modern code.

MR. STAUDENMEIER: Yes.

MR. BAJOREK: Joe? Just one other point to add on that.

This is Steve Bajorek from Office of Research.

Another one of the major goals was oriented towards resources on each of these codes. Each one of these RELAP TRAC-P, TRAC-B took it's own maintenance group in order to keep the code up to date, add models to that as technical improvements are made to -- grid spacers are one. You would have to add it to all of those codes. By combining these codes and doing the code consolidation, it's enabled the agency to focus its resources on one set of coding, maintaining one code so that as improvements to the models are made, we can do it across the board and benefit all types of analysis; large break, small break, PWRs, BWRs and have one team that's able to use that same code to extend it to things that we really don't understand at this point, like the ACR-700 and other types of reactor systems.

MR. STAUDENMEIER: Yes. I mean the first goal of this was to encompass all the capabilities and also modernize the software architecture so that it's easy to add new models into. And I think we've had great success in doing that. The code is much earlier to develop for than any of those predecessor codes. You'll see that later in John Mahaffy's talk.

A lot of the things that you used to have to do by hand like memory management of components and setting pointers within a big global container array, software compilers can do that for you because Fortran 95, which we're using, has dynamic memory allocation, derived variable types. So we have names for everything everywhere and we don't have big container arrays and pointers into that, which was a source of a lot of errors in these codes in the past.

So, I mean the first part of this success is getting a modern maintainable software architecture where you can add new models to the code and extend it in the future with things you haven't thought of yet. And I think we've been very successful in that and very successful in encompassing the capabilities of the original code. And you'll see that we're now in active development of things well beyond that in one of my later sides.

MR. BANERJEE: What was the reason to stay with Fortran historical when most modern CFD codes are written in C now, all the commercial ones?

MR. STAUDENMEIER: I think part of it is that we have a lot of people that know how to program in Fortran. Fortran 90 or 95 is, I think, is a lot cleaner to program in than C or C++. It doesn't let you wonder off into bad areas because C++ is a very big language and there's a lots of things you can use in that poorly.

Fortran 90 or 95 has essentially all the things that we need for our code in it.

MR. BANERJEE: So why do Fluent and Star CD and CFX use C?

MR. STAUDENMEIER: I don't know. You'd have to ask them.

MR. BANERJEE: So you decided that you know why you'd use Fortran in spite of all these other guys doing --

MR. STAUDENMEIER: I know why we use Fortran, yes. And I don't know why they use C or C++. But I think it was a good choice to use Fortran. Now that Fortran 2000, the next iteration is going to have object oriented features. So anything that we would have liked about C++ that that would have that we would like to us, we could use that and add it into the code. But I think we have a nice clean readable code now in Fortran 90 or 95. I mean, there's not really any features that C++ has that I think we really need to use.

I mean, you could think of features that you could use just to use, but I don't think we've been really limited in our code development by the language.

MR. BANERJEE: You had a point, John?

DR. MAHAFFY: Yes. This is John Mahaffy.

The decision, in fact, it wasn't something that was arbitrary. Vic Ransom may remember, there was a committee of illustrious experts that was conveyed, what was it, seven eight years ago?

DR. RANDOM: Seven, I think. Yes.

DR. MAHAFFY: Yes. And the purpose was to begin setting some of the parameters for some of this code development.

Two of the people on that committee were, in fact, computer scientists. But Lloyd Le Broc was one of them and there was an illustrious guy from UCLA whose name that I'm blanking on right now.

But, they both agreed with the idea of Fortran 90. And the fellow from UCLA, the way he put it was that Fortran, at that time 90. We're into 95 now and moving into 2000 whatever. But the way he put it was the language itself, as Joe said, contains what you need to know, but it's more compact language. There's less you need to know to be expert. His advice was that unless you have a core of people who are doing the program continuously all the time, you don't want to deal with C++ because it takes a greater effort to maintain the kind of level of expertise you need in C++.

And the NRC was talking about, and if it gets into this easily maintainable, easily extensible architecture, you're going to be rolling in engineers for a module here, a module there. And, you know, you can't guarantee they're C++ gurus. That's touch to do.

And I've been there. I've almost done that. And I've been back. And I think that advice was good.

We have within our own history, there was a fellow that worked largely on the architecture that we'll be discussing in a little bit here, named Skip Dearing from Los Alamos. Skip Dearing, you know, most of his adult professional life while it was available, programmed in C++. His conclusion after working on this was we were better off sticking Fortran 90.

So, I mean there are bits and pieces of sound evidence that, you know, first of all the decision was not arbitrary. And secondly, it was a good decision. It's given us something that's easier within the context of the NRC to keep up and extend.

MR. BANERJEE: Is it because the numerical framework of the system is pretty invariant? Nothing changes much, whereas in most codes you tear out the numerics and replace it somewhere, the internal projection method?

DR. MAHAFFY: No, no. The numerics varies. I go in there and change things from time-to-time, other people do. Fortran 90 is an easily modularizable language. There's lots of things you can do there. Again, as Joe said, the things you need that are in C++ are in Fortran 90. We're able to do it.

I mean, we can sit down. This is probably not the context. You could have a later meeting to look at programming details of this, if it's interesting to you. But it's held up very well for us over the years.

MR. BANERJEE: Okay. And it's easily parallalizable?

DR. MAHAFFY: Yes. And I'll address that when I get up and talk to you.

MR. STAUDENMEIER: Yes. I don't know if you've looked at Fortran 95, but I mean the vector and array language, and it is very powerful and it can make for very clean programs.

MR. BANERJEE: I've worked with 90, yes.

DR. SIEBER: What language are they teaching young engineers in college these days?

MR. BANERJEE: C.

MR. STAUDENMEIER: Java.

DR. SIEBER: Java.

MR. BANERJEE: Java and C, that's what they're teaching.

DR. SIEBER: Okay.

MR. KELLY: Joe Kelly from Research.

And the decision to stay with Fortran is, as they said, a rational one although I did lobby against it at the time. But it's also part of a substantive of a larger decision. And the larger decision was to consolidate these capabilities into an excellent code rather than starting from start and doing a complete total rewrite not having a single line of code the same, which is exactly what would have happened if we had decided to make such a radical language switch as going from Fortran to C++.

Once the decision to stay with a code was made, that allowed one really strong positive, was we already had an operating code. So that as we upgraded the architecture and migrated to Fortran 95 we could make a change in the way the internal communications was done and compare the answers, and it better be identical to what it was before. If not, we knew we had a problem.

So all along the development path as we're making these drastic changes to the code architecture, the code was testable. Whereas, if we had started from scratch and had to do a complete total rewrite, it would have been years before we got to the position where we had a code that could do calculations.

So there were a lot of positives for

sticking with Fortran 95 including these capabilities into an existing code.

CHAIRMAN WILLIS: Okay. Joe, could we move along?

DR. RANDOM: One thing further on that, there was at the time the decision was made, there wasn't a revolutionary model that came to surface that would replace, say, the two flow model that was embedded in these codes. And so there wasn't a lot of motivation for starting over again, I don't believe. As far as I know, that's still the situation today.

MR. STAUDENMEIER: Okay. Yes. I think--

CHAIRMAN WILLIS: The only thing I'd add to this slide here is does this code fulfill the requirements as spelled out in the Reg. Guide and SRP that you guys wrote?

MR. STAUDENMEIER: I'd say not yet. We haven't done an extensive review of the physical models, an assessment of the physical models yet.

CHAIRMAN WILLIS: At least you should be consistent with your own guidelines.

MR. STAUDENMEIER: We will be doing that. And we will be consistent with that Reg. Guide.

CHAIRMAN WILLIS: So we can look at your guidance and we can check off the direction, it did run, said had to be done at that point?

MR. STAUDENMEIER: Right now I'd say no, because we've stuck with the existing correlations in the code to keep it a running code. We're going to be examining the correlations in the future. You'll see some presentations tomorrow from Joe Kelly about developing new physical models. And I think you'll see that his development process and his physical models is following things as stated in they're stated in the Reg. Guide.

DR. RANDOM: Joe, one thing I'd like to hear myself as you go along, is how peer review has been brought in to kind of guide the overall project. Certainly in the past, the NRC utilized peer review a great deal. I mean, some of the people might not have liked it, under the Tom regime and whatnot, but it was pretty powerful in terms of keeping everything acceptable, I guess, within the technical community and across the board. And I'd be interested in what has been done along those lines as you've evolved the code and what your plans are for the future on that.

MR. STAUDENMEIER: Yes. In terms of peer review, other than the initial meeting where they talked about -- well, there was an initial meeting of the experts, then there was a meeting in Annapolis where worldwide people convened to talk about requirements for future codes.

Beyond that there really hasn't been much peer review outside some use of the code by CAMP members or NRR and providing feedback based on that. So this is our first venture into, I guess, peer review or outside review of a code. I think it's an important part of the code development, and I think we really -- once we get our first assessment on hand, I think we really need to go out and have everything peer reviewed and look where to go beyond that for the future.

And Office of Research within the past year has initiated more of a peer or a push to having important office products peer reviewed. And I think that's a good idea. And, hopefully they'll put the money behind it to go out and actually do that. And I'm all for that.

DR. RANDOM: Well, certainly when you look at some of the documentation, I mean you can't get a two or three day meeting. A person has to really sit down with them that material and kind of pour over it if they're to provide anything meaningful in the way of feedback.

MR. STAUDENMEIER: The documentation will actually probably be the limiting thing in when we can actually go out for a peer review. All the documentation is in draft form essentially now.

DR. RANDOM: Right.

MR. STAUDENMEIER: And needs to be updated to a final form so we can actually give it to outside people to look at and provide comments on.

MR. BANERJEE: In particular, the peer view for the correlations used and everything which relates to physics is very important. So, in the past these things have been pretty ad hoc. Somebody saw some relationship in the literature and stuck it in, and then it would come up front of this advance core review group or something, and see that there are other possibilities.

So, it's pretty essential to do that. Because if I recall, a lot of the correlations even today are pretty arbitrary anyways. They're chosen because somebody knows they exist, or they may not be the best. And then they've been tuned.

I can give you a couple of examples off line.

MR. STAUDENMEIER: Yes, I'm sure. I know that's been true in the past. I mean, we definitely want technically defensible correlations that are recognized as good correlations and that meet the test against experimental data, code assessment against experimental data. So that is a good area for a detailed peer review or in the future we could have a presentation where we present the whole closure model package that's used in the code.

MR. BANERJEE: And how it's implemented, actually.

MR. STAUDENMEIER: Right. How it's implemented is important in many cases on how you average quantities over cells and what density you actually feed in or what void fraction you feed into evaluate the correlation. There's many ways you could average or pull numbers in to get numbers back out. So that is very important in these types of codes.

Okay. I guess a second goal of this project was maintain our investment --

CHAIRMAN WILLIS: I think we've all read this.

MR. STAUDENMEIER: Input models, okay. Are there any more questions? Provided productivity enhancing graphic analysis environment, all in --

CHAIRMAN WILLIS: I think that when you say as good as or better, you need -- there are probably several metrics, it's not just one metric in terms of comparison with data. It's other things.

MR. STAUDENMEIER: Right now, I guess our metrics are a comparison with data and code robustness, is what I'd say. And we haven't worried so much about one time yet, but that's something we're going to start looking at now.

CHAIRMAN WILLIS: And the adaptability to different platforms and things like that?

MR. STAUDENMEIER: Right. Okay.

DR. SIEBER: So this was run on something other than a 1950s workstation, right?

MR. STAUDENMEIER: Right. It'll run on your Windows PC.

DR. SIEBER: Okay.

CHAIRMAN WILLIS: Does it run on the Mac?

MR. STAUDENMEIER: Yes. Joe Kelly develops on a McIntosh. Mac OS10, not OS9. And so if you have OS10 you can run it.

DR. RANDOM: Another thing that would be interesting on, from reading the NRC's research program and whatnot, it's amazing the number of organizations involved in this. I mean, you have ISL, you have Penn State, you have the NRC people working on it. You have Los Alamos working on it. How do you coordinate all that, you know? And certainly from my experience in the past, I know today with the internet and things like that, communication is better, certainly. But still, it's who is the driving force, you know? Who ultimately makes the decision as to what; we're going to do it this way or that way?

MR. STAUDENMEIER: I mean the NRC is the driving decision, essentially. We have periodic codevelopment meetings where we gather everyone together, hash out issues and come up with a way to move forward. So --

DR. RANDOM: Do you resolve differences among the different developers that are working on it?

MR. STAUDENMEIER: Yes, people throw out ideas for what -- what their ideas how to solve it, other people will provide feedback and maybe say, "Oh, I think this is a better direction to move on that." And it works surprisingly well.

I mean, Jennifer Uhle was carrying the load essentially by herself for quite a long time in this development project. I mean, right now we're probably at our peak number of in-house NRC developers. I'm sure it was -- it gets fairly chaotic at times now with all the people involved. And I really don't know she handled it. She did an enormous job in just keeping the project moving forward when she was in charge of it, because she was essentially carrying --

DR. RANDOM: How many people do you have working on it in-house?

MR. STAUDENMEIER: Well, I'll get to that in a slide.

DR. RANDOM: Okay. Go ahead.

MR. STAUDENMEIER: But we have -- okay.

DR. RANDOM: I'll wait until then.

MR. STAUDENMEIER: Yes. Actually SNAP, SNAP won't run on Mac OS10 yet. There's some Java 3-D libraries that haven't been ported to there yet or aren't fully running yet. I think I saw that there's beta versions of those out. So within the next year, SNAP will probably run that, too.

DR. KRESS: SNAP an acronym?

MR. STAUDENMEIER: SNAP is symbolic numerical -- I don't know.

CHAIRMAN WILLIS: We have a whole presentation. We have two hours presentation.

MR. STAUDENMEIER: Yes. You'll see plenty of SNAP in the afternoon.

But essentially our model of computer SNAP is our frontend, and it's a graphical user interface. It's essentially a computing environment for all the

NRC codes. You'll develop models in SNAP, modify them, clip a menu item in SNAP that says go run this model. It will either run it locally on your machine or it can connect to a Linux cluster sitting off somewhere else, or any other machine sitting off somewhere else. Run the job and come back and post-process answers within SNAP.

You'll see it this afternoon. It's a very powerful environment. I don't know the last time you've seen SNAP. But it's very much improved.

DR. RANDOM: Who is developing SNAP now?

MR. STAUDENMEIER: Ken Jones at APT. He has his own small company called APT.

DR. RANDOM: So it started out on ISL and then it's migrated to Ken Jones?

MR. STAUDENMEIER: It started out as ISL. I think Ken at the time was working at either ISL or it was Syntec, I guess, at the time.

DR. RANDOM: Yes.

MR. STAUDENMEIER: And Ken was working in maybe one of their subsidiaries or as a contractor. Ken's back here. He can answer that this afternoon when he comes up to talk about his history.

DR. RANDOM: He was the XMGR developer, wasn't he?

MR. STAUDENMEIER: He did a lot of work with XMGR, yes. Okay.

CHAIRMAN WILLIS: And you've already talked about the rest of it.

MR. STAUDENMEIER: Yes. We're supporting commodity platforms because you really can't beat price performance and even outright performance, there's not much performance to gain. By spending ten times as much as you spend on a PC, you don't get much performance gain by that extra amount of money.

We're supporting standard programming languages to make everything portable and network and software file formats, Fortran 05, Java, there's some bridge code in C and C++ that's linked in.

Yes, Java, SNAP is written in Java now. Originally that was written in C++ and it was moved over to Java a few years ago.

Using standard networking protocols.

CHAIRMAN WILLIS: So most people can use it; that's really the message here?

MR. STAUDENMEIER: Right.

Standard file formats we're sticking to because there's lots of openly available software and tools to manipulate this file formats. So we don't want to reinvent anything. We want to leverage all that work out there that's done by other people.

CHAIRMAN WILLIS: You don't want to get something which is just peculiar to the NRC?

MR. STAUDENMEIER: That's right.

CHAIRMAN WILLIS: Okay.

MR. STAUDENMEIER: And when I started here, I mean all the codes, a lot of codes relied on essentially proprietary software packages like SPLAY or things like that you had to pay $20,000 a year for a license or things like that. And we want to move away from that.

In my opinion, our tools are much better than they were then and they're using openly available software protocols.

TRACE development team. The internet and desktop workstations has made it possible to develop a code like this without the physical infrastructure of a national laboratory. Back in the '70s or '80s you essentially needed a super computer to develop and run these codes on. Now PCs are fast enough and for our codes, they're essentially as fast as the fastest computer you can buy. Because your codes don't really take advantage of the advance architectures that are in what are called super computers right now.

CHAIRMAN WILLIS: Okay. We can read this

one, too, I think.

MR. STAUDENMEIER: Okay. In-house, we have five people, five separate bodies, I'd say about half of our time we spend on code development. Quite a few contractors have been involved in the effort. Applied Programming Technology is Ken Jones' company on SNAP. ISL, Los Alamos and Penn State and Purdue.

DR. RANDOM: Is University of Maryland helping out? I saw something in some of the research stuff that looked like they were involved.

MR. STAUDENMEIER: University of Maryland had been doing some code assessment for us. And Professor Wolfe retired in the last year or so, so there's no one there to lead the assessment effort anymore, so we don't have work being done there now.

DR. RANDOM: Where is that?

MR. STAUDENMEIER: University of Maryland.

DR. RANDOM: Maryland? Right.

MR. STAUDENMEIER: Yes. And the professor there that, I guess, headed the assessment effort, he has retired so we don't have work there anymore. But we have had work there in the past for assessment.

DR. RANDOM: Okay.

DR. KRESS: Was that where the transient area model was being developed?

MR. STAUDENMEIER: Excuse me?

DR. KRESS: The interfacial area model, was that where that was being developed?

MR. STAUDENMEIER: No. That was being developed at Purdue, the interfacial area transport, I believe. I don't know. Is that correct, Joe? Okay.

DR. RANDOM: Also, UCLA they're doing the subcool boiling model development. But I don't know how closely that's tied in with this.

MR. STAUDENMEIER: I'll cover that in a later side.

DR. RANDOM: Okay.

MR. BANERJEE: But one of our concerns when we saw that presentation was not the quality of the work, but the quality of the tie-in with TRAC.

MR. STAUDENMEIER: Of which?

MR. BANERJEE: Of TRAC or whatever.

MR. STAUDENMEIER: The subcool boiling model?

MR. BANERJEE: Yes.

MR. STAUDENMEIER: Well, it isn't tied in with TRAC yet, but it will be.

DR. RANDOM: Yes.

MR. STAUDENMEIER: And I'll get to that in one of these slides.

DR. RANDOM: And you'll also cover how this work at Purdue will get --I actually don't have anything on that in my slide, because that's not in our near term plans to integrate interfacial area transport, at least I don't think it is. But eventually we're looking at integrating that into the code to eventually to replace all standard --

CHAIRMAN WILLIS: Our recommendation when we looked at this work and the Penn State work and the other work, was don't have a research program where they develop it there way and then assume that five years from now somebody's going to translate it into TRAC. TRAC has got to go along with the work they're doing so that they're developing parameters and structure which is capable with TRAC from day one.

MR. KELLY: This is Joe Kelly from Research.

And that's very true. And at least for the Purdue work, I'm going to be doing that over the next couple of years. And I have a design to task to develop a small pilot code. And the purpose of that pilot code is to determine what is the most efficacious way to implement an interfacial area transport within a two code framework. Because there's certain different ways you can do it. You need two bubble momentum equations or can you get by with one. If so, how do you -- and so and so forth. So that's the idea, is to have a research tool. And I will be working to do that.

MR. BANERJEE: When are we going to review this program? Because that are controversial aspects as to whether such an approach actually can work? And if it does work, what does it buy you? So I think it would be interesting to have that up for some review.

CHAIRMAN WILLIS: Yes. We'll do that.

MR. BANERJEE: Because it's been going on for a long time --

CHAIRMAN WILLIS: We haven't heard about it for a long time.

MR. STAUDENMEIER: And we can put in a request as wanting to see details of that experimental program in it. We would try to accommodate that.

Configuration control is maintained at the NRC. Later presentation today will show essentially our configuration control and testing.

Current status. Architectural change is nearly complete. We're trying to work out the last bugs in the RELAP5 translation in running.

We're debugging existing models. We have improvements in physical models and numerics in active development.

CHAIRMAN WILLIS: How about the input text? Can you just take RELAP input text and use them?

MR. STAUDENMEIER: Right. You can do that. I'll go to the next slide -- not everyone yet, but we have --

DR. RANDOM: You have to do that through SNAP, right?

MR. STAUDENMEIER: Through SNAP, yes.

DR. RANDOM: One question that would be interesting is the sort of trade off between SNAP exclusively and improving the input capabilities of TRAC, say, which have been quite crude from the beginning and I think still are probably not very user friendly.

MR. STAUDENMEIER: And that's right. I mean, SNAP has always been seen as the new input processor. Essentially SNAP is -- right ASCII input processing and TRACE is still there in essentially TRACE format which is very similar to TRAC-P format and TRAC-B format. I guess we see as phasing that capability out over some period of years, that SNAP is to the point now where I think almost anyone would want to use SNAP. There's still some remaining bugs to be sorted out. But SNAP has these capabilities you're talking about like loop closure and things like that.

DR. RANDOM: Well, are you eventually going to -- why not just integrate these two together, you know, so that basically SNAP provides the interface to TRACE on the same machine?

MR. STAUDENMEIER: That's what is going to happen. SNAP is essentially going to develop your input model in SNAP. It dumps a file. TRACE reads that file in and you don't -- right now there's an ASCII format file its dumping. In the future for RELAP it dumps a binary format file that TRACE some reading of. But in the future you probably, ten years from now, wouldn't even think about ASCII input decks or -- at least people learning the code from now on probably won't think that much about ASCII input decks. They'll see SNAP as their input processor.

And you'll see in this afternoon's presentation that SNAP is now very powerful and there's very few reason why you wouldn't really want to use. I mean, there's some small bugs to be worked out, but it's very powerful for most uses and greatly increases productivity during renodding -- well, let me talk about that in my later.

Okay. Current priorities, complete RELAP file support, complete interim reflood model, which you'll hear about from Joe Kelly tomorrow. Some condensation work, Joe Kelly is also working on, he'll talk about tomorrow.

CHAIRMAN WILLIS: This is available for use?

MR. STAUDENMEIER: TRACE is available for use. Interim reflood model isn't into the base version.

CHAIRMAN WILLIS: Yes, but it's available without these improvements, is it?

MR. STAUDENMEIER: Right. It is available without these improvements. And people are using it.

CHAIRMAN WILLIS: How about the runtime?

MR. STAUDENMEIER: Well, I'll talk about runtime in a couple of slides coming up.

CHAIRMAN WILLIS: Okay. Let's talk about all these things later so we can move on then.

MR. STAUDENMEIER: Yes. Current priorities. We need to add a rod bundle interfacial drag model and complete modularization of the interfacial drag. That's one thing we're missing in our correlations package that TRAC-B and RELAP5 had. That's a simple addition. It's going to be a best correlation for rod bundle interfacial drag and bubbly--

CHAIRMAN WILLIS: So if you haven't done the second bullet, how do you know it's any good?

MR. STAUDENMEIER: Well, we have done some preliminary assessments so we know what the big problem areas are and we know where it does fairly well right now. And Steve Bajorek will talk about that in a talk tomorrow. But we need to get to the point where we'll stabilize the code. It's almost at that point now. We'll stabilize, go through more complete assessment and then identify things after we've done this last bit of model improvement and stabilization.

CHAIRMAN WILLIS: So you'll stabilize but not fossilize?

MR. STAUDENMEIER: Correct.

MR. KELLY: Joe Kelly from Research again.

And I can give a little bit more information on the rod bundle interfacial drug because it actually was the code assessment that was done at the University of Maryland that identified a deficiency in the high pressure PWR small break LOCA type conditions in the current TRACE models.

And they also went in and put in -- and checked it for those conditions. Then when I was doing the interim reflood model, I ran into the same problems and I checked it for those conditions for the low pressure level swell conditions and it did a much superior job to anything we had now. And so that's when the decision was made to go ahead and put it in and redo all of the assessment with that model.

MR. STAUDENMEIER: Yes. And we've also done some BWR related calculations with -- included in special code versions, and it's done a better job in those -- predicting void profiles in those situations, too.

DR. FORD: Excuse me. The second bullet. Will we be seeing a demonstration of some of these calculation versus physical data?

MR. STAUDENMEIER: Tomorrow Steve Bajorek has some.

MR. BAJOREK: Yes. Yes. Tomorrow I'd like to try to go through a summary of all of the assessments that we've done in the past year. I'm not going to go back to 2002 and look at the interfacial drag, but between now and tomorrow I'll try to get a little bit more on that. But I want to give an indication on how the code is performing with respect to the code consolidation, show some of the flaws that we need to fix and what our plan is for the next year, and what assessments we're going to be picking up on in order to try to get the job of improving the code, proving its accuracy and quantifying what we've got there.

MR. STAUDENMEIER: I think that's a good topic for maybe a future meeting where we could have a full two day meeting on just showing code assessment results. I mean, you could easily fill up a two day meeting with that.

DR. FORD: And how much is the runtime improved? You say "improve runtime performance." A factor of ten?

MR. STAUDENMEIER: No. The runtime, a priority is to improve the runtime performance. In some models it actually runs faster than the old code. I'd say, most cases it runs slower right now. But we haven't concentrated on runtime improvement and optimism yet. We're just concentrating on stabilizing the architecture and getting the code working robustly.

DR. FORD: Wasn't that one of the --

MR. STAUDENMEIER: That is one of the final goals, and I think that we will be getting big runtime improvements. And actually, I have a slide coming up where I just fix the physical model that improved runtimes greatly. I think the things -- we have to worry about fixing our physical models first. I think they're causing a lot of runtime and robustness problems.

And in running these codes, usually the biggest performance hit isn't actually the actual runtime for time step. Even if it was a factor of two greater. I mean it's parts sometimes during these calculations where the code just bogs down and runs at real low time steps and have a solitary behavior, and it's usually due to bad physical models or bugging the code somewhere and not the actual time per time step taken in trying to refer to the time step.

CHAIRMAN WILLIS: Are you talking about runtimes of days or something like the old codes rather than minutes?

MR. STAUDENMEIER: If you make a big enough model, you can make it run as long as you want. I mean, most things that we look at, I mean --

CHAIRMAN WILLIS: Regulatory needs someone wants to sit down with this code, have a question and get an answer within the hour, not next week.

MR. STAUDENMEIER: And I think most things you can get answers within the hour.

DR. FORD: Well I thought in the very

beginning of the program, as you had mentioned earlier on as to what your expectations were, that (a) an absolute is that it must predict the physical phenomena. But the other thing I would have thought is that you must decrease the runtime.

MR. STAUDENMEIER: Yes. And as I said, we are going to decrease runtime and that is a goal of ours.

CHAIRMAN WILLIS: Orders of magnitude. Computers improve by orders of magnitude every, whatever it is, 2½ years or something. You should be doing the same thing with this code. It's ten times faster every three years.

MR. STAUDENMEIER: I mean, the code does follow Miller's law. It doesn't violate Miller's law. As computers get faster, it gets a lot faster with the computers --

MR. BANERJEE: But nothing in the code follows Miller's law.

MR. STAUDENMEIER: I mean, some things it does -- you'll see, John Mahaffy will talk about some parallel processing capability that you could get runs like wall clock speed ups on, not reducing the actual total CPU time. But there's other advanced numerical stuff that we're looking at that could increase runtime implicitness of the numerical methods, that's where we think we'll see a big runtime increase.

But I think right now our biggest runtime increase is going to be debugging our physical model package --

CHAIRMAN WILLIS: Is someone going to show us?

MR. STAUDENMEIER: You'll see a slide in a couple of minutes if you just wait, where I'll show you how improving a physical model will improve your runtime.

CHAIRMAN WILLIS: Okay. So you're actually going to show data. On all these measures of success, you're going to give us evidence?

MR. STAUDENMEIER: That's right.

CHAIRMAN WILLIS: Great.

MR. STAUDENMEIER: Legacy deck support. Input decks from these codes will run with little or no modification.

Okay. SNAP. You'll see plenty of SNAP this afternoon, so I'll skip over this.

Right now we've finished pretty much the corridor oriented details and we're moving towards more of an engineering oriented interface, and you'll see examples of that this afternoon.

CHAIRMAN WILLIS: Some members of the Committee, not necessarily those who are here, don't understand why you use SNAP rather than going for something that's commercially available.

MR. STAUDENMEIER: Well, SNAP's at a commercial company and it is available now.

CHAIRMAN WILLIS: I know. But it's specific for this -- than something that's there already for other purposes and out there in the marketplace.

MR. STAUDENMEIER: I don't know of any products like that. I mean, we'd have to contract with -- do a sole source contract with some software company that has a similar product, I guess, and get them to adapt it to all our codes that we want to put in. But actually now you'll see SNAP has created this open environment that pretty much anyone can take their code and write a plugin for SNAP without modifying the core of SNAP and make their code with SNAP if it has this sort of a component junction type of paradigm.

So someone could take RETRAN and write a SNAP plug in for RETRAN and RETRAN could use all the infrastructure of SNAP or any other code that looks like that could use the infrastructure of SNAP.

DR. RANDOM: Does SNAP also handle the output display, like the XMGR?

MR. STAUDENMEIER: SNAP goes well beyond. SNAP right now uses XMGR. It's now called AcGrace. XMGR has become an open source project on the internet called Grace. And then AcGrace is all virgin of it. That links to all the NRC codes. But it's going to go well beyond that in the next few years. It does animations. Right now, it doesn't do it yet, but it'll essentially take your graphical input mask and it will animate that in output. It'll do what the MPA did or it will connect off and do interactive capabilities with calculations. You can make masks like MPA masks and animate that mask.

DR. RANDOM: Well, in that sense then is TRACE than just a module of SNAP?

MR. STAUDENMEIER: It's the computational engine.

DR. RANDOM: All right.

MR. STAUDENMEIER: I mean SNAP is the front end and back end, and that's what it's advance computational engine. It feeds -- or TRAC essentially into the computational engine post-process results.

DR. RANDOM: Well, that seems like the logical way to go.

MR. STAUDENMEIER: I mean, it would be more of that -- I mean essentially the original idea was to remove all input processing from TRAC, but it turned out we are limited in the number of Java developers we had and had more Fortran developers, so more of that was kept in TRAC. In the future, we'd like to move more of that input processing out of TRAC and move it all over into SNAP and have it just -- TRAC just picks up this binary snapshot of this data. The system at the beginning of the calculation runs through it, dumps this out. But SNAP post-processes this stuff. So I mean the ASCII interface will essentially disappear in the future.

DR. RANDOM: In that sense are you able to observe the calculation as it's proceeding?

MR. STAUDENMEIER: Yes.

DR. RANDOM: In a graphical sense?

MR. STAUDENMEIER: Yes.

DR. RANDOM: Yes.

MR. CARUSO: Is SNAP going to be also used for CONTAIN and maybe MELCOR --

MR. STAUDENMEIER: CONTAIN support will be finished the end of December, I think is the current schedule. MELCOR is sometime next year.

MR. CARUSO: Okay.

MR. STAUDENMEIER: Or parts will be integrated into SNAP.

MR. CARUSO: Okay.

MR. STAUDENMEIER: Okay. Now we get to runtime performance. Everybody's been asking about runtime.

DR. KRESS: Before you go there, you know one of the traditional ways to improve runtime is to lengthen the finite difference time step at the expense of accuracy. When you compare, say, these runtimes to different codes like TRAC, are you using the same time step or are you getting the same accuracy, or how --

MR. STAUDENMEIER: We're getting the same accuracy and there's some time step controlling sizes that limits time step sizes based on rate of change of stated variables in the code.

DR. KRESS: Yes, you have --

MR. STAUDENMEIER: We don't truly limit by truncation error yet. That may be something to look at the future as having a measure of truncation error. We don't look at that yet. If things are changing fast in our variable, it limits the rate of change of how fast the variable can change and reduces the time step based on that.

Okay.

CHAIRMAN WILLIS: So it's longer than the code that you had before?

MR. STAUDENMEIER: For runtime performance, actually the small model which run really fast and you don't really care that they've slowed down so much right now, run slower than the --

CHAIRMAN WILLIS: This may surprise you. I must be completely naive because, again, I start with the assumption that you're trying to make orders of magnitude improvement rather than just run as fast as it did before.

MR. STAUDENMEIER: The orders of magnitude improvement were never in time for time step. They were always looked at being able to take larger time steps and decreasing the number of time steps in a calculation. And work is being done --

CHAIRMAN WILLIS: The time isn't computer time. It's the physical time being modeled.

MR. STAUDENMEIER: Right.

CHAIRMAN WILLIS: And if the computer can make the same number of calculations faster, if it can, now it must be able to run the code quicker?

MR. STAUDENMEIER: Oh, yes. I mean, if I compared this to the machine that was it running on when we started the project. This is on the same machine, essentially, that the two different codes--

CHAIRMAN WILLIS: Okay. But if you look at machines --

MR. STAUDENMEIER: This is two different codes on the same machine.

CHAIRMAN WILLIS: Okay. But if you look at machines, you've probably got an improvement which is really significant?

MR. STAUDENMEIER: I mean, that wouldn't be a fair comparison.

CHAIRMAN WILLIS: How much is that?

MR. STAUDENMEIER: To take either one of these codes and compare it to how fast it ran on a computer seven years.

CHAIRMAN WILLIS: Yes. If a guy from NRR wants to run a calculation with this thing, how much faster can it be run today than it could be run when you started?

MR. STAUDENMEIER: And I can't remember what was available seven years ago. But the calculations are much faster.

I know when I ran calculations seven years ago, they took a lot longer time than they take now.

CHAIRMAN WILLIS: Is it ten percent faster or ten times faster? What is it?

MR. STAUDENMEIER: Well, I'd say it's more on the order of five to ten times faster now with the computer speed.

DR. MAHAFFY: This is John Mahaffy.

Figure a factor of ten.

CHAIRMAN WILLIS: A factor of ten. Yes.

MR. BANERJEE: Why is the time going up?

MR. STAUDENMEIER: Why is the time going up? We haven't tracked down what the big runtime hit is. This is measuring truly performance per time step, essentially. So it's for the small problems, the small one do problem.

MR. BANERJEE: The time steps are the same?

MR. STAUDENMEIER: Time steps are the same, essentially. I mean, they're slightly different because we're having little bugs fixed here and there to make the progression of the calculation slightly different.

So essentially we're taking 1.7 times more per time step on simple calculations.

Fortran 90, depending on how you implement interfaces and call interfaces, different compilers have different hits, runtime hits. This is in a specific compiler, specific computer. And these numbers may be different on a different computer and a different compiler.

This is on our -- compiler on an AMD Athlon chip --

DR. SIEBER: This is all 32-bit.

MR. STAUDENMEIER: Right. It's as 32-bit. The code is double precision, but it's 32-bit chip essentially.

DR. SIEBER: Yes. On the other hand, with the new 64-bit chips you got a lot more of addressable memory. So, you don't have to use virtual memory.

MR. STAUDENMEIER: No. We're not using virtual memory yet.

DR. SIEBER: Oh, you aren't.

MR. STAUDENMEIER: Our calculations are well within the limits of 32-bit memory addressing. Except for maybe post-processing and graphical output files.

DR. SIEBER: These codes are intering codes, right?

MR. STAUDENMEIER: Yes, you'll see a presentation on that, the merits and how that works.

DR. SIEBER: So is double precision good enough to allow the itering to occur without hunting on a PC?

MR. STAUDENMEIER: Yes, we think so.

DR. SIEBER: Okay. Have you ever sat down and looked at the answers to see where the iteration-- how the iteration converges?

MR. STAUDENMEIER: I haven't specifically done that. I know John Mahaffy has done that, and he's added some new checks in looking at residuals as they converge versus time sets. And you could possibly ask him about that during his presentation.

DR. SIEBER: Okay.

DR. FORD: I just want to make sure that I understand. You said that the time steps are the same between that used in TRACE as compared to TRAC, therefore the accuracy is the same?

MR. STAUDENMEIER: Yes.

DR. FORD: The accuracy in terms of observation versus calculation?

MR. STAUDENMEIER: Well, I mean, I'll cover that in the next slide. I mean, we fixed some bugs and make the answers better, but essentially the accuracy is close to the same, yes.

DR. FORD: And so this --

MR. STAUDENMEIER: It'll lay curves essentially over the top of each other, in most cases.

DR. FORD: And so this increase in runtime is due purely to the difference in complexity between TRACE and TRAC?

MR. STAUDENMEIER: Well, I think it's due largely to the ability of Fortran 95 compilers to optimize code. And they're sensitive in some ways of transferring data between -- in subroutine calls and things like that. And we have to track that down and see what's going on. But it's been demonstrated in the past that different compilers don't handle some things very well and we have to go back through and look and see how we're transferring data. And essentially, we're going to have to do some runtime code profiling and see where this extra time is being taken up in the new code and look at what we can do to fix that. But I think we should be able to get the 1-D things back down to the TRAC, Fortran 77 level. We just have to find out what's slowing it down now.

And if you see down at the bottom of the list, the largest problems are running faster in this code per time step, and that's partly due to a new matrix solver that's in the code. So when you have multiple vessels or even just one large vessel and multiple 1-D connections to that vessel, this new matrix solver speeds things up quite a bit.

CHAIRMAN WILLIS: If I want to run a reactor model as to BWR stability or something, and I want to run a model, every channel instead of having just hot channel and the rest, that's a tremendous increase in the number -- amount of things I'm going to compute.

MR. STAUDENMEIER: Right.

CHAIRMAN WILLIS: I mean, it's a hundred times as much or something. So you need to have a corresponding increase in speed in order to say well now do this.

MR. STAUDENMEIER: Well, I don't think it will be a problem. I think we'll be able to with our Peach Bottom and Ringhals work. Like turbine trip modeling every channel in the core, we're going to do calculations in under an hour. I think. I'm confident of that. Because we do them very fast now.

MR. BANERJEE: Where is the code spending its time?

MR. STAUDENMEIER: I don't know yet. We have to do some runtime profile. We haven't gotten to that point to track down where the performance hits are.

MR. BANERJEE: Where does TRACP-F77 spend its time?

MR. STAUDENMEIER: It depends on the problem. Like this last problem, the AP-600 with multiple 3-D components and many 1-D to 3-D junctions, the last time I checked, which was when I was doing AP-600 calculations, I did profiling and actually found a performance improvement at the time with how some matrix multiplications were being done. It was spending essentially half it's time on the matrix solution for that every time step. And I put in a small modification and it cut it essentially, the matrix solution time down in half and we spending a quarter of its time in matrix solution then.

So, I would expect that this new matrix solver, since it's faster than the old code, it's spending less of its time in the matrix solution and more in just filling up matrices with the coefficients of the equations and things like that and transferring data back and forth between components and things that need it.

DR. RANDOM: Was the level if implicitness seen between the old version and the new in terms of interface drag and treatment --

MR. STAUDENMEIER: Yes, the level of implicitness and those things is the same. One thing that's been improved is if you had flow loops, 1-D flow loops that came out of, say, a radial face and then went back around after a while and connected into an axial interface, you had to use semi-implicit numerics in the vessel. But now with the new matrix solver, it's been changed that you can run sets everywhere with those type of connections. But essentially it's the same level of implicitness in everything.

We have some work ongoing that's looking at increasing the implicitness first in interfacial heat transfer. And then wall heat transfer and fully implicit 1-D components. And then move on to look at full implicit 3-D components.

DR. RANDOM: This drop in performance must be due to the change in the way you modularize the code, I guess, and communication between the different parts of it?

MR. STAUDENMEIER: Yes. That's what it is and how Fortran 95 optimizes these new code constructs.

I know of some places in the 3-D vessel where stepping through matrices is done in wrong order in IJK space in a multi-dimensional array. That's stepping through the last index first and looked at -- you can probably get like ten percent speed up on the large vessel problems in that case for that. We have to search through the code and look for where the runtime is. I mean, that's all it comes down to, is we're going to look through and study it and try to solve it.

DR. FORD: We're never going to let you get off this graph, I'm afraid.

I've got a higher level question.

MR. STAUDENMEIER: That's too bad. The next one --

DR. FORD: Is there an easy answer to the question as to what's the value of TRAC? The accuracy you say is the same. The individual codes that it's modeling, like TRAC-P and things like this, and yet the runtime is greater. So apart from the academic enjoyment of having an all seeing, all dancing code, what's it value?

MR. STAUDENMEIER: As I said, it's not finished yet and the runtime will get faster. Right now it's at the stage of the project where we've completed these architecture enhancement and debugging things. And you'll see on the next slide it runs much faster, it some cases with some bug fixing --

DR. SIEBER: We may not get to that slide.

MR. STAUDENMEIER: Here it comes. TRACE

Robustness. Preliminary code assessment results have identified some robustness and physical model problems.

I recently had some error corrections to the annual-mist interfacial drag in critical flow model and we've greatly improved robustness.

CHAIRMAN WILLIS: It's not just robustness. It's not just robustness when it's just reducing its time steps. Is that when you mean by robustness?

MR. STAUDENMEIER: Well, robustness also means the code used to just shut down sometimes with minimum time steps. It's the minimum time step wants to reduce it further, but it can't because you've limited the minimum time step you can run at. So robustness is one way to characterize that is how stably it makes it through a calculation.

And after these improvements, the code gives better results in almost every problem I've seen and vastly better results in lots of problems.

Like, you can see, TLTA took 142000 time steps before these corrections. Now it's down to 17000 time steps. SSTF, another BWR LOCA test, more than a factor of 40.

CHAIRMAN WILLIS: So you'd expect --

MR. STAUDENMEIER: Browns Ferry more than--

CHAIRMAN WILLIS: --run time, wouldn't you--

MR. STAUDENMEIER: Well, yes, this will vastly -- these weren't the tests that I used to check runtime.

Semi-Scale, reduces it by -- I don't know, like on the order of less than 10 percent but the improvement in accuracy of what PCT vastly improved in that calculation.

LOFT, about 30 percent decrease for a small break.

Here's a case where the LOFT calculation, a large break, wouldn't make it through the calculation with these improvements. It runs through and it's running faster up to the point of where the original code stopped running also.

And other than interfacial drag, there's a bug I suspect somewhere in interfacial heat transfer causing some similar robustness problems in some other calculations. Actually, Joe Kelly I think has identified at least one place in interfacial heat transfer. And we'll go assessment or else look through our other correlations and see where the code is causing problems and you could see some other step change in runtime improvements when we get out the clutches in our correlations package.

DR. FORD: Not being a computer guy, but what's the benefit of this? You're showing that there's a decrease in the time steps to do a specific calculation. Well, so what?

MR. STAUDENMEIER: You can do more calculations if you run it 40 times as fast. You may be able to do more sensitivity calculations on the accident that you're looking at or things like that, or you just get the results back faster. It could change from being limited by how fast it takes the code to run something to being limited by analyzing the results. But also sensitivity calculations for uncertainty analysis, you can make lots of parameter variations, and look at how sensitive the results are to the things you're looking at and changing.

DR. FORD: Yes. And there's less things to go wrong, presumably. But that's not born out by your previous graph.

MR. STAUDENMEIER: As I said, this is totally different runs and --

DR. FORD: Okay.

MR. STAUDENMEIER: Before you were

complaining it wasn't running fast, now I show it runs fast and now it's something else. What good is it when it runs fast?

DR. SIEBER: He's never satisfied.

MR. BAJOREK: Joe, this is Steve Baj9orek again.

I think what might be being missed is the improvement overall between what the code does and what it's doing on the new platforms.

If we look at the last three cases on here, the LOFT and the Semi-Scale, I can't remember the exact numbers, but Semi-Scale, for example, I think is running in a couple of hours or so to get one of the transients done.

MR. STAUDENMEIER: It was about 300 seconds, actually. But at least this one. There's another Semi-Scale one I didn't run that may have been taking hours. If it was taking hours, I think it'll run much faster now if I run it.

MR. BAJOREK: But the Semi-Scale cases are running in a relatively short period of time. I don't know if it's minutes or a couple of hours. But running similar cases with a vendor's code several years ago, running Semi-Scale cases of similar complexity took days, closer to a week.

LOFT, the large break and also the small break, again I think they are running now in a couple of hours or, you know, an afternoon, something like that. Again, when we had done assessments with a different code to run similar cases, in fact the L3-1 in particular, it took days to run.

You know, so we don't have exact numbers on that, but if we're looking at the overall increase in performance, we're now running assessments that same day, that same afternoon that had previously taken us quite a long time.

Now, one of the other things that I think is very important, is we do have some cases that are still taking days to run. And as we're getting away from code consolidation and spending more time on the constituent models, we're finding the bugs and clutches. And I think the first three cases are good examples of when we've had the time to look at the models, find the bugs and fix them, we're taking those remaining cases that have been taking days to run, and those are also going back to the point where they can run in hours or minutes.

DR. SIEBER: Rather than dwell on runtime, I think there's another thing that you said that makes me curious, and that is that you get better answers, which I interpret meaning more accurate answers. How do you know that answer A is more accurate than answer B?

MR. STAUDENMEIER: Most of the cases I've showed, we have --

DR. SIEBER: You can adjust --

MR. STAUDENMEIER: We have experimental data, so most of these I've showed are test facilities. So a big comparison in test facilities. And there's other measures of the calculation, like one may be very oscillatory, that's an obvious numerical instability. When you fix it, it gives you smooth curves.

DR. SIEBER: Right. I don't think of that as accuracy, though. It's more of a stability issue. But you can adjust the answer you get, right, to some extent?

MR. STAUDENMEIER: I mean nodalization you can adjust.

DR. SIEBER: Yes.

MR. STAUDENMEIER: The answers somewhat by changing nodalization. Other than that, like going in and changing the correlation parameters, you can't really adjust the things. I mean, we have a correlations package that's been developed and compared to separate effects data. And we run them against these integral type test facilities which I showed for these runs to show that the whole code is holding together.

I think with a pass codes like REPLAP4 had a lot of adjustable parameters --

DR. SIEBER: That's right.

MR. STAUDENMEIER: -- that you could set in the input. Things like RELAP5 and TRAC and those codes, the big step in those was removing all these adjustable parameters in those so that the users can't have a big impact over the answers.

DR. SIEBER: But the correlation coefficients are really just nobs. You only set them once, though.

MR. STAUDENMEIER: That's right. I mean, they're nobs -- I mean, some of them are purely empirical correlations, some of them are semi-empirical based --

DR. SIEBER: Right.

MR. STAUDENMEIER: -- on some sort of physical model. And you have to set those coefficients in the thing, I mean we're not at the point where we can -- it's too complex to get them for first principles. You have to set them with some sort of test data, compared to some sort of test data. And then you live with them from there on, though.

DR. SIEBER: That's right. They're built in.

MR. STAUDENMEIER: Right.

DR. SIEBER: One final question in this area, if I accept that it is more accurate and you're basing that on test data, how accurate is it really? Are we talking five percent, ten percent? And what parameters, output parameters tend to give you the worst levels of accuracy?

MR. STAUDENMEIER: Well Steve Bajorek will cover that in more detail and show you some actual plots. I don't have plots here to show. I probably should have threw in a plot or two. But I was showing actual results for Steve tomorrow.

MR. BAJOREK: Well, we'll go into that tomorrow.

DR. SIEBER: All right.

MR. BAJOREK: That's been an issue in code development for a long time, what matrix should you use to try to gauge the accuracy of a code.

DR. SIEBER: That's right.

MR. BAJOREK: I'm going to talk about that tomorrow and show how we're going to try to get away from a simple subjective approach which has been used through the code consolidation and up to now to try to get some quantifiable matrix on the code performance. And that's an important part of our focus over the next few years.

DR. SIEBER: Okay. Well, I'm interested in that, so I'll be eager to listen tomorrow.

MR. STAUDENMEIER: Specifically for reflood, Joe Kelly will be going over reflood modeling and show you what's important and show you plots of important parameters.

CHAIRMAN WILLIS: The code is keeping track of uncertainties. It'll give you a best estimate prediction and uncertainties and so. And you can see if the data are within these uncertainties.

MR. STAUDENMEIER: Yes. We don't keep track of uncertainties now. But that's something --

DR. SIEBER: Well, do that. Then you might know what to make of the fact of the scatter. That's the ultimate panacea, because once you know the uncertainties, you can make a judgment as to what the accuracy really is.

MR. STAUDENMEIER: Yes, that's part of our future goals is to --

DR. SIEBER: That's a keep.

MR. STAUDENMEIER: Yes. Is to implement some measure of uncertainty into the code so that you'll get a measure of uncertainty out with your calculations.

Documentation, they're all in draft form. Users guide input format is up to date. We need to change that to recognize that SNAP exists and will become the user interface essentially.

Need work on modeling guidelines.

Theory manual, mostly update. We will be revising it as correlations packages get revised.

Assessment. It's out of date. It'll be revised when new assessment is completed.

Programmers manual.

CHAIRMAN WILLIS: Well, I guess we're going to talk about this, but in the theory from what we've seen is that people take something like momentum equation, and declare this is it. And sometimes it's a justification which is a little hooky. But they declare it. And then they go and work tremendously on the details of the coefficients in the correlations. But the modeling of this conservation of something also has more idealizations. And there's no attempt to parameterize that or correct it, or anything at all. That may be one of the major sources of error the way you've reduced maybe a stokes equation to something which describes a box, which it doesn't look like the real geometry and all that. There's a lot of assumptions in there, a lot of errors introduced by doing that. They're not quantified in anyway at that stage.

And then there's a lot of stuff on how accurate is some correlation for heat transfer, which is little detail really compared with this big simplification of some conservation law.

So really, there ought to be some attempts--

MR. STAUDENMEIER: Yes. We don't address that in detail. I guess we always thought it has been taken care of in our assessment calculations against integral experiments that have these errors in that they show good results.

CHAIRMAN WILLIS: If you make it, say, a one dimensional analysis of something which is multidimension, is that you're introducing errors which may be a factor of two right there.

MR. STAUDENMEIER: Yes.

CHAIRMAN WILLIS: And if that's inherent in the structure of the code, you're not going to catch it by tweaking all these correlations.

MR. STAUDENMEIER: Yes, that's correct. And I guess one thing is to use the correct tool for the problem. And if you need real multidimensional type of effects, you're not going to get it in these type of codes.

CHAIRMAN WILLIS: You need to put in something else.

MR. STAUDENMEIER: You need to go to something like CFD.

CHAIRMAN WILLIS: Plus factor or something which is correlatable?

MR. STAUDENMEIER: Possibly, yes. Well, that's one thing that we'll look at in our theory guide. It doesn't really talk about those simplifications and approximations, but including documentation on that. Because I know we'll be pushed on it from here. But, yes, I think it's a good idea to have that even without being pushed on.

CHAIRMAN WILLIS: Well my old approach used to be to say I start the documentation and if the first ten pages look hooky, I don't need to read the rest of the document at all.

MR. STAUDENMEIER: Actually, we're also addressing another one of your concerns, which is to get rid of TRAC-Es in the theory manual instead of what the variable name in the code is for interfacial drag or heat transfer coefficient to say this is the interfacial drag coefficient defined like this.

Okay. Deficiencies. Documentation is not yet complete is one major deficiency. Some things that have been identified through calculations that have been --

CHAIRMAN WILLIS: Is that not yet complete, because there's something inadequate about parts of it or is it not complete because it's missing commas and punctuation marks, or that type?

MR. STAUDENMEIER: A lot of it is it just hasn't been detailed review. Like the users guide is mostly accurate. It's missing user guidelines.

CHAIRMAN WILLIS: But you know what's in TRAC. And the original TRAC is developed in the old days. I mean, when it first was developed I understand documentation was so bad that no one who wasn't a developer of it knew what really happened. And it was major effort to write down clearly what is actually in the code and why.

MR. STAUDENMEIER: Right.

CHAIRMAN WILLIS: But you're not at that stage. You really know what's in the code and why?

MR. STAUDENMEIER: Yes. My understanding

is the code was revised or documentation was revised greatly after CSAU offered to document this --

CHAIRMAN WILLIS: Well, after they found what the code really was.

MR. STAUDENMEIER: Yes.

CHAIRMAN WILLIS: Yes.

MR. STAUDENMEIER: And that's the current state of the documentation. I mean we've changed formats some and --

CHAIRMAN WILLIS: But now it makes sense.

MR. STAUDENMEIER: We make error corrections.

I think it makes sense. I mean, there's still some TRAC-Es that has to be removed, but that will be upgraded as we move on. I mean, the documentation --

CHAIRMAN WILLIS: But the outsider can look at it and say this makes sense to me?

MR. STAUDENMEIER: I think so, for the most areas. I mean, there's some areas where it probably needs better explanations. But I think for the most part it's understandable to at least someone that's working in the field.

DR. RANDOM: I've read a little bit of the theory manual, and I say that latest version that came out in 2002 I think is a great improvement over past documentation. It's very readable. I haven't got far enough to dig into a lot of the appendices to know how complete the explanations are.

And a lot of it is very up front, as a matter of fact, in terms of what the effect of the numerics are on stabilization and that type of thing, and what it's done, which often times has sort of slipped under the rug in the past. And I think that's a step in the right direction, too.

MR. STAUDENMEIER: Yes. And I think it also tries to show basic correlations in the code and where they're bad or not doing very well.

DR. RANDOM: Yes.

MR. STAUDENMEIER: So I think that documentation, even though it's draft, it's fairly complete and it's readable. But we need to make improvements.

I mean, the biggest thing -- most out of date document now is probably the programmers manual, because that was done a few years ago and there's been substantial changes to that. But we see that as the last priority because the developers have been working and are familiar with it. But then to turn it over to someone else that hasn't seen the code, that's where we really need the programmers manual so someone, a new person that gets added to the project can pick this up. He needs to change part of the code, he can look in the programmers manual, see what needs to be changed, what you're not supposed to touch.

Like I know a big complain of John Mahaffy's is the code's architecture has improve so much, it's too easy to find out what's going on and modify. And people keep breaking his parallel processing part of the code because of that, because they're not leaving stuff alone that they should. And the coding is very accessible now and it's easy.

DR. RANDOM: You're talking about members of the team doing this?

MR. STAUDENMEIER: Yes.

CHAIRMAN WILLIS: Now, we've had presentations of the momentum equation by other code developers which would get a C- or a D as a grade. Let's hope it doesn't happen with yours.

MR. STAUDENMEIER: Well, we're not going to give -- I mean, we're going to cover code equations today. I think a detailed presentation on momentum equation -- no, we've heard concerns about the momentum equation and we will be documenting --

CHAIRMAN WILLIS: Are you doing the same

thing as they did?

MR. STAUDENMEIER: We will be documenting it and making sure it's well documented.

CHAIRMAN WILLIS: At least it doesn't show the straight pipe as a pump or something like that, or a bend as a pump?

MR. STAUDENMEIER: No, it doesn't show a bend as a pump.

CHAIRMAN WILLIS: So it's not as bad as that.

DR. SIEBER: Just done once.

DR. FORD: When will these deficiencies be resolved?

MR. STAUDENMEIER: All of them? I don't know. I mean, we'll work on them as we can and prioritize them. But giving a schedule on resolving all these issues is just a wild guess based on nothing -- wild guesses as surrogates for facts, I don't know.

CHAIRMAN WILLIS: Arbitrary judgment.

DR. FORD: A deficiency to me means that you buyer beware. In other words, it's not correct in certain aspects. And yet you're using it, you say, for AP1000. What's the risk associated with using it with these current deficiencies?

MR. STAUDENMEIER: Well, I guess the key

thing is understanding what the deficiencies are and if they're important in your calculation. AP1000 has some robustness problems that I'm looking at that I think are going to be traced to condensation and interfacial heat transfer. I'm not sure yet, but that's my feeling for it.

So, up to the point where it has the problems, I think the code is doing okay. And the calculation beyond that point, who knows. But I thin that that will be fixed and we'll be doing some assessment against test data for that, too.

Essentially it's understanding the limitations and what it's okay to use the code for right now and what it's not okay to use the code for.

DR. SIEBER: These passive plants that are the driving forces for various denominators are so small that accuracy is critical, in my view. If you have a safety system that's a pump system, you know, the forces are very large there and you know that when you turn the pump on, the water's going to flow provided there's water there to flow.

And so I think that as you go through this process, you really have to pay close attention to the accuracy that you're getting. Because it could mean the difference between an interfering phenomena existing and not existing.

MR. STAUDENMEIER: Yes, I agree with that. That's one reason why we have passive and integral test data.

DR. SIEBER: Yes. Right.

MR. STAUDENMEIER: And we're going to do detailed comparisons against that before declaring the code is good.

DR. SIEBER: Right.

MR. STAUDENMEIER: For ESBWR, NRR's using it for some calculations. They've had some success with at least getting to the point of minimum vessel inventory and beyond that. We haven't really moved into the long term phase of the accident.

DR. SIEBER: Okay.

MR. STAUDENMEIER: It could be talked about in a future meeting on ESBWR.

MR. BAJOREK: It's also what's driving the order in which many of these are getting resolved.

At this point RELAP is being used to base the regulatory decisions for AP1000. We've done a fair number of AP1000 calculations with TRAC and they actually compare quite favorably with the RELAP calculations.

DR. SIEBER: Okay.

MR. BAJOREK: Which is a bit of a assessment in and of itself.

DR. SIEBER: Yes. A secondary order assessment.

MR. BAJOREK: Yes. Yes. You get into the question of which code you start to believe at that point.

DR. SIEBER: Yes.

MR. BAJOREK: But we don't have the time and the schedule to do all of the assessments that we would want to do with TRACE to quantify its accuracy for AP1000. That's why NRR has gone with RELAP for its work with AP1000.

ESBWR, however, we're going to be using TRACE version coupled with CONTAIN in order to get better dynamics between what goes on in containment. Because of that, most of the near term work is going to focus on getting the condensation models correct. Because those are going to be the most important in that analysis.

DR. SIEBER: Right.

MR. BAJOREK: Things like the horizontal flow maps, the interfacial drag in the core, which would be very important for AP1000, can wait a bit. Okay. Just because they're at a lower priority because it's covered elsewhere.

DR. SIEBER: Yes.

DR. FORD: So on slide 11 where you say that TRACE is being currently or planned to be used for ESBWR AP 1000 and ACR-700, what you're really saying is that you're already using RELAP. And there's a difference between TRACE and RELAP, you use RELAP, is that right?

MR. BAJOREK: That's right. And right now we would say that RELAP has the better assessment base for what's going on in AP1000. The results from AP1000 TRACE calculations are useful. They help us with assessments. In some cases we use them to help evaluate the plant, but we wouldn't have to depend on those because we have another avenue for evaluation.

DR. FORD: Right.

MR. STAUDENMEIER: Yes. Because like for AP-600, I think probably on the order of 50 to 100 man-years of work was put into making RELAP work for AP-600 if you count in all the code assessment and fixes and analysis was done for that. It was an astounding amount of people working on it.

So to reproduce that in a short period of time with any code would be impossible. But, I mean, we're going to look at assessment against some of those same things. We're not going to do -- we're going to have to spend that total amount, probably, on assessing a new code just because of all the things we learned during the past project, we'd be smarter about doing some things. We were just feeling our way along back then. But we'll be assessing against key test data that was done part of that program.

CHAIRMAN WILLIS: When you say this deficiency, I wonder what the measure is of that? I mean, you say the horizontal two face flow models are deficient. Does that mean that at the separate effects level when you compare with a simple test, you get the wrong answer or does that mean something else?

MR. STAUDENMEIER: Just that there was never a lot of effort put into the TRAC horizontal flow model, mainly because of its past emphasis on large break LOCA calculations.

CHAIRMAN WILLIS: So how do you measure it to the --

MR. STAUDENMEIER: For small breaks, I mean horizontal flow regime map become more important for small breaks and there wa never a lot of emphasis put on that.

CHAIRMAN WILLIS: Yes. Let's just take it take horizontal two phase flow models. I mean, I can take the models in TRAC.

MR. STAUDENMEIER: Right.

CHAIRMAN WILLIS: And say I'll now use them to predict what happens in oil pipeline. And off from the pressure drop by a factor of five or something. That doesn't sound very good for oil pipelines. And yet it may be that when you use it a nuclear reactor it doesn't matter.

MR. STAUDENMEIER: And it could be. But currently we haven't even assessed the horizontal two phase flow map yet.

CHAIRMAN WILLIS: But the old excuse for not improving these has been oh it has no effect on PCT or something, even though it's got the physics all wrong and so on, and it doesn't work for all pipelines and other applications. It doesn't have any effect on PCT, so we don't worry about it. But the danger there is you say okay now we've got the code, we'll use it to predict something else like water inventory in the passive system and it may turn out it has a tremendous effect on that.

MR. STAUDENMEIER: That's right.

CHAIRMAN WILLIS: But these errors have been accepted over the years because they didn't seem to effect some measure of success.

MR. STAUDENMEIER: That's right.

CHAIRMAN WILLIS: It's now no longer the measure of success.

MR. STAUDENMEIER: And we know --

CHAIRMAN WILLIS: We have to be careful, I think, about what you mean by --

MR. STAUDENMEIER: We know applications where it is important and RELAP has better horizontal flow mapping than TRACE does.

CHAIRMAN WILLIS: Depending on what measure?

MR. STAUDENMEIER: On what measure? Comparison to experimental data against phenomena.

DR. RANDOM: Is that part of that the out take models, for example you know in horizontal stratified flow if you had the break on the top of the pipe, the side or the bottom, well you got different flow splits. I don't know whether those models are in TRAC or not.

MR. STAUDENMEIER: Yes, they have the same model that RELAP has.

DR. RANDOM: Well, they do in the current one, I know I read.

MR. STAUDENMEIER: Yes. The current. It's been in there for at least the last ten years, I know.

DR. RANDOM: What is it? More the transition from stratified flow to slug flow, that type of thing and it's not comparable or --

MR. STAUDENMEIER: Yes. Looking at hot leg like when you have voided water or water in a hot leg and looking at entrainment up in the steam generators or things like that. I think that's the cases where in small break LOCAs and you get into like condensational reflux modes that TRACE or TRAC really hasn't been assessed against experimental data. But I mean these are the type of things we're going to look through our correlations package, identify places where it's deficient and replace the models or improve the models that are there. And it's just going to be a process, a structure process of going through reviewing them, identifying deficiencies and replacing them.

Right now Joe Kelly's doing that with the rethread model. He'll be doing it with some condensation he's working. And we'll look at other correlation packages in the future.

The other deficiencies --

CHAIRMAN WILLIS: Well you always get the question, though, from the engineering types who say "Well, you're just being academic. You're looking for some perfect correlation or perfect." Well, that's not what we're doing. It doesn't matter from our engineering purposes. Something crude is okay. But you have to be able to answer that, because they have some truth to what they say sometimes.

MR. STAUDENMEIER: Yes, there is. I mean, you don't want to spend time -- I mean we have limited developer time and we don't want to spend that developer time defining unimportant details. We want to spend that developer time improving the places where the code needs the most improvement.

CHAIRMAN WILLIS: You have to have a code which lets the user detect this. I mean, when it's being used for a new application, there's got to some way that the user can run the code quickly enough and come in say ah-ha, for this particular use, this particular part of the code is giving me trouble, and can recognize that. Otherwise if it's just used blindly, it might have deficiencies which are not even seen.

MR. STAUDENMEIER: That's true. I mean, to do that you have to understand the phenomena it's going to happen in your accident and recognize whether the code has models in to cover that or not, or maybe they weren't meant to cover that, but it still does okay.

CHAIRMAN WILLIS: Well, you have to have that sort of curiosity and you have to have a code which lets you pursue your curiosity.

MR. STAUDENMEIER: Yes.

MR. CARUSO: Does the code -- are you including, I want to say intelligence in the code that alerts the user if he becomes too creative in setting up his models with models like horizontal heat transfer? I mean, that tells him well, you are setting up a model that's outside the range of applicability or you're getting into areas where this is uncertain or there are problems?

MR. STAUDENMEIER: Well, the theory manuals ideally will document that and show you the range of where the correlations have been assessed over.

SNAP in terms of modeling, plant modeling was working on putting the expert wizards type of things in there to help you set up a plant model and nodalization that you need for what type of calculation you're doing and things like that. So we are trying to build in -- give the user the information he needs in order to help him along with this type of calculation he's doing.

I guess if I go into every one of the deficiencies, I think we'll be over schedule here.

CHAIRMAN WILLIS: So, I understand.

MR. STAUDENMEIER: If you have any particular ones we could --

CHAIRMAN WILLIS: Maybe later on some of you can do a presentation on one or two deficiencies, so we can at least see the approach.

MR. STAUDENMEIER: Okay.

CHAIRMAN WILLIS: Isn't that true, there's going to be a presentation somewhere on --

MR. STAUDENMEIER: Well, reflood heat transfer, you'll see past deficiencies in the code --

CHAIRMAN WILLIS: And the condensation model and things --

MR. STAUDENMEIER: Right.

CHAIRMAN WILLIS: Sure.

DR. SIEBER: Everything is tomorrow afternoon.

MR. STAUDENMEIER: Yes. People following me will answer all of your questions. You'll need to ask them.

Future development work. Right now there's development in progress adding a droplet field that eventually could be used in entrainment models or better reflood models. But the structure is going in the code to add a droplet field to the code.

DR. RANDOM: At another velocity field than you've put in?

MR. STAUDENMEIER: Right. It'll be a liquid continuous phase and a droplet phase, so there'll be, yes, separate equations tracking the droplets from the film.

ATLATS offtake model, coding has been completed. One of our interns working under Steve Bajorek that came from OSU where ATLATS facility is, implemented a correlation for the ATLATS offtake into the code. It's not in the mainstream version yet. We still have to put it through some more testing and assessment.

DR. RANDOM: We ever resolve what the offtake behavior was due to? You know, there was some question whether there was slugs going back and forth or whether it was true entrainment.

MR. STAUDENMEIER: I don't know. I'm not familiar with the experimental program. But I'll have to become familiar with it --

DR. RANDOM: I hate to have a correlation for a phenomena that's not there, incorporate in the--

MR. STAUDENMEIER: It'll be something that the user has to fit the ATLATS offtake model on, so it's not something that's going to be on by default. So it's something that you'll have to make a conscious decision to set it on and, presumably if you do make this conscious decision, you know something about the model.

DR. RANDOM: Or willing to go back again, like you mentioned, it's wise to try to avoid those kind of things if you can.

MR. STAUDENMEIER: Yes.

DR. RANDOM: Because that allows the user then the option of --

MR. STAUDENMEIER: Yes. Eventually we'd like to get rid of these type of options and have the code decide automatically, but it's not at that state now.

Steve, do you want to say something about it?

MR. BAJOREK: I think from the tests and the visualization it was clear that a lot of the entrainment was coming from transition to a slugging and oscillatory behavior in the region between the offtake and the steam generate inlet plenum. I guess we would sort of look at that as some type of a change in the flow pattern that's going on in that region of the hot leg as opposed to a change in the kind of entrainments that I think is typically thought of like in the Schrock type correlations, droplet shear off of a smooth stratified or wavy interface.

While we have the coding set up so that you could use the correlation that was developed from ATLATS, we think that in the long run proper implementation of that is going to mean looking at that in conjunction with the horizontal flow pattern matters to make sure it's implemented only for the right conditions as opposed to just blindly turning it on or off for a particular transient.

CHAIRMAN WILLIS: Right. Joe, you going to go a few minutes more?

MR. STAUDENMEIER: Yes.

CHAIRMAN WILLIS: I think we can probably read the slides.

MR. STAUDENMEIER: Okay.

DR. SIEBER: There's only one left.

DR. FORD: The last one bullet. Has anyone done an analysis of what the net value of doing this is? For instance, the cost of development must be considerable versus the benefit? Has anyone sat down and done what's the net value of doing this?

MR. STAUDENMEIER: To having an

independent assessment capability for reactor regulation, is that what you're --

DR. FORD: No. The coupling of TRACE to the other NRC codes?

MR. STAUDENMEIER: Oh, coupling of TRACE to other NRC codes? I mean, some calculations I mean you just get to the point -- I mean it speeds user productivity that he doesn't have to transfer data by hand. But there's some calculations where it goes beyond just being a convenience to becoming necessary where you get tight feedback between the different things that you really need to solve a couple of calculation for it. So, for that I mean the benefit is you're actually able to solve the true physical problem.

And the cost you'd have to look at, I guess, is what your uncertainty would be in solving it in a detached manner and making approximations versus solving it in a true coupled manner. But the cost isn't actually going to be very much to couple like a fuel rod code to TRACE. Now with our new ECI capability. Like CONTAIN was coupled to TRACE in a matter of a couple of months, essentially. And essentially no modifications to TRACE except add some additional variables that CONTAIN wanted to see. But essentially it was dropping a few subroutines into CONTAIN and then it was magically able to talk to TRACE and they're able to communicate and run together. So in terms of code development, it was a very small cost.

I mean putting the ECI infrastructure into TRACE to being with was a fairly large investment, but that was known up front and that was one of the design requirements is we wanted to be able to communicate with other codes to do these sort of coupled calculations. And add couple capability fairly quickly then have to integrate this whole FRAP code into TRACE, we can have this communications interface present in a quicker manner.

DR. SIEBER: I understand the niceness, if you like, for the requirement to combine models for two different physical entities where physically there's a joint. But if that's to be done at the expense of increasing the runtime by a factor of ten because of complexity, you know, you have to come into some value judgment. Has that been known?

MR. STAUDENMEIER: Well, I mean, I don't think it's going to increase runtime by a factor of ten. But, yes, you're right it does have to have some sort of judgment on that. But mainly we're being driven by user requirements. Everything users ask us to do, I mean if it's totally crazy and think, oh, it's going to take a million dollars to do this and you're only going to use it for one calculation, that doesn't matter, then we would recommend not doing it. But for something that's of general use and interest to a wide number of people, then we make the investment and do it to gave capability to our user community that they need for doing some sort of analysis.

CHAIRMAN WILLIS: I'd like to take a break at 10:30. But if there's some high level questions. We're going to get into lots of detail later.

Do you have a high level question?

The runtime for your presentation was two hours, and that is what we have achieved.

MR. CARUSO: I have one high level question.

CHAIRMAN WILLIS: A final question.

MR. CARUSO: Is any of the TRACE development work aimed at gas reactors or nonwater cooled reactors?

MR. STAUDENMEIER: Well, there was work in TRACE in the gas reactors. It was thought they were adding modifying their 3-D kinetics to be able to model the gas reactor geometry. So it was more PARCS work. There would have been some models that would have had to be added to TRACE for like a pebble bed core heat transfer thing and maybe some pressure drop correlations through a medica like that.

That work was stopped. Yes, I don't know. Someone I think probably -- I think it was decided maybe that MELCOR would do most of the calculations.

MR. BAJOREK: Yes. Joe?

Yes. We looked at the situation and since there wasn't anything driving HTGR to the gas reactors in 2003, we decided to stop the work on TRACE and made the decision that in the future because those reactors are actually going to need a new regulatory framework, a better avenue would be to use MELCOR because that has the various tracking for fission products, fission product release. It can handle single phase and it looked as though it's going to be a lot easier to teach MELCOR to do the heat transfer in a pebble bed or a single phase prismatic core than it would be to try to get TRACE to do fission product release and track multiple constituents through the containment.

CHAIRMAN WILLIS: Okay. Can we --

MR. STAUDENMEIER: We have packages added to do liquid metal heat transfer. Some people have used it for that. Out at Los Alamos they're using for that for like accelerator production.

Okay. We're using experimental data --

CHAIRMAN WILLIS: We've read all that already.

MR. STAUDENMEIER: Okay. Summary of upcoming presentation.

CHAIRMAN WILLIS: Yes, we know that.

MR. STAUDENMEIER: You'll see the presentations.

CHAIRMAN WILLIS: Yes, we know that. We'll see that. Take a break until 10:45.

Thank you, Joe, for being very communicative and robust.

(Whereupon, at 10:35 a.m. a recess until 10:48).

CHAIRMAN WILLIS: Okay. Let's get back into session. We're going to hear from Professor John Mahaffy from Penn State.

DR. MAHAFFY: Well, good morning. I was asked to present some material that's going to be review for at least two of you that I recognize here, just to bring some other Committee members up to speed.

I had provided a large amount of material.

I would say most of it you'll see me flash by the viewgraphs. Jump up and yell if you want to ask a question on it, but I don't want to cover all of this unless there's something key that you want to go through.

I'm going to talk, first of all, in an overview of the architecture of the code. But let me come back to somewhere that Joe began.

The architecture of code was driven by three high level requirements that were sent before any of us laid down any work on this thing that we now call TRACE. The coding had to be readable and understandable; the coding had to be maintainable; it had to be extensible.

And let me remind you of something based on some of the questions.

There was an oh, by the way, when you're done we want it to run faster than the previous applications.

But these are the primary driving items, and it's difficult to do check boxes on these things, but we try to do it for you.

Some of the things that help you out, we've heard discussion from Joe. I'll talk a little about the evidences in the recent development. Now you have people going in doing things that those of us who have been developing code for 30 years have some idea of what it took to do it in the old days and what's happening now. And these things are working a lot better in terms of development efforts. Bringing in engineers, getting specific models implemented, the engineers go away and do something else. And they're not necessarily part of the core team of developers.

Computer science modular and object oriented. The curious thing here, and I didn't really think about this until rather recently, a fellow named Bill Reed, some of you may know from a long time in Las Alamos, he actually used those words when he was pitching what became the TRAC code. And there is enough of that that survived in TRAC, I think drove the decision as that's the point where we're going to start an evolutionary approach that Joe talked about. It was another design decision. Do we write from scratch or do we evolve?

I think history has shown that the evolutionary approach has paid off for us. We have something that is not recognizable as TRAC anymore.

DR. RANDOM: One comment on that, though.

DR. MAHAFFY: Yes.

DR. RANDOM: It seemed like the read made it modular by component, meaning steam generator, pressurizer, this type of thing and it specialized the modeling to those components.

DR. MAHAFFY: Yes.

DR. RANDOM: That turned out to be a mistake because of its inflexibility and later was modified.

And I don't know, are you still at that point in TRACE where it's modular by component.

DR. MAHAFFY: What you see, there are different levels of modularity in TRACE. The modularity by component, and let's just use this as an opportunity to move forward. Let's see if there's anything here.

Yes, let me answer that question a minute down the line, can I do that for you?

DR. RANDOM: All right. Sure.

DR. MAHAFFY: When we get to the right viewgraphs.

General characteristics of code. The one thing I want to talk to you about is that as time goes on here, people are not going to be thinking in a public level in terms of this acronym TRACE. It was already said before; it becomes a module contained in the symbolic nuclear analysis package, all right? And this will take a minute. Here's our little sketch of what's going on in this package.

We've got a bunch of programs that can talk to each other. SNAP is really the overall driver. And I've SNAP a lot. I used it as a basis of a course, and it is a very powerful tool. Even now when they talk about bugs, yes, there are still bugs in there. But even with the bugs in that for somebody whose been developing input models for a long time, this is incredibly powerful. You know, I can do a whole more with SNAP until I hit some bug that limits me, and I have to go off to a text editor, than I could ever do before.

But you look at it, there's a computational engine. This actually has this picture we've shown the input processing that used to be done for TRAC pulled off. And at one point we had these things physically separated for a brief moment in history. They're actually tied more tightly together at the moment than they ought to be. But if you look at the flow of what's going on, the old TRAC series stuff comes in through an ASCII processor and it moves into the computational engine. RELAP5 is coming in through SNAP, that's what ought to be happening anyway. There some information from RELAP5 that moves into something that looks like this to get converted a little more, as was discussed. That's a history in available developers. And then this computational engine can operate in parallel. We'll talk about that later with other support applications. 3-D neutron kinetics is one of them.

But in terms of the future, all this stuff; these blocks through here, this is just so that we don't waste -- you know, Lord knows how many hundreds of thousands of man-years of effort in these archival input models of plants and experiments. But when you're doing something new, and this stuff goes away, you're coming straight into the ModelEditor interface that you'll see later and it's talking in a little different way with its own input checking and whatnot, with the computational engineer. All right.

DR. RANDOM: Out of curiosity, what kind of file exists after you get through going to SNAP in terms of an input file? I assume you can give it a name and call it --

DR. MAHAFFY: It's called a TPR, it's a platform independent binary file. Are you going to talk about that, Ken?

MR. JONES: Yes.

DR. MAHAFFY: So let Ken take care of that file for you.

DR. RANDOM: It's nothing that they use or would look at directly, though?

DR. MAHAFFY: No, it wouldn't. And, you know, if the user gets nervous this thing can spit out ASCII input decks if you want to go look at them. There are no plans in any foreseeable future to destroy ASCII input decks. But us old guys that get nervous about this stuff have all retired and have gone onto greener pastures somewhere. And the young kids that don't care and have more faith in this visual stuff, move on. And then you'll see it go away. But, yes, it's here.

DR. RANDOM: It seems like both a benefit and a danger, though?

DR. MAHAFFY: Yes.

The interesting thing, listen to Ken talk about some of the checking. There's some interesting stuff that goes on that builds up a heavy -- there's a lot of things you can do in terms of feedback that gives you a high degree of confidence of what's into SNAP and gotten past to the code and back again. There's a lot of power in cross checking to make sure that what the code is seeing in its internal data structure is what you meant it to see in terms of the geometric representation. But I'm not going to talk about that.

Now, your question. Here is part of modularity. You still see loops driving certain generations that go through pipe components, channel components, pump components. And under that you might be doing special things in terms of your heat transfer correlations for a pipe wall versus a channel. And so there is this modularity by component that allows you within physical regions to do different things as appropriate in terms of your correlated information. All right. So that's there.

DR. RANDOM: You say loop driving? You mean--

DR. MAHAFFY: In terms of the code, okay? Right up here there is -- in Fortranland it's a DO loop. All right. There's a DO loop that runs over all components in the system.

DR. RANDOM: And then it'll call and sequence?

DR. MAHAFFY: Yes. And each component it'll work it's way down and it'll be able to access. From a practical standpoint, a lot of this stuff is the same. Okay. The fluid physical properties are -- well, we're not going to talk about that today. I mean you can have isolated loops in a system where one loop is a liquid sodium loop and one loop is a water loop. So, you know, these components may in fact be using different fluid properties, depending on where they are.

DR. RANDOM: Well, I assume there are more components like trols, trips?

DR. MAHAFFY: Yes.

DR. RANDOM: Neutronics?

DR. MAHAFFY: Trols and trips. The neutronics really is that block called parts. We're really not doing --

DR. RANDOM: -- higher yet?

DR. MAHAFFY: Yes, let's go back up. It doesn't want to back for me. There we go.

Neutronics really is isolated off in another computational package. There are point kinetics calculations that can still be done, but we're pretty much steering away from those. It was important historically, but most of the interesting calculations now you want to do a better job in your neutron kinetics. And that gets passed and you have to worry about where this occurs in your computational flow. I'll talk about that a little bit later on.

So that's the kind of modularity by

component.

There's also a modularity by computational mesh. And coming from RELAP5 world, you probably recognize that sort of thing more than the component modularity. So that there are a lot of calculations going on at a low level on the hydrodynamics that you're seeing.

Okay. Here are the things we operate on for (1) the hydrodynamic solution. Here are the things we operate on for 3-D hydrodynamic solution. Here are the things that operate in the conduction space for the walls. So there's another class of modularity there.

Data structure. A lot of the data still gets stored by components and we're using something called a derived type. And this probably doesn't mean anything to you unless you're a Fortran 90 programmer. It looks like a structure in C. But for each component it has a component array that marches to the components and then within the components you specify properties.

There are also global arrays that are accessed for the solution that look more like your mesh dependence. But I'm not going to go --

DR. SIEBER: Are these the indices that

you're showing, you know, of the schematics --

DR. MAHAFFY: I've got an index here over components.

DR. SIEBER: Yes.

DR. MAHAFFY: And here's the index over volumes within the component. Now, we have -- again, when you get to the global array and you see more of this with some recent stuff that's not in the official code. When you get out to the global solution, you are seeing data structures that also look more like what you're used to in RELAP5 where you stream through every variable in the whole system.

DR. SIEBER: Like for one point.

DR. MAHAFFY: Yes.

DR. SIEBER: But here you're saying through all the pressure for all the cells in that particular --

DR. MAHAFFY: Yes. Basically you've got an array of pressures, all cells in a given component are grouped together. That sort of modularity occurs in the component data structure.

I'm not going to go into this unless you guys have some computer science questions. You can look at this. This tells you how data gets passed around.

Here's an important feature that's new and it has to do with the language, and we use it very strongly. We have this thing called name based pointer assignment. Pointer is an object in any kind of advanced language.

What we have, and we build it into the data structure of TRACE, any kind of variable that's accessible within the code, we have subroutines that when passed an ASCII name that's unique to a given variable, they will return a programming pointer object that points to the location in memory where that variable is available. So actually I lied. You got to give a name, a component, a cell or something like that to get at it.

But we now have a capability to do a lot different. What was the old computational science term? You know, it's data driven computing is what it is. So it can do some interesting things there.

DR. SIEBER: Well are you saying by this, that if you want to couple another code to it, that you can link it by giving it the --

DR. MAHAFFY: We'll see this with the ECI. This started with the ECI. I can send an ASCII request into TRACE that says give me void fraction, and I give it a name. I call it ALP, but you can change your dictionaries.

Here's the component where it is. Here's the cell where it is. Set up a link so I can pump this out at certain time intervals. And it's very cognizant of this sort of stuff. But we also do it internally.

For instance, when I need to set up by boundary conditions in a given component space so it knows what's going on next door, we no longer have hard wire code that really is saying, all right, you got to move this, this, this and this. What we have is very generic code that's pumping information based on an table of pointers, okay. And those pointers are assigned by lists in a module in our code. So that if I need to add a new variable that needs to be moved between two components or various other things, I just add it to one or more of these lists that tell me what needs to be moved and when. It makes things a whole lot similar in terms of programmers coming into this code and doing something useful.

And here's a sample if you want to get into it. I'm going to skip it. Of how we set up some of these pointer assignment routines based on ASCII names.

Here's how we move data, and really the

bottom line, this set of lines is important. Once we've set up all these pointers to tell you where the data's coming from and where the data's got to go, that's all done at initialization before the computation gets running away, moving data is a really tight small programming structure. It is very, very fast. It's not one of our timing problems in this code.

Other features. Okay. Everything in here is allocatable. There are no fixed chunks of memory in here. We assign memory on fly based upon the structure of the input.

Another thing that was important. When we moved from the old Fortran data structures, which are these ugly container arrays with variables that tell you where things are to the new data structure, this was all done in a scripting constructs. And it was done in a way that the data structure now has very unique patterns and names in it so that if we come down the line five years from now with enough morphs in the language or better wisdom about how our data should be laid out to take advantage of speeds of new generations of compilers, it's not that much work to go in there and do really massive surgery on this data structure.

Now, you want everything to go from a component orientated data structure to a purely RELAPish data structure that you know and love from the old days, in fact I've done that in a half a day with some experiments, just running some strips. It's quick.

We've got a rich set of information. Pardon the TRACEese here. I've given you some names of some arrays if some of you ever want to dig into it. But there are arrays of information that tell you in all kinds of flexible ways whose talking to who and how.

MR. CARUSO: Did you discover very many bugs in the old TRAC container arrays when you took them apart?

DR. MAHAFFY: Yes. And this is something I want to talk to you about. We'll do it now.

This whole business of evolution versus revolution. Okay. When we started with the hold TRAC code and we went through -- we do things in two ways. We try to isolate this. There are steps when you're moving and advancing the code, and there are less of these happening now than there were over the last six years -- there are steps where you know in principle that what you have done to that code should change not a single bit in the answer. These are what we call null results. And as a programmer I know that I'm about to engage null results. I've rewritten the numerical schemes in large ways in various places, and I know null results or I know that it's a round off area results and I can put metrics on bounding on that, too.

And I go in and we run the previous version of the code. We run the new version off of a basis set of assessment that Chris Murray has pulled together over the years. Every time I create an update to the code, I'm running over a thousand test problems to prove to myself that I have not undermined some capability that's already in that code. Okay. And I'll run that set of a thousand test problems. And if I get non-null answers when I believe I shouldn't, I'm going in there and I'm finding out why. And the why is often bugs that have been in there. Within the last six months, I've found two bugs that I know have been in there for 25 years.

This evolutionary approach has done a remarkable job in terms of cleaning out bugs that had been inherited from the predecessor code. And the way in which we're doing our testing, I can tell you that I've put a -- you know, I'm not going to claim I've put zero bugs in that code over the last six years. I undoubtedly have. It's the nature of programming. But I will tell you that I've put by two orders of magnitude less bugs into that code than over a comparable period of six years 25 years ago because of these testing procedures.

DR. RANDOM: John, one quick question. This new structure and all, is this your brain child or did you work with other people or did it --

DR. MAHAFFY: No. This was a collaborative effort and there's dangers with design by committee, obviously. Yes, I contributed significant amounts, other people did significant things. Skip Dearing did wonderful things for us during this period of time.

DR. RANDOM: Now he's at Las Alamos.

DR. MAHAFFY: He's a Las Alamos. He's not even involved in reactor safety anymore. He's moved on to other things.

DR. RANDOM: Any others that --

DR. MAHAFFY: Yes. I mean, Skip Dearing was involved. In terms of actual programmers, Susan Woodruff was involved. Paul Giguere. You can look at the whole set of code developers. Again, we have these coordination meetings and we've had them since the beginning. And people sit down and they look at what's being done. And you can look at anybody who was on the list of code developers in this project over a wide range, you've seen the various organizations. If you sit down, these things get presented. People kick them around a little bit. They get changed.

So you're not going to find one person that's done it. It's been a remarkable collegial experience. People have been very friendly in the way they've dealt with these things.

DR. RANDOM: I'm just trying to get a feel for whether this was one person's brain child or whether there was some diversity.

DR. MAHAFFY: A lot of the data structure that you're seeing here if you look for a dominate person in it, it was probably Skip Dearing. He's the guy that applied it. Before there was Skip -- before we really started coding, there was a group of us. I was involved, Chris Murray was involved a little bit, I believe, and the cluster at Las Alamos where we looked at a whole range of potential data structures and we did timing studies based on existing compilers to see what was reasonable and what wasn't.

And this is one of the reasons I put this caveat. You know, if we don't like this data structure, we can change it. Because the compiler capabilities have changed. You've heard this before. The compilers as they change with time deal with different things at different levels of efficiency. And from the very beginning we wanted to be able to produce something that if we have to morph it into another form to take advantage of some new generation of compilers, we can do it and we don't have to spend six months at it. As I say I've done some major changes on this in six hours.

DR. RANDOM: Well, the underlying thing, I guess, is driven by computer science considerations and, I guess, flexibility you just mentioned.

DR. MAHAFFY: Yes.

DR. RANDOM: Were those the motives for it?

DR. MAHAFFY: Those are the motives, and again the back -- you know, readability, maintainability, extensibility.

DR. RANDOM: Yes.

DR. MAHAFFY:

You know, we've sat down there. That's Farouk talking. He's laid those. He chiseled those in stone tablets and he brought those down to us and we've kept those in our minds ever since.

DR. RANDOM: Now you say radiability; this has to do with things like indices, pointers, understand what they are?

DR. MAHAFFY: Indices, pointers. And, again, you look into that array you saw that viewgraph set. Let's go back and peer at this thing.

DR. RANDOM: Also there was some discussion of Fortran, and I would guess that one reason for sticking with Fortran is it's from an engineering point of view, quite readable.

DR. MAHAFFY: That's right. That's can write readable C. I can write -- you know, every year there's an international obfuscated C contest. All right. And believe me, you can obfuscate C a whole lot better than you can Fortran.

I'm guilty of writing some really obfuscated Fortran, as some of the people sitting in this room can attest, and I make a conscious effort to clean things up. And I think everybody has. We kick each other around if things get out of line.

But, yes, it's readable. And you had the comment about some complaints. You want to hear the dark side of things. It is so easy to find data in here that every now and again you'll get a developer whose in a hurry, decides to take a shortcut to get at some data, and they violate the rules that I've laid down in terms of data flow to maintain parallel calculation capabilities. You know, "Gee, I know where that is, I'll just grab it." Well, you can't grab at this point in the computational cycle because it's not a synchronization point.

And we have all kinds of automated services that are list driven that move the data when it needs to be moved. You want data moved from one part of the calculation to another, you adhere to certain standards of populating lists and letting our service to move data take care of it for you.

Extensible architecture. The one thing that I want to impress on you is that we have looked ahead in this process. We're not sitting here with blinders about reproduce the capabilities of predecessor codes. There are enough of us who thought about these problems over enough years that we've gone in there. And I've just produced a list of some of the high obvious items.

We want to be able to eventually to implement higher order numerical methods. The numerical methods in any of our reactor safety codes can't be published in a reputable journal anymore, right? They want at least second order accuracy.

More implicit numerical methods. I've got a number of tasks working on that. We're moving up from the old semi-implicits in the sets to fully implicit where necessary.

DR. RANDOM: When you say "higher order," I don't know if you want to talk about this now, but on time, for example, it's only first order --

DR. MAHAFFY: That's right.

DR. RANDOM: -- and space --

DR. MAHAFFY: We're pushing the space first. We'll look at the time later.

DR. RANDOM: Why is that? I'm curious because you run hundreds of thousands of time. You would think that the --

DR. MAHAFFY: Yes. You know, the reason for that is personal experience as anything. We had a major effort in the TRAC program, circa 1980, where we put a number of man-months into taking that thing up to second order accurate in time. And the amount of difference it made in runtime and accuracy was minimal.

Now, we had a much poorer assessment set in those days. Let's face it. The capabilities now for running a large set of test problems have radically improved. And we want to revisit that issue. Okay. For a much more limited -- what we were running, maybe 50 standard assessment problems that we do over and over again; for those we didn't see improvement results. And that's, you know, a personal bias and I kind of step back from that.

In terms of higher order methods, one thing here. I've got a graduate student working on that. You can, for instance, there's a method out there called Quickest. It's a Leonard method. It advertises itself as both higher order and space and time. Well, if you look at that, my student ran for me a Richardson extrapolation sensitivity study on all of that. And despite all the claims, it still boils down to a first order accurate in time and space because of little details that it forgave that Leonard doesn't mention in his papers. I expected them. But it's got a lower order that it defers to and then stay at an upwind different scheme. So it's still something useful.

DR. RANDOM: Maybe you want to talk about this later sometime, but when you have a code with discontinuities buried within it like void fractions going from zero to one.

DR. MAHAFFY: Yes.

DR. RANDOM: Why those points tend to

limit the accuracy you can achieve.

DR. MAHAFFY: That's right.

DR. RANDOM: And high order methods don't do you much good.

DR. MAHAFFY: Well, what you do is, you know, in a high order method is you have to -- a high order method by itself is pretty useless. You've got to at the very least have some kind of a flux correction method in there to capture the discontinuities. And you have to be willing to stand up in front of people and admit up front that, you know, when I get a discontinuity, folks, I'm falling back to something that's formerly first order accurate because that's the best I can do.

You know, I could wave my hands and do fancy mathematics, but it's going to boil down to first order accuracy to handle those discontinuities robustly.

Yes, Sanjoy?

MR. BANERJEE: Yes. A couple of questions.

All these higher order methods and space work if you're really dealing with PDEs and things. But when you get to the sort of momentum equations that we've seen where there are junctions and things coming into boxes, and flying out of them.

DR. MAHAFFY: Yes. And again --

MR. BANERJEE: Weird angles. How do you do higher order for those things?

DR. MAHAFFY: At some point you don't. Okay. And if and when we're ready to come and tell you folks that we believe we have a higher order method that is robust enough to survive in this kind of environment, you'll see qualifiers on it.

CHAIRMAN WILLIS: But do you ever do it for things like these boxes that Sanjoy is talking about; only one dimensionalize, idealized concepts?

DR. MAHAFFY: The really idealized stuff, okay, you're right.

CHAIRMAN WILLIS: You can't.

DR. MAHAFFY: If I've got a plenum, what I can do for you is that I can tell you that on a purely mathematical basis that each of the fluxs coming into that box I have evaluated to check into it for accuracy. But what does that mean in terms of everything getting mixed up in that box, you know?

You've lost it. Sure.

MR. BANERJEE: One of the problems that Graham had, and I had with some of the previous presentations was that even at very low order it was very hard to have a rational scheme for preserving mass momentum and energy in these boxes. Maybe mass and energy --

DR. MAHAFFY: Mass and energy, but momentum.

MR. BANERJEE: Is very hard?

DR. MAHAFFY: Yes. And you're going to see, as I go into another talk on the mesh topology, that we have not resolved that yet. And I'll talk about that later, if you'll allow me to delay that particular issue.

MR. BANERJEE: Sure.

CHAIRMAN WILLIS: I think the NRC should offer a prize to the first person who resolves that issue.

MR. BANERJEE: Well, I think it has to be multidimensional.

DR. MAHAFFY: No. I mean, the resolution--

MR. BANERJEE: Because this is a multidimensional system.

DR. MAHAFFY: Here's your answer. Okay. If you want to resolve that issue, if you want to resolver a lot of the things that you worry about and I worry about on momentum equation, and what you want to do is get the NRC to sign up to be a participant in the Neptune Project, which France kicked off a couple of years ago, and that is model the whole thing with CFD. Right. Okay.

MR. BANERJEE: These are truly multidimensional components.

DR. MAHAFFY: Yes.

MR. BANERJEE: And to try to make them work by one dimensional doesn't work.

Let me say one other thing about higher order schemes. There are schemes higher order which will handle this discontinuities, like variance of the INNO schemes where, you know, you sense the discontinuity and you take -- you never take a difference across it.

DR. MAHAFFY: That's right.

MR. BANERJEE: So that's not the problem.

DR. MAHAFFY: It's another flavor of mesh matching. People give these things different names and they wave their hands a bit. But these techniques are old and they just keep getting resurfaced in a number of different ways.

MR. BANERJEE: Right. But the real problem is the physics of how to do this across complex geometries which are inherently multidimension.

DR. MAHAFFY: And the answer is you're doing some kind of CFD. That's what it does.

The other thing we can do for you --

CHAIRMAN WILLIS: And you can beat it to death with numerical methods, but if you got some fundamental gross approximation in the physics, you're not going to get any better accuracy.

DR. MAHAFFY: Yes.

DR. RANDOM: Well, I think that's an important point. And as we move towards some of these risk-informed methods, you know, uncertainty fit into these becomes a key issue. And how to evaluate the uncertainty of one thing versus another. And I think some of these probabalistic methods that are evolving are very powerful in that direction where you feed in all the sources of uncertainties, do your 60 calculations.

DR. MAHAFFY: Yes.

CHAIRMAN WILLIS: If you have all the uncertainties.

DR. RANDOM: Right.

CHAIRMAN WILLIS: But in many cases you don't.

MR. BANERJEE: You mean let your momentum go over 360 to --random number generator.

DR. MAHAFFY: Yes. But you can have a whole two day meeting on uncertainties and still not cover that.

CHAIRMAN WILLIS: So this is your last slide on architecture?

DR. MAHAFFY: I hope so.

CHAIRMAN WILLIS: Okay.

DR. MAHAFFY: Other things, additional field equations. I'm in the business of putting a droplet equation in now. What we've done as served us well for making that easy.

Improve solution procedures. We've isolated the equation solvers from the generation of the terms in the equations. It's easy to slap in a new solver. You'll see me putting a Priloft method in there pretty soon.

Additional component modules. That's been happening. People do it. There's a procedure for doing that. We actually even have on our developer's page, you have access to that.

CHAIRMAN WILLIS: That will be something like a T?

DR. MAHAFFY: Well, we're hoping to get rid of the T. I'll talk about that. But there are various components. Somebody in the audience could talk about some of the things that are going on for the ESBWR. But --

MR. BANERJEE: Let me ask you before you go, you're using still some pressure velocity coupling type of formulation for the numerical map?

DR. MAHAFFY: At the heart of it we're using what's called a semi-implicit method.

MR. BANERJEE: Right.

DR. MAHAFFY: And if you trace that back far enough, it's the old ICE method from Hirt and company.

MR. BANERJEE: Now, how easy is it for you to incorporate a method where you have something like a Jacobean like CATHARE?

DR. MAHAFFY: It better be easy, because I've got to get that done in the 1-D within the next year.

MR. BANERJEE: Okay. So you'll actually calculate that Jacobean --

DR. MAHAFFY: Yes.

MR. BANERJEE: By algebra somehow?

DR. MAHAFFY: No. That's not my proposal.

MR. BANERJEE: Okay. Why not?

DR. MAHAFFY: If you look at it, there's a lot of literature on this. And if you look at it from a rational computational basis, ideally you've got to do a mix. But if you think about it, there are certain, in effect, functions and regards to their constituent relationship that gives you interfacial heat transfer, a surface. Think of that as some function of my primary system variables.

From a purely computational standpoint it is quicker for me on those types of functions, certainly, to evaluate it twice for two different values of pressure, take a difference and divide by the difference in pressure than it is goes through and get the algebraic --

MR. BANERJEE: But wouldn't you have to do the algebraic once by using Mathmatica?

DR. MAHAFFY: No, no. no. I'm telling you after I've done that -- yes, I understand that. I also use mathmatica to generate my algebraic derivatives. That's the way I do it. I've done that for a long time.

You know, right now you use Mathmatica, you used to use Maxima.

But that's done. It's in your code. What I'm telling you is you look at the code that Mathmatica spit out to give the derivative of interfacial heat transfer on the vapor side with respect to pressure.

MR. BANERJEE: Right.

DR. MAHAFFY: Okay. And you do an operation count. The operation count in that is higher than if I just evaluate the function twice, do my two differences and my division.

MR. BANERJEE: That's by --

DR. MAHAFFY: Yes. You're just doing a finite difference analysis.

MR. BANERJEE: So you treat it together as the most --

DR. MAHAFFY: What we're trying to do, yes.

MR. BANERJEE: Okay. So you are going through a Newton-Raphson sort of procedure?

DR. MAHAFFY: Yes.

MR. BANERJEE: That makes me happy.

DR. MAHAFFY: Yes.

MR. BANERJEE: Because I've always the pressure velocity coupling to be difficult to --

DR. MAHAFFY: Be careful. The pressure -- we have always within the pressure velocity coupling scheme, what we call semi-implicit, it's the same semi-implicit method that's been used. We'll look at this in RELAP5 with a couple of differences. But the difference that we're following is that you look at it -- we're not doing a single linearization. The semi-implicit method represents a set of coupled non-linear equations. When solved properly they will rigorously conserve mass --

MR. BANERJEE: In total or for each component?

DR. MAHAFFY: For each volume they were rigorously conserve mass.

MR. BANERJEE: How do you do that by adjusting one pressure?

DR. MAHAFFY: Well, you're adjusting temperatures, too.

MR. BANERJEE: It doesn't work. You have to adjust the volume fraction as well as the pressures, which is your --

DR. MAHAFFY: You're adjusting all of those things. We can look at it. Those were slides I wanted to skip in the numerical method, but if you want to, we'll look at them.

MR. BANERJEE: You can do all time. But Brian Spalding tried this and she put a outer loop with --

DR. MAHAFFY: Yes, yes. Brian Spalding, I sat down with Patankar, circa 1980, and I told him what was wrong with their methodology. And I think he finally caught on. But --

MR. BANERJEE: Anyway, the whole thing becomes academic if you are going into the Newton-Raphson proceeding.

DR. MAHAFFY: We are doing Newton-Raphson. And I can convince you, we rigorously conserve mass right now.

MR. BANERJEE: For each phase, not for the total?

DR. MAHAFFY: For each phase. Yes.

MR. BANERJEE: Okay. You can do that.

DR. MAHAFFY: Yes.

CHAIRMAN WILLIS: Can we move on to the next.

DR. MAHAFFY: Yes, let's get on.

DR. RANDOM: John, just a clarification here. Accomplishing that by an iterative scheme.

DR. MAHAFFY: Accomplishing it by an iterative scheme. You know, we're taking what's in RELAP5, but we just keep reiterating.

DR. RANDOM: Yet, but you told me you're talking about --

MR. BANERJEE: You're talking about a iterative scheme at the converters.

DR. MAHAFFY: Yes. And we converge. And it is structured as a Newton-Raphson -- I like to say when I grew up Newton-Raphson was a method that didn't update to Jacobean. And Newton updated to Jacobean each iteration. We do a Newton scheme within that way of definition. All right.

Let's talk about the field equations. I want to go through this quickly. You guys interrupt me when there's something you want to see.

Basically there's some general background histories here. And the background is that in the beginning, I can remember being at meetings with Vic at various times, and RELAP5 would be doing one thing because they were the code that was designed to run as fast as possible and they were trying to capture the approximations in as little bit of equations as they could.

TRAC was the code that was supposed to just only be run once every three years to double check RELAP5 or whatever you needed. And it was all kinds of detail that was supposed to be in there, and perhaps wasn't. But the important thing is there was a converging evolution between these things. And the thing that's worked for us is that if you look there's not a lot of difference between the actual field equations in RELAP5 and TRAC. And we tried to look at those.

And a lot of what goes on, you know, I can talk to in terms of simplified equations. If you really look at when you go to 1-D modeling, this is what you get in terms of converting your divergence operator over to 1-D averaging.

To me, the equivalent of this in the momentum equation is where you get into your biggest error problems. Okay. And I'm not going to satisfy you in any kind of near term situation because I'm stuck with this area averaging convention.

CHAIRMAN WILLIS: This is simply a definition of divergence, really.

DR. MAHAFFY: Yes. But the idea -- the other thing it involves here, it's right here. Okay. When you do this, and you do your error averaging there's this assumption that the product of the averages is equal to the average of the products. That's what kills you in any of these equation sets. That's the biggest source of error in --

MR. BANERJEE: But you can always put a distribution coefficient.

DR. MAHAFFY: You can put a distribution coefficient in. And we can look at that as time goes on.

MR. BANERJEE: Why do you use, just as a matter of discussion, the momentum equation is a vector equation. And Brian -- I mean Graham and I and many of us has problems because when you go into these volumes and things like that, it's very hard to do this properly in terms of the linear momentum equation because your directions were changing.

DR. MAHAFFY: Yes. What you want to do, I didn't reproduce it here. I didn't have time and we don't have time to talk about it. But if you look at this -- I suspect Vic's been through this exercise, too. But if you take this equation and you do a rigorous -- the easiest way to look at it is you do a volume average of equation --

CHAIRMAN WILLIS: You reach a place where you can't solve it.

DR. MAHAFFY: What you do is you end up with these distribution functions that you don't have.

CHAIRMAN WILLIS: But even then you have forces from the wall you don't know, all kinds of things. But that's not get into that.

DR. MAHAFFY: I can get rid of those walls forces, but that's another time. I've gone through that exercise.

MR. BANERJEE: I've actually changed this to a mechanical energy equation, which is a scalar question. But at the end of it it doesn't really help you very much either.

DR. MAHAFFY: No.

MR. BANERJEE: You see, you end up more or less at the same point.

DR. MAHAFFY: This is something --

MR. BANERJEE: We knew it came from multi-dimensionality.

DR. MAHAFFY: We can spend a whole day talking about this.

CHAIRMAN WILLIS: We've got to move on.

DR. MAHAFFY: Yes.

DR. RANDOM: Well, there are two points here, actually. There are multidimensional applications for multidimensional treatment and then there are 1-D. And if you talk about, an example, oil pipelines or chemical plant systems, you are not going to treat those multi-dimensionally. And consequently, if you reduce it to one dimension, these are no longer vector equations. They're actually scaler equations. The only thing that has any vector property is the body force. And the only components that's left is the one that's projected on the spacial dimension.

CHAIRMAN WILLIS: There's no way you can get something like the momentum equations for a T just by writing down some averaging law. It doesn't work.

DR. RANDOM: That's right.

MR. BANERJEE: We could even in pipelines, we've moved to 3-D components wherever their terrain changes and drafts and --

DR. MAHAFFY: Well, but somewhere you have to have boundary conditions that are taken from the 1-D, you know. So you match them up there.

MR. BANERJEE: But that's the trick how to do that.

CHAIRMAN WILLIS: Let's move on.

DR. MAHAFFY: Yes. Again, you want to set up a day where we can discuss momentum equations, that's fine. But we shouldn't mess with it here. And I will concede that, you know, there are things we can do to improve it, and I'll be happy to sit down with you and show you how we get what we get. And then where I see the places where we could do something to improve it. But, again, I go back to the Neptune Project if you really want to get this right.

MR. BANERJEE: Or will you be able to incorporate in this structure 3-D components if we found out laws of how to get the distribution from a

1-D to a 3-D and vice versa?

DR. MAHAFFY: Yes. That's the big issue. And you've seen it before in Joe's presentation. I'm going to talk about it again. One of the ideas with this whole code coupling issue is we want to be able to couple this so that TRACE in effect becomes boundary conditions to a CFD code. And you hit the nail on the head. The real trick in that, and I'm involved with some OECD work where we're talking about benchmarks on this and whatnot. The real trick is getting that distribution to convert your 1-D solution into something useable as an input to a CFD where you're going from --

CHAIRMAN WILLIS: Here's what you should do. You should have straight pipes or a simple 1-D. But when you get into, say, going into the downcomer with a T, with a little piece of CFD there.

DR. MAHAFFY: Yes.

CHAIRMAN WILLIS: Then you have -- plenum, and you have a piece of CFD because that's what you need to model what's happening.

DR. MAHAFFY: The code is structured so that if you need to, you can do that. We don't have specific coupling on this specific code to some CFD -- it's doable, it's a formality.

MR. BANERJEE: But you can plug it in?

DR. MAHAFFY: You can plug it in.

MR. BANERJEE: Provided you get these rules, okay, to do.

DR. MAHAFFY: Yes.

CHAIRMAN WILLIS: Okay.

DR. MAHAFFY: Again, the other thing that's interesting in TRACE is that we actually quietly have two sets of flow equations. We do something a little different when we switch into 1-D, and it's done in a conservative way. You guys have seen all this stuff.

CHAIRMAN WILLIS: These are differential equations.

DR. MAHAFFY: Yes.

CHAIRMAN WILLIS: And you're going to apply them to boxes, right?

DR. MAHAFFY: The boxes are in the solution methodology, all right? We just --

CHAIRMAN WILLIS: It always bothers me. I mean, you want to start with the box and write the equation for the box.

DR. MAHAFFY: Well, when somebody tells me field equations what I want to do is I want to write the differential equation is that I then average over volumes to get the boxes. Once I average over the volumes, I'm starting to get into my numerical methods, really. But --

CHAIRMAN WILLIS: When you average over the volume, you get what you could have got by writing the equation for the box right away.

MR. BANERJEE: That's just a methodology.

CHAIRMAN WILLIS: Okay.

MR. BANERJEE: Let's get on with them. If you want to do.

CHAIRMAN WILLIS: Because the differential equations came from little boxes.

DR. MAHAFFY: Yes, they did.

MR. BANERJEE: Either one way or the other.

CHAIRMAN WILLIS: Okay.

DR. MAHAFFY: We can solve it all, you know. The computers are getting big enough at some point we just solve the Boltsman equation and we're done.

MR. BANERJEE: It all came from an action, pure and simple.

CHAIRMAN WILLIS: Who phased flow. Okay.

DR. MAHAFFY: Okay. Here's one thing I wanted to highlight, something that remains to be done. If you look at RELAP5, it's going to allow you to calculate any number of non-condensable gas species. Right now TRACE is inherited from the old TRAC code a scheme where it supports a wide range of non-condensable gases, but you can only track one at a time. You make a decision up front, you're tracking nitrogen in a system and that's what you do.

There is work that's actually largely been completed out at LANL, and it's based on some stuff that was lost years ago to get back to the full capabilities RELAP5 did.

One thing I will say, it looks like we need to get extra stability to push it a little beyond the numerical representation that was done in RELAP5. We can talk about details of that if you want to.

The other thing that is in TRACE are trace species. You can put any number of mass equations in there you want as long as it's a trace species that there's an interact --

CHAIRMAN WILLIS: So you can do chemical reactions and things like that?

DR. MAHAFFY: Yes. As long as they're not interacting thermal-dynamic. You got to watch it. If you're doing chemical reactions and that contributes thermal-dynamically to the system, we're not going to do that for you.

CHAIRMAN WILLIS: It doesn't appear in the energy equation.

DR. MAHAFFY: As long as it's not enough energy to contribute to the energy equation we're okay. But, for instance, I've got a project through the DOE where we're doing water chemistry. Okay. And we got all these trace species and we can put in whatever chemicals we want, and we can follow them around. And you can exchange. You can set up things so that you have a trace species in the liquid and it comes out into the gas. But it's not going to do your dissolved nitrogen problem because it's a trace species. It doesn't --

CHAIRMAN WILLIS: So it'll boron? It'll do boron, for instance.

DR. MAHAFFY: Well, boron's already there in a special field anyway.

MR. BANERJEE: But boron will have a vapor pressure, too, right?

DR. MAHAFFY: You could do it, but at no point do we do anything with boron that contributes to the thermal-dynamics of the system.

CHAIRMAN WILLIS: Energy equations you mean?

DR. MAHAFFY: Boron just sits there and it does what it does. It plates out. It interacts. Well, it contributes to the thermal-dynamics of the system to the extent it interacts with the neutron kinetics. But that's where boron's important.

MR. BANERJEE: But let me understand why it's so difficult to put some of the stuff into the thermal-dynamics. I mean, it's just a chemical reaction, right?

DR. MAHAFFY: Yes.

CHAIRMAN WILLIS: What's the problem?

DR. MAHAFFY: It's easy to write mass equations, okay, that stand alone. Given a velocity field from a full set of your fluid equations that you've solved, it's easy to generate any number of mass equations where your densities are being propagated around by a predetermined velocity field.

MR. BANERJEE: Species condensation equation.

DR. MAHAFFY: Yes. Yes. That's easy to do. As soon as these species start feeding back into the pressures, I've got some dissolved nitrogen in a liquid, it comes out into the gas, there's enough coming out into the gas, it's noticeably changed the pressure. Okay. Then I've got to add coupling equations--

CHAIRMAN WILLIS: So you don't do partial pressures?

DR. MAHAFFY: We do partial pressures.

CHAIRMAN WILLIS: Okay.

DR. MAHAFFY: Okay. But we're not -- we don't have that coupled in yet. And these are tasks later down the line. But I'm telling you --

MR. BANERJEE: Do you see any problems in principle or just that you wanted to defer that task?

DR. MAHAFFY: No. It's a task that hasn't been done yet.

MR. BANERJEE: Okay.

DR. MAHAFFY: This was an easy thing to do. There was a need for people at times to look at trace species, whether you're doing water chemistry, you're doing radionuclides that you're trying to follow around, it was an easy thing to put in the code. It's in there. I'm just making sure you understand that there's a class of these things that we don't have yet.

MR. CARUSO: What defines a trace?

DR. MAHAFFY: What defines a trace species is that it does not interact thermal-dynamically with the rest of your system.

MR. CARUSO: And there's a metric to determine that so it doesn't raise the pressure by--

CHAIRMAN WILLIS: It's an inert gas --

MR. CARUSO: -- more than one percent?

DR. MAHAFFY: You could define whatever metric you want. I mean, my metric is that I don't care what's going on, it won't interact thermal-dynamically with this system.

MR. BANERJEE: It's a passive scaler that sits in the liquid.

DR. MAHAFFY: Yes, that's right. It's a passive scaler that sits in the vapor.

MR. CARUSO: All you solve for is the mass fraction.

DR. MAHAFFY: That's all you're doing.

MR. CARUSO: Right?

DR. MAHAFFY: That's all you're doing. It's there, it's useful, but it's not as useful as--

MR. BANERJEE: But doesn't transfer from one place to the other.

DR. MAHAFFY: It will transfer. Okay. Again, you can have a passive scaler --

MR. CARUSO: So can you do the --

DR. MAHAFFY: -- in a liquid. It can transfer to a passive scaler in the vapor. But in doing so, it can't effect the total pressure of that vapor.

MR. CARUSO: So you can't calculate opening a soda can?

DR. MAHAFFY: Not now.

MR. CARUSO: Not yet?

DR. MAHAFFY: Not yet. That's something, one of the NRC guys can tell you about a schedule on that, but it's something we failed with.

CHAIRMAN WILLIS: Because if I shake up the -- I get different answers.

DR. MAHAFFY: Yes.

DR. RANDOM: A point of clarification. If you inject nitrogen into the system, though, it is considered to be -- it's partial pressure detracts from the steam --

DR. MAHAFFY: Yes. See, the nitrogen is followed by this special non-condensable equation.

DR. RANDOM: Okay. So you have a special vapor component?

DR. MAHAFFY: Yes. There is an equation for the nitrogen.

MR. BANERJEE: Yes, you see on the right hand side which has to be changed if you're transferring from one place to the other.

DR. MAHAFFY: That's absolutely correct. That's right. And that hasn't been implemented in that. You've seen the stack of things we had to do, and this wasn't even on somebody's list that you saw. It is on lists that do exist.

Phase change terms. Everybody uses heat conduction limited model. That's what's still in there. All right. Energy equations. They're nonconservative in form. We have a project scheduled to move to a fully conservative energy equation. I don't know when that's going to happen, but it's not too distant in future.

Wall energy -- yes?

DR. RANDOM: What do you mean backed by the primes on the h_v and h_l? You say you've gone to a donor type --

DR. MAHAFFY: Yes. It's the same donor --

DR. RANDOM: So you don't superheat the vapor?

DR. MAHAFFY: That's right.

DR. RANDOM: Yes.

DR. MAHAFFY: Everything you're familiar with RELAP5 land in terms of donored enthalpies goes on TRACE. And I will tell you, if you're looking for physical flaws, the reason -- is everybody here familiar with this game of donoring enthalpies?

If you go back to your heat conduction model, all right. And you've got your source of heat to the interface and you're dividing by your latent heat of vaporization in effect. Those primes tell you it's not really a latent heat of vaporization. If I'm boiling, so I'm taking liquid and I'm making vapor, I'm doing what Vic refers to as a donor enthalpy. The enthalpy of the vapor, the vapor is appearing at the interface so it appears at the saturation temperature. But the liquid coefficient, it's donored from the bulk liquid. That liquid enthalpy is evaluated at the bulk liquid temperature.

If I don't do that --

CHAIRMAN WILLIS: You don't conserve energy if you don't do it right, do you?

DR. MAHAFFY: Well, you conserve energy. Okay. But what will happen is you can get into situations where you think of it as you're taking away just a limited spectrum of your Maxwellian curve. Okay.

I've got a liquid at a temperature of 300 degrees and I'm pulling off liquid at 450 degrees. Well, what happens is liquid starts cooling down and you can get into these runaways where your liquid gets colder, your steam gets hotter.

Now, what I maintain is that this artifact really is a result of imperfections in my knowledge of the interfacial heat transfer areas and heat transfer coefficients.

CHAIRMAN WILLIS: Because of your lumping or your averaging, or whatever? You're using the liquid as if it's at some temperature which is not the same as the interface temperature. But that's all you can do.

MR. BANERJEE: But then I have a problem. Basically the q there, let's say the q_{il}.

DR. MAHAFFY: Yes.

MR. BANERJEE: Is calculated as being a heat transfer coefficient into a difference of temperatures, right.

DR. MAHAFFY: Yes.

MR. BANERJEE: So the temperature of the liquid is what you would call a h prime. CPTL is equal to h prime L. It has to be.

DR. MAHAFFY: Be careful. These h --

MR. BANERJEE: No, I'm saying the h on top.

DR. MAHAFFY: This h here is your saturation.

MR. BANERJEE: Is the enthalphy.

DR. MAHAFFY: And enthalphy.

MR. BANERJEE: So h prime --

DR. MAHAFFY: Yes.

MR. BANERJEE: -- $_l$ is equal to c_p into t_l. It has to be.

DR. MAHAFFY: Okay.

DR. RANDOM: There's something wrong with that slide.

MR. BANERJEE: Now I go on.

DR. MAHAFFY: There was a copy job. You're getting two liquids instead of a liquid and a vapor. I apologize.

MR. BANERJEE: No. But that's all right. Let me go back to the q_{il}. q_{il} is equal to a heat transfer coefficient, h_{il}.

DR. MAHAFFY: Yes.

MR. BANERJEE: Into an interfacial area a_i. Into this bracketed quantity which is the saturation temperatures interface minus the temperature of the bulk liquid.

DR. MAHAFFY: Yes.

MR. BANERJEE: That gives you the heat flux.

DR. MAHAFFY: That gives you the heat flux of the interface.

MR. BANERJEE: Okay. And then q_{ig} would be the corresponding -- yes. That's fine. You don't worry about it.

So you get the sum of the two gives you the next heat flux at the interface.

DR. MAHAFFY: Yes.

MR. BANERJEE: Okay. Now you should be dividing by H_{fg} strictly. I don't see why you are dividing it by this quantity far away because -- the vaporization is the next heat flux divided by the latent heat.

DR. MAHAFFY: That's right.

MR. BANERJEE: You've already taken the drop in temperature --

DR. MAHAFFY: Yes.

MR. BANERJEE: -- into account in calculating q_{ia}.

DR. MAHAFFY: That's right. And that was the final point I was trying to make.

This game here is a code developer's fix to a problem in lack of precision in evaluating these terms right here. If I had done these right, you're absolutely correct. Everything would have worked out.

What's really happening is that I'm not doing these correctly in a full consistent manner. And when I do those incorrectly in the wrong direction, I get into these runaway situation where, you know, I start off -- if I've got subcooled liquid and I'm boiling it from hot vapor, okay. The more boil -- well, you get -- that runaway may be legitimate.

But you get these temperature runaways that are nonphysical. And if I condense super heated steam, I can get the steam heating up as I condense it from subcooled water. And it's because I'm not doing these interfacial heat transfer coefficients.

MR. BANERJEE: I can see that you could have problems with the interfacial heat transfer coefficient in areas. But I don't think that it's easy to justify the fix that you've put in for --

DR. MAHAFFY: No. No. I'm not going to stand up here --

MR. BANERJEE: It doesn't compensate. How do you know it's --

DR. MAHAFFY: As a duly physicist, I'm not going to stand up and try to defend this on a physical bases.

CHAIRMAN WILLIS: But the problem is, Sanjoy, that you could -- what you say is right for calculating the flux of the interface.

MR. BANERJEE: Right.

CHAIRMAN WILLIS: But then with the box thing, they immediately assume that the stuff which is injected then gets the properties of the average vapor, which it doesn't get. But anything in the box has the average vapor properties.

DR. MAHAFFY: You guys wanted to hear some of the ugly downsides of codes, this is one that's been in there for 25 years.

CHAIRMAN WILLIS: No, this should be looked at.

MR. BANERJEE: Yes, it should certainly be examined carefully.

CHAIRMAN WILLIS: Something we can examine next time.

DR. MAHAFFY: Well, it's more than just next time. What you're going to find is, you know, this is something that certainly I've watched and, hopefully, the next generation developers when I move on somewhere will also watch, as I say, as these things get better, and that may come with the interfacial area transport equations or whatever, we need to go back and get this fixed. No question about it.

MR. BANERJEE: I will say that this should be tabled as an item, because it's indefensible. What you can say is that we should put lock down certainty somehow into the coefficient because that is defensible.

DR. MAHAFFY: Yes.

MR. BANERJEE: But to do this doesn't make any sense.

DR. MAHAFFY: You guys can schedule all kinds of meetings. I'll tell you that --

CHAIRMAN WILLIS: We'll do it in the future.

DR. MAHAFFY: But right now these guys are hard to defend, too.

CHAIRMAN WILLIS: You don't know AI very well.

DR. MAHAFFY: No. Compared to what you don't know here and what I didn't know here, is small potatoes.

But, yes, Joe?

MR. KELLY: I'll just jump in for a second, because I -- this is Joe Kelly.

What you could really do to make me feel a lot better that is write that first equation in a slightly different way so that you have your interfacial heat transfer, divide by your normal latent heat. Then you'd have another term which would account for the energy it takes to bring that phase up to the saturated temperatures.

CHAIRMAN WILLIS: That's right. That's right.

MR. KELLY: It's just straight forward energy balance.

DR. MAHAFFY: You can do that.

MR. KELLY: It's just it's very, very confusing the way that's written. It goes against, you know, your experience. That's all it is.

CHAIRMAN WILLIS: It doesn't have to be confusing.

DR. RANDOM: Well, I think the problem is there are no correlations for the heat transfer coefficient to do that --

CHAIRMAN WILLIS: That's another problem. That's another problem.

MR. KELLY: Right. Because the temperature of the phase has to change as it approaches the interface. You can't do that.

MR. BANERJEE: I would say there are reasonably good coefficients for the heat transfer correlations -- coefficients, but there aren't for the interfacial areas. I think that's the truest statement. Because it's more difficult to get it.

It's a very interesting problem in turbulence, by the way. It goes -- you can show that the factual dimension of an area over a length scale has to go to 2.34, which suggests that if this is very churned up, that you'll have to get to that ratio.

CHAIRMAN WILLIS: It has to do with averaging. You have a vapor space that has a certain temperature. And you remove vapor at a lower temperature. It's artificial. Then you have heat supplementing behind, it's going to be hotter. So it's ridiculous. It makes no sense.

MR. BANERJEE: It's a box problem, yes.

MR. KELLY: Exactly.

CHAIRMAN WILLIS: Right. Well, let's move on.

DR. MAHAFFY: Anyway, I've presented it before the way Joe's talked about, but then you have to get into defending the partition of energy between--

CHAIRMAN WILLIS: Yes. Now let's go on.

DR. MAHAFFY: -- coefficients. Okay.

MR. BANERJEE: We will defend this offline sometime.

DR. MAHAFFY: I want to skip over this

unless somebody's particularly interested. This is just the way we do the single phase.

CHAIRMAN WILLIS: Yes, we're going to go through this I think in my detail some other date.

DR. MAHAFFY: Yes. Yes.

CHAIRMAN WILLIS: We've seen all of this kind of stuff before.

DR. MAHAFFY: Momentum equation, again, you now this isn't going to make you happy because are just field equations.

CHAIRMAN WILLIS: These are just --

MR. BANERJEE: Except I have a problem with the momentum equation even for a straight pipe there. Because you have a hydraulic head term for a horizontal pipe if you ever tried to do that.

DR. MAHAFFY: Well, this one's -- no, this actually will work out once you --

MR. BANERJEE: Unless you add a term.

DR. MAHAFFY: When we actually do the averages on this and come up with a term, we don't get head terms in a straight --

CHAIRMAN WILLIS: Okay.

DR. MAHAFFY: You have to do something special to get the head terms --

MR. BANERJEE: How do you get flow? Let's say you had a pipe.

DR. MAHAFFY: Yes.

MR. BANERJEE: I had more liquid at one end.

DR. MAHAFFY: Yes.

MR. BANERJEE: It was running the pipe and running out the other end.

DR. MAHAFFY: Yes.

MR. BANERJEE: How do you handle it --

DR. MAHAFFY: It's not encompassed in these equations.

CHAIRMAN WILLIS: It's not?

DR. MAHAFFY: That's correct. You have to go into the depths of the manual and to how it does a horizontally stratified flow regime, and you'll see a modification of these equations. It involves a void fraction derivative.

MR. BANERJEE: Right.

DR. MAHAFFY: Yes. That's true.

Let me show you one thing. This is probably the biggest place we differ for now from RELAP5

CHAIRMAN WILLIS: You have no added mass?

DR. MAHAFFY: We have no virtual mass terms. They're not there. I'm just being up front about that. And, again, it has to do with two things.

Number one is that, again, our source that we've evolved from is TRAC. TRAC was always in this form. RELAP5 has been in this form for a very long. Vic can tell you when that started.

There were huge fights. Dennis Lyles used to get into fights with various people on various boards about this. And the bottom line was that we were never provided examples where this made a profound difference in the answers.

And I invite you to provide those examples, and we'll be happy to go in and make this change, and schedule it. It's not a big deal.

CHAIRMAN WILLIS: You have to have a pretty rapid acceleration to make a difference.

DR. MAHAFFY: Yes. It's not something that passed the so-what test, really, for us. And so rather than introduce extra complexity in a numerical solution procedures, we've kept this level of approximation --

CHAIRMAN WILLIS: But if you understand accelerating a bubble through a liquid, all of the inertia is in the added mass.

DR. MAHAFFY: Oh, yes.

CHAIRMAN WILLIS: So it's got to give

completely the wrong answer to that problem.

MR. BANERJEE: Yes. Whether this is important for the specific transients you're looking at or not, there's a classical example in the first five pages of Landau and Lipshitz on an oscillating bubble. And your bubbles are not going to oscillate if you do that.

CHAIRMAN WILLIS: Well, if you take a bubble in a pipe, sitting there like a sperm level.

DR. MAHAFFY: Yes.

CHAIRMAN WILLIS: And hit it with a hammer, the bubble will move.

DR. MAHAFFY: Yes.

CHAIRMAN WILLIS: Whereas it wouldn't. If you didn't an area mass in, it wouldn't move.

MR. BANERJEE: It will give you a completely wrong answer if you have even a bubble trapped in a pipe and you move the pipe.

CHAIRMAN WILLIS: Right and accelerate. Yes, or accelerate it anyway. It's like hitting it with a hammer.

DR. MAHAFFY: I understand. I understand exactly what you're saying. But --

MR. BANERJEE: Probably there's so much numerical diffusion in these codes anyway that --

DR. MAHAFFY: Yes.

MR. BANERJEE: What do you find? Nobody knows what comes out anyway.

DR. MAHAFFY: If we had versions --

CHAIRMAN WILLIS: All the experiments are correlated using the friction factor, which sort of takes account of this in some way.

DR. MAHAFFY: We had versions of TRAC with both forms of these equations, and we couldn't get the lower more complicated form to make any significant difference in the answers, and numerical diffusion may be part of it, classic problems.

What I'm inviting you to do by putting this viewgraph, in your wisdom if you can come up with assessment problems that should be added to our assessment set relevant to nuclear reactor safety, yes, I mean, I can produce lots of simple test problems, as you say, that exhibit this term. But, you know, if those things come in, then that's fine. We'll get that in there and that'll go on the list. But for right now, this is not one of our high priority items.

That's my statement. You can cogitate on it and we welcome any kind of creative solutions.

One thing that's gone on, the interfacial drag there's a difference there. You get these distribution factors.

Joe, are those distribution factors in some of the modifications you got scheduled to go?

MR. STAUDENMEIER: The distribution factors are stuck in in the computation of interfacial drag coefficients. The equations themselves don't have the distribution factor.

DR. MAHAFFY: Okay.

MR. STAUDENMEIER: There's a ratio multiplier that takes those into account.

DR. MAHAFFY: Okay. So we're effectively going to be adopting those --

CHAIRMAN WILLIS: This RELAP thing is a very strange looking --

DR. MAHAFFY: Well, again, it's a way of adjusting for the fact that you've got profiles of velocity across a given region, is what's going on there. Part of your area averaging results.

MR. BANERJEE: Would you accept as a test problem a pipe full of bubbles and being fluid fairly high velocity and predict the void waves? Is this relevant? The dynamic waves, not the kinematic waves?

CHAIRMAN WILLIS: The acoustic. The acoustic.

MR. BANERJEE: What you see is, of course, is that the void fraction is not uniform. It forms waves. And you see this in fluid -- as well. Eventually they lead to bubbles in fluid -- this is the famous Jackson instability.

DR. MAHAFFY: Okay.

MR. BANERJEE: But if you did it with bubbles in a liquid, you get these void waves. And people have measured the velocity of these void waves and their amplitudes. Is this a problem that would be of any interest to the --

DR. MAHAFFY: I can't off the top of my head relate it to anything. But I'm not thinking along those lines at the moment, so --

MR. BANERJEE: Well, the question is, is the propagation rate of void waves of any importance in nuclear reactors, or is anything of any importance? I don't know, other than liquid falling out of the hole.

CHAIRMAN WILLIS: We could go onto this forever.

DR. MAHAFFY: Yes.

CHAIRMAN WILLIS: There are all kinds of assumptions here which are a bit hooky.

DR. MAHAFFY: Yes.

DR. RANDOM: Well, a lot of these coefficients were part of the incorporation of the drift flux model for interface drag computation. Like the absolute value of the velocity difference is actually in the FIG, you know. So there's still the same kind of dependence.

DR. MAHAFFY: What Joe just said, he's picking that up. Joe has got some changes to the interfacial drag model already that are coming in. It's not there.

I mean, we identified very early a lot of models that were high probability changes, and interfacial drag was very high on the list. You know, you and I sat through a lot of the review of the RELAP5 for the AP-600. And there's a whole lot of models that just -- there was a huge amount of effort in justifying them and then pulling them together that really need to be reviewed still. But that's partially--

CHAIRMAN WILLIS: Well, the message I get from this is that TRACE has all the faults of all the codes we've seen from other sources.

DR. MAHAFFY: In terms of the field equations, sure.

MR. BANERJEE: And leaves even a few terms out here and there.

DR. MAHAFFY: Yes.

MR. BANERJEE: Compared to what should be there. I mean, it exists. Whether it's important, that's another problem.

DR. MAHAFFY: That's right.

DR. RANDOM: Well, it may contain all the faults, you know. Are there any improvements that could or should be made?

DR. MAHAFFY: Yes. Yes. And I understand that we can look at this momentum thing in detail, but again mass is conserved rigorously, energy will be conserved rigorously. Those are two of the biggest things you got to worry about.

Solution flow. I just want to do this at a very high level. If you want to get into details, we can worry about it later.

But basically everything that's done in the solution should be recognizable to people who have looked at these codes over the last 20 years. Okay.

CHAIRMAN WILLIS: What does SETS mean?

DR. MAHAFFY: SETS is stability enhancing two step method. I developed that in 1979.

CHAIRMAN WILLIS: Okay.

DR. MAHAFFY: It's been in TRAC running

robustly since 1982. A version of it called Nearly Implicit has been in RELAP5, not running quite so robustly until very recently. But it is -- it embarrasses me some as to some point that this method is still used. It was my response to Three Mile Island. All right.

In the beginning Vic and I were tied down to large break LOCAs, right? That was the first thing we had to worry about. He started worrying about other things before we did, but you got to a case if you look at the semi-implicit numerical method, you've got this material collosability limit. Okay. Time step has got to be less or equal to a characteristic cell dimension divided by a flow velocity. And when you're trying to run a transient that runs on for hours or days, and you're taking those small time step sizes which are very small compared to the characteristic time scale for change in any physical parameter in the system, you're killing yourself. Because that time set size is well below anything that reflects accuracy, even in a first order time accurate method. So we needed something that limited that method. We didn't want to go to fully implicit. SETS is a way of adding a stabilizer step to each of the key equations, momentum, energy and mass, that wipes out that material collosability limit. It's cheaper by about a factor of five in terms of cost per cell per step than a fully implicit method.

It still has the problems --

CHAIRMAN WILLIS: What's the technique, something relaxation, or something?

DR. MAHAFFY: What are you looking at?

CHAIRMAN WILLIS: Well, maybe makes it semi-implicit. I forgot -- I'm looking for a piece of jargon.

DR. MAHAFFY: Yes, semi-implicit --

MR. BANERJEE: -- corrector.

DR. MAHAFFY: Yes, it is.

MR. BANERJEE: That's all it is.

DR. MAHAFFY: In effect, what you're doing with the SETS method, you use the semi-implicit method to get your velocities with which you're going to move things around. And then you go with those velocities fixed. You write some equations that look like fully implicit mass and energy equations, and away you go.

DR. RANDOM: I think you basically said it, but the difference between whether you want to use one or the other is whether or not material propagation effects are important or not.

DR. MAHAFFY: Yes.

DR. RANDOM: Because the more implicit schemes will diffuse that propagation. And there are a number of cases where you do not want the diffusion and --

CHAIRMAN WILLIS: -- stability, you time step a millisecond and nothing of interest is happening --

DR. MAHAFFY: Right. That's right.

DR. RANDOM: And that's in stationary systems.

DR. MAHAFFY: But this is what Vic saying. Really, there are a whole class of these slow transients that all they really are are an evolution from one quasi steady configuration --

CHAIRMAN WILLIS: That's right.

DR. MAHAFFY: -- to another.

CHAIRMAN WILLIS: That's right.

DR. MAHAFFY: And then you want some kind of much more implicit method.

CHAIRMAN WILLIS: Right.

DR. MAHAFFY: But if you're zinging waves around somewhere or continuity waves of any sort, you should be using semi-implicit. And that -- that is in there. You can choose semi-implicit, you can choose a SETS base method. We really ought to be automating that at some point in the future to adapt these things. We haven't done it yet.

DR. RANDOM: But, John, along that line if you run the sets method with the same kind of time step algorithm that goes along with the semi-implicit, don't you get the same limit in terms of diffusion? I mean, there's no benefit to it, basically, but it should produce the same result.

DR. MAHAFFY: Yes. If you look at what it happens is that in terms of diffusion, you could run the classic diffusion problems where you take a heavy side step function and you propagate it through a 1-D flow channel with a fixed velocity field. And what you see that for a semi-implicit or an explicit method, since we've fixed velocity, your diffusion is maximum at the lowest time step size and it lets less as you approach the stability limit.

DR. RANDOM: Right.

DR. MAHAFFY: We have a fully implicit method, and this should make sense logically, at the minimum time step size it's going to have the same numerical diffusion as the explicit or semi-explicit method, it has the time step size increases it just gets worse.

DR. RANDOM: Well, that's what I was

saying.

DR. MAHAFFY: Yes.

DR. RANDOM: I mean, if you learn the SETS method at the minimum time step --

DR. MAHAFFY: At the minimum time -- if your time step size is significantly less than the material collosability limit, then it doesn't matter.

DR. RANDOM: Right.

DR. MAHAFFY: Yes.

MR. BANERJEE: It's just that the diffusion coefficient for an exclusive matter has a difference in it. So as you get to a number of one it goes almost to zero rather than the implicit one that you've got.

DR. SIEBER: Yes. Right.

DR. MAHAFFY: But the implicit one if you run it at its minimum time step becomes the same.

MR. BANERJEE: Yes. Of course.

DR. RANDOM: Then why are they using the same spacial difference -- in fact, there are some things that are interesting about implicit schemes. Sometimes if you run them at less than the limit, they become unstable, which is a kind of peculiar result.

DR. MAHAFFY: To capture the key importance of this particular viewgraph up here, TRACE is following a RELAP5ish way of doing things rather than what was in TRAC years ago. We've isolated the generation of the coefficients in your equation from the solution of the equation. And that's probably the key important solution architecture feature in here. So that I can go in there and I can put in the linear equation solver DuJour very quickly and and that's helped us a lot.

MR. BANERJEE: How are you solving? So you get a form of a Poisson equation still?

DR. MAHAFFY: Yes. Within the semi-implicit step if you look at the footprint of it, it's the same footprint as a 5 point poisson operator. But you've got these scattered things due to your odd flow topology. And what we're doing is we're giving it to a sparse linear system solver. We're doing direct solution. There is no iterative solution on these linear systems because we just don't have a high enough variable count in any kind of reasonable system.

CHAIRMAN WILLIS: But then it's a problem with implicit methods when you have a regime change occurring in the time step? You've got a regime change occurring in the time step and you're switching from one correlation to another and you're trying to use the implicit method, can you get into trouble where the computer doesn't know which one to use and it solves it sort of --

DR. MAHAFFY: Implicit method always has problems when you get discontinuities --

CHAIRMAN WILLIS: Right. Well, you do, watching that correlations of heat transfer which change --

DR. MAHAFFY: What we try to do, and you know this can be subject of another ACRS meeting if you'd like, you're going to have to talk to Joe Kelly about this.

CHAIRMAN WILLIS: Okay. So maybe he'll tell us.

DR. MAHAFFY: There are excluders in there.

CHAIRMAN WILLIS: Okay.

MR. KELLY: When you go to fully implicit, there's another way of handling it, and that's with the CATHARE team does. At the first pass through the equations, they set their flow regimes and -- regimes. Those are then frozen for the duration of the time step. They will go back and update the heat transfer correlations within that regime, but they freeze it to that regime and don't allow it to change, just to avoid what you're speaking of.

CHAIRMAN WILLIS: Yes. Okay.

DR. RANDOM: John, are you going to cover the single phase, the two phase transition where you get into over extraction of mass and mass --

DR. MAHAFFY: I zinged by those while you weren't looking. If you want, we can go offline and talk about that.

DR. RANDOM: Well, you have that at the --

DR. MAHAFFY: No. What we do is that, again, because we're doing an interactive solution, we watch as we iterate on the solution. So we come into a time step and there's a void fraction of .001. And in iteration number two or iteration number three of our solution that goes to zero, there's logic in there that looks and says does that zero really make sense. Is that what you want to do? It's looking at your at your face teams terms, it's looking at your flux terms. And once it decides that zero is where it wants to be, it clicks it switch and it goes from the full two phase equations to a set of equations that contain b-mass, b-energy, a statement on -- let's say on the void fraction zero end it'll say alpha equals zero. It'll say t-vapor equals t-sat. And that's it.

And if you think from a physical

standpoint about what that's doing. Number one, it rigorously enforces conservation of mass again. But you can interpret that physically really as coming up with a value for the integral of gamma from the beginning of the time step to the end of the time set that's consist with elimination of the phase that's left in the system.

CHAIRMAN WILLIS: You're going to have trouble if you use different equations and you're going to go through a transition on alpha equal .001 to something where it's now going to be subcooled. You can't use a d-alpha dt that goes through that boundary. You're going --

DR. MAHAFFY: Well, I mean, as we say, you know we're doing a discontinuous fit to our equation section. It's worked very smoothly over the years. That's one thing we haven't had problems with.

I was asked to just draw up some solution diagrams to pictorially show you what's going on.

This emphasizes --

CHAIRMAN WILLIS: This is another one of these loops you never get into and never come out of?

DR. MAHAFFY: No, no. I mean, again look at the bottom line here. The arrows are showing you how information flows within a time step, but there's implication here. When you need information one of these boxes that you don't see an arrow in, are just coming from the previous time step.

And so we start a time step, we evaluate our control system. The control system could add momentum sources, which go in here and here. It can add actions on your control rods which feed into the neutron kinetics. And it can add boundary condition to my heat condition equations. So it's feeding in on a purely explicit basis with the beginning of the time step.

Our heat transfer coefficients are all explicit. And these are wall and interfacial. So they're coming down here. They feed into the semi-implicit part of the equations, the heat conduction equations. There's no line down here because of a game we play that isolates all the source terms up here. That's algebra. Friction coefficients here and here for momentum equations. And that's it.

We've got at iterative solution. This is the key thing that we're doing.

MR. BANERJEE: But what is that, because you were telling me that this interim procedure is the one that actually allows you to get mass conservation in each phase.

DR. MAHAFFY: Yes. Yes. And let me flash forward in a minute and I'll stop as I go pass my mass equation on a numerical scheme, and I hope I can convince you of that.

Very shortly we hope to be in a mode like this where our heat transfer coefficients, our heat conduction equation is all integrated with the semi-implicit step. It doesn't increase the size of our matrices, so it makes a certain amount of sense. We're doing studies on that now. We'll see how it plays out.

DR. RANDOM: Well, the heat transfer coefficients are normal in your functions but you're going to iteratively update them, is that right?

DR. MAHAFFY: Yes. They're impact will be included in the Jacobean.

DR. RANDOM: What do you do if you changer from boiling, for example, to single phase or change regime, basically?

DR. MAHAFFY: Well, that was what he was talking about. And Joe's answer is the first answer that I'm going to give you.

DR. RANDOM: What was that?

MR. BANERJEE: What kind of Jacobean?

DR. MAHAFFY: You freeze your flow regime.

DR. RANDOM: Okay.

CHAIRMAN WILLIS: Well, at some point you'd better -- you've got some zeros here.

DR. MAHAFFY: If the alpha goes less than zero, all of this kicks back to my single phase equation substitution and it will capture that correctly and it will conserve mass.

DR. RANDOM: I have a question, John, on the control system. That's a set of ordinary differential equations and the old RELAP5 didn't have an implicit solution scheme. You had to order them in the proper order.

DR. MAHAFFY: Yes.

DR. RANDOM: And have you gotten rid of that?

DR. MAHAFFY: Yes. Birol could talk to it. We basically have the iterative capabilities in there in the control system --

CHAIRMAN WILLIS: What you just said is it goes to the single phase. I got two phase coming in here, like a single phase. I got the interface. It's just double tracking business, really.

DR. MAHAFFY: Well, the level --

CHAIRMAN WILLIS: An uncommon use of a differential equation assuming it's continuous, to catch that box where you have two phase coming in and single phase going out. You don't have a d-alpha dx, which makes any sense.

DR. MAHAFFY: No. The only time you have that is when you have level tracking engaged and you have vertical flow.

CHAIRMAN WILLIS: But you have it lots of times going on in the circuit.

DR. MAHAFFY: The only time -- yes. You've identified a potential --

CHAIRMAN WILLIS: Every time you have a slug of liquid and then some two phase -- you've got--

DR. MAHAFFY: I don't disagree with you. If it's in a horizontal pipe and you got a slug of liquid zinging through there, it's not going to capture that correctly. It's going to smear it out. That's part of your numerical diffusion.

If I'm in a vertical riser, we've got a very good level tracking algorithm that will take care of that for you. But once you go into horizontal flow, there's nothing we do except --

MR. BANERJEE: Are you going to talk about this level tracking later?

DR. MAHAFFY: No, we're not. We could spend a half day on that one.

MR. BANERJEE: Then let me ask you the question, why does your level tracking methodology work for vertical flow and not for horizontal flow?

DR. MAHAFFY: You could put it in a horizontal flow. It's just a question of making your decision.

Birol, do you want to make any intelligent comments on that for him?

MR. BANERJEE: Or any comment of any sort?

MR. AKTAS: This is Birol Aktas from ISL.

The reason we use only in the vertical components is because the interface is register at a point in the scale equations. You know, it's horizontal. And then we can handle, for example, crossing of the interface across a cell boundary and it's an instantaneous crossing across the cell face because, you know, the equations are solving for scale properties and the interface itself is treated as a point. But you have, let's say, an inclined pipe then when the water level is rising part of the interface will be in a cell J minus one and part of it will be in the next cell, which is very difficult to handle. So what we do is we limit the level tracking only to vertical components.

MR. BANERJEE: And it's horizontal?

MR. AKTAS: On horizontal --

MR. BANERJEE: It's even worse, right?

MR. AKTAS: Yes.

CHAIRMAN WILLIS: But you could do horizontal as long as the interface is vertical?

MR. AKTAS: Yes. That's a very good point, yes.

MR. BANERJEE: But this methodology that you're using is actually just treating as the point.

MR. AKTAS: Yes.

MR. BANERJEE: Have you looked at things like level sets that you could treat -- with a zero on the interface?

MR. AKTAS: Yes. However, you have to remember that everything is averaged across the cross sectional area, so we don't have any variations in the lateral direction of the pipe flow, for example. So we don't have that information. All we have is a velocity in the direction for the flow and the void fraction distribution along, again, axis of the flow. We don't have anything in the left or to the right.

So if you had an inclined pipe, then we would again -- algebraically I think it's possible. I tried various things with the three dimension version of the level tracking because we have a three dimensional fluid flow component vessel. And some of my attempts are fruitless, so I just resorted back to the vertical components.

MR. BANERJEE: I see. It's an artifact because of the one dimensionality?

MR. AKTAS: Yes.

MR. STAUDENMEIER: Excuse me. Let me interject something here. Let's keep in mind the object of the presentations, which is to present an overview. If we go into detail on anything that pops into your mind, we're never going to make it through here. But you can flag it as something that you'd like to see an additional detailed presentation done at a future meeting where we can go in and discuss all the details.

CHAIRMAN WILLIS: I mean, what we're getting out of this, though -- what I'm getting out of this is that all the questions that we've always asked about all codes are coming up again.

DR. MAHAFFY: That's right.

CHAIRMAN WILLIS: Is that right?

DR. SIEBER: That's because we're used to old codes.

CHAIRMAN WILLIS: But there isn't some magic bullet that's been discovered.

MR. BANERJEE: Well, maybe they did the introductive procedure and can actually conserve mass in both phases, then it'll be maybe new.

DR. MAHAFFY: Okay. Let me show you, when I converge my iteration, okay, this is just for single phase flow but you can talk about this for two of these equations, for vapor and liquid. I converge my iteration to the point that this equation really is satisfied so that I can write the numerical equivalent of a volume integral of this whole equation. And what I get then is that the integral over volume of the new time density is equal to the integral or volume of the old time density with the integral over volume of this divergence term, which boils down to just whatever fluxs you have into your system minus the fluxs you have out of your system.

CHAIRMAN WILLIS: Yes. But that divergence term is also going on with time during this little time dt. It's changing, too. The low end in there can be old or new or some average throughout the time step.

DR. MAHAFFY: This is new, this is old. Basically what I do is I define -- I take this equation and it is what it is. You know, this relates to time levels. I integrate this -- just take this equation as it is and I integrate it over volume. Then I get a relationship between the new --

CHAIRMAN WILLIS: The new mass --

DR. MAHAFFY: The old mass and the integration of this over all volume --

CHAIRMAN WILLIS: But that other thing is changing through the time step. The flow in and out isn't just a constant. So that --

DR. MAHAFFY: But these numbers, at the time I do the integration, it's when I've solved this equation and these numbers are just numbers. But they're consistent numbers so that the flux out of one cell is exactly equal to flux into another cell and they just cancel all the way across the line.

MR. BANERJEE: But the row n end being at the old time step, is that a -- I mean, if it was row n plus one, I can see that that satisfies mass conservation.

DR. MAHAFFY: And, again, you know it doesn't matter where I evaluate time levels. I'm taking a snapshot. I've solved this equation. I now want to make some statement about mass conservation. I've solved the equation. These are just numbers, whether they're new numbers or old numbers; they're numbers and they're numbers that cancel out in a way that the flux from one cell out matches the flux into the next cell.

CHAIRMAN WILLIS: Even though it's not quite the right flux, at least it's balanced, right?

DR. MAHAFFY: Yes. And it is the right flux within the time order accuracy that I defined for my system.

CHAIRMAN WILLIS: Okay.

DR. MAHAFFY: Within first order accuracy in time, that's the right flux.

CHAIRMAN WILLIS: Well, we can look at this again another time.

DR. MAHAFFY: Yes.

CHAIRMAN WILLIS: But, again, we are not really the quality control on this TRAC code. We can't get into every little detail.

DR. MAHAFFY: We can show you mass edits, okay.

MR. BANERJEE: The only thing I have to ask is that this is a very standard procedure because you can it in many ways. You solve the pressure Poisson equation and you can correct the velocity field based on that Poisson equation and you can satisfy that equation.

DR. MAHAFFY: Yes.

MR. BANERJEE: I mean, I can do it ten thousand different ways.

DR. MAHAFFY: Yes.

MR. BANERJEE: Okay. The problem that arises is that if you have two such equations and you have only one pressure Poisson equation, you're trying to drive this thing to conserve mass in the gas and the liquid. The question I have for you is I can find a way to do it for the mixture. I can do it for the volume. But I can't drive one Poisson equation for the --

DR. MAHAFFY: The answer to your question is that I'm not really solving one equation.

MR. BANERJEE: Oh.

DR. MAHAFFY: I isolate one equation. It's all algebra. The numerical method is structured in a way that I can insolate a single pressure equation, but in doing so I still have a set of equations in this case that involve temperatures for my single phase flow, and I've got to come to deal with those, too. But that's a back substitution step. After I've solved for my pressures, I've got another set of equations that now get my temperature changes correct, too.

MR. BANERJEE: But the temperature has a

very small effect on a liquid density, for example, right? So it's going to be a very stiff system.

DR. MAHAFFY: It is a stiff system if it's all liquid.

MR. BANERJEE: Yes.

DR. MAHAFFY: But it does turn out to solve it quite well. It's better than the density based method for a purely liquid system.

MR. BANERJEE: But what you're really saying is through the temperature you're adjusting the volume fraction then in some way?

DR. MAHAFFY: Well, in two phase flow, okay, this list of variables now increases. I've got pressure. I've got two temperatures and I've got a void fraction as an unknown that I actually have to go through the solution.

MR. BANERJEE: And this iteration you do has it involved all of those, like the void fraction--

DR. MAHAFFY: Yes, they're all there. I'm solving for all of those.

MR. BANERJEE: My God, how does it converge?

DR. MAHAFFY: It's a Newton method and it does a pretty good job of converging. Convergence is not one of our big problems with this code in general.

MR. BANERJEE: So you integrate on the -- to keep it simple if it was an isothermal system, you would iterate on the pressure and the volume fraction? I mean, the pressure comes up because it's all a Poisson equation effectively. And then you iterate on the volume fraction, keeping the temperatures out of it right now?

DR. MAHAFFY: No, they're all iterated together.

MR. BANERJEE: Let's say it's isothermal--

DR. MAHAFFY: Well, it's isothermal then I'm actually going to get at iteration, I am simultaneously altering my pressures and my void fractions --

MR. BANERJEE: Right.

DR. MAHAFFY: -- in a way that's consistent with a pure Newton solution of this set of algebraic equations.

MR. BANERJEE: Do you have that written down somewhere?

DR. MAHAFFY: Yes, it's here. It's in the TRACE manual. There are probably some papers laying around.

MR. BANERJEE: I'd be very interested in seeing that.

CHAIRMAN WILLIS: That's why you have the reduction of the equations coming in --

DR. MAHAFFY: Yes.

CHAIRMAN WILLIS: That's all part of this whole thing?

DR. MAHAFFY: That's all in there.

MR. BANERJEE: Okay.

CHAIRMAN WILLIS: Your homework assignment is to read that manual.

DR. MAHAFFY: If you read this set of viewgraphs, it's all there in a condensed form. And I'd be happy to go through it in a more detailed for you.

MR. BANERJEE: Is it in detail in the manual, there's no problem.

DR. MAHAFFY: Yes, it's in the manual, too.

CHAIRMAN WILLIS: It's all based on assuming that these little boxes can be modeled with these different equations.

DR. MAHAFFY: Yes. Okay.

CHAIRMAN WILLIS: Some of them can.

DR. MAHAFFY: And again, you know all this is saying is the difference in terms of predecessor codes is that we're iterating RELAP5 is doing exactly the same thing we are during the first iteration. We just go ahead and do a relinearization and come back until it converges to some criteria. And so what we're missing, RELAP5 has mass conservation error checks to adjust time step size. That's going to be directly one of our checks, although you can pick it up through checks on residuals that normally don't get activated. You've got to be a developer to get that going right now. But that's what's going on now.

If somebody wanted to do a RELAP5ish reproduction of the calculation, what you would do in TRACE is that you would relax the convergence criteria to the point that it was nonsense; instead of one in a million, you'd do a million parts in one for your convergence ratio. But you would also have to activate some kind of a time set control in the code to capture the capabilities of RELAP5 to do the single shot linearization and still do a decent job on mass conservation.

We've talked about some of the implicit things that are going on.

Mesh topology.

DR. FORD: Before you go onto that.

DR. MAHAFFY: Yes.

DR. FORD: I wish someone would calibrate me. I'm hearing all these detailed questions from the experts with calling into account or questioning the base of this whole methodology. And we're very late into the development. Are we in a series "oh heck" situation or is this just academic nuances?

MR. BANERJEE: I think the "oh heck" question is with regard to the one dimensional description of a system which has strong three dimensional characteristics in places. So it's really hard to get anything sensible and those regions are of a one dimensional region.

In a pipe I don't think it matters very much, you know, what we're doing --

DR. FORD: Well, I heard --

MR. BANERJEE: In a lower plenum it might, you know.

CHAIRMAN WILLIS: Well, I could try to answer your question here. I mean, it somewhat of an insider looks at this thing and says "Gee, wiz, how can they make all these other assumptions and shortcuts and yet get answers which are good?"

DR. MAHAFFY: Right.

CHAIRMAN WILLIS: And that's never really been answered. The answer always comes back "Well, we've used these things for many years and we figured out to make them work and we're getting good enough answers for nuclear regulation."

But the problem then that the outsider has, or maybe the ACRS members, okay I can accept that this is sort of experience of 30 years that you can use this thing with all its faults for this problem, because you've assessed it and you've compared it with all kinds of data and so on, it's good enough in spite of all kinds of things. But when you use it on a new problem, we have no idea.

DR. FORD: Well, I heard one physical scenario which I can understand, this question of bubbles in pipes. And it's a no nevermind under normal situations. But suppose you have an accident situation where a pipe is rapidly accelerated? It is not, if I understand your argument, covered by not taking into account this thing that's in RELAP5. Is that a "so what?"

CHAIRMAN WILLIS: There might be cases where if you had a spacecraft and you had a bubble in the gas tank and you wanted to know where the bubble is and does it get near the offtake pipe and so on, it might be really critical.

DR. FORD: Well, what about a nuclear reactor which is under severe accident?

CHAIRMAN WILLIS: Probably not.

DR. FORD: Not a problem? I'm asking for calibration here.

CHAIRMAN WILLIS: That's where the staff has to very experienced and wise, and curious to probe these weaknesses and are they sensitive -- is the decisions they're trying to make sensitive to these weaknesses in the code. You cannot have someone who blindly uses a code developed by somebody else with no idea about its weaknesses and then tries to reach really good conclusions.

DR. FORD: I can understand what you're saying, Graham. I'm just trying to get calibrated myself as to where we -- do we have a weakness here.

DR. SIEBER: That's where all the testing programs come into play.

DR. FORD: Tomorrow afternoon.

DR. SIEBER: You go and model a new phenomenon for your new plant, and you get a result. And you ask yourself the questions is this result reflect actually what would happen in your plant for those issues that you are unsure about that differ from the one that you know about, you go and test that.

CHAIRMAN WILLIS: Your battery is running out.

DR. SIEBER: This battery is off.

CHAIRMAN WILLIS: It's is pacemaker.

MR. KELLY: If I could jump in for a quick comment. This is Joe Kelly.

On a couple of occasions, Professor Wallace has said something about he used this code for a new application, can you believe it. Well, absolutely not.

DR. SIEBER: Yes, right.

MR. KELLY: Okay. And that can't be emphasized strongly enough. If you are going to use any of these codes in a new application, the very first thing you have to do is take a look at it, figure out what's important and assess for the things that are important over the range of parameters for which they're important.

DR. SIEBER: That's right.

MR. KELLY: And you have to do that structured kind of assessment before you believe any answer from any of these codes.

DR. KRESS: And your question of severe accidents, it's not applicable to severe accidents. We've got so many uncertainties in the severe accident field that it just swamps these kinds of uncertainties. And they're taken care of some other way.

DR. MAHAFFY: But everything we do in the business of simulation involves approximations.

CHAIRMAN WILLIS: That's right.

DR. MAHAFFY: If I solve the Navier-Stokes equations with, you know, as much dimensionality as I want, I still got approximations in my turbulence models, I got approximations due to the fact that it's not the Boltzmann equation. I go to a Boltzmann equation, I've got approximations on my force fields and potential functions. And it goes on.

And what's been said about the assessment is the bottom line. Anytime you're simulating any system, you have to establish an assessment base and see if your approximations are good enough.

DR. RANDOM: And another way of saying this, these tools are not predictive. You cannot predict what's going to happen beyond the database that they were build on.

DR. SIEBER: That's right.

CHAIRMAN WILLIS: Which if you had a reactor look like this one, this is a backup slide or something?

DR. MAHAFFY: No. We're into mesh

conductivity now and --

CHAIRMAN WILLIS: But, you see, there's some false physics about what you're allowing the code to do here, which under some circumstances could be quite important.

DR. MAHAFFY: Yes. This is a separate issue that I was asked to address, and I want to do this with just a few slides.

CHAIRMAN WILLIS: Could you just take four minutes?

DR. MAHAFFY: I can do that. And let's do this in two, and two for my last presentation and we'll be done with it.

Basically, it's my belief and what we need to do is have Vic sit down and review the documentation in detail, but we have captured absolutely all of the mesh conductivity that was available in RELAP5 with TRACE. That was one of the things we were forced to do. It was a good idea. But you're absolutely right. There's all kinds of stuff here.

You can see my little notation here. If I've got a side junction, it's not going to move momentum across here correctly. If I want to do that, what I need to do is turn on my 3-D mesh. But these are conductivity issues that come up in RELAP5, and they're there. So we could reproduce archival decks and run problems that people have run before.

This is a zipper that's in RELAP5. I can put a cross connection between two parallel pipes.

CHAIRMAN WILLIS: So it's getting closer to CFD in that case?

DR. MAHAFFY: Well, you know this was RELAP5's way of dealing with two and three dimensional geometries before you had RELAP5 3-D or whatever it's called now. You know, we reproduced this to be able to use archival decks; that's all. We tell the users use the three dimensional geometry if this is important. Okay.

We have some troubles now we have to deal with from plena with the branch component. Right now we only catch two of the three really straight on through and the other guys at a right angle.

CHAIRMAN WILLIS: These are all the problems with the other code, particularly with the momentum.

DR. MAHAFFY: Again, see my answer is if that's important, you're going to go to a two dimensional or a three dimensional calculation representation, which you can get. It's there. This is just conductivity issues.

Also, we take care of the gravity in RELAP5 and TRACland.

CHAIRMAN WILLIS: I have a comment on the bends. In bends there's a centrifugal force which can often be much bigger than gravity. So the bend is behaving like a horizontal pipe enhanced gravity, and yet it's never taken into account.

DR. MAHAFFY: You have to do that through your lost coefficients.

CHAIRMAN WILLIS: Because there's a physics there which isn't captured just by lost codes.

DR. RANDOM: Well, just on the one you show, one is a two node problem the other one a three node problem. How do you map those back and forth?

DR. MAHAFFY: How do you map those back and forth? We don't. All right. We treat -- the internal of the code, it knows whether it's in RELAP geometry or whether it's in TRAC geometry.

Really internal to the code. The way we're treating this is that RELAP5 has basically an elevation change across the link of a cell. That's how it works.

DR. RANDOM: Right. Right.

DR. MAHAFFY: Okay. In old TRACland what

there was was an angle of a given face to the vertical. That's why I draw this picture.

DR. RANDOM: It's a junction property.

DR. MAHAFFY: It's a junction property.

DR. RANDOM: Right.

DR. MAHAFFY: What happens in TRACE is that we have separate elevation changes for each half cell internally. So I actually can do funny combinations of this. I can tell you what the elevation change is from this face to this center, and I can give you a separate elevation change from this center to this edge. And I can preserve my gravitational handprint --

DR. RANDOM: In TRACE you actually can do the problem on the left?

DR. MAHAFFY: That's right.

DR. RANDOM: Okay.

DR. MAHAFFY: When you read in a RELAP5 deck, it's engaged the elevation changes in the right way in TRACE so that it captures exactly that model. Okay. And that's all I really want to say.

You know, we've got some stuff on component models you can look at. We can talk to you about them.

CHAIRMAN WILLIS: This is all in the CDs

that we have?

DR. MAHAFFY: Yes.

CHAIRMAN WILLIS: All right.

DR. MAHAFFY: Coupling to other codes, I'm out of time. We talked about this before. We talked about. But basically we're doing multi-tasking, we're trying to bring lots of tools to bear on a complicated system simulation. We've tried to provide an interface that allows you to plug in a lot of different simulation tools for TRACE, a CFD code, CONTAIN is already there, REMIX is there. A picture of maybe multi-processes.

The ECI is just a name for our interface that does it. It's a convenient way of plugging things together.

We don't have time to talk about the details. I can talk to you about it --

CHAIRMAN WILLIS: I think you've done a good job, though, of sort of going through all this stuff and if we're interested in the details, we can look them up.

DR. MAHAFFY: Yes.

CHAIRMAN WILLIS: So it's very helpful.

DR. MAHAFFY: Okay. Then that's all --

CHAIRMAN WILLIS: Is that in this mesh

topology?

DR. MAHAFFY: There is a separate one with a label for --

CHAIRMAN WILLIS: Of what?

DR. MAHAFFY: Coupling.

CHAIRMAN WILLIS: There's a new handout called coupling.

DR. MAHAFFY: There's a lot on our website also that will tell you about this thing. And I'm available.

CHAIRMAN WILLIS: Thank you, John. It's very helpful.

We'll take a break now from 12:30 to 1:30 for lunch.

We're off the record.

(Whereupon, at 12:30 p.m. the Subcommittee was adjourned, to reconvene this same day at 1:33 p.m.)

A-F-T-E-R-N-O-O-N S-E-S-S-I-O-N

CHAIRMAN WILLIS: Okay. Come back into session please. We're looking forward to hearing about SNAP. The presenter will be Ken Jones from Applies Programming Technology.

MR. JONES: We'll just be giving a quick design overview of the SNAP and all the pieces that make up this package.

Then I'll be talking about the redesign work we did on the model as over the last two years we've done a lot of redesign on the model from the ground up.

I'll touch on the current ongoing efforts, our future plans and then I'll go into a demo of the ModelEditor.

This is the overall system architecture. This is all written in Java with the exception with the AcGrace, which is Legacy plotting package. This is a combination of C and C++.

There's four areas of SNAP. The access control portion is -- for distributed calculations. It's a Java application that wraps a relationship database, but it's a commercial relational database, either ORACLE, SYBASE or PostgreSql can all be used.

There's a database administrative tool

that talks with this Java application, so it allows you to maintain the system without having a database administrator that knows a lot about database.

There's a file archival portion to that.

There's a free processor that consists of this ModelEditor and there's also a jEdit plugin for ASCII files. And there's also a command guide line guide submittal for submitting jobs to the runtime.

The runtime consists of two servers. There's a databank server, which is a job application that wraps the NRC databank, the service's request from the client applications for experimental data. From the client applications you can list of the experiments list of data channels within that experiment and then actually access the data.

There's a calculation server which wraps each one of these analysis codes. All the communication with the analysis code is through the calculation server. So the client applications don't need to know the specifics of how to read the RELAP5 or TRACE files. That's all encapsulated inside the calculation server.

We've used this plugin architecture extensively. This allows us to create an API which gives us a standard interface between the analysis code and the server -- the main core portion of the application. This way we can easily add an additional analysis code. I'll get into that in a little more detail.

There's a job status application. This runs -- it's really a utility application that will give a list of all the runs that are running on a specific server that will allow you to look at the ASCII output from a calculation and also plot the data from job status using the AcGrace post-processor.

The two post-processing applications are beta. The visual engineering data analyzer. This gives us a two dimensional and three dimensional graphic of the data being calculated by the calculations as well as plot capability through AcGrace.

CHAIRMAN WILLIS: Now this NRC databank, that's all the experiments that have been run that might possibly want to be analyzed?

MR. JONES: Exactly. It's a collection of all the experimental data.

CHAIRMAN WILLIS: It has all the description of the systems and everything in their right form so it can be just plugged in?

MR. JONES: There's a standard file

architecture for each of these files that the data transfers into. So it has a list of -- for each experiment there's a set of files. And each file has header information to identify what all those data channels are that are contained in the file.

CHAIRMAN WILLIS: You search for the files or is there a search engine for them, or what?

MR. JONES: It goes through a tree structure. I can't demonstrate it here because I'm not connected to the network, but --

MR. BANERJEE: There's not tags or how does it find the data?

MR. JONES: The databank itself is really structured with just a directory structure of experiments.

MR. BANERJEE: Oh, I see.

MR. JONES: Okay. The job application that wraps it reads that directory structure so you can query it from the post-processing application to find out what all those different experiments are.

MR. BANERJEE: You got a key word query or how do you find it, the problems of experiments?

DR. MAHAFFY: Well it's in the tree structure so you can by facility and such.

MR. CARUSO: Well, I think Dr. Wallis'

question was, though, and at the input for modeling the experiments. For example, the database has LOFT experimental data in it. Does it have the geometric and physical data for the LOFT facility that would allow someone to create a new LOFT model, or does it just have the pressures and temperatures and flows that were measured during the experiment?

MR. JONES: Just the measured experimental data.

CHAIRMAN WILLIS: So you have to create the model for the system somehow?

MR. JONES: If you wanted to, yes.

CHAIRMAN WILLIS: Nodalize LOFT in some new way, you have to start from the beginning?

MR. JONES: Well, you'd start with an existing LOFT model.

CHAIRMAN WILLIS: That's in the databank or is that just from --

MR. JONES: No, that's not in the databank. That would be in an existing analysis.

CHAIRMAN WILLIS: So there's another --

MR. JONES: Yes, this server just gives us an easy way of accessing experimental data so we can bring them in --

CHAIRMAN WILLIS: You have an electronic

description of the facility in which the tests were run.

MR. JONES: Yes, that's right. This isn't really meant to store that type of data. The runtimes really are just for transient data for analysis.

CHAIRMAN WILLIS: I would think it would go with the data set. The databank's no good at all until it's keyed to an electronic description of a physical system.

MR. JONES: You know, it may have some of that in it.

MR. STAUDENMEIER: I mean, we have a catalog of input decks for different test facilities. So you would look up in some sort of --

CHAIRMAN WILLIS: But you've got to take them and load them into this somehow or --

MR. STAUDENMEIER: Yes, you have to take them and load them in. It can't read your mind yet as to what one you want in.

CHAIRMAN WILLIS: So there's not a databank of those things the way there is for the data?

MR. STAUDENMEIER: There's a set of decks, they had haven't put in -- actually, there are some facility decks in the databank. It's not all our decks, but there are some test facility decks in the databank for different things which it may or may not do -- it would have to be modified. There may be a base deck there that you'd have to modify for the specific test that you want to run. But you have to load it in my hand. The data experiment and the input deck aren't one package. You have to get them separate. Bring them in separately. That's I guess, a possible thing you could have in the databank when you click on the data, you could also have another hyperlink that appears right there to the input deck that you have. And that's something --

MR. MURRAY: Well, that's in the databank now. You can view this. There's an internal website that you can go to, to res5nrc.gov that shows this whole structure and how things are organized.

So each experiment is cross referenced by, you know, you can start off just say do a separate effects test and it takes you through a list of those. And you can move down in various different ways.

That databank is storing the input decks for particular facilities if they exist. The data and usually electronic reports that pertain to that facility or at least a bibliography for the reports if they're not in a nice electronic format.

Now, how you integrate it with what's Ken has got right now --

CHAIRMAN WILLIS: Well the data points taken in a lot of test, and they're not necessarily at the same place you have a node in your computer model, are they?

MR. JONES: Right. Usually the input decks are set up with instrumentation in mind. Someone that set up the deck was aware of the instrumentation for at least the test he wanted to run and what the key measurements were. But there are in some of these reports that are attached to the data, there are P&ID drawings or facilities --

CHAIRMAN WILLIS: What if you wanted to run a LOFT test with your computer, and you wanted have the node size or something, would it take a thousand years to recast everything in terms of this new node size and relate it to the physical geometry?

MR. JONES: The purpose of the model is to be able to automate that test, the renodalizing existing model. But in this case we'd start with an existing nodalization and then just use the ModelEditor to, say, split every node in a pipe in half, if you wanted to do that. And I'll demonstrate some of that in a couple of minutes.

MR. BANERJEE: Can I ask you about the visual engineering data analyzer. Is it using some sort of packages which --

MR. JONES: It uses the Java 2-D and Java-3D APIs for visualization. It's based on the JavaBean architecture. So not only do we have a plugin API for codes, but we also have a JavaBean architecture that allows additional visual elements to be added without changing the core of the --

MR. BANERJEE: Can it do a sort of rendering of surfaces and things, or what does it do?

MR. JONES: Well, I'll demonstrate it in just a few minutes. But what it will give you a 2-D or 3-D animation of the calculation, either while the calculation is running, you can run it like in an MPA type mode where you're interesting directly with the calculation or you can just run it in a replay mode where you're looking at the calculation as it's running.

It also allows you to connect to multiple calculation servers and multiple calculations at the same time. So in the post-processor you can bring in multiple data sources and animate them in the same display. It will take care of that interpolation on the time frame so the data's consistent.

DR. RANDOM: How much of this is code that you've written versus some of it available as commercial software?

MR. JONES: Well, AcGrace was modified. You know, that's a modified version of Grace. About half of it is original Grace code.

DR. RANDOM: Purchased it and then --

MR. JONES: No, this is an open source.

DR. RANDOM: Oh, it's open source. What about the top, this SNAP database server?

MR. JONES: Of course, these are commercial packages. PostgreSql is an open source database. ORACLE and SYBASE as, of course, commercial databases. The rest of it's all written with --

DR. RANDOM: Okay.

MR. JONES: There is a JEdit open source editor that we use, that we have our own plugin for.

There's also a configuration tool here that's used to set global data. You can also use it to start and stop the server application.

About two years ago we started a redesign of the ModelEditor portion. What we really worked on was trying to provide a consistent and extensible infrastructure. The original design was largely hard wired for RELAP5. It was more of a proof of concept for a ModelEditor. It really wasn't extensible for adding these additional codes.

There are also redundant data layers, which was confusing for the analyst because he had to enter data in both a physical layer and a model layer. And a lot of times there were conflicts between that information.

Now, when we had started on this, the work on beta had already been done. We had a working 2-D/3-D package. We pulled the drawing routines out of there. They were cleaner, and we've added those to the ModelEditor.

We changed the fundamental way data is structured inside the ModelEditor. Originally it was a single view ModelEditor. If anybody's seen it a couple of years ago, basically you had one view of the data. And all the data for the elements was tied directly to the visual elements on the screen. We've broken that connection so the data is stored in its own internal database. This way we can work this multiple view design.

It also allowed us to implement a consistent undo/redo architecture.

The original design had restart capability, but it didn't keep track of how the changes were made from when the run was actually submitted. We've gone through and added the restart capability so that when we generate a restart deck, it only includes the modified components.

We've gone through a reworked all the editing dialogues. A lot of them were originally hand coded. We've redone in using IDE so they're largely generated.

We've switched to a plugin design, and I'll get into that in a little bit more detail. But basically you don't have to modify any of the core to add an additional code. It reduces the overall maintenance on the software.

MR. BANERJEE: When you say there was a beta version available to you, where did that come from?

MR. JONES: The ModelEditor portion was originally developed by the ISL group. We started working on it two years ago.

MR. BANERJEE: And they did it for RELAP primarily, is that it?

MR. JONES: Yes, it worked for RELAP at that time. There was some TRAC work that had been done on it. But it really needed redesign to push it forward.

MR. BANERJEE: How long did they work on it?

MR. JONES: Several years.

MR. BANERJEE: Was there quite a bit of that that you could use or --

MR. JONES: We tried to use as much as we could, but there was a lot that we really had to -- you know, we tried to keep like the IO code as much as we could. But a lot of the dialogues had to be replace. And most of the infrastructure really had to be replaced. But instead of starting over and writing it from scratch, we just incrementally refactored the code continuously until we came up with a new design.

Chester, you want to --

MR. GINGRICH: Yes. This is Chester Gingrich at the -- I'm the project manager for this code.

We were able to save a lot of the base coding and the RELAP, most of the RELAP import and export capabilities were saved from the ISL version.

MR. JONES: Okay. The new architecture let us move into this multi-view design, which I'll show.

And one of the other things we've added is a patch processing capability.

The plugin design, the whole idea is we want to make this an extensible architecture so that we don't have to modify any of the base portion of the code. So we take all the analysis specific classes; all the IO, the dialogues, that type of information and we move it into a plugin portion.

You have a clearly defined interface that has to be supported in order to implement this plugin.

And the whole key to this is a standard API that we publish that anyone can pick up and just add their own plugins following the API.

Now, we have two types of plugins. There's an analysis code plugin which defines an analysis code, of course. And then there's also a feature plugin. Someone's interested in adding a new feature to the SNAP package, they can implement a feature plugin and that option will be a, you know, they can options to the menus and batch commands and things like that through this feature plugin interface.

Right now we have analysis code plugins for TRACE and RELAP5. The CONTAIN one is under development. It should be ready by the end of the year. And there's a RELAP5 3-D plugin which is really a --

DR. RANDOM: Whose doing the RELAP5 3-D?

MR. JONES: We're doing the RELAP5 3-D.

DR. RANDOM: You're doing it yourself?

MR. JONES: That's right.

DR. RANDOM: The NRC's not funding it?

MR. JONES: That's right.

DR. RANDOM: Pardon?

MR. JONES: That's correct.

DR. RANDOM: Yes. I'm just envisioning that DOE applications and --

MR. JONES: Beddes is interested in the RELAP5 3-D.

DR. RANDOM: Yes. All right.

CHAIRMAN WILLIS: Is RELAP5 3-D going to be better than TRACE?

MR. JONES: I can't really answer that. I don't think so.

CHAIRMAN WILLIS: It seems rather odd you're developing two, two plugins here for --

MR. JONES: Well, the RELAP5 3-D is really just an add-on. We call it a piggyback plugin which modifies the RELAP5 plugin. So if you have a slightly different version of the code, we can use this approach to support, you know, slightly different --

CHAIRMAN WILLIS: It doesn't mean three

dimensional, does it?

MR. JONES: It has a 3-D --

CHAIRMAN WILLIS: It's much more complicated than RELAP5 itself?

DR. RANDOM: Well, it's an -- original RELAP5, so you have a 3-D vessel and you have multiple 3-D vessels.

MR. JONES: Most of the components are identical as far as the inputs.

DR. RANDOM: I think the inputs are virtually the same, right?

MR. JONES: Pardon?

DR. RANDOM: Only minor differences in the input, I guess, to accommodate --

MR. JONES: Exactly.

DR. RANDOM: -- some of these 3-D features.

MR. JONES: That's right.

DR. RANDOM: In fact, I think Wagner designed in the ability to put in the 3-D input into the original codes, even though it wasn't used.

MR. JONES: Right.

We have a RELAP5 to TRACE conversion, which is done as one of these feature plugins. There's also a display generator that is under development, which will take the original displays that are done in the ModelEditor and generate them for the post-processor.

This is a picture of the new multi-view design. It follows a SDI design approach, so we have separate windows here for the different application, different portion of the application.

This is the main window here we you can create models, all the import and exportive files is done through these menus up here. You can submit calculations from here. You can them out, and that kind of thing.

We've moved to this component navigator approach here, which is a reflection of the internal database in the ModelEditor. You can open multiple models here. And this will give you a listing of all the components that are in the model categorized by type. You can edit these components directly from here.

And you could also create views from the navigator. You can generate different views of the data.

The idea is we moved away from the single view of the data to this internal database structure that lets us generate different ways of looking at the same information.

There's a little message window down here that keeps a log of any messages that are coming through. If the message is tied to a component, if you click on it, it'll select up herein the navigator. There's a convenience features like that.

MR. CARUSO: I'm just curious. You have a Westinghouse flow loop model but it's calling up the steam tables with deuterium oxide.

MR. JONES: Those just automatically get loaded, the d2o and the first ones are the h2o.

MR. CARUSO: Okay. You're not modeling heavy water, Westinghouse PWR?

MR. JONES: They just automatically get loaded.

DR. SIEBER: There's a little heavy water.

MR. JONES: Yes.

DR. RANDOM: Out of curiosity, will this work with the ATHEANA code?

MR. JONES: That's a good question. We haven't --

DR. RANDOM: It's just sort of a subpart of the RELAP5 3-d?

MR. JONES: It wouldn't be difficult to

have that plugin to support it, or any other code.

We also have these ASCII views, which will give you the ASCII representation of the input deck for anyone of the components it's selected.

It uses a component navigator. Again, here's the model node. You can have multiple models open at the same time. There's main categories, hydraulic components, for example to list all the types for a giv4en model types.

All these categories are determined from the plugin, not from anything in the ModelEditor itself.

Here we've expanded the pumps node. This lists, there two pumps in this one. Here it will show you what all the heat connections to that pump.

And the whole idea is we're trying to provide a logical representation of the model and it'll display all the data through this navigator component.

Down here there's also these views. It's a different type of node. The views themselves are handled as components in the ModelEditor This lets us create a drill down type capability so you can treat a view as a component in another view. Now, I'll demonstrate that in a couple of minutes.

We've also added user defined constants and equations. We have a Python interpreter that will read this set of user defined constants and variables that the user can set up Python equation and run through that and generate data that can then be brought into anyone of these dialogues.

Okay. These are 2-D views. Okay. So they can contain any kind of component type. The original model had one view with three different layers. You can only look at the control systems or the hydro or the heat structure. Layered here you can mix and match the data anyway you want in a view to create a data that the analyst is interested in.

It also has annotation capability. So you can put in these labels and pictures and that kind of information can all be inserted into the views.

And you can either edit the components. And you can either edit the components off the navigator nodes or you could edit them directly from these views.

Anyone of these components also can be displayed in as many views as you want. So, for instance, this secondary side of the steam generator, you may want to show it over here in one view and because you're just showing the steam generator, you may want to include that same component in another view. Since all the data is tied to a single database and then any modifications in one view will be automatically reflected in the other.

DR. RANDOM: Now, are those views the user has to build this view, site the arrangement?

MR. JONES: Well, we tried to get it as close as we can.

DR. RANDOM: In secondary, you know elevation would be lined up or as what you've shown in there, but I'm wondering did the user do that or just the code?

MR. JONES: This is a user going in and laying these out. But we try to get as close as we can, but it's kind of a difficult process trying to lay that. It's one of the areas we'd like to improve, just the initial layout.

Okay. We've gone through and cleaned up a lot of the dialogues.

There's a vessel dialogue where we're adding flow area fractions. Here we're showing the table. And as you select components in the table, this will highlight the cross section view. This is tapped down. So this showing which cells are selected. It's really the intersection of the selected region on the side view and the top down view.

And you can either select over here and have it highlight in the table, or vice versa. Either one will be consistent back and forth.

This is a 3-D vessel. Again, we're using a Java 3-D representation of the data here. As you select things over here you'll be visually see exactly what levels are being selected.

DR. RANDOM: Can you pick it up and rotate it and change the perspective.

MR. JONES: Yes. I'll demonstrate that when I get to that point.

Okay. We also have this ASCII view. On all the components there's a show ASCII menu item. All these views, including the ASCII view are modeled as listeners in the code. So any changes to the component's data is automatically reflected in these views. So if you go into the dialogue and change the dx, this window automatically updates the current data.

We do syntax highlighting. It's kind of hard to see up there, but it's color coded with comments and that kind of thing.

Right now we're supporting TRACE only.

Joe's looking on the RELAP5

MR. STAUDENMEIER: It's ready to go in.

MR. JONES: Yes. That will be in this week, so that's their next release.

We also have reference documentation. We have this pdf viewer, it exports HTLM or pdf. And off any of the components you can hyperlink directly into the users manual. So from the pumps node if you go to "Reference Docs" there's a list of different documentations that's available. Right now it's only set up for TRACE, and it's only set up for the users manual. But it's designed so that we can add as many different references as we want. So if we want to -- you have a hyperlink from the component to some document -- you know the theory manual, say, section on the pump, we can do that.

We'd also like to extend that later so that the user can define their own document and reference this for a specific component.

DR. RANDOM: Are you going to talk about input diagnostics and what SNAP will do in terms of --

MR. JONES: Sure. Sure. Yes, I'll do that in the demo portion.

We have these user defined constants, variables and functions. Either the constants or the variables can be used in the editing dialogues anywhere that we're itering a floating point value. It doesn't support the energy data fields right now, just the floating point.

These values can also include the engineering unit, so if you change between SI and British it will automatically update the calculation accordingly.

You can set up a parametric constant to create a series of calculations so that when you generate the ASCII deck and submit the run you select one of the constants as a parametric. It'll vary that over the specified range and create one model for each case in the parametric set.

DR. RANDOM: How hard is it going to be in the future if you decide to use this nonparametric statistical approach to assessing the uncertainty associated with the code calculation where some of the things that you want to change will be actually things that, presumably in the correlations that are built into the code, like your heat transfer coefficient, the friction coefficient or multipliers on those and then use monte carlo techniques to select the set and then do a run and continue in that process? And then analyze the final set of runs statistically?

MR. JONES: The systems really designed to really facilitate being able to do those type of calculations. We want to be able to do batched runs with this approach so that you can vary your initial set of constants and then --

DR. RANDOM: Well, some of these constants may not appear, you know, in the usual way. They're not going to be links or diameters, or volumes or --

MR. JONES: Well, the only thing we're changing are inputs to the code request --

DR. RANDOM: Right. Most of the things I'm talking about you'd be having to go into the code and actually change the numerical values or put multipliers on them, or something like that.

MR. JONES: In that case we'd really have to have had more code input that would allow you to change that --

DR. RANDOM: It's up to the NRC to decide if they want that capability in time.

MR. JONES: Sure.

MR. STAUDENMEIER: Actually, this is just on my -- there's a limited set of multipliers that you can define for input already. And that could easily be expanded or have an ECI type or some kind of communication where SNAP can have a set, a whole set that you control and vary over a given range. They could become variables instead of constants in the code that SNAP is able to modify.

MR. JONES: Okay. I talked about the pipe, but I'll show that in the demo.

Okay. We have these user defined functions. This is just an example of a user defined function here where these are all inputs on this side. So the number of rods per row, pellet radius, clad radius, inside and outside diameter. You take these inputs, feed through this Python calculation and they'll generate these calculated values.

Now, any of the initial constants for the calculated values can then be brought into the dialogue. And if we change any of these inputs, of course, it goes through automatically and calculates new data for the calculation.

This is also demonstrating some of the ability to create your own user defined pictures, and you can annotate the views. So that when you come back to this in six months or a year, you can really see what all these inputs are and the user can -- basically has this drawing capability to annotate their calculation.

MR. CARUSO: Can you model all the new,

the currently extant BWR fuel types with --

MR. JONES: We have that in there. We have the new BWR channel model in SNAP.

MR. CARUSO: Length rods and --

MR. JONES: Length rods. It hasn't been well tested. I know we have some issues with it. That's one of the things we'll be working on over the next month.

DR. RANDOM: You can accommodate things like good spacers and blockage associated with those?

MR. JONES: We model everything that's in the --

DR. RANDOM: The can?

MR. JONES: That comes through the model or through the code.

DR. RANDOM: Input, I guess, huh?

MR. JONES: Yes. Anything in the input can be modeled.

MR. CARUSO: Can you do the siemens -- no, not siemens. ABB subchannels, BWRs that are subdivided into subchannels?

MR. JONES: I believe so.

MR. STAUDENMEIER: Yes. This is Joe Staudenmeier.

TRACE has been modified, it's an advanced channel now that can model all the events BWR fuel geometries. And I don't know if SNAP has support for all of those yet, but it will have support for all of those.

MR. JONES: This is our ongoing efforts.

We do have support for all that right now. It needs to be tested. I know there's some issues with it.

We're currently working on this CONTAIN plugin which follows a different design pattern than the RELAP and TRACE based plugins. Those used user -- well the developer had to create the dialogues for each one of the components in that case. This new JavaBean architecture will automate a lot of that process. So we define what the attributes for each component are and then we're able to reuse a lot of the code for the editor.

This also gives us the ability to do a multi-selection edit. So you'll be able to select multiple pipes and then modify a single attribute for all the pipes at one time, which can't be done with the earlier architecture.

We'll be adding support for the new ECI communications package.

Let's see, the RELAP5 ASCEE viewer, which is going into this next version.

We to add additional renodalization options. Right now in TRACE if the inlet and outlet flow areas for a cell are different, we're using the closet cell areas. We're going to change that to make it a little bit smarter so it can sense abrupt area changes on either the inlet or outlet.

We're continuing with testing and validation. We have a set of 970 TRACE calculations that we use for testing. At this point we're able to import all of them and generate ASCII decks that get through input processing on TRACE for all the cases except for 64. And that works continuing on that.

We usually put out a new version in every one to two weeks. It's available on a website. You can automatically update the application through the website.

So our future plans, we want to provide tighter integration between the pre and post-processor. Since we're sharing a lot of that code, a lot of the drawing code between the ModelEditor and data, we want to be able to automatically create this -- we want to be able to use those displays from the ModelEditor directly in the post-processors so the analyst doesn't have to create the same drawing twice.

We're adding the MELCOR and the PARCS codes.

We want to be able to add more advanced copy/paste functions. We want to be able to do a scaled copy of a loop. So if you copy -- and we want to be able to select an entire loop and put in a scale factor and say .5 would scale everything, all the cross sections by .5 appropriately.

We want to be able to add component libraries so that you can just go out and select the type of component you want, pull in from your model.

Along the same lines, we want to be able to support reference models so that you can go to the database and check your existing nodalization against a reference model and revert your nodalization back from that reference model.

MR. BANERJEE: What does AcGrace do there?

MR. JONES: It's a 2-D plotting package. It used to be XMGR5. AcGrace is a new version of that.

We want to leverage the user defined functions and create engineering templates for things like lost coefficient calculation.

We want to add additional RELAP5 and TRACE conversion assistance.

We also want to create these links to user's defined documentation so if inside your model you want to create a link to, let's say these HTML pages in the databank that describe the LOFT facility, you'll be able to create that automatically and the next time you bring them all put, you'd be able to link right to it.

Right now for RELAP5 we can generate a complete 3-D representation that's consistent with all the elevations and that. For TRACE we haven't done that work yet. We can generate a 3-D vessel view, but we don't have the associated piping.

The other thing we want to do is -- I mean the problems we have with AcGrace is that it's an x11 based application. It's written in MOTIF and requires an xserver to be running to be able to use it, which is cumbersome on the PC architecture. There is freeware that you can use to use it, but it's not -- it's difficult to work with and it's a high maintenance application because it has to be compiled on each platform. The rest of the software is all written Java. We just compile it once and the distributed compiled version runs on every platform.

This is the configuration tool right here. And this is used to set all the global information for all the different applications; the location of the Java virtual machine, where the logs are going to written, any calculation servers.

This was originally set up for PDM. That functionality doesn't work right here.

It tells it where the AcGrace executable is located.

We also have web support is done through this application. You can submit a bug report or check for software updates from the website. It'll automatically go out, grab the updates and load them in your machine.

This is the calculation server. It supports -- you know, for each plugin you can define one or more analysis geodes. So if you want to be able to run -- well, you could support multiple versions of TRACE, for example, on a given machine. And when you submit the Java, you just tell it which version you want to run. Go ahead and start that server.

This is the new -- I'm just going to open this. Yes, this is the main application. Window is the component navigator. This is where it lists all the components, and you can go in and directly edit these directly from model. You don't have to go through a view. They're also listed here.

The other thing we can do is look at the reference documentation directly from any one of these nodes. And it's fairly quick. You know, just pops to that section. And that's a fairly good size pdf file that it's working with. The performance is very good right now.

MR. CARUSO: If you wanted to see what the pump curve-- for these pump models, does it pull something like that up?

MR. JONES: It doesn't show the curve. No. It'd be in a table format.

MR. CARUSO: Okay.

MR. JONES: Something like that.

MR. CARUSO: So the user has to develop the table and input it?

MR. JONES: That's right.

MR. CARUSO: Suppose I wanted to enter a negative loss coefficient for some pipe, could I do something like that?

MR. JONES: There's a lot of checking that goes on for each component. In some cases we're checking at the dialogue level, in other cases we check the model. Up there, there's a check model list done before each job submitted that would give a list of the problems.

MR. CARUSO: I just asked about that one because I've seen that one used. And I just wondered if that was something that was still allowed.

MR. JONES: That may be allowed right now, but -- it still allows that. But, yes, it doesn't appear to have a problem with that.

MR. CARUSO: So it can still be created?

MR. JONES: Well, absolutely. But we're at a point where it's easy to add those -- you know, the ability to check different parameters, either on a component level or on the model level.

This is an overview of the component. These are actually just images of different views. And if go down, this is the actual component here or the actual view that has the components in it. And I can go properties here and see the actual properties.

As I select over here, I get the visual feedback so it shows me what I'm editing.

MR. BANERJEE: So do you have these already in the schematic --

MR. JONES: No. These are generated from the input deck. So this reflects the current geometry in the component. So --

MR. BANERJEE: The steam generator's

always generated that same way, right?

MR. JONES: No. This is from imputing an original TRACE deck. And that it'll generate -- you know, it's based on the actual data we have here. So, for example, if I change --

MR. BANERJEE: I guess the shape's is -- you generate your shapes yourself?

MR. JONES: Right, from the geometric data.

MR. BANERJEE: Yes.

MR. JONES: So if I change the data here, I make this segment longer, then you'll see it gets stretched a little bit. And if I were to apply that, it would show up over there that these are drawn to scale.

MR. BANERJEE: Okay.

MR. JONES: By default. You can turn that option on and off.

MR. BANERJEE: So if you got an input deck, say a RELAP deck from someone, it knows how to take it apart and display it?

MR. JONES: Exactly. It's parse the deck, fill out all these components in the navigator. And then you can go in and edit it through the navigator. You can edit through this --

MR. BANERJEE: The are identified in the deck which, say, it's a steam generator something --

MR. JONES: Yes, we read it the same way the codes would read it, and that's part of the plugin that it's being able to parse that ASCII input deck.

Now, the other thing we can look at is the ASCII view of the data for any component. So if I go in here and -- I changed that value again.

MR. BANERJEE: So can you do the reverse, generate a deck from a diagram?

MR. JONES: Yes. Yes. What'll happen is as you make these changes -- in fact, here's a piece of a deck. This ASCII view is the input that goes to the code. Okay. And here I'm going to change this value slightly. And this is one of the dx terms here. So, hopefully, we'll get a change over there. We'll apply it. And then you can see the actual volume changed over there.

And if I were to undo it. See, it pops back to the original value. So you can always take a look at the data here, then again the reference documentation, you can either go to the navigator or you can go directly to the component and look at it. So when you have annotated ASCII version that the users manual, and that's the drawing of it.

One of the other things we can do is renodalize the component. So, for instance, if we want to split this one up into five uniform cells, we can do that. Apply it and you'll see the generated deck is automatically updated accordingly.

And this is an interesting one, too, because all the ones that are shown in this kind of red or pink here have heat structures tied to them.

We know like the pipe it goes in and renodalizes the heat structure at the same time. It does that all automatically for you.

Go back to this overview. Now what you use this for, this gives you the drill down capabilities. So you could have one drawing that is basically your own plant and then these images represent views that you can then go down and drill to.

And this is just the generated picture. If you come into these components and look at its properties, you either create a snapshot which does to the picture and then generates a Raster image of what you have. Or, you can go out and select any kind of Rester image, a jpeg or png, or any of that. You see then it automatically, you know, you can use those images to really annotate how the model's laid out.

DR. RANDOM: In what stage does it check for loop closure?

MR. JONES: It'll check for loop closure on an initial ASCII import and whenever you go to run the model, or if you click up here, this checks the entire model for loop closure. And it went through the loop closure check and it got 1.33 to the minus 5 was a loop closure error, okay?

DR. RANDOM: It tells you what loop that's in?

MR. JONES: No, it doesn't give you any real useful information, other than you got problems.

DR. RANDOM: So the user has to go back and find it someway?

MR. JONES: And the way we do that is under this tools there's a loop check selection option. So if you're in a 2-D view, you can select a group of components and say loop check selection, and it just loop checks the ones that are selected.

DR. RANDOM: And that's my outlining you mean or --

MR. JONES: Yes. The selected components show up in the --

DR. RANDOM: Just in that part, huh?

MR. JONES: Yes, in -- anything that's

selected that has that red box around it. So you can use that loop check selection to zero in on exactly where the problem is in your loop check.

Now, what we want to do is extend that even further and create a loop check audit report that would give you a listing for all the selected components. What's the inlet elevation, the outlet elevation and then connection elevations for that component. And then with that information on the table, it should be really easy to go through and find out exactly where the problems are.

DR. RANDOM: Yes.

MR. JONES: Okay. Just to show that loop check. If I go in here and change the length of one of the -- let's see. Just modify a link and apply it. And now I'll do a loop check. That's the kind of message you see right now. It's saying there's a loop closure failure of 2.85 meters. And it's giving a list of the components that it found the loop.

CHAIRMAN WILLIS: What was it?

MR. CARUSO: Screensaver.

MR. JONES: It's another TRACE sample. This is just a simple pipe. But here we're using these user defined functions to bring data into the model itself. So here we have the diameter, d zero -

d zero diameters and d one is the other diameter.

DR. RANDOM: Do those tampered cells imply that there there's a smooth transition?

MR. JONES: Yes, those would be conical right now. There are no abrupt area change at that point.

DR. RANDOM: Abrupt area shows up and it's just a sudden change in area then?

MR. JONES: No. Right now it's going to use the area of the junction at each location, so you can see it. It would actually going to look like a cone.

CHAIRMAN WILLIS: Just like it does there, huh?

MR. JONES: Right. Right. Yes, we're not using the actual volume to create the drawing.

Okay. In this case we have this user defined variable. Actually we have one user defined variable. The flow areas are calculated. Those are huge invariables. There's two user defined constants. The zero d is one and then there's a -- user defined constant. And those can be either changed over here through the navigator or since these are just additional components, you can drop them right on the display and look at the ASCII. You can come in here and change the value. And you can see the value gets changed in the drawing as well as down in the code here we change the value.

There are also -- they keep track of what engineering units they're in. So if we switch to British, you'll see all the data values are characterized in British units as well as the input deck is all converted to British.

And if we come in and renodalize, say you want to renodalization a set and split each one of those cells into two, again, it just automatically splits it all out for you.

MR. CARUSO: Could you go back to your master, your overall drawing.

MR. JONES: Well, this is a different model. This is just a single pipe model. But I'll be going back to that in just a minute.

Anyway, that's just an example of how you can use these user defined functions.

Another one is -- this is a BWR fuel element. Here we're able to do -- you know, we can just input values and these are the calculated values we're using. This is just an example.

And here's where we enter the code to do these -- type calculations. And it's just a Python interpreter. It has color syntax checking. And it automatically generates this list of input values and output values and if you have any kind of error in the code, it have syntax checking and that kind of thing.

And the other thing we can do is add views, you know, actual images as annotations. So in this case if you wanted to include something like that in your drawing, you know in your model, you can just include those types of references directly.

Now, that's a little different than the external references that we want to add where the user is actually going out to something on the internet and adding annotation that way. This is actually embedded right in the model itself.

MR. CARUSO: That's nice, because you can include that as documentation. If you had something strange that you were modeling, you can include a picture of it so people could try to figure out what you were doing.

MR. JONES: Exactly. And the idea is if you have a idea or something like that, you could either embed it directly in the model or you could have this referenced to it externally. And, you know, as long as it doesn't move if it's an external reference, it'll come up through a hyperlink.

I'll go back to the --

DR. SIEBER: I take it that the numbers you put in for the fuel are cold numbers and not hard numbers?

MR. JONES: That's just an example. Those are just -- I don't think they correspond to any real data. In that case they were just an example.

DR. SIEBER: Yes, I know. But if it were real, would they be cold numbers?

MR. JONES: I think they typically are --

DR. SIEBER: Because the gap closes, you know.

MR. JONES: Right, right, right.

DR. SIEBER: When you heat up the element.

MR. JONES: Well, since it's a Python calculation, the user could even put in any kind of -- you know, could calculate that thermal expansion and temperature could be one of the inputs.

DR. SIEBER: Yes.

MR. JONES: And it could calculate the full area that way.

DR. SIEBER: Yes.

MR. STAUDENMEIER: The code has a simple gap type of model. If you put in conditions what it is and get changes in temperature, and the gap model to calculate that, I don't know how well it works.

DR. SIEBER: Probably as well as --

MR. STAUDENMEIER: Actually, one of our fuel rod codes coupling in FRAPTRAN or FRAPCON by putting in more sophistication fuel rod modeling. And it'll take care of any limitations in one of the internal modelers --

DR. MAHAFFY: This is John Mahaffy.

To give you a really precise answer, you're imputing more than just dimensions. You're imputing initial conditions so that the dimensions that you input are consistent with whatever initial temperatures you input.

DR. SIEBER: Okay.

MR. JONES: This is where you can zoom in and rotate the component. In this case, see we have two axial rings -- or, I'm sorry. Seven axial rings. Seven axial levels. Two rings and four azimuthal vectors. And as you select them over here or do a multiple section, you know, you can see where you are on that. And this is all drawn to scale on the reading view.

The teal colored section is zero angle data section. And it's just put in there for reference so that you can tell where you are.

Another thing we added was this capability of looking at the -- being able to select the data based on the intersection of this top down view and the side view. So if we select, say, three levels here and as we select different locations over here, or multiple locations, we can edit all those simultaneously. It was just a convenience to edit values. And it works in both directions. So if we select things over here in the table you can see exactly what azimuthal sector you're working in and where your levels are but the data's changing.

And it works on the face value as well. You just select which space you're working, say the radial face. And the level. And that would show you in the table exactly which values need to be changed, so you can just change them that way.

The other thing we can show is any connections in the vessel. This is showing where all the external connections are coming into the vessel. They're all -- you see this one in particular, pipe 43 is like a cold leg coming in the vessel. It's hooked to cell two. The ring to sector 4 level 6. And it will highlight that location. Up here you'll be able-- a little easier to see as it gets rotated. And as you select it, it'll show you exactly which face that connections being made to in the vessel.

One of the other things we have is the 3-D renodalization. In this case you have to select which axis you're going to renodalize and if I wanted to renodalize the number of radial rings, for example, this is going to come up with a little dialogue and it shows me, okay, I have two rings right now. And there's those red dots indicate that there's a fuel rod. You see structures in those quadrants.

If I select the inner ring and split that into two, it'll spawn new heat structures and it'll adjust the power components so now it's gone through and created these new either fuel elements and for channel, and scaled them based on the area that was selected.

Like I say, we are five model of -- I'm going to go ahead and submit this calculation.

Here's where that parametric selection would be. At the point where you submit the job, you could select which constant you want to vary parametrically and it would generate multiple runs. There's no parametric variable set up in this case.

Here's where you would get hat list of RELAP5 executables from the calculation server.

On this case I'm on locally, but you could also submit this calculation across the internet to any other machine.

Go ahead and start this pause. And in a message down here, the job has been submitted.

Now I'll open this job status utility. It's just connecting to the local server. And you see we have this job that was just submitted. It's in a paused state right now.

And even though we're not using the database, it does keep track of when the run was made and that type of information. Current status of it. Who submitted it.

Now, from here we can send an interactive command through this menu, we can resume the calculation and pause it. And that's all I'm going to do. I don't want to run it out too far. So that's in a paused state right now.

Go ahead and bring up the visual engineering data analyzer. And this is really the MPA replacement. And this is analogous to a mask here. Right now this is user. You know, the analyst has to sit down and create these masks. We do want to be able to leverage those displays that are created in the ModelEditor so you can automatically bring them in here and animate those as well.

MR. BANERJEE: So this is not being imported from the setup that you were showing for the initial deck?

MR. JONES: No, that's right. None of this is generated. This is created through this application. If you switch to design mode, you can modify any one of these components. This is the setup display, you now, if you wanted to add a --

MR. BANERJEE: So you have components you drag and drop in there or what?

MR. JONES: That's right. Yes, I just added an analyze dial, for example.

DR. RANDOM: You also have to relate what volume is going to be used to show different --

MR. JONES: Yes, exactly. This is using this JavaBean architecture. So for this component, you see this is a list of all its parameters. And one of them -- now right now I'm not connected to a data source. I need to add a data source.

MR. BANERJEE: So you create this once again? Why don't you just create it from your input deck?

MR. JONES: That's what we want to do. We want to -- unless we started with a ModelEditor that only had a single view and they're basically just iconic views. What we want to be able to do is use those views that we're generating now in the modelizer and bringing them into the -- we really haven't done much work on this over the last two years. So we want to merge that new work back into this.

So this would be the list of the data channels, for instance, for that. And now that component would be animated.

DR. RANDOM: Which component is that now?

MR. JONES: I think I selected AC tank 190. So it's just giving us a list of our data fields coming in.

Our design mode. Now, this is in a replay mode right now. It's going to run up until the point the calculation is paused. We're running interactively right now with this calculation. So if I--

DR. RANDOM: You have already linked the regions on the mask?

MR. JONES: Oh, yes.

DR. RANDOM: With data?

MR. JONES: That's right. This mask is all set up.

DR. RANDOM: And colored it appropriately.

MR. JONES: Yes, exactly. So these are all the data channels that drive that particular element.

DR. RANDOM: The blue is subcooled water and the green, I guess, is saturated, right?

MR. JONES: This is the map we use for color.

DR. RANDOM: Okay.

MR. JONES: And the user can set up whatever color scale they want.

DR. RANDOM: Right.

MR. JONES: But we use one range for subcooled liquid, another for steam and then another color range for the other.

In this case, I picked the same color at the top of this range, at the bottom of this. But you could use whatever scale you feel is appropriate there.

Now, we are interactive in this case, so I can come in here and open a break. And then when I hit play and resume the calculation, it'll actually go into a transient type --

DR. RANDOM: Where's the break?

MR. JONES: This is on the code like --

DR. RANDOM: You don't illustrate it, I guess, huh?

MR. JONES: We could stop at anytime and

move back. So we can move between this replay mode and the interactive mode and run up to the end of the interactive portion. And then if you hit play again, it'll ask you if you want to resume the calculation. And then, again, you're tied directly into the calculation.

DR. RANDOM: I guess you can automate like on the valves, click on them, flow --

MR. JONES: Yes, any one of these --

DR. RANDOM: The pumps?

MR. JONES: Yes. The options so it'll send the command to the code. And this is all done through that core interface that's flowing between data and the calculation server and then down to the analysis code.

Then resume the calculation.

Okay. So this is just a simple 2-D display. You can create as many of these as you want. Here's a 2-D of the vessel.

DR. RANDOM: Can you import these from old MPA decks?

MR. JONES: It wouldn't be difficult to add that capability, but you can't right now. Since we know -- you know, the data's all there. It's just a matter of writing a program to do that --

MR. BANERJEE: Right now it's all manually done, basically.

MR. JONES: That's right. But it will be automated from the ModelEditor soon.

MR. BANERJEE: Right.

MR. JONES: And then the MPA map, it'd be nice to be able to bring those in, but that's a fair amount of work to be able to read that data format.

Now, the other thing we can look at are the -- there's 3-D representation of the same data. It should come back to this guy first and show where this data came from.

On these 2-D displays there's a pop-up. There's view hydro at 3-D. This will go through and generate a 3-D representation of the vessel or all the hydro components, I should say. And then you can rotate to see how they're laid out.

Now, they're all drawn to scale as far as the elevations go. But since they're 1-D components, you can see there is these -- you can see these hot legs coming out of the vessel. Just go out to the end. We don't have any way of bringing those back in automatically.

You can use this tool to rotate the pivot around the Z axis at any of these junctions to make it a cleaner display.

DR. RANDOM: Well, have you retained all the -- only RELAP5 used to have the xyz coordinates that could be input. Some of them weren't used, but mainly the elevational thing was used. But the other ones are there if somebody wanted to create a real representation.

MR. JONES: Well, the other thing we could do is take the -- is input the facts that don't have that data in and then use this tool to rotate these components around and then export that xyz data.

DR. RANDOM: But the new TRACE code does have all the coordinates of where these components lie in space?

MR. JONES: No. I think we're going to be in the same kind of position there --

DR. RANDOM: Plot space, huh?

MR. JONES: Right. Right. Yes. We can do a lot of the generation, but there's still places where you have to pivot that data.

So again, this data all gets animated.

DR. RANDOM: Another feature that you could build in that sort of would be an indication of errors is at an xyz space you don't close, both elevation wise as well as azimuthally and whatnot, it's an indication something is not known about the system.

MR. JONES: Well, I think the real advantage is being able to see in a 3-D space what the data really looks like.

DR. RANDOM: Yes.

MR. JONES: One of the things you'll notice is --

DR. RANDOM: In fact, if there was any intention to build this, and I notice you did say that the gravity director can be in any direction; and if that's true, you need to close in all the 3-D space if the gravity does not correspond to the vertical dimension, why then you're going to have trouble.

MR. JONES: Sure.

DR. RANDOM: You anticipated that? Because I notice you have made gravity a variable.

DR. MAHAFFY: Gravity is a variable right now, but it should scale at the face. It's an angle relative to the Z axis.

DR. RANDOM: Right.

DR. MAHAFFY: What we have in our data structure is additional information to in effect give you a vector of gravity, but that's not available by input. So it just sits there in the data --

DR. RANDOM: You only use the projection of the gravity vector on the vertical dimension to give you any type of dyrostatic --

DR. MAHAFFY: We can extend things from scale to extract the information off the data --

DR. RANDOM: I know the Navy at one time, I think even in RELAP5, they do have dynamic system --

DR. MAHAFFY: That's in TRACE

This is John Mahaffy, by the way.

That's in TRACE in the vessel. Your vessel of gravity can be any three dimension angle off of your vessel.

DR. RANDOM: But not in the 1-D?

DR. MAHAFFY: The 1-D really is a pointer view and you get in relation to the axis. But that's it.

DR. RANDOM: I don't know if nuclear wants to bother with or not.

MR. JONES: I'm just letting this calculation run out.

Now, as you're connected to this data source, you can switch into this design mode and work with the live data. So you can create the display and edit the display while you're connected to the calculation.

Unlike the MPA where you had a separate application that was used to create the displays, but then you had to bring in -- it was labor intensive process of trying to map those data channels to the visual elements. Here it's all done in the one application.

MR. BANERJEE: Can you also graph the data and look at it?

MR. JONES: Yes.

DR. SIEBER: By time.

MR. BANERJEE: By time.

DR. SIEBER: Yes.

MR. JONES: There's a couple of ways we can do it. We can use graphs like this, you know, a strip chart type representation. That can of thing. And those can be as involved as you want, of course.

DR. SIEBER: You can put more than one parameter on the chart, right?

MR. JONES: Yes, this one does. You just switch to design mode. And then this guy, it allows you to set up to six data channels.

MR. BANERJEE: So you deduct the strip chart.

MR. JONES: That's the strip chart.

MR. BANERJEE: But what about can you see it, what would the length of one transient to the point you pause or something?

MR. JONES: Like this?

MR. BANERJEE: Yes.

MR. JONES: Yes. This is -- we've linked this application and the job status application so that it goes out to the calculation server and grabs that data and plots it.

Again, it's kind of a -- a little bit of a pain to use AcGrace because you'll see this little x here, you had to be running an x server at the time. And, you know, it is it an application that had to be complied on a machine where all this other software. We just compile on one machine and then we can run it everywhere. It's a lot easier to live with.

DR. RANDOM: You mentioned having 3-D plotting capability?

MR. JONES: That's what we'd like to add.

DR. RANDOM: Oh.

MR. JONES: That's int he wish list.

DR. RANDOM: I see.

MR. JONES: And that's replacing this with a 2-D/3-D plot application that still has the -- type capability that XMGR5 and AcGrace had where you could, you know, create your equation and analyze your data with it.

MR. BANERJEE: Are you going to actually just plug in existing package for that or are you going to --

MR. JONES: We have to assess what's currently available out there. You know, really kind of look at what's in the open source domain.

MR. BANERJEE: We'll allow you to do that.

MR. JONES: Yes. I think we're going to be able to get most of it from that. And then we just tie in our own IO routines so we're able to read all these different data formats.

This is that job status application. And it's the calculation we just ran. We can also view the output from here. But this will give us a view of all the generated output.

Some points of interest here where we can select like major edits in this case. There it goes through, it parses the whole deck, finds all the major edits and then you can just go down through the list here and it'll take you right to that location in the file. So we're looking at major edit 200 at this point.

And you can do the same thing for warning messages. The whole idea is that we're just trying to make a productive environment to perform the calculations, so to make it easy to edit the calculations, submit them and then analyze all the results from --

MR. BANERJEE: If you're sending the calculations to be done over the internet to a different computer, are you receiving the results over the internet as well or do you --

MR. JONES: Only the portions you look at.

MR. BANERJEE: I see.

MR. JONES: This is a good example of that. This file could be a gigabyte worth of data somewhere on the other machine. The only portion that comes into this viewer is the portion you're actually looking at, with about a 50 line buffer on each end. And it caches all the data that's been brought in.

All this parsing is done over on the calculation server side. So the calculation server goes through and finds all those points of interest. And then we go through the core interface to locate, you know, to get together that information. And then this output viewer goes to the calculation server and says okay, I want lines 500 to 600, and it just brings those portions in.

MR. BANERJEE: So it brings them in,

whatever you are viewing?

MR. JONES: Exactly.

MR. BANERJEE: It's a very small file or data stream?

MR. JONES: Yes. Exactly. The idea is we want to be able to run it. And I've run this over my dialup modem. And I can submit this calculation over my cellphone and bring the data back and analyze it.

MR. CARUSO: Every night between these two buildings, 1500 computers that sit all night and don't do anything. With this system you could have them all running sensitivity studies all night?

MR. JONES: Absolutely.

DR. RANDOM: They'd have to be written from your car.

MR. CARUSO: They're all networked. They're all networked.

DR. SIEBER: What? All the NRC computers here?

MR. CARUSO: Every person in this agency has one on their desk.

DR. SIEBER: Great. A lot of passwords to remember.

MR. BANERJEE: People do this for the SECY project anyway.

MR. KELLY: We could have something on the web that everyone can download so they can run in the background.

MR. CARUSO: They wouldn't even have to know that it was --

MR. BANERJEE: Free screen saver.

MR. CARUSO: You could do that in this building right now. You could do that, right?

MR. KELLY: Well, actually, no. Because it has to have Java 1.4 and that's controlled by, you know --

MR. CARUSO: Sierra --

MR. KELLY: It has to be a special request.

DR. SIEBER: They only have one license.

MR. JONES: One of the things we are looking at for the installation is including the Java virtual machine with the distribution. We're probably about six months away from doing that. So, the only catch is it's dependent on which platform you already have to have. You know, will have to support -- you know, let the package be a VM for each platform you want to run on.

That's everything I had to demonstrate. Is there questions?

MR. CARUSO: Take a break?

MR. STAUDENMEIER: I guess one thing I'd like to interject before this, I mean mostly now SNAP has been oriented towards code information and things that you see through the code ASCII input deck. But you've seen a sample of these user defined functions and templates for like a channel or a pipe. And eventually we'd like that to be interfaced -- have these templates that come up and help them model actual nuclear reactor components and we could imagine building in things like a ideal loss coefficients that you go look up the geometry and -- a loss coefficient for that geometry. But they help the user along and you can modeling complex geometries with what they need to know about the hardware and not all the intricacies of dx's and flow areas and volumes for every cell that that can automatically recalculate by just a few simple inputs that the user defines.

DR. RANDOM: What do you do for RELAP5? You know, it has it's own -- change, loss coefficient calculations scheme -- but in order to match their result you would have to have the same lost coefficient, I guess, unless somehow you imported that model?

MR. STAUDENMEIER: That model has not been part -- forwarded over, as far as I know. One thing that is going to be imported over is a Reynolds number dependent loss coefficient. But that's something that could be imported over in the future also.

TRAC has an abrupt area change model that you implement through interface. It's actually a flag on the faces that you import.

DR. RANDOM: You can calculate a lost coefficient.

MR. STAUDENMEIER: It'll calculate a lost coefficient automatically or also what the user -- additional loss coefficient on top of what it recalculates automatically also.

DR. RANDOM: Do you know what that --

DR. MAHAFFY: It's basically a -- contraction. It's some kind of a --

DR. RANDOM: You can a --

DR. MAHAFFY: Yes, but it's probably not identical to the one in RELAP5. Someone would have to do a study to see what the true results are.

DR. RANDOM: Well, most of it's textbook type stuff. Based on the --

DR. MAHAFFY: Yes. That's right. It was taken out of textbook. But whether it's the same textbook that RELAP5 used or not, I don't know.

CHAIRMAN WILLIS: Okay. We gained some time. We're going to take a break. I think we should take it until about 3:20. And then we'll get back to the program with Chris Murray.

DR. RANDOM: What time?

CHAIRMAN WILLIS: 3:20. Until 3:20. Break until 3:20.

(Whereupon, at 3:03 p.m. a recess until 3:23 p.m.)

CHAIRMAN WILLIS: Sound effects again. What was that? Was that the rewinding the tape or something?

MR. BANERJEE: That's the evacuation signal.

MR. MURRAY: We are ready.

CHAIRMAN WILLIS: Okay. Go ahead.

MR. MURRAY: My name is Chris Murray. For those of who don't know me, I've been on the development team now for about five years, but I've been sort of in the back of the process and more day-to-day operations and not on the phase.

I'm the code caretaker for TRACE. I handle updates come to me, and I handle, you know, applying those updates to our official version of the code. And making sure that they're properly tested, reviewing the updates to make sure that they meet at least some minimum level of our code standards and making sure they look all right.

Today I'm going to talk a little bit about our automated testing and assessment tools that I've developed for TRACE. And my goal is to give you an overview of the testing process that we follow. Some of it's been in place for a while, some of it we're phasing to. And I'll also give a walkthrough of the tools that we use to accomplish that testing process or follow through on that testing process.

I don't want to get into real low level details. A lot of what's involved in the automated testing is making sure the file is in the right directories and moving files places, and things like that. And I wasn't going to try and get down at that level.

Our in-house testing process has three tiers. The first tier I call the code-wide regression/verification testing. The purpose of this portion of the testing, which is something we do on a day-to-day as updates are applied, is to ensure that the existing behavior is not broken and also that new features are implemented correctly.

So what happens is a developer will submit an update to me and they may add a new feature to the code. Ideally, they've created some set of test problems that test that function out to make sure it's functioning the way they think it should function. And so I'm taking those input decks that they supply me, making sure that there are enough input decks and exercise that feature properly, and fold those into our test suits and run the entire set ensuring that no existing behavior's been broken in the process.

So a developer may only run some subset of our test problems. But I'm making sure, hey, across the board nothing's -- we haven't regress in any way.

Ideally, that would be a large set of problems. You want them short running that turnaround time is fairly quick. Less than a day is ideal. We would run them against each code update. And, ideally, we would run those across multiple platforms.

Right now I'm not doing across all platforms that the code does run on. Mostly it's just PCs that I run the code on. This is something that I need to address, but I need to restructure our build environment so that it's all automated. Right now it's too manual a process to really carry out day-to-day. But the rest of it mature.

We've been doing this now -- well, the

whole five years or so. And I took this process over from Jennifer Uhle when she went on. Actually, a few months before that.

The second level of testing that we've conceived of is what we call robustness testing. About maybe a year ago or so, we started to notice when we started get in the assessment, we started to notice robustness problems. Initially, I think it was my concept was these initial set of testing wouldn't-- you were fairly confident that you weren't causing new problems to occur. And I don't think that that was entirely the case. We found that some test problems that used to perform well or used to run to the end were suddenly dying. So we conceived an intermediate level of testing to address the deficiencies in the regression test methodology. But we don't want the full overhead or the overhead of a full assessment. A full assessment could take months, to be kind to the process.

Quickly, another goal is to quickly assess the overall health of the code from a quality results perspective. It is the code predicting results compared to some limited set of data as well as it used to? Is the code time step being bogged down in certain places of the code? Things like that.

DR. SIEBER: Do you compare the results against a standard? And if so, what is the standard?

MR. MURRAY: Well, the standard would be the data for that problem, the experimental data.

DR. SIEBER: Yes, but you're looking at results and not the input, right?

MR. MURRAY: That's right. We're looking at the results.

DR. SIEBER: So you use some other run that's sort of prior to the modification as the comparison media?

MR. MURRAY: That would also be the case. What you'd do is you would look at the data, the previous version's calculation and the current version.

DR. SIEBER: Okay.

MR. MURRAY: And so you'd see that you either improved things or you made them worse. And you could even look at the data to see how that got closer to the data or further away.

DR. FORD: So in that very first bullet, the top, you could for the word "behavior," you could substitute "accuracy" or "agreement with data?"

MR. MURRAY: No, no, not really. I mean what I'm looking at this level has -- I don't know.

Have you upset the time step backup behavior so now is a flag being turned on that shouldn't have been turned in? Is the input checking that used to work, not working anymore?

So you're also looking at features in the code. I mean, you're also going to see in a lot of cases if you've made the results worse. But that's not really what --

CHAIRMAN WILLIS: You're not testing the comparison with data, are you?

MR. MURRAY: No, not at that level.

CHAIRMAN WILLIS: Is it running? Is the machine running as it was supposed to run?

MR. MURRAY: Yes.

DR. SIEBER: And it doesn't oscillate or--

MR. MURRAY: That's right.

DR. SIEBER: And in the order that it's supposed to do thing?

MR. MURRAY: I mean, it would if I added any trip type to the code. If I give an input deck that's supposed to cause that trip to turn on at 10 seconds, is it turning on at 10 seconds? Are you seeing that behavior that would lead to that?

DR. SIEBER: Yes.

DR. FORD: So nowhere on this slide is the question of does it predict what is in fact seen?

MR. MURRAY: Yes, not yet. That's the next slide.

DR. FORD: Okay.

DR. RANDOM: Were any of these simple solutions that have exact solutions or simple problems that have exact solutions?

MR. MURRAY: Yes. One I'll show later is, you know, minometer and U2 --

DR. RANDOM: Well, for those, certainly you can do an assessment?

MR. MURRAY: Yes, that's right. That's right.

MR. STAUDENMEIER: The robustness testing is actually looking against test data to see if it's giving the right answer.

DR. FORD: Oh, okay.

MR. STAUDENMEIER: Not as detailed a look as you would give in a full assessment, but it's a quick look and spot check like a canary in a coal mine to make sure things haven't diverted --

DR. SIEBER: On the other hand, my understanding was that in robustness space you're looking to see that the program actually runs without going on some wild loop someplace as opposed to looking at an error criteria in the analytical result. Is that correct? You get a result that looks like the other ones, but the most important feature is to make sure that the code runs the way it's supposed to run?

MR. MURRAY: Yes, I think generally it's that you haven't done something that's slowed down the code greatly, introduced a problem that, as I say, it could take --

DR. SIEBER: Or plot it off on --

MR. STAUDENMEIER: -- ten as many time steps.

DR. SIEBER: Different work, yes.

MR. STAUDENMEIER: Or that the accuracy as compared to some experimental criteria haven't degraded vastly like you've broken one of the physical models in the code.

DR. SIEBER: Okay.

MR. MURRAY: Because this box in here, we presumably take at least several days to run through those cases. Because they're designed to be longer running cases. We would probably run those less frequently.

You see, there's some updates where we expect or done the same before you expect updates aren't going to change results. And for those updates you're fairly confident that if you get through this with no results, that you're not going to see any changes to the data.

That's not always true. When we added new heat structure, you're changing results on a wide scale. And across a large number of test problems. And what you need, is you need some longer running test problems that are going to catch really places where you really set the calculation off in different ways. And those happen at usually major updates.

DR. SIEBER: Have you made any attempt to -- well, 40 years ago we used to flow chart the codes and then test all the branches to make sure that -- you know, because sometimes you run the code and you don't hit all the branches. You hit the certain series depending on what the problem is.

MR. MURRAY: One of the things that I try to encourage in the process, and I'm not always successful largely because the overhead's a little large, is what we call test coverage profiling.

DR. SIEBER: Okay.

MR. MURRAY: And you aim for 100 percent coverage of the code that you've modified. And so what you want to do is when the developer develops an update, they also should be developing test problems that show and demonstrate 100 percent coverage of every line that they've touched.

DR. SIEBER: Okay.

MR. MURRAY: And that's happened in a couple of places. I did it one update that I made to the control system a few years ago. But the tools that we do that with aren't very -- they're kind of cumbersome to use.

And I try to structure things in our development process that automate things and allow you to let the computer do the things the computer is good at and let the developer worry about, you know, doing the stuff that they're good at.

And so getting people to really jump on that bandwagon has been tough, because it is time consuming to run through that process.

DR. SIEBER: Tedious. It's tedious?

MR. MURRAY: Yes.

DR. SIEBER: Okay. Thank you.

MR. MURRAY: Yes.

Next slide. And the third level of testing that we conceive of here is developmental assessment. Nothing new there. That's comprehensive assessment of code accuracy and the ability to predict the relevant phenomena.

I really want to phase this sort of testing in more frequently. And to do that, you really need to automate the process, which means generating the figures automatically and having some way of really visualizing all that output and being able to parse that and focus the developer's attention really where there attention needs to be.

Eventually, and we're not using these now on any regular basis or even this whole process isn't really in place, but eventually use goodness-of-fit strategies to focus the developer's attention on the trouble spots. Now, goodness-of-fit would be quantitative figures of merit that, you know, demonstrate is the modified version of the code closer or worse as compared to the data that it was before.

And these are not silver bullet type processes or solutions. They way you generate these figures of merit really can lead to false/positives, but as a means of focusing a developer's attention, I think you can gain something there.

There are tools that exist. I'm going to show some of those later, that allow us to generate these goodness-to-fit that these figures of merit are.

We need to gain experience using them. We have none at this point.

DR. FORD: Are there acceptance criteria, preset acception criteria? For instance, prediction is within plus or minus ten percent or whatever it might be of the data?

MR. STAUDENMEIER: This is Joe Staudenmeier.

We haven't defined acceptance criteria yet, but we'll have to go through on a case-by-case basis and define quantitative assessment criteria. That's what we want to move towards anyway. In the past, acceptance criteria have been more qualitative, like you look at all those major trends and as close to the data. But we do want to move into more quantitative mode testing with some of these automated testing tools. So we're not there yet, but we're moving in that direction.

DR. SIEBER: When you test the accuracy of a code, you're testing it against data that comes from testing facilities, most of which is scaled because the facilities are not the same size as the actual problem that you're running. How do you tell the difference between calculational error and scaling error, or can you or do you even try?

MR. MURRAY: I know we're not trying. I'm not even sure how we'd go about that. Joe may know.

MR. BAJOREK: This is Steve Bajorek.

That really wouldn't be an issue with the stuff that Chris is trying to do here. This is checking in the updates and making sure the hundreds or thousands of decks that you have now run and execute.

When it goes back to comparisons to data, that's the -- I think you're going to start talking about pd script later on in your presentation, this is one of the automation tools that we'll have for taking code results, comparing them to experimental data.

DR. SIEBER: Right.

MR. BAJOREK: And out of that comparison getting a comparison that we can quantify.

DR. SIEBER: Right.

MR. BAJOREK: Whether it's a quench time. I mean, you don't like us to hang on PCTs.

DR. SIEBER: Right.

MR. BAJOREK: But parts of that transient by which you can pick out of the transient -- the simulation and the data itself.

DR. SIEBER: Right.

MR. BAJOREK: And we'll move to some quantifiable metrics there. I'll talk a little bit about that tomorrow.

I think what Chris is getting at now are what are the tools that are going to make us automate the process so that when we make a code change or come up with a new revision to the code, we can do that quickly without having to repeat months and months and years of assessment.

DR. SIEBER: Right. Okay. I'll mark that down for tomorrow afternoon.

MR. BAJOREK: Okay. But with regards to the integral tests and the scalability, one way that that's been approached in the past is to take a look at several integral facilities that have the same processes going on in them and then showing that you don't have a scaling bias with the results.

DR. SIEBER: And that's to look at differing facilities that require different scaling?

MR. BAJOREK: That's right. For example, in --

DR. SIEBER: So they don't drop out?

MR. BAJOREK: -- ECCS bypass, you may want to look at 1/15th and 1/5th scale tests from Creari and the full scale UPTF tests. And hopefully if you are under predicting a certain process in one, you're doing it in all. It's not -- there's not a discernable trend in that as you go to full scale your results at the lower -- the smaller scaled facilities aren't valid anymore.

DR. SIEBER: Okay.

CHAIRMAN WILLIS: Well, it's the question we've had before of noding. Do you node a pipe that's 2 feet high with 10 nodes and you put in phenomena and all that? You got a pipe that's 100 feet high with 10 nodes. The nodes are now ten feet long and their description of the phenomena in there may not be as -- be quite different from what it was for the much shorter nodes. Because the transient time for a bubble -- something is different, you know. The noding relative to the phenomena has now changed.

MR. BAJOREK: Yes. And as we are moving ahead in our assessment, what we are doing is rather than trying to just preserve the nodalization that people have used in the past, we're now trying to make those, you know, gravitate towards something that we want to stay with for the full scale PWRs and BWRs and eventually the other types of reactors, preserving that same type of nodaliztion techniques and the integral and the separate effects facilities so we don't induce --

CHAIRMAN WILLIS: You're going to do ten nodes for a pipe even though the scale has changed?

Because, you know, you really ought to have a code which is robust against changes in noding. That's the real test I think you should have.

MR. STAUDENMEIER: Yes. I think that is what we need to do.

Another thing in relating scaling is where there's distortions in phenomena between the test facility, the scale test facility and the plant. Just because of the reduced size, you won't have that flow regime that allowed in the test facility that may be allowed in the plant, or a gross distortion in the phenomena based on the correlations, have a correlations scale from small scale things to large scale things. The phenomena change and cause a distortion in the response, and that's a difficult thing.

CHAIRMAN WILLIS: But that's not what he's talking about here, is it? That's something else?

MR. STAUDENMEIER: That's not what Chris is talking about at all. That's a lot more involved process relating top down scaling, button up scaling with test facilities to plant analysis, and that's a very long involved process to do that.

MR. BANERJEE: You know, robustness with regard to noding is desirable, but hard to get.

There's a paper by Jeff Hewitt and Jayanthi in the *General Of Fluid Mechanics* in the last years where they took this model for a pipeline and they showed that you had to really node down very, very finely to get a mathematically converted answer. In fact, it converged. They got very good answers. And it agreed with everything.

So there was this sort of -- it wasn't even uniformingly converting it. It did something like that to this.

So this suggests that mathematical conversions would be difficult to obtain, even in the straight pipes where you might get it. I think it would be almost impossible to obtain in these peculiar components where you're sort of fudging 3-D effects into 1-D effects. I think everything you change the noding, you've changed the answer, like the plenums.

CHAIRMAN WILLIS: That's not very reassuring, though, is it.

MR. BANERJEE: Well, I think you're taking a 3-D situation and making it 1-D.

MR. BAJOREK: What code were they using? What code were they using in that study?

CHAIRMAN WILLIS: It's safe with 13 nodes--

MR. BANERJEE: They wrote a code base, but it looks like TRAC. I mean, the pipeline code that people use now called PLAC is one of them, and was a variation of TRAC with the hydrodynamic head terms added to it. So it actually -- everything is well posed, everything works well. There's nothing wrong with it.

MR. BAJOREK: One way of getting around the scaling issue is to make sure in your assessment you are choosing either tests which are well scaled; I mean things like APEX and things where you've really done a good scaling analysis, good scaling evaluation, or to stay with test facilities which are essentially full scale.

UPTF when it comes to ECCS bypass. Even some of, like test 8 for looking at flow patterns in the hot leg. Most of the reflood tests that we'll be simulating, full scale, full height bundles, relatively large scale in terms of the number of rods. So that you should not have a whole lot of atypicalities due to the size of the bundle itself.

MR. BANERJEE: Full scale is good, but when you do sort of the APEX type scaling, it's very scary because I can tell you, this is a true story. In chemical reactors they were doing reduced height scaling for emergency release. They were taking that to full scale reactors and the full scale reactors were blowing up during the emergency release. And the reason was that level height, level swell does not scale with height. You have to have the right filling to start with.

So, there's no substitute for full height there.

MR. BAJOREK: I was using APEX as an example as opposed to, let's say LOFT where we know there's serious distortions. At least I'm not aware of going through a detailed scaling evaluation to try to quantify what those distortions are.

MR. BANERJEE: Yes. LOFT is another story, yes.

MR. BAJOREK: Yes.

MR. BANERJEE: But I don't know what you do. There's sort of an unfortunate situation because full height scaling is okay. But there could be other effects to the diameters. So just to depend on that and not to seek mathematical convergence with -- I mean that would be really reassuring. I don't think you can obtain that. So you really have to have some mishmash

of these things which, hopefully, will make a good story.

MR. BAJOREK: I think we've talked for AP1000, we look at the APEX tests as being good and well scaled when it comes to looking at flows into and out of the system. The ADS performance, the IRWST and the CMT. When it comes to level swell in the core, it's no good. You've got to go back to the tests like--

MR. BANERJEE: Full scale.

MR. BAJOREK: Yes. Full scale THETIS and G1 and G2 in order to really evaluate your codes.

DR. RANDOM: It depends on the phase of the accident you're looking at. Like in PUMA, it's like from 125 PSI on, you know. And at that point it's primarily single phase liquids draining into minor two phase mixtures. And those scale quite well.

MR. BANERJEE: As long as there is no level swell.

DR. RANDOM: Not a significant amount.

MR. BANERJEE: Right. Is it single phase?

DR. RANDOM: Well, you know, it's boiling in the heated section of the core but with a large vapor region above it, in effect. You know these are rather tall vessels with quite an upper plenum, in effect. And for those kind of situations I think it's quite reasonable.

Now others may not, you know. It depends on the reason your reactor is blowing up, I guess.

MR. BANERJEE: What if you're blowing up due to level stress, which was what was happening.

DR. RANDOM: In other words, it plugged up the outloader or the relief valve?

MR. BANERJEE: The relief valve.

DR. RANDOM: Yes.

CHAIRMAN WILLIS: We're getting a bit off.

DR. SIEBER: Yes, we are.

CHAIRMAN WILLIS: I think when your screensaver comes up, it's time to reassert yourself.

MR. KELLY: I'll set mine for 30 seconds tomorrow.

CHAIRMAN WILLIS: Yes.

MR. MURRAY: I'm just going to impress that really what I'm trying to show are the tools that I'm either developing or have developed or in the process that I want to use to keep the code from regressing in performance. And also, the way that I'm trying to not lose the investment in assessments that we do and the development work that goes on, it's important that we don't recreating the wheel every time we create a new code or something.

Developmental assessment and automating that process, the frequency of execution of an automated assessment really depends, I think, on the state of the development program. Certainly you'd run it every time you're ready to release a code. And maybe a month or two beforehand so you have time to respond to issues that you might find. Although I would also say, or I would probably also think that you'd run this even when you're trying to make major modeling changes.

You may have two or three people making some fundamental changes to the constituent relationships, or something like that. And they ought to be either running that themselves to make sure that they're not changing behavior in ways that they didn't really want to --

DR. RANDOM: Who is the configuration guru that determines when control updates going into the code?

MR. MURRAY: That's me.

DR. RANDOM: You?

MR. MURRAY: That's me.

DR. RANDOM: Uh-huh. So people have to have done their work before they can actually have the update incorporated into the code?

MR. MURRAY: Yes. Basically what happens, what a developer does is they do their work on some version that was the latest at the time they started the work. And they go through the process of developing the code, and they develop their test problems and run them. And they show that their updates are working correctly.

And then usually what they'll do is they'll grab my regression set and they'll run that as well. Now they know that their update isn't going to change everything.

At that point I've got scripts that build these HTML webpages for them that summarize the update. And they type in the summary of what the update does, who did it, who reviewed it. There's usually an internal developer within our development group that reviews their update. Every subroutine that changed, how they changed it or the intent of the change. They will state whether the manuals need updating or have been undated.

And then they'll say, okay, here's the test problems I had. And if they ran the whole test suite, they'll say they ran the whole test suite. And then they'll say what were the results. And they either go no results or they'll tell me which test problems changed.

And in a lot of cases, you know, I'll see updates where somebody will say, okay, I changed -- or I ran these three test suites and this problem and this problem changed. That's all they'll tell me. And I'll say well wait a second, you didn't tell me that that update, that the changes you made caused that temperature change in this problem. If you can justify to me that you understand that difference that you caused in the code, then I'll let the update go through.

Ideally, I ought to even be looking at lines of code that they change. And to see degree, I'm able to do that. For very large updates, it's a very tedious process. And if I've got ten updates sitting in our holding bin waiting to go in, I don't always look at every single line of code.

DR. RANDOM: I know Dr. Mahaffy was mentioning that, you know, sometimes people would make changes that would destroy the synchronization basically of parts. And it would seem like somebody like himself, almost, must have to at least examine it and approve.

MR. MURRAY: Yes. What was happening before, and that was true, it was because a regression set, having incorporated parallel testing, that's one issue. Because parallel testing is a little bit more complicated than just running a simple input deck. You've got to make sure the -- are in place that are allowing the communication to happen.

All my testing is done in a four processor Dell with some Pentium III chips on it.

And the scripts that I had weren't able to really run some code in parallel. That's since changed. Now I will know if somebody's impacted the parallel, you know, I'm going to know about it right away.

Now if you actually look at the parallel test suite, it's fairly simplistic. If somebody upsets the ability for a heat structure to communicate across processes, right now I'll tell you my test suite's not going to catch that. The test suite's only as good as the test problems you have in there. But we take the view of let's get it in there. And as long as you consistently add those test problems here, you're never going to regress. You can only get better as time goes on. So there are known gaps, but you do the best that you can with what you have and move forward. And as you find bugs, the next time somebody does find a bug in the parallel that's not addressed in the test suite, that test problem comes in. And you guarantee you're not going to have that problem again, or that same problem again.

And that was sort of the process. I haven't really talked too much detail about the tools we use.

These are sort of the requirements that you have for creating a set of automated tools.

Number one would be scalability. As I was just saying, you need some way of easily adding those test problems that developers are submitting or that people who are doing assessments are creating input decks and creating figures and things like that. You need some way of putting all that information in. I don't want to spend all my time manually recreating their work.

Preserve the historical integrity. This is important. I want to be able to revert an entire system back to some previous state of the code. As you increase the code, sometimes you might find bugs that are really bugs in the input deck, and we'll go in and we'll fix the input decks.

Well if I don't have some way of going back. Let's say two years I want to run an old version of the code, I'm going to run that. But now

I'm going to have a whole string of input decks that are dying because the input deck changed, not the code necessarily. And then I'm going to spend a lot of my time just trying to figure out why is this input deck dying because of something we changed four years ago. So you have to preserve that historically integrity and tie a snapshot of your system to every version of the code as you march forward.

CHAIRMAN WILLIS: This would be automatize, though. You wouldn't have to recreate the past two decks because they go back to the past --

MR. MURRAY: That's right. That's exactly right.

Eliminate maintenance points. I sort of went over three levels of testing, and as much as possible would like to share, just have one input deck that exists. I don't want that same input deck existing in multiple directories and multiple places because if I find a flaw that requires I modify an input deck or I just want to change the input deck so it runs out further in time, or I remove a nameless option that input deck, I want to change it one place. I don't want to go and try and figure out every location on my file system I need to change that input deck. So I want one copy.

Flexibility in operation. I want it to be an easy to run one test problem -- you know, one particular test problem in that 1000 -- actually 1200 test suite as it is to run all the test problems. And I may run my entire suite of tests and I may identify three that have changed. And just as a quick check, I might say okay, let me modify the code. Let me just see if I fix those three test problems. I don't want to have to rerun all 1000 at that particular moment. I will eventually before I check the code in. But as I'm just trying to fix a bug or something, I want to be able to run just the test problems that I want to run.

It's got to be automatize, that's what we're talking about here.

The ability to summarize vast amounts of information in a way that focuses the developer's attention on the hotspots. With a 1000 plus test problems, it's very easy to just get lots in a long list of directories and files and trying to shift through numbers. And there ought to be a nice easy way to summarize that, capture that and say here's where's a problem, here's where's a problem.

And exploit multi-processor environments to enhance runtime. Bays of Beowulf clusters now, you'd like to be able to spawn off jobs on multiple processors. So you're running -- if I've got a 1000 test problems and I've got ten processors. Okay. Each processor only has to run 100 decks instead of 1000 on one CPU so wall clock time gets better.

Those are the requirements. And I'm trying to say hear that I sort of built that and met those requirements. It's a little bit of a cheat, because I define the requirements based on what I built.

We have what I'm calling our automated testing framework. I like to think of these are production quality tools. I try to make all this stuff was adequately documented for the developers, because I'm intending for developers and users to be able to take this stuff and be able to incorporate it into their own, either operational environment or a development environment.

Rigorous error checking so that they know whether my scripts are performing the way they should or not. And a high degree of flexibility so that they're useful across a wide range of uses.

All the input models are maintained in one place and are all under revision control. So I can always step back. With each update, I try to tag each update with the code version that I'm checking in at that particular time. And having in one place makes the process scanable. So if a developer submits a set of ten input decks to go along with an update, I copy the files over. And I just drop them into a single directory and they're in my test suites. That's all I got to do. I don't have to play around with really manually modifying my scripts all the time.

The test problems are organized in the test suites. And I'll go into more on that in a minute. But it's a way of grouping related sets of test problems for modularity.

It's platform independent. I use a scripting language called Perl, so anybody can run Perl who can run this stuff. And on the output side, it's based on HTML webpage type files.

I have three subsystems which, naturally, mirror the three levels of testing I've presented above.

And I'm able to visually parse the results. And this is fairly new, this part. Up until this past summer most of my testing process I went through in checking for null differences was really scanning files, you know, doing directory listings and looking at sizes of files.

Our regression testing is 1246 test problems. It's test suite driven, so I do test suites or regroup test problems by either functionality or feature.

I have a test suite in my system that I call "choke flow." And the test problems that are in that test suite all are focused on demonstrating that the choke flow models as it was designed. And different test problems can be shared amongst different suites. So in some cases I'm actually running more tests than this 1246, but some of those are duplicates that I'm rerunning. And that's an inefficiency in my process that I eventually want to address.

I could have that same Marvican test that's demonstrating choke flow off in a test suite that's testing the time step backup capability, because that test problem happens to do time step backups.

It's shared-memory parallel. So I run on a four processor Dell, so I'm actually a four process system. It takes me five hours to run my set, do all 1000 test problems. If I was running on a single processor, it would be five hours times four.

It's under configuration control. All

files are tagged by code version.

I run this suite for every new version.

I check for null differences against the previous against the previous version. In most cases when the developer expects it to be null, they're nulled before the code gets checked in.

I've got a web-based runtime statistics facility that allow us to mine our test results and understand whether or not on a global scale is the code getting -- it gives us a sense as to whether code is getting better or worse. It's a feature that John added to the code this past summer in response to some of his work.

And our user interface is geared more towards developers at this point.

DR. FORD: You can just give us an idea. The 1246 test problems, these are temperature, time, transient in a given reactor because of action and you're seeing how the --

MR. MURRAY: There's some full plant models that are just cut back in time. We won't run our large big LOCA out to its full transient.

DR. FORD: Okay.

MR. MURRAY: We might cut off in a transient after 100 seconds instead of 500 seconds.

DR. FORD: So TRACE has been through an examination in 1246 different scenarios?

MR. MURRAY: Yes. Different configurations of input. A lot of time what we might--

DR. FORD: With this?

MR. MURRAY: Yes. 1246 --

DR. FORD: So a very extensive examination?

MR. MURRAY: Yes. Not data. I mean all of these 1246 are not compared -- don't necessarily have data associated with them.

DR. FORD: It's just the word "test" is fooling me. 1246 tests is not 1246 experiments?

MR. MURRAY: I should have wrote 1246 input decks.

DR. FORD: Ah.

MR. MURRAY: I'm sorry.

DR. FORD: Okay. No, that's okay.

MR. MURRAY: To me the word becomes interchangeable.

DR. FORD: Okay.

MR. MURRAY: And a lot of times what the developer does, is they may take a full plant model that has a certain configuration and they may have added a new pump model to the code. So they'll just take that deck and tweak one number in the input deck and feed it back and say okay, this is my test for that. And, yes, it's essentially the same as some other input deck with one number might be different. But that's a valid test. I mean, there's millions upon millions of combinations of numbers that you could have in an input deck. And, ideally, you'd test them all. But I'm not going to live that long.

We runtime statistic features, which is part of our regression set now. We added this this past summer. It allows us to quickly assess the overall impact of a particular update. But we're actually looking at metrics like, you know, end time of every test, number of time steps it took for that test problem, total CPU time, the total number of iterations. And you can run down, I guess, and read them all here. But these are different metrics that sort of give the developer a feel for the health of the code after they've applied their update.

Use a web-based summary of those results that I'm going to show here. And here, after I've run my test set and I have a script that generates this page here. It shows have any of my test problems failed.

The green flag. You know, green means good and red would mean bad. So if this is a number greater than soon, this would be red. And it's a visual cue that, ops, the developer has changed something.

Some of the decks are designed to fail because they're testing input errors that are supposed to be generated if you feed it erroneous input.

And you have your runtime statistics, which you click on these links and it brings you to a page that'll give a summary of that information, which I'll show.

And finally, look at the differences. If, ideally, a lot of these test problems should be null because we're changing internals about the code, internal structure not necessarily may not be designed to change results.

And here would be a failure summary, and here I just have a -- you can link to the input deck and was it expected to fail. If there were any here that shouldn't have failed, they would be cued red. Those would be listed at the top of the summary. So it's a way the developer can immediately see what's wrong, they can right there and they can see okay, this is what's wrong and it helps their productivity and my productivity.

This would one of the runtime statistics pages. And here you can see, this is an update. I ran this for a previous version that we just checked in and the one before, and that was designed to get some CPU time back. There was a couple of places in the code where we noticed we could get some quick -- some better results for runtime. It wasn't acting significant, like we were talking about this morning.

At any rate, you can see that here. Across the entire suite, 708 test problems, it actually got better. And a few got worse. But if you look at a lot of these numbers, you're going to see they're going so fast that it may not mean too much to you and that's why maybe 60 are worse because you're in the level of -- you're below the resolution of the processor.

DR. FORD: Isn't runtime platform dependent?

MR. MURRAY: Yes. This is all on the same platform.

DR. SIEBER: Do you have different times or do you run the old case and the new case together, or do you have different times for different platforms?

MR. MURRAY: They're run on the same platform.

DR. SIEBER: Okay.

MR. MURRAY: I mean, they're run consecutively. I'll run the old case maybe two days ago, the new case to be run today.

DR. SIEBER: Okay.

MR. MURRAY: But it's a dedicated machine. Nothing else ever gets run on that. So I'm guaranteed 100 percent. And, yes, there will be variation just from, you know, operating system nuances that may also lead to these -- that you might see some test problems get a little worse than better.

DR. SIEBER: If your machines are like mine, after five years it's a new machine.

MR. MURRAY: Yes.

DR. SIEBER: And so what do you do? Rebaseline everything or --

MR. MURRAY: Yes, that's what I have to do. In fact, I mean this year this machine that I'm running on is due for replacement.

DR. SIEBER: It sounded like it.

MR. MURRAY: I'll just have to rerun it.

DR. SIEBER: Going to move up to Pentium IV.

MR. MURRAY: Yes.

DR. SIEBER: $800.

MR. MURRAY: Well, the machine I'll buy will be $25,000. But yes, these four process machines, the cost gets -- grows expediential the more processors you add.

DR. SIEBER: Okay.

DR. RANDOM: You've already covered this, but where did all these tests come from? Developer's tests?

MR. MURRAY: Yes, developers put them together when they submit their updates. A lot of times, I don't know if you were here when I said it, a lot of these decks are just permutations of existing decks, too. And so -- I mean, they'll give it a different name. And when you see this, R Power Test 12 and 2 is probably the same input deck with a little bit different input.

But, you know, if a test isn't there to address some feature that the person added, then they have to create a new one. And it's sort of my job to make sure that they're adding those test problems.

And one of the perpetual problems I've always had is making sure there's enough test problems. Because I'm not sure who indicated it before, you really want to do some -- you sort of want to assure you're testing every flow path. Well, I mean a lot of updates, two problems isn't enough to test every flow path. We might need ten or 20. And it's sort of my job to make sure that there are enough tests.

DR. RANDOM: Well there are some software packages available to determine how much of the code has been exercised in any given run.

MR. MURRAY: Yes. Our visual Fortran environment has that. And when I got my master's thesis, I'd done exactly that for TRAC-B.

DR. RANDOM: Yes.

MR. MURRAY: And I created a set of scripts that drove this visual Fortran profiler and allowed you to assess across a wide number of test problems the total test coverage. But that has not been incorporated in here. It's an extra level of effort.

I know what I did to write those scripts. I know what I did to write here. Eventually I'll find the time to sort of incorporate them. But it's a trivial exercise at this point.

DR. RANDOM: There are also packages, I guess maybe the developers know better, but you can find dead code or --

MR. MURRAY: Yes.

DR. RANDOM: Would never be reached.

MR. MURRAY: Yes. That's different. And I haven't -- that would be something, an additional some tools you --

DR. RANDOM: Compilers find that kind of thing? Some compilers will.

DR. MAHAFFY: This is John Mahaffy.

We have run such packages during the last five years at various intervals. It's not something to consider frequently. But particularly in the really large change zones among the code structure and database, there was that kind of exercise it went through to try to locate dead codes, just get rid of it. There's probably dead code now.

DR. SIEBER: If it's truly dead, why do you care?

DR. MAHAFFY: It's just --

DR. SIEBER: It's clutter?

DR. MAHAFFY: It's clutter. It's back to our readability requirement.

DR. SIEBER: Yes.

DR. MAHAFFY: We try to get the stuff that shouldn't be there out to make the code more readable.

MR. MURRAY: This page is just meant to sort of show, these are the sort of metrics we're looking at on a global scale to see have you made the results better or worse within the context of this test suite and what these tests are meant to show. And this was a useful update because it did show, yes, we made some updates that were intended to get back a little bit of runtime in our initialization portion of the code runs. And you can almost say, yes, well we did. Because you wouldn't have seen this big difference. In a normal test run that doesn't effect runtime, I usually get about a 50/50 split here. You know, some run a little worse, some run a little better because the test problems are so quick running you lose resolution of that just from OS overhead.

And finally, there's the difference results. And we look at -- we test for significant differences which I sort of define as -- in our output file. Because our output file the numbers don't have -- go out to full precision. Any file size that's greater than about 20k is usually indicative -- that's 20 kilobytes. That's usually indicative of some pretty major flaws.

And the text in blue indicates minor or cosmetic differences. Sometimes you just might change the way some -- you might fix a typo in an output file. And you usually don't see those kinds of differences need the file sizes greater than 20 kilobytes. That's just based on experience.

And text in black indicates non-null differences in the message file.

And you can see here, we have some differences in our parallel test problem, but they're very minor. And actually, if you would going looking at these output files, you're just going to see it's the CPU time that's reported. So there's nothing -- I see these four problems come up different every single time I run the suite.

Our robustness and assessment frameworks. We rely on the same common underlying toolset for both, because I'm essentially doing the same thing. It's just a matter of scale how many tests I'm really looking at.

It's going to be modular and test suite driven. The inputs are grouped by either test facility, the assessment project or end all functionality.

The input data that we use for defining the assessment cases is collected and organized through a spreadsheet using a tool we call AV script.

And it's for auto validation script. It's a four dimension matrix of input data where you define on separate worksheets of your spreadsheet cases you want to run, the figures you want to generate, the data sets you want to plot. And if you're going to define figures and merit, this helps you define the input to other tools that we use to generate those figures of merit.

Just to note, we use a spreadsheet, but the input is actually really text files. It's not tied to anyone specific spreadsheet. It's just the spreadsheets used as an easy way to collect the data.

CHAIRMAN WILLIS: Well, other organizations have a similar process.

MR. MURRAY: Yes.

CHAIRMAN WILLIS: So other organizations.

MR. MURRAY: Sure.

CHAIRMAN WILLIS: I think Sandia has, for instance, a whole thing about how they test codes. Is this all done from scratch or is it much the same as what everybody else does?

MR. MURRAY: Most of what I've done here is sort of, I've just been collecting knowledge over the last five years. I mean, like -- the ones we use here, are ones developed me. This AV script was actually by ISL for doing RELAP assessments. The regression set in its current incarnation, was really driven by me pulling it altogether. Like I say, the runtime stat8istics was really a feature John added to the code that I then took a lot of the work he did with some scripts and stuff and modified them.

CHAIRMAN WILLIS: So you're really building on NRC experience, nobody else's?

MR. MURRAY: It's all NRC experience.

And again, it's got a web-base summaries for viewing results.

On this assessment for our robustness suites, we can either drive RELAP5 TRAC-B or TRACE cases. You know, it can automatically generate your figures for you. And you can usually look at time history plots, parametric plots or axial plots. All kinds of different plot formats that will allow you to import those formats into your reports and things like that.

We filter our data sets through a program called ACAP to generate out quantitative figures of merit. We're not doing that now in the process that I've outlined, but the ability is there once we understand things better.

And by default, it will only run and

execute files that are out of date. So if you change an input deck, then it reruns it. But if you don't, it leaves it alone so that you have the ability to really fine tune things without having to rerun everything.

And I'll go through these pages very quickly. This is just sort of our spreadsheets that show the configuration information about your run and where executables are located and things like that.

I can override a lot of this information from the command line as I'm running my tools so I don't have to keep coming into these and updating for every version of the code.

This would define cases that you tag with ID numbers, the input name and the code version that you're going to run.

The mnemonic value for a version. And is a TRACE run or a RELAP run, it's location.

And what you're going to plot. This is actually what figures you've defined and things like titles and all that information you need to create a figure. Scale, axis and titles, and things like that.

And then you have one last page where you define exactly what you're going to plot from which code version you're going to get that information and on which figure it's going to sit. And then some information about that line so that it puts it in a legend so you understand what you're looking at.

Once you've run through the system, you've run your cases, it's generated your figures, it'll pull up this web-based summary. And it'll tell you the code version ID. And in this case it's telling you to look at the Excel input, but that won't always be the case if you feed it a specific version of the code.

DR. SIEBER: I take it all this stuff runs under Windows?

MR. MURRAY: Windows or UNIX.

DR. SIEBER: UNIX.

MR. MURRAY: And at the very top, assuming that you have cases where you've generated figures of merit, I'm going to show some example. It'll tell you, okay, how many figures of merit did I generate across my entire suite of test problems. How many are worse than before, how many have improved and how many are unchanged. And then it gives some summary of each test suite.

In this case, I've got a LOFT test suite that has a LOFT program that I've run. And some level tracking test suite that's targeting the level tracking model. These are just some samples that I've started putting together.

And it gives you the figure and figure of merit results so you can quickly look at the figures and have what's called a Figure Quicklook that is sort of a page that shows you all the figure one after another.

And you get your summary of results. And it shows you the figure. If you click on the link, it takes you to that particular figure so you can look at it.

In this case, you can see I've artificially generated a couple of figures of merit just to show that the process works. These two particular figures are actually to -- well, I can show you.

There we go. Resolutions a little off.

Here to give you a Figure Quicklook page. You click on. You just do the figures one after the next of different plots.

DR. RANDOM: Will that read different plots?

DR. SIEBER: Yes.

MR. MURRAY: What?

DR. RANDOM: You have three cases, is that right, or --

MR. MURRAY: In here?

DR. RANDOM: Well, that's one example.

MR. MURRAY: This would actually be one particular case I just ran just with a modified code.

DR. RANDOM: In what, the three different parameters or --

MR. MURRAY: No. It's just liquid temperature and in two different locations of the vessel.

MR. BANERJEE: What was the first reading?

CHAIRMAN WILLIS: Is it supposed to be that green one, which is really -- is that the right answer?

MR. MURRAY: Yes, this is the right answer.

CHAIRMAN WILLIS: Okay.

MR. MURRAY: These two have level tracking and they all delay each other, and these two have been the upwind solution or with the old level tracking for--

CHAIRMAN WILLIS: That's pretty --

DR. SIEBER: Could you go back to the first frame.

CHAIRMAN WILLIS: It's pretty poor --

MR. MURRAY: Looking at the results --

DR. SIEBER: That one. All right. That's pretty noisy --

MR. MURRAY: This flow -- tracking. And now this yellow and green right here -- and you don't see all the noise. But it sort of demonstrated better results.

A developer would look at this and if they saw that, okay, they started to see something going out here where it didn't used to happen, they would say oh wait a second. Something's happened.

DR. RANDOM: I'm wondering how much noise there is in the green and the yellow. It's not clear.

MR. MURRAY: No, it's not. This was based on -- I pulled this together based on Birol's thesis. These are cases he ran for his thesis. Now, a lot of these cases I don't think would have been useful.

In this case, I would never be going and comparing upwind the level tracking. I'm interested in level tracking for this version, level tracking for a previous version. And that's really sort of what I would be looking at.

DR. RANDOM: Well, that's why he wanted to look at the level tracking results. I mean, they apparently don't completely get rid of the problem.

MR. MURRAY: There's other ways. You have to read Birol's thesis to really understand. That's not my sphere of expertise.

DR. FORD: But what you're trying to do there is to see -- this compares stability of before and after modification?

MR. MURRAY: Yes.

DR. FORD: Is that right?

MR. MURRAY: Yes. And presumably if you have data, you would be comparing back to data. For that particular case, it was a contrived case for his thesis.

DR. RANDOM: And the cold water being pumped into a pipe and steam condensing on the top.

MR. MURRAY: Yes.

DR. FORD: When you look at the one with the temperature time --

MR. MURRAY: Yes.

DR. FORD: -- you had a square weight and you said that's the right answer and the other was --

MR. MURRAY: Well, that was a level tracking answer versus an upwind. I mean, you're in some sense trying to show that level tracking is better than before or better than no level tracking.

DR. FORD: Well, could you go to that

particular graph?

MR. MURRAY: Sure.

MR. BANERJEE: As soon as you show data, you're in trouble.

MR. MURRAY: Yes. I'll remember that next time.

CHAIRMAN WILLIS: Keep talking with words.

DR. FORD: In your job --

MR. MURRAY: Yes.

DR. FORD: -- what conclusion would you come to based on that?

MR. MURRAY: At this point? If I ever looked at this figure, I would say okay I haven't changed any results for the code. Because -- actually, no. I wouldn't even come to that conclusion at all with this because I'm not comparing base to modify it at all.

DR. FORD: What are you comparing?

MR. MURRAY: This is for two different locations in the model.

DR. FORD: So you don't have just before and after --

MR. MURRAY: Well, this particular plot is not a before and after.

DR. FORD: What are we seeing then? What

are we seeing.

MR. MURRAY: Ask Birol, I'm not really sure.

DR. SIEBER: You should never have put these up here. I mean, you'll never get out of this now.

MR. BANERJEE: Now we have -- believe me. Okay. Quickly move on.

DR. SIEBER: No, let's see some more.

MR. BANERJEE: Ideally, no data. Words, you won't get any questions.

MR. MURRAY: Well, the one thing I do want to show --

CHAIRMAN WILLIS: The manometer one.

MR. MURRAY: This being my life here, I guess.

If you look, this would have been a manometer problem. Without level tracking, you saw the damping.

CHAIRMAN WILLIS: With no damping?

MR. MURRAY: Yes. Or with damping.

CHAIRMAN WILLIS: But there's no --

MR. MURRAY: Yes.

CHAIRMAN WILLIS: There's no damping in the analytical value version?

MR. MURRAY: That's right. That's right.
MR. BANERJEE: So there's no --
DR. RANDOM: One has level tracking and the other one not?
MR. MURRAY: Yes. Well, no, no, no. That was no level tracking to analytic. This plot here would have been with level tracking versus the analytic solution or a base in modified version, okay. And you see in this -- between the two I haven't changed results at all.
If I saw the blue line suddenly shifting here or something, that would be a visual cue to me say, hey something's wrong. If I don't understand it specifically, that's when I go back to the developer. If it's something within my sphere of knowledge or I understand enough about it, then I'll --
MR. BANERJEE: So how does level tracking get rid of numerical viscosity?
MR. MURRAY: You're out of my --
MR. BANERJEE: Is this too good to be true?
MR. MURRAY: You're out of my realm.
MR. STAUDENMEIER: Save that for a later presentation.
MR. MURRAY: But the point I'm just trying

to show is the process. Because ideally I might have those two solution sets next to each other, and you would have seen a figure of merit went from very poor, one being perfect fit and zero being no fit or a random, basically. You see the figure of merit go up. And now this was -- I contrived a case and used our tool in a very contrived way. So these numbers really mean nothing, except just to show you that it's a process that's there that we can use.

CHAIRMAN WILLIS: I thought the -- didn't change --

MR. MURRAY: No, no, no. This is one --

CHAIRMAN WILLIS: From one column and two is the same.

MR. MURRAY: This was versus analytic. And it wasn't --

MR. BANERJEE: The figure 19 and 20.

MR. MURRAY: Yes. The figure really didn't change in the base code versus analytic it were here. In the modified code versus analytic is here.

CHAIRMAN WILLIS: So it's the same?

MR. MURRAY: It's the same.

CHAIRMAN WILLIS: Right.

MR. MURRAY: So I would be looking at this percent of change here. So there's something wrong.

CHAIRMAN WILLIS: So nothing is being gained?

MR. MURRAY: Yes.

DR. MAHAFFY: This is John Mahaffy.

Let me tell you what's been gained in terms of this analysis, and that is time. That column that he's showing you where he's got a zero difference between two figures of merit, that says that the keeper of the code does not even have to look at the figures to decide about changes between code versions. They're both behaving in the same way. They're giving you the same figure of merit. Nothing important has happened here. I can move on and concentrate on comparative results somewhere else, otherwise Chris will quit his job the first time he has to go through a full set like this and they'll be on to somebody else.

You're trying to minimize your human intervention and focus it on the cases that are obviously problems where you've got to go back to the developer and come up with a resolution.

MR. MURRAY: In some sense, I'm relying on the fact that there's already been somebody whose gone through this assessment before this and really has done that analysis and shown, okay, what's going on here, I understand why this is happening. There's code deficiencies, there's not. And once they've done that, now I have a framework that we don't lose that. And from that point forward at least I can make sure that we're not moving backwards.

And I think that's always been the point. You know, Joe's going to get up tomorrow and say that he's got this new reflood or interim reflood model that's performing better than before. And in the past I've seen cases when I was at Penn State, we added a new reflood model based on the Neptune facility to TRAC-B. And we fixed -- we had this set of test cases and they're going to run my test suite, and there's a whole bunch of others that now are seeing temperatures go off the roof and off the scale. And how do you get a --

CHAIRMAN WILLIS: That would explain what happened with some of these more complicated problems.

MR. MURRAY: Yes.

CHAIRMAN WILLIS: You fix something to fix -- say, Penn State and you find it does a nice job on something else.

MR. MURRAY: But with this framework what I'm arguing, I guess, is that now we know what's happening across a whole wide spectrum of problems and across all of our assessments. You just haven't lost anything in the process of going through these assessments.

MR. BANERJEE: Can we see the top of the figure? What these columns mean again?

MR. MURRAY: Oh, sure. Sure. This was just a figure of merit. A version --

CHAIRMAN WILLIS: It's before and after, right?

DR. SIEBER: Yes, right.

MR. MURRAY: Base versus sum deference--

CHAIRMAN WILLIS: Well, it's before and after, isn't it?

MR. MURRAY: Yes. Modified version versus sum deference. And then you have links to the output files so you can go and link and look at them directly.

CHAIRMAN WILLIS: What are all these blanks in the table? Because nobody's done that?

DR. RANDOM: A point-by-point comparison or --

MR. MURRAY: Yes. I'm going to go into that a little bit. Like I'm trying to say, we don't -- there's a whole lot of magic involved in how we get those and we're not yet using them. But in some areas we want to try to look into.

CHAIRMAN WILLIS: Magic?

MR. BANERJEE: May I ask what this figure of merit is?

MR. STAUDENMEIER: That's most problems we haven't developed figures of merit yet for. There's this ACAP tool which is automated code assessment program. And what's going to have to be done is developers doing developmental assessment with a given problem are going to have go and use ACAP and come up with something that represents a good figure of merit for that problem and define it. And when that happens, then you'll have a base figure of merit to compare to. Right now Chris is just showing a demonstration how it works on a contrived problem. But we haven't gone through and defined figures of merit yet to take advantage of this automated comparison yet. But that's work that will be done in the future to help us do this more quantitative comparison between results and not less qualitative. But in things that will flag things to Chris once figures of merit are available and developers submit changes. Our developers are doing their own changes in code testing, it'll flag areas to them where they've changed things greatly where maybe they didn't want to change things greatly, that they have to go back and look at.

So it's more a productivity enhancing tool for both developers and for Chris in terms of maintaining the code.

DR. SIEBER: Once you get that, you can further automate this by having it generate a nasty email to the developer.

DR. MAHAFFY: This is John Mahaffy.

Let me make one more comment on this process that Chris alluded to. It's actually deeper than that.

As a developer, as he said, I have all these tools. If I'm behaving responsibility I've run all this, I've scanned those figure of merit changes and in my report to Chris when I submit the update, I've told him where I've seen the changes and I've explained why each of those changes has occurred within the contest of my update. His only job then, if I've done that right, is to go in, rerun on his machine and confirm what I have seen.

CHAIRMAN WILLIS: But not everything. They didn't have to confirm everything.

DR. SIEBER: Just spot check, presumably?

MR. MURRAY: If I'm going to automate it, there's no reason why I can't --

DR. SIEBER: Yes.

MR. MURRAY: I've only balked --

CHAIRMAN WILLIS: In development. I think it might -- if you have too much of a checking process, you might --

DR. MAHAFFY: Again, it's automated.

CHAIRMAN WILLIS: Yes. If it happens quickly enough, it's okay.

DR. MAHAFFY: It may be tough for some of the young kids around here. But believe me, having gone through the process 25 years ago, this is a breeze.

CHAIRMAN WILLIS: Well, then age optimum for this.

MR. MURRAY: Yes.

DR. SIEBER: Well, actually, this is not new ground. Software development occurs in many fields in testing this way. And so it's not unique. It's pretty comprehensive, though. I like that.

MR. STAUDENMEIER: Yes. I think it's a step beyond anything that's been done in the past with any of the NRC codes previously.

CHAIRMAN WILLIS: How you going to put this in the right guide for the vendor codes?

MR. STAUDENMEIER: We can't specify what testing methods they follow, just that they have testing methods that meet Appendix B.

MR. MURRAY: I'm now hesitating to go ahead.

CHAIRMAN WILLIS: Well, quickly.

MR. MURRAY: With ACAP that's a tool that was developed by Penn State a number of years ago for trying to help us generate these figures of merit. It gives the developer numeral statistical measures as well as data conditioning techniques to apply to data sets to sort of try to come up with a figure merit.

Now, how the developer uses the tool and how they define a figure of merit based on which statistical techniques is really I think the question. And to me it's, because I'm not a statistician, it is a little bit magic; that's why I used that word. For somebody whose got a good grounding in statistics, there's going to be probably some good basis for why you would choose certain techniques to certain data sets.

I think we need a period of time to grow comfortable with how the numbers really behave. The process that was outlined by the developers of this software called for really a set of experts to sit down and look at every case and say, okay, this is how we define a figure of merit for those cases and then you use that from that time forward.

Further study improvements. With respect ACAP's time series data analysis techniques, we've got a couple of different techniques for doing time series analysis. Most of them are for parametric type or axial plot type data but the curves are smoother and the statistics don't change with time.

And I just gave a listing, there's all the different techniques. Time series data, really, wavelet transform and the bill time is some of the work that Daria had done.

CHAIRMAN WILLIS: The colors don't mean anything there?

MR. MURRAY: I pulled this out of a different one, and I'm not really sure of what the colors mean. This is off a different presentation.

I know that these are just beta conditioning and they're a different color. So maybe there is some scheme here.

MR. CARUSO: So these are all different methods that you're considering?

MR. MURRAY: These are all methods that are built into this software tool that I'm going to show a snapshot of.

MR. CARUSO: They're built in?

MR. MURRAY: Yes. And there's more we could add. Certainly part of the recommendations was we needed to explore more. This wavelet transform type statistics. And we're using a morelet, what they call a morelet transform here. There's other kinds of transformers you can do that ought to be probably looked at.

MR. BANERJEE: So what is the transform doing? You're trying to get rid of noise or what?

MR. MURRAY: The best way I could characterize it, because I don't really know too much about it, it's an enhanced fourier transform. And it's able to I think characterize more local effects --

MR. BANERJEE: I guess you could go back and educate me. Why would you need to use a fourier transform on the data? What are you looking for?

MR. STAUDENMEIER: As I said before, Chris isn't covering that in his presentation, but it depends on the data set. Your analysis method would-- I mean maybe you're looking at also fourier data for like instability or something like that. Like Frigg instability tests or things like that and want to look at frequency compared to data or frequency compared to code versions that you're predicting and a fourier transform is an actual way to transform the data. For other data it may not be. Maybe something is like a pressure decay. I don't know what kind of method you would analyze that, but some kind of a measure that would look at how far off the pressure curve is--

CHAIRMAN WILLIS: But it tells you how wiggly things are. Look at this data here. These are different predictions or something?

MR. MURRAY: Sure. Sure.

CHAIRMAN WILLIS: Some are wrigglier than others.

MR. MURRAY: Yes. I mean, this blue line in the middle was the reference set.

CHAIRMAN WILLIS: Right.

MR. MURRAY: And all these others were some -- off of that. And, you know, this was actually data right out of Doria's paper on his --

DR. RANDOM: Well, I think that you can use the FRT. That's the only one I'm familiar with here. Is that you can look at the different frequencies of the function and see whether they're growing or decaying or shifting, or what is going on. That can be fairly valuable in the numerical method to see that as you approach the critical wave length that they are indeed decaying.

CHAIRMAN WILLIS: Might introduce some sort of a periodicity.

MR. MURRAY: What with?

CHAIRMAN WILLIS: Nodding. If you have the right of nodding which was tuned to some wave propagation.

DR. RANDOM: Well, as you node more finally or more nonuniformally, I guess, you can introduce different frequencies, of course.

CHAIRMAN WILLIS: Right.

DR. FORD: What is this particular graph showing us?

MR. MURRAY: This is with the -- it just is showing that you would have set of -- if you're using the tool, you would input some data sets and you would choose off of this palette some set of different statistical techniques that you could even weight them, you know, by some factor and use those to generate figures of merit that would get output here. That's all that that's -

DR. FORD: Is that not just using a statistical approach that gives you the answer you want? Is that an unkind --

MR. MURRAY: Well, certainly you can -- my feeling is that you can do that. I mean, those figures I showed you in the webpage results, I tuned that to be the answer I wanted it to be. And that to me is the danger of this, you know.

CHAIRMAN WILLIS: You don't have to understand this at all. You got these different curves and one of this measured or something?

MR. MURRAY: The blue is the preference set.

CHAIRMAN WILLIS: And are these measured?

MR. MURRAY: Yes.

CHAIRMAN WILLIS: And the red is a reasonable approximation to it, the wiggly one?

MR. MURRAY: Yes.

CHAIRMAN WILLIS: The green is sort of out of --

MR. MURRAY: Way out.

CHAIRMAN WILLIS: Yes.

MR. MURRAY: And what this was showing was that -- let's see for data set one, which is the red you had an overall figure of merit of 1.94. And here you had some data set 2, which is yellow, which is over here. And your figure of merit was a little lower. And this is saying okay, that looks like a better fit to the data even with the flow behavior than this yellow. But what those mean, I couldn't tell you.

CHAIRMAN WILLIS: You can say did it match the right wave transform, you might conclude the red was really bad because it puts in a wave which isn't there.

MR. MURRAY: You might.

CHAIRMAN WILLIS: And so I'm not sure you've got the right --

MR. MURRAY: Well, you're right. And that's I'm trying to tell you that we're not using it now and --

DR. RANDOM: Are those real code outputs or they just idealizations of different terms in a --

MR. MURRAY: No. This was contrived data that d'Auria made up for his paper on --

MR. BANERJEE: Right. But I think going back to the principle it depends on why you're doing this. Imagine that red is the flow rate.

MR. MURRAY: Yes.

MR. BANERJEE: Okay. And the blue is the actual flow rate. Then adding those oscillations could give you a much different heat transfer coefficient than a flow without oscillations. Therefore, I mean just using it without knowing what the end product is can be very dangerous.

I mean, a figure in isolation has no meaning in some sense.

MR. MURRAY: That's right.

MR. BANERJEE: You have to look at the whole pattern.

MR. MURRAY: And you're relying on -- that's why we group these assessment cases here. You're presumably looking at a lot of different cases.

MR. BANERJEE: So it's more like a pattern recognition problem then taking isolated figures of merit.

MR. MURRAY: Yes, that's right.

MR. STAUDENMEIER: As I said, developers will have to go through case-by-case and come up with figures of merit that work well for that specific test case. Then once they're defined, that can be automated and run with looking at those figures of merit for that specific test case. But there is human knowledge that needs to be put in into making sure that you've chosen good figures of merit for that specific test case that you're looking at.

CHAIRMAN WILLIS: -- calculations they would run codes like this and you got all these different answers. And then the bottom line would be it only changes PCT by two degrees, therefore it doesn't matter.

DR. SIEBER: Right.

CHAIRMAN WILLIS: So we don't care about these wiggles or these differences.

MR. STAUDENMEIER: Actually they do care about that stuff in Appendix K cases. Even when they change it, they look at more in-house -- I mean what they submit to the NRC is that it's a change to this amount because of reporting requirements in the law. I mean, they're responsible developers and they're going to be looking at the important things in-house when they change the code, presumably. I mean, there's cases where that hasn't happened and we've come down hard on them. But generally I think they're responsible people out there that want to do the right thing. And what you see is a two degree change.

I mean, people are looking to make sure that they're getting qualitatively good results even in Appendix K. Appendix K calculations contain lots of realistic models in them. It's only some models are specified by regulation that have to be in a certain way.

CHAIRMAN WILLIS: Well, usually Appendix X models we see is the great stacks of output of wiggles with no comparison with the data of any sort whatsoever.

MR. STAUDENMEIER: Well, it depends. I mean, some are better than others. The ones that haven't changed since 1973 can be like that, but others have been updated quite a bit since then and do quite well against data.

MR. BANERJEE: I guess it's going to be very difficult to come up with figures of merit. I mean, to me it seems that even a very experienced person will have trouble. And any new developer or getting into relatively recently, I don't see very easily they can come up with something.

DR. MAHAFFY: This is John Mahaffy.

And I wrote the original proposal for this ACAP thing. Just to summarize, it's in line with everything that's been said here. ACAP is a tool for you to survey various figures of merits relative to a specific set of data. But as he said before, let me emphasize, it was from the beginning part of the description of this ACAP is, it's just a tool. You have to have a group of real experts sit down, think carefully about the meaning of the data and the possible ways it can change and select a relevant one or more figure of merit generator from this and then lock into that for future assessment.

MR. CARUSO: Do you think that this would be useful in helping develop the infrastructure to support risk-informed regulation?

DR. MAHAFFY: It might. I mean, I don't have enough opinion on risk-informed regulation to speak intelligently on that.

CHAIRMAN WILLIS: So this is your last slide?

MR. MURRAY: The last one.

DR. SIEBER: It's better than the previous one. Let's controversy.

MR. MURRAY: In terms of future improvements for this framework, things I'd like to do would be automatic regenerating our DA manual. Once you've got the figures, importing those into a manual that's got a well-defined structure would be really useful to us. The only thing you've got to do then is have somebody make sure that the conclusions that are reached in the manual are in line with the new figures that you generated. But you cut down a lot of work there. I think that there's advances now in software technology that this is really possible.

I'd like to modify our robustness and assessment tools to better exploit the multi-processor environments. I run our regression suites on an inter-multiprocessor set. My tools aren't yet capable of handling that on the assessment and robustness testing side. They need to be advanced for that.

And extend, again, ACAP with more ways to generate figures of merit.

Fold qualification of code uncertainty into the process. You're going to talk about that a little tomorrow with Steve. You're talking, I guess, a little bit on measures of merit, I guess.

Further improve methods for presenting and mining massive amounts of information generated during a typical assessment.

CHAIRMAN WILLIS: Well, someone asked about risk assessment or risk-informed. You really need to know uncertainties to do a good job.

MR. CARUSO: You'd have to do that.

CHAIRMAN WILLIS: The risk that the code may be wrong so that you may be giving an answer which is misleading.

MR. MURRAY: And in this vein of improvements, something would be a better tracking of runtime. Right time my tools really aren't looking at the code runtime or the assessment robustness sets. That's something you really want to know there as well, and they need to be extended.

CHAIRMAN WILLIS: Okay.

MR. MURRAY: That's it.

CHAIRMAN WILLIS: Now we're going to meet tomorrow for about the same length of time.

DR. SIEBER: Yes. Everything is going to be in the afternoon.

CHAIRMAN WILLIS: Is there anything else that the Subcommittee wants to do today?

No. So I think we're ready to recess. We'll recess until 8:30 tomorrow morning.

Thank you all very much. See you tomorrow.

(Whereupon, at 4:51 p.m. the Subcommittee was adjourned, to reconvene tomorrow morning at 8:30 a.m.)