Result of email vote concluded on Wednesday May 11 2011.
All eligible voters except Francoise and Tom sent in their votes.
[9 mantis items: 3046, 2506, 3531, 3394, 2905, 3254, 3298, 3054]
PASS: 2935, 3254, 2905,
NOT PASS: 3045, 2506, 3531, 3394, 3298, 3054
(1) Mantis 3046 7 Yes 3 No
YES: Scott, Steven, Alex, Gord, Dave, Ray, Neil S.,
NO: Mark, Arturo, Neil,
(2) Mantis 2506 4 Yes 3 No 3 Abstain
YES: Scott, Mark, Arturo, Alex,
No: Gord, Dave, Ray
Abstain: Steven, Neil S., Neil
(3) Mantis 3531 9 Yes 1 No
YES: Arturo, Scott, Steven, Alex, Gord, Dave, Ray, Neil S., Neil,
No: Mark,
(4) Mantis 3394 9Yes 1 No
YES: Mark, Arturo, Scott, Steven, Alex, Gord, Dave,Ray, Neil S.,
NO: Neil
(5) Mantis 2905 10 Yes 0 No
YES: Mark, Arturo, Scott, Steven, Gord, Alex, Dave, Ray, Neil S., Neil
(6) Mantis 3254 10Yes 0No
Yes: Mark, Arturo, Scott, Steven, Gord, Alex, Dave, Ray, Neil S., Neil
(7) Mantis 3298 7 Yes 3 No
YES: Scott, Steven, Gord, Alex, Dave, Ray, Neil S.,
NO: Mark, Arturo, Neil,
(8) Mantis 3054 9 Yes 1No
YES: Mark, Scott, Steven, Gord, Alex, Dave,Ray, Neil S., Neil
NO: Arturo
(9) Mantis 2935 10 Yes 0 No
YES: Mark, Arturo, Scott, Steven, Gord, Alex, Dave, Ray, Neil S., Neil,

(1) Mantis 3046 ___Yes ___No
YES: Scott, Steven, Alex, Gord, Dave, Ray, Neil S.,
NO: Mark, Arturo, Neil,
Mark: I have some concern over the wording of the proposal. It says:
"If the dotted name fails to resolve in the scope of the randomize() with object class, it shall be resolved following normal resolution rules in the scope containing the inline constraint."
I thought we had changed or were planning to change the LRM to say if the first token of a dotted name resolves to a variable, but later tokens cannot be resolved, it is an error.
Arturo:
The term "strictly downwards manner" is not well defined. Also, the language is a bit awkward in treating dotted names specially when all it should say is that normal rules apply, which already deal with the first name of a dotted name. I do agree with the outcome of the examples shown in the Mantis.

[Gord]But normal rules do not apply. If the first name is not present in the target class, one does not want to do a hierarchical search upwards from the location of the class type definition. One wants to terminate the search immediately and start the search in the context surrounding the inline constraint, not the context of the class object type. That is why a clarification is needed.

Feel free to suggest better wording. But I don't think that "normal search" is going to be correct.
Neil: I agree with a lot of what Arturo mentioned.
To tell you the truth, I don't think this set of changes is even necessary.
I agree with the summary of the examples described in the mantis item, but it seems to me that the existing text leads to this conclusion.

(2) Mantis 2506 ___Yes ___No
YES: Scott, Mark, Arturo, Alex,
No: Gord, Dave, Ray
Abstain: Steven, Neil S., Neil

Gord: Given the additional changes going into 2506 from other comments and the overall size of the proposal, I would like to have a final proposal go through unchanged before voting in favor.

Dave: Would like a final review in a CC.
Ray: The BNF changes need to be replicated in Annex A. Also does there need to be a restriction on the visibility of functions declared within the scope of a cross.

Abstain: Steven, Neil S., Neil
[Neil: I didn't have time to review this one in detail.]

(3) Mantis 3531 ___Yes ___No
YES: Arturo, Scott, Steven, Alex, Gord, Dave, Ray, Neil S., Neil,

[Steven: I am not seeing blue text for the additions in A.8.4, though I see it for Syntax 11-8. If this isn't just a problem with my viewer, then this should be fixed.

No: Mark,
Mark: I have some reservations about this change.
The LRM describes 'null' as a special value. It is only
legal in a few contexts. There are no casting rules between 'null' and
integer types. It does not have a self-determined type. I am worried we need
more text to restrict where this can be used. Apparently it is already in
primary, so we probably already have this problem.

(4) Mantis 3394 ___Yes ___No
YES: Mark, Arturo, Scott, Steven, Alex, Gord, Dave,Ray, Neil S.,
NO: Neil
Neil: It isn't clear to me that this example is invalid.
Aren't both the existing text and the proposed text legal?
I agree with the change in the comment for dest2.

(5) Mantis 2905 ___Yes ___No
YES: Mark, Arturo, Scott, Steven, Gord, Alex, Dave, Ray, Neil S., Neil

Comments:
Arturo:
The sentence seems somewhat tautological: constraint_set is ..
constraint set. Perhaps we can find better verbiage – the intent is
to define a constraint_set as either a single constrain or a group of
constrains bracketed by curly braces. I'm voting yes because the same
language exists in implication so this is no worse.

(6) Mantis 3254 ___Yes ___No
Yes: Mark, Arturo, Scott, Steven, Gord, Alex, Dave, Ray, Neil S., Neil
Comment: [Arturo]The sentence seems somewhat tautological: constraint_set is .. constraint set. Perhaps we can find better verbiage – the intent is to define a constraint_set as either a single constrain or a group of constrains bracketed by curly braces. I'm voting yes because the same language exists in implication so this is no worse.

(7) Mantis 3298 ___Yes ___No
YES: Scott, Steven, Gord, Alex, Dave, Ray, Neil S.,
[Scott Friendly amendment: The new text should be in blue.]
[Steven: BTW, it would also make sense to allow 'this' in initializers on non-static properties. These are effectively executed in the constructor, which is a non-static method. And these initializers can presumably refer to other class properties using the normal scoping rules, and this is an implicit reference to 'this' already.]

NO: Mark, Arturo, Neil,
Mark: What about the use of 'this' in randomize with blocks.
Doesn't that need to be included in this section also?
Arturo:
Must mention constrains: A friendly amendment (which may need discussion):
The this keyword shall only be used within nonstatic class methods, constrains, inlined constrain methods, or embedded covergroups (see 19.4) ;
Neil:
By making this change, it creates a new problem in the existing text.
The following sentence assumes 'this' can only be used in a class subroutine.

"The this keyword denotes a predefined object handle that refers to the object that was used to invoke the subroutine that this is used within."

Friendly amendments:
The new text should be in blue and not underlined.
The mantis item should also be shown at the top of the proposal.

(8) Mantis 3054 ___Yes ___No
YES: Mark, Scott, Steven, Gord, Alex, Dave,Ray, Neil S., Neil
NO: Arturo

Arturo:
Although allowing $countones and $onehot in constrains is reasonable. There is no explicit proposal that describes the semantics associated with such constrains. For example, do these function calls partition the constrains (as with other "allowed" functions) or are they to be treated differently? If the semantics are the same as for other function calls, the LRM can simply state that constrains will allow functions *__and system function__* that adhere to the existing limitations.

(9) Mantis 2935 ___Yes ___No
YES: Mark, Arturo, Scott, Steven, Gord, Alex, Dave, Ray, Neil S., Neil,
[Steven: I haven't actually seen the fixed version, but I will take Shalom's word for it.

Mark:
(1) Mantis 3046 ___Yes _X__No
http://www.eda.org/svdb/view.php?id=3046
[Dotted names within inlined constraints]
[proposal: 3046_inline_dotted_names.pdf]

I have some concern over the wording of the proposal. It says:

"If the dotted name fails to resolve in the scope of the randomize() with object class, it shall be resolved following normal resolution rules in the scope containing the inline constraint."

I thought we had changed or were planning to change the LRM to say if the first token of a dotted name resolves to a variable, but later tokens cannot be resolved, it is an error.

(2) Mantis 2506 _X__Yes ___No
http://www.eda.org/svdb/view.php?id=2506
[Non-trivial coverage space shapes and joint conditions are difficult to specify with covergroups]
[proposal: Proposal for Mantis 2506_v5.pdf]

(3) Mantis 3531 ___Yes _X__No
http://www.eda.org/svdb/view.php?id=3531
[null should be allowed in constant expressions]
[proposal file: 3531_null.pdf]

I have some reservations about this change. The LRM describes 'null' as a special value. It is only legal in a few contexts. There are no casting rules between 'null' and integer types. It does not have a self-determined type. I am worried we need more text to restrict where this can be used. Apparently it is already in primary, so we probably already have this problem.

(4) Mantis 3394 _X__Yes ___No
http://www.eda.org/svdb/view.php?id=3394
[invalid example for dynamic array]
[proposal: 3394_illegal_example.pdf]

(5) Mantis 2905 _X__Yes ___No
http://www.eda.org/svdb/view.php?id=2905
[BNF bug for attribute instance along with timeunits_declaration]
[proposal: 2905_attribute_timeunits.pdf]

(6) Mantis 3254 _X__Yes ___No
http://www.eda.org/svdb/view.php?id=3254
[18.5.6 if-else constraints mistakenly uses the work "block" when it means "set"]
[proposal: 3254_constraint_set.pdf]

(7) Mantis 3298 ___Yes _X__No
http://www.eda.org/svdb/view.php?id=3298
[Use of 'this' in a coverpoint expression]
[proposal: 3298_this_covergroup.pdf]

What about the use of 'this' in randomize with blocks. Doesn't that need to be included in this section also?

(8) Mantis 3054 _X__Yes ___No
http://www.eda.org/svdb/view.php?id=3054
CLOSE as duplicate of 3202 (sv-ac)
[$countones and $onehot system functions in constraints]

(9) Mantis 2935 _X__Yes ___No
http://www.eda.org/svdb/view.php?id=2935
CLOSE: already fixed in new version of the LRM and uploaded [Correction to example in 9.7. in 1800-2009]

Arturo:
Below are my votes:

(1) Mantis 3046 ___Yes _**X**_No

The term "strictly downwards manner" is not well defined. Also, the language is a bit awkward in treating dotted names specially when all it should say is that normal rules apply, which already deal with the first name of a dotted name. I do agree with the outcome of the examples shown in the Mantis.

(2) Mantis 2506 _**X**_Yes ___No

(3) Mantis 3531 _**X**_Yes ___No

(4) Mantis 3394 _**X**_Yes ___No

(5) Mantis 2905 _**X**_Yes ___No

(6) Mantis 3254 _**X**_Yes ___No

The sentence seems somewhat tautological: constraint_set is .. constraint set. Perhaps we can find better verbiage – the intent is to define a constraint_set as either a single constrain or a group of constrains bracketed by curly braces. I'm voting yes because the same language exists in implication so this is no worse.

(7) Mantis 3298 ___Yes _**X**_No

Must mention constrains: A friendly amendment (which may need discussion):
The this keyword shall only be used within nonstatic class methods, constrains, inlined constrain methods, or embedded covergroups (see 19.4) ;

(8) Mantis 3054 ___Yes _**X**_No

Although allowing $countones and $onehot in constrains is reasonable. There is no explicit proposal that describes the semantics associated with such constrains. For example, do these function calls partition the constrains (as with other “allowed” functions) or are they to be treated differently? If the semantics are the same as for other function calls, the LRM can simply state that constrains will allow functions *__and system function__* that adhere to the existing limitations.

(9) Mantis 2935 _**X**_Yes ___No

Arturo

Steven:

(1) Mantis 3046 _X_Yes ___No

(2) Mantis 2506 ___Yes ___No _X_Abstain

I have not followed the discussions well enough to vote on this.

(3) Mantis 3531 _X_Yes ___No

I am not seeing blue text for the additions in A.8.4, though I see it for Syntax 11-8. If this isn't just a problem with my viewer, then this should be fixed.

(4) Mantis 3394 _X_Yes ___No

(5) Mantis 2905 _X_Yes ___No

(6) Mantis 3254 _X_Yes ___No

(7) Mantis 3298 _X_Yes ___No

BTW, it would also make sense to allow 'this' in initializers on non-static properties. These are effectively executed in the constructor, which is a non-static method. And these initializers can presumably refer to other class properties using the normal scoping rules, and this is an implicit reference to 'this' already.

(8) Mantis 3054 _X_Yes ___No
CLOSE as duplicate of 3202 (sv-ac)

(9) Mantis 2935 _X_Yes ___No
CLOSE: already fixed in new version of the LRM and uploaded

I haven't actually seen the fixed version, but I will take Shalom's word for it.

Alex Gran:

> (1) Mantis 3046 _X_Yes ___No
> http://www.eda.org/svdb/view.php?id=3046
> [Dotted names within inlined constraints]
> [proposal: 3046_inline_dotted_names.pdf]
>
> (2) Mantis 2506 _X_Yes ___No
> http://www.eda.org/svdb/view.php?id=2506
> [Non-trivial coverage space shapes and joint conditions are difficult
> to specify with covergroups]
> [proposal: Proposal for Mantis 2506_v5.pdf]
>
> (3) Mantis 3531 _X_Yes ___No
> http://www.eda.org/svdb/view.php?id=3531
> [null should be allowed in constant expressions]
> [proposal file: 3531_null.pdf]
>
> (4) Mantis 3394 _X_Yes ___No
> http://www.eda.org/svdb/view.php?id=3394
> [invalid example for dynamic array]
> [proposal: 3394_illegal_example.pdf]
>
> (5) Mantis 2905 _X_Yes ___No
> http://www.eda.org/svdb/view.php?id=2905
> [BNF bug for attribute instance along with timeunits_declaration]
> [proposal: 2905_attribute_timeunits.pdf]
>
> (6) Mantis 3254 _X_Yes ___No
> http://www.eda.org/svdb/view.php?id=3254
> [18.5.6 if-else constraints mistakenly uses the work "block" when it
> means "set"]
> [proposal: 3254_constraint_set.pdf]
>
> (7) Mantis 3298 _X_Yes ___No
> http://www.eda.org/svdb/view.php?id=3298
> [Use of 'this' in a coverpoint expression]
> [proposal: 3298_this_covergroup.pdf]
>
> (8) Mantis 3054 ___Yes ___No
> http://www.eda.org/svdb/view.php?id=3054
> CLOSE as duplicate of 3202 (sv-ac)
> [$countones and $onehot system functions in constraints]
Abstain - Arturo had already voted NO on this one. I want to fully understand Arturo's objection before making my decision.

> (9) Mantis 2935 _X_Yes ___No

> http://www.eda.org/svdb/view.php?id=2935
> CLOSE: already fixed in new version of the LRM and uploaded
> [Correction to example in 9.7. in 1800-2009]
>
>

Gord:
Yes to all except 2506.

Given the additional changes going into 2506 from other comments and the overall size of the proposal, I would like to have a final proposal go through unchanged before voting in favor.

Gord.

Yes to all except 2506. Would like a final review in a CC.

Dave Rich
Verification Technologist
Mentor Graphics Corporation
New Office Number: 510-354-7439

(1) Mantis 3046 _X_Yes ___No

(2) Mantis 2506 ___Yes _X_No

The BNF changes need to be replicated in Annex A. Also does there need to be a restriction on the visibility of functions declared within the scope of a cross.

(3) Mantis 3531 _X_Yes ___No

(4) Mantis 3394 _X_Yes ___No

(5) Mantis 2905 _X_Yes ___No

(6) Mantis 3254 _X_Yes ___No

(7) Mantis 3298 _X_Yes ___No

(8) Mantis 3054 _X_Yes ___No

(9) Mantis 2935 _X_Yes ___No

Regards,
Ray Ryan

(1) Mantis 3046 ___Yes _X_No

I agree with a lot of what Arturo mentioned.
To tell you the truth, I don't think this set of changes is even necessary.
I agree with the summary of the examples described in the mantis item, but it seems to me that the existing text leads to this conclusion.

(2) Mantis 2506 ___Yes ___No
2506_v5.pdf

I didn't have time to review this one in detail.

(3) Mantis 3531 _X_Yes ___No
(4) Mantis 3394 ___Yes _X_No

It isn't clear to me that this example is invalid.
Aren't both the existing text and the proposed text legal?
I agree with the change in the comment for dest2.

(5) Mantis 2905 _X_Yes ___No
(6) Mantis 3254 _X_Yes ___No
(7) Mantis 3298 ___Yes _X_No

By making this change, it creates a new problem in the existing text.
The following sentence assumes 'this' can only be used in a class subroutine.

"The this keyword denotes a predefined object handle that refers to the object that was used to invoke the subroutine that this is used within."

Friendly amendments:
The new text should be in blue and not underlined.
The mantis item should also be shown at the top of the proposal.

(8) Mantis 3054 _X_Yes ___No CLOSE as duplicate of 3202 (sv-ac)
(9) Mantis 2935 _X_Yes ___No