



UDDI Specifications TC

UDDI as the registry for ebXML Components

Technical Note

Document identifier:

uddi-spec-tc-tn-uddi-ebxml

Current version:

<http://www.oasis-open.org/committees/uddi-spec/doc/tn/uddi-spec-tc-tn-uddi-ebxml-20040219.htm>

Field Code Changed

Deleted: 20031020

Latest version:

<http://www.oasis-open.org/committees/uddi-spec/doc/tn/uddi-spec-tc-tn-uddi-ebxml.htm>

Previous version:

<http://www.oasis-open.org/committees/uddi-spec/doc/tn/uddi-spec-tc-tn-uddi-ebxml-20031020.htm>

Author:

Keisuke Kibakura, Fujitsu

Editors:

Luc Clément
Daniel Feygin, [UnitSpace](#)
Tony Rogers, Computer Associates

Deleted: , Microsoft

Deleted: Unitspace

Contributors:

Tom Bellwood, IBM
Jacques Durand, Fujitsu
Sam Lee, Oracle
Joel Munter, Intel
Dale Moberg, Cyclone Commerce
Claus von Riegen, SAP
Alok Srivastava, Oracle
Max Voskob, MSI Business Solutions

Abstract:

This document describes the way to model ebXML-based services and ebXML components such as CPP and BPSS, and provides a practical guidance on how to use a UDDI registry as the registry for ebXML components.

Status:

This document is updated periodically on no particular schedule. Send comments to the editor.

36
37
38
39
40
41
42

Committee members should send comments on this technical note to the uddi-spec@lists.oasis-open.org list. Others should [submit](http://www.oasis-open.org/committees/comments/form.php?wg_abbrev=uddi-spec) comments [via http://www.oasis-open.org/committees/comments/form.php?wg_abbrev=uddi-spec](http://www.oasis-open.org/committees/comments/form.php?wg_abbrev=uddi-spec).

For information on whether any intellectual property claims have been disclosed that may be essential to implementing this technical note, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the UDDI Spec TC web page (<http://www.oasis-open.org/committees/uddi-spec/>).

Deleted: subscribe to and send

Deleted: to the

Field Code Changed

Deleted: uddi-spec-comment@lists

Formatted: Default Paragraph Font

Deleted: list. To subscribe, send an email message to uddi-spec-comment-request@lists.oasis-open.org with the word "subscribe" as the body of the message

Table of Contents

44	1	Introduction	4
45	1.1	Problem Statement.....	4
46	1.2	Goals.....	4
47	1.3	Non-goals.....	4
48	1.4	Terminology	4
49	2	Technical Note Solution.....	5
50	2.1	Definitions	5
51	2.2	Technical Note Behavior.....	5
52	2.2.1	Reference Scenario.....	5
53	2.2.2	Modeling Outline and Rationale	6
54	2.2.3	ebXML Specifications Taxonomy tModel.....	7
55	2.2.4	Common ebXML tModels	8
56	2.2.5	Registering ebXML Services.....	12
57	2.2.6	Searching ebXML Services and Components	15
58	2.2.7	Scenario Variation: Using a CPA Template.....	16
59	2.2.8	Scenario Variation: Using Role Information.....	17
60	3	References	19
61	3.1	Normative.....	19
62		Appendix A. Acknowledgments.....	20
63		Appendix B. Revision History.....	21
64		Appendix C. Notices.....	22
65			

Deleted: 20

Deleted: 20

66 1 Introduction

67 This UDDI Spec Technical Committee Technical Note (TN) provides technical guidance on how
68 to use UDDI registries within the ebXML framework of B2B services. Specifically, it addresses the
69 issues related to enabling automated discovery of ebXML framework components, such as
70 Collaboration Protocol Profile and Business Process Specification Schema, using UDDI.

71 By adopting the technical guidance of this TN, users will enable trading partners and their Web
72 services and ebXML infrastructures to interact using UDDI as a common registry.

73 1.1 Problem Statement

74 Multiple consortia have initiated pilot projects using the ebXML framework for business-to-
75 business transactions, while corporations have also begun adopting ebXML technologies for
76 internal use. At the same time Web service technologies, which have significant momentum due
77 to unprecedented industry support, are also being rolled out. UDDI can play a major role in
78 enabling trading partners and their Web services and ebXML infrastructures to interact using
79 UDDI as a common registry. This is the focus of this Technical Note.

80 In addition to being a universal technology for publication and discovery of service metadata,
81 UDDI also enables discovery of ebXML framework components such as Collaboration Protocol
82 Profile and Business Process Specification Schema. This capability can help enable
83 interoperability among trading partners that use UDDI and ebXML framework components.
84 However, a prescribed methodology of modeling services and components which are conformant
85 to ebXML specifications is required to make interoperable solutions possible.

86 1.2 Goals

87 This note provides technical guidance:

- 88 • to model an ebXML-based service;
- 89 • to register an ebXML-based service and ebXML components in a UDDI registry; and
- 90 • to query a UDDI registry for an ebXML-based service and ebXML components.

91 1.3 Non-goals

92 This note does *not* intend:

- 93 • to merge UDDI and ebXML registry technologies;
- 94 • to substitute an ebXML registry with a UDDI registry; or
- 95 • to migrate entity data from an ebXML Registry to a UDDI registry (or vice versa).

96 1.4 Terminology

97 The key words *must*, *must not*, *required*, *shall*, *shall not*, *should*, *should not*, *recommended*, *may*,
98 and *optional* in this document are to be interpreted as described in [RFC2119].

99

2 Technical Note Solution

2.1 Definitions

101 ebXML

102 ebXML is a common XML-based e-business framework which consists of some distinct
103 technologies.

104 ebXML Registry

105 ebXML Registry is intended to provide registry and repository services in the ebXML
106 framework. [\[ebRR1\]](#)[\[ebRR2\]](#)

107 ebXML Message Services

108 ebXML Messaging Services provides a method for conducting e-business transactions
109 within the framework of ebXML technologies. It extends the SOAP protocol to address
110 specific business-to-business transaction requirements such as authentication, reliability
111 and non-repudiation. [\[ebMS1\]](#)[\[ebMS2\]](#)

Deleted: [ebMS1]

Deleted: [ebMS2]

112 ebXML Collaboration Protocol Profile and Agreement 113 (CPPA)

114 This is a specification which defines CPP and CPA. [\[ebCPPA1\]](#)[\[ebCPPA2\]](#)

Deleted: [ebCPPA1]

Deleted: [ebCPPA2]

115 ebXML Collaboration Protocol Profile (CPP)

116 A CPP defines one business partner's technical capabilities to engage in electronic
117 business collaborations with other partners by exchanging electronic messages. These
118 capabilities include business processes, document formats, and technical communication
119 parameters required to communicate with a trading partner. Before starting to do
120 business with a partner via ebXML, trading partners create and exchange their CPPs,
121 recognize the other's technical capability, negotiate for common ground, and build a CPA
122 (Collaboration Protocol Agreement).

123 ebXML Collaboration Protocol Agreement (CPA)

124 A CPA documents the technical agreement between two (or more) partners to engage in
125 electronic business collaboration. It may be formed from two (or more) CPPs, or may be
126 created from a CPA template.

127 ebXML CPA template

128 CPA template is a trading partner's "fill in the blanks" proposal to a prospective trading
129 partner. It has place-holding values that are intended to be replaced by the actual values
130 when a CPA is derived. CPA template may be used to form a CPA as the alternative
131 method to merging two CPPs.

132 UN/CEFACT - ebXML Business Process Specification 133 Schema (BPSS)

134 BPSS provides a machine readable description of a business process. [\[ebBPSS\]](#)

Deleted: [ebBPSS]

135 2.2 Technical Note Behavior

136 2.2.1 Reference Scenario

137 What follows is a typical use case which this Technical Note will use as its reference scenario:

- 138 • a fictitious consortium, named "ABC Consortium" creates a standardized business
139 process for a given industry vertical and describes it as an ebXML BPSS;

- 140 • fictitious seller Company A and buyer Company B each create their own CPPs;
- 141 • Company A implements and registers an ebXML-based service S_a which is conformant to
- 142 the standard process defined by ABC Consortium.

143

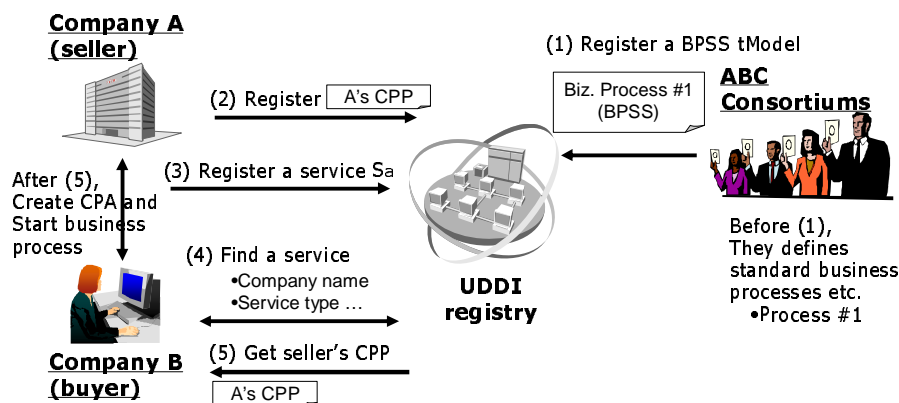
144 Using the guidance provided for by this Technical Note, these partners model their ebXML
 145 components and interact as follows to enable their business processes:

- 146 1. ABC Consortium creates a tModel to represent the standard business process and
 147 publishes it.
- 148 2. Company A registers the location of its CPP as a business service.
- 149 3. Company A registers a service S_a .
- 150 4. Company B finds A's service S_a using a UDDI registry.
- 151 5. Company B locates A's CPP using a UDDI registry.

152

153 Figure 1 shows these five steps.

154



155

Figure 1: Reference scenario

156 These five steps show how Company B obtains Company A's CPP. Company B can then begin
 157 to negotiate and build a CPA, allowing it to do business with Company A. What happens after the
 158 discovery is complete is specific to ebXML and is out of this document's scope.

159 A CPA provides run-time information required for trading partners' business collaboration
 160 software components. A CPA is generally only accessible to the parties that created it and is not
 161 disclosed to any third party, since it implies business partnership between the parties to a CPA,
 162 which can be regarded as a potential trade secret. Therefore this TN does not model any
 163 recommended registration of the CPA in UDDI.

164 The following sections provide specific modeling recommendations covering this scenario.

165 2.2.2 Modeling Outline and Rationale

166 ebXML services and components must be modeled to match UDDI data structures. The bases for
 167 modeling are as follows:

- 168 • A company that provides ebXML-based services is modeled as a businessEntity.
- 169 • An ebXML-based service is modeled as a businessService.

- 170 • A tModel is created to represent each ebXML specification.
- 171 • A company's CPP is modeled as a businessService belonging to the businessEntity
172 whose capabilities it describes. This fits the one-to-many cardinality of company-to-CPP
173 relationships, which are never shared with external parties, just as a businessEntity's
174 contained businessServices.
- 175 • A business process definition (BPSS) is modeled as a tModel. A BPSS instance can be
176 regarded as one of technical characteristics of a service interface, and can be used by
177 multiple organizations.

178 This Technical Note defines three types of tModels:

179 **Specifications Taxonomy tModel**

180 This taxonomy is used to categorize each ebXML specification represented by a
181 Common ebXML tModel (see below) to aid in their discovery.

182 **Common ebXML tModels**

183 These tModels refer to specific versions of ebXML specifications/concepts, such
184 as CPP v1.0, UN/CEFACT BPSS v1.10, Message Service v2.0, and so on.

185 **Proprietary tModels**

186 These tModels refer to specific instances or implementations of ebXML
187 specifications. For instance, a particular business process, described in an
188 ebXML BPSS, is an instance of BPSS, and it can be defined as a proprietary
189 tModel.

190 The models described in this Technical Note apply equally to both UDDI v2 and v3 registries,
191 though all the examples are written in terms of UDDI v2.

192 **2.2.3 ebXML Specifications Taxonomy tModel**

193 ebXML is a framework that consists of several distinct technical specifications, each of which
194 must be represented by tModels in a UDDI registry. UDDI tModels are usually tagged with
195 categorizations denoting information that facilitates their discovery. That is why before defining
196 tModels representing ebXML specifications, we define an ebXML Specifications Taxonomy
197 tModel, which is used to categorize these tModels (described in the following section).

198 **ebXML Specification Taxonomy tModel:**

199 **tModel Name:** *ebxml-org:specifications*
200 **tModel Description:** ebXML Specifications Taxonomy
201 **tModel UDDI Key (V3):** uddi:ebxml.org:specifications
202 **Derived V1, V2 format Key:** uuid:da52cf72-2cb2-39a9-ad1e-577e66d8a6f6
203 **Categorization:** categorization
204 **Checked:** No

205

```

206 <tModel tModelKey="uuid:da52cf72-2cb2-39a9-ad1e-577e66d8a6f6">
207 <name>ebxml-org:specifications</name>
208 <description xml:lang="en">ebXML Specifications Taxonomy</description>
209 <overviewDoc>
210 <overviewURL>http://www.oasis-open.org/committees/uddi-
211 spec/doc/tn/uddi-spec-tc-tn-uddi-ebxml.htm#ebxmlspec</overviewURL>
212 </overviewDoc>
213 <categoryBag>
214 <keyedReference
215 tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
216 keyName="uddi-org:types"
217 keyValue="categorization" />

```

Deleted: _home.php?wg_abbr
ev=
Deleted: -jc/
Deleted: 39b756d62ab4

218
219
220
221
222
223

```
<keyedReference
  tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
  keyName="uddi-org:types"
  keyValue="unchecked" />
</categoryBag>
</tModel>
```

Deleted: 39b756d62ab4

224 The following values are defined for this ebXML Specifications taxonomy. These values are
225 useful for classifying ebXML-related tModels and are helpful for others who want to find those
226 tModels.

- 227 **ebXML:CPPA** ebXML Collaboration Protocol Profile and Agreement
- 228 **ebXML:MS** ebXML Message Service
- 229 **ebXML:BPSS** UN/CEFACT - ebXML Business Process Specification Schema

230 Please note that this tModel is a categorization system for ebXML framework specifications and
231 does not represent ebXML specifications themselves.

232 2.2.4 Common ebXML tModels

233 In UDDI, a service interface (which is called a binding template in UDDI) is associated with
234 tModels, which refer to the technical specification of the interface. Those tModels represent
235 service type information. This collection of tModels is sometimes referred to as the technical
236 fingerprint of the service interface.

237 In the case of ebXML specifications, Message Service, CPP, CPA template¹ and BPSS all
238 convey relevant service type information. It can be captured in tModels, which we will refer to as
239 common ebXML tModels. “Common” means that these tModels are relied on by all users who
240 use UDDI as the registry for ebXML services and components. Since these tModels are
241 categorized by ebXML Specifications Taxonomy (ebxml-org:specifications) defined in the
242 previous section, they are easy to find.

243 ebXML Message Service v1.0 tModel

- 244 **tModel Name:** **ebxml-org:MessageService:v1_0**
- 245 **tModel Description:** ebXML Message Service v1.0
- 246 **tModel UDDI Key (V3):** uddi:ebxml.org:messageservice:v1.0
- 247 **Derived V1, V2 format Key:** uuid:c8692873-1842-3b32-b980-8fa6d16676d2
- 248 **Categorization:** specification

¹ This tModel is not used in the scenario, but is used in its variation. See Section 2.2.7.

249

```

250 <tModel tModelKey="uuid:c8692873-1842-3b32-b980-8fa6d16676d2">
251 <name>ebxml-org:MessageService:v1_0</name>
252 <description xml:lang="en">ebXML Message Service v1.0</description>
253 <overviewDoc>
254 <overviewURL>http://www.ebxml.org/specs/ebMS.pdf</overviewURL>
255 </overviewDoc>
256 <categoryBag>
257 <keyedReference
258 tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
259 keyName="uddi-org:types"
260 keyValue="specification" />
261 <keyedReference
262 tModelKey="uuid:da52cf72-2cb2-39a9-ad1e-577e66d8a6f6"
263 keyName="ebXML Message Service"
264 keyValue="ebXML:MS" /> <!-- Message Service spec -->
265 </categoryBag>
266 </tModel>

```

Deleted: 39b756d62ab4

267

268 ebXML Message Service v2.0 tModel

269 **tModel Name:** ebxml-org:MessageService:v2_0

270 **tModel Description:** ebXML Message Service v2.0

271 **tModel UDDI Key (V3):** uddi:ebxml.org:messageservice:v2.0

272 **Derived V1, V2 format Key:** uuid:9a3b93be-515e-34c7-90c7-b05cdfdba8c3

273 **Categorization:** specification

274

```

275 <tModel tModelKey="uuid:9a3b93be-515e-34c7-90c7-b05cdfdba8c3">
276 <name>ebxml-org:MessageService:v2_0</name>
277 <description xml:lang="en">ebXML Message Service v2.0</description>
278 <overviewDoc>
279 <overviewURL>http://www.oasis-open.org/committees/ebxml-
280 msg/documents/ebMS_v2_0.pdf</overviewURL>
281 </overviewDoc>
282 <categoryBag>
283 <keyedReference
284 tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
285 keyName="uddi-org:types"
286 keyValue="specification" />
287 <keyedReference
288 tModelKey="uuid:da52cf72-2cb2-39a9-ad1e-577e66d8a6f6"
289 keyName="ebXML Message Service"
290 keyValue="ebXML:MS" /> <!-- Message Service spec -->
291 </categoryBag>
292 </tModel>

```

Deleted: 39b756d62ab4

293

294 ebXML CPP v1.0 tModel

295 **tModel Name:** ebxml-org:CollaborationProtocolProfile:v1_0

296 **tModel Description:** ebXML Collaboration Protocol Profile v1.0

297 **tModel UDDI Key (V3):** uddi:ebxml.org:collaborationprotocolprofile:v1.0

298 **Derived V1, V2 format Key:** uuid:e3f3df4f-b221-33b4-a3ff-17b21410c565

299 **Categorization:** specification

300

301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318

```

<tModel tModelKey="uuid:e3f3df4f-b221-33b4-a3ff-17b21410c565">
  <name>ebxml-org:CollaborationProtocolProfile:v1_0</name>
  <description xml:lang="en">ebXML Collaboration Protocol Profile
v1.0</description>
  <overviewDoc>
    <overviewURL>http://www.ebxml.org/specs/ebCCP.pdf</overviewURL>
  </overviewDoc>
  <categoryBag>
    <keyedReference
      tModelKey="uuid:clacf26d-9672-4404-9d70-39b756e62ab4"
      keyName="uddi-org:types"
      keyValue="specification" />
    <keyedReference
      tModelKey="uuid:da52cf72-2cb2-39a9-adle-577e66d8a6f6"
      keyName="ebXML Collaboration Protocol Profile and Agreement"
      keyValue="ebXML:CPPA" />      <!-- This belongs to CPPA spec -->
  </categoryBag>
</tModel>

```

Deleted: 39b756d62ab4

319

320 ebXML CPP v2.0 tModel

321 **tModel Name:** ebxml-org:CollaborationProtocolProfile:v2_0

322 **tModel Description:** ebXML Collaboration Protocol Profile v2.0

323 **tModel UDDI Key (V3):** uddi:ebxml.org:collaborationprotocolprofile:v2.0

324 **Derived V1, V2 format Key:** uuid:43ed4af4-eacf-3b20-95d1-c7c197f5d9d0

325 **Categorization:** specification

326

327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344

```

<tModel tModelKey="uuid:43ed4af4-eacf-3b20-95d1-c7c197f5d9d0">
  <name>ebxml-org:CollaborationProtocolProfile:v2_0</name>
  <description xml:lang="en">ebXML Collaboration Protocol Profile
v2.0</description>
  <overviewDoc>
    <overviewURL>http://www.ebxml.org/specs/ebcpp-2_0.pdf</overviewURL>
  </overviewDoc>
  <categoryBag>
    <keyedReference
      tModelKey="uuid:clacf26d-9672-4404-9d70-39b756e62ab4"
      keyName="uddi-org:types"
      keyValue="specification" />
    <keyedReference
      tModelKey="uuid:da52cf72-2cb2-39a9-adle-577e66d8a6f6"
      keyName="ebXML Collaboration Protocol Profile and Agreement"
      keyValue="ebXML:CPPA" />      <!-- This belongs to CPPA spec -->
  </categoryBag>
</tModel>

```

Deleted: oasis-open

Deleted: committees/ebxml-cppa/documents/ebCPP

Deleted: _

Deleted: 39b756d62ab4

345

346 ebXML CPA v1.0 Template tModel

347 **tModel Name:** ebxml-org:CollaborationProtocolAgreement:v1_0:Template

348 **tModel Description:** ebXML Collaboration Protocol Agreement v1.0 Template

349 **tModel UDDI Key (V3):** uddi:ebxml.org:collaborationprotocolagreement:v1.0:template

350 **Derived V1, V2 format Key:** uuid:5ab4e3af-2e67-3a4f-b9b7-92a436be8f43

351 **Categorization:** xmlSpec

352

```
353 <tModel tModelKey="uuid:5ab4e3af-2e67-3a4f-b9b7-92a436be8f43">
354 <name>ebxml-org:CollaborationProtocolAgreement:v1_0:Template</name>
355 <description xml:lang="en">
356 ebXML Collaboration Protocol Agreement v1.0 Template
357 </description>
358 <overviewDoc>
359 <overviewURL>http://www.ebxml.org/specs/ebCCP.pdf</overviewURL>
360 </overviewDoc>
361 <categoryBag>
362 <keyedReference
363 tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
364 keyName="uddi-org:types"
365 keyValue="xmlSpec" />
366 <keyedReference
367 tModelKey="uuid:da52cf72-2cb2-39a9-ad1e-577e66d8a6f6"
368 keyName="ebXML Collaboration Protocol Profile and Agreement"
369 keyValue="ebXML:CPPA" /> <!-- This belongs to CPPA spec -->
370 </categoryBag>
371 </tModel>
```

Deleted: 39b756d62ab4

372

373 ebXML CPA v2.0 Template tModel

374 **tModel Name:** ebxml-org:CollaborationProtocolAgreement:v2_0:Template

375 **tModel Description:** ebXML Collaboration Protocol Agreement v2.0 Template

376 **tModel UDDI Key (V3):** uddi:ebxml.org:collaborationprotocolagreement:v2.0:template

377 **Derived V1, V2 format Key:** uuid:86c6fcb3-8c73-33a7-8635-38438b76aee7

378 **Categorization:** xmlSpec

379

```
380 <tModel tModelKey="uuid:86c6fcb3-8c73-33a7-8635-38438b76aee7">
381 <name>ebxml-org:CollaborationProtocolAgreement:v2_0:Template</name>
382 <description xml:lang="en">
383 ebXML Collaboration Protocol Agreement v2.0 Template
384 </description>
385 <overviewDoc>
386 <overviewURL>http://www.ebxml.org/specs/ebcpp-2_0.pdf</overviewURL>
387 </overviewDoc>
388 <categoryBag>
389 <keyedReference
390 tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
391 keyName="uddi-org:types"
392 keyValue="xmlSpec" />
393 <keyedReference
394 tModelKey="uuid:da52cf72-2cb2-39a9-ad1e-577e66d8a6f6"
395 keyName="ebXML Collaboration Protocol Profile and Agreement"
396 keyValue="ebXML:CPPA" /> <!-- This belongs to CPPA spec -->
397 </categoryBag>
398 </tModel>
```

Deleted: oasis-open

Deleted: committees/ebxml-cppa/documents/ebCPP

Deleted: _

Deleted: 39b756d62ab4

399

400 UN/CEFACT - ebXML Business Process Specification Schema v1.10 tModel

401 **tModel Name:** untmg-org:BusinessProcessSpecificationSchema:v1_10

402 **tModel Description:** UN/CEFACT - ebXML Business Process Specification Schema
 403 v1.10
 404 **tModel UDDI Key (V3):** uddi:untmg.org:businessprocessspecificationschema:v1.10
 405 **Derived V1, V2 format Key:** uuid:1a2a88af-54f8-316c-aaf1-e1fc2ef1c0e9
 406 **Categorization:** specification
 407

```

408 <tModel tModelKey="uuid:1a2a88af-54f8-316c-aaf1-e1fc2ef1c0e9">
409 <name>untmg-org:BusinessProcessSpecificationSchema:v1_10</name>
410 <description xml:lang="en">UN/CEFACT - ebXML Business Process
411 Specification Schema v1.10</description>
412 <overviewDoc>
413
414 <overviewURL>http://www.untmg.org/downloads/General/approved/ebBPSS-
415 vlpt10.zip</overviewURL>
416 </overviewDoc>
417 <categoryBag>
418 <keyedReference
419 tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
420 keyName="uddi-org:types"
421 keyValue="specification" />
422 <keyedReference
423 tModelKey="uuid:da52cf72-2cb2-39a9-ad1e-577e66d8a6f6"
424 keyName="UN/CEFACT - ebXML Business Process Specification Schema"
425 keyValue="ebXML:BPSS" /> <!-- This belongs to BPSS spec -->
426 </categoryBag>
427 </tModel>
  
```

Deleted: 39b756d62ab4

428

429 2.2.5 Registering ebXML Services

430 The tModels defined so far enable us to model and register ebXML-based services in a UDDI
 431 registry. In accordance with the scenario presented in section 2.2.1, we show the details of each
 432 step one by one.

433 A consortium registers a tModel of a BPSS instance

434 In this section, the step below is shown:

- 435 1. ABC Consortium creates a tModel to represent the standard business process and
 436 publishes it.

437 It is natural to regard a business process definition as one of technical characteristics of a service
 438 interface, since it describes business choreography which the service follows. Every registered
 439 businessService conformant to an ebXML BPSS description should be associated with a tModel
 440 of the corresponding BPSS instance.

441 In the reference scenario, a fictitious consortium, ABC Consortium, defines a standard business
 442 process (here, we call it "ABC Standard Process #1"), which is widely used by the consortium
 443 members. The consortium describes it using a BPSS instance.

444 Assuming that the BPSS instance is located at "http://abc.org/process_1.bps", we define the
 445 tModel like below:

446 **tModel Name:** *abc-org:StandardProcess:1*
 447 **tModel Description:** ABC Consortium's Standard Process #1

448 **tModel UDDI Key²:** uuid:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxx1

449 **Categorization:** xmlSpec

450

```
451 <tModel tModelKey="uuid:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxx1">
452 <name>abc-org:StandardProcess:1</name>
453 <description xml:lang="en"> ABC's standard process #1</description>
454 <overviewDoc>
455 <overviewURL>http://abc.org/process_1.bps</overviewURL>
456 </overviewDoc>
457 <categoryBag>
458 <keyedReference
459 tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
460 keyName="uddi-org:types"
461 keyValue="xmlSpec" />
462 <keyedReference
463 tModelKey="uuid:1a2a88af-54f8-316c-aaf1-el1fc2ef1c0e9"
464 keyName="UN/CEFACT - ebXML Business Process Specification Schema
465 v1.10"
466 keyValue="untmg-org:BusinessProcessSpecificationSchema:v1_10" />
467 </categoryBag>
468 </tModel>
```

Deleted: 39b756d62ab4

469 ABC Consortium registers this tModel using `save_tModel`. Later, it is referenced from a service
470 whose choreography is conformant to this BPSS instance.

471 **Company A registers CPP**

472 In this section, the step below is shown:

473 2. Company A registers the location of its CPP as a business service.

474 In UDDI, all business services must be associated with a business entity. A company must first
475 create a businessEntity if it decides to register an ebXML service. This Technical Note
476 recommends that the location of the CPP should be registered as an accessPoint in a
477 businessService. Assuming that Company A's CPP is located at "http://a.com/a.cpp"³, its
478 businessEntity and businessService are as follows:

```
479 <businessEntity>
480 <name xml:lang="en">Company A</name>
481 <contacts>...</contacts>
482 <businessServices>
483 <businessService>
484 <name xml:lang="en">Company A's CPP</name>
485 <description xml:lang="en">Company A's CPP</description>
486 <bindingTemplates>
487 <bindingTemplate>
488 <accessPoint URLType="http">
489 http://a.com/a.cpp
490 </accessPoint>
491 <description xml:lang="en">A's CPP is here!</description>
492 <tModelInstanceDetails>
493 <tModelInstanceInfo
```

² In the examples in this technical note, v2 format keys are used. Note that entity key compatibility with v3 and earlier versions of UDDI should be considered unless the publisher intends to use this proprietary tModel in a closed environment only (e.g. a private registry). See Section 10 "Multi-Version Support" in [\[UDDIV3\]](#).

Deleted: [UDDIV3]

³ In this scenario, we assume that Company A has just one CPP. Though it is possible for a business entity to have multiple CPPs, the modeling of such a case is more complex.

```

494         tModelKey="uuid:e3f3df4f-b221-33b4-a3ff-17b21410c565">
495         <description>
496             ebxml-org:CollaborationProtocolProfile:v1_0
497         </description>
498         </tModelInstanceInfo>
499     </tModelInstanceDetails>
500 </bindingTemplate>
501 </bindingTemplates>
502 </businessService>
503 </businessServices>
504 </businessEntity>

```

505 The business service above can be regarded as a service which delivers A's CPP. Company A
506 issues save_business API call with above contents within save_business element to register
507 its CPP. If Company A is already registered in a UDDI registry, its CPP can be registered by
508 issuing a save_service API call using businessService defined above.

509 **Company A registers an ebXML-based service**

510 In this section, the step below is shown:

511 3. Company A registers a service S_a.

512 Company A already has a businessEntity which contains one businessService that delivers CPP.
513 Now it can add a service S_a which uses ebXML Message Services protocol, and is conformant to
514 the Standard Process #1 defined by ABC Consortium. Assuming that S_a is an online wine shop,
515 and is provided at "http://a.com/wine/acceptPurchaseOrder", the new businessService is as
516 follows:

```

517 <businessService businessKey="...Company A's businessKey...">
518 <name xml:lang="en">A's online wine shop</name>
519 <description xml:lang="en">an ebXML-based service</description>
520 <bindingTemplates>
521 <bindingTemplate>
522 <accessPoint URLType="http">
523     http://a.com/wine/acceptPurchaseOrder
524 </accessPoint>
525 <description xml:lang="en">A's ebXML-based service</description>
526 <tModelInstanceDetails>
527 <tModelInstanceInfo>
528     tModelKey="uuid:c8692873-1842-3b32-b980-8fa6d16676d2">
529     <description>ebxml-org:MessageService:v1_0</description>
530 </tModelInstanceInfo>
531 <tModelInstanceInfo>
532     tModelKey="uuid:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxbp1">
533     <description>abc-org:StandardProcess:l</description>
534 </tModelInstanceInfo>
535 </tModelInstanceDetails>
536 </bindingTemplate>
537 </bindingTemplates>
538 <categoryBag>
539 <keyedReference>
540     tModelKey="uuid:CD153257-086A-4237-B336-6BDCBDCC6634"
541     keyValue="50.20.22.05"
542     keyName="UNSPSC:Sparkling_Wine" /> <!-- Service category info -->
543 </keyedReference>
544 </categoryBag>
</businessService>

```

545 Company A issues save_service API with above contents to register the service.

546 2.2.6 Searching ebXML Services and Components

547 In this section, we show how Company B (buyer) finds a seller and retrieves a seller's CPP.
548 Since the examples below are for explanation, they may sometimes be a little redundant.

549 Company B finds a service

550 In this section, the step below is shown:

551 4. Company B finds A's service S_a using a UDDI registry.

552 UDDI provides some powerful search capabilities. It is possible to find by business name, service
553 category, service type, and so on. What follows are examples that illustrate uses of some of the
554 UDDI APIs and the return structures of services and tModels registered in accordance with this
555 TN. There are more (efficient) ways to obtain and query UDDI if richer use of categorization is
556 employed for example.

557 In our scenario, it is assumed that Company B (buyer) is capable of doing business using ebXML
558 framework, adopts the ABC Consortium's Standard Business Process #1, and wants to consume
559 a service which is provided by an ebXML-ready seller.

560 The sample XML below finds a service provider who has an ebXML CPP (which means that it is
561 ebXML-ready):

```
562 <find_business>  
563 <tModelBag>  
564 <!-- ebxml-org:CollaborationProtocolProfile:v1_0 -->  
565 <tModelKey>uuid:e3f3df4f-b221-33b4-a3ff-17b21410c565</tModelKey>  
566 </tModelBag>  
567 </find_business>
```

568 Company B issues the above inquiry and gets a businessList like below as a result:

```
569 <businessList>  
570 <businessInfos>  
571 <businessInfo businessKey="...Company A's businessKey...">  
572 <name>Company A</name>  
573 <serviceInfos>  
574 <serviceInfo serviceKey="...serviceKey of CPP...">  
575 <name>Company A's CPP</name> <!-- Company A has a CPP! -->  
576 </serviceInfo>  
577 </serviceInfos>  
578 </businessInfo>  
579 </businessInfos>  
580 </businessList>
```

581 Thus Company B finds that Company A is a seller who provides an ebXML-based online service.
582 Next, Company B checks if Company A provides a service that is conformant to the ABC
583 Consortium's Standard Business Process #1 adopted by Company B.

584 The example below finds Company A's service which uses ebXML Message Services as
585 communication protocol and which is conformant to the Standard Process #1 defined by the ABC
586 Consortium.

```
587 <find_service businessKey="..Company A's businessKey...">  
588 <tModelBag>  
589 <!-- ebxml-org:MessageService:v1_0 -->  
590 <tModelKey>uuid:c8692873-1842-3b32-b980-8fa6d16676d2</tModelKey>  
591 <!-- abc-org:StandardProcess:1 -->  
592 <tModelKey>uuid:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxx1</tModelKey>  
593 </tModelBag>  
594 </find_service>
```

595 Thus Company B has found Company A's ebXML service which meets the requirements. The
596 next step is to obtain Company A's CPP to negotiate communication parameters.

597 **Company B obtains A's CPP**

598 In this section, the step below is shown:

599 5. Company B locates A's CPP using the UDDI registry.

600 The sample `find_service` invocation below allows to locate Company A's CPP.

```
601 <find_service businessKey="...Company A's businessKey...">
602   <tModelBag>
603     <!-- ebxml-org:CollaborationProtocolProfile:v1_0 -->
604     <tModelKey>uuid:e3f3df4f-b221-33b4-a3ff-17b21410c565</tModelKey>
605   </tModelBag>
606 </find_service>
```

607 Company B issues the above inquiry and gets a `serviceList` like below as a result:

```
608 <serviceList>
609   <serviceInfos>
610     <serviceInfo businessKey="..Company A's businessKey..."
611       serviceKey="...serviceKey of CPP...">
612       <name>Company A's CPP</name>
613     </serviceInfo>
614   </serviceInfos>
615 </serviceList>
```

616 Once Company B has the `serviceKey` of the business service which contains the information
617 about A's CPP, it can issue a `get_serviceDetail` to retrieve the access point for the CPP.

```
618 <get_serviceDetail>
619   <serviceKey>"...serviceKey of CPP..."</serviceKey>
620 </get_serviceDetail>
```

621 Company B gets full information of the `businessService` and knows that Company A's CPP is
622 located at "http://a.com/a.cpp". Now Company B can fetch that CPP and then initiate negotiations
623 with Company A. This calls for Company B providing its own CPP, negotiating the definition of a
624 CPA⁴, and then consuming S_a.

625 In a real system, the inquiries shown in the examples above would not be used since they are
626 redundant and inefficient. This section is provided to give an understanding of the concepts
627 involved.

628 **2.2.7 Scenario Variation: Using a CPA Template**

629 While [\[ebCPA2\]](#) suggests that a CPA is formed from two CPPs, it also mentions an alternative
630 approach which uses a CPA template to create a CPA. Since a CPA template represents one
631 company's proposed configuration of service interface to perform business collaboration, the
632 other trading partner can simply replace the placeholder values to form an agreed CPA. It is an
633 easier way of negotiating a CPA.

Deleted: [ebCPA2]

634 In our scenario described in section 2.2.1, a CPA template provided by Company A (seller) can
635 be used as an alternative to Company A's CPP. In this case, Company A can publish a business
636 service shown below at the Step 2:

```
637 <businessService>
638   <name xml:lang="en">Company A's CPA Template</name>
639   <description xml:lang="en">
640     CPA template provided by Company A
641   </description>
642   <bindingTemplates>
643     <bindingTemplate>
```

⁴ The ways to exchange CPP and to negotiate CPA are not defined here.


```

644     <accessPoint URLType="http">
645         http://a.com/a.cpa-template
646     </accessPoint>
647     <description xml:lang="en">CPA template is here!</description>
648     <tModelInstanceDetails>
649         <tModelInstanceInfo
650             tModelKey="uuid:5ab4e3af-2e67-3a4f-b9b7-92a436be8f43">
651             <description>
652                 ebxml-org:CollaborationProtocolAgreement:v1_0:Template
653             </description>
654             </tModelInstanceInfo>
655         </tModelInstanceDetails>
656     </bindingTemplate>
657 </bindingTemplates>
658 </businessService>

```

659 The rest of the steps are similar to the original scenario except that this variation uses a CPA
660 template instead of a CPP.

661 2.2.8 Scenario Variation: Using Role Information

662 In the reference scenario, there appears to be an unspoken agreement with regard to "role" (i.e.
663 buyer vs. seller) between a publisher and an inquirer that the former provides a service and the
664 latter consumes the service.

665 Generally a BPSS instance describes all the roles involved in a business process, such as 'buyer'
666 and 'seller'. When there are multiple parties providing a service within the same BPSS instance,
667 we cannot suppose such implied agreement. Therefore it is impossible to determine which role of
668 the ones described in a BPSS instance the service published to a UDDI registry belongs to.

669 Role information is one of the characteristics of the service provider, which varies with the service.
670 Modeling service roles in UDDI data structures in the way that aids discovery would be useful for
671 an inquirer to filter out irrelevant service registrations.

672 Company A registers role with service information

673 In Step 3 of the reference scenario, Company A could provide its role with a `keyedReference` in
674 the `categoryBag` of the `businessService` like below:

```

675     <categoryBag>
676         <keyedReference
677             tModelKey="uuid:CD153257-086A-4237-B336-6BDCBDCC6634"
678             keyValue="50.20.22.05"
679             keyName="UNSPSC:Sparkling_Wine" /> <!-- Service category info -->
680         <keyedReference tModelKey="UUID:A035A07C-F362-44dd-8F95-
681 E2B134BF43B4"
682             keyName="role"
683             keyValue="seller" />
684     </categoryBag>

```

685 In this example, `uddi-org:general_keywords` `tModel` is used to describe that the service acts as a
686 "seller". Generally a business process (i.e., a BPSS instance) has its own vocabulary to specify a
687 role, such as "seller", "shipper", "retailer", "service provider", and so on. Therefore, it is impossible
688 to define a fixed word set to create a universal role taxonomy in advance. By using
689 `general_keywords` taxonomy, a publisher can identify his role within the context of a specific
690 business process.

691 A publisher's role is described in its CPP, using the role vocabulary that is defined by a BPSS
692 instance. A publisher should pick the name of the role from its CPP, and use it as the
693 `keyedReference`'s `keyValue` so that a knowledgeable inquirer can use it for a query.

694 **Company B finds a service using role**

695 At the Step 4 in the reference scenario, Company B could find a service using role information. In
696 the case that an inquirer seeks a service provider whose service is conformant to a certain BPSS
697 instance, it is quite natural to suppose that the inquirer is very familiar with the business process,
698 otherwise they would fail to collaborate even if they came to know each other.

699 Therefore, we can assume that Company B knows the business process well, which means that
700 the role name of the counter party Company B looks for is also known at the time of inquiry.

701 Company B could filter out undesirable results by using adding the below categoryBag to a
702 find_service inquiry:

```
703 <categoryBag>  
704   <keyedReference tModelKey="UUID:A035A07C-F362-44dd-8F95-  
705   E2B134BF43B4" keyName="role" keyValue="seller" />  
706 </categoryBag>
```

707 The rest of the steps are similar to the original scenario.

708

3 References

3.1 Normative

- 710 **[RFC2119]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,
711 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 712 **[UDDIV3]** L. Clément et al, *UDDI V3.0 Published Specification*,
713 <http://uddi.org/pubs/uddi-v3.00-published-20020719.pdf>, OASIS UDDI
714 Spec TC Committee Specification, July 2002.
- 715 **[ebBPSS]** UN/CEFACT Techniques and Methodologies Group (TMG), *ebXML -*
716 *Business Process Specification Schema Version 1.10*,
717 <http://www.untmg.org/downloads/General/approved/ebBPSS-v1pt10.zip>,
718 Oct 2003.
- 719 **[ebCPA1]** ebXML Trading-Partners Team, *Collaboration-Protocol Profile and*
720 *Agreement Specification Version 1.0*,
721 <http://www.ebxml.org/specs/ebCPP.pdf>, May 2001.
- 722 **[ebCPA2]** OASIS ebXML Collaboration Protocol Profile and Agreement Technical
723 Committee, *Collaboration-Protocol Profile and Agreement Specification*
724 *Version 2.0*, <http://www.ebxml.org/specs/ebcpp-2.0.pdf>, September
725 2002.
- 726 **[ebMS1]** ebXML Transport, Routing & Packaging Team, *Message Service*
727 *Specification Version 1.0*, <http://www.ebxml.org/specs/ebMS.pdf>, May
728 2001.
- 729 **[ebMS2]** OASIS ebXML Messaging Services Technical Committee, *Message*
730 *Service Specification Version 2.0*,
731 <http://www.ebxml.org/specs/ebMS2.pdf>, April 2002.
- 732 **[ebRR1]** OASIS ebXML Registry Technical Committee, *Registry Information*
733 *Model v2.0*, <http://www.ebxml.org/specs/ebrrim2.pdf>, December 2001.
- 734 **[ebRR2]** OASIS ebXML Registry Technical Committee, *Registry Services*
735 *Specification*, <http://www.ebxml.org/specs/ebrrs2.pdf>, December 2001.
- 736

737 **Appendix A. Acknowledgments**

738 The author(s) would like to thank Joel Munter, Sean MacRoibeaird, and Ed Mooney for their
739 contributions to this work.

Appendix B. Revision History

Rev	Date	By Whom	What
1	January 8, 2003	Keisuke Kibakura	Initial version.
2	January 31, 2003	Keisuke Kibakura	Clarification statement added: <ul style="list-style-type: none"> - Support for UDDI v2 and v3. - No intent to substitute ebXML RR. Improved modeling: <ul style="list-style-type: none"> - Restructured spec taxonomy - Addition of common tModels Brief outline of modeling Terminology clarification <ul style="list-style-type: none"> - CPPA, CPP, CPA, CPA template Added scenario variation.
3	February 27, 2003	Keisuke Kibakura	Updates based on the F2F discussion: <ul style="list-style-type: none"> - Added CPA positioning. - Added modeling rationale. - tModel keys changed. - Added footnote on multiple CPPs. - Removed errors in the example.
4	March 19, 2003	Luc Clément Daniel Feygin Tony Rogers Keisuke Kibakura	Reviewed by editors.
5	April 16, 2003	Keisuke Kibakura	Refinement of tModel categorization. Added new section on role information.
6	May 8, 2003	Keisuke Kibakura Daniel Feygin Tony Rogers Luc Clément	Final Draft
7	September 20-22, 2003	Daniel Feygin	Incorporated ebXML JC's input.
8	October 18, 2003	Luc Clément	Updated the ebXML BPSS references to the UN/CEFACT BPSS 1.10 spec
9	February 19, 2004	Daniel Feygin	Corrected an error in the UDDI v2 key of the <code>uddi.org:types</code> taxonomy Updated a number of overviewURLs to point to revised location of ebXML specs Changed the overview URL of the <code>ebXML Specification Taxonomy tModel</code> to point to section 2.2.3 of this Technical Note.

742

Appendix C. Notices

743 OASIS takes no position regarding the validity or scope of any intellectual property or other rights
744 that might be claimed to pertain to the implementation or use of the technology described in this
745 document or the extent to which any license under such rights might or might not be available;
746 neither does it represent that it has made any effort to identify any such rights. Information on
747 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS
748 website. Copies of claims of rights made available for publication and any assurances of licenses
749 to be made available, or the result of an attempt made to obtain a general license or permission
750 for the use of such proprietary rights by implementors or users of this specification, can be
751 obtained from the OASIS Executive Director.

752 OASIS invites any interested party to bring to its attention any copyrights, patents or patent
753 applications, or other proprietary rights which may cover technology that may be required to
754 implement this specification. Please address the information to the OASIS Executive Director.

755 **Copyright** © OASIS Open 2003-[2004](#). *All Rights Reserved.*

756 This document and translations of it may be copied and furnished to others, and derivative works
757 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,
758 published and distributed, in whole or in part, without restriction of any kind, provided that the
759 above copyright notice and this paragraph are included on all such copies and derivative works.
760 However, this document itself does not be modified in any way, such as by removing the
761 copyright notice or references to OASIS, except as needed for the purpose of developing OASIS
762 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual
763 Property Rights document must be followed, or as required to translate it into languages other
764 than English.

765 The limited permissions granted above are perpetual and will not be revoked by OASIS or its
766 successors or assigns.

767 This document and the information contained herein is provided on an "AS IS" basis and OASIS
768 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO
769 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE
770 ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
771 PARTICULAR PURPOSE.