

Document Number: N3695
Date: 2013-06-28
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 2013/03/11-2013/06/10

Contents

Minutes for 2013/03/11 SG5 Conference Call	2
Minutes for 2013/04/08 SG5 Conference Call	10
Minutes for 2013/04/29 SG5 Conference Call	14
Minutes for 2013/05/13 SG5 Conference Call	23
Minutes for 2013/06/03 SG5 Conference Call	25
Minutes for 2013/06/10 SG5 Conference Call	30

Minutes for 2013/03/11 SG5 Conference Call

Minutes by Mark

> Start Time: Monday, March 11, 2013, 12:00 PM US Pacific Time (08:00:00

> PM in GMT) End Time: 1:00 PM US Pacific Time (duration: one hour)

>

>

>

> The current secretary rota list is:

>

Paul M, Hans, Victor, Mike Spear, Jens Maurer, Tatiana,
Michael W, Justin, Maged, Torvald, Michael Scott, Mark

>

> 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

Mike Spear

Michael Scott

Torvald

Victor

Hans

Maged

Justin

Mark Moir

Michel Wong

> 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
- >
- > 1.4.1 Maged has the IBM legal contact names, checking Oracle and Intel to set up a joint time to meet.
- > Intel trying to connect with a lawyer, Oracle and IBM has
- > identified their legal rep. Intel has internal dependencies.
- > IBM/Intel/Oracle people to keep after their respective lawyers to
- > free up the current spec for submission; use Bristol as leverage

No update, still waiting for response from Intel legal.

- >
- > 1.4.2 Victor to attempt to create new doc, possibly extending N3341
- > - others to help/review/object as fast as possible

Done.

- > 1.4.3 Mark to write up SBDBD or "reducing annotation requirements
- > for transaction_safe functions". - possibly to be incorporated into
- > Victor's doc

Done.

- >
- > 1.4.4. Torvald to write up his alternate exception proposal

Some discussions on exceptions escaping, but proposal action item carried over.

- > 1.4.5. Tatiana or Justin to send around source for N3341

Done.

- > 1.4.6 Everyone update your status for Bristol meeting: Bristol UK
- > registration deadline Feb 12: preferred dates of our meeting (Tuesday
- > April 16 to Thursday April 18)
- >
- > The following link will take you to the WG21 tab of the ACCU
- > booking site:
- >
- >
- > <http://www.cvent.com/events/accu-2013/custom-22-09ec03b22c4f4a0a832e28126a4585fc.aspx>
- > Select wg21 registration only
- > select full wg21 event
- > select Marriot if possible

- > select April 16-18 hotel as a minimum, but you are welcome to
- > increase this time.
- >
- > The longer invite N3397:
- > <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2012/n3397.pdf>
- >
- > > Michael W: Who is going to Bristol?
- > > Justin: Tatiana is not going
- > > Justin: Justin might go.
- > > Torvald: Going
- > > Jens: Going
- > > Mark: 50/50
- >
- > Mark is most likely not going.
- >
- > > Maged: Might go
- > > MLS: Not going
- > >
- > Victor: likely going.

No update discussed.

- >
- > 1.4.7. Everyone: Review important links:
- > a. Prioritize TM discussion ahead of Bristol based on the following
- > page
- > <http://wiki.edg.com/twiki/bin/view/Wg21bristol/SG5>
- >
- > b. Writing Standarese for current TM draft
- > <http://jmaurer.awardspace.info/wg21/tmspec.html>
- >
- >
- > 2. Main issues
- >
- > 2.1 Continue to Discuss P2 item of Suggestion 3 Unified of
- > <http://wiki.edg.com/twiki/bin/view/Wg21bristol/SG5>
- >
- > Actively continue with the current exception discussion on how
- > nesting, cancel, and exceptions interact. Make a decision on the
- > exception behavior of the strong/atomic transactions once we have a
- > detailed proposal for abort-on-exception behavior that addresses all the
- > issues. Make the decision on [[outer]], [[may_cancel_outer]] and
- > cancel-throw in the context of the same discussion.
- >
- > 2.1.1. Review Victor's proposal writeup

Editor's note: I am sorry that I was unable to do a very good job of capturing this conversation. Between audio problems and trying to listen, think and type at the same time, I really lost parts of the conversation. Hopefully there are some useful keywords below to job peoples' memories at least :-). Apologies to those I've misrepresented or failed to represent.

Maged: happy with it given that more controversial stuff about cancellation has been removed.

Discussion of `transaction_callable`.

Mark: no change - it is still a semantic-free hint.

Someone: If we're getting rid of `transaction_safe`, why not get rid of `transaction_callable` too?

Mark: We're not getting rid of `transaction_safe`, we're making it optional. `transaction_callable` is already optional.

Michael: my students say it's harder to figure out what to annotate as callable than safe.

Mark: Some compilers may completely ignore `transaction_callable`, others may do amazing optimisations. So question is really for tools, QOI, not language spec.

Hans: if we infer safe-by-default, why don't we assume `transaction_callable`?

Someone: Do we need `transaction_not_callable`?

Someone: We already have `transaction_unsafe`.

Mark: Does not serve same purpose, as `transaction_unsafe` means not to be used in atomic transactions, not that it won't/can't be used in relaxed transactions.

[lost some discussion here, someone suggested there was disagreement about what `transaction_callable` means]

Mark: what disagreement?

Michael Spear: my impression of `transaction_callable` is "please produce an instrumented version even though I know I may need to go

irrevocable sometimes".

Others: agreed.

Mark: I also agree, but would not state it in implementation terms. We are just saying "this function may be used in relaxed transactions". This gives the compiler a hint, and it may optimise on that basis or not.

Hans: how to do we handle indirect calls?

Torvald: anything that could have an indirect call should have a clone. So we need it to be part of the type. Definitely harder for implementation to ...

Victor: suppose no transaction_safe type for function pointers. Then no way to ensure transaction_safety.

Torvald: is annotation viral on function pointers?

Hans: also virtual functions.

Hans: inheritance must respect transaction_safe annotations.

Hans: seems like explicit annotations are needed.

Victor: ?

Mark: different topic.

Michael S.: tradeoff, choosing to require annotations to avoid dynamic checks. We should make the explicit.

Mike S.: what about a way to explicitly state it, otherwise you get a runtime error?

Victor: could be ...

Justin: Could consider dynamic cast. Will cast at runtime.

Mike S: You didn't put in the effort to annotate the function pointer, so you don't get the benefit of static checking.

Michael S: ...

Something about inverse cast?

Victor: indirect function call in a transaction that turns out to be unsafe.

Torvald: this was easier with SBD, now it's SBDBD.

Mark: right.

General consensus(?): we like SBDBD, but we need to figure out this issue.

Michael S: haven't yet heard convincing argument that we can't do static checks. E.g., what's different about const?

Torvald: trying to avoid syntactic overhead of const.

Torvald: could have one safe target, one unsafe. Can't tell at compile time which it might be.

Michael S: we just decided we need dynamic checking for function checking.

Torvald: could check statically, but conservatively. It's an option.

[lost the conversation here]

Hans: AI - send an example showing why we are forced to do dynamic checking.

Michael W: Are we comfortable with sending two documents: summary since Portland and next spec outline?

More discussion of what needs to be done to "next spec outline".

Mark: Consider just sending one document (summary since Portland), as deadline is close, not enough time to do a good job of draft spec?

Victor: still have two semantic questions to resolve: indirect calls, exception escaping.

Michael: submit even half baked.

Mark: move to send first paper, defer decision on other until later in the week.

All: agreed.

> 2.1.2. Review Mark's SBDBD proposal writeup

Captured (to some pathetic extent :-)) in previous section.

> 2.1.3. Review Torvald's proposal writeup

Not done yet. Torvald considering getting something ready for Bristol submission.

> 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

All: give feedback on summary of learnings document.

Mark and Victor: edit to reflect feedback, prepare for submission.

All: give feedback on draft spec outline.

All: consider whether to submit draft spec outline.

Mark: send titles of four potential documents to Michael W, include Torvald proposal. And to make this a self-fulfilling action item:

- 1) Summary of progress since Portland.
- 2) Draft outline of next version of spec.
- 3) Summary of discussions on explicit cancellation.
- 4) Torvald's alternative proposal for cancellation and exceptions.

Michael W: request numbers for four documents.

>

>

> 5. Closing process

- >
- > 5.1 Establish next agenda
- >
- > 5.2 Future meetings: Mar 11, teleconference
- >
- > Jan 14: Priority discussion (DONE)
- > Jan 21: Discuss P2 item (DONE)
- > Feb 04: Continue P2 item (Bristol registration ends about Feb 12)
- > (DONE)
- > Feb 25: Continue P2 item; writing begins for Pre-Bristol deadline
- > (DONE)
- > Mar 04: Discuss early writeups on Victor + P1 (SBDBD) item +
- > Torvald proposal (Pre-Bristol mailing deadline: 15 March 2013) (DONE)
- > Mar 11: Continue review of writeups (Further discuss Bristol paper
- > submissions: March break 1st week, many away at conference next week)
- > Apr 01: Prep for Bristol, mostly slide prep (last call before
- > Bristol;also Easter Monday, this is not a joke)
- > Apr 15: Bristol

Next meeting April 1st. No, really.

- > 5.3 Adjourn

Michael W: move to adjourn.

Mark: second.

Minutes for 2013/04/08 SG5 Conference Call

Minutes by Victor

> The current secretary rota list is:

Paul M, Hans, Mike Spear, Jens Maurer, Tatiana, Michael W, Justin, Maged, Torvald, Michael Scott, Mark, Victor

> 1.1 Roll call of participants

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

> 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

>

> 1.4.1 Maged has the IBM legal contact names, checking Oracle and Intel to set up a joint time to meet.

> Intel trying to connect with a lawyer, Oracle and IBM has identified their legal rep. Intel has internal dependencies.

> IBM/Intel/Oracle people to keep after their respective lawyers to free up the current spec for submission; use Bristol as leverage

It is going higher and higher at IBM: Oracle exec called IBM exec in charge of standards (Gerald Lane?), who wants to know what exactly the Oracle lawyers want in addition to the ISO documents.

> 1.4.2 All: give feedback on summary of learning document. Mark and Victor: edit to reflect feedback, prepare for submission.

DONE.

> 1.4.3 All: give feedback on draft spec outline. All: consider whether to submit draft spec outline.

DONE. Decided not to send.

> 1.4.4 Mark: send titles of four potential documents to Michael W, include Torvald proposal. And to make this a self-fulfilling

> action item:

>

- > 1) Summary of progress since Portland.
- > 2) Draft outline of next version of spec.
- > 3) Summary of discussions on explicit cancellation.
- > 4) Torvald's alternative proposal for cancellation and exceptions.

DONE.

- > 1.4.5 Michael W: request numbers for four documents.

DONE.

- > 1.4.6 Everyone update your status for Bristol meeting: Bristol UK registration deadline Feb 12: preferred dates of our meeting (Tuesday April 16 to Thursday April 18)

DONE.

- > 1.4.7. Everyone: Review important links:
 - > a. Prioritize TM discussion ahead of Bristol based on the following page
 - > <http://wiki.edg.com/twiki/bin/view/Wg21bristol/SG5>

- >
- > b. Writing Standarese for current TM draft
- > <http://jmaurer.awardspace.info/wg21/tmspec.html>

- >
- >
- > 2. Main issues

- >
- > 2.1 Review slides for Bristol based on the following submissions:
 - > <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2013/#mailing2013-03>

- >
- > Summary of Progress Since Portland towards Transactional Language Constructs for C++
- > <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2013/n3589.pdf>
- > Summary of Discussions on Explicit Cancellation in Transactional Language Constructs for C++
- > <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2013/n3591.pdf>
- > Alternative cancellation and data escape mechanisms for transactions
- > <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2013/n3592.html>

Victor and Torvald talked through their respective slide sets. See those slides. Some comments made about slides recorded below (this is spotty). Mostly feedback that will be incorporated into slides.

Victor:

Slide 6: Note that although relaxed transactions can be implemented with SGL, this is not recommended. Benefit of relaxed transactions is no instrumentation

Slide 8: Note that no instrumentation needed for reads and writes in trivial implementation

Slide 9: composability vs failure atomicity distinction? (Mike Spear)

Slide 11: modularity issue is part of what makes semantics hard to understand

Slide 12: flat vs closed only matters if transactions may be (partially) cancelled. Since we have no explicit cancellation

Torvald:

3. Cancellation: This is different than exceptions. You can use this for speculation.

6. Victor: Don't we agree that cancellation and exception should be separate?
Michael Scott not convinced.

8. Can call `escaping_memcpy` inside a transaction, it reads data transactionally, writes it out non-transactionally, communicate data outside of tx

Not limited to cancellation; its semantics is defined with respect to cancellation.
"abort" is forced by system, not user; transaction must retry.

There may be thread-level speculation under the cover: not used directly by programmer

Can't generate automatically, reference counting is incompatible with data escape.

9. Declare certain memory (allocated by special allocator) as escaping: all writes to it escape

Hans: high-level form is more convenient, but more error prone

Michael Wong: may be a problem that different objects of the same type might use different allocators.

Victor: Perhaps safer for each declaration to say explicitly that it is escaping, rather than every declaration delimited by `begin/end`

Hans: could also standardize the scoped form

(Both alternative forms require syntactic support.)

11. Only transactions with attaching cancellation handler (CH) can be cancelled

13: will not terminate -> will not call `terminate()`

14: picks the nearest enclosing with matching cancellation handler

This preserves closed nesting, can know at compile time where we need closed nesting and

where we don't

15. combine cancel with data escape

16. use data escape mechanism: $i = 3$ (not 1) after transaction

Note that here, `transaction_cancel` takes an argument: a function to execute (which takes the cancellation handler as an argument).

Requires function object to escape: make clear that this is implicit for the argument of `transaction_cancel`

This can be done in a library

> 4.2 Review action items

TODO: Victor and Torvald to update their slides, send out new versions.

TODO: Inquire from Oracle what exactly their lawyers want beyond ISO documents.

Next meeting at Bristol (Apr 15), then phone meeting on Apr 29.

Minutes for 2013/04/29 SG5 Conference Call

Minutes by Mike Spear

Start Time: Monday, April 29, 2013, 12:00 PM US Pacific Time (08:00:00 PM in GMT) End Time: 1:00 PM US Pacific Time (duration: one hour)

The current secretary rota list is:

Paul M, Hans, Mike Spear, , Tatiana,Michael W, Justin, Maged, Torvald,Michael Scott,Mark, Victor, Jens Maurer

Updated to

Paul M, Hans, Tatiana,Michael W, Justin, Maged, Torvald,Michael Scott,Mark, Victor, Jens Maurer, Mike Spear

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

Present:

Mike Spear, Michael Wong, Jens Maurer, Maged Michael, Torvald Riegel, Victor Luchangco, Justin Gottschlich, Tatiana Shpeisman, Michael Scott, Mark Moir, Hans Boehm.

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 (limited to 5 min)

1.4.1 Maged has the IBM legal contact names, checking Oracle and Intel to set up a joint time to meet.

Mark: Not done yet, but some progress in understanding.

Michael W: Should continue this over email.

Mark: Awaiting response from Oracle on one issue, will email once updated.

1.4.2 Victor and Torvald to update their slides, send out new versions.

Done

1.4.3. Everyone: Review important links:

a. SG5 Chair report and minutes from Bristol April 17 meeting

<http://wiki.edg.com/twiki/bin/view/Wg21bristol/SG5>

b. Writing Standarese for current TM draft

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

2. Main issues (50 min)

2.1 Bristol meeting report (Michael, Victor, Torvald, Hans, Maged ,Jens)

Michael W: Many straw polls, lots of feedback from Bristol. Summary of impressions/feedback:

Michael W: sumamry of straw polls -- many polls; overwhelming suggestion to move quickly with standarese for a simple first version. They cancelled an evening session to give SG1 more time, and this lets us delay to Chicago.

-- No consensus on changing name of relaxed transactions

-- Negative on explicit cancel

-- Even on cancel on escaping transaction

-- Weak yes to commit on escaping exception

-- Simple data escape: yes, but tied to cancel on escape

-- On advanced data excape: clear no

-- On atomic and relaxed transactions: both were a clear yes

In post-meeting meeting, Victor volunteered to write up a simple specification, which Jens is following up with wording / managing minute details and standarese.

Jens adds: very few people in the room apart from TM regulars when the presentation went on, and only 3-4 of them were vocal in voicing an opinion vs. listening, so results of polls could change if we had a bigger group voting. Note: the document is at best a rough guideline, good enough for the

short term.

Michael: It's a data point, not long-term guidelines.

Jens: We now have non-zero data points, but still we may get other data points later.

Justin: Question about escaping (commit/cancel) exceptions, but nothing about noexcept for transactions. Maged states that the guidance was "keep it simple". Michael W confirms that there is awareness that we'd need to define /something/ on exception. Victor suggests we focus on looking forward, not interpreting attitudes from people. Michael Scott worries that noexcept is going to cause lots of try/catch(all) to litter TM code if we go with noexcept, like catch(InterruptedOperationException) in Java. Mark? and Justin agree.

2.2 Review future plans, goal and away times (Michael)

Michael W: plan is to take a simplified specification and move it to standardized wording. If we make a document available for Chicago, we can announce a work item for a tech specification, which can go to SG22, get public comments, i.e., national bodies' standardization organizations. If we get 5 votes from national bodies, we get approved as a technical specification, but need 2 more rounds of ballots first, which go up to more levels of national bodies. Each ballot takes 3-6 months, so once we get a work item (and we attach a timeframe), it still will take at least a year (3 is usually the most generous timeframe). Usually folks ask for 3, so they can finish early.

Jens: But to submit at Chicago, SG22 wants to be sure that we're likely to be able to finish. It's conceivable that we could be done in very short order, but a tech spec gets more publicity to help things get implemented and move along more quickly. Note that features don't change radically during the comment process.

Hans asks if there's an advantage in getting a tech spec early, can we combine two ballots. Jens: we could combine the first two ballots under some cases, but it still takes at least a year. It takes time.

Mark: Concern that document isn't going to serve the purpose, that the manner of drafts we wrote before would be more valuable (vs. standardese) because they are more readable.

Michael W: could have a supplemental document/book/set of papers about what the standard /really/ says. It is normal procedure to have these other documents. Jens: but they don't come out of the committee process

necessarily.

Tatiana: A compromise is for us to derive a sufficient number of committee papers in support of the standardese, and we could set a goal to have these ready for Chicago.

Michael S: Have we given up on revising the current draft spec?

Michael W: We need to give ourselves the freedom to write the standard specification; using that document will require legal permission. Mark and Tatiana suggest that we can get to that point.

Michael S: Is the current spec supposed to get turned into the standardese?

Victor: Jens will write 'diffs' of how to add these features to the current C++ spec. Standardese isn't document saying how TM fits, it's explaining how the C++ specification needs to be changed to support TM, so that compiler implementors know what they need to change.

Michael W: Do we need to do more on the old draft spec, or can we just write something new after we do the standardese?

Victor: We could do standardese and the new document concurrently. Victor had a different structure, but he did abbreviate some of the content that wasn't changing from the old draft specification. The back-references were really about doing things quickly, we can pull that text in eventually (e.g., memory model, inheritance of annotations) -- there is enough substance, but not the details that were in the draft specification. He wants to use that as an outline, build it into a new draft specification, with some direct copy-and-paste.

Tatiana: Could we try to come up with a new informal specification while Jens moves it into standardese?

Victor: We need (first) to get a conceptual agreement on the contents of the first version. It won't satisfy all of us, but we need something that doesn't have too many objections, hopefully the only objections are what's missing, but the document must be an acceptable compromise that people will actually be able to use. It can't be too impoverished so people won't use it, but it can't be too rich and complex.

Tatiana: We should have footnotes about stuff that might get dropped from the final specification for anything that is of question.

Victor: We shouldn't have more than 3-4 of these.

Consensus that we should iron these things out now at the conceptual level.

Michael W: Then let's look at the Victor proposal. There have been a few issues raised, might take 1-2 meetings to go over.

Jens: Can we start with what isn't objectionable.

(Now Victor leading discussion): Goal is a conceptually complete proposal from the simplified proposal put forth earlier. Reminder: don't worry about spelling/syntax. Focus is on concepts: atomic txns with one of three types to handle exceptions (cancel is the only one with slightly complex semantics on exception; commit and noexcept are straightforward).

Justin: concern about use of attributes being misused; Michael W suggests tabling it -- it's a spelling issue, we know they need to become keywords. Victor agrees -- table it, we know this needs to change, let's not worry about spelling, it'll probably become keywords.

Hans: regarding noexcept: it /is/ a keyword with a different meaning in the rest of the language. Victor suggests that again we defer spelling, we could call it 'terminate_on_escape'.

Victor: The simplified proposal includes description of transaction unsafe. It has a (probably inadequate) discussion of safe by default (SBD), which led to some questions embedded in the text. Memory model not specified, since the old draft spec is mostly fine, and needs to be handled by Jens with Victor and Hans when the time comes.

Relaxed transactions are all as follows from our previous discussions. Open question about transaction_callable, do we need it? He thinks no, due to SBD. Mark: since it has no semantic meaning, we can probably ignore it. Tatiana: let's leave this for another discussion.

Transaction safety differs from old spec in terms of when conditions are checked. This was trickier than expected, but a few questions: should it cover lambdas? It probably needs to be the same, that's what we intend, it's just a question of being consistent with how C++ treats lambdas vs. functions (if at all).

Another open question: transaction_unsafe on function pointers. This can't always be done at compile time or link time.

Everything about inheritance we moved in with unsafe/safe. Victor would like us to drop the text on const? attributes.

Most serious question was on overriding on virtual functions, and how it

would work, but Victor isn't sure he understands the issues well enough.

Victor kept transaction expressions and functional blocks. They are easy to specify, but may not be necessary (slightly favor keeping, but not going to fight it; Robert advised that we'll get more feedback for not having it when people want it).

Class attributes were removed. They were motivated about having to annotate stuff as safe, and this was supposed to be a convenience, and it goes away once we have safe-by-default.

Tatiana: Simple alternative is to have some kind of scope mechanism to handle this. Probably doesn't need to go to Chicago, but there are ways to handle this.

Victor: Right, it's convenience, we'll see if people complain, and if so we can figure out what to do about it. If SBD isn't convenient enough, we'll deal with it. Not same as inheritance stuff, which has to be there for type safety.

Victor: what open questions? Expressions, function blocks, anything else?

Michael S: Need a more careful look at memory model. The redefinition of relaxed overcomplicates it, and now that relaxed is described in terms of a lock, this gets much simpler while also making isolation much clearer.

Victor: yes, that was part of why to dodge; it might not get drastically simpler, but this area definitely needs to change, and it's on Victor's todo list. He didn't have time to get to it for this meeting. Also, we know what we intend, so it's not controversial. We just have to do it.

Tatiana: Do we intend to allow locks and atomics inside of atomic transactions? Old spec said no other synch inside of atomic transactions. It was handled via `transaction_unsafe`. But now if we wanted to support it, it would affect the memory model section.

Michael S: We don't want to support them now. Justin agrees. Victor agrees, but emphasize that the point is "not now", so let's not lock ourselves out of adding them in the future. We should make sure the memory model makes sense given our new understanding of relaxed transactions, and that might make it easier to handle this (locks/atomics) in the long run. The memory model is going to be tricky.

Tatiana: Regarding libraries, do we need to specify which library functions have to be safe/unsafe?

Jens: Yes. We can have boilerplate wording, just highlighting what **isn't** unsafe, such as containers, but we have to do something. SBD doesn't help for the specification, as it's an implementation detail. We must say what functions are required to be safe or unsafe. Probably we can just say everything is allowed to be unsafe, except for those in a list we specify. There are going to be some portability challenges here that we'll need to work out.

Torvald: Advise a practical route with an agenda starting from what is essential, then see what we can achieve before Chicago. If we focus on the other stuff first, we might not make the deadline.

Victor agrees. He wants to know what in his proposal do we agree on, what don't we agree on. Then we can iron out those pieces, before moving on to libraries and other contentious stuff.

Michael W: Action item for May 13: Need to work on some of these items.

Jens: Let's start with the memory model first. Michael, Victor, Jens, and Tatiana want to look at this.

Tatiana wants a pointer to the latest SBD work. Hans thinks this is the most significant hole at the moment, followed by the list of exceptions we do/don't support. Thinks integral types aren't necessarily going to encourage the right programming style. Maybe we should have a subclass of it.

Mark: What I said is that SBD is mostly about what we don't **say**. In other words, things the previous spec **required** (regarding safety attributes of functions to be called within transactions) are no longer required. Thus, SBD is most naturally reflected in a complete proposal that does not impose such requirements.

Action items:

- Jens will send memory model text
- Tatiana wants the SBD text (who will send it)? Mark: it's about what is **not** safe, so we need a proposal first.
- Should we subclass exceptions?

Justin will chair next meeting.

April 29: Post Bristol report, Transact report

May 13: Michael away

May 27:

June 10
June 24:
July 8:
July 22:
Aug 5
Aug 19:
Sept 9: Michael away
Sept 23: Chicago C++ Std Meeting

2.3 Transact Report (Justin, Tatiana, Michael Spear)

Deferred to an unspecified later meeting.

3. Any other business

N/A.

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)

-- Jens will send memory model text

-- [Unspecified Lead] Should we subclass exceptions?

5. Closing process

5.1 Establish next agenda

5.2 Future meetings: May 13, teleconference

Justin will chair next meeting on May 13th.

April 29: Post Bristol report, Transact report

May 13:

May 27:

June 10

June 24:

July 8:

July 22:

Aug 5

Aug 19:

Sept 9:

Sept 23: Chicago C++ Std Meeting

5.3 Adjourn

Minutes for 2013/05/13 SG5 Conference Call

Minutes by Hans Boehm

May 13, 2013 SG5 Minutes

Attendees:

Justin Gottschlich
Maged Michael
Michael Wong
Mike Spear
Victor Luchangco
Jens Maurer
Mark Moir
Hans Boehm
Michael Scott
Tatiana Shpeisman

1-1.3: Minutes and agenda approved

1.4 Action items from last time:

1.4.1 Legal update:

Mark: Lawyers met last Friday to discuss acceptable copyright.

Justin: Tatiana also following through on Intel side.

Action Item 1: Michael Wong to talk to IBM lawyer.

1.4.2 Jens to send memory model text:

Memory model discussion has been active on mailing list.

1.4.3 Should we subclass exceptions from `std::exception`?

Jens: Integers add no difficulties. Allow them.

No fundamental difficulty in supporting predefined subclasses.

Can also allow user-defined subclasses that don't add data members.

Victor: Issue is fairly orthogonal to the rest of the specification.

Conclusion: Look at allowing such subclasses of `std::exception`.

Mark: Sounds good. Avoid making this too inflexible.

2.1 Memory model wording:

Michael Scott gave overview.

We currently get atomicity for free, since atomic transaction bodies are sync-free code.

Currently the nonnormative statement about sequential consistency is accidentally restricted to locks and `seq_cst` atomics.

Needs tweaking: Should allow transactions.

Jens: Uneasy about forward-looking statement, since path to allowing synchronization inside atomic transactions is unclear.
Conclusion: Keep the gist of current wording.

Action Item 2: Michael Scott to tweak memory model non-normative wording.

2.2 Atomicity wording:
Included under 2.1 discussion.

2.3 Other work on standardese:
Need wording for cancellation.
Jens: Snapshot values of memory locations at beginning of atomic transaction. Restore snapshot on cancel, one transaction at a time.

Action Item 3: Jens to add some wording that cancelled transactions can still be racy in programs that are not data race free.

Jens:
Next major item: Make transaction-safety typing rules air-tight.

Also need actual transaction block wording (clause 6). Current wording doesn't distinguish atomic and relaxed transactions properly.

Action Item 4: Everyone help Jens pin down transaction safety rules and with clause 6.

Tatiana: No agreement yet on removal of transaction expressions.

Move Jens' and Victor's page to EDG wiki.

Action Item 5: Victor to add web page for his minimalist proposal.

Action Item 6: Jens to do the same and add link to Victor's proposal.

Jens: Would prefer to remove transaction expressions in favor of lambdas.
Discuss at next meeting. Also look at function transactions.

Action Item 7: Everyone to consider whether transaction expressions should be part of TS.

2.4: Future plans, away times

Action Item 8: Everyone to send Justin and Michael Wong their conflicting dates for SG5 meetings through September 23.

Minutes for 2013/06/03 SG5 Conference Call

Minutes by Tatiana

SG5 meeting on 6/3/2013

Attendees:

Justin Gottschlich

Jens Maurer

Tatiana Shpeisman

Michael Scott

Maged Michael

Hans Boehm

Michael Wong

Torvald Riegel

Victor Luchangco

Yuan Zhihao

1.2-1.3 Minutes and agenda approved.

1.4. Action items from previous meeting

1.4.1 Action Item: Michael Wong to talk to IBM lawyer.

Michael Wong has contacted IBM lawyer, but not connected yet. Ongoing.

1.4.2 Action Item: Michael Scott to tweak memory model non-normative wording.

Done. In the main section of the call, we should make sure everybody can live with the current state of 1.10 at <http://wiki.edg.com/twiki/pub/Wg21bristol/BasicFirstProposal/tmspec.html>

1.4.3 Action Item: Jens to add some wording that cancelled transactions can still be racy in programs that are not data race free.

Jens: Done, see wiki page above.

I consider clause 15 in a reasonable state right now, so please have a look. (It doesn't carve out the set of "transaction-safe exceptions", as I've temporarily named them, but that should probably go into the library section anyway.)

1.4.4 Action Item: Everyone help Jens pin down transaction safety rules and with clause 6.

Jens: Still open with me.

1.4.5 Action Item: Victor to add web page for his minimalist proposal.

Done:

<http://wiki.edg.com/twiki/bin/view/Wg21bristol/BasicFirstProposal>

1.4.6 Action Item: Jens to do the same and add link to Victor's proposal.

Done:

<http://wiki.edg.com/twiki/pub/Wg21bristol/BasicFirstProposal/tmspec.html>

1.4.7 Action Item: Everyone to consider whether transaction expressions should be part of TS.

Discussed at meeting.

1.4.8 Action Item: Everyone to send Justin and Michael Wong their conflicting dates for SG5 meetings through September 23.

Ongoing.

2. Main issues.

2.1 Should transaction expressions be part of TS?

Victor: We all agree that equivalent functionality can be expressed with lambdas, it's just ugly. So, the question is whether we need transaction expressions as syntactic sugar.

Michael Scott: Everybody agrees. Let's leave transaction expressions out for the first round.

Tatiana: Agree with Victor's statement. Ok to leave them out for the 1st round.

Justin: OK to leave them out for the 1st round.

Justin: The item is closed

Jens: Asking Victor to update the wiki page with this info.

Action item: Victor to update the wiki page to reflect the decision not to include transaction expressions in the phase one.

2.2 Should transaction function blocks be part of TS?

Jens: We agree that function transaction blocks are not necessary for normal functions. Do we need function transaction blocks for constructors? It's probably not well guided idea. A post showed that this construct does not solve all the problems.

Victor: Michael Spear gave an example of how function transaction block is useful.

Michael Scott: I am not sure I understood all the discussions. Is the only purpose to wrap the constructor initializers and the body of the transaction in a single transaction?

Michael Wong: The work around is to move initialization in the body. There is no syntax to initialize arbitrary combinations of the initializers inside a transaction.

Tatiana: We should not get rid of the ability to execute the whole constructor as a transaction just because we cannot do any part of the constructor as a transaction

Jens: The current facility is brittle because assignment would not be executed as part of a transaction.

Victor: Current functionality is limited

Yuan: Everybody understands that if we do not provide this functionality there is no workaround. Why user would want to have initialization inside a transaction?

Justin: I've written code that uses this kind of functionality. For example, if I do a unique id in an object and a pool of ids is a shared container. So, I personally would lean toward leaving it inside the spec for the 1st round.

Michael Scott: Multiple inheritance is another use case. There is no workaround to execute multiple super constructors inside the same transaction.

Hans: I don't see a good reason to have initialization being atomic.

Jens: Shouldn't you design your module that gives ids so that it uses synchronization itself?

Justin: It depends on the level of composition you want to achieve.

Michael Scott: Imagine that I got an object that is a mini check point. I want a constructor to look at the data pulled from structures from the base classes.

Jens: One can use a workaround ...

Victor: Explaining what Jens says – create a function, put everything inside it and make it atomic.

Jens: ... [elaborates on the prior point]

Tatiana: Mixing modularity and synchronization has its drawbacks.

Hans: Jens solution, although harder to write, is better at expressing the real intent

Justin: Multiple member initialization is still useful.

Hans: But what you are looking for is executing all member initialization as a transaction, not initialization + bodies.

Justin: Arguing for function transaction blocks.

Victor: If one needs to execute initialization atomically and cannot execute the body in a transactional way, the current proposal does not solve the problem.

Justin: That's a valid counter-argument. Is Jens suggesting to send an example illustrating his workaround?

Jens: That's my suggestion.

Michael Scott: I don't agree that it's just a problem of running initializers. But it's esoteric enough that it could be left out of the 1st phase.

Tatiana: It expresses the functionality that one cannot express in another way. Is keeping it in that bad? Taking things out is easy.

Victor: The problem is that the functionality is not what we want. Also, moved by Robert Geva's argument that we need user demand before introducing features.

Yuan: One could put everything in a class to execute constructor as a transaction. ...

Jens: Let's move on after people looked at the example on the mailing list.

Everybody agrees.

Michael Scott: Anybody who wants a feature will be well advised to post a killer application this week.

Action item: Anybody who wants transaction function blocks should post a killer application this week.

2.3 Is memory model wording in Jens write-up acceptable?

Ongoing for the next meeting. Hans and Michael need to continue the discussion about non-normative reading.

Everybody is OK with the normative part.

2.4 Is clause 6 in Jens write-up acceptable?

Jens: People should look at clause 6 at the earliest opportunity

Victor: I looked at it and it looks OK to me on my first looking at it. I prefer elimination of the brackets. We thought that noexcept might be a bad word here.

Discussion on noexcept.

Discussion on using __ as proposed by Hans on the mailing list vs. real keywords. The agreement leans towards using real keywords.

Jens: Will remove __ in the current write up.

Michael Scott: When do we have a discussion on the spelling of keywords?

Jens: When we are happy with everything else.

Summary of new action items:

1. Action item: Victor to update the wiki page to reflect the decision not to include transaction expressions in the phase one.
2. Action item: Anybody who wants transaction function blocks should post a killer application this week.
3. Action item: Hans and Michael Scott to continue working on non-normative part of the memory model

4. Action item: Everybody comment on clause 6
5. Action item: Everybody please read clause 15
6. Action item: Jens to get rid of `__` in `__transaction_atomic` and `__transaction_relaxed`.

Minutes for 2013/06/10 SG5 Conference Call

Minutes by Michael Wong

On Tuesday, June 4, 2013 4:37:59 PM UTC-4, Justin Gottschlich wrote:
Start Time: Monday, June 10, 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)

The current secretary rota list is:

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

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 Wong, Michael Scott, Hans Boehm, Mark Moir, Torvald Riegel, Zihao, Justin, tatiana, Jens joined at 3:40

All the following are approved unanimously unless otherwise noted.

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 Action Item: Michael Wong to talk to IBM lawyer.

Michael just returned from long trip. No connection yet. Will keep trying.

1.4.2 Action item: Victor to update the wiki page to reflect the decision not to include transaction expressions in phase one.

Leave on.

.

1.4.3 Action item: Anybody who wants function transaction blocks should post a killer application this week.

Done. Hans to communicate outcome to Jens

1.4.4 Action item: Hans and Michael Scott to continue working on non-normative part of the memory model.

Done.

1.4.5 Action item: Everybody comment on clause 6

(<http://wiki.edg.com/twiki/pub/Wg21bristol/BasicFirstProposal/tmspec.html>).

Leave on.

1.4.6 Action item: Everybody please read clause 15

(<http://wiki.edg.com/twiki/pub/Wg21bristol/BasicFirstProposal/tmspec.html>).

Leave on.

1.4.7 Action item: Jens to get rid of `__` in `__transaction_atomic` and `__transaction_relaxed`.

Done.

1.4.8 Action Item: Everyone to send Justin and Michael Wong their conflicting dates for SG5 meetings through September 23.

Leave

1.4.9 Action Item: 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.

Done.

2. Main issues (50 min)

2.1 Should function transaction block be part of TS1?

Zihao is convinced by Hans,

2 things we want

1. make the initialization and the evaluation of the members transactional, currently it involves both, Jens work around seems to work is exactly what we want

Justin feels rewrites of base classes is still burdensome, solution so far is not acceptable,

Zihao feels our function transaction block does not do what we want

but Jens' formulation does

Justin feels we do not know exactly what we want

transaction (`a(b++)`)

3 things

1. `b++`

2, `a(b++)`

3. part of function body

MS: will V1 be unusable without it

MM: prefer to wait until user yelling about it

JG: nervous that without this feature, users are offered a big workaround
MS: put the new call explicitly inside the transaction but breaks modularity, but is usable
ZZ: later may need to add syntactic sugar
MW: is anyone else having reservations about leaving this out for V1
TS: I do
MM: I do too, dont think its a killer for round one
HB: work around expresses our intent much better then the language feature
JG: is our language too coarse grain
HB: all the examples only care about the init list, you shouldnt be able to tell whether it is done atomically
MM: if you cant tell, is not the performance prob due to a failure of optimization
MM: can we say the function body is not included in the transaction and we dont believe you when you say you can tell differently
HB: this is different then what we have in the past, this may make sense
MM: 2 reasons why we might want something finer grain ... (i missed the rest)
HB: not convinced yet by examples shown so far, and the refactor is not worst
TS: if it is for implementers, you will need to add it in first TS
TR: depends on cost of adding TM block later on, most impl may view it as not a huge cost, so can be added later
HB: always a cost for adding features
There is a cost for programmers
ZZ: we need a good enough syntax
Technical Reason we are dropping this feature for V1
1. we want something that only handles the evaluation (Justin et all does not entirely agree with that)
2. the language feature we have is not necessarily all that we need, but we don't know what else to add to it without user feedback
Closed as not in V1. tatiana, and Justin agrees.

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

HB: posed an alternative and is acceptable to both MS and HB:
MM: have some questions ...
HB: main thing we want to preclude are weakly ordered atomic operations, in that case you dont get interleaving basis so the statement makes no sense
MM: so this is kind of a conservative statement
TS: any sc execution have this property
MS: can have a irrelevant memory operations appear in the middle
MM: isnt there another ordering that is true unless there is a race in the program
TS: how about any sc execution have equivalent execution
MW: What is equivalent?
MS: claim with out a proof, but if we look at the proof we see it is constructive
HB: does not change when u make those transactions contiguous
does not have to super precise, just try not to say anything that is not technically true
MS: if there is one sc execution, then there is another one with the property we want
TS: imagine different thread interleaving that is non equivalent, tx in T1 that interferes with memory access in T2, ... if such an execution exists, does not tell me anything about atomicity at all

MM: if it is DRF, then there is a sc execution with whatever property we want. .

TS: not good enough, ...

want one that applies to atomics but not relaxed tx

MS: will give it another stab (AI)

JM: joined.

2.3 Is clause 6 acceptable

(<http://wiki.edg.com/twiki/pub/Wg21bristol/BasicFirstProposal/tmspec.html>)?

MS: no objection

JM: no objection last time either

MW: will leave as AI one more week, and then close.

2.4 Is clause 15 acceptable

(<http://wiki.edg.com/twiki/pub/Wg21bristol/BasicFirstProposal/tmspec.html>)?

MS: a small type, one section found to be unclear and suggested an alternative wording, Ask people (AI)

MW: will leave it on AI for one more week and give Jens a chance to address

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

We didn't get started on this. Will defer to next meeting. Some preliminary discussion.

1. Is it okay to omit the `transaction_callable` attribute?

MS: OK to omit,

TS: will defer

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

JM+TS+MM+JS: like to omit

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

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

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?

2.6 Next standardese wording topic?

JM: standardese wording for transactional safety, then see how it works with SBD, work with

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]

No Transaction block support in V1

4.2 Review action items (5 min)

1. Action Item: Michael Wong to talk to IBM lawyer.

Michael just returned from long trip. No connection yet. Will keep trying.

2 Action item: Victor to update the wiki page to reflect the decision not to include transaction expressions in phase one.

3.Action item: Everybody comment on clause 6

(<http://wiki.edg.com/twiki/pub/Wg21bristol/BasicFirstProposal/tmspec.html>).

.

4 Action item: Everybody please read clause 15

(<http://wiki.edg.com/twiki/pub/Wg21bristol/BasicFirstProposal/tmspec.html>).

5. Action Item: 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.

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

5. Closing process

5.1 Establish next agenda

5.2 Future meetings: June 24, teleconference

Mainly look at 2.5 Outstanding questions about Victor's minimalist proposal (see Victor's post on 2013/05/28: "Questions about basic proposal").

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 IWOMP/Innovate) (DONE)

June 10: Discuss function transaction blocks, clause 6, clause 15 (Victor away, Mike Spear 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