

**OASIS Web Services Business Process Execution Language
(WSBPEL) Technical Committee**
Face-to-face meeting
Walldorf, Germany
(hosted by SAP)
31 March – 1 April 2004

March 31

Mar 31 9am-5:30 pm

- Roll call/determine quorum
- Review/accept minutes from Mar. 3 and Mar. 17 meeting.
- Nomination/volunteers to take minutes
- Review agenda.
- Issue discussion.
 - 10 - [Satish Thatte](#) (compensation handling: 25, 75, 83, 94, 108) and 1- [Satish Thatte](#) (scopes: 101)
 - 11 - [Danny van der Rijn](#), 48, 63 - [Yaron Goland](#) (data handling: 12, 51, 90, 102, 103),
 - 4 - [Ivana Trickovic](#)/[Dieter Roller](#) (new constructs: 43, 52, 63)
 - 33 - [Bernd Eckenfels](#) and 37 [Yuzo Fujishima](#) , 78 - [Yaron Goland](#) (correlation: 8, 26, 31, 66, 96)
 - 34 - [Monica Martin](#) (related standards: 15, 71, 86, 87)
 - 107 - [Yaron Goland](#) and 91, 97, 99, - [Dieter Koenig](#) (abstract processes: 82, 109)
 - 3PM issue 53 - [Peter Furniss](#) (process coordination: 30, 54-59)
- - 4:30 pm BoF on ip
- Updates
 - Dinner plans, next f2f – discuss over lunch
 - Revisit issues 10 and 1 with examples of use/abuse – after other issues
 - Vote on issue 25 – when quorum achieved or by [kavi](#) if not possible at f2f
 - Continue data handling: 48, 63 - [Yaron Goland](#) (data handling: 12, 51, 90, 102, 103),

Scribe for first period: Tony Fletcher, Choreology

Roll Call and Introductions

The roll-call was taken and we went round the table introducing ourselves.

Unfortunately the meeting did not have quorum.

It was agreed to move straight to the discussion of issues as the meeting was unable to pass any formal motions including agreeing previous minutes.

Issues discussion

Issue 10 - Serialization of compensation

Satish presented his thoughts on Issues 1 (Permeability of scopes) and 10. Refer to the slides entitled Tailored Scopes (Satish, Dieter roller and Frank Leymann). Slides are available below:

Scopes are not Naturally Homogeneous

- BPEL uses scopes as a hook to hang many capabilities and behaviors that apply to a process region
- One such behavior is reflected in the presence of fault and compensation handlers—let us call this “reversibility”
 - An orthogonal capability is the ability to declare local variables
- The decision to make the reversibility behavior apply to all scopes is a part of the problem
 - Forces all scopes to have implicit fault/compensation handlers
 - Any restrictions relative to links crossing reversible scope boundaries apply to all scopes and thus become hard to justify
- Key part of the proposal is to define explicit attributes to specify the intent of a scope, thus allowing “tailoring” of scope behavior
 - **Not all scopes are reversible**
 - We can invent new attributes such as “permeable” (see below) and define their relationship to other attributes

Proposed Scope Tailoring Attributes

- Currently there are three proposed
 - **Reversible:** has error recovery behavior, *i.e.*, either explicit or implicit fault and compensation
 - **Permeable:** Allows links leaving the scope to communicate status immediately as soon as it is determined at the source activity—impermeable scopes force the status communication to wait until the scope is completed, and the status communicated can be “true” only if the scope completed successfully
 - **Serializable:** as before (just shortened the name from variableAccessSerializable)
 - Need defaults for the first two (mutual dependence must be accounted for, thus makes schema “interesting”)

Mutual Interactions of Attributes

- A control path leaving and reentering an impermeable scope via links would create a control cycle, and therefore this statically detectable condition would be prohibited
- Serializable scopes must be impermeable otherwise they risk deadlock with other serializable scopes
 - But we can eliminate the link crossing ban for serializable scopes
- Serializable scopes must be reversible because otherwise the serializability environment for inner reversible scopes becomes indeterminate
 - see next slide
- Reversible scopes should be impermeable because source activities may be compensated after they communicate the status of links originating in them. Permeability would either “break the premise” of the link status or would require us to address “cascading compensation” (not in this version)
 - Therefore the “effective source” and target of a link must always be in the same reversible scope
 - Therefore the effective source of a link crossing the boundary of an impermeable scope is the impermeable scope as a whole
 - *The status of a link leaving an impermeable scope is communicated only if the impermeable scope is completed (successfully or otherwise)*

Serializability and Reversibility

- There is a side issue left over from the resolution of Issue 79
- The **serializability** environment for reversible scopes nested directly or indirectly inside serializable scopes is uncertain
 - Such nested scopes cannot themselves be serializable
 - When their compensation handlers are called from a fault handler of the serializable scope, they share the **serializability** environment of the ancestor scope
 - When those same handlers are called from the compensation handler of the serializable scope they do not
 - We need to amend the resolution of Issue 79 to make the compensation handler of a serializable scope implicitly serializable
- If we allowed serializable scopes to be non-reversible, there would be no way to make the **serializability** environment of nested reversible scopes predictable
- We therefore end up with a hierarchy of scope properties:
 - serializable > reversible > impermeable

Benefits of “scope tailoring” proposal

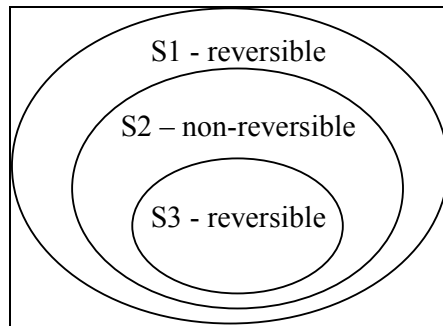
- Solves Issue 1
 - The problem cited had to do with a link firing and then its source activity getting compensated as a result of a fault in the scope at the source—the link would not now be able to fire until all danger of such compensation without similar compensation of the target was past
- Solves Issue 10
 - It is no longer possible to have other than depth first order of control-flow-dictated completion since we ban circular dependencies via links between reversible scopes
- Improves the solution for Issue 79, and moreover
 - Serializable scopes can now be made impermeable rather than banning links crossing their boundaries altogether
 - Makes it much easier to use serializable scopes for concurrency control in <flow> branches

Scopes are containers for activities. The issue is about when and how we should allow information to escape from a scope, especially given that a scope may be reversed later.

Links can make one scope be dependent on another one. If the link fires then there is the promise that work has been done by the first scope (promisor). If this scope is then compensated the promise is effectively retracted but the promisee not told. Therefore cascade the compensation? Suggested answer is No.

An impermeable scope does not allow information to escape from the scope until it cannot be compensated. Currently all scopes are reversible as they have an actual or implicit fault handler/compensation handler.

One suggestion is to put an attribute on the scope to say whether it is reversible or not. Non reversible scopes are transparent.



If scope S1 compensates then compensate scope S3.

Alternatively make all scopes impermeable? Simpler?
Who will use BPEL - sophisticated programmers only?

Yaron (Goland): programmers understand the idea of straightforward compensation readily. Deep nesting and a mix of local and global variables can cause problems however.

Business people design the forward path and business logic, then IT people add in the compensation scheme?

Maciej (Szeffler): however all Scopes cause implementation problems.

Frank (Leymann): business re-engineers tend to want to try to introduce parallelism. Non permeable scopes help to create parallelism. Permeable scopes restrict parallelism.

Martin (Chapman): I support issue 10 proposal (compensation order) but am against issue 1 proposal (permeability). Would like to see all scopes reversible and impermeable.

Yaron: need a use case that is compelling to introduce permeable scopes - do not want to introduce complexity unless really necessary.

Do not have the scope there at all? However this loses the other capabilities of scopes such as the ability to define local variables and correlation set definitions and a group of activities. A scope is associated with wrapping stuff up and hiding it (unlike sequence and flow that you can 'see' through).

Yaron: really need time to think through use cases for this feature, which have not been put on the table yet.

Satish (Thatte) moved on and put up the slide 'benefits of scope tailoring'. The proposal is that only reversible scopes have a compensation handler.

Maciej proposed reserving scopes for the impermeable case, but allowing activities to declare local variables and correlation sets - this suggestion was not accepted at this point.

Scopes do group together activities and set up an environment for them. This could be to add code, acid transactionality, time performance characteristics as well as local variables and correlation sets.

Alex (Yiu) suggested extending the serialisable attribute to differentiate isolation levels.

Diane (Jordan) suggested taking a break from the discussion to come up with examples for and against, and coming back later to look at the examples. Satish suggested that we were agreeing that to solve the issues 1 and 10 we needed impermeable scopes and this proposal was a solution to Issues 1 and 10.

Diana invited everyone to look at the issues list by category. (Issue 10, for instance, is in the compensation category). We should look at other issues in the same category.

Issue 25 - Consistent enablement of compensation handlers

Remove process instance compensation handlers.

There seemed to be consensus on this resolution in the meeting and so it was agreed to vote on this when we achieve quorum (which will now be the next teleconference or via a Kavi ballot as this face-to-face meeting did not achieve a quorum at any point). Could have a pick/event handler at the end of the process instead to receive a compensate or finalise message, and could cover with a time out as well. This approach would then be totally manual with no help from BPEL syntax mechanisms. Yaron has sent a resolution proposal for issue 25 to the list.

Issue 75 - Locally Scoped partnerLink declarations

This issue concerns compensation handlers but is not directly about them. This issue suggests making partnerLinks statically typed but able to have a variable number of instances. Currently need to declare at design time the number of partnerLinks the process has, and this can limit degree of parallelism achieved. Currently need to re-use partner links serially.

Yaron: main motivation for raising issue 75 was to have self-contained segments of code - locally declare partnerLinks.

Satish: there are different interpretations

- it is a visibility thing, or
- it is about exploding the number of instances of partnerLinks.

Fixed number of end points that a process exposes but want to talk to a variable number of instances - asymmetric - but much of BPEL looks symmetric.

ACTION ITEM: Yaron agreed to outline his suggestion and think about the symmetry/asymmetry considerations and the reasons/use cases.

Need to consider system administration of where the process is talking to and the capability to create new addresses to send messages back to it on. Both new instances of existing partners just missing address, and new instances to new partners. No syntax in BPEL for this address. To day can do it serially but issue 75 allows to do it in parallel.

Issue 83 - Garbage Collecting Compensation Handlers

Yaron introduced this issue. May need a lot of state in a compensation handler. Some processes can last a long time (e.g. annuity - can back out of a new contract only for 'x' days after sale). In this case only need the compensation handler for 'x' days. Currently need keep all the compensation handler state for the complete duration not just 'x' days.

Generally there are situations where a compensation handler can no longer be run - can build logic into the handler but this issue is about releasing resources so they can be garbage collected.

Meeting Break: 11:00 AM
Meeting Restarted at 11:14 AM

Scribe for second period: Phil Rossamando, Unisys

Issue 83: Garbage Collecting Compensation Handlers

Yaron: Recap: Kick off discussion (Issue 83). Instead of garbage perform implicit collection.

Satish: Observed that we have the notion of implicit behavior of compensation handlers. He thinks this is a special feature for a special case. If process is going on, compensation handler has a scope.

Suggested to create compensation handler with empty inside.

Yaron: Observed a problem with the above solution. ... Should you be willing to consider side affects of empty.

It was then observed that no reachability analysis is done in BPEL.

Don't actually want to do garbage collection explicitly but rather implicitly.

Yaron: Observed that we need a way to null references to compensation handlers so they can be garbage collected.

Tony: Observed that the BPEL engine may be able to do the garbage collection.

Martin: Believes the programmer should be given control over when garbage collection is performed.

Alex: Need Boolean condition to decide when garbage collection to be performed.

Yaron: Has sympathy for Alex's ideas. Directly specify reason why compensation handler should be garbage collected within the compensation handler should be garbage collected.

ACTION ITEM: Yaron agreed to engage with Alex to further discuss this issue (i.e., Issue 83).

Issue 83 Discussion Continues:

Ben: Different scopes should be able to say I'm not interested in this compensation handler any more.

Alan: Doesn't like to have indication in compensation handler.

Dieter: What we're discussing here is not just for compensation handlers, (e.g., fault handlers, event handlers, etc.) are also potentially involved. This may become a major effort.

Peter: Thinks he agrees with Dieter. Doesn't want to play games.

Yaron: Strongly disagrees with Dieter concerning similarity of Compensation handlers and Fault handlers.

Specifically, Yaron observed that "Fault handlers created with scope and die with the scope." Compensation handlers have no associated scope. (Has infinite nature)

Uso: What ever mechanism we have for garbage collection it must be the same for all compensation handlers.

Yaron: One scope can have one compensation handler.

Danny: Want to make things run more efficiently. May not want to include in spec. Relegate to implementation specific extension.

Yaron: Doesn't care but will tie customer to specific engine. Agrees with Danny maybe we don't care.

Yaron: Should we go on to next topic.

Dianne: Suggests we defer until later but consider Danny's above point further.

Yaron: Observed that the merits of proposal outweigh cancellation.

ACTION ITEM: Yaron will investigate this issue further.

Yaron suggests we go on to the Issue 108: Parallel Compensation.

Diane: Every issue must be decided. Don't prioritize issue but just go into them.

Issue 82: Description of abstract processes in spec

Abstract process issue 82 mentioned but No further discussion.

Issue 94: Allow both "compensate" and other activities in compensation or fault handler

Issue 94: Bug so fixes it. Alex to get proposal to vote on issue 94.

Issue 108: Parallel Compensation

Issue 108: (Yaron introduced) Parallel Compensation. If I have an implicit compensation handler, should it be serial?

There is a problem here in that two different versions of BPEL could handle default compensation handlers differently.

Current spec does not provide enough guidance to handle the situation properly.

Make handling of default compensation handlers explicit.

Need clarification.

Sadish: We did not set the BAR properly. Mandate serial behavior for compensation handlers. Default compensation handles should be serial.

Peter: Compensation handlers should be under programmer control.

Danny: Engine could decide if things were run in parallel. If we want to enable parallel compensation handlers than it should be an explicit attribute of the handler.

Sadish: Interesting scopes that run in parallel but can be serializable.

Tony: Parallel scope serialized running them backwards should fix the problem.

Dieter: If we do things serially, it takes longer. Engine should decide if parallel or serial execution should be used.

Yaron: Feels default serializable may be appropriate.

Phil: Feels that serializable is preferred default mode for engine and flag should exist which says that compensation can be done in parallel.

Sadish: Doesn't think explicit rules for parallelism can be done the spec.

Yaron: Feels default behavior should be serial and explicit action to be taken otherwise.

Frank: If two scopes start in arbitrary order than ...

Danny: Given two programmers they will each do it wrong.

Peter: If you have compensation handlers which can run in any order that will be broke if not run in proper order, they will be broke.

Alan: Implicit compensation handler. Can go for conservative semantics for default compensation handler.

Frank: No way to explicitly say that all compensation handlers should be run in parallel.

Yaron: If I wasn't using the default compensation handler than we could use flow trick, add flag to indicate compensation handler be run in parallel.

Sadish: Thinks Dieter's approach is worth considering. Need to think very carefully relative to issue 10 (i.e., **Issue 10: Serialization of compensation.**)

Danny: Even if you could determine invocation order of compensation handlers, you still wouldn't have a record of start order.

Yaron: If I change variables should I worry about others fooling around with them? Compensation handlers should by default be serializable. The only reason for not doing this would be performance based.

ACTION ITEM: Dieter: Action item to sort out the proposed solutions and come up with a recommendation.

Scopes:

Issue 101: Local Variables Overriding Enclosing Scope.

Sadish: Proposes the opposite to Danny's proposed solution. If we change scopes than variables may have different values.

Yaron: When you interpose a variable within a new scope you have performed a violent action. Adopt solution as proposed by Danny.

Jeff: What is bug?

Yaron: No methods in BEPEL. Need cut and paste reusability.

ACTION ITEM: Phil will post suggestions for using scope resolution operators.

Frank: feels scope resolution operators are inappropriate but didn't say why exactly.

Danny: Cut and paste does not to work and so don't know what resolution is for reuse. Lexical scoping not an end-all.

ACTION ITEM: Danny volunteered to send out proposal on this issue.

Lunch started at 1:30PM.

Lunch ended at 2:30PM.

Discussion on Issue 53

PeterF: See presentation mailed out by Peter on March 30

Process coordination

The presentation was issued before the tentative resolution Issue 25 was received

Y Axis on slide 4 of presentation represents increasing visibility of effect

We need to decide what we are going to call document we are working on.

Slide 7 Top Left equates to Issue 25 Resolution as proposal for current doc
Top right is proposal for round 2 but not for now (not questioning whether there will be a round 2).

Jeff: drop for now and hope something will happen in future

Slide 9 – Starts in some known state

Slide 10 – perfect compensation

Round 2 - Reversible AND contingent

MonicaM – can we talk about function rather than external specification names – especially ones that may or may not be in standards bodies

The conclusion doesn't mention specification. Round 2 discussions will depend on what state of other specifications is at that point in time.

Monica – worthwhile including discussion of other specs.

AlastairG – BPEL spec being prod should remove refs to spec BT spec e.g. WS-BA – inappropriate because it pre-judges work being performed elsewhere

At least 3 other specifications including CAF & BA – stand best chance of being vanguard spec – hope that work taking place in all others is taken into account – recognise points of others – premature – too unclear too fuzzy at moment.

MartinC – broadly agree with Choreology – that this issue should be deferred till later – implications that remove trans and coordination (thought had done at last meeting including removal of appendix C and voted on it –but couldn't find minutes to check). Latest editors' draft is still dated Dec 03 and still has these references to other WS-specifications.

MartinC - can we inspect the minutes from Melbourne meeting to check?

DianeJ – thinks there was lots of discussion at Melbourne meeting but was not put to the vote

DianeJ – suggested that Choreology proposal is put to formal vote

TonyF Proposes for issues close 30, 53 – 59 with no change to spec but marked as revisitable after this round and that Appendix C is removed

Seconded by MartinC

Not quorum so Straw poll instead as indication to rest of TC what the feeling of F2F was

SatishT – state machine in appendix C (Figure 2) helpful in compensation and fault handling so may consider keeping that in some form

MonicaM – will get on to this in issue 34 – should look at BPEL's requirements in this area – may be should revisit appendix C rather than just remove

TonyF – would like to vote on motion as proposed – so that can indicate whether we have lost valuable information – should raise new issue of what should be added to spec. May be would use appendix C as initial text and some parts could go into main body of spec as appropriate

PeterF – includes details of BA spec but this has changed so BPEL spec is not correct. BPEL spec would need to have this diagram reworked as some of arcs need to be renamed

SatishT – rework appendix C

AlastairG – am sympathetic to state diagram that describes what happens in respect of compensation handling. Think that appendix C so heavily marked with external refs that it would have to be completely reworked. Agrees with what TonyF suggest – remove appendix C and include appropriate explanatory material. Don't want it to become a halfway house or amendment. Would think that since not quorate straw poll should

MartinC – suggest that delete appendix C and that Satish should propose new solution.

AlastairG – formal point would like to with draw motion made in January.

DianeJ – can't because not quorate – don't think ever formally ever tabled so probably doesn't really need to be revoked

Straw Poll – see issues list for formal

MartinC - should state transaction and Coordination protocols

Phil seconded

Result = 27 in favour, 1 against

Issue 11

Formal ballot rather than TC to make sure of result rather than relying on TC conf call that may or may not be quorate

DianeJ – what will we be voting on for issue 11

MartinC – vote on what was in minutes – will vote on MC's proposal

AlastairG – what's happening with issue 25?

DianeJ – straw poll already and will be formally dealt with in next TC call

DianeJ still lots of issues to go through after sub committee reports

End of March 31 Meeting.

Note:

A BOF was held on 3/31 to allow people to discuss licensing concerns. Minutes from the BOF are not provided since the BOF was not considered to be part of the official TC meeting.

Members of the BOF eventually returned to a discussion of the BPEL Issues list. Although the Issue list discussion occurred during the time allotted for the BOF, minutes of the discussion were captured since it concerns the business of the TC.

Issues Discussion Again

Look at other data handling issues 48 & 63?

Issue 12

SatishT – Kevin raised this issue initially. However, between 1.0 and 1.1 added XML types. Since WSDL doesn't allow to do this WS will be using a message. It is effectively a shortcut. Goes some way to eliminating assignment.

DianeJ – defer since Kevin not primed to provide info in this

Issue 48

Should close this and raise another issue

Lots of discussion

YaronG – should close 48 because 13 solves but doesn't solve efficiency

AlexY – open new issue to solve of examine and manipulate issue. Can use input from issue 11 as an example. Not trying to define language binding, need to define placeholder. Want to ensure schema is validated after manipulation.

Straw poll

**Yaron prop, Maciej seconded
16 in favour, 2 nos and 2 abstentions.**

Issue 51

YaronG – there are 2 issue in 51

- 1) Text in 9.3.1 doesn't seem to allow copying between message variable and non message variable
- 2) For an assign to be legal schema of source must be equal to or a sub-set of the schema of the destination

Intention for static analysis can tell whether legal or not.

Use case concerned about. Temp variable to build up XML structure assign of type any and variable of type not any.

Understands that has to be sub-set of destination.

Want to be able to do in 2 steps

- 1) static analysis to work out whether could succeed still possibility of run time error
- 2) schema validation at runtime expensive. Validation by promise!

Issue 63

AlexY – Xquery or XPath has for each. Array too restrictive

Danny – if want to receive multiple times into subsequent slots. Relates to Issue for and 11.

SatishT – if going to construct number of messages (each may be multipart). 4 based on good use case – should follow up on this.

YaronG – Do we want to close this in favour of 4 i.e. Park 63 until discussed 4?

Issue 90

DianeJ will discuss with Kris to forward a proposal

Issue 100

Danny – contingent on 11 being solved

MartinC – do need to work out where the schema validation points are

Danny – schema should be valid all the time until this occurs don't need to resolve

Alex – will work with Danny on 11, 48, 100 on joint resolution

Issues 102, 103

Bug fix – need proposal to fix

AlexY - Want to make pseudo schema forward compatible

SatishT – suggest wait for Alex's proposal before revisiting this
AlexY/YaronG - no -103 is independent

Maciej – dealing with implicit schema – can use in other places within spec. If want to iterate over collection of messages.
AI Alex will write up a proposal

DianeJ will summarize action items

Meeting Adjourned at 18:00
Meeting Starts tomorrow at 09:00

April 1

Apr. 1, 9am - 2:30pm

- Liaison subgroup report
- Use case subgroup report
- Requirements and Issues Coordination subgroup report.
- Specification Editing subgroup report.
- Implementation subgroup report.
- New business
- Continue issue discussion from above list
 - 4 - Ivana Trickovic/Dieter Roller (new constructs: 43, 52, 63)
 - 33 - Bernd Eckenfels and 37 Yuzo Fujishima , 78 - Yaron Goland (correlation: 8, 26, 31, 66, 96)
 - 34 - Monica Martin (related standards: 15, 71, 86, 87)
 - 107 - Yaron Goland and 91, 97, 99, - Dieter Koenig (abstract processes: 82, 109)
 - Revisit issues 10 and 1 with examples of use/abuse – after other issues
 - Vote on issue 25 – when quorum achieved or by kavi if not possible at f2f

Scribe: Phil Rossomando

Ivana: Presented Report Bundles Requirements presentation on issue 4.



Microsoft PowerPoint
Presentation

(Ivana's presentation is also available from the Calendar Documents folder)

Issue 4: Dynamic parallel processing

Presentation/Proposal Posted

Dieter Roller and Ivana Trickovic put together the presentation

Invocation

Multiple activity instances for same invocation type activity

Different types of invocation mechanisms.

<invoke> and example of synchronous invocation

<invoke> - <receive> activities for asynchronous activities.

Send same request to different partners

Send different request to a partner.

Etc. see Ivana's foils.

Yaron: Observed that Ivana is attempting to recreate the or-join.

Yaron: Is concerned about the proposal and made many comments to this fact.

What is an or-join asks Dieter: N of M joins said Yaron.

Compensation

WCL message may have different parts.

Yaron: Extremely concerned about Array representations and suggest that XML schema be the focus. (see Scenario 1 foil)

XPath expressions.

Sadish: Raised his own set of issues.

Related Issue 103: Standardizing \$varName syntax for XPath to refer to a BPEL variable

Yaron: Sees a multi-partner link problem.

Alex: Says we don't want to invent a new XPath concept.

The question is: How to design bundle elements. Says Ivana.

She suggested the introduced of a new element: <bundle>

An activity can be a scope.

Yaron: Raised another serious concern about how partners are supposed to work in BPEL.

Says Yaron: Two partner links required to talk to two different people.

Spec not clear on the meaning of partner links says Yaron based on the differing understandings between him and Ivana. This causes him serious concerns.

Solution suggested locally scoped partner links. Yaron agrees.

Diane: Takes on-line attendance.

Ivana's presentation continues:

Completion conditions:

Peter: Raised concerns about using macros to capture completion conditions.

Yaron: Countered with reason why a macro may be useful. Must decide if a macro is worth creating.

Yaron: Again raised the issue of slippery slope.

Peter: How does application decide if decision is appropriate? How is it communicated once it is decided that it is correct?

Error handling within a Bundle: How is it to be done?

Bernd: Inside activity within bundle can have fault handler says Yaron and Bernd.

Bernd: Skip branch elements if you want to ignore it catch the fault within the bundle activity itself.

Yaron: Completion conditions of bundle spec is the general concern raised by group. Says burden of proof that there is no problem is with Ivana.

Sid (On Phone): Disagrees with Yaron's concerns and way of presenting them.

Mashie: Sees what Ivana is driving at. His main concern is the <ignore> element and the branch/condition activity.

Peter: Distinguishing how application decides what to do and how BPEL engine address this how.

Alex: Bundle element should ignore "ignore all" for a while.

Alex: Should always have a scope element within a bundle fault handler would be implicitly generated. Ignore all is confusing.

Danny: Instead of a branch attribute use an XPath attribute.

Yaron: Concerned about using faults to handle success condition. Scale of changes required is making him concerned.

Phil: Agrees with Yaron that Fault should not indicate a successful completion of a Bundle.

Sid: feels we should further consider Ivana's proposal. Need more time to analyze the proposed solution.

Mashie: What about the case where you have a series of dynamic invokes, may have problems with correlations.

Bernd: Confused about implicit way of getting same index. Should think of something else. Say an explicit index is needed.

Danny: Satisfies question you raised addressed through use of different partner links.

Sadish: Agrees with Danny. Partner link implicitly replicated. Can't tell from whom the message came from after you receive it. Question identical messages to different partner.

Danny: If you're sending to different end points you can correlate on end points.

Mashie: Disagrees

Peter: Iterating over response messages. Need to enhance interrater definition. Is it possible to have a bundle with only a receive? Or a receive followed by an invoke.

Dieter: There must be something that distinguishes message, they cannot be "identical".

Alex: Invoke and receive. Can handle Invoke part with correlation set. Receive part of message must be different.

Yaron: Group design not the optimal way of solving problem.

Mashie: What we are discussing has impact on partnerLink addressing.

Ivana: Do we want to use macros or use a first class element? This is an initial proposal. Feels we have valid scenarios. Include bundle element as a feature. Data handling is an issue. Will it appropriately handle WCL messages.

Sadish: Much rather see that we have a workable proposal than chase a too difficult solution.

Diane: Group needs to agree that the scenario is valid.

Yaron: Number of use cases here that can be covered. No vote now. Need consensus identified within a sub-group.

Diane: Need to identify impact of individual use cases.

Issue 33 Correlation Bernd Eckenfels (related: 8 correlation sets immutable, 26 receive/reply, 31 auto correlation key,

66 no/multiple match, 78 multi-start initiate and 96 opaque correlation) Initial discussion surrounded race conditions. I need to understand the scope and use of correlation.

- a. Find correct instance for incoming message.
- b. Check outgoing messages.
- c. Decide on Initiation of new instance.

Samples are confusing - examples do not work. Affects other related issues (see above) and a new issue: Does the correlation set place the value in the variable?

For ship order example: set property

Why do we have check on correlation key on outgoing messages?

Thatte: Need for initiate.

Question surrounding on outbound that is not initiate. Correlation is a 'conversation' and a key for an abstract conversation between multiple parties. Assert if part of same conversation. This establishes the contract.

Szefler: Those who have read the specification, the examples and descriptions are not obvious to Satish's definition.

Chapman: Multiple outstanding receive-reply is restricted.

Thatte: This relates to Issue 26.

vanderRijn: From bundle discussion, do we need ability to initiate correlation sets from other than messages?

Thatte: This may relate to Issue 96.

vanderRijn: What about <assign> into a correlation set?

Yiu: There are two level of opaque-correlation set definition and then property level (96). Need the concept of assertion.

Goland/Chapman: If the related issues are addressed and descriptions in the technical specification may alleviate Issue 33.

Note: The team didn't agree to this.

Chapman: Need to take the issues together to define what correlation should do.

Goland: There is no motivation to do this in spec editing team.

Eckenfels: Reference callback example.

Using sequence, flow etc., the engine cannot be conformant to the specification. The specification does not specify I have to accept messages I am not waiting for. A strict definition at the protocol level in the engine could reject a message. Need to make sure the engine is listening. Show dependency of sequence between <receive> and <invoke> in a flow. Need to state race conditions.

Goland: Are we implicitly or explicitly saying we expect queues? How do you communicate to the engine my intention?

Szefler: Why do we react thus to queues - this is a normal expectation for the implementor?

Eckenfels: This is a design issue. The more activities I have, the more chance I miss the sequence required.

Szefler: This is a matter of degree.

Eckenfels: Specification does not allow me to have two activities together in sequence that come one after the other (not defined constraint).

Thatte: Agree with Szefler. The receive activity is a trigger a broadcast of my intent to receive a message. The forward looking is implicit.

Szefler: Assume messaging is reliable.

Eckenfels: Use correlation log to understand candidates to incoming messages. We should put clarification in the specification. Need non-normative description and in general make more explicit what is normative or non-normative in the specification.

vanderRijn: Are you saying that While there is the appearance of a race condition, implementations deal with it?

Goland: This is a problem is not usefully solved. BPEL is transport and implementation agnostic.

Eckenfels: Provide a rationale in the specification, however. See: <http://www.oasis-open.org/archives/wsbpel/200403/msg00240.html>.

Would prefer to provide updated verbiage and engage on Kavi after straw poll.

Jordan: Do straw poll today and address at next meeting.

Eckenfels: There are multiple options: 1. Original proposal 2. More formal specification 3. No action

Jordan: Resolve with clarification in specification seem preferred.

Eckenfels: This is the motion.

"The implementation should be able to deal with responses, which are received before the actual receive is activated.

This is required to reduce the risk of race conditions, between requesting a service and receiving its response."

Yiu: In the specification, a default container received a message but cannot find a correlation set. What should the container do? This should be specified. Does this occur in Issue 33?

Eckenfels: BPEL generic fault may be confusing especially if BPEL is not known by a partner. Need to open a new issue.

Furniss: Related to Issue 66? This condition involves an undeliverable message.

Thatte: Where does the BPEL engine send back the faults to? We don't know bindings.

Yiu: Agrees his issue is handled in Issue 66.

Straw Poll

Vote: 22 Yes, unnamed party abstains as they are muted by the operator.

Issue 37 Yuzo Fujishima See: <http://www.oasis-open.org/archives/wsbpel/200403/msg00207.html>

Case 1: Non-start activities

Three proposals (see brief).

Proposal 1-B is more aligned for multi-start activities.

Case 2: For start activities

This is an area of confusion.

Disallow both initiate="yes" and "no."

Allow a new value "start."

Goland: Can I have an initiate that is not a start? Must all initial activities be start activities? Specification is ambiguous here.

Fujishima: The "yes" has two different meanings depending on whether the activity is start or non-start.

Eckenfels: We have an option not to allow multi-start activities but use a <pick>.

Roller: That will not work; there are different semantics.

Eckenfels: You can have a receive inside a pick.

Fujishima: Proposal is to merge 1-a (Always throw correlation violation fault) and 2-a (see above).

Thatte: Pushing back on handling multi-start here.

Goland: This relates to Issue 96.

Fujishima: Agree with Goland to not address multi-start until we talk about dedicated start activity.

Trikovic: This only has to do with abstract processes.

Fujishima: For executable process, many faults can be thrown. The behavior is largely undefined for abstract process. Proposal is for executable processes only.

**Lunch started at 12:30PM.
Meeting restarted at 1:30PM.**

Topic: Next F2F

Every three months the spec is not done we need to do another F2F.

September is five months away so June/July.

Potential June 21 thru 25.

Diane asks for hosting – food, room, internet access (preferable).

ACTION ITEM: Diane will put a ballot out – preference for west coast but will depend on hosting offers.

Topic: Issue 11: Query in <to> close should allow assigning to new locations

Danny van der Rijn (re)stated the proposal for extra elements in the assign – insertbefore, insertafter, append, remove, rename. We have previously decided by straw poll that this is worth pursuing, so the discussion focuses on how it should be done.

Alex had some suggestions to make the proposal more DOM friendly vs xupdate centric.

Danny may be happy to go for DOM friendliness.

Ivana, will this help amalgamating multiple messages. Probably not necessarily as wsdl messages are multi part.

Related to issues 103 and \$variables.

Danny: Do we do this or not

Danny motion to adopt proposed solution sent in email issue 11: preparing for discussion fri march 25. 3:33pm.

Alex seconded.

Procedural discussion whether people have had time to absorb the proposal.

Satish: simple “objections”. Assignment in current BPEL is simple, limited and relative fragile because BPEL was not intended to solve full data handling issues. Other proposals would need adding to complete the story Danny started and is a potential slippery slope. So do we take on this slippery slope or do we actually cut back on exiting features and defer all data handling to other specs!

Yaron: Customer interest in BPEL is protection w.r.t the potential to swap vendors. So what is the longevity in this space?

Xupdate may provide lots of answers but not yet a standard. Plus agrees with Satish about slippery slope. So why promote this incomplete and soon to be outdated proposal. So take assign out of executable and allow a watered down form in abstract.

Bernd: facilities must be in the spec for portability reasons. But agrees with Yaron that proposed solution may not be enough and may need an external spec that should be referenced by bpel.

Maciej: people want to do simple stuff and need the facilities to do simple stuff. Otherwise they get frustrated. So either adopt this or take Satish's proposal to remove.

Alex: how long do we have to wait for xupdate! So until then no portable simple xml manipulation for bpel. Others non-schema aware languages have the same problem. Without this executable bpel is not portable.

Ivana: Do not agree to exclude assign. It is less important for abstract.

Yaron: this proposal still doesn't let people do simple stuff e.g ability to deal with intermediate states that are not schema consistent. So kidding that this is a simple solution as to be useful needs to solve some real hard problems. On BPEL being non-portable, it was never intended to be a complete solution. Never intended to solve the xml data manipulation in a portable way.

Danny: If bpel is just programming in the large, just remove assign. Assign the way it is though is not defensible to needs to be resolved or removed. If xupdate is going to take a while then its not instant legacy.

Jeff: Need to make the tradeoff of functionality vs complexity. Current discussion not helping with these tradeoffs.

Martin: said needs at least a way to wrap a transformation in a portable way.

Bernd: Different models for different people with different tools and editors. BPEL should be able to have complex structure manipulations and BPEL tools should be supported.

Alex: who will be inserting the java – probably not the bpel person.

BPEL should have some manipulation facilities. Maybe related to issue 108. Schema validation boundaries need at least defining.

Satish: Agree to Alex about schema validation boundaries. Manipulating xml was not in scope of charter and is also a big slippery slope. The fact that other groups are taking so long indicates it's a difficult problem. So might prefer Martins proposal.

Yaron: issue 48 is related (Xml transformation support). Rather than 11 that introduces new operations, look at how issue 48 could work with at least one binding such as XSLT.

John: We are not trying to define a programming language, BPEL is just about exchange formats.

Straw Poll:

9 yes, 10 no, 6 abstain.

Satish needs alternatives to choose from.

Alternate 1: remove the assign completely

Tony: won't be able to do for loops without assign.

Cannot remove assign cause of partnerlink assignment

Satish: too crude a way to state this so need to think of proper wording. So this removal wont really work

Another alternate: leave as is.

Straw Poll:

How many can live with assign as is currently defined?

Yes =17

Staw Poll:

How many people are unhappy with how assign is today?

Yes= 8

Another alternate is Martin's encapsulation.

ACTION ITEM: Martin to write up xml transformation/assignment replacement.