



1

2

Position Paper: Local vs global

3

Proposal 01, 28 May 2002

4

Document identifier:

5

p-desre-localvsglobal-01 ([Word](#))

6

Location:

7

<http://www.oasis-open.org/committees/ubl/ndrsc/pos>

8

Authors:

9

Fabrice Desré, France Télécom <fabrice.desre@francetelecom.com>

10

Abstract:

11

This position paper outlines several options related to the use of global elements when designing the UBL schema library.

12

13

Status:

14

This is V01 of the local vs. global position paper intended for consideration by the OASIS UBL Naming and Design Rules subcommittee and other interested parties. It is still incomplete; however, parts of the recommendations made here have already been incorporated into the *Universal Business Language Naming and Design Rules Specification*.

15

16

17

18

19

If you are on the ubl-ndrsc@lists.oasis-open.org list for subcommittee members, send comments there. If you are not on that list, subscribe to the ubl-comment@lists.oasis-open.org list and send comments there. To subscribe, send an email message to ubl-comment-request@lists.oasis-open.org with the word "subscribe" as the body of the message.

20

21

22

23

24

Copyright © 2002 The Organization for the Advancement of Structured Information Standards [OASIS]

25

26 **Table of Contents**

27	1	Introduction	3
28	2	Requirements.....	3
29	3	XSD design	3
30	3.1	Global elements	4
31	3.2	Namespace interaction	4
32	4	Options.....	6
33	4.1	Current UBL recommendation	7
34	4.1.1	Schema definitions.....	7
35	4.1.2	Instances.....	8
36	4.1.3	Requirements analysis.....	9
37		Appendix A. Notices	14
38			

39 1 Introduction

40 The UBL schema library consists of a collection of data types, which are used either in the UBL
41 schemas themselves or in third parties schemas ("extended", "customized" schemas).

42 The Naming and Design Rules subcommittee (NDRSC) must make some decisions to specify
43 how valid UBL schemas can be built. Among these decision, the NDRSC must specify if the UBL
44 types should be backed up by global elements, and whether these elements should be
45 namespace-qualified or not.

46 The original decision on this matter (see [wd-ublndrsc-ndrdoc-08.doc](#)) is to have globally
47 declared qualified root elements, and locally declared unqualified sub elements.

48 2 Requirements

49 The following requirements were identified by the NDRSC :

- 50 • **Compatibility with previous NDR decisions**, e.g. element naming conventions (see [wd-
51 ublndrsc-ndrdoc-08.doc](#)).
- 52 • **Modeling flexibility** : content models must be able to include two elements of the same type.
53 The local vs. global decision must not preclude the design of complex types that carry several
54 fields of the same type.
- 55 • **Context rules friendliness**
56 The ability to use normal mechanisms of the context methodology without adding burden.
57 Several approaches exist for context methodology : Paella and TAAT.
- 58 • **XSLT/XPath friendliness**
59 The local vs. global decision has potential impacts on what the instances look like, and what
60 the resulting XML data model is. We have to come with a non intrusive solution allowing the
61 use similar of XPath expressions for standard and customized UBL instances.
- 62 • **Ease of processing**
63 We must take into account that some applications need to parse documents fragments
64 easily, without having the whole document context knowledge.

65 3 XSD design

66 In this section we will outline some XSD design considerations useful for the following sections of
67 this document.

68 3.1 Global elements

69 A global element can be built in two ways :

- 70 • Using a standalone element

```
71 <xs:element name="info">  
72   <xs:complexType>  
73     <xs:sequence>  
74       <xs:element name="name" type="xs:string"/>  
75       <xs:element name="age" type="xs:int"/>  
76     </xs:sequence>  
77   </xs:complexType>  
78 </xs:element>
```

79

- 80 • Using a type reference

```
81 <xs:complexType name="infoType">  
82   <xs:sequence>  
83     <xs:element name="name" type="xs:string"/>  
84     <xs:element name="age" type="xs:int"/>  
85   </xs:sequence>  
86 </xs:complexType>  
87  
88 <xs:element name="info" type="infoType"/>
```

89 Each of these schemas be used to validate the same instance :

```
90 <?xml version="1.0" encoding="ISO-8859-1" ?>  
91 <info>  
92 <name>Fabrice Desré</name>  
93 <age>29</age>  
94 </info>
```

95

96 3.2 Namespace interaction

97 The previous example uses unqualified elements. However we will use namespaces in UBL for
98 several purposes (modularity, versioning, extensions), so we must be aware of the
99 consequences.

100 When writing schemas, each module (i.e. file) end up with its own namespace target. A common
101 use case is then to reuse these schemas by importing them (using <xsd:import/>). The behavior
102 of such imported components can be described that way :

- 103 • Imported types don't preserve the namespace of their original schema. The newly
104 declared elements belong to the new namespace.
- 105 • Referenced global elements preserve the namespace of their original schema.

106 Here are two sample schemas to exhibit these differences :

107

108

File : localvsglobal.xsd

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.oasis-open.org/ubl"
  xmlns:ubl="http://www.oasis-open.org/ubl"
  elementFormDefault="qualified">

  <xs:element name="info">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name" type="xs:string"/>
        <xs:element name="age" type="xs:int"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="infoType">
    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="age" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>

  <xs:element name="infoRef" type="ubl:infoType"/>

</xs:schema>
```

135

136

File : customized.xsd

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.foo.org/"
  xmlns:ubl="http://www.oasis-open.org/ubl"
  xmlns:foo="http://www.foo.org/"
  elementFormDefault="qualified">

  <xs:import namespace="http://www.oasis-open.org/ubl"
    schemaLocation="localvsglobal.xsd"/>

  <xs:complexType name="myFirstInfo">
    <xs:sequence>
      <xs:element name="info" type="ubl:infoType"/>
      <xs:element name="country" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
```

153
154
155
156
157
158
159
160
161
162
163
164
165

```
<xs:element name="info1" type="foo:myFirstInfo"/>

<xs:complexType name="mySecondInfo">
  <xs:sequence>
    <xs:element ref="ubl:info"/>
    <xs:element name="country" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="info2" type="foo:mySecondInfo"/>

</xs:schema>
```

166

167 And two valid instances :

168
169
170
171
172
173
174
175
176

```
<?xml version="1.0" encoding="UTF-8" ?>
<foo:info1 xmlns:ubl="http://www.oasis-open.org/ubl"
  xmlns:foo="http://www.foo.org/">
  <foo:info>
    <foo:name>Fabrice</foo:name>
    <foo:age>29</foo:age>
  </foo:info>
  <foo:country>France</foo:country>
</foo:info1>
```

177

178
179
180
181
182
183
184
185
186

```
<?xml version="1.0" encoding="UTF-8" ?>
<foo:info2 xmlns:ubl="http://www.oasis-open.org/ubl"
  xmlns:foo="http://www.foo.org/">
  <ubl:info>
    <ubl:name>Fabrice</ubl:name>
    <ubl:age>29</ubl:age>
  </ubl:info>
  <foo:country>France</foo:country>
</foo:info2>
```

187
188

We can see that in the first instance (info1), the UBL type "info" has been borrowed and is now referred as "foo:info". On the other side, the second instance preserve the real origin of the type.

189

4 Options

190

I will consider two options:

191

- The current UBL recommendation

192

- An alternative one, trying to better fulfill our requirements.

193 For each one, a sample UBL schema will be given, along with a customized one and the relevant
194 instances.

195 **4.1 Current UBL recommendation**

196 The current UBL recommendation dictates to :

- 197 • Use qualified elements for root elements in the standard UBL library.
- 198 • Use unqualified elements for every other elements in the standard UBL library.
- 199 • Extension writers must use qualified elements.

200

201 **4.1.1 Schema definitions**

202 An hypothetical UBL order schema :

```
203 <?xml version="1.0" encoding="UTF-8" ?>
204 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
205           targetNamespace="http://www.oasis-open.org/ubl"
206           xmlns:ubl="http://www.oasis-open.org/ubl"
207           elementFormDefault="unqualified">
208
209   <xs:complexType name="orderType">
210     <xs:sequence>
211       <xs:element name="header" type="ubl:headerType"/>
212       <xs:element name="item" type="ubl:itemType"
213 maxOccurs="unbounded"/>
214     </xs:sequence>
215   </xs:complexType>
216
217   <xs:complexType name="headerType">
218     <xs:sequence>
219       <xs:element name="from" type="xs:string"/>
220       <xs:element name="to" type="xs:string"/>
221     </xs:sequence>
222   </xs:complexType>
223
224   <xs:complexType name="itemType">
225     <xs:sequence>
226       <xs:element name="desc" type="xs:string"/>
227       <xs:element name="price" type="xs:double"/>
228     </xs:sequence>
229   </xs:complexType>
230
231   <xs:element name="orderDoc" type="ubl:orderType"
232 form="qualified"/>
233
234 </xs:schema>
```

235 An extension/reuse of this schema to build a custom order :

```
236 <?xml version="1.0" encoding="UTF-8" ?>
237 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
238           targetNamespace="http://example.org/foo"
239           xmlns:ubl="http://www.oasis-open.org/ubl"
240           xmlns:foo="http://example.org/foo"
241           elementFormDefault="qualified">
242
243   <xs:import namespace="http://www.oasis-open.org/ubl"
244             schemaLocation="currentUBL.xsd"/>
245
246   <xs:complexType name="fooHeaderType">
247     <xs:sequence>
248       <xs:element name="from" type="xs:string"/>
249       <xs:element name="to" type="xs:string"/>
250       <xs:element name="date" type="xs:dateTime"/>
251     </xs:sequence>
252   </xs:complexType>
253
254   <xs:element name="fooOrderDoc">
255     <xs:complexType>
256       <xs:sequence>
257         <xs:element name="header" type="foo:fooHeaderType"/>
258         <xs:element name="item" type="ubl:itemType"
259           maxOccurs="unbounded"/>
260         <xs:element name="desc" type="xs:string"/>
261       </xs:sequence>
262     </xs:complexType>
263   </xs:element>
264
265 </xs:schema>
```

266 4.1.2 Instances

267 Here are the basic and the customized instances :

```
268 <?xml version="1.0" encoding="UTF-8" ?>
269 <ubl:orderDoc xmlns:ubl="http://www.oasis-open.org/ubl">
270   <header>
271     <from>Fabrice</from>
272     <to>NDRSC</to>
273   </header>
274   <item>
275     <desc>first item</desc>
276     <price>10</price>
277   </item>
278   <item>
279     <desc>second item</desc>
280     <price>20</price>
281   </item>
```


282

```
</ubl:orderDoc>
```

283

284

```
<?xml version="1.0" encoding="UTF-8" ?>
```

285

```
<foo:fooOrderDoc xmlns:ubl="http://www.oasis-open.org/ubl"
```

286

```
xmlns:foo="http://www.foo.org/">
```

287

```
<foo:header>
```

288

```
<foo:from>Fabrice</foo:from>
```

289

```
<foo:to>NDRSC</foo:to>
```

290

```
</foo:header>
```

291

```
<foo:item>
```

292

```
<foo:desc>first item</foo:desc>
```

293

```
<foo:price>10</foo:price>
```

294

```
</foo:item>
```

295

```
<foo:item>
```

296

```
<foo:desc>second item</foo:desc>
```

297

```
<foo:price>20</foo:price>
```

298

```
</foo:item>
```

299

```
<foo:desc>A sample order</foo:desc>
```

300

```
</foo:fooOrderDoc>
```

301

4.1.3 Requirements analysis

Compatibility with previous NDR decisions	+++ Implied by this option
Modeling flexibility	+++ Elements use type reference inside complex content models, thus allowing
Context rules friendliness	??
XSLT/XPath friendliness	--- Consider the Xpath expression needed to access an item's price. Since the item type is simply reused, we can expect similar solutions for both : <ul style="list-style-type: none"> • Within the standard instance : //item/price • Within the customized instance : //foo:item/foo:price Worst, an XSLT template matching rule to display an item would be : <ul style="list-style-type: none"> • <xsl:template match="item"> for the standard library. • <xsl:template match="foo:item"> for the customized library. This implies that standard stylesheets can't display

	customized documents in a backward compatible way (they could by using the local-name() Xpath function, but this means that we don't care at all about namespaces !).
Ease of processing	-- There is no way looking at the instance to know that <item/> and <foo:item/> have indeed the same content model. This implies that fragment processing application must be aware of this (how? Using the standard dictionary and an augmented one ?).

302 4.2 Globally qualified solution

303 This is an attempt to provide a design pattern that will fulfill at best our requirements. The
304 previous analysis showed us that the main drawbacks of the current solution are in the
305 "namespace qualifying" area.

306 The so called "globally qualified" solution can be viewed as an attempt to :

- 307 • Localize type definitions inside a namespace to allow type reuse.
- 308 • Create aliases to these types to preserve original namespace in customized schemas.

309 4.2.1 Schema definitions

310 An hypothetical UBL order schema :

```

311 <?xml version="1.0" encoding="UTF-8" ?>
312 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
313           targetNamespace="http://www.oasis-open.org/ubl"
314           xmlns:ubl="http://www.oasis-open.org/ubl"
315           elementFormDefault="qualified">
316
317   <xs:complexType name="orderType">
318     <xs:sequence>
319       <xs:element name="header" type="ubl:headerType"/>
320       <xs:element name="item" type="ubl:itemType"
321 maxOccurs="unbounded"/>
322     </xs:sequence>
323   </xs:complexType>
324
325   <xs:complexType name="headerType">
326     <xs:sequence>
327       <xs:element name="from" type="xs:string"/>
328       <xs:element name="to" type="xs:string"/>
329     </xs:sequence>
330   </xs:complexType>
331
332   <xs:complexType name="itemType">

```

```

333 <xs:sequence>
334 <xs:element name="desc" type="xs:string"/>
335 <xs:element name="price" type="xs:double"/>
336 </xs:sequence>
337 </xs:complexType>
338
339 <xs:element name="orderDoc" type="ubl:orderType"/>
340
341 <xs:element name="headerTypeRef" ref="ubl:headerType"/>
342 <xs:element name="itemTypeRef" ref="ubl:itemType"/>
343 </xs:schema>

```

344 An extension/reuse of this schema to build a custom order :

```

345 <?xml version="1.0" encoding="UTF-8" ?>
346 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
347 targetNamespace="http://example.org/foo"
348 xmlns:ubl="http://www.oasis-open.org/ubl"
349 xmlns:foo="http://example.org/foo"
350 elementFormDefault="qualified">
351
352 <xs:import namespace="http://www.oasis-open.org/ubl"
353 schemaLocation="currentUBL.xsd"/>
354
355 <xs:complexType name="fooHeaderType">
356 <xs:sequence>
357 <xs:element name="from" type="xs:string"/>
358 <xs:element name="to" type="xs:string"/>
359 <xs:element name="date" type="xs:dateTime"/>
360 </xs:sequence>
361 </xs:complexType>
362
363 <xs:element name="fooOrderDoc">
364 <xs:complexType>
365 <xs:sequence>
366 <xs:element ref="ubl:headerTypeRef"/>
367 <xs:element ref="ubl:itemTypeRef" maxOccurs="unbounded"/>
368 <xs:element name="desc" type="xs:string"/>
369 </xs:sequence>
370 </xs:complexType>
371 </xs:element>
372
373 </xs:schema>

```

374 4.2.2 Instances

375 Here are the basic and the customized instances :

```

376 <?xml version="1.0" encoding="UTF-8" ?>
377 <orderDoc xmlns="http://www.oasis-open.org/ubl">
378 <header>

```

```

379     <from>Fabrice</from>
380     <to>NDRSC</to>
381 </header>
382 <item>
383     <desc>first item</desc>
384     <price>10</price>
385 </item>
386 <item>
387     <desc>second item</desc>
388     <price>20</price>
389 </item>
390 </orderDoc>

```

391

```

392 <?xml version="1.0" encoding="UTF-8" ?>
393 <foo:fooOrderDoc xmlns:ubl="http://www.oasis-open.org/ubl"
394     xmlns:foo="http://www.foo.org/">
395   <foo:header>
396     <foo:from>Fabrice</foo:from>
397     <foo:to>NDRSC</foo:to>
398   </foo:header>
399   <ubl:itemTypeRef>
400     <ubl:desc>first item</ubl:desc>
401     <ubl:price>10</ubl:price>
402   </ubl:itemTypeRef>
403   <ubl:itemTypeRef>
404     <ubl:desc>second item</ubl:desc>
405     <ubl:price>20</ubl:price>
406   </ubl:itemTypeRef>
407   <foo:desc>A sample order</foo:desc>
408 </foo:fooOrderDoc>

```

409 4.2.3 Requirements analysis

Compatibility with previous NDR decisions	++ We'll probably need to write a recommendation for the naming of elements that are type aliases.
Modeling flexibility	-- In each module, we can reuse a type several times in a content model. But a customized schemas can't reference several times an imported type.
Context rules friendliness	??
XSLT/XPath friendliness	+++ Consider the Xpath expression needed to access an item's

	<p>price. Since the item type is simply reused, we can except similar solutions for both :</p> <ul style="list-style-type: none"> • Within the standard instance : //ubl:item/ubl:price • Within the customized instance : //ubl:itemTypeRef/ubl:price <p>In the same vein an XSLT template matching rule to display an item would be :</p> <ul style="list-style-type: none"> • <xsl:template match="ubl:item ubl:itemTypeRef"> for both. <p>This shows that we don't have to update the standard stylesheets to provide a default display to customized instances.</p>
Ease of processing	<p>+++</p> <p>The application only needs to know about the UBL standard element name and the aliased name that can possibly appear in a customized instance.</p>

410

411 **5 Issues**

412 The pending issues so far are :

- 413 • Modeling flexibility with the globally qualified solution
- 414 • Context rules friendliness : we need to know more about TAAT formalism to analyse this.

415

Appendix A. Notices

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

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

428 Copyright © The Organization for the Advancement of Structured Information Standards [OASIS]
429 2001. All Rights Reserved.

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

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

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