

4

WSDL Technical Note

5 **Document identifier:**

6 wsdl-TN-V2.00-Draft-20020926

7 **Location:**

8 [http://tbd]

9 **Authors (alphabetically):**

10 John Colgrave, IBM colgrave@uk.ibm.com

11 Karsten Januszewski, Microsoft karstenj@microsoft.com

12 **Abstract:**

13 This document is an OASIS UDDI Technical Note that defines a new approach to using WSDL in a UDDI
14 Registry.

15 **Status:**

16 This document is a working draft.

17 Committee members should send comments on this document to the uddi-spec@lists.oasis-open.org list.
18 Others should subscribe to and send comments to the uddi-spec-comment@lists.oasis-open.org list. To
19 subscribe, send an email message to uddi-spec-comment-request@lists.oasis-open.org with the word
20 "subscribe" as the body of the message.

21 Copyright

22 *Copyright (C) OASIS Open September 2002. All Rights Reserved.*

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

30 *The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or*
31 *assigns.*

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

36 **Table of Contents**

37	Copyright	2
38	<i>Table of Contents</i>	3
39	1 Introduction	6
40	1.1 Goals and Requirements	6
41	1.2 Relationship to Version 1 Best Practice	6
42	1.3 tModel Keys	7
43	2 Rationalizing Two Data Models: WSDL & UDDI	7
44	2.1 WSDL Data Model	7
45	2.1.1 portType	7
46	2.1.2 binding	7
47	2.1.3 service and port	7
48	2.1.4 import	7
49	2.2 UDDI Data Model	8
50	2.2.1 tModels	8
51	2.2.2 businessService & bindingTemplate	8
52	2.3 Mapping WSDL and UDDI	8
53	2.4 References to WSDL Components	9
54	2.5 Mapping WSDL 1.1 in UDDI V2	9
55	2.5.1 wsdl:portType → uddi:tModel	9
56	2.5.2 wsdl:binding → uddi:tModel	9
57	2.5.3 wsdl:service → uddi:businessService	10
58	2.5.4 wsdl:port → uddi:bindingTemplate	10
59	2.5.5 soap:address → uddi:accessPoint	10
60	2.6 Mapping WSDL 1.1 in UDDI V3	10
61	2.6.1 wsdl:portType → uddi:tModel	11
62	2.6.2 wsdl:binding → uddi:tModel	11
63	2.6.3 wsdl:service → uddi:businessService	11
64	2.6.4 wsdl:port → uddi:bindingTemplate	12
65	2.6.5 soap:address → uddi:accessPoint	12
66	3 A Complete Example	12
67	3.1 WSDL Sample	12
68	3.2 UDDI V2 Model	13
69	3.2.1 UDDI portType tModel	13
70	3.2.2 UDDI binding tModel	14
71	3.2.3 UDDI businessService and bindingTemplate	14
72	3.3 Sample V2 Queries	15
73	3.3.1 Find tModel for the portType StockQuotePortType in the namespace http://example.com/stockquote/	15
74	3.3.2 Find bindings for portType	15
75	3.3.3 Find Implementations of portType	16
76	3.3.4 Find implementations of binding	16
77	3.3.5 Find SOAP Implementations of portType	16
78	3.3.6 Find SOAP/HTTP Implementations of portType	16

79	3.3.7	Find the portType of a binding.....	16
80	3.4	UDDI V3 Model.....	16
81	3.4.1	UDDI portType tModel.....	16
82	3.4.2	UDDI binding tModel.....	17
83	3.4.3	UDDI businessService and bindingTemplate.....	17
84	3.5	Sample V3 Queries.....	18
85	3.5.1	Find tModel for the portType StockQuotePortType in the namespace http://example.com/stockquote/.....	18
86	3.5.2	Find bindings for portType.....	19
87	3.5.3	Find Implementations of portType.....	19
88	3.5.4	Find Implementations of binding.....	19
89	3.5.5	Find SOAP Implementations of portType.....	19
90	3.5.6	Find SOAP/HTTP Implementations of portType.....	19
91	3.5.7	Find the portType of a binding.....	20
92	4	References.....	20
93	 Appendix A: Canonical tModels	
94		20
95	A.1	WSDL Entity Type tModel.....	20
96	A.1.1	Design Goals.....	20
97	A.1.2	Definition.....	20
98	A.1.3	Values.....	20
99	A.1.4	Example of Use.....	21
100	A.2	WSDL Namespace tModel.....	21
101	A.2.1	Design Goals.....	21
102	A.2.2	Definition.....	21
103	A.2.3	Values.....	21
104	A.2.4	Example of Use.....	21
105	A.3	WSDL Local Name tModel.....	22
106	A.3.1	Design Goals.....	22
107	A.3.2	Definition.....	22
108	A.3.3	Values.....	22
109	A.3.4	Example of Use.....	22
110	A.4	WSDL portType Reference tModel.....	22
111	A.4.1	Design Goals.....	22
112	A.4.2	Definition.....	22
113	A.4.3	Values.....	23
114	A.4.4	Example of Use.....	23
115	A.5	SOAP Protocol tModel.....	23
116	A.5.1	Design Goals.....	23
117	A.5.2	Definition.....	23
118	A.5.3	Example of Use.....	24
119	A.6	SOAP over HTTP Transport tModel.....	24
120	A.6.1	Design Goals.....	24
121	A.6.2	Definition.....	24
122	A.6.3	Example of Use.....	24

123	Appendix B: Using XPointer in overviewURL	
124		25
125	B.1 XPointer Syntax.....		25
126			

127 1 Introduction

128 The Universal Description Discovery and Integration (UDDI) specification provides a platform-independent way of
129 describing and discovering Web services and Web services providers. The UDDI data structures provide a
130 framework for the description of basic service information, and architects an extensible mechanism to provide
131 detailed service access information using any standard description language. Many such languages exist in
132 specific industry domains and at different levels of the protocol stack. The Web Services Description Language
133 (WSDL) is a general purpose XML language for describing the interface, protocol bindings and the deployment
134 details of network services. WSDL complements the UDDI standard by providing a uniform way of describing the
135 abstract interface and protocol bindings of arbitrary network services. The purpose of this document is to clarify the
136 relationship between the two, describe how WSDL can be used to help create UDDI business service descriptions.
137 The importance of mapping WSDL consistently and thoroughly is critical to the utility of UDDI.

138 1.1 Goals and Requirements

139 The primary goals of this new mapping are to represent sufficient information from the WSDL documents to allow
140 the following types of queries without further recourse to the source WSDL documents, and to allow the appropriate
141 WSDL documents to be retrieved once a match has been found. Given that the source WSDL documents can be
142 distributed among the publishers using a UDDI registry, a UDDI registry provides a convenient central point where
143 such queries can be executed.

144 This new best practice would satisfy the following types of queries for both design-time and run-time discovery:

- 145 1) Given the namespace and/or local name of a wsdl:portType, find the tModel that represents that
146 portType.
- 147 2) Given a tModel representing a portType, find all tModels representing bindings for that portType.
- 148 3) Given a tModel representing a portType, find all bindingTemplates that represent implementations of
149 that portType.
- 150 4) Given a tModel representing a binding, find all bindingTemplates that represent implementations of that
151 binding.
- 152 5) Find all bindingTemplates that represent implementations of an extended binding, for example all
153 SOAP/HTTP implementations, of a portType.

154 Some aspects of the mapping allow information to be retrieved directly without further queries being necessary.
155 For example, given the tModel representing a binding, it is possible to retrieve the key of the tModel representing
156 the portType that the binding refers to. Other aspects of the mapping may require multiple queries to be issued to
157 UDDI.

158 A further goal is to allow deployment WSDL to be generated. With this new mapping, it is possible to generate a
159 wsdl:service element directly from the mapping in UDDI.

160 1.2 Relationship to Version 1 Best Practice

161 This document builds on *Using WSDL in a UDDI Registry, Version 1.07* [2], providing an expanded modeling
162 practice which encompasses the flexibility of WSDL.

163 As a Technical Note, this document does not replace the existing Best Practice. If the additional flexibility is not
164 required the existing Best Practice can continue to be used, particularly when the UDDI model is published
165 manually.

166 It is anticipated that implementations of the approach described in this Technical Note will be developed and that
167 once experience with those implementations is obtained this Technical Note will become a Best Practice.

168 A final goal is to be compatible with the existing Best Practice in that a model published using the approach
169 described in this document should be usable by a client that uses the Version 1 Best Practice approach.

170 **1.3 tModel Keys**

171 The tModel keys presented in this draft document are fictitious. When this document is approved the real tModels
172 will be published to the UDDI Business Registry and the tModel keys in this document will be updated to the real
173 values.

174 **2 Rationalizing Two Data Models: WSDL & UDDI**

175 A brief discussion of the two respective data models, WSDL and UDDI, follows. For a complete explanation of the
176 these specifications, see [1] and [3].

177 **2.1 WSDL Data Model**

178 A review of WSDL in the context of the goals and requirements will help guide a new mapping practice in UDDI.

179 **2.1.1 portType**

180 The central construct in WSDL is the portType. A portType is an abstract collection of operations that may be
181 supported by one or more Web services. A WSDL portType defines these operations in terms of message
182 definitions which usually rely on the XML Schema language to describe the representation of each message. A
183 single WSDL file may contain multiple portType entities. Each portType is uniquely identified by the combination of
184 its local name and the target namespace of the definitions element that contains the portType.

185 WSDL portTypes may be "implemented" by more than one Web service. Web services that purport to support a
186 given portType must adhere not only to the message formats that are part of the WSDL definition, they must also
187 adhere to the semantic agreement that is implicitly part of the portType. This allows applications to treat two web
188 services as substitutable if and only if they implement a common portType.

189 **2.1.2 binding**

190 WSDL portTypes are defined independently from the transport protocol used to transmit the messages. To allow
191 transport protocol details to be expressed in WSDL, one must define a second construct, known as a binding. A
192 WSDL binding adds transport specifics to a particular WSDL portType. Again, a single WSDL file may contain
193 multiple bindings. A WSDL binding specifies its portType through a QName reference. The portType
194 implemented by a binding may or may not be in the same target namespace as the binding itself. Like a portType,
195 a binding is uniquely identified by the combination of its local name and the target namespace of the definitions
196 element that contains the binding.

197 **2.1.3 service and port**

198 Finally, WSDL defines a service as a collection of named ports, each of which is bound to a particular portType
199 through a named binding. A service may expose multiple ports in order to make a single portType available over
200 multiple protocols. Or, a service may expose multiple ports in order to expose more than one portType from a
201 single logical entity. A WSDL port specifies the binding it implements through a QName reference.

202 **2.1.4 import**

203 The import directive in WSDL allows the separation of these different entities into multiple files. As such, a WSDL
204 file may be composed of a single portType, multiple portTypes, a single binding that imports its portType definition,
205 multiple bindings, a single service or multiple services, etc. The WSDL data model provides great flexibility in
206 terms of composition and reusability of WSDL entities.

207 Given this, the critical components of a WSDL file in terms of composition and identity are the target namespace of
208 the definitions element and the local names which identify each portType, binding, service and port within the target
209 namespace.

210 2.2 UDDI Data Model

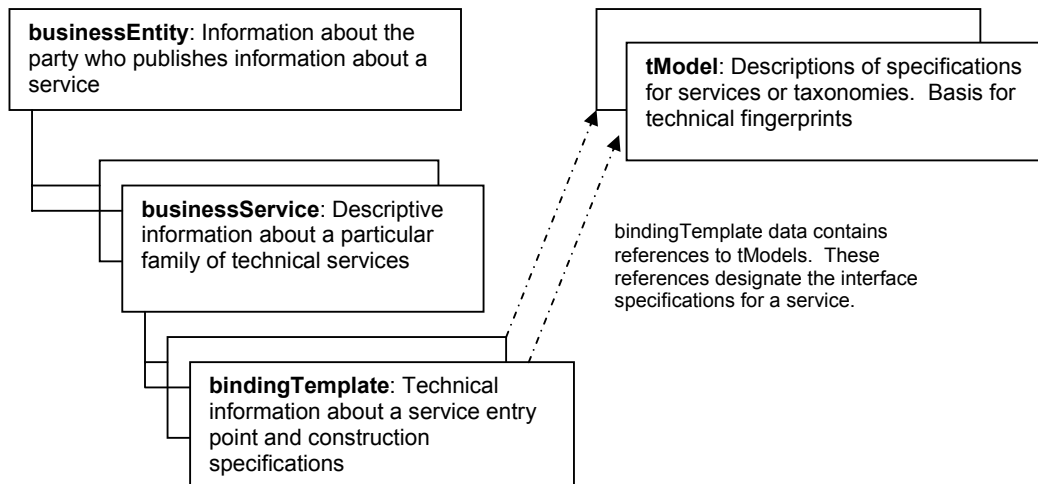
211 As an aid to understanding the sections ahead, we provide here a brief overview of two UDDI data structures that
 212 are particularly relevant to the use of WSDL in the context of a UDDI registry: the tModel, also known as the service
 213 type definition, and the businessService.

214 2.2.1 tModels

215 TModels represent unique concepts or constructs and provide the ability to describe compliance with a
 216 specification, a concept, or a shared design. TModels have various uses in the UDDI registry. In the case of
 217 mapping WSDL-described Web services, tModels are used to represent technical specifications like wire protocols,
 218 interchange formats and sequencing rules. When a particular specification is registered with the UDDI repository as
 219 a tModel, it is assigned a unique key, which is then used in the description of service instances to indicate
 220 compliance with the specification.

221 Each tModel contains an overviewURL which provides an address where the specification itself can be retrieved,
 222 for example, a WSDL file.

223 TModels can be decorated with searchable metadata by associating identifierBags and categoryBags with that
 224 tModel. These bags contain keyedReferences: name/value pairs associated with a unique identifier. This
 225 metadata can be used to search for these tModels through UDDI Inquiry API calls.



226 2.2.2 businessService & bindingTemplate

227 Services themselves are represented in UDDI by the businessService data structure, and the details of how and
 228 where the service is accessed are provided by one or more bindingTemplate structures. The businessService
 229 might be thought of as a logical container of services. The bindingTemplate structure contains the accessPoint of
 230 the service itself, as well as references to the tModels it is said to implement.

231 2.3 Mapping WSDL and UDDI

232 To represent information about a WSDL file (such as target namespace, QName relationships and resource
 233 locations) so that it is consistently described and discovered requires a mapping practice that acknowledges the
 234 variety and flexibility within the WSDL model itself. This can be accomplished by a comprehensive use of UDDI's
 235 key entity structures (tModels) and typed meta-data construct (keyedReferences).

236 By representing WSDL portType and binding entities as UDDI tModels, one can accurately represent the building
 237 blocks of "abstract" WSDL interface information in UDDI. Similarly, by representing WSDL service and port as,
 238 respectively, UDDI businessService and bindingTemplate, one achieves a similar parallel. Moreover, by
 239 representing the target namespace and dependencies of a WSDL entity as keyedReferences on the UDDI entities,
 240 one can represent strongly typed metadata about the WSDL entity within UDDI.

241 There are two important things to note about this mapping, especially as compared to the Version 1 *Using WSDL in*
 242 *a UDDI Registry Best Practice*.

243 First, this new approach means that a single WSDL file may be mapped to multiple tModels. For example, a WSDL
244 file located at <http://www.example.com/example.wsdl> which contains a single portType definition and 2 binding
245 definitions will map to three distinct tModels in UDDI. This decision differs significantly from the Version 1 Best
246 Practice, which always mapped the entirety of a WSDL file to a single tModel. The rationale for this new mapping
247 decision is allows for the modularity of WSDL to be expressed through UDDI. By decomposing WSDL into multiple
248 tModels, one can accurately model in UDDI exactly which portTypes and bindings a given Web service supports,
249 as opposed to being constrained to asserting that a Web service always supports the entirety of the WSDL file,
250 which may not be the case.

251 While there is an increased amount of data from a WSDL file modeled in UDDI, this new approach is in accord with
252 the original Best Practice in that it does not attempt to use UDDI as a repository for *all* of the data in a WSDL file.
253 Just as in the Version 1 Best Practice, one still must go outside of the UDDI registry to retrieve the portType and
254 binding information necessary for software applications to work with that Web service.

255 **2.4 References to WSDL Components**

256 As part of mapping WSDL in UDDI it is necessary to refer to various components in the WSDL document(s).
257 These references occur as overviewURL values. As noted above, in this mapping several tModels can refer to the
258 same WSDL file. The particular WSDL component SHOULD be determined by using the metadata contained
259 within the tModel's categoryBag. Alternatively, the overviewURL value MAY contain a fragment identifier which
260 identifies the particular WSDL component, but this is not required. If the optional fragment identifier is used then
261 the syntax described in Appendix B MUST be used.

262 **2.5 Mapping WSDL 1.1 in UDDI V2**

263 **2.5.1 wsdl:portType → uddi:tModel**

264 A wsdl:portType MUST be modeled as a uddi:tModel.

265 The uddi:name element of the tModel MUST be the value of the name attribute of the portType.

266 The tModel MUST contain a categoryBag and the categoryBag MUST contain at least two keyedReferences:

- 267 1. A keyedReference with a tModelKey of the WSDL Namespace tModel and a keyValue of the
268 target namespace of the definitions element that contains the portType.¹
- 269 2. A keyedReference with a tModelKey of the WSDL Entity Type tModel and a keyValue of
270 "portType".

271 The tModel MUST contain an overviewDoc with an overviewURL containing the location of the WSDL file.

272 **2.5.2 wsdl:binding → uddi:tModel**

273 A wsdl:binding MUST be modeled as a uddi:tModel.

274 The uddi:name element of the tModel MUST be the value of the name attribute of the binding.

275 The tModel MUST contain a categoryBag and the categoryBag MUST contain at least three keyedReferences:

- 276 1. A keyedReference with a tModelKey of the WSDL Entity Type tModel and a keyValue of "binding".
- 277 2. A keyedReference with a tModelKey of the WSDL Namespace tModel and a keyValue of the
278 target namespace of the definitions element that contains the binding.
- 279 3. A keyedReference with a tModelKey of the portType tModel and a keyValue of the tModelKey that
280 models the portType to which the binding relates.

281 The tModel categoryBag MAY contain an additional keyedReference for the uddi.org types taxonomy of wsdlSpec
282 for backward compatibility².

¹ WSDL 1.1 does not require the usage of a targetNamespace, but such a practice is not recommended. In the event that a WSDL file without a targetNamespace is registered in UDDI, it will not have a WSDL Namespace keyedReference and queries for these tModels based solely on the tModel name could return multiple results because no namespace can be specified.

² By categorizing a wsdl:binding tModel according to the Version 1 UDDI/WSDL Best Practice, backward compatibility is maintained. However, wsdl:portType tModels should not be categorized with this designation, as the wsdl:portType tModel will not contain sufficient information to compose a complete WSDL binding.

283 The tModel MUST contain an overviewDoc with an overviewURL containing the location of the WSDL file.

284 **2.5.3 wsdl:service → uddi:businessService**

285 A wsdl:service MUST be modeled as a uddi:businessService. An existing businessService MAY be used or a new
286 businessService MAY be created.

287 If a new businessService is created, the uddi:name of this businessService MUST be the value of the name
288 attribute of the wsdl:service³.

289 The bindingTemplates of the businessService MUST include bindingTemplates that model the ports of the service,
290 as described in the following sections.

291 The businessService MUST contain a categoryBag and the categoryBag MUST contain three keyedReferences:

- 292 1. A keyedReference with a tModelKey of the WSDL Entity Type tModel and a keyValue of "service".
- 293 2. A keyedReference with a tModelKey of the WSDL Namespace tModel and a keyValue of the
294 target namespace of the definitions element that contains the service.
- 295 3. A keyedReference with a tModelKey of the WSDL Local Name tModel and a keyValue that is the
296 value of the name attribute of the wsdl service.

297 **2.5.4 wsdl:port → uddi:bindingTemplate**

298 A wsdl :port MUST be modeled as a uddi:bindingTemplate.

299 The bindingTemplate tModelInstanceDetails element MUST contain the following tModelInstanceInfo elements:

- 300 1. A tModelInstanceInfo with a tModelKey value of the key of the tModel that models the wsdl:binding
301 that this port implements. The instanceParms of this tModelInstanceInfo MUST contain the
302 wsdl:port local name.
- 303 2. A tModelInstanceInfo with a tModelKey value of the key of the tModel that models the portType⁴.

304 If the wsdl:binding contains a soap:binding then the bindingTemplate MUST include the following additional
305 tModelInstanceInfo elements:

- 306 1. A tModelInstanceInfo with a tModelKey value of the key of the standard tModel that represents the
307 SOAP extension to WSDL.
- 308 2. A tModelInstanceInfo with a tModelKey value of the key of the standard tModel that represents the
309 HTTP transport for SOAP.

310 Other binding extensibility elements are handled in a similar fashion.

311 **2.5.5 soap:address → uddi:accessPoint**

312 A soap:address MUST be modeled as a uddi:accessPoint in the bindingTemplate that models the wsdl:port that
313 contains the soap:address.

314 The value of the URLType attribute of the accessPoint MUST correspond to the transport specified by the
315 soap:binding, or "other" if no correspondence exists. In the case of the HTTP transport for example, the value of
316 the URLType attribute MUST be "http".

317 The value of the accessPoint MUST be the value of the location attribute of the soap:address.

318 **2.6 Mapping WSDL 1.1 in UDDI V3**

319 The differences in UDDI data structures that are relevant are:

- 320 1. A bindingTemplate can have a categoryBag.

³ Because an existing businessService could be used and it is unlikely in this case that the businessService name is the same as the WSDL service name, the keyedReference in the categoryBag MUST be used to retrieve the WSDL service name, not the businessServiceName.

⁴ While it may seem redundant to create tModelInstanceInfos for both portType and binding, doing so allows one to query for services based on known portTypes, as opposed to just known bindings.

- 321 2. An accessPoint has a useType attribute not a URLType attribute and one of the values of useType
322 is wsdlDeployment which indicates that the address information should be retrieved from the
323 referenced WSDL document.
- 324 3. An overviewURL now has an optional useType attribute and a standard value of “wsdlInterface”
325 has been defined to indicate “an abstract interface document”. This mapping assumes that
326 “wsdlInterface” is used with tModels that represent both portTypes and bindings.

327 **2.6.1 wsdl:portType → uddi:tModel**

328 A wsdl:portType MUST be modeled as a uddi:tModel.

329 The name of the tModel MUST be the value of the name attribute of the portType.

330 The tModel MUST contain a categoryBag and the categoryBag MUST contain at least the following
331 keyedReferences:

- 332 1. A keyedReference with a tModelKey of the WSDL Namespace tModel and a keyValue of the
333 target namespace of the definitions element that contains the portType.
- 334 2. A keyedReference with a tModelKey of the WSDL Entity Type tModel and a keyValue of
335 “portType”.

336 The tModel MUST contain an overviewDoc with an overviewURL containing the location of the WSDL file. The
337 value of the useType attribute of the overviewURL MUST be “wsdlInterface”.

338 **2.6.2 wsdl:binding → uddi:tModel**

339 A wsdl:binding MUST be modeled as a uddi:tModel.

340 The name of the tModel MUST be the value of the name attribute of the binding.

341 The tModel MUST contain a categoryBag and the categoryBag MUST contain at least the following
342 keyedReferences:

- 343 1. A keyedReference with a tModelKey of the WSDL Entity Type tModel and a keyValue of “binding”.
- 344 2. A keyedReference with a tModelKey of the WSDL Namespace tModel and a keyValue of the
345 target namespace of the definitions element that contains the binding.
- 346 3. A keyedReference with a tModelKey of the portType tModel and a keyValue of the tModel that
347 models the portType to which the binding relates.

348 The tModel categoryBag MAY contain an additional keyedReference for the uddi.org types taxonomy of wsdlSpec
349 for backward compatibility.

350 The tModel MUST contain an overviewDoc with an overviewURL containing the location of the WSDL file. The
351 value of the useType attribute of the overviewURL MUST be “wsdlInterface”.

352 **2.6.3 wsdl:service → uddi:businessService**

353 A wsdl:service MUST be modeled as a uddi:businessService. An existing businessService MAY be used or a new
354 businessService MAY be created.

355 If a new businessService is created, the uddi:name of this businessService MAY be the value of the name attribute
356 of the wsdl:service.

357 The bindingTemplates of the businessService MUST include bindingTemplates that model the ports of the service,
358 as described in the following sections.

359 The businessService MUST contain a categoryBag and the categoryBag MUST contain the following
360 keyedReferences:

- 361 1. A keyedReference with a tModelKey of the WSDL Entity Type tModel and a keyValue of “service”.
- 362 2. A keyedReference with a tModelKey of the WSDL Namespace tModel and a keyValue of the
363 target namespace.
- 364 3. A keyedReference with a tModelKey of the WSDL Local Name tModel and a keyValue that is the
365 value of the name attribute of the wsdl service.

366 2.6.4 wsdl:port → uddi:bindingTemplate

367 A wsdl:port MUST be modeled as a uddi:bindingTemplate.

368 The bindingTemplate tModelInstanceDetails element MUST contain the following tModelInstanceInfo elements:

- 369 1. A tModelInstanceInfo with a tModelKey value of the key of the tModel that models the wsdl:binding
370 that this port implements.
- 371 2. A tModelInstanceInfo with a tModelKey value of the key of the tModel that models the portType.

372 If the wsdl:binding contains a soap:binding then the bindingTemplate MUST include the following additional
373 tModelInstanceInfo elements:

- 374 1. A tModelInstanceInfo with a tModelKey value of the key of the standard tModel that represents the
375 SOAP binding.
- 376 2. A tModelInstanceInfo with a tModelKey value of the key of the standard tModel that represents the
377 HTTP transport.

378 Other binding extensibility elements are handled in a similar fashion.

379 The bindingTemplate MUST contain a categoryBag and the categoryBag MUST contain the following
380 keyedReferences:

- 381 1. A keyedReference with a tModelKey of the WSDL Entity Type tModel and a keyValue of "port".
- 382 2. A keyedReference with a tModelKey of the WSDL Namespace tModel and a keyValue of the target
383 namespace of the definitions element that contains the port.
- 384 3. A keyedReference with a tModelKey of the WSDL Local Name tModel and a keyValue of the local
385 name of the port.⁵

386 2.6.5 soap:address → uddi:accessPoint

387 A soap:address MUST be modeled as a uddi:accessPoint in the bindingTemplate that models the wsdl:port that
388 contains the soap:address.

389 The useType of the accessPoint SHOULD be set to endPoint and the content of the accessPoint SHOULD be the
390 value of the location attribute of the soap:address. Alternatively, the useType of the accessPoint MAY be set to
391 wsdlDeployment and the content of the accessPoint SHOULD be a reference to the WSDL file containing the
392 address information.

393 3 A Complete Example

394 Consider the following sample based on the WSDL file presented in the WSDL 1.1 specification. This sample
395 shows how a single WSDL file is decomposed into two tModels (one for the portType and one for the binding) and
396 one businessService with one bindingTemplate. It then shows the kinds of UDDI API queries that can be used for
397 the purposes of discovery.

398 3.1 WSDL Sample

```

399     <?xml version="1.0" encoding="utf-8" ?>
400     <definitions
401         name="StockQuote"
402         targetNamespace="http://example.com/stockquote/"
403         xmlns:tns="http://example.com/stockquote/"
404         xmlns:xsd="http://example.com/stockquote/schema/"
405         xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
406         xmlns="http://schemas.xmlsoap.org/wsdl/">
407         <import
408             namespace="http://example.com/stockquote/schema/"
409             location="http://location/schema.xsd" />

```

⁵ Note that in the v3 modeling, the local name of the port is captured in a keyedReference, whereas in v2, it was captured in the tModelInstanceInfo instanceParms.

```

410     <message name="GetLastTradePriceInput">
411         <part name="body" element="xsd1:TradePriceRequest" />
412     </message>
413     <message name="GetLastTradePriceOutput">
414         <part name="body" element="xsd1:TradePrice" />
415     </message>
416     <portType name="StockQuotePortType">
417         <operation name="GetLastTradePrice">
418             <input message="tns:GetLastTradePriceInput" />
419             <output message="tns:GetLastTradePriceOutput" />
420         </operation>
421     </portType>
422     <binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
423         <soap:binding style="document"
424             transport="http://schemas.xmlsoap.org/soap/http" />
425         <operation name="GetLastTradePrice">
426             <soap:operation soapAction="http://example.com/GetLastTradePrice" />
427             <input>
428                 <soap:body use="literal" />
429             </input>
430             <output>
431                 <soap:body use="literal" />
432             </output>
433         </operation>
434     </binding>
435
436     <service name="StockQuoteService">
437         <documentation>My first service</documentation>
438         <port name="StockQuotePort" binding="tns:StockQuoteBinding">
439             <soap:address location="http://location/sample"/>
440         </port>
441     </service>
442
443 </definitions>

```

444 Note that this wsdl file has one portType, one binding, one service and one port. As such, this sample represents
445 the simplest wsdl file. Also note that the location of this WSDL is at <http://location/sample.wsdl>.

446 3.2 UDDI V2 Model

447 3.2.1 UDDI portType tModel

```

448     <tModel tModelKey="uuid:e8cf1163-8234-4b35-865f-94a7322e40c3" >
449         <name>
450             StockQuotePortType
451         </name>
452         <categoryBag>
453             <keyedReference
454                 tModelKey="uuid:c0ec5422-44e9-4774-b2f1-5a89eca389f3"
455                 keyValue="http://example.com/stockquote/"
456             />
457             <keyedReference
458                 tModelKey="uuid:7950ce8d-259a-4a9a-8f13-e51632fe12d5"
459                 keyValue="portType"
460             />
461         </categoryBag>
462         <overviewDoc>
463             <overviewURL>
464                 http://location/sample.wsdl
465             </overviewURL>
466         </overviewDoc>
467     </tModel>

```

468 3.2.2 UDDI binding tModel

```

469     <tModel tModelKey="uuid:49662926-f4a5-4ba5-b8d0-32ab388dadda">
470         <name>
471             StockQuoteSoapBinding
472         </name>
473         <categoryBag>
474             <keyedReference
475                 tModelKey="uuid:c0ec5422-44e9-4774-b2f1-5a89eca389f3"
476                 keyValue="http://example.com/stockquote/"
477             />
478             <keyedReference
479                 tModelKey="uuid:7950ce8d-259a-4a9a-8f13-e51632fe12d5"
480                 keyValue="binding"
481             />
482             <keyedReference
483                 tModelKey="uuid:acc0db62-2866-4081-8986-d62a7bf75ab3"
484                 keyValue="uuid:e8cf1163-8234-4b35-865f-94a7322e40c3"
485             />
486             <keyedReference
487                 tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
488                 keyValue="wsdlSpec"
489             />
490         </categoryBag>
491         <overviewDoc>
492             <overviewURL>
493                 http://location/sample.wsdl
494             </overviewURL>
495         </overviewDoc>
496     </tModel>

```

497 Note how this tModel creates a link to the portType tModel through placing the tModelKey of the portType tModel
498 above as its keyValue.

499 3.2.3 UDDI businessService and bindingTemplate

```

500     <businessService
501         serviceKey="102b114a-52e0-4af4-a292-02700da543d4"
502         businessKey="1e65ea29-4e0f-4807-8098-d352d7b10368">
503         <name>StockQuoteService</name>
504         <bindingTemplates>
505             <bindingTemplate
506                 bindingKey="f793c521-0daf-434c-8700-0e32da232e74"
507                 serviceKey="102b114a-52e0-4af4-a292-02700da543d4">
508                 <accessPoint URLType="http">
509                     http://location/sample
510                 </accessPoint>
511                 <tModelInstanceDetails>
512                     <tModelInstanceInfo
513                         tModelKey="uuid:454ef824-994f-4ce3-a892-3d97ad952707">
514                         <description xml:lang="en">
515                             This tModel represents the SOAP binding
516                             extensibility element.
517                         </description>
518                     <tModelInstanceInfo
519                         tModelKey="uuid:26de470c-0dab-4471-89f9-57450a524005">
520                         <description xml:lang="en">
521                             This tModel represents the HTTP transport using SOAP.
522                         </description>
523                     </tModelInstanceInfo>
524                     <tModelInstanceInfo
525                         tModelKey="uuid:49662926-f4a5-4ba5-b8d0-32ab388dadda">
526                         <description xml:lang="en">
527                             This tModel key represents the wsdl:binding that this

```



```

528         wsdl:port implements.
529     </description>
530     <instanceDetails>
531         <instanceParms>StockQuotePort</instanceParms>
532     </instanceDetails>
533 </tModelInstanceInfo>
534 <tModelInstanceInfo
535     tModelKey="uuid:e8cf1163-8234-4b35-865f-94a7322e40c3">
536     <description xml:lang="en">
537         This tModel represents the wsdl:portType that
538         this wsdl:port implements.
539     </description>
540 </tModelInstanceInfo>
541 </tModelInstanceDetails>
542 </bindingTemplate>
543 </bindingTemplates>
544 <categoryBag>
545     <keyedReference
546         tModelKey="uuid:7950cc8d-259a-4a9a-8f13-e51632fe12d5"
547         keyValue="service"
548     />
549     <keyedReference
550         tModelKey="uuid:c0ec5422-44e9-4774-b2f1-5a89eca389f3"
551         keyValue="http://example.com/stockquote/"
552     />
553     <keyedReference
554         tModelKey="uuid:248ccb56-6365-446d-847a-91171ae79740"
555         keyValue="StockQuoteService"
556     />
557 </categoryBag>
558 </businessService>

```

559 3.3 Sample V2 Queries

560 This section shows how to perform various UDDI V2 queries given the model of the example.

561 3.3.1 Find tModel for the portType StockQuotePortType in the namespace 562 <http://example.com/stockquote/>

```

563 <find_tModel generic="2.0" xmlns="urn:uddi-org:api_v2">
564     <name>StockQuotePortType</name>
565     <categoryBag>
566         <keyedReference tModelKey="uuid:7950ce8d-259a-4a9a-8f13-e51632fe12d5"
567         keyValue="portType"/>
568         <keyedReference tModelKey="uuid:c0ec5422-44e9-4774-b2f1-5a89eca389f3"
569         keyValue="http://example.com/stockquote/" />
570     </categoryBag>
571 </find_tModel>

```

572 This should return the tModelKey uuid:e8cf1163-8234-4b35-865f-94a7322e40c3.

573 3.3.2 Find bindings for portType

```

574 <find_tModel generic="2.0" xmlns="urn:uddi-org:api_v2">
575     <categoryBag>
576         <keyedReference tModelKey="uuid:7950ce8d-259a-4a9a-8f13-e51632fe12d5"
577         keyValue="binding"/>
578         <keyedReference tModelKey="uuid:acc0db62-2866-4081-8986-d62a7bf75ab3"
579         keyValue="uuid:e8cf1163-8234-4b35-865f-94a7322e40c3"/>
580     </categoryBag>
581 </find_tModel>

```

582 This should return the tModelKey uuid:49662926-f4a5-4ba5-b8d0-32ab388dadda.

583 3.3.3 Find Implementations of portType

```

584     <find_binding generic="2.0" xmlns="urn:uddi-org:api_v2">
585         <tModelBag>
586             <tModelKey>uuid:e8cf1163-8234-4b35-865f-94a7322e40c3</tModelKey>
587         </tModelBag>
588     </find_binding>

```

589 This should return the bindingKey f793c521-0daf-434c-8700-0e32da232e74.

590 3.3.4 Find implementations of binding

```

591     <find_binding generic="2.0" xmlns="urn:uddi-org:api_v2">
592         <tModelBag>
593             <tModelKey>uuid:49662926-f4a5-4ba5-b8d0-32ab388dadda</tModelKey>
594         </tModelBag>
595     </find_binding>

```

596 This should return the bindingKey f793c521-0daf-434c-8700-0e32da232e74.

597 3.3.5 Find SOAP Implementations of portType

```

598     <find_binding generic="2.0" xmlns="urn:uddi-org:api_v2">
599         <tModelBag>
600             <tModelKey>uuid:e8cf1163-8234-4b35-865f-94a7322e40c3</tModelKey>
601             <tModelKey>uuid:454ef824-994f-4ce3-a892-3d97ad952707</tModelKey>
602         </tModelBag>
603     </find_binding>

```

604 This should return the bindingKey f793c521-0daf-434c-8700-0e32da232e74.

605 3.3.6 Find SOAP/HTTP Implementations of portType

```

606     <find_binding generic="2.0" xmlns="urn:uddi-org:api_v2">
607         <tModelBag>
608             <tModelKey>uuid:e8cf1163-8234-4b35-865f-94a7322e40c3</tModelKey>
609             <tModelKey>uuid:454ef824-994f-4ce3-a892-3d97ad952707</tModelKey>
610             <tModelKey>uuid:26de470c-0dab-4471-89f9-57450a524005</tModelKey>
611         </tModelBag>
612     </find_binding>

```

613 This should return the bindingKey f793c521-0daf-434c-8700-0e32da232e74.

614 3.3.7 Find the portType of a binding

615 This is simply the keyValue of the keyedReference in the categoryBag with tModelKey="uuid:acc0db62-2866-4081-
616 8986-d62a7bf75ab3" and therefore no query is required once the tModel of the binding is obtained.

617 3.4 UDDI V3 Model

618 3.4.1 UDDI portType tModel

```

619     <tModel tModelKey="uddi:e8cf1163-8234-4b35-865f-94a7322e40c3" >
620         <name>
621             StockQuotePortType
622         </name>
623         <categoryBag>
624             <keyedReference
625                 tModelKey="uddi:c0ec5422-44e9-4774-b2f1-5a89eca389f3"
626                 keyValue="http://example.com/stockquote/"
627             />
628             <keyedReference
629                 tModelKey="uddi:7950ce8d-259a-4a9a-8f13-e51632fe12d5"
630                 keyValue="portType"

```



```

631         />
632     </categoryBag>
633     <overviewDoc>
634         <overviewURL useType="wsdlInterface">
635             http://location/sample.wsdl
636         </overviewURL>
637     </overviewDoc>
638 </tModel>

```

639 3.4.2 UDDI binding tModel

```

640     <tModel tModelKey="uddi:49662926-f4a5-4ba5-b8d0-32ab388dadda">
641         <name>
642             StockQuoteBinding
643         </name>
644         <categoryBag>
645             <keyedReference
646                 tModelKey="uddi:c0ec5422-44e9-4774-b2f1-5a89eca389f3"
647                 keyValue="http://example.com/stockquote/"
648             />
649             <keyedReference
650                 tModelKey="uddi:7950ce8d-259a-4a9a-8f13-e51632fe12d5"
651                 keyValue="binding"
652             />
653             <keyedReference
654                 tModelKey="uddi:acc0db62-2866-4081-8986-d62a7bf75ab3"
655                 keyValue="uddi:e8cf1163-8234-4b35-865f-94a7322e40c3"
656             />
657             <keyedReference
658                 tModelKey="uddi:c1acf26d-9672-4404-9d70-39b756e62ab4"
659                 keyValue="wsdlSpec"
660             />
661         </categoryBag>
662         <overviewDoc>
663             <overviewURL useType="wsdlInterface">
664                 http://location/sample.wsdl
665             </overviewURL>
666         </overviewDoc>
667     </tModel>

```

668 Note how this tModel creates a link to the portType tModel through placing the tModelKey of the portType tModel
669 above as its keyValue.

670 3.4.3 UDDI businessService and bindingTemplate

```

671     <businessService
672         serviceKey="uddi:102b114a-52e0-4af4-a292-02700da543d4"
673         businessKey="uddi:1e65ea29-4e0f-4807-8098-d352d7b10368">
674         <name>StockQuoteService</name>
675         <bindingTemplates>
676             <bindingTemplate
677                 bindingKey="uddi:f793c521-0daf-434c-8700-0e32da232e74"
678                 serviceKey="uddi:102b114a-52e0-4af4-a292-02700da543d4">
679                 <accessPoint useType="endPoint">
680                     http://location/sample
681                 </accessPoint>
682                 <tModelInstanceDetails>
683                     <tModelInstanceInfo
684                         tModelKey="uddi:454ef824-994f-4ce3-a892-3d97ad952707">
685                         <description xml:lang="en">
686                             This tModel represents the SOAP binding
687                             extensibility element.
688                         </description>
689                     </tModelInstanceInfo
690                         tModelKey="uddi:26de470c-0dab-4471-89f9-57450a524005">

```

```

691         <description xml:lang="en">
692             This tModel represents the HTTP transport using SOAP.
693         </description>
694     </tModelInstanceInfo>
695     <tModelInstanceInfo
696         tModelKey="uddi:49662926-f4a5-4ba5-b8d0-32ab388dadda">
697         <description xml:lang="en">
698             This tModel key represents the wsdl:binding that this
699             wsdl:port implements.
700         </description>
701     </tModelInstanceInfo>
702     <tModelInstanceInfo
703         tModelKey="uddi:e8cf1163-8234-4b35-865f-94a7322e40c3">
704         <description xml:lang="en">
705             This tModel represents the wsdl:portType that
706             this wsdl:port implements.
707         </description>
708     </tModelInstanceInfo>
709 </tModelInstanceDetails>
710 <categoryBag>
711     <keyedReference
712         tModelKey="uddi:7950cc8d-259a-4a9a-8f13-e51632fe12d5"
713         keyValue="port"
714     />
715     <keyedReference
716         tModelKey="uddi:c0ec5422-44e9-4774-b2f1-5a89eca389f3"
717         keyValue="http://example.com/stockquote/"
718     />
719     <keyedReference
720         tModelKey="uddi:248ccb56-6365-446d-847a-91171ae79740"
721         keyValue="StockQuotePort"
722     />
723 </categoryBag>
724 </bindingTemplate>
725 </bindingTemplates>
726 <categoryBag>
727     <keyedReference
728         tModelKey="uddi:7950cc8d-259a-4a9a-8f13-e51632fe12d5"
729         keyValue="service"
730     />
731     <keyedReference
732         tModelKey="uddi:c0ec5422-44e9-4774-b2f1-5a89eca389f3"
733         keyValue="http://example.com/stockquote/"
734     />
735     <keyedReference
736         tModelKey="uddi:248ccb56-6365-446d-847a-91171ae79740"
737         keyValue="StockQuoteService"
738     />
739 </categoryBag>
740 </businessService>

```

741 3.5 Sample V3 Queries

742 This section shows how to perform various UDDI V3 queries given the model of the example.

743 3.5.1 Find tModel for the portType StockQuotePortType in the namespace 744 <http://example.com/stockquote/>

```

745     <find_tModel xmlns="urn:uddi-org:api_v3">
746         <name>StockQuotePortType</name>
747         <categoryBag>
748             <keyedReference tModelKey="uddi:7950ce8d-259a-4a9a-8f13-e51632fe12d5"
749                 keyValue="portType"/>

```

```

750         <keyedReference tModelKey="uddi:5165f13f-139c-4649-adf8-89f66e437f55"
751         keyValue="http://example.com/stockquote/" />
752     </categoryBag>
753 </find_tModel>

```

754 This should return the tModelKey uddi:e8cf1163-8234-4b35-865f-94a7322e40c3.

755 3.5.2 Find bindings for portType

```

756     <find_tModel xmlns="urn:uddi-org:api_v3">
757         <categoryBag>
758             <keyedReference tModelKey="uddi:7950ce8d-259a-4a9a-8f13-e51632fe12d5"
759             keyValue="binding" />
760             <keyedReference tModelKey="uddi:acc0db62-2866-4081-8986-d62a7bf75ab3"
761             keyValue="uddi:e8cf1163-8234-4b35-865f-94a7322e40c3" />
762         </categoryBag>
763     </find_tModel>

```

764 This should return the tModelKey uddi:49662926-f4a5-4ba5-b8d0-32ab388dadda.

765 3.5.3 Find Implementations of portType

```

766     <find_binding xmlns="urn:uddi-org:api_v3">
767         <tModelBag>
768             <tModelKey>uddi:e8cf1163-8234-4b35-865f-94a7322e40c3</tModelKey>
769         </tModelBag>
770     </find_binding>

```

771 This should return the bindingKey f793c521-0daf-434c-8700-0e32da232e74.

772 3.5.4 Find Implementations of binding

```

773     <find_binding xmlns="urn:uddi-org:api_v3">
774         <tModelBag>
775             <tModelKey>uddi:49662926-f4a5-4ba5-b8d0-32ab388dadda</tModelKey>
776         </tModelBag>
777     </find_binding>

```

778 This should return the bindingKey f793c521-0daf-434c-8700-0e32da232e74.

779 3.5.5 Find SOAP Implementations of portType

```

780     <find_binding xmlns="urn:uddi-org:api_v3">
781         <tModelBag>
782             <tModelKey>uddi:e8cf1163-8234-4b35-865f-94a7322e40c3</tModelKey>
783             <tModelKey>(key of tModel representing SOAP extensibility
784             element)</tModelKey>
785         </tModelBag>
786     </find_binding>

```

787 This should return the bindingKey f793c521-0daf-434c-8700-0e32da232e74.

788 3.5.6 Find SOAP/HTTP Implementations of portType

```

789     <find_binding xmlns="urn:uddi-org:api_v3">
790         <tModelBag>
791             <tModelKey>uddi:e8cf1163-8234-4b35-865f-94a7322e40c3</tModelKey>
792             <tModelKey>(key of tModel representing SOAP extensibility
793             element)</tModelKey>
794             <tModelKey>uddi:1be8abb8-86e3-4ee5-99ac-b7fffbbed8110</tModelKey>
795         </tModelBag>
796     </find_binding>

```

797 This should return the bindingKey uddi:f793c521-0daf-434c-8700-0e32da232e74.

798 3.5.7 Find the portType of a binding

799 This is simply the keyValue of the keyedReference in the categoryBag with tModelKey="uddi:acc0db62-2866-4081-
800 8986-d62a7bf75ab3" and therefore no query is required once the tModel of the binding is obtained.

801 4 References

- 802 1. UDDI Version 2.0 Data Structure Reference, July 7, 2002. Available at [http://uddi.org/pubs/DataStructure-V2.03-](http://uddi.org/pubs/DataStructure-V2.03-Published-20020719.pdf)
803 [Published-20020719.pdf](http://uddi.org/pubs/DataStructure-V2.03-Published-20020719.pdf).
- 804 2. Using WSDL in a UDDI Registry, May 21, 2002. Available at <http://uddi.org/pubs/wsdlbestpractices.pdf>
- 805 3. Web Services Description Language (WSDL) 1.1, March 15, 2000. Available at <http://www.w3.org/TR/wsdl>
- 806 4. XML Pointer Language (XPointer) Version 1.0, January 8, 2001. Available at <http://www.w3.org/TR/WD-xptr>

807 Appendix A: Canonical tModels

808 A.1 WSDL Entity Type tModel

809 A.1.1 Design Goals

810 This tModel provides a typing system based on the WSDL entity which the UDDI entity represents.

811 A.1.2 Definition

812 **Name:** uddi.org:wsdl:types

813 **Description:** WSDL Type Category System

814 **V3 format key:** uddi:7950ce8d-259a-4a9a-8f13-e51632fe12d5

815 **V1,V2 format key:** uuid:7950ce8d-259a-4a9a-8f13-e51632fe12d5

816 **Categorization:** categorization

817 **Checked:** no

818 A.1.2.1 tModel Structure

```
819     <tModel tModelKey="uuid:7950ce8d-259a-4a9a-8f13-e51632fe12d5" >
820         <name>uddi.org:wsdl:types</name>
821         <overviewDoc>
822             <overviewURL>
823                 http://uddi.org/pubs/uddi_wsdl_technical_note_v2.htm#wsdlTypes
824             </overviewURL>
825         </overviewDoc>
826         <categoryBag>
827             <keyedReference tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
828                 keyValue="unchecked" />
829             <keyedReference tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
830                 keyValue="categorization" />
831         </categoryBag>
832     </tModel>
```

833 A.1.3 Values

834 While this is an unchecked taxonomy, there are only four values that should be used with this taxonomy:

ID	Description	UDDI Entity
portType	Represents a UDDI entity categorized as a wsdl:portType	tModel
binding	Represents a UDDI entity categorized as a wsdl:binding	tModel

service	Represents a UDDI entity categorized as a wsdl:service	businessService
port	Represents a UDDI entity categorized as a wsdl:port	bindingTemplate (v3 only)

835

836 **A.1.4 Example of Use**

837 A tModel representing a portType tModel would have a categoryBag representing its type:

```

838     <categoryBag>
839         <keyedReference tModelKey="uuid:7950ce8d-259a-4a9a-8f13-e51632fe12d5"
840         keyName="WSDL Entity type" keyValue="portType" />
841         ...
842     </categoryBag>

```

843 **A.2 WSDL Namespace tModel**844 **A.2.1 Design Goals**

845 Because each WSDL entity has an associated target namespace, there needs to be a way to capture that target
846 namespace. The WSDL Namespace tModel serves this purpose. *More than one tModel might be categorized with*
847 *the same namespace tModel* – in fact, this mapping would be quite common, as many WSDL files use a common
848 target namespace for <wsdl:portType>, <wsdl:binding> and <wsdl:service> elements.

849 **A.2.2 Definition**850 **Name:** uddi.org:wsdl:namespace851 **Description:** A tModel used to represent WSDL namespaces852 **V3 format key:** uddi:c0ec5422-44e9-4774-b2f1-5a89eca389f3853 **V1,V2 format key:** uuid:c0ec5422-44e9-4774-b2f1-5a89eca389f3854 **Categorization:** categorization855 **Checked:** No856 **A.2.2.1 tModel Structure**

```

857     <tModel tModelKey="uuid:c0ec5422-44e9-4774-b2f1-5a89eca389f3" >
858         <name>uddi.org:wsdl:namespace</name>
859         <overviewDoc>
860             <overviewURL>
861                 http://uddi.org/pubs/uddi_wsdl_technical_note_v2.htm#wsdlNamespace
862             </overviewURL>
863         </overviewDoc>
864         <categoryBag>
865             <keyedReference tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
866             keyValue="unchecked" />
867             <keyedReference tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
868             keyValue="categorization" />
869         </categoryBag>
870     </tModel>

```

871 **A.2.3 Values**

872 The values used in this taxonomy come directly from the different target namespaces being represented by
873 instances of this tModel.

874 **A.2.4 Example of Use**

875 A namespace keyedReference would be as follows:

```

876     <categoryBag>
877         <keyedReference tModelKey=" uuid:c0ec5422-44e9-4774-b2f1-
878         5a89eca389f3" keyName="namespace" keyValue="urn:foo" />
879         ...
880     </categoryBag>

```

881 A.3 WSDL Local Name tModel

882 A.3.1 Design Goals

883 Because each WSDL entity has a name attribute which is key to identifying that WSDL entity, there needs to be a
 884 way to capture that attribute in the mapping. In the case of portType and binding, this is mapped to the tModel
 885 name element. However, in the case of the wsdl:service entity, the name element of the businessService is not
 886 appropriate and, in the case of wsdl:port, the bindingTemplate entity does not have a name element. The WSDL
 887 Local Name tModel serves this purpose of capturing that information in these two constructs.

888 A.3.2 Definition

889 **Name:** uddi.org:wsdl:localname

890 **Description:** A tModel used to represent WSDL local names

891 **V3 format key:** uddi:248ccb56-6365-446d-847a-91171ae79740

892 **V1,V2 format key:** uuid:248ccb56-6365-446d-847a-91171ae79740

893 **Categorization:** categorization

894 **Checked:** No

895 A.3.2.1 tModel Structure

```

896     <tModel tModelKey="uuid:248ccb56-6365-446d-847a-91171ae79740" >
897         <name>uddi.org:wsdl:localname</name>
898         <overviewDoc>
899             <overviewURL>
900                 http://uddi.org/pubs/uddi_wsdl_technical_note_v2.htm#wsdlLocalName
901             </overviewURL>
902         </overviewDoc>
903         <categoryBag>
904             <keyedReference tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
905                 keyValue="unchecked" />
906             <keyedReference tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
907                 keyValue="categorization" />
908         </categoryBag>
909     </tModel>
  
```

910 A.3.3 Values

911 The values used in this taxonomy come directly from the different target namespaces being signified by this tModel.

912 A.3.4 Example of Use

913 A namespace keyedReference would be as follows:

```

914     <categoryBag>
915         <keyedReference tModelKey="uuid:248ccb56-6365-446d-847a-91171ae79740"
916             keyValue="StockQuoteService" />
917         ...
918     </categoryBag>
  
```

919 A.4 WSDL portType Reference tModel

920 A.4.1 Design Goals

921 There is also a need to be able to relate a wsdl:binding tModel to the wsdl:portType that it implements. In order to
 922 accomplish this goal, a tModel needs to be established that allow arcs to be created between tModels. This is
 923 accomplished by placing the tModel key of a given tModel as the keyValue of a keyedReference. In order to create
 924 these arcs, a tModel is needed that represents the wsdl:portType.

925 A.4.2 Definition

926 **Name:** uddi.org:wsdl:portTypeReference

927 **Description:**

928 **V3 format key:** uddi:acc0db62-2866-4081-8986-d62a7bf75ab3

929 **V1,V2 format key:** uuid:acc0db62-2866-4081-8986-d62a7bf75ab3

930 **Categorization:**

931 A.4.2.1 tModel Structure

```

932     <tModel tModelKey="uuid:acc0db62-2866-4081-8986-d62a7bf75ab3" >
933         <name>uddi.org:wSDL:portTypeReference</name>
934         <description xml:lang="en">This tModel is a namespace tModel that can
935         be used to identify other tModels as portType tModels.</description>
936         <overviewDoc>
937             <overviewURL>
938
939                 http://uddi.org/pubs/uddi_wSDL_technical_note_v2.htm#portTypeReference
940             </overviewURL>
941         </overviewDoc>
942         <categoryBag>
943             <keyedReference tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
944             keyValue="categorization" />
945             <keyedReference tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
946             keyValue="unchecked" />
947         </categoryBag>
948     </tModel>

```

949 A.4.3 Values

950 The values used in this taxonomy come directly from the tModelKeys that represent wSDL:portType tModels.

951 A.4.4 Example of Use

952 One would add the following keyed reference to signify wSDL:binding that implemented a known portType:

```

953     <categoryBag>
954         <keyedReference
955             tModelKey="uuid:acc0db62-2866-4081-8986-d62a7bf75ab3"
956             keyName="wSDL:portType Reference"
957             keyValue="uuid:e8cf1163-8234-4b35-865f-94a7322e40c3"
958         />
959     ...
960 </categoryBag>

```

961 Note that the keyValue is a GUID, which, if queried for using get_tModelDetail, would return a tModel that
962 represented a given portType.

963 A.5 SOAP Protocol tModel

964 A.5.1 Design Goals

965 The WSDL specification outlines usage of the SOAP extension to WSDL. This tModel is used to signify that a
966 given bindingTemplate uses the SOAP extension.

967 A.5.2 Definition

968 **Name:** http://schemas.xmlsoap.org/wSDL/soap/

969 **Description:** Used to represent the WSDL SOAP extension

970 **V3 format key:** uddi:454ef824-994f-4ce3-a892-3d97ad952707

971 **V1,V2 format key:** uuid:454ef824-994f-4ce3-a892-3d97ad952707

972 **Categorization:** protocol

973 A.5.2.1 tModel Structure

```

974     <tModel tModelKey="uuid:454ef824-994f-4ce3-a892-3d97ad952707">
975         <name>http://schemas.xmlsoap.org/wSDL/soap/</name>
976         <overviewDoc>
977             <overviewURL>
978                 http://uddi.org/pubs/uddi_wSDL_technical_note_v2.htm#soap
979             </overviewURL>
980         </overviewDoc>
981         <categoryBag>
982             <keyedReference
983                 tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
984                 keyName="Protocol"

```

```

985         keyValue="protocol"
986     />
987 </categoryBag>
988 </tModel>

```

989 A.5.3 Example of Use

990 In the case when a uddi:bindingTemplate implements a wsdl:binding, it is important to model the protocol used by
991 that binding. In such a way, a query can be run: show all bindings that support a given protocol.

```

992     <bindingTemplates>
993         <bindingTemplate
994             bindingKey="f793c521-0daf-434c-8700-0e32da232e74"
995             serviceKey="102b114a-52e0-4af4-a292-02700da543d4">
996             <accessPoint URLType="http">
997                 http://location/sample
998             </accessPoint>
999             <tModelInstanceDetails>
1000                 <tModelInstanceInfo
1001                     tModelKey="uuid:454ef824-994f-4ce3-a892-3d97ad952707">
1002                     <description xml:lang="en">
1003                         This tModel key represents the soap protocol
1004                     </description>
1005                 </tModelInstanceInfo>
1006                 ... ..
1007             </tModelInstanceDetails>
1008         </bindingTemplate>
1009     </bindingTemplates>

```

1010 A.6 SOAP over HTTP Transport tModel

1011 A.6.1 Design Goals

1012 Other transport tModels may be defined in the future.

1013 A.6.2 Definition

1014 **Name:** http://schemas.xmlsoap.org/soap/http

1015 **Description:** Used to represent soap over http

1016 **V3 format key:** uddi:26de470c-0dab-4471-89f9-57450a524005

1017 **V1,V2 format key:** uuid:26de470c-0dab-4471-89f9-57450a524005

1018 **Categorization:** transport

1019 A.6.2.1 tModel Structure

```

1020     <tModel tModelKey="uuid:26de470c-0dab-4471-89f9-57450a524005" >
1021         <name>http://schemas.xmlsoap.org/soap/http</name>
1022         <overviewDoc>
1023             <overviewURL>
1024                 http://uddi.org/pubs/uddi_wsdl_technical_note_v2.htm#http
1025             </overviewURL>
1026         </overviewDoc>
1027         <categoryBag>
1028             <keyedReference tModelKey="uuid:c1acf26d-9672-4404-9d70-39b756e62ab4"
1029                 keyName="Wire/transport protocol"
1030                 keyValue="transport" />
1031         </categoryBag>
1032     </tModel>

```

1033 A.6.3 Example of Use

1034 In the case when a uddi:bindingTemplate implements a wsdl:binding, it is important to model the transport used by
1035 that binding. In such a way, a query can be run: show all bindings that support a given transport.

```

1036     <bindingTemplates>
1037         <bindingTemplate
1038             bindingKey="f793c521-0daf-434c-8700-0e32da232e74"
1039             serviceKey="102b114a-52e0-4af4-a292-02700da543d4">

```



```
1040         <accessPoint URLType="http">
1041             http://location/sample
1042         </accessPoint>
1043         <tModelInstanceDetails>
1044             <tModelInstanceInfo
1045                 tModelKey="uuid:26de470c-0dab-4471-89f9-57450a524005">
1046                 <description xml:lang="en">
1047                     This tModel key represents the wsdl:binding transport
1048                 </description>
1049             </tModelInstanceInfo>
1050             ... ..
1051         </tModelInstanceDetails>
1052     </bindingTemplate>
1053 </bindingTemplates>
```

1054 [Appendix B: Using XPointer in overviewURL](#)

1055 **B.1 XPointer Syntax**

1056 As mentioned earlier, it is optional to have a fragment identifier in an overviewURL that identifies a WSDL
1057 document. It is RECOMMENDED that applications that use the overviewURL use the UDDI metadata, the values
1058 in keyedReferences etc., to precisely define the target WSDL component. If an application does not use the UDDI
1059 metadata then the fragment identifier provides a generic mechanism to achieve the same identification of a
1060 particular WSDL component.

1061 As the WSDL 1.1 schema does not allow for id attributes on WSDL elements, we cannot simply use a fragment
1062 identifier of the form #foo.

1063 If the optional fragment identifier is used, the syntax defined by Xpointer [4] MUST be used for the fragment
1064 identifier.

1065 Considering Example 2 in the WSDL 1.1 W3C Note, the form of the reference to the portType named
1066 StockQuotePortType would be
1067 [http://example.com/stockquote/stockquote.wsdl#xpointer\(//definitions/portType\[@name="StockQuotePortType"\]\)](http://example.com/stockquote/stockquote.wsdl#xpointer(//definitions/portType[@name=)
1068 whereas the full Xpointer syntax for the same reference, including the namespace, would be
1069 [http://example.com/stockquote/stockquote.wsdl#xmlns\(x=http://example.com/stockquote/definitions\)
1070 xpointer\(//x:definitions/portType\[@name="StockQuotePortType"\]\)](http://example.com/stockquote/stockquote.wsdl#xmlns(x=http://example.com/stockquote/definitions) xpointer(//x:definitions/portType[@name=).