

Date – 09/18/2012

Attendees: CJ Clark, Adam Cron , Adam Ley, Bill Bruce, Bill Eklow, Bill Tuthill, Brian Turmelle, Carl Barnhart, Carol Pyron, Craig Stephan, Dave Dubberke, Dharma Konda, Francisco Russi, Hugh Wallace, John Braden, John Seibold, Josh Ferry, Ken Parker, Peter Elias,

Missing with pre-excuse: Wim Driessen,

Missing: Lee Whetsel, Matthias Kamm , Mike Richetti, Neil Jacobson, Ted Cleggett, Brian Erickson, Scott Wilkinson, Jason Chodora , Roger Sowada, Kent NG, Sam McMillan, Sankaran Menon, Ted Eaton, Heiko Ehrenberg, Jeff Halnon, Rich Cornejo, Roland Latvala,

Agenda:

- 1) Patent Slides and Rules of Etiquette
- 2) Use LiveMeeting “Raised Hand” to be recognized and take the floor
- 3) Draft submitted for ballot on 9/14 (per WG vote). Ballot open until end of October.
- 4) Discussion of use of iCall outside of iProc in top level and application specific files.

The grammar in the draft prevents the description of iCall outside of an iProc. We have defined PDL to document procedures and create a library of (hopefully) pre-validated IP and IC iProcs. However the ‘environment’ the use of an IC or the IP may vary widely. From some perspectives it is useful to be able to document how (looping for instance) and what (which iProcs) should be called and in what order. IC supplied example files and the like currently are prevented from specifying an iCall.

Meeting Called to order at 10:40 am EST

Minutes:

Review Patent Slide – Slide Presented to the Group.

Solicited input from anybody who is aware of patents that might read on our standard.

No Response

Review of Working Group Meeting Guidelines

No Objections

Draft has been submitted for ballot on 9/14.

Open to 11:59 PM October 31st

iCall Discussion

Current Problem - Grammar in draft now prevents description of iCall outside of an iProc
Carl – what is the use mode? This is the same as what is stated in standard. PDL is documentation language for how to use test procedures. Standard has been very careful to stay away from tool issues. Seeing calling the top level is a tool issue.

CJ – iProcs are not procs. iProcs can't be entered on the command line. Must use an iCall.

Carl – once you have registered the PDL commands in a TCL environment. TCL script can include PDL commands. Can source TCL script and iSource a PDL file.

Carl – needs a starting file and entry point. No different here. When you bring it up, you can specify a TCL command to do what you want for the top level stuff, our job in 1149.1 was to document PDL commands and how to use test features. Not our job to define how a tool would define it.

CJ – what do you do with PDL 0?

Carl – TCL script can do initial iSource and iCall. Everything underneath can be PDL 0

Carl – Nothing in the standard from using a TCL file with PDL commands in it. No changes to the standard needed.

Carol – sees both sides of discussion. PDL defined today are library of documented procedures. Vendor may want to give example. Is it in TCL or PDL?

Not necessarily broken but would like to see it in a standardized view.

CJ – how do you do it in PDL0? Only required to use PDL0 and TCL is option.

TCL seems to be outside the specification of the grammar for PDL0

Carol – could give example in a reference manual style – do this then this then this

Tools won't be doing PDL0 for a while. (prediction)

Josh – always considered more on the scale of a DLL rather than an executable.

Are software systems out there that don't have entry points? Up to the implementer to pick where the entry point is. If we force each PDL to have a single entry point, Josh will be limited if he needs to share info between 2 different PDL files.

Carl – already a way to say –export on an iProc. Each –export is a potential entry points.

CJ – how do we document the entry points as users? No method now

Josh – that is tool provider and the PDL provider to show examples. Chip to chip dependant. And would not like that in a standard. Might be limiting.

Hugh – definitely need an entry point to indicate where the beginning is. But that may change. Could use a “main” and use that as your entry point.

Can have a bunch of library elements that describe what happens.

Where you start is a tool thing.

If we want to specify a file- that is way of indicating the top.

If you do it with iCall that takes away your ability to use parameters.

Would go with Carl viewpoint.

iProc can be called by TCL or called by themselves if compiled. Make a callable iProc

CJ – no restrictions on passing arguments to an iProc from an iCall

Adam L – concurs with Carl. Believes that it was deliberate that iCall was left from the definition. We don't have a naked iCall in any of our examples. Created a syntax that doesn't have the iCall and no examples to support the iCall.

Bill B- can be left as a tool vendor issue which is where it is now.

Willing to do that

To say the language doesn't allow a naked iCall is correct, but with a tool that is how you are going to get it evoked.

To make a solution that will solve many problems is difficult. The best we can do is come up with some sort of standardization that gets the ball rolling.

CJ Recluses himself while making comments, Carol takes over as Chair.

CJ – would be a mistake to need TCL to document the test. Already developed a mechanism called the iCall. Used to call a proc.

No command line that is practical because you would have to write iProcs on the command line. Vendors should be able to supply a sequence and order to run

All the examples presented have naked iCalls. They may not have been captured in the standard.

Want to specify what procs are being called in the example files. No problem passing parameters. Can be done in an iCall. If you don't supply an argument then the defaults are used.

In PDL0 there is no other choice.

CJ has returned to the chair.

Hugh – a way of handling what you are saying is nominating a procedure like main.

If you have main at the top level that it can be obvious what you want to execute.

If you have an iProc called main. If it is compiled than you can execute that.

CJ – main is a library element from the Vendor if he wants the customer to execute something all the time. Not an entry point.

Hugh – not aware that main was taken. Call it iProc entry point.

Carl – naked iCall not sufficient. In PDL0 there is no way of passing arguments to a naked iCall.

Hugh – makes a motion but it is not seconded

Bill B – probably need to go to an abstraction to solve this. Execution file that says what you want to do. List the procs, make it like a run file. Defines what you want to do for a run.

CJ – that was the intent of a top-level iProc.

Carl – can have multiple top-level iProcs

CJ – there is a top-level a grammar in the standard. And a single top-level grammar in the standard does not let you do an iCall. No way to specify an entry point.

Need a method to specify in the top-level the procs that need to be called to make the documentation complete.

Carl – that is called an iProc

CJ – than what do you do.

Carl – that is a tool issue.

Hugh – can you have more than one top-level file?

Carl – yes.

CJ – no restriction

Hugh – which one do you execute? That is the question

The Chair asks if anyone would like to make a motion regarding this.

Adam C – POI no one is making a motion to fix this.

Brian - *Motion that the PDL grammar be corrected to allow iCalls in top-level PDL files.*

Bill Bruce seconds.

Adam L – POI PDL grammar already allows iCalls in top-level PDL files. Some qualifying language is missing. Standard already meets the letter of the Motion.

CJ – spirit of the motion is that one can't have an iCall in the file either iProc_group0 or iProc_group1

Carl – just add outside the iProc to the motion.

Adam L – asking that the language be precise so we don't argue about it either

Friendly amendment “*outside of an iProc*”

Amendment accepted by Brian and Bill B

Adam L – POI at least 2 discussion of examples of top level in PDL that may lead one to believe that naked iCalls are permissible. Will provide that as background.

Discussion ended.

Question called

Yes 5

Bill B. Craig S.

Bill T. Josh Ferry.

Brian T.

No 5

Adam L. Ken P.

Carl B. Peter E.

Hugh W.

Abstain 7

Adam C. Dave D. Francisco R. John s.

Bill E. Dharma K. John B.

Motion doesn't pass. It was a tie.

Chair did not feel it be appropriate to vote. And the vote stands at 5/5/7

Adam L gives a brief overview on how to get to the ballot for the standard

IEEE standards -> eTools -> sign in with IEEE account/password (sign in with myProject) you will see Announcements.

Click on Balloting Tab.

Click on Manage myBallot Activity

To log your vote put vote in Vote column

Abstentions do count against. Needs to clear at least 75% non abstention participation.

Then you can add your comments.

Attached files can be added. Only can be attached through single comment submittal.

Offline Comment Submittal allows you to upload ballot comments. Can download 4 different templates that can be used to create comments. And then upload the comments in the file template that you downloaded.

Column labeled “must be satisfied” the only way to set this field with a Yes is to submit the comments or edit them when your vote is disapproved.

Meeting adjourned: 12:13 pm EST.

Summary of Motions Voted on

1 Motions voted on

Motion that the PDL grammar be corrected to allow iCalls in top-level PDL files outside of an iProc.

Result was a Tie. Motion did not pass

5/5/7

Next Meeting: 9/25/2012 10:30 AM EST

NOTES:

1149.1 working group website - <http://grouper.ieee.org/groups/1149/1/>

To Join the meeting

<https://www.livemeeting.com/cc/intellitech/join?id=2CQ2PQ&role=attend&pw=n%26d%5DNqX%284>

Meeting time: Tuesdays 10:30 AM (EST) (Recurring)

AUDIO INFORMATION

-Computer Audio(Recommended)

To use computer audio, you need speakers and microphone, or a headset.

-Telephone conferencing

Use the information below to connect:

Toll: +1 (218) 862-1526

Participant code: 11491

FIRST-TIME USERS

To save time before the meeting, [check your system](#) to make sure it is ready to use Office Live Meeting.

TROUBLESHOOTING

Unable to join the meeting? Follow these steps:

1. Copy this address and paste it into your web browser:

<https://www.livemeeting.com/cc/intellitech/join>

2. Copy and paste the required information:

Meeting ID: F9R6S6

Entry Code: k/d6<@M6j

Location: <https://www.livemeeting.com/cc/intellitech>

If you still cannot enter the meeting, [contact support](#).

NOTICE

Microsoft Office Live Meeting can be used to record meetings. By participating in this meeting, you agree that your communications may be monitored or recorded at any time during the meeting.