

Document Scope

This document represents the DRAFT Resolution Log of the six SCA Charters submitted for OASIS Member Review on June 29 and July 02, 2007. It is submitted prior to the combined Convener call for the six charters which takes place on July 17th. A final version of this document will be posted as the charters are finalized, ready for the Call for Participants.

Issues Raised

Issue 1: Why have a TC for every language?

Raised by: David RR Webber.

Original Text: Why have a TC for every language?

Is it just me - but I thought the whole point of SCA was that there is a common consistent approach - that is language agnostic - and that you should not need a TC for every language out there implementing SCA...

Rather - from the base SCA work - you should be able to develop simple Appendices - for BPEL, Java, C++, VB, Delphi, Ruby, SQL et al as non-normative guides.

Am I missing something here? Or is this really just a case of everyone deciding they have to own their own little patch of the turf?!?

Disposition: No changes to any of the six proposed SCA charters are required.

Response:

You are right that one of the main goals of SCA is to standardize a common and consistent approach across heterogeneous technologies/platforms for developing and deploying service oriented applications. To that end, SCA does indeed define a common model as well as extensive set of language-neutral syntax that address various different functionalities such as for – building composites, expressing bindings to communication technologies, declaring non functional/QoS requirements, etc.

In addition, to achieve portability and to simplify the development of SCA compliant components using the various, popular languages/platforms, SCA defines mappings of the SCA concepts and the language-neutral syntax to the native model/syntax of individual language/platforms. In some cases, the applicable mappings may be relatively simple and short. However, in other cases, especially where the development platforms are complex and rich in functionality (e.g. Java EE, Spring), mapping of SCA to the native concepts and syntax involves significant amount of work.

Please note that, we considered it highly important that the SCA mappings to languages/platforms be clear, complete and normative (as opposed to your suggestion of non-normative guidelines) to ensure that the adoption of SCA in

corresponding markets becomes easier, and more importantly creating portable composites remains a strong possibility.

In summary, our goal of standardizing comprehensive support for developing SCA in different language/platform environments in a manner that is native and simpler for the corresponding developer community has led to creation of the various different SCA language binding specifications. As you may notice by reading the input contribution material on the C&I specifications of Java, Spring, BPEL, etc, the content of each of these specifications is sufficiently large and unique to justify their existence as standalone documents, and creation of separate TCs to track their further evolution.

Note: A full thread of the issue being debated can be found at: <[Mail thread](#)>

Impacted Charters: None.

Issue 2: Inconsistent terminology regarding TC life cycle?

Raised by: Sanjay Patil.

Original Text: Inconsistent terminology regarding TC life cycle?

The charters for the SCA-C and the SCA-J have the following statement under the Deliverables section:

- Ratification of the above specifications as OASIS Standards, including a brief period to address any errata, will mark the end of the TC's lifecycle.

I think the above statement conflicts with what is said under the Maintenance sub section of the Deliverables section in the same charters:

- Once the TC has completed work on a deliverable and it has become an OASIS standard, the TC will enter "maintenance mode" for the deliverable. ...

Disposition: Issue Accepted. This will be corrected in the final submission.

Response:

Issue agreed. The text "*Ratification of the above specifications as OASIS Standards, including a brief period to address any errata, will mark the end of the TC's lifecycle*" will be removed from the two charters.

It is the intention of the charters to ensure that the life cycle of each TC must include a period of maintenance until there is confidence that the deliverables are stable.

Impacted Charters: SCA-C and SCA-J.

Issue 3: Incorrect Proposed Working Title?

Raised by: Robin Cover.

Original Text: Incorrect Proposed Working Title?

=====

i. Proposed working title
Service Component Architecture Assembly Specification

=====

I wonder whether the "Proposed working title" given in the SCA-C draft charter is correct, or (?) perhaps an error. The title "Service Component Architecture Assembly Specification" matches the title of the specification named in the Proposed Charter for the SCA-Assembly TC, presented here:
<http://lists.oasis-open.org/archives/tc-announce/200706/msg00014.html>

Here's the list of "Proposed working title"s for the six TC charters (as numbered by their release)

- 1: Service Component Architecture Assembly Specification
- 2: Service Component Architecture Policy Framework Specification
- 3: Service Component Architecture Web Service Binding Specification
Service Component Architecture Java Message System (JMS) Binding Specification
- Service Component Architecture J2EE Connector Architecture (JCA) Binding Specification
- 4: SCA Java Component Implementation Specification
SCA Java Common Annotations and APIs Specification
SCA Spring Component Implementation Specification
SCA EJB Session Bean Binding Specification
SCA Java EE Integration Specification
- 5: Service Component Architecture BPEL Client and Implementation Specification
- 6: Service Component Architecture Assembly Specification

I note also that the "Deliverables" section for the SCA-C proposal identifies four separate "specifications":

=====

Deliverables

The TC has the following set of deliverables:

- A revised SCA Client and Implementation Model Specification for C++, together with a complete set of conformance statements
- A revised SCA Client and Implementation Model Specification for C, together with a complete set of conformance statements
- A complete Test Suite specification for the SCA C++ Specification including documents and the related materials described in the scope section
- A complete Test Suite specification for the SCA C Specification including documents and the related materials described in the scope section

These specifications will reflect refinements, corrections or material technological improvements with respect to the input documents and in accordance with this charter. Ratification of the above specifications as OASIS Standards, including a brief period to address any errata, will mark the end of the TC's lifecycle. The TC may choose to vary the number of the specification documents and their titles.

Robin Cover
OASIS, Chief Information Architect

Disposition: Issue Accepted. This will be corrected in the final submission.

Response: Issue accepted; you have found the obvious copy&paste error that we should have spotted!
Section 2i will be corrected to list the following:-

- SCA Client & Implementation Specification for C
- SCA Client & Implementation Specification for C++

Impacted Charters: SCA-C-C++

Issue 4: Words imply a solution to eventing & conflates eventing with sync-async

Raised by: Martin Chapman.

Original Text: Words imply a solution to eventing & conflates eventing with sync-async

The proposed SCA Assembly TC charter as currently worded implies a solution to eventing, and also conflates eventing with sync/async invocations.

Bullet 11 of the in-scope section says:

- 11. The handling of service interfaces, including the nature of the message exchange patterns and the handling of synchronous, asynchronous and one-way interactions. Techniques including Pub/Sub and Queue handling form part of this description.

I think it would be better to remove the "Techniques..." sentence and to add a new bullet at the end of the in-scope list

- Support for eventing, pub/sub and queuing systems within the assembly model.

Disposition: Issue Accepted. This will be corrected in the final submission.

Response: Proposed change to the SCA Assembly TC charter as follows:

Change the bullets at the end of the in-scope section to read as follows:

- 11. The handling of service interfaces including the nature of the message exchange patterns and the handling of synchronous, asynchronous and one-way interactions.
- 12. Support for Eventing, Publish/Subscribe and Queueing models

Impacted Charters: SCA-Assembly.

Issue 5: Use of the Java Trademark

Raised by: Robin Cover.

Original Text: Use of the Java Trademark

A question has arisen about the Proposed Charter for the SCA J TC [1]; it's been requested that I seek clarification on this list.

The Name of the proposed TC as given in the "PROPOSED CHARTER FOR REVIEW AND COMMENT" (1. Normative information, a) is:
OASIS Service Component Architecture for Java(tm) (SCA J) Technical Committee

The Charter text also states (following the References, and apparently at the end of Section "2. Non-normative information regarding the startup of the TC") that:

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Now... The OASIS Technical Committee Process [2] specifies in "Section 2.2. TC Formation" that the name of a TC designated in a Charter may not "include any trademarks or service marks not owned by OASIS."

(1)(a) The name of the TC, such name not to have been previously used for an OASIS TC and not to include any trademarks or service marks not owned by OASIS. The proposed TC name is subject to TC Administrator approval and may not include any misleading or inappropriate names. The proposed name must specify any acronyms or abbreviations of the name that shall be used to refer to the TC.

Also... according to the IPR Policy (Section 5.3 Trademarks) [3] Trademarks or service marks that are not owned by OASIS shall not be used by OASIS, except as approved by the OASIS Board of Directors, to refer to work conducted at OASIS, including the use in the name of an OASIS TC, an OASIS Committee Specification or OASIS Standard, or incorporated into such work.

Questions:

1. Does OASIS have ownership of the Java trademark?
2. If so, by what means has ownership of the Java trademark been conveyed to OASIS or otherwise established?

3. If not, has the OASIS Board of Directors granted an exception according to the rules of IPR Policy Section 5.3?
4. Independent of the answers to questions in #1-#3 above (and assuming that the legal requirements for ownership were to be met according to the rules of TC Process Section 2.2), is deemed necessary that the **official name** of the TC be spelled using the symbol "(tm)" [Java(tm)] – or (e.g.,) would it be sufficient to arrange for use of the symbol "(tm)" just in the Charter Section 1.b "Statement of Purpose"?
Thanks for clarifications.
Robin Cover

===== References:

[1] <http://lists.oasis-open.org/archives/tc-announce/200707/msg00003.html>³

Proposed Charter for OASIS Service Component Architecture for Java(tm) (SCA J) Technical Committee - 4 of 6

[2] <http://www.oasis-open.org/committees/process.php#2.2>³

[3] 5.3 Trademarks

With respect to CONTRIBUTIONS

<http://www.oasis-open.org/who/intellectualproperty.php#contributions>

Disposition: Issue Accepted. This will be corrected in the final submission.

Response:

Thanks for catching this. I think the answer is actually quite simple and avoids the various pitfalls which you so ably describe. We had meant to remove the Java(tm) reference in the title. I guess you can never have too many eyes reviewing proposed text.

I believe that the issue will be resolved simply by appropriate renaming. To be consistent we will change the name of all the language specific TC's to be something like: Service Component Architecture/<lang> (SCA/<lang>) Technical Committee e.g. SCA/J, SCA/BPEL, SCA/C-C++.

Descriptive material, specifications, etc. can refer to trademarks (kind of like fair use), as long as it is appropriately attributed. A search through lots of OASIS documents will turn up lots of references to trademarks, although many don't have the "foo is a trademark of bar" (an oversight i presume). Hence I don't think we need to do anything more than fix the long and short names.

It might be worth raising an issue (with TC Process committee I guess) to look into whether clarifying material needs to be added to the Process document, staff procedures, etc. to require that trademarks have appropriate attribution.

Please let me know if you don't think this will resolve the issue.

Impacted Charters: SCA-J

Issue 6: Vacation Conflicts with Initial TC Call

Raised by: Pete Wenzel.

Original Text: Vacation Conflicts with Initial TC Call

As Sept 3 is a major US holiday this year (Labor Day), the organizers should expect minimal participation from US-based parties on that date. Please consider rescheduling for later that week, to give more individuals an opportunity to join the TC at the very important initial meeting.

Disposition: Issue Accepted. This will be corrected in the final submission.

Response:

Thank you; this was an oversight that we had missed. Before we finalize all of the SCA Charters, we'll coordinate a schedule for the initial TC teleconferences, with a target of avoiding that Monday (and maybe Tuesday).

Impacted Charters: Potentially all six charters.

Issue 7: Pointer to OASIS Solution Deployment Descriptor TC recommended

Raised by: Brent Miller.

Original Text: Pointer to OASIS Solution Deployment Descriptor TC recommended

Regarding the proposed SCA-Assembly TC charter item #14:

14. Packaging and deployment of SCA related artifacts, including the relationship to a runtime and characteristics of the runtime are part of the specification. SCA Artifact resolution, plus the use of existing non-SCA mechanisms. SCA Contributions and their metadata. SCA packaging format.

This seems to relate to the OASIS Solution Deployment Descriptor (SDD) TC, which specifies metadata for software packaging and deployment (see http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=sdd). As the chair of the OASIS SDD TC, I encourage the SCA-Assembly group to investigate SDD as one way that item #14 of the proposed charter might be satisfied. The SDD TC would be willing to work with the SCA-Assembly TC, once formed, in this regard.

Disposition: Issue Accepted. This will be corrected in the final submission.

Response:

Proposed change to the SCA-Assembly TC charter:

Add the following text at the end of charter item #14:

Consideration should be given to the use of the OASIS Solution
Deployment Descriptor work [3]

Add the following reference item at the end of the charter:

[3] OASIS Solutions Deployment Descriptor TC

http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=sdd

Impacted Charters: SCA-Assembly.

Issue 8: Consider adding OSGi integration

Raised by: Jeff Mischkinsky.

Original Text: Consider adding OSGi integration

I was wondering if the group had considered some activities/work around integration of SCA and OSGI; at least putting it in scope to allow TC(s) to work on the topic if there is sufficient interest.

Disposition: Issue Accepted. This will be corrected in the final submission.

Response:

Thank you for the observation. Given the start of the Enterprise Expert Group at OSGi, it would seem sensible to cover future dependency requirements for improved OSGi integration within scope of the SCA-J TC. If OSGi were to produce a C++ specification, such a scope addition would seem sensible for SCA-C++ also.

Hence, we propose the following words to the two charters:

The TC MAY address the following work areas. Before work commences on any of the above work items, the TC Charter MUST be clarified (by following the TC Charter Clarification rules defined in the OASIS Technical Committee Process document) to scope each of these work items.

(For the SCA-J charter):

Integration of SCA with OSGi including packaging and deployment of SCA related artifacts in an OSGi environment; use of OSGi to deploy and manage SCA artifacts; use of SCA services from OSGi services, use of OSGi service from SCA services, integration with OSGi registry and definition of required metadata for such integration.

(For the SCA-C-C+ charter):

Should the OSGi community produce a specification for C++, then: Integration of SCA with OSGi including packaging and deployment of SCA related artifacts in

an OSGi environment; use of OSGi to deploy and manage SCA artifacts; use of SCA services from OSGi services, use of OSGi service from SCA services, integration with OSGi registry and definition of required metadata for such integration.

Impacted Charters: SCA-J and SCA-C-C++