Review of TN Using BPEL in a UDDI Registry

Contributors:

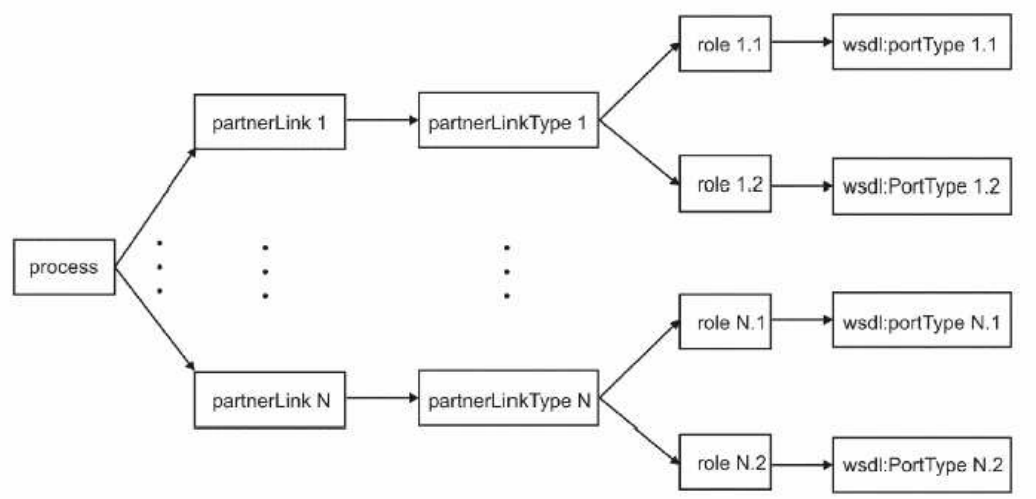
1. Jan Pridal, Systinet
2. Svatopluk Dedic, Systinet
3. Ales Lipovy, Systinet
4. Zdenek Svoboda, Systinet
5. Luc Clément, Systinet

This review touches two different areas – firstly the TN is considered from the BPEL point of view, and secondly from the UDDI standpoint (including correctness of example structures and calls).

1 Mapping the BPEL Process to UDDI Structures

1.1 Issue 1 - Incongruence of Mapping

There is a possible future issue with mapping BPEL to UDDI as it is proposed in this TN. The wsdl:portType is used in the BPEL specification in a following manner:



A `process` contains references to a set of `partnerLinks` where each `partnerLink` references exactly one `partnerLinkType`. In turn, each `partnerLinkType` contains references to one or two different `roles` (played by each of the services in the conversation). Each `role` in turn references an interface which is defined as `wsdl:portType`. It is common practice for these `wsdl:portTypes` to originate from separate namespaces.

The mapping proposed by the TN does not reflect this hierarchical structure but rather flattens it to a list of wsdl:portTypes (tModels) referenced from the process (tModel). Thus the `role` semantic is lost which may be problematic – the extent of which has not been ascertained. That said, the goals set out by this TN (i.e. the queries that should be enabled by this TN) are achieved using this simple flat model.

- [Issue 1] It should be ascertained whether this incongruence in the mapping is problematic.

1.2 Issue 2 - Incongruence of Mapping

What follows is further input relating to issues of incongruence.

Concrete bindings of partners are out of scope of the BPEL process. However, bindings are specified for each partnerLink separately. Consider the case where the same `PortType` (e.g. corp:ApproverPortType) is used in two different `partnerLinks` (manager, supervisor). Each `partnerLink` represents a different level in the corporate approval hierarchy.

With BPEL, the deployer can use different bindings for these `partnerLinks`. The local role of each `PartnerLink` can be bound specifically to allow the business process engine to distinguish incoming communication originatingfrom different `PartnerLinks`.

The proposed BPEL-to-UDDI mapping retains the remote role binding but does not allow it to describe bindings of the local roles to different endpoints as it is typically done in BPEL scripts. Since the bindings are assembled at the process tModel, the relationship between the remote PortType (tModel) and the matching local PortType (tModel) from the `partnerLinkType` definition is lost.

As a result an implementer of the remote role's tModel cannot tell, without additional information, which bindingTemplate of the process tModel it should use for `partnerLinkType` communication to the business process.

- [Issue 2] It should be ascertained whether this incongruence in the mapping is problematic.

1.3 Mapping the bpws:role

In accordance to the TN the bindingTemplate that provides the binding for a tModel (representing the wsdl:portType) that is referenced from the BPEL process tModel, should also contain a tModelInstanceInfo as an indication that it provides a support of that BPEL process. There is no information which bpws:role in the BPEL process this binding supports. This is result of the flat mapping issue mentioned above.

- [Issue 3] It should be ascertained whether this incongruence in the mapping is problematic.

1.4 Mapping of a bpws:process

The mapping is not fully understood the reference from binding template to process tModel. There seems to duplication. For example from the bindingTemplate one can find the associated portType tModel and find all process tModels that reference this portType. Perhaps these references do not represent the same relationship.

- [Issue 4] Could this be clarified?

2 Editorial Comments

The list of editorial comments follows; each issue is introduced by the TN section number.

2.1 Figure 1

The image representing the BPEL process in the top left corner uses the term ‘action’ that is not used in the BPEL specification and is not defined in the TN. The term action represents the following set of BPEL activities: bpws:receive, bpws:reply, bpws:invoke.

- [Issue 5] This info should be included in the text.

2.2 Figure 2

- [Issue 6] The quotation characters should be altered and represented as “”.

2.3 Section 3.1.2

- [Issue 7] The BPEL Entity Type tModel is defined as a unchecked taxonomy; it would be valuable for it to be checked.

2.4 Section 3.1.2.1

- [Issue 8] The tModel listing contains TBD section in its overviewURL.

2.5 Section 3.2.2.1

- [Issue 9] The tModel listing contains TBD section in its overviewURL.

2.6 Section 3.2 - Definition of a WSDL portType Reference tModel

In section 3.2 of the TN the WSDL portType Reference tModel is introduced. A similar tModel is already defined in *Using WSDL in a UDDI Registry, Version 2* TN though in a slightly different context (reference from wsdl:binding tModel to wsdl:portType tModel). It is however defined so generally that it represents the category system used to reference a wsdl:portType tModel. This is actually what the proposed tModel in BPEL mapping TN should do.

- [Issue 10] Is it really necessary to introduce a new category tModel here?
- [Issue 11] Couldn't we reuse the already defined uddi-org:wsdl:portTypeReference tModel?
- [Issue 12] Should the *Using WSDL in a UDDI Registry, Version 2* be generalized to allow for its reuse by this TN? Doing so now is warranted given that Errata #1 is currently under consideration.

2.7 Section 3.2.3

The TN states: 'Valid values for this category system are tModelKeys.' This sentence should probably modified as: 'Valid values for this category system are tModelKeys

of tModels categorized as wsdl:portTypes.'

- [Issue 13] Consider revising text

2.8 Section 4

The example given in this section contains redundant listings and can be simplified – the listing of BPEL process in section 4.1 can be shortened to contain only the partnerLinks element as the rest is of no meaning to TN.

- [Issue 14] Consider revising text

2.9 Section 4.2.3

There are incorrect tModelKeys in tModelInstanceInfos:

- Instead of 'uuid:e1...' there should be 'uuid:a1...'.
- Instead of 'uuid:a1...' there should be 'uuid:a2...'.

- [Issue 15] Update tModelKeys

2.10 Section 4.3.4

- [Issue 16] The first sentence should read 'Find all implementations of ReservationAndBookingTickets process.'
- [Issue 17] Another problem is that the calls do not return all implementations of the process as the text suggests but all the implementations of parts (identified by wsdl:portType each) of the process. There is no such concept of a binding representing the whole implementation of the process introduced in this TN.
- [Issue 18] The closing tag of find_service call is incorrect (find_binding).

2.11 Section 4.5

- [Issue 19] All sample queries in this section and its subsections have wrong namespace, instead of 'urn:uddi-org:api_v2' there should be 'urn:uddi-org:api_v3'.
- [Issue 20] The attribute 'generic' should be removed.

2.12 Section 4.5.4

- [Issue 21] The first sentence should read 'Find all implementations of ReservationAndBookingTickets process.'
- [Issue 22] Another problem is that the calls do not return all implementations of the process as the text suggests but all the implementations of parts (identified by wsdl:portType each) of the process. There is no such concept as a binding representing the whole implementation of the process introduced in this TN.
- [Issue 23] This section is incorrect as it does not talk about UDDI V3 API but instead contains text copied from the UDDI V2 sample queries section. [And moreover the closing tag of find_service call is incorrect (find_binding).] The section should contain following text:

In UDDI V3 API the serviceKey attribute is optional in find_binding call. It's then possible to find all implementations of a process with a single call:

```
<find_service xmlns="urn:uddi–org:api_v3">
   <tModelBag>
      <tModelKey>
         uddi:TravelAgent.com:ReservationAndBookingTicketsProcess
      </tModelKey>
   </tModelBag>
</find_service>
```