Document Number: N3717
Date: 2013-08-30
Authors: Michael Wong
Project: Programming Language C++, EWG, SG5 Transactional Memory
Reply to: Michael Wong <michaelw@ca.ibm.com>

SG5: Transactional Memory (TM) Meeting Minutes 2013/06/24-2013/08/26

Contents

Minutes for 2013/06/24 SG5 Conference Call

Minutes by Maged Michael

The current secretary rota list is:

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

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

Michael Scott, Victor, Mike Spear, Michael Wong, Maged, Zihao, Torvald

1.2 Adopt agenda

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

1.4 Review action items from previous meeting (5 min)

1.4.1. Michael Wong to talk to IBM lawyer.

All agreeing to talk among the lawyers to clear the obstacles to release the specification to ISO

1.4.2 Victor to update the wiki page to reflect the decision not to include transaction expressions and blocks in phase one.
Done (Will be done by end of the meeting).

1.4.3 Everybody comment on clause 6 (http://wiki.edg.com/twiki/pub/Wg21bristol/BasicFirstProposal/tmspec.html).
(This is the last meeting we will continue this AI; but we will still accept changes in future)

Done.

1.4.4 Everybody please read clause 15
(http://wiki.edg.com/twiki/pub/Wg21bristol/BasicFirstProposal/tmspec.html).
(This is the last meeting we will continue this AI; but we will still accept changes in future)

Done.

1.4.5. Everyone to review Victors questions on the minimalist proposal (see Victor's post on 2013/05/28: "Questions about basic proposal" or see 2.5 below). Discussion of these questions in this meeting.

Will discuss this meeting.

1.4.6 Michael Scott to give another stab at non-normative memory model writeup

Done.
New AI: All to review Michael Scott's writeup.

1.4.7 Considering going to Chicago C++ Standard meeting, Hotel Booking deadline Sept 6, 2014:
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2013/n3577.pdf

The meeting will be mostly about C++14.
SG5 may get an official vote on a work item for a new TS.
There might not be discussion of TM details.

2. Main issues (50 min)

2.1 Outstanding questions about Victor's minimalist proposal (see Victor's post on 2013/05/28: "Questions about basic proposal").

1. Is it okay to omit the transaction_callable attribute?

Consensus that transaction_callable is not needed in v1.

There are concerns about code bloat due to instrumentation by default on the one hand and poor performance due to lack of instrumentation on the other. Consensus is to drop transaction_callable in v1 and deal with the code bloat and performance issues if they arise in the future.

2. Is it okay to omit class attributes for transaction-safety?

Consensus on omitting class attributes for transaction safety in v1.

3. Should we allow transaction_unsafe to modify a function-pointer declaration, and if so, what effect does it have?

Consensus that it is only documentary. To be forbidden in v1.

4. Should empty atomic transactions be barriers? If not, how do we define "empty"?

Consensus to treat empty transactions as barriers in v1.
In the future, relaxed syntax can be added.

5. Do the safe-by-default rules forbid a base class that defines a virtual function that executes only transaction-safe code and a derived class that overrides that virtual function with one that executes transaction-unsafe code, even if neither of them specify any transaction-safe attributes? If so, is this restriction acceptable? If not (i.e., if such a base class and derived class is allowed), can the virtual function be called within a transaction for an object of the base class? What if at run time, that object turns out to be of the derived class?

2.2 Is non-normative part of memory model write-up acceptable?

2.3 Is clause 6 acceptable
(http://wiki.edg.com/twiki/pub/Wg21bristol/BasicFirstProposal/tmspec.html)?

2.4 Is clause 15 acceptable
(http://wiki.edg.com/twiki/pub/Wg21bristol/BasicFirstProposal/tmspec.html)?

2.5 Review of recent talks

2.6 Goal of SG5

2.7 Next standardese wording topic?

2.8 Review future plans and away times

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 meetings: July 8, teleconference

April 29: Post Bristol report (DONE)
May 13: Discuss memory model wording, atomicity wording (Michael Wong at C++Now) (DONE)
June 3: Discuss transaction expressions, function transaction blocks, memory model wording and other standardese topics (Michael Wong at OpenMP and Innovate) (DONE)

June 10: Discuss function transaction blocks, clause 6, clause 15 (Victor away)
June 24: Discuss Outstanding questions about Victor's minimalist proposal, Review recent TM talks at TRANSACT, ACCU, ADC++, C++Now. Goal of SG5 (Justin at HotPar, Hans at HotPar).
July 8:
July 22:
Aug 5
Aug 19:
Sept 9: (Michael Wong at IWOMP, Justin at PACT)
Sept 23: Chicago C++ Std Meeting

5.3 Adjourn

Minutes for 2013/07/08 SG5 Conference Call

Minutes by Justin Gottschlich

1. Opening and introductions

1.1 Roll call of participants

Hans Boehm
Justin Gottschlich
Victor Luchangco
Maged Michael
Mark Moir
Michael Scott
Mike Spear
Michael Wong

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. Michael Wong to get status of TM draft from lawyers.
In progress (still open).

1.4.2 Victor to update the wiki page to reflect the decision not to include transaction expressions and function transaction blocks in phase one.
Done.

1.4.3. Everyone to review Victor's questions on the minimalist proposal (see Victor's post on 2013/05/28: "Questions about basic proposal" or see 2.5 below). Discussion of these questions in this meeting.
In progress (might close by this meeting).

1.4.4 Considering going to Chicago C++ Standard meeting, Hotel Booking deadline Sept 6, 2014:
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2013/n3577.pdf
In progress (will remain on agenda until September 6, 2014).

Who's going?

2. Main issues (50 min)

2.1 Outstanding questions about Victor's minimalist proposal (see Victor's post on 2013/05/28: "Questions about basic proposal").

1-4 answered.
5. Do the safe-by-default rules forbid a base class that defines a virtual function that executes only transaction-safe code and a derived class that overrides that virtual function with one that executes transaction-unsafe code, even if neither of them specify any transaction-safe attributes? If so, is this restriction acceptable? If not (i.e., if such a base class and derived class is allowed), can the virtual function be called within a transaction for an object of the base class? What if at run time, that object turns out to be of the derived class?
Victor explains problem and how it applies to virtual functions. In essence, a base class can declare a virtual function that is transaction safe. Then a derived class from that base class can override the transaction safe virtual function and make it transaction unsafe. Then a function can take as a parameter a pointer to the base class and invoke a call to the virtual function. Such invocation can lead to a call to either a transaction safe or transaction unsafe call, depending on which actual object is passed to the function.

Mark: What if we disallow safe-by-default for virtual functions, so virtual functions must be declared as transaction safe or transaction unsafe. If there is no safety declaration, then the virtual function is inferred to be transaction unsafe.

Mike Spear: Seems like this might be the only way to solve this problem without a run-time error.

Hans: It seems consistent with how we handle function pointers.

Mark: How often are virtual functions used?

Michael Wong: There are used, but it seems that their usage might be decreasing.

Mark: The question then becomes does safe-by-default fulfill the promise of automatically handling transaction safety?

Mike Spear: There is another alternative such that virtual functions can be automatically transaction safe and then this could avoid some viral annotation issues. I’m not advocating for such an approach, but rather pointing out that such an approach exists.

Mark: Okay. But the selling point of SBD is you can just compile your program and it is correct. If we violate this guarantee then that usefulness seems to be diminished.

Mark: Another option is using a compiler flag that allows all virtual functions to be transaction safe. So, then by default we could have virtual functions as transaction unsafe, unless the compiler flag is used to make virtual functions as transaction safe, in which case the virtual functions are transaction safe by default.

Victor: We might have some other non-standard document that includes suggestions on how we might include some discussion about what a compiler might want to do.

Michael Scott: Is there a one sentence explanation why we want the default for virtual functions to be transaction unsafe?

Victor explains the reason why this problem can lead to run-time errors in transaction safety.

Mike Spear: Is there a transaction unsafe annotation?

Victor: Yes.

Decision: unannotated virtual functions are transaction unsafe.

Action item: Victor to modify Jens list of transaction unsafe items to include virtual functions.

Hans: On a completely different topic, static variables within a function need to call initializers upon the first pass of executing the code. Because of this, some isolation issues might arise if such initialization occurs within an atomic transaction.

Discussion about static variables within functions.

Action item: Hans to send out example on how if a function local static is used inside an atomic transaction then isolation might be violated with naïve implementations.

2.2 Is non-normative part of memory model write-up acceptable?

2.3 Is clause 6 acceptable
(http://wiki.edg.com/twiki/pub/Wg21bristol/BasicFirstProposal/tmspec.html
)?

2.4 Is clause 15 acceptable
(http://wiki.edg.com/twiki/pub/Wg21bristol/BasicFirstProposal/tmspec.html)?

2.5 Review of recent talks

2.6 Next standardese wording topic?
2.7 Review future plans and away times

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 meetings: July 22, teleconference

April 29: Post Bristol report (DONE)
May 13: Discuss memory model wording, atomicity wording (Michael Wong at C++Now) (DONE)
June 3: Discuss transaction expressions, function transaction blocks, memory model wording and other standardese topics (Michael Wong at OpenMP and Innovate) (DONE)
June 10: Discuss function transaction blocks, clause 6, clause 15 (Victor away)
June 24: Discuss Outstanding questions about Victor's minimalist proposal, Review recent TM talks at TRANSACT, ACCU, ADC++, C++Now. Goal of SG5 (Justin at HotPar, Hans at HotPar).
July 8: Discuss Outstanding questions 5 onwards about Victor's minimalist proposal, Review recent TM talks at TRANSACT, ACCU, ADC++, C++Now, Hot PAR.
July 22:
Aug 5
Aug 19:
Sept 9: (Michael Wong at IWOMP, Justin at PACT)
Sept 23: Chicago C++ Std Meeting

5.3 Adjourn

Minutes for 2013/07/22 SG5 Conference Call

Minutes by Michael Scott

SG5 minutes
Monday, July 22, 2013, 12:00 PM US Pacific Time
Participating: Hans Boehm, Jens Maurer, Maged Michael, Michael Scott, Tatiana Shpeisman, Mike Spear, Michael Wong, Zhihao Yuan

The current secretary rota list is:
Torvald, Mark, Victor, Jens Maurer, Mike Spear, Hans, Tatiana, Zhihao, Michael Wong, Maged, Justin, 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).

Agenda:

1. Opening and introductions

1.1 Roll call of participants
1.2 Adopt agenda

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

1.4 Review action items from previous meeting (5 min)
1.4.1. Michael Wong to talk to IBM lawyer.
   No news from lawyers yet. Michael W. will ping them again.

1.4.2. Victor to update wiki BasicProposal to reflect decision and reasoning from last meeting: unannotated virtual functions are transaction unsafe.

   Done.

1.4.3. Victor to modify Jens list of transaction unsafe items to include virtual functions.

Done. Jens: NB: list is not necessarily complete.

1.4.4. Hans to send out example on how if a function local static is used inside an atomic transaction then isolation might be violated with naïve implementations.
Done; further discussion below.

1.4.5. Considering going to Chicago C++ Standard meeting, Hotel Booking deadline Sept 6, 2014:
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2013/n3577.pdf

Michael W. doesn't anticipate a lot of TM discussion at the meeting: it will largely be devoted to C++'14 issues.

Jens: BUT, the people who will be most interested in TM will be the ones least committed to C++'14 issues. (Jens and Hans will be both, though.)

Michael W. hopes, nonetheless, to submit our planned standardese wording ahead of time. Jens agrees this is wise. Goal: get "air time" in the Evolution committee (chaired by Stroustrup).

Michael Scott: do we need to agree on names of constructs first?
Answer: no.

2. Main issues (50 min)

2.1 Outstanding (additional) questions about Victor's minimalist proposal (see Victor's post on 2013/05/28: "Questions about basic proposal").
1-5 previously answered.

6. If a virtual function declaration specifies the transaction_safe attribute, must an overriding virtual function declaration also specify the attribute, even though it is redundant (because if the overriding function were not transaction-safe, the program would be rejected)? Also, if a virtual function has multiple declarations, including one that specifies the transaction_safe attribute, must all its declarations specify the attribute? If not, must any particular declaration specify it?
We previously agreed that overriding virtual functions cannot be transaction unsafe when the base version was transaction safe. The question here is whether all declarations and the definition of the overriding versions have to _say_ they're transaction safe.
Consensus: yes. This follows the pattern of exceptions, and prevents the programmer from accidentally adding a transaction safety declaration to a method of a class that has a derived class

in which the method is not safe.

Jens: Note that we are requiring the compiler/linker to catch inconsistencies across compilation units -- a stronger requirement than exists for exceptions.

7. Is the dynamic initialization of function-local statics transaction-safe (assuming the code in the initialization expression is transaction-safe) even though it likely involves some nonatomic synchronization under the covers?

Problem: Initialization of function-local statics implicitly introduces synchronization. We don't want to break that: if I create create an object in a transaction, I don't want to be able to see concurrent initialization of statics by another thread. But implementation implications aren't clear: would all initializations of statics have to be protected by transactions under the hood?

Michael Scott: Perhaps we could we protect initialization with the same global lock that lets atomic and relaxed transactions interoperate.

Hans: We need to think carefully about performance implications (e.g., wrt memory order) on subsequent _uses_ of statics.

Mike Spear: even in the worst case, perhaps desirable (always atomic) semantics would impose new costs only on transaction-safe methods.

Will continue discussion online.

2.2 Is non-normative part of memory model write-up acceptable?

Yes. Jens will turn the blue sentence black.

2.3 Is clause 6 acceptable?
(http://wiki.edg.com/twiki/pub/Wg21bristol/BasicFirstProposal/tmspec.html)

(This is the syntax for transaction statements.)
Previously accepted.
Jens will turn blue words black.

2.4 Is clause 15 acceptable?
(http://wiki.edg.com/twiki/pub/Wg21bristol/BasicFirstProposal/tmspec.html)

(This is exception handling.)
Previously accepted.
Jens will turn blue words black.

# not reached, due to lack of time:
#
# 2.5 Review of recent talks: TRANSACT, ACCU, ADC++, C++Now, Hot PAR.
#
# 2.6 Next standardese wording topic?
#
# 2.7 Review future plans and away times
#
# 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 meetings:

Aug 5, teleconference
Aug 5: Continue discussion of outstanding question 7;
Review recent TM talks at TRANSACT, ACCU, ADC++, C++Now, HotPar.
Aug 19:
Sept 9: (Michael Wong at IWOMP, Justin at PACT)
Sept 23: Chicago C++ Std Meeting

5.3 Adjourn

Minutes for 2013/08/05 SG5 Conference Call

Minutes by Mark Moir

Attendees: Mark, Jens, Mike Spear, Hans, Michael Scott, Michael Wong,
Maged, Justin, Victor, Tatiana

Summary of action items (included inline below as well):

- AI: Michael Wong to follow up with lawyers regarding copyright assignment.
- AI: Michael Wong to encourage discussion of implementation approaches for function-local static initialization
- AI: Everyone please take a look at safe-by-default standardese,
Section 8.4.4 in particular.
- AI: Victor to follow up on new syntax and adapt basic proposal accordingly
- AI: Someone to volunteer to lead Sept 9 meeting in absence of Michael and probably Justin
- AI: Michael Wong to add special meeting for Aug 26th.

Next meeting is Aug 19th.

> The current secretary rota list is:
Torvald, Victor, Jens Maurer, Mike Spear, Hans, Tatiana, Zhihao,
Michael Wong, Maged, Justin, Michael Scott, Mark,

> Agenda:

> 1. Opening and introductions

> 1.1 Roll call of participants
> 1.2 Adopt agenda
- Yes, with changes from Michael

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

> 1.4 Review action items from previous meeting (5 min)
> 1.4.1. Michael Wong to talk to IBM lawyer.
- MW: Lawyers not responding.
- AI: Michael Wong to follow up with lawyers regarding copyright assignment.

> 1.4.2. Considering going to Chicago C++ Standard meeting, Hotel
> Booking deadline Sept 6, 2014:
> http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2013/n3577.pdf

> 2. Main issues (50 min)

> 2.1 Outstanding (additional) questions about Victor's minimalist
> proposal (see Victor's post on 2013/05/28: "Questions about basic
> proposal").

> 1-5 answered.
> 6. If a virtual function declaration specifies the transaction_safe
> attribute, must an overriding virtual function declaration also
> specify the attribute, even though it is redundant (because if the
> overriding function were not transaction-safe, the program would be
> rejected)? Also, if a virtual function has multiple declarations,
> including one that specifies the transaction_safe attribute, must
> all its declarations specify the attribute? If not, must any
> particular declaration specify it?

=> Further discussion:

- If base method annotated as tm_safe, must overrifing method also be annotated?
- Jens sent two suggestions, analogous to virtual methods and exceptions, resp.
- JM: Neither option will be liked by all.
- Michael Scott:
  - motivation for SBD was to not break existing code
  - virtual function different
- JM: feature called override: intended to override
  - slight mistake means not an override
  - people already leaning to a bit towards requiring explicit repetition
- Mark: Just talking about virtual funtions?
- JM: redeclaration case: "I already told you, why do I have to say so again?"
- MS: have to respecify args anyway?
- JM: Yes, but that is needed for disambiguating different functions.
- MS: Right.

- MM: can we settle redeclaration case?
- JM: current wording, no SBD for function pointers, need explicit annotation
- MM: can you explain?
- JM: Current wording says if taking address of a function and assigning the result to a tm_safe pointer, function must be explicitly tm_safe.
- MM: What about taking of address of function being an indication that it's tm_safe?
- JM: It's ok, in that case; we have things like that (e.g., template

argument deduction), but it doesn't cover all cases; e.g., overload resolution -- different functions with and without tm_safe arguments (otherwise identical); another example auto p = &foo.
- MW: let's not spend whole meeting on it; go for straw poll at
- Chicago

- MM: Suggest as a general principle that we require explicit annotation when we need to, but---in spirit of SBD---resist going down the slippery slope by then requiring it in more places that it's not needed.
- General agreement.

- Conclusions based on this principle:

- Function pointers: taking address of a function and assigning it to a tm_safe function pointer is taken to be implicit indication that function is required to be tm_safe.

- Redeclaration: if a function is declared tm_safe in one declaration or definition, then it is not required that this be explicitly repeated on other declarations

- Virtual functions: if a method id declared safe in the base class, then an overriding function is required to be tm_safe, but need not be explicitly declared as such.

- We agreed on these decisions, understanding that the committee might override them, but at least in that case, it's not *us* putting additional unnecessary burden on programmers.

> 7. Is the dynamic initialization of function-local statics
> transaction-safe (assuming the code in the initialization expression
> is transaction-safe) even though it likely involves some nonatomic
> synchronization under the covers?

- Question about whether it's OK to leave initialization performed by a transaction that aborts.
- MS: Isn't initialization just an optimisation, and we could do it earlier?
- MM & JM: No that's bad, initialization may depend on arguments
- HB: may also be a function of thread that initialized it.
- OK, if transaction aborts, initialization must be undone.
- Discussion of potential implementations (somewhat confused/confusing)
- JM: all we need is response from implementators that it can be implemented reasonably
- Hans: my impression is that fast path can be same, initialization code must "use txn"

- More discussion of possible implementation approaches
- VL, MS, MM: start with allowing clean semantics, see if any pushback
- JM: don't want to "sneak" past the committee, should highlight it
- MM: should bring it to attention of committee with concrete input from implementors, not confused guesses

-AI: Michael Wong to encourage discussion of implementation approaches for function-local static initialization

> 8. Discuss additional topics from Jens

JM: SBD language needs a bit more work, but mostly there
AI: Everyone please take a look at safe-by-default standardese, Section 8.4.4 in particular.

> 2.2 Write up assignments.

- JM: Mailing deadline Aug 30, we need standardese by then
- MW: need papers describing decisions
- Victor: have been keeping issues of simplified proposal on wiki page
- MW: we have drafts in bullet form, need to move towards papers
- MW: heavy travel in September, will need someone else to guide meetings, esp. Sep 9
- AI: need volunteer to chair Sept 9
- AI: Michael Wong to schedule additional meeting for Aug 26th
- JM: Basic proposal needs to be adapted to current syntax
- AI: Victor to follow up on new syntax and adapt basic proposal accordingly
- JM: we are missing convincing examples in new syntax
- Agreed
- MW: had discussion about relaxed transactions, Bjarne said don't call it a transaction if it's not a transaction
  (Editorial note: we agreed to defer a number of issues, including this one, until after Bristol. We did not achieve our goals for Bristol, so we seem to have tacitly extended deferral of these issues. I'm just sayin'.)

- MW: Availability in coming weeks?
- Mark, Tatiana, Victor, Jens expect to be around

> 2.3 Discussion of user impressions, and identification of issues to
>   be incorporated into the next revision of the specification"
> 2.4 Review future plans and away times

> 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 meetings: Aug 19, teleconference
AI: Someone to volunteer to lead Sept 9 meeting in absence of Michael
and probably Justin

AI: Michael to add special meeting for Aug 26th.

> April 29: Post Bristol report (DONE)
> May 13: Discuss memory model wording, atomicity wording (Michael Wong at C++Now) (DONE)
> June 3: Discuss transaction expressions, function transaction blocks, memory model wording and other standardese topics (Michael Wong at OpenMP and Innovate) (DONE)
> June 10: Discuss function transaction blocks, clause 6, clause 15 (Victor away)
> June 24: Discuss Outstanding questions about Victor's minimalist proposal, Review recent TM talks at TRANSACT, ACCU, ADC++, C++Now. Goal of SG5 (Justin at HotPar, Hans at HotPar).
> July 8: Discuss Outstanding questions 5 onwards about Victor's minimalist proposal,
> July 22: Continue discussion of outstanding questions 6, 7
> Aug 5: Continue discussion of outstanding 6, 7. Review recent TM talks at TRANSACT, ACCU, ADC++, C++Now, Hot PAR.
> Aug 19:Write up of proposal for Pre-meeting mailing deadline.
> Sept 9: (Michael Wong at IWOMP, Justin at PACT)
> Sept 23: Chicago C++ Std Meeting

> 5.3 Adjourn

Minutes for 2013/08/19 SG5 Conference Call

Minutes by Torvald Riegel

Roll call:

Zihao Yuan (zy), Victor Luchangco (vl), Justin Gottschlich (jg), Michael Wong (mw), Mike Spear (ms), Jens Maurer (jm), Maged Michael (mmi), Mark Moir (mmo), Torvald Riegel (tr)

> The current secretary rota list is:
>
Victor, Jens Maurer, Mike Spear, Hans, Tatiana, Zihao, Michael Wong, ,Maged,Justin, Michael Scott, Mark,
>
> 1.4 Review action items from previous meeting (5 min)
> 1.4.1. Michael Wong to talk to IBM lawyer.
Posted on reflector. Done.

AI everyone that cares: Have a look.
>
> 1.4.2. Michael Wong to encourage discussion of implementation
> approaches
> for function-local static initialization
Done.

zy: What about TM code linking to non-TM code?
ms, vl: the code doing the initialization is in one function, and that is either txsafe or not.
mmo: initialization could touch data elsewhere, but that would be a race.

Continue discussion on the reflector.

> 1.4.3. Everyone please take a look at safe-by-default standardese,
> Section 8.4.4 in particular.
mmo: should keep this as AI, need more people to look at it.
vl: gist of it is okay, just minor concerns about wording and such.

New AI All: Look at all of the standardese.
AI MW: add a link to what has to be reviewed.

> 1.4.4. Victor to follow up on new syntax and adapt basic proposal
> accordingly
Done?

>
> 1.4.5. Someone to volunteer to lead Sept 9 meeting in absence of
> Michael and probably Justin
>
> Michael Should be Available

MW might be available, or will find someone else.

>
> 1.4.7. Jens and Victor to update Standarese and Basic proposal with
> Aug 5 Decisions.

???

> 1.4.8. Considering going to Chicago C++ Standard meeting, Hotel
> Booking deadline Sept 6, 2014:
> http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2013/n3577.pdf

Evolution should have time during the Chicago meeting.

> 2. Main issues (50 min)
>
> 2.1 Standarese Review
>
> 2.2 Basic Proposal writeup Review

MW: In the next week we need to have both the standardese and the basic proposal papers written.

vl: volunteers for the basic proposal paper.
jm: rephrase questions into something else. content should stay though.
vl: intent is to resolve those.

jm: can we merge both papers, at least intro/rationale?
vl: yes maybe merge.

AI vl: will write the basic proposal, jm and vl will then see whether merging can be done.

AI mw: get paper numbers.

mmo: will try to come up with a list of examples. all of us should contribute to the final set of examples.
ms: volunteers to contribute examples.
jg: likewise.

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

[Didn't get all the details. But JM and VL seemed to agree with each other.]

jm: Asymmetry: transaction_safe is part of type, transaction_unsafe is just an annotation on function decls.

vl: what's the state of overloading with txsafe?
jm: in current state of proposal one can't overload based on txsafe vs. !txsafe. similar example: can't overload based on return type of function.
mmo: what does "part of the type" mean beyond overloading?
jm: it makes sure that one cannot assign a non-txsafe function to a txsafe function pointer, for example. so if lhs = rhs, then lhs and rhs must have the same type (modulo casts etc.).

Minutes for 2013/08/26 SG5 Conference Call

Today's meeting notes by Michael Wong, so Victor can focus on talking ...
These notes were approved in a special email dated Aug 30 “First draft of basic Proposal”

On Tuesday, August 20, 2013 3:55:26 PM UTC-4, Michael Wong wrote:
Start Time: Monday, Aug 26, 2013, 12:00 PM US Pacific Daylight Time (07:00:00 PM in GMT)
End Time: 1:00 PM US Pacific Daylight 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, August 26, 2013, 12:00 PM US Pacific Time
916-356-2663, Bridge: 4, Passcode: 3369465

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:

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

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
Victor, Jens, Mike Spear, Michael Wong, Maged, Mark, Torvald, Hans, Tatiana, Zihao
1.2 Adopt agenda

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

1.4 Review action items from previous meeting (5 min)
1.4.1. All to look at Legal Draft Wording for releasing original doc to ISO
Done.
1.4.2. All to review Standarese:
http://wiki.edg.com/twiki/pub/Wg21bristol/BasicFirstProposal/tmspec.html
Done.
1.4.3.Michael to get paper number for pre-meeting mailing of at least 2 documents
  1. SG5 TM proposal (possibly merged Basic Proposal, Rationale, Wording or be separate, TBD)
  2. SG5 minutes
Done.
1.4.4. Victor to Write-up Basic Proposal in a paper form, merging Rationale
Working on it.
1.4.5. Mark to lead a group (Mike Spear, Justin) to work on examples. We need lots of examples as the wider Evolution Group has exponentially less TM knowledge
Working on it.
Michael Should be Available
1.4.6. Considering going to Chicago C++ Standard meeting, Hotel Booking deadline Sept 6, 2014:
http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2013/n3577.pdf

2. Main issues (50 min)

2.1 SG5 TM Proposal writeup Review
Example thread: Mark filled in a lot, need someone who speak C++ to check them carefully, several bold comment paras mentioning various issues to discuss. Should summarize those here:
1.6 Examples illustrating that partial effects of transactions that cause std::terminate to be called cannot be observed
1.13 Example illustrating function-local static initialization in atomic transactions
Mark mentioned 2 possible choices. Push forward to get it right, or outlaw it
Hans mention a third choice where we make function local statics (FLS) transactional.
Hans:What do we guarantee done atomically, whole wording is vague
Jens: FLS does not seem to related memory model
Hans: what do impl to, is it only on the constructor call
Jens: lock recursive local mutex, any other have to wait for the first to complete
Hans: so it is the entire RHS including the actual init, because RHS might call the fn to be initialized recursively
Jens:footnote saids it must not introduce deadlocks
Hans:in essence, it is already saying it needs to be init transactionally. We need to think about to phrase it
Victor: see if we get the issue right, the wording will be adjusted after feedback
Hans: want it to act like a relaxed transaction

no place we require an impl to support that
Victor: implied that we dont say that it is unsafe
Jens: practical suggestion is that we wont have a solution in this call, just say in explainatory section that this is open issue, we want this semantic, but it is not reflected yet in the wording
Victor: in the note tx safe section, the formal wording does not reflect this
Note: The dynamic initialization of function-local statics is transaction-safe (assuming the code
in the initialization expression is transaction-safe) even though it likely involves some nonatomic
synchronization under the covers.
Hans: require tx in atomic mode, revert to irrevocable ode when u notice it is not tx safe, we cant avoid that
Jens: even if we outlaw it , not a big deal, current std seem deficient
Hans: concern about deadlock issues there
Mark: capture more of the issues so we dont rediscover it again
Victor: AI: will update this issue
Hans: no consensus that we can get it right

1.6 Examples illustrating that partial e
ffects of transactions that cause std::terminate
to be called cannot be observed
Jens: might do unsafe stuff
Mark: does not matter it does unsafe stuff, does not comit, this thread makes no progress, other thread does not obserev this thread
we are not sure, may need input from implementer
Mike S: does that include racy programs?
Victor: racy programs allow arbitrary behaviour
Mike S: make sure this example that a TM impl does not imply a redo base (?)
Victor: I need to say something about that so not surprising
Torvald: need to rollback, and not commit, so as to allow terminate
Mark: rollback is not needed, just need to not commit
now impl call std::terminate, half executed tx is still half execuuted, (pending and not committed)
Jens: concern that terminate handler does unsafe thing
Mark: since we dont need to commit this tx, then we dont care
Hans: are we running code inside a tx?
Victor:terminate is not inside tx
Mike S: tx calls some fn, changes value of FLS, then while terminate running, it must access that FLS (like a std library function accessing it), even though my tx is invisible, Mark point is to not worry about my tx not visible to other thread, but can the effect be visible to thread running the handler
Victor: should handler see FLS be init?
Mark: what the thread does that is terminating, it needs to figure out for itself, other thread can see effect of half terminate tx can do all kinds of stuff

Torvald: we need to specify for user what the behaviour be? Agree for other thread it does not matter, for the same thread we need to say something
Hans: want the debug handler to see state before rollback
Mark: been trying to reflect others concerns and listen and understand
Victor: why allow the terminate handler be able to see
Jens: terminate handler is a ptr, declared to be not tx safe, understand we dont care for it to be tx safe. rest of language say we can call tx unsafe stuff from inside atomic tx, so we cant be in there
Mark: constraints for programmer is different then for impl
understand that the terminate handler is a fn ptr and that might be changed by the transaction
Hans: like FLS, we also want it to backoff dynamically
Jens: say current standarese does not reflect this in 1.6
Michael: committe may also want to dive into this, we need to decide whether we want them to help, or wait for us to decide
zihao: should say something our expected behaviour,
Victor, dont think we want to require it to be tx safe
Mark: question: what is the behaviour of that (terminate) thread? Can we leave that to impl dependent? dot require tx be rollback, when throwing exception from a noexcept tx, that would require rollback, and the whole point of noexcept tx is to be able to optimize them
Zihao: terminate handler must terminate program, cannot return

Victor to discuss Basic Proposal paper
all feedback added except dynamic init of function local statics, and we have an agreed to direction on this topic
atomic vs relaxed: why do we have these 2 things?
Jens: very surprising that we have 2 different kinds
we need motivation as to why there are 2 different things
we will not fix the naming yet

Mark: this will be first paper we will talk about relaxed and SGL
Tatiana: hope we dont design relaxed tx in terms of locks
Victor: standarese is unchanged, so not the issue, just the standarse says it is equivalent to having a mutex lock ....
Tatiana: equivalent is not a problem, will reread
Mark: we agreed on the semantics is since the rest of the committee heard from us, can have a concrete discussion
now we can answer them highlight
Mike S: section 4 should provide 1 sentence example, can output data without worry about races
Victor: code examples should be in the example, not eager to add a para
Mark: can we add pointers frm one section to another
Michael: like to get original sources so that we can make last minute changes
Mark: ask Jens to read Victor's sections
Jens: AI: will adjust it with a red pen and send it to Victor
Also Michael should practice putting the pieces together

Michael: Ok, can already to latex
Tatiana, AI: please motivate lawyers to publish Legal Draft Release wording with original document

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 meetings: Sept 9, teleconference

April 29: Post Bristol report (DONE)
May 13: Discuss memory model wording, atomicity wording (Michael Wong at C++Now) (DONE)
June 3: Discuss transaction expressions, function transaction blocks, memory model wording and other standardese topics (Michael Wong at OpenMP and Innovate) (DONE)
June 10: Discuss function transaction blocks, clause 6, clause 15 (Victor away)
June 24: Discuss Outstanding questions about Victor's minimalist proposal, Review recent TM talks at TRANSACT, ACCU, ADC++, C++Now. Goal of SG5 (Justin at HotPar, Hans at HotPar).
July 8: Discuss Outstanding questions 5 onwards about Victor's minimalist proposal,
July 22: Continue discussion of outstanding questions 6, 7
Aug 5: Continue discussion of outstanding 6, 7. Review recent TM talks at TRANSACT, ACCU, ADC++, C++Now, Hot PAR.
Aug 19:Discussion of Std writeup. Write up of proposal for Pre-meeting mailing deadline.
Aug 26: Review Write up
Sept 9: (Michael Wong at IWOMP, Justin at PACT) will work on presentation for C++ Std meeting

Sept 23: Chicago C++ Std Meeting

5.3 Adjourn