Document Number: N4182
Date: 2014-10-10
Authors: Michael Wong
Project: Programming Language C++, SG5 Transactional Memory
Reply to: Michael Wong <michaelw@ca.ibm.com>

SG5: Transactional Memory (TM) Meeting Minutes 2014/07/14-2014/10/06

Contents

Minutes for 2014/07/14 SG5 Conference Call

Minutes by Hans

Attending:

Michael Wong
Justin Gottschlich
Victor Luchangco
Mike Spear
Torvald Riegel
Jens Maurer
Hans Boehm

Adopted agenda

Minutes approved

Michael Talking about WG21 meeting:

Had 2nd and 3rd LEWG reviews, LWG review.

CWG: Code bloat issue with safe-by-default is deal breaker.

EDG: No control over linker. Useless code may be preserved.

Victor: No code bloat at high optimization with whole program analysis.

Went back to EWG with alternate, not safe-by-default design.

Justin: With explicit annotations, only annotated functions have code duplication.

Mike S.: Duplicate compilation vs. code bloat

extern template problem is back.

transaction-callable may be back.

Jens: CWG and EWG consensus against sbd. Shouldn't bloat unrelated code. May change implementation where definitions are visible. Most standard library definitions are visible because they are implemented in headers, even if we make them transaction-safe.

Jens: Need transaction-safety specified for more of the library, notaby other containers.

Jens: Appending to a string is conditionally transaction-safe.
Expressible in English, not in code.

Jens: Want explicit instantiations and "extern template". Calls will see regular extern function references without transaction-safe annotations.

Mike: Does transaction-safety have to be part of the type?

Victor, Jens: Yes, it already is.

More important to go through library.

Jens: Halfway through wording update. Core review is not complete; more issues may be coming up. But also need to go through library. Standard transaction-cancellation compatible
exception type needs to be added.

Victor: Transaction-safety treatment is essentially reverting to old version?

Jens: Yes.

Victor: Will LWG object to standard library code bloat?

Jens: In principle, there is no binary code involved. Minor bloat in binary part of standard library appears acceptable.

Michael W: Still need core wording resolved. Will call a CWG teleconference.

Michael W: Scheduled Oct. 6 teleconference to review pass over library.

Michael W: Also needs transaction_callable?

Others: Needed for initial version?

Hans & others: May still need transaction_callable for things like occasional logging.

Mike S: Probably want something like on-commit handlers instead.

Mike S. Transaction-callable already there. Will be supported anyway.

Hans & others: Want to make the same name usable everywhere.

Discuss a bit more later. Don't spend much time on it.

*** Action Items:

Mike S: Post example where transaction_callable is needed (done in the meantime)

Jens: Core wording

Mike W.: Arrange CWG conference call

Mike W.: Organize standard library wording

Mike S. & Torvald to get in touch about gcc library (done in the meantime)

Minutes for 2014/07/28 SG5 Conference Call

Meeting minutes by Michael W.

On Thursday, July 24, 2014 3:14:34 PM UTC-4, Michael Wong wrote:

Start Time: Monday, July 28, 2014, 12:00 PM US Pacific Time (07:00:00 PM in GMT)
End Time: 1:00 PM US Pacific Time (duration: one hour)

US primary phone number (CA): 916-356-2663 US auxiliary phone number (MA): 978-553-2663 US auxiliary phone number (TX): 512-314-3030
US toll-free phone number: 888-875-9370 EU phone number (UK): +44 1793 402663

Monday, July 28, 2014, 12:00 PM US Pacific Time
916-356-2663, Bridge: 2, Passcode: 9160508

When you hear a single beep, someone has joined the call. When you hear a double beep, someone has dropped from the call. With large numbers of participants, audio interference can be a problem. Please try to keep your phone muted whenever possible. If your phone does not have a mute button, the bridge will mute or un-mute your line if you dial *6.

The current secretary rota list is (the person who took notes at the last meeting is moved to the end)

Michael Scott, Mark, Michael Wong, Victor,
Maged, Mike Spear, Torvald, Justin, Tatiana, Jens Maurer, Hans

Michael Wong
Victor
Maged
Mike Spear
Hans
Paul
Justin
Jens

Reminder: We use the Secretary Rota to determine who is responsible for minutes at any given meeting. The first name on the list that is present at the meeting will be responsible for them. Upon completing the minutes, they should move their name to the end of the rota. In face-to-face meetings, minutes duties will be assigned for a morning session or an afternoon session or an evening session (if applicable) so as to distribute the load fairly (but not too fine grained; consider it a transaction).

Agenda:

1. Opening and introductions

1.1 Roll call of participants

1.2 Adopt agenda

Approved

1.3 Approve minutes from previous meeting, and approve publishing previously approved minutes to ISOCPP.org

Approved

1.4 Review action items from previous meeting (5 min)

1.4.1 Mike S: Post example where transaction_callable is needed
DONE

1.4.2. Jens: Core wording fixed for all comments from Rapperswil
Done.

1.4.3. Mike W.: Arrange CWG conference call
Not yet. Aiming for August Review?
CWG seem to have it monthly

1.4.4. Mike W.: Organize standard library wording
LWG review Oct 6.

1.4.5. Mike S. & Torvald to get in touch about gcc library
Done. Connected with Jonathan Wakely, the GCC Lib maintainer.

1.4.6. Mike W.: Produce a Working Draft

1.4.7. All: read latest wording draft from Jens.

1.4.8. All: Consider Attending UIUC Urbana meeting Nov 3-8; hotels held till Oct 1
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2014/n3828.pdf
Hotel details not in above paper:
https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/bBV-tlIe33A

Plan: Core review, Library review, vote to create a Working Draft paper
Tentative: Hans, Michael, Jens, Torvald, Maged

This one will be a critical vote assuming we can get all our tasks done.

2. Main issues (50 min)

2.1 C++ Standard library transactionalization plans

Mike Spear/Michael Wong

go container by container, is either easily TM-safe or made TM-safe without much big effort
big difference was we start with list and pull things in as needed
transition being done now:
entire GCC_V3 folder is being checked and is very empirical about it
can say what was impact to size of binary
most of Std library is in template: answer is that very small amount of source line of code was augmented
JOnathan Wakely helping
GCC source repository
"containers" section are done
"iterators" and "numerics" is open to be done
AI: invite all to read through the words between calls, 2 style of edits are there, one is conditional transaction safety needs to be checked carefully, and 2 is direct phrasing of transaction-safety

2.2. optimize_for_synchronized attribute formerly transaction_callable

Mike Spear has posted here the example
https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/wxY3T8qhQ3M

Mike Spear talked about the memcache example:
can we have the fprintf in some escape/on-commit handler with an atomic block
can use thread-local variables to address the problem
do we want to do all that yet? This case is also not one of those

another on the spot example is accesses to volatile variables in memcache, we replaced every access to volatile with transactions

these are the use cases, may be gather it into examples for the paper

thread-optimized allocators/thread-local allocation cache, then go to some backup allocator, if it is in some third party library
nice example, it is not clear that escape is good enough, have some correctness condition, and calling from outside the transaction may not be good enough it
AI: add a few more of these examples: Mike Spear example, mmap example (Hans), volatile, Mike Spear
AI: Jens to add this to core-language section, optimized_for_synchronized is better then previous, this is a function-modifier attribute

This feature is approved. No objection by all on the call.

AI: Mike Wong should expose to EWG by writing a paper to outline the changes and gather the examples. We should include 2.3 outcome to this paper as well.

2.3. Transaction-safety of virtual functions

Jens has posted example here:
https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/AIzRZQxRiaA

That means the destructor of std::bad_cast must be tx-safe.
However, it is virtual, because std::exception's destructor is.
The destructor of std::bad_cast calls the base class destructor
(i.e. the one for std::exception). So this one must be
(declared to be) tx-safe, as well. Now our rules for virtual
functions say that any destructor in a class derived from
std::exception needs to be declared tx-safe. That invalidates
a lot of tx-unrelated user code that becomes invalid as soon
as you switch on tx support in your compiler, because there
are probably a lot of user-defined exception classes out
there that have never heard of TM, thus do not annotate
their destructors as tx-safe, yet derive from std::exception
(eventually).

For legacy code, if they turn on TM but not use, they should not be affected.
Affects virtual destructors of exception class, also what() returns a char *,

relax rule in Jens' proposal here means: a virtual fn declared tx-safe in base class is not viral in derievd classes, quality not inherited, if you end up calling a derived class that is not tx-safe, then you can get undefined behavior.
can add hole in type system, if the whole hierarchy's vfunc is not tx-safe (when the base class vfunc is made tx-safe)
can use C++11 override to say that you intentionally override the base class and this may add some safety back
some of these options are orthogonal
some feel uncomfortable with all the choices
Michael: option 1 can introduce hole for all vfunc hierarchy where the base class has been made tx-safe, and you want to to a dynamic cast

Victor has a hybrid idea: 2 ways to declare a vfunc tx-safe, current way, but add new way,say derived classes may not be
can order of virtual and tx-safe convey the meaning
dynamic casts inside tx can be ...

Justin feel we already have special behavior for destructors for tx-safety (so option 2 or 3)

Hans feel that vfunc dynamically checking need to be extended to fn-ptr

AI: all need to think about the different preferences on how to resolve issue

Should we relax just for the std, and use a harder rule for user code

Jens like idea but concern about wording complexity

bad_cast could be replaced with bad_alloc, just

2.4. Core Wording review adjusted for removing SBD:
https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/5jgAMxzSrDY

http://jmaurer.awardspace.info/wg21/tmspec.html

3. Any other business

4. Review
4.1 Review and approve resolutions and issues [e.g., changes to SG's working draft]
4.2 Review action items (5 min)

This feature (synchronized_for_optimized) is approved. No objection by all on the call.

AI: invite all to read through the words between calls, 2 style of edits are there, one is conditional transaction safety needs to be checked carefully, and 2 is direct phrasing of transaction-safety

AI: add a few more of these examples: Mike Spear example, mmap example (Hans), volatile, Mike Spear
AI: Jens to add this to core-language section, optimized_for_synchronized is better then previous, this is a function-modifier attribute

AI: Mike Wong should expose to EWG by writing a paper to outline the changes and gather the examples. We should include 2.3 outcome to this paper as well.

AI: all need to think about the different preferences on how to resolve issue

5. Closing process
5.1 Establish next agenda
5.2 Future meeting:
Next meeting: Aug 11 Telecon

Meeting agendas in past and future (since the last Std meeting)

July 14: reviewed Rapperswil changes

July 28: close on optimize_for _synchronized

Aug 11: close on virtual function issue, review core wording,
Aug 25: review core and library wording
Sept 8: prepare for core telecon review+library progress; Michael/Hans away

Sept 22: review library wording; Michael away
Oct 6: LWG review special telecon; paper on changed for EWG; Oct 10 mailing deadline, Michael away but should be able to callin
Oct 20: review paper and presentation for EWG; Michael away but should be able to callin

Nov 3: C++ Std meeting Urbana

Minutes for 2014/08/11 SG5 Conference Call

Minutes by Michael S.

Start Time: Monday, Aug 11, 2014, 12:00 PM US Pacific Time (07:00:00 PM in GMT)
End Time: 1:00 PM US Pacific Time (duration: one hour)

The current secretary rota list is (the person who took notes at the last meeting is moved to the end)

Victor, Maged, Mike Spear, Torvald, Justin, Tatiana, Jens Maurer, Hans, Michael Wong, Michael Scott

Reminder: We use the Secretary Rota to determine who is responsible for minutes at any given meeting. The first name on the list that is present at the meeting will be responsible for them. Upon completing the minutes, they should move their name to the end of the rota. In face-to-face meetings, minutes duties will be assigned for a morning session or an afternoon session or an evening session (if applicable) so as to distribute the load fairly (but not too fine grained; consider it a transaction).

1. Opening and introductions

1.1 Roll call of participants

Maged, Mike Spear, Victor, Michael Scott, Michael Wong, Hans, Jens

Mark Moir not active on the list for now; removed from rota.
Should we change the time? Not unless the Europeans request.

1.2 Adopt agenda

No objection.

1.3 Approve minutes from previous meeting, and approve publishing previously approved minutes to ISOCPP.org

No objection.

1.4 Review action items from previous meeting (5 min)

1.4.1. Add a few more of these (optimize_for_synchronized) examples: Mike Spear example, mmap example (Hans), volatile, Mike Spear

Mike posted a couple things yesterday; see the reflector.

1.4.2. Jens to add this to core-language section, optimized_for_synchronized is better then previous, this is a function-modifier attribute.

Done as section 7.6.6: http://jmaurer.awardspace.info/wg21/tmspec.html

1.4.3. Invite all to read through the Std words between calls, 2 style of edits are there, one is conditional transaction safety needs to be checked carefully, and 2 is direct phrasing of transaction-safety.

Continuing item.

1.4.4. Mike Wong should expose to EWG by writing a paper to outline the changes and gather the examples. We should include 2.3 outcome to this paper as well.

Continuing item.

1.4.5. All need to think about the different preferences on how to resolve issue of transaction-safety of virtual functions.

Main item for today's call.

1.4.6. Mike W.: Arrange CWG conference call.
Not yet. Aiming for August Review?
CWG seem to have it monthly.

15 Sept. call is for concepts.
We're on for Monday, Sept 29. 1-3 ET.
Keeps us in front of LWG.

1.4.7. Mike W.: Organize standard library wording
LWG review Oct 6 3-4 ET

Our usual conf. call time.

Michael Scott: are we planning to loop back to discussion of empty transactions and serialization order?

Michael Wong: prob. not for this "cycle" (UIUC meeting). Don't think the issue will be a show-stopper as long as people understand we're still working on the issue. (NB: Michael W. suspects that Chandler may

be
opposed to including atomic transactions in any form, which is a bigger issue. Jens suspects that others may have similar concerns, but that their opinions may change with time and experience.)

Mike Spear: Would be good to have some sort of mechanism to accumulate feedback from users of technical standard implementations (gcc in particular).

1.4.8. Mike W.: Produce a Working Draft

Still to come.
Jens: What we have at present is his html.
Want to have something more formal to be the subject of a vote at UIUC, for "Working Draft" status. At that point the doc will probably become LaTeX rather than HTML. Subsequent changes will require a full-committee vote at a meeting.

1.4.9. All: Consider Attending UIUC Urbana meeting Nov 3-8; hotels held till Oct 1
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2014/n3828.pdf
Additional details:
https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/bBV-tlIe33A

Plan: Core review, Library review, vote to create a Working Draft paper
Tentative: Hans, Michael, Jens, Torvald, Maged
Michael Wong: This one will be a critical vote assuming we can get all our tasks done.

2. Main issues (50 min)

2.1 C++ Standard library transactionalization updates

Mike Spear: Not a lot to report this week.

2.2. Transaction-safety of virtual functions

Jens has posted example here:
https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/AIzRZQxRiaA

Michael Wong and Maged Michael worked through the proposed solutions.
Not really happy with any of them.

Options:
-- create safety hole only for exception hierarchy
-- or do it for all virtual destructors
-- or all virtual functions

-- or identify particular hierarchies, and treat them differently.
   Victor prefers this option.

Michael Wong: can two versions solve the problem for virtual destructors?
Victor/Jens: probably not. Still stuck with runtime errors.

Goal is to be able (in the unusual case) to have a transaction-safe construct that has transaction-UNsafe derived class (use of which would result in runtime error).
Mike Spear: how about "transaction_safe_noinherit" ?
Jens: agree with that.
Or maybe "explicit transaction_safe"?
(Or even "private transaction_safe"?)
So virtual functions are by default NOT transaction_safe;
unless they're labeled "transaction_safe", in which case their overrides are also transaction_safe;
unless the base method is labeled "transaction_safe_noinherit" ("explicit transaction_safe"), in which case overrides must be separately labeled (as "transaction_safe" or "explicit transaction_safe") if they want to be transaction_safe.

Hans: What about function pointers? Don't the same issues arise?
Michael Scott: Isn't that more properly a cast?
Jens: No -- compiler has to generate transaction-safe code; can't expect to turn an arbitrary pointer into a transaction-safe one.
A pointer analogy to the virtual function proposal wouldn't be a cast;
it would be a use of an explicitly labeled pointer.
As a side benefit, it would solve the terminate() problem.
Emerging consensus around this sort of solution, but no vote.

Michael Scott: also need to think about member function pointers, and how they tie into transaction safety, overrides, and "explicit".

2.3. Core Wording review adjusted for removing SBD:
https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/5jgAMxzSrDY
http://jmaurer.awardspace.info/wg21/tmspec.html

Didn't get to this.

3. Any other business

Didn't get to any.

4. Review
4.1 Review and approve resolutions and issues [e.g., changes to SG's

working draft]
4.2 Review action items (5 min)

All: conjure up function pointer examples to send to the mailing list.

Minutes for 2014/08/25 SG5 Conference Call

Minutes by Victor

Start Time: Monday, Aug 25, 2014, 12:00 PM US Pacific Time (07:00:00 PM in GMT)
End Time: 1:00 PM US Pacific Time (duration: one hour)

The current secretary rota list is (the person who took notes at the last meeting is moved to the end)

Maged, Mike Spear, Torvald, Justin, Tatiana, Jens Maurer, Hans, Michael Wong, Michael Scott, Victor,

Agenda:

1. Opening and introductions

1.1 Roll call of participants

Mike Spear, Victor, Maged, Hans, Michael Scott, Justin

1.2 Adopt agenda

Adopted.

1.3 Approve minutes from previous meeting, and approve publishing
previously approved minutes to ISOCPP.org

Approved.

1.4 Review action items from previous meeting (5 min)

1.4.1. invite all to read through the Std words between calls, 2 style of edits are there, one is conditional transaction safety needs to be checked carefully, and 2 is direct phrasing of transaction-safety
http://jmaurer.awardspace.info/wg21/tmspec.html

Continuing

1.4.2. All: transaction-safety explicit (no-inherit) wrt function pointers

This was just something for us to think about for this meeting, to help the discussion.

1.4.3. Michael Scott: also need to think about member function pointers, and how they tie into transaction safety, overrides, and "explicit”.

This was just something for us to think about for this meeting, not just for Michael Scott. We discussed it further at the meeting.

1.4.4. Mike Wong should expose to EWG by writing a paper to outline the changes and gather the examples. We should include 2.3 outcome to this paper as well.

Continuing (no update since Michael was absent)

1.4.5. Mike W.: Arrange CWG conference call
We're on for Monday, Sept 29. 1-3 ET.
Keeps us in front of LWG.

Continuing (no update since Michael was absent)

1.4.6. Mike W.: Organize standard library wording
LWG review Oct 6 3-4 ET

Continuing (no update since Michael was absent)

1.4.7. Mike W.: Produce a Working Draft

Continuing (no update since Michael was absent)

1.4.8. All: Consider Attending UIUC Urbana meeting Nov 3-8; hotels held till Oct 1
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2014/n3828.pdf
Additional details:
https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/bBV-tlIe33A
Plan: Core review, Library review, vote to create a Working Draft paper
This one will be a critical vote assuming we can get all our tasks done.

Likely: Hans, Michael, Jens, Torvald, Victor, Tatiana

Possibly: Maged, Justin
Unlikely: Michael Scott, Mike Spear

2. Main issues (50 min)

2.1 C++ Standard library transactionalization updates
Mike Spear/Michael Wong

Mike Spear reports that they went through all the containers identified in the standard library (not yet all the iterators though), and modified them as needed so they can be executed from transactional context. Across all 8 containers, approx 15 annotations were needed. Two issues came up:
- it was clunky to declare as transaction-safe the comparator functions passed to ordered maps.
- some helper functions throw exceptions that contain transaction-unsafe strings
  - this is expected to be deprecated in the future, so it should be a transient problem
  - for now, we get around it by declaring these as tm_pure

Hans: What about transaction-safety of shared pointers?

Mike Spear: We didn’t look at this. (AI for Mike to look at this.)

Mike also reports that they can easily generate statistics on the number of annotations required, resulting code bloat, etc., which we could include in the papers for the WG21 meeting.

2.2. Transaction-safety of virtual functions (focus on function pointers, member function pointers)

We recapped this issue, with a general consensus favoring having “explicit” transaction-safe annotation for virtual functions and function pointers.

Maged: Why do we need such a general solution? Why not just treat exception destructors specially?

Victor: Problem arises for other methods of exception. Also possibly for other base classes that are widely used and extended in legacy code. Also, function pointer helps with the problem we had with std::terminate call backs.

Michael Scott: We have a strong preference for static checking. Might this undermine that?

Victor: Yes, but we need some way to handle the case for std::exception. Also, this allows legacy code to be amended relatively easily so that it is easier to adopt transactions (whereas explicitly listing exception destructors for special treatment does not). (Note that that this does go against the principle of providing minimal functionality and waiting for users to complain if they actually need more. The argument for it is that users may just avoid using transactions entirely if it would break code in legacy derived classes.)

Justin: Okay, sounds good, but I want to think more about it.

Victor: Let's try to get definite agreement next meeting.

Also, we all agreed that none of the proposed "spellings" were ideal. Suggestions are welcome.

2.3. Core Wording review adjusted for removing SBD:
https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/5jgAMxzSrDY
http://jmaurer.awardspace.info/wg21/tmspec.html

2.4. Change in call time.

No European members were present, so we didn't propose any change.

3. Any other business

None proposed.

4. Review
4.1 Review and approve resolutions and issues [e.g., changes to SG's working draft]
4.2 Review action items (5 min)

AI (Mike Spear: Look into transaction-safety for shared pointers.
AI (All): Read and give feedback to Jens about wording
AI (All): Think about proposed solution to transaction-safety for virtual methods/function pointers
AI (Michael Wong): Several holdovers:
- outline changes in paper for EWG;
- schedule CWG concall;
- organize LWG review of standard library changes

- produce working draft

5. Closing process

5.1 Establish next agenda

Get closure on virtual methods/function pointers

5.2 Future meeting:

Next meeting: Sept 8 Telecon

Past and future Meeting agendas
July 14: reviewed Rapperswil changes
July 28: approved optimize_for _synchronized
Aug 11: review core wording, close on virtual function issue
Aug 25: decide on virtual function issue wrt function ptrs, member fn ptrs, use explicit
Sept 8: Michael away
Sept 22: Michael away
Oct 6: LWG review special telecon; Oct 10 mailing deadline, Michael away but should be able to callin
Oct 20: Michael away but should be able to callin
Nov 3: C++ Std meeting Urbana

Minutes for 2014/09/08 SG5 Conference Call

Minutes by Maged

Start Time: Monday, Sept 8, 2014, 12:00 PM US Pacific Time (07:00:00 PM in GMT)
End Time: 1:00 PM US Pacific Time (duration: one hour)

The current secretary rota list is (the person who took notes at the last meeting is moved to the end)

Mike Spear, Torvald, Justin, Tatiana, Jens Maurer, Hans, Michael Wong, Michael Scott, Victor, Maged.

Agenda:

1. Opening and introductions

1.1 Roll call of participants

Maged, Michael Scott, Jens Maurer, Victor, Michael Wong, Hans, Mike Spear, Justin, Tatiana.

1.2 Adopt agenda

Adopted

1.3 Approve minutes from previous meeting, and approve publishing previously approved minutes to ISOCPP.org

Approved

1.4 Review action items from previous meeting (5 min)

1.4.1. invite all to read through the Std words between calls, 2 style of edits are there, one is conditional transaction safety needs to be checked carefully, and 2 is direct phrasing of transaction-safety
http://jmaurer.awardspace.info/wg21/tmspec.html

Comments from Michael Scott. Ongoing.

1.4.2. All: approving solution for transaction-safety explicit (no-inherit) wrt function pointers and the naming

Ongoing.

1.4.3. Michael Spear: consider tx-safety of shared pointers

Carry over.

1.4.4. Michael Wong should expose to EWG by writing a paper to outline the changes and gather the examples. We should include 2.3 outcome to this paper as well.

Ongoing.

1.4.5. Michael Wong.: Arrange CWG conference call We're on for Monday, Sept 15. 1-3 ET. (CHANGED from Sept 29) Keeps us in front of LWG.

Changed to Sep 15 from Sep 29.

1.4.6. Michael Wong.: Organize standard library wording LWG review Oct 6 3-4 ET

Keep as a reminder.

1.4.7. Michael Wong.: Produce a Working Draft

Carry over

1.4.8. All: Consider Attending UIUC Urbana meeting Nov 3-8; hotels held till Oct 1
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2014/n3828.pdf
Additional details:
https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/bBV-tlIe33A
Plan: Core review, Library review, vote to create a Working Draft paper
Likely: Hans, Michael, Jens, Torvald, Victor, Tatiana
Possibly: Maged, Justin
Unlikely: Michael Scott, Mike Spear
This one will be a critical vote assuming we can get all our tasks done.

Keep as reminder.

2. Main issues (50 min)

2.1. Vote on Transaction-safety of virtual functions (focus on function pointers, member function pointers)
Should it support function pointers/member function pointers
Jens has posted example here:
https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/AIzRZQxRiaA

Victor: Brief overview of problem.
M Wong: Concerned about explicit tx_safe for function pointers. OK with this for member function pointer. It seems pointless to apply ixplicit tx_safe to function pointers without object.

Victor: Programmers would expect function pointers and virtual functions to behave similarly
Justin: You can use a function pointer for cases that are transactional and non-transactional.
Hans: Example of function that takes function pointer as argument. So making the function pointer tx_safe can break legacy code.
Victor: foo() takes function pointer as argument. bar can be declared explicit tx_safe.
M Wong / M Scott: This is conflating two issues (1) A function that MAYBE tx_safe with (2) a function that is tx_safe and might be overridden as tx_unsafe. Let us separate the issues.
Hans / Victor: The issues are related and solution applies to both cases.
M Wong: There is a difference in that virtual function is surrounded by class abstractions and function pointers don't have the same encapsulation.
M Scott: Do we have a compelling use case similar to virtual functions.
Hans: Thinks that his example is compelling. The difference is that the virtual function issue appears in standard libraries but the function pointer issue will appear in user code all over the place.
M Wong: Agrees. But we should use different constructs.
Jens: Virtual function solution is a must have. Function pointers solution is nice to have. OK to get feedback after TS is published. Disagrees with M Wong that we use different syntax. The two issues are essentially similar. Favors the same syntax.
M Spear: Code for ordered containers in STL. If code is moved to a different compilation unit this problem would arise. Not always possible to do dynamic test. Supports different syntax because virtual functions must have a definition and the compiler can always do static checking, while the compiler may not be able to do static checking for function pointers.
Victor / M Scott: Example. Class A has virtual function foo() declared explicit tx_safe. Someone else that not using TM derived class B from A. B.foo() need not be tx_safe.
M Wong: "maybe" can be a place holder for the case with function pointers.

NOTE from Michael Spear:
I noticed a minor typo in the minutes: instead of "Not always possible to do dynamic test.", it should be "Not always possible to do dynamic cast." The point was that in a transaction_safe function, we only have one version of the source code, so we can't know that it's necessary to do 'dynamic_cast<void (*)(void) transaction_safe>(my_func_ptr)'... in fact, such code would cause run-time errors in non-transactional/legacy code.

Third, I noticed that in the minutes, there wasn't any mention of Michael Scott's brief aside that we ought to consider std::function. Coincidentally, I received an email from my student Chao Wang this evening noting that in some work he's trying to do inside of GCC right now, he's run into the problem that std::function's operator() cannot be called from within a transaction. This seems closely related to the discussion today. For example, suppose that std::function::operator() was transaction_safe_maybe (or whatever we end up calling it). If that were true, then when compiling a TU with TM support turned on, if a lambda is assigned to a std::function object, the compiler would need to try to produce two versions of the lambda. If it failed to do so, it would just generate one, and there would be a run-time error if the std::function was used within a transaction. But since it should be valid to create lambdas within a transaction, even when those lambdas are not transaction_safe, it seems that there's no natural way to use the compilation context to avoid dynamic checking... the transaction_safe_maybe mechanism appears to be the most viable option.

- Mike

Action item for Victor: Summarize the issues in this discussion.

Jens / Victor: Will ignore function pointers for now.Will focus on virtual functions.

2.2. Core Wording review adjusted for removing Safe-By-Defaul:
https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/5jgAMxzSrDY

http://jmaurer.awardspace.info/wg21/tmspec.html

Core review meeting next Monday Sep 15.

Action item: All should read spec and send feedback yes or no.

3. Any other business

4. Review
4.1 Review and approve resolutions and issues [e.g., changes to SG's working draft]
4.2 Review action items (5 min)

5. Closing process
5.1 Establish next agenda
5.2 Future meeting:
Next meeting: Sept 22 Telecon

Past and future Meeting agendas
July 14: reviewed Rapperswil changes
July 28: approved optimize_for _synchronized
Aug 11: review core wording, close on virtual function issue
Aug 25: decide on virtual function issue wrt function ptrs, member fn ptrs, use explicit
Sept 8: Michael away; vote on tx-safety of virtual functions; Std Wording review; Sept 15 CWG review 1-3 ET
Sept 22: Michael away; LWG review in the next SG5 meeting
Oct 6: LWG review special telecon; Oct 10 mailing deadline, Michael away but should be able to callin
Oct 20: Michael away but should be able to callin
Nov 3: C++ Std meeting Urbana

Minutes for 2014/09/22 SG5 Conference Call

Minutes by Mike Spear

Start Time: Monday, Sept 22, 2014, 12:00 PM US Pacific Time (07:00:00 PM in GMT)
End Time: 1:00 PM US Pacific Time (duration: one hour)

US primary phone number (CA):   916-356-2663   US auxiliary phone number (MA):   978-553-2663   US auxiliary phone number (TX):   512-314-3030
US toll-free phone number: 888-875-9370 EU phone number (UK):   +44 1793 402663

Monday, September 22, 2014, 12:00 PM US Pacific Time
916-356-2663, Bridge: 3, Passcode: 9879819

When you hear a single beep, someone has joined the call. When you hear a double beep, someone has dropped from the call. With large numbers of participants, audio interference can be a problem. Please try to keep your phone muted whenever possible. If your phone does not have a mute button, the bridge will mute or un-mute your line if you dial *6.

The current secretary rota list is (the person who took notes at the last meeting is moved to the end)

Mike Spear, Torvald, Justin, Tatiana, Jens Maurer, Hans, Michael Wong, Michael Scott, Victor, Maged.

(Mike Spear taking minutes)

Agenda:

1. Opening and introductions

1.1 Roll call of participants

Present: Maged, Michael Scott, Jens, Victor, Mike Spear

1.2 Adopt agenda

No objections.

1.3 Approve minutes from previous meeting, and approve publishing previously approved minutes to ISOCPP.org

No objections.

1.4 Review action items from previous meeting (5 min)

1.4.1. invite all to read through the Std words for Std Library section (half way down the page)
http://jmaurer.awardspace.info/wg21/tmspec.html

October 6th is the LWG meeting. CWG meeting went well.

1.4.2. Victor to summarize tx-safety of virtual functions
Done

Victor sent an email, which was very comprehensive.

1.4.3. Michael Wong should expose to EWG by writing a paper to outline the changes and gather the examples. We should include 2.3 outcome to this paper as well.

1.4.4. Michael Wong.: Organize standard library wording LWG review Oct 6 3-4 ET

We will keep this on the minutes to make sure that this is scheduled. Jens hasn't heard anything on the LWG reflector yet.

1.4.5. Michael Wong.: Produce a Working Draft

1.4.6. All: Consider Attending UIUC Urbana meeting Nov 3-8; hotels held till Oct 1
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2014/n3828.pdf
Additional details:
https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/bBV-tIIe33A
Plan: Core review, Library review, vote to create a Working Draft paper
Likely: Hans, Michael, Jens, Torvald, Victor, Tatiana
Possibly: Maged, Justin
Unlikely: Michael Scott, Mike Spear
This one will be a critical vote assuming we can get all our tasks done.

Victor may be less "likely" than he was previously. Not certain yet.

2. Main issues (50 min)

2.1. Review Core wording telecon from last Monday.
https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/oJtYjkxtqJM

There were only relatively minor comments about wording.

Jens: there are a few extension-like things that we should present to the group item-by-item:

- The syntax changes where the CWG felt that the transaction_safety annotation wasn't worked in completely seamlessly won't need EWG approval.

- We need EWG approval for (1) optimize for synchronized; and (2) "maybe" for transaction_safety.

The changes are enumerated in the current HTML document.

These changes are addressed, and should not be controversial:

- When throwing a pointer to a function, should the catch clause match the pointer type exactly, or can it skip "transaction_safe". We should allow removal of the type qualifier.
-- Catch handlers are in a separate section, so Jens had to add additional text to address this.

- We allow, but don't require transaction_safe annotation on lambdas

- There was a loophole when maybe_transaction_safe overrides transaction_safe. This should not be allowed.

- Template argument deduction: if the parameter list includes pointer to function that is not transaction_safe, but it is given a pointer to a transaction_safe function, no error should result.

- The '?' operator: if one branch returns a pointer to function, and the other returns a pointer to transaction_safe function, then there should be an automatic conversion that drops the transaction_safe qualifier.

Jens: Note, too, that there is much less "blue" than before. This is a direct consequence of the CWG teleconference.

2.2. Standard Library Wording review (please scroll half way down the page), not Standard Core review

http://jmaurer.awardspace.info/wg21/tmspec.html

Library review meeting Monday Oct 6 3-4 ET in SG5 call.

Action item: All should read spec and send feedback yes or no.

Jens will need feedback in advance of the Oct 6th meeting. It begins with Chapter 17. Our goal should be to ensure that we agree with the claims about what should and should not be transaction_safe. We should also make sure that Jens did not accidentally deviate from what we, individually, thought was the consensus of the group.

Jens: Once the invitation goes to the group, it's too late to make significant changes to the document, so he needs that feedback soon... next 2--3 days would be best.

2.3. Further discussion on Transaction-safety of virtual functions (focus on function pointers, member function pointers)
Should it support function pointers/member function pointers
Jens has posted example here:
https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/AIzRZQxRiaA

One open issue is the wording. "maybe" is probably not going to be the final spelling.

Jens: We don't have a good example for function pointers.

Through trial and error, the group has largely concluded that neither the collections nor algorithms library really need this (C-style qsort is an exception, but can be handled via overloading). Note: Both collections and algorithms use templates heavily. Also, note that function parameters being transaction_safe is sufficient to enable overloading.

However, Jens and Hans both believe this will be useful somewhere.

Victor suggests that we don't put this into the wording yet, because we don't have a strong example. Perhaps we should do a straw poll at the meeting, but we don't have consensus now for function pointers.

Maged: noinherit is descriptive, but does not extend well to function pointers. Is there a requirement to be stingy with the creation of new keywords?

Jens: not really, but this can complicate the rules that govern parsing.
Michael: could we use a token, like '?', instead of a keyword?
Jens: it's possible. The main goal is to be unambiguous, so this might not go over well. We are already using a lot of new keywords, to include some that are very long.
Victor: we can postpone the naming until the LWG meeting.

Poll: does anyone object to extending to function pointers? *no*. We do not have any record of anyone against this proposal. If EWG asks, it appears that nobody on SG5 has a pushback against doing this.

"Dynamic" is a possible word (Maged suggested, Ville had mentioned it earlier). But it doesn't really work either.

General consensus: for now, don't do function pointers, and stick with "noinherit". When (if?) we add function pointers, we can change the name. We should leave this up to the committee.

3. Any other business

Proposal to move meeting time forward 1 hour (i.e., to 2:00 PM Eastern) after the Urbana meeting.

4. Review
4.1 Review and approve resolutions and issues [e.g., changes to SG's working draft]

Agree to rename "maybe" to "noinherit", and to delay decision on function pointers.

4.2 Review action items (5 min)

Michael Wong to ensure that the LWG phone conference is set.

Michael Wong should expose to EWG by writing a paper to outline the changes and gather the examples. We should include 2.3 outcome to this paper as well.

Michael Wong: Produce a Working Draft

All: send feedback to Jens about the Library section of the HTML at http://jmaurer.awardspace.info/wg21/tmspec.html. Please do this within the next 2 days, so changes reach the LWG before they begin reviewing the document.

Jens: rename "maybe transaction safe" to "transaction safe noinherit".

5. Closing process
5.1 Establish next agenda
5.2 Future meeting:

Minutes for 2014/10/06 SG5 Conference Call

Minutes by Justin

Start Time: Monday, Oct 6, 2014, 12:00 PM US Pacific Time (07:00:00 PM in GMT)
End Time: 1:00 PM US Pacific Time (duration: one hour)

US primary phone number (CA):   916-356-2663   US auxiliary phone number (MA):   978-553-2663   US auxiliary phone number (TX):   512-314-3030
US toll-free phone number: 888-875-9370 EU phone number (UK):
+44 1793 402663

Monday, October 06, 2014, 12:00 PM US Pacific Time
916-356-2663, Bridge: 1, Passcode: 7839577

When you hear a single beep, someone has joined the call. When you hear a double beep, someone has dropped from the call. With large numbers of participants, audio interference can be a problem. Please try to keep your phone muted whenever possible. If your phone does not have a mute button, the bridge will mute or un-mute your line if you dial *6.

The current secretary rota list is (the person who took notes at the last meeting is moved to the end)

Torvald, Tatiana, Jens Maurer, Hans, Michael Wong, Michael Scott, Victor, Maged, Mike Spear, Justin

Agenda:

1. Opening and introductions

1.1 Roll call of participants

Michael Scott, Mike Spear, Justin, Maged, Jens (chairing start), Michael Wong (chaired remainder starting Item 1.4.3), Victor

Jens: is anyone from library working group on call? No one from LWG is on call.

1.2 Adopt agenda

Jens: the agenda is no longer appropriate. Let’s postpone change to agenda until we are in the main issue.

1.3 Approve minutes from previous meeting, and approve publishing previously approved minutes to ISOCPP.org

Approved.

1.4 Review action items from previous meeting (5 min) Note these ACTION Items will assume to be done and not reviewed in order to save time.

1.4.1. invite all to read through the Std words for Std Library section (half way down the page)
http://jmaurer.awardspace.info/wg21/tmspec.html

Jens: Victor sent feedback, but haven’t had time to incorporate yet.

Jens: Has anyone else sent email about updates?

Action item: Jens to integrate Victor’s comments.

1.4.2. Jens: rename "maybe transaction safe" to "transaction safe noinherit".

Closed.

1.4.3. Michael Wong should expose to EWG by writing a paper to outline the changes and gather the examples. We should include 2.3 outcome to this paper as well.

In progress. Should have update in a few days.

1.4.4. Michael Wong.: Organize standard library wording LWG review Oct 6 3-4 ET

Done?

1.4.5. Michael Wong.: Produce a Working Draft

In progress.

1.4.6. All: Consider Attending UIUC Urbana meeting Nov 3-8; hotels held till Oct 1
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2014/n3828.pdf
Additional details:
https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/bBV-tlIe33A
Plan: Core review, Library review, vote to create a Working Draft paper
Likely: Hans, Michael, Jens, Torvald, Victor, Tatiana
Possibly: Maged, Justin
Unlikely: Michael Scott, Mike Spear
This one will be a critical vote assuming we can get all our tasks done.

Michael: Ideal time to attend is Tuesday, Wednesday and Thursday.

2. Main issues (50 min)

2.1. Standard Library Wording review (please scroll half way down the page), not Standard Core review

http://jmaurer.awardspace.info/wg21/tmspec.html

Library review meeting Monday Oct 6 3-4 ET in SG5 call.

Action item: All should read spec and send feedback yes or no.

Discussion:

Michael Wong: What will we need for UIUC meeting?

Jens: We need to have a working draft, or something fairly close to a working draft on Friday at the meeting, so we can have a meaningful vote. We’re not at that stage yet because we have not had an LWG review yet and we have not yet had EWG review some changes since our last pass.

Mike Spear: Have we expanded the scope with regard to STL?

Jens: We have expanded the scope with regard to STL containers. We went from having annotations for one container to having annotations for all containers.

Mike Spear: Right, but we haven’t added any new annotations that are visible to the programmer.

Jens: Agreed.

Michael Wong: Okay, but we will likely have to have a LWG review and possible an EWG review at UIUC. Will we need a CWG review?

Jens: My opinion is we probably should have a CWG review at the discretion of the head of CWG.

Jens: Cannot have any meaningful review on Monday. My HTML webpage should become a specification and paper that we present. However, we need to be prepared to get the draft rolling very shortly after the UIUC meeting. It would be great to have post-UIUC meeting to have a solid draft together.

Michael Wong: Will check with Herb with respect to what is needed for meeting.

Jens: Have you allocated an N numbers from Clark?

Michael Wong: Will record after call.

Action Item: Michael Wong to secure N numbers from Clark.

Papers:

1. HTML wording
2. Paper for evolution working group
3. Minutes paper

Action Item: Discuss with Herb … what?

Action Item: Michael Wong to send reference paper with author list to Jens as a reference for specification.

Michael Wong: Should we try to have a second LWG phone call on Oct 20?

Jens: Probably not as we will still need to meet with LWG at the C++ Standards Meeting as we will unlikely have consensus to say LWG approves.

--

Michael Wong: What should we discuss on Oct 20?

Jens: The EWG meeting should be discussed in the coming week because it should be included in the pre-meeting mailing.

Jens: Might be good to make progress on Chandler’s open issue. However, it may be challenging to discuss this new research idea via a phone call.

Mike Spear: Can someone remind me the open issue?

Jens: The compiler analyzes that an atomic block is actually not using any shared memory, so the atomic block can be eliminated.

Michael Wong: Do we have enough TM examples?

Jens: I have come up with some examples in the TM spec. Synchronized example, atomic noexcept example, optimized for synchronized example, transaction safe example, noinherit example. No example for transaction cancelation, but not sure we need one.

Michael Wong: Issue with memory model should be on agenda.

Jens: Review papers we have written.

Mike Spear: Do we want data for STL containers?

Jens: It seems like the current three papers we are submitting don't have a place for them. But, we might be able to create a new N numbered paper

Victor: When should I be at the UIUC meeting?

Michael Wong: Review will happen Tuesday, Wednesday, and Thursday. On Friday, I will need some TM expert there just in case someone opens up a can of worms. So we will need someone to stay through Friday 5:00pm.

Victor: Do we have a specific schedule?

Jens: Unfortunately, not yet.

3. Any other business

4. Review
4.1 Review and approve resolutions and issues [e.g., changes to SG's working draft]
4.2 Review action items (5 min)

Action Item: Jens to integrate Victor's comments.

Action Item: Michael Wong to secure N numbers from Clark. Three of them.

Action Item: Michael Wong to discuss with Herb the procedure aspects of the draft.

Action Item: Michael Wong to send reference paper with author list to Jens as a reference for the specification.

Action Item: (Lower Priority) Mike Spear and Michael Wong to try to create STL containers data paper by October 10.

5. Closing process

Chandler's memory model discussion.

5.1 Establish next agenda
5.2 Future meeting:
Next meeting: Oct 20 Telecon

Past and future Meeting agendas
July 14: reviewed Rapperswil changes
July 28: approved optimize_for _synchronized
Aug 11: review core wording, close on virtual function issue
Aug 25: decide on virtual function issue wrt function ptrs, member fn ptrs, use explicit
Sept 8: Michael away; vote on tx-safety of virtual functions; Std Wording review; Sept 15 CWG review 1-3 ET
Sept 22: Michael away; LWG review in the next SG5 meeting
Oct 6: LWG review special telecon; Oct 10 mailing deadline, Michael away but should be able to callin
Oct 20: Michael away but should be able to callin
Nov 3: C++ Std meeting Urbana