



PPS (Production Planning and Scheduling) Part 1: Core Elements, Version 1.0

Public Review Draft 03

24 Oct 2009

Specification URIs:

<http://docs.oasis-open.org/pps/v1.0/pr03/pps-core-elements-1.0.doc>
<http://docs.oasis-open.org/pps/v1.0/pr03/pps-core-elements-1.0.html>
<http://docs.oasis-open.org/pps/v1.0/pr03/pps-core-elements-1.0.pdf> (Authoritative)

Previous Version:

<http://docs.oasis-open.org/pps/v1.0/cs01/pps-core-elements-1.0-cs01.doc>
<http://docs.oasis-open.org/pps/v1.0/cs01/pps-core-elements-1.0-cs01.html>
<http://docs.oasis-open.org/pps/v1.0/cs01/pps-core-elements-1.0-cs01.pdf>

Latest Version:

<http://docs.oasis-open.org/pps/v1.0/pps-core-elements-1.0.doc>
<http://docs.oasis-open.org/pps/v1.0/pps-core-elements-1.0.html>
<http://docs.oasis-open.org/pps/v1.0/pps-core-elements-1.0.pdf>

Technical Committee:

OASIS Production Planning and Scheduling TC

Chair(s):

Yasuyuki Nishioka, PSLX Forum / Hosei University

Editor(s):

Yasuyuki Nishioka, PSLX Forum / Hosei University
Koichi Wada, PSLX Forum

Related work:

This specification is related to:

- Universal Business Language 2.0

Declared XML Namespace(s):

<http://docs.oasis-open.org/ns/pps/2009>

Abstract:

OASIS PPS (Production Planning and Scheduling) specifications deal with problems of decision-making in all manufacturing companies who want to have a sophisticated information system for production planning and scheduling. PPS specifications provide XML schema and communication protocols for information exchange among manufacturing application programs in the web-services environment. Part 1: Core Element especially focuses on information model of core elements which can be used as ontology in the production planning and scheduling domain. Since the elements have been designed without particular contexts in planning and scheduling, they can be used in any specific type of messages as a building block depending on the context of application programs.

Status:

This document was last revised or approved by the PPS TC on the above date. The level of approval is also listed above. Check the “Latest Version” or “Latest Approved Version” location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee’s email list. Others should send comments to the Technical Committee by using the “Send A Comment” button on the Technical Committee’s web page at <http://www.oasis-open.org/committees/pps/>.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (<http://www.oasis-open.org/committees/pps/ipr.php>).

The non-normative errata page for this specification is located at <http://www.oasis-open.org/committees/pps/>.

Notices

Copyright © OASIS® 2007-2009. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

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

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

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The names "OASIS", PPS are trademarks of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

Table of Contents

1	Introduction.....	6
1.1	Terminology.....	6
1.2	Normative References.....	6
1.3	Non-Normative References.....	6
1.4	Terms and definitions.....	6
2	Primitive Elements.....	8
2.1	Structure of primitive elements.....	8
2.2	List of primitive elements.....	9
2.2.1	Party element.....	9
2.2.2	Plan element.....	10
2.2.3	Order element.....	10
2.2.4	Item element.....	10
2.2.5	Resource element.....	10
2.2.6	Process element.....	11
2.2.7	Lot element.....	11
2.2.8	Task element.....	11
2.2.9	Operation element.....	11
3	Relational Elements.....	12
3.1	Structure of relational elements.....	12
3.2	List of relational elements.....	13
3.2.1	Compose element.....	13
3.2.2	Produce element.....	14
3.2.3	Consume element.....	14
3.2.4	Assign element.....	14
3.2.5	Relation element.....	14
4	Specific Elements.....	15
4.1	Structure of specific element.....	15
4.2	List of specific elements.....	16
4.2.1	Location element.....	16
4.2.2	Capacity element.....	16
4.2.3	Progress element.....	17
4.2.4	Spec element.....	17
5	Eventual Elements.....	18
5.1	Structure of eventual element.....	18
5.2	List of eventual elements.....	19
5.2.1	Start element.....	19
5.2.2	End element.....	19
5.2.3	Event element.....	19
6	Accounting Elements.....	20
6.1	Structure of Accounting element.....	20
6.2	List of accounting elements.....	21
6.2.1	Price element.....	21
6.2.2	Cost element.....	21

7	Administrative Elements	22
7.1	Structure of Administrative Elements	22
7.2	List of Administrative Elements	22
7.2.1	Priority element	23
7.2.2	Display element	23
7.2.3	Description element	23
7.2.4	Author element	23
7.2.5	Date element	23
8	Data Elements	24
8.1	Qty element	24
8.2	Char element	24
8.3	Time element	25
9	Conformance	27
A.	Object Class diagram	28
B.	Cross reference of elements	29
C.	Acknowledgements	31
D.	Revision History	32

1 Introduction

This document prescribes how to describe contents of the XML messages which are used for exchanging the information on Production Planning and Scheduling by some application software programs.

If information is exchanged between production planning and scheduling applications, the enterprise can develop systems comparatively easily at a low cost and make them more competitive for the whole enterprise. To make matters better, the systems will be able to have high extendability in future.

This specification aims at production planning and scheduling for all kinds of products and services provided by manufacturing enterprises. Production scheduling explained in this specification can be divided into scheduling in the whole enterprise including some areas and sites, and detailed scheduling within an individual area and work-centers.

The scope of this specification doesn't include optimization logic for solution, special knowledge of individual enterprises, concrete solution methods for production planning and scheduling, and planning for the total supply chain.

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1.2 Normative References

- [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- [PPS02] OASIS Public Review Draft 03, PPS (Production Planning and Scheduling) Part 2: Transaction Messages, Version 1.0, <http://docs.oasis-open.org/pps/v1.0/pr03/pps-transaction-messages-1.0.pdf>
- [PPS03] OASIS Public Review Draft 03, PPS (Production Planning and Scheduling) Part 3: Profile Specifications, Version 1.0, <http://docs.oasis-open.org/pps/v1.0/pr03/pps-profile-specifications-1.0.pdf>

1.3 Non-Normative References

- [PSLXWP] PSLX Consortium, PSLX White Paper - APS Conceptual definition and implementation, <http://www.pslx.org/>
- [PSLX001] PSLX Technical Standard, Version 2, Part 1: Enterprise Model (in Japanese), Recommendation of PSLX Forum, <http://www.pslx.org/>
- [PSLX002] PSLX Technical Standard, Version 2, Part 2: Activity Model (in Japanese), Recommendation of PSLX Forum, <http://www.pslx.org/>
- [PSLX003] PSLX Technical Standard, Version 2, Part 3: Object Model (in Japanese), Recommendation of PSLX Forum, <http://www.pslx.org/>

1.4 Terms and definitions

Plan

Unit for intensive information of related orders corresponding to a specific period on a discrete time scale, or calculated information based on the schedule under the related orders. This can represent actual results when the related events have been occurred.

- 43 **Order**
44 Unit of requirement describing concrete item, resource or operation in a specific place at a
45 specific time. This can also represent the results to the requirement.
- 46 **Party**
47 Customer who is a sender of an order and has a demand to make a decision, or supplier who is a
48 receiver in case that a decision-maker sends the demand that can't be handled inside.
- 49 **Item**
50 Object to be produced or consumed by production activities. The quantity or the quality of item is
51 changed during the production activity. Examples include product, parts, module, unit, work in
52 process and materials.
- 53 **Resource**
54 Object that can provide essential function for production activities. The capacity of function is
55 used during production activity, and is available again after finishing the production. Examples
56 include equipment, machine, device, labor and tool.
- 57 **Process**
58 Segment of production activities indicating a certain production line or method. This takes
59 duration from start time to end time, and gives added value to the producing item. One process
60 may have two or more than two processes detailed in the lower levels.
- 61 **Lot**
62 Instance of a specific volume of item that exists in a specific place at a specific time. Generally
63 the specific time corresponds to start or end of an operation, and the specific volume is equal to
64 the quantity of item produced or consumed by the operation.
- 65 **Task**
66 Unit of necessity to execute a specific operation at a specific time, indicating the volume of used
67 capability provided by the applicable resource. This can represent both capacity value provided
68 by resource at a specific time point, and aggregated total value of capacity provided by resource
69 during specific duration.
- 70 **Operation**
71 Actual processing element to be executed by a specific task, and to produce or consume a
72 specific lot. It is a concrete instance of particular processes in production activities.

73

2 Primitive Elements

74

2.1 Structure of primitive elements

75

Primitive elements are the minimum series of element that corresponds to the most basic domain objects.

76

The type of this element SHOULD be represented with the following XML schema and SHOULD fulfill the following constraints.

77

78

79

```

<xsd:complexType name="PrimitiveType">
  <xsd:sequence>
    <xsd:element ref="Compose" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Produce" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Consume" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Assign" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Relation" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Location" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Capacity" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Progress" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Spec" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Start" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="End" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Event" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Price" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Cost" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Priority" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Display" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Description" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Author" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Date" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:string" use="required"/>
  <xsd:attribute name="key" type="xsd:long"/>
  <xsd:attribute name="name" type="xsd:string"/>
  <xsd:attribute name="parent" type="xsd:string"/>
  <xsd:attribute name="type" type="xsd:string"/>
  <xsd:attribute name="status" type="xsd:string"/>
  <xsd:attribute name="party" type="xsd:string"/>
  <xsd:attribute name="plan" type="xsd:string"/>
  <xsd:attribute name="order" type="xsd:string"/>
  <xsd:attribute name="item" type="xsd:string"/>
  <xsd:attribute name="resource" type="xsd:string"/>
  <xsd:attribute name="process" type="xsd:string"/>
  <xsd:attribute name="lot" type="xsd:string"/>
  <xsd:attribute name="task" type="xsd:string"/>
  <xsd:attribute name="operation" type="xsd:string"/>
</xsd:complexType>

```

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

- *id* attribute SHOULD represent an identifier of this element.

119

- *key* attribute SHOULD represent a key used in the local applications.

120

- *name* attribute SHOULD represent the name of this element.

121

- *parent* attribute SHOULD represent the identifier of the inherited element of this element.

122

- *type* attribute SHOULD represent the division of this element.

123

- *status* attribute SHOULD represent the status of this element.

124

- *party* attribute SHOULD represent an identifier of the party associated with this element.

125

- *plan* attribute SHOULD represent the identifier of the plan associated with this element.

126

- *order* attribute SHOULD represent the identifier of the order associated with this element.

- 127 • *item* attribute SHOULD represent the identifier of the item associated with this element.
- 128 • *resource* attribute SHOULD represent the identifier of the resource associated with this element.
- 129 • *process* attribute SHOULD represent the identifier of the process associated with this element.
- 130 • *lot* attribute SHOULD represent the identifier of the lot associated with this element.
- 131 • *task* attribute SHOULD represent the identifier of the task associated with this element.
- 132 • *operation* attribute SHOULD represent the identifier of the operation associated with this element.
- 133
- 134 • *Compose* element SHOULD represent the element corresponding to part of this element.
- 135 • *Produce* element SHOULD represent the relation that this element produces.
- 136 • *Consume* element SHOULD represent the relation that this element consumes.
- 137 • *Assign* element SHOULD represent the relation that this element uses.
- 138 • *Relation* element SHOULD represent the relation to other primitive elements.
- 139 • *Location* element SHOULD represent the location where this element exists.
- 140 • *Capacity* element SHOULD represent the capacity status of this element.
- 141 • *Progress* element SHOULD represent the progress of this element.
- 142 • *Spec* element SHOULD represent the specification of this element.
- 143 • *Start* element SHOULD represent the start event of this element.
- 144 • *End* element SHOULD represent the completion event of this element.
- 145 • *Event* element SHOULD represent the optional event under this element.
- 146 • *Price* element SHOULD represent the price of this element.
- 147 • *Cost* element SHOULD represent the cost of this element.
- 148 • *Priority* element SHOULD represent the priority of this element.
- 149 • *Display* element SHOULD represent how to display this element.
- 150 • *Description* element SHOULD represent the description of this element.
- 151 • *Author* element SHOULD represent the author of this element information.
- 152 • *Date* element SHOULD represent the date of this element information.

153 2.2 List of primitive elements

154 This specification defines nine primitive elements: *Party*, *Plan*, *Order*, *Item*, *Resource*, *Process*, *Lot*, *Task*,
 155 and *Operation*. The type of those elements SHOULD be represented with the following XML schema.

156

```

157 <xsd:element name="Party" type="PrimitiveType"/>
158 <xsd:element name="Plan" type="PrimitiveType"/>
159 <xsd:element name="Order" type="PrimitiveType"/>
160 <xsd:element name="Item" type="PrimitiveType"/>
161 <xsd:element name="Resource" type="PrimitiveType"/>
162 <xsd:element name="Process" type="PrimitiveType"/>
163 <xsd:element name="Lot" type="PrimitiveType"/>
164 <xsd:element name="Task" type="PrimitiveType"/>
165 <xsd:element name="Operation" type="PrimitiveType"/>

```

166

167 2.2.1 Party element

168 *Party* element represents a customer or a supplier. Customer is an object that requests some products or
 169 services to the enterprise. The requests are sent to a person who is in charge of production planning and

170 scheduling. Supplier is an object providing some products or services to the enterprise. Supplier
171 receives orders form the enterprise, and provides corresponding items, resources or processes for the
172 enterprise.

173 2.2.2 Plan element

174 *Plan* element represents a value planned for particular products or services. The value shows volume of
175 the products or services required or resulted during certain period of time. Typical cases of planning
176 period include day, week and month.

177 2.2.3 Order element

178 *Order* element represents an object of information produced to request some products or services. Order
179 is source to create production orders that are finally dispatched to the plant floor. Orders can be divided
180 into inventory order, capacity order and production order according to the type of request.

181

182 Example: Ten "A" products are requested.

```
183 <Order id="Z01" item="A">  
184 <Spec type="quantity"><Qty value="10"/></Spec>  
185 </Order>
```

186 Example: Three labors in group "B" are requested.

```
187 <Order id="Z02" resource="groupB">  
188 <Spec type="quantity"><Qty value="3"/></Spec>  
189 </Order>
```

190 Example: Switching operation is requested two times.

```
191 <Order id="Z03" process="change01">  
192 <Spec type="quantity"><Qty value="2"/></Spec>  
193 </Order>
```

194 Example: Order which consist of 10 of "A" and 5 of "B" is totally 3,000 yen.

```
195 <Order id="Z00">  
196 <Compose order="Z01"/>  
197 <Compose order="Z02"/>  
198 <Price value="3000" unit="yen"/>  
199 </Order>  
200 <Order id="Z01" item="A">  
201 <Spec type="quantity"><Qty value="10"/></Spec>  
202 </Order>  
203 <Order id="Z02" item="B">  
204 <Spec type="quantity"><Qty value="5"/></Spec>  
205 </Order>
```

206

207 2.2.4 Item element

208 *Item* element represents a product, component, parts, work in process (WIP), raw material and other
209 items. Item is produced by any processes, and after that, it is consumed by another processes. Lot is an
210 instance of the corresponding item.

211 2.2.5 Resource element

212 *Resource* element represents a resource, which is an object enabling production, transportation, storage,
213 inspection and other various services. As resource can produce tasks to execute operations, it is
214 assigned to an operation by considering its volume of capacity.

215 **2.2.6 Process element**

216 *Process* element represents a process that has a function to produce value. Process can be defined as a
217 segment of activities in production process. It produces and consumes production items by being
218 executed during certain period of time.

219 **2.2.7 Lot element**

220 *Lot* element represents a production lot. Production lot is an object corresponding to a concrete item that
221 actually exists in a specific place at a specific date and time. Lot is produced by an operation and finally
222 consumed by another operation or discarded.

223 **2.2.8 Task element**

224 *Task* element represents a task, which is an object showing the usage of a specific resource capability for
225 a specific period of time. Schedule may request a certain volume of task for each resource assigned to
226 execute the appropriate operations.

227

228 Example: Task corresponding to the volume that 3 labors work load is required for 2 days

```
229 <Task id="T01">  
230 <Capacity type="human"><Qty value="3"/></Capacity>  
231 <Capacity type="duration"><Qty value="2" unit="day" /></Capacity>  
232 </Task>
```

233

234 **2.2.9 Operation element**

235 *Operation* element represents a segment of activities that is actually dispatched to plant floor. Operation
236 identifies an executable function at a specific place on a plant floor for a specific time. Operation is
237 associated with a specific lot and task by executing those activities.

238

3 Relational Elements

239

3.1 Structure of relational elements

240

Relational elements represent any relations between primitive elements. A relational element can have properties. The type of this element SHOULD be represented with the following XML schema and SHOULD fulfill the following constraints.

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

```
<xsd:complexType name="RelationalType">
  <xsd:sequence>
    <xsd:element ref="Location" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Capacity" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Progress" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Spec" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Start" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="End" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Event" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Price" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Cost" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Priority" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Display" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Description" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Author" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Date" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Qty" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Char" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Time" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:string"/>
  <xsd:attribute name="key" type="xsd:long"/>
  <xsd:attribute name="name" type="xsd:string"/>
  <xsd:attribute name="type" type="xsd:string"/>
  <xsd:attribute name="status" type="xsd:string"/>
  <xsd:attribute name="apply" type="xsd:string"/>
  <xsd:attribute name="party" type="xsd:string"/>
  <xsd:attribute name="plan" type="xsd:string"/>
  <xsd:attribute name="order" type="xsd:string"/>
  <xsd:attribute name="item" type="xsd:string"/>
  <xsd:attribute name="resource" type="xsd:string"/>
  <xsd:attribute name="process" type="xsd:string"/>
  <xsd:attribute name="lot" type="xsd:string"/>
  <xsd:attribute name="task" type="xsd:string"/>
  <xsd:attribute name="operation" type="xsd:string"/>
</xsd:complexType>
```

280

281

- *id* attribute SHOULD represent an identifier of this element.

282

- *key* attribute SHOULD represent a key used in the local applications.

283

- *name* attribute SHOULD represent the name of this element.

284

- *type* attribute SHOULD represent the division of this element.

285

- *status* attribute SHOULD represent the status of this element.

286

- *apply* attribute SHOULD represent application type of this element. This element is a disjunctive (OR) content under the parent element, if the attribute value is "*disjunctive*".

287

288

- *party* attribute SHOULD represent an identifier of the party associated with this element.

289

- *plan* attribute SHOULD represent the identifier of the plan associated with this element.

290

- *order* attribute SHOULD represent the identifier of the order associated with this element.

- 291 • *item* attribute SHOULD represent the identifier of the item associated with this element.
- 292 • *resource* attribute SHOULD represent the identifier of the resource associated with this element.
- 293 • *process* attribute SHOULD represent the identifier of the process associated with this element.
- 294 • *lot* attribute SHOULD represent the identifier of the lot associated with this element.
- 295 • *task* attribute SHOULD represent the identifier of the task associated with this element.
- 296 • *operation* attribute SHOULD represent the identifier of the operation associated with this element.
- 297
- 298 • *Location* element SHOULD represent the location associated with this element.
- 299 • *Capacity* element SHOULD represent the capacity status of this element.
- 300 • *Progress* element SHOULD represent the progress of this element.
- 301 • *Spec* element SHOULD represent the specification of this element.
- 302 • *Start* element SHOULD represent the start event of this element.
- 303 • *End* element SHOULD represent the completion event of this element.
- 304 • *Event* element SHOULD represent the optional event under this element.
- 305 • *Price* element SHOULD represent the price of this element.
- 306 • *Cost* element SHOULD represent the cost of this element.
- 307 • *Priority* element SHOULD represent the priority of this element.
- 308 • *Display* element SHOULD represent how to display this element.
- 309 • *Description* element SHOULD represent the description of this element.
- 310 • *Author* element SHOULD represent the author of this element information.
- 311 • *Date* element SHOULD represent the date of this element information.
- 312 • *Qty* element SHOULD represent the quantity of this element.
- 313 • *Char* element SHOULD represent the qualitative value of this element.
- 314 • *Time* element SHOULD represent the time of this element.
- 315

316 3.2 List of relational elements

317 This part of specifications defines five relational elements: *Compose*, *Produce*, *Consume*, *Assign*, and
 318 *Relation*. Relational element defines relationship between the parent element and those that characterize
 319 the element. The type of this element SHOULD be represented with the following XML schema.

320

```
321 <xsd:element name="Compose" type="RelationalType"/>
322 <xsd:element name="Produce" type="RelationalType"/>
323 <xsd:element name="Consume" type="RelationalType"/>
324 <xsd:element name="Assign" type="RelationalType"/>
325 <xsd:element name="Relation" type="RelationalType"/>
```

326

327 3.2.1 Compose element

328 *Compose* element defines a hierarchical relation between the parent element and another same primitive
 329 element that addresses one level upper or lower than the target element. This element can represent
 330 that the object referred to in this element composes or be composed by the parent element.

331

332 Example: Product “A” family includes product “A1” and product “A2”.

```
333 <Item id="A">
334 <Compose type="child" item="A1"/>
335 <Compose type="child" item="A2"/>
336 </Item>
```

337 Example: Product "B" is assembled with 2 of parts "C1" and 3 of parts "C2".

```
338 <Item id="B">
339 <Compose type="child" item="C1"><Qty value="2"/></Compose>
340 <Compose type="child" item="C2"><Qty value="3"/></Compose>
341 </Item>
```

342 Example: 2 of parts "C1" are used for product "B1", and 5 of parts "C1" are used for product "B2".

```
343 <Item id="C1">
344 <Compose type="parent" item="B1"><Qty value="2"/></Compose>
345 <Compose type="parent" item="B2"><Qty value="5"/></Compose>
346 </Item>
```

347

348 3.2.2 Produce element

349 *Produce* element defines a relation between processes and items, or a relation between operations and
350 lots. This element can show the quantity of the item or lot produced by the process or operation
351 respectively, or how many items or lots are produced by the process or the operation respectively.

352 3.2.3 Consume element

353 *Consume* element defines a relation between processes and items, or a relation between operations and
354 lots. This element can show the quantity of the item or lot consumed by the process or operation
355 respectively, or how many items or lots are consumed by the process or operation respectively.

356 3.2.4 Assign element

357 *Assign* element defines a relation between processes and resources, or a relation between operations
358 and tasks. This element can show the volume of capacity provided by the resource or task assigned for
359 the process or operation respectively, or how many resources or tasks are used.

360 3.2.5 Relation element

361 *Relation* element can show that the parent element has a specific relation to other primitive elements.
362 This element can additionally define relational classes between primitive elements. Examples include
363 precedence relations and pegging relations.

364

4 Specific Elements

365

4.1 Structure of specific element

366

Specific elements are defined to represent any properties of the primitive element. This element MAY be described more than once on the same parent element if the value is historical. Those multiple properties have time stamp. The type of this element SHOULD be represented with the following XML schema and SHOULD fulfill the following constraints.

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

```
<xsd:complexType name="SpecificType">
  <xsd:sequence>
    <xsd:element ref="Start" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="End" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Event" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Price" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Cost" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Priority" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Display" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Description" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Author" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Date" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Qty" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Char" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Time" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:string"/>
  <xsd:attribute name="key" type="xsd:long"/>
  <xsd:attribute name="name" type="xsd:string"/>
  <xsd:attribute name="type" type="xsd:string"/>
  <xsd:attribute name="status" type="xsd:string"/>
  <xsd:attribute name="apply" type="xsd:string"/>
</xsd:complexType>
```

394

395

- *id* attribute SHOULD represent an identifier of this element.

396

- *key* attribute SHOULD represent a key used in the local applications.

397

- *name* attribute SHOULD represent the name of this element.

398

- *type* attribute SHOULD represent the division of this element.

399

- *status* attribute SHOULD represent the status of this element.

400

- *apply* attribute SHOULD represent application type of this element. Specification of the element is relative, if the value is "*relative*".

401

402

403

- *Start* element SHOULD represent the start event of this element.

404

- *End* element SHOULD represent the completion event of this element.

405

- *Event* element SHOULD represent the optional event under this element.

406

- *Price* element SHOULD represent the price of this element.

407

- *Cost* element SHOULD represent the cost of this element.

408

- *Priority* element SHOULD represent the priority of this element.

409

- *Display* element SHOULD represent how to display this element.

410

- *Description* element SHOULD represent the description of this element.

411

- *Author* element SHOULD represent the author of this element information.

- 412 • *Date* element SHOULD represent the date of this element information.
- 413 • *Qty* element SHOULD represent the quantity of this element.
- 414 • *Char* element SHOULD represent the qualitative value of this element.
- 415 • *Time* element SHOULD represent the time of this element.

417 4.2 List of specific elements

418 For specific elements, this part of specifications has four elements: *Location*, *Capacity*, *Progress*, and
 419 *Spec*. The type of this element SHOULD be represented with the following XML schema.

420

```
421 <xsd:element name="Location" type="SpecificType"/>
422 <xsd:element name="Capacity" type="SpecificType"/>
423 <xsd:element name="Progress" type="SpecificType"/>
424 <xsd:element name="Spec" type="SpecificType"/>
```

425

426 4.2.1 Location element

427 *Location* element represents a location. When the expression of location has structure, multiple values
 428 can be set by describing different names of the data. Change of the location depending on time can also
 429 be represented by multiple values.

430

431 Example: Customer's address

```
432 <Party id="ABC Inc.">
433 <Location type="address"><Char value="123 ABC street"/></Location>
434 <Location type="city"><Char value="Cambridge"/></Location>
435 <Location type="state"><Char value="MA"/></Location>
436 <Location type="code"><Char value="02139"/></Location>
437 <Location type="country"><Char value="USA"/></Location>
438 </Party>
```

439

440 4.2.2 Capacity element

441 *Capacity* element represents volume of capability provided by resources, items or processes. In the case
 442 of resource capability, it may show available amount of corresponding tasks. In the case of Items, it
 443 shows the available amount of Lots. And for Processes, it shows maximum ratio of production. All of this
 444 information is represented in a time horizon.

445

446 Example: Inventory level of "material01"

```
447 <Item id="material01">
448 <Capacity><Qty value="150"/></Capacity>
449 </Item>
```

450 Example: Temporal change of the material

```
451 <Item id="material01">
452 <Capacity><Qty value="150"><Time value="2005-04-10T00:00:00"/></Capacity>
453 <Capacity><Qty value="200"><Time value="2005-04-17T00:00:00"/></Capacity>
454 </Item>
```

455 Example: Material location information: Stock of "material01" is 150 located at "storage01"

```
456 <Item id="material01">
457 <Location value="storage01"/>
```

458 <Capacity><Qty value="150"/></Capacity>
459 </Item>

460

461 **4.2.3 Progress element**

462 *Progress* element represents progress of order and operation, or status of lot and task. This element
463 shows the latest data, status or progress at a specific time point. This element MAY represent a change
464 of time-dependent values.

465 **4.2.4 Spec element**

466 *Spec* element represents various specifications for primitive elements. The content can be represented
467 with a pair of a spec name and a value. This element can also represent time-dependent change of the
468 value. The value of the specification is represented with one data type of a numerical value, characters
469 and date time. Spec elements with the same specification name under a common parent element
470 SHOULD be represented with the same data type.

471

5 Eventual Elements

472

5.1 Structure of eventual element

473

Eventual elements represent any properties that occur at one time point. Any type of events can be specified by using this element. The type of this element SHOULD be represented with the following XML schema and SHOULD fulfill the following constraints.

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

```
<xsd:complexType name="EventualType">
  <xsd:sequence>
    <xsd:element ref="Priority" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Display" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Description" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Author" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Date" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Qty" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Char" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Time" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:string"/>
  <xsd:attribute name="key" type="xsd:long"/>
  <xsd:attribute name="name" type="xsd:string"/>
  <xsd:attribute name="type" type="xsd:string"/>
  <xsd:attribute name="status" type="xsd:string"/>
  <xsd:attribute name="apply" type="xsd:string"/>
  <xsd:attribute name="condition" type="xsd:string"/>
  <xsd:attribute name="value" type="xsd:string"/>
</xsd:complexType>
```

497

498

- *id* attribute SHOULD represent an identifier of this element.

499

- *key* attribute SHOULD represent a key used in the local applications.

500

- *name* attribute SHOULD represent the name of this element.

501

- *type* attribute SHOULD represent the division of this element.

502

- *status* attribute SHOULD represent the status of this element.

503

- *apply* attribute SHOULD represent application type of this element. The condition of this element is exclusive, if the value is "exclusive".

504

505

- *condition* attribute SHOULD represent the condition of this element.

506

- *value* attribute SHOULD represent the qualitative value of this element.

507

508

- *Priority* element SHOULD represent the priority of this element.

509

- *Display* element SHOULD represent how to display this element.

510

- *Description* element SHOULD represent the description of this element.

511

- *Author* element SHOULD represent the author of this element information.

512

- *Date* element SHOULD represent the date of this element information.

513

- *Qty* element SHOULD represent the quantity of this element.

514

- *Char* element SHOULD represent the qualitative value of this element.

515

- *Time* element SHOULD represent the time of this element.

516

517 **5.2 List of eventual elements**

518 This part of specifications defines three eventual elements: *Start*, *End*, and *Event*. The *Start* and *End* are
519 special cases of *Event* element. The type of this element SHOULD be represented with the following XML
520 schema.

521

```
522 <xsd:element name="Start" type="EventualType"/>  
523 <xsd:element name="End" type="EventualType"/>  
524 <xsd:element name="Event" type="EventualType"/>
```

525

526 **5.2.1 Start element**

527 *Start* element represents a start event of orders, processes or operations. In case of order, this element
528 represents an event at the earliest start time of corresponding operations.

529 **5.2.2 End element**

530 *End* element represents an end event of orders, processes or operations. In case of order, this element
531 represents an event at the latest end time of corresponding operations.

532 **5.2.3 Event element**

533 *Event* element represents an event associated with a customer, supplier, item, resource, process or
534 operation. Event brings any action or any status change at a specific time point. In general, the status
535 value of item or resource changes discontinuously before the event.

536

537

6 Accounting Elements

538

6.1 Structure of Accounting element

539

Accounting element represents any accounting information such as profit revenue and cost spending.

540

Price and cost associated with goods and services are the target of the elements. The type of this

541

element SHOULD be represented with the following XML schema and SHOULD fulfill the following

542

constraints.

543

544

```
<xsd:complexType name="AccountingType">
  <xsd:sequence>
    <xsd:element ref="Priority" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Display" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Description" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Author" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Date" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Qty" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Char" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Time" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:string"/>
  <xsd:attribute name="key" type="xsd:long"/>
  <xsd:attribute name="name" type="xsd:string"/>
  <xsd:attribute name="type" type="xsd:string"/>
  <xsd:attribute name="status" type="xsd:string"/>
  <xsd:attribute name="value" type="xsd:string"/>
  <xsd:attribute name="condition" type="xsd:string"/>
  <xsd:attribute name="apply" type="xsd:string"/>
</xsd:complexType>
```

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

- *id* attribute SHOULD represent an identifier of this element.

566

- *key* attribute SHOULD represent a key used in the local applications.

567

- *name* attribute SHOULD represent the name of this element.

568

- *type* attribute SHOULD represent the division of this element.

569

- *status* attribute SHOULD represent the status of this element.

570

- *apply* attribute SHOULD represent application type of this element. The condition of this element is exclusive, if the value is "exclusive".

571

572

- *condition* attribute SHOULD represent the condition of this element.

573

- *value* attribute SHOULD represent the qualitative value of this element.

574

575

- *Priority* element SHOULD represent the priority of this element.

576

- *Display* element SHOULD represent how to display this element.

577

- *Description* element SHOULD represent the description of this element.

578

- *Author* element SHOULD represent the author of this element information.

579

- *Date* element SHOULD represent the date of this element information.

580

- *Qty* element SHOULD represent the quantitative value of this element.

581

- *Char* element SHOULD represent the qualitative value of this element.

582

- *Time* element SHOULD represent the temporal value of this element.

583

584 **6.2 List of accounting elements**

585 For accounting elements, *Price* element and *Cost* element are defined in this specification. The type of
586 this element SHOULD be represented with the following XML schema.

587

```
588 <xsd:element name="Price" type="AccountingType"/>  
589 <xsd:element name="Cost" type="AccountingType"/>
```

590

591 **6.2.1 Price element**

592 *Price* element represents a price. This element can be used to represent price information of primitive
593 element and some properties.

594 **6.2.2 Cost element**

595 *Cost* element represents a cost. This element can be used to represent cost information of primitive
596 element and some properties.

597

598

7 Administrative Elements

599

7.1 Structure of Administrative Elements

600 Administrative elements represent any administrative information, which is not the main body of the
601 problem domain but the information how to deal with the domain information. The type of this element
602 SHOULD be represented with the following XML schema and SHOULD fulfill the following constraints.

603

```
604 <xsd:complexType name="AdministrativeType">  
605 <xsd:sequence>  
606 <xsd:element ref="Qty" minOccurs="0" maxOccurs="unbounded"/>  
607 <xsd:element ref="Char" minOccurs="0" maxOccurs="unbounded"/>  
608 <xsd:element ref="Time" minOccurs="0" maxOccurs="unbounded"/>  
609 </xsd:sequence>  
610 <xsd:attribute name="name" type="xsd:string"/>  
611 <xsd:attribute name="type" type="xsd:string"/>  
612 <xsd:attribute name="status" type="xsd:string"/>  
613 <xsd:attribute name="apply" type="xsd:string"/>  
614 <xsd:attribute name="condition" type="xsd:string"/>  
615 <xsd:attribute name="value" type="xsd:string"/>  
616 </xsd:complexType>
```

617

- 618 • *name* attribute SHOULD represent the name of this element.
- 619 • *type* attribute SHOULD represent the division of this element.
- 620 • *status* attribute SHOULD represent the status of this element.
- 621 • *apply* attribute SHOULD represent application type of this element. The condition of this element is
622 exclusive, if the value is “exclusive”.
- 623 • *condition* attribute SHOULD represent the condition of this element.
- 624 • *value* attribute SHOULD represent the qualitative value of this element.
- 625
- 626 • *Qty* element SHOULD represent the quantitative value of this element.
- 627 • *Char* element SHOULD represent the qualitative value of this element.
- 628 • *Time* element SHOULD represent the temporal value of this element.

629

7.2 List of Administrative Elements

631 For administrative elements, *Priority*, *Display*, *Description*, *Author* and *Date* elements are defined in this
632 specification. The type of this element SHOULD be represented with the following XML schema.

633

```
634 <xsd:element name="Priority" type="AdministrativeType"/>  
635 <xsd:element name="Display" type="AdministrativeType"/>  
636 <xsd:element name="Description" type="AdministrativeType"/>  
637 <xsd:element name="Author" type="AdministrativeType"/>  
638 <xsd:element name="Date" type="AdministrativeType"/>
```

639

640 **7.2.1 Priority element**

641 *Priority* element represents the priority of the primitive element or the parent element. This information is
642 used to make a decision for planning or scheduling.

643 **7.2.2 Display element**

644 *Display* element is an element to set how to display the parent element. This element can specify colors
645 or display locations on the screen.

646 **7.2.3 Description element**

647 *Description* element is an element to set an optional comment of the parent element. The comment data
648 type is a character string.

649 **7.2.4 Author element**

650 *Author* element represents the author and its related information such as the authoring date. This
651 information is not about the target domain model, but information processing model.

652 **7.2.5 Date element**

653 *Date* element is an element that shows the creation date, expire date, revising date, and so forth. This
654 information is for administrative use of the domain model.

655

8 Data Elements

656

8.1 Qty element

657

Qty element SHOULD represent quantitative information. This element can be used to represent the quantitative numerical data by decimal type data format. Unit of the value can be set in this element, and representation of fraction is available. The type of this element SHOULD be represented with the following XML schema and SHOULD fulfill the following constraints.

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

```
<xsd:element name="Qty">
  <xsd:complexType>
    <xsd:attribute name="name" type="xsd:string"/>
    <xsd:attribute name="type" type="xsd:string"/>
    <xsd:attribute name="status" type="xsd:string"/>
    <xsd:attribute name="apply" type="xsd:string"/>
    <xsd:attribute name="condition" type="xsd:string"/>
    <xsd:attribute name="value" type="xsd:decimal"/>
    <xsd:attribute name="count" type="xsd:long"/>
    <xsd:attribute name="unit" type="xsd:string"/>
    <xsd:attribute name="base" type="xsd:decimal"/>
  </xsd:complexType>
</xsd:element>
```

675

676

- *name* attribute SHOULD represent the name of this element.

677

- *type* attribute SHOULD represent the division of this element.

678

- *status* attribute SHOULD represent the status of this element.

679

- *apply* attribute SHOULD represent application type of this element. The condition of this element is exclusive, if the value is “exclusive”.

680

681

- *condition* attribute SHOULD represent the condition of this element.

682

- *value* attribute SHOULD represent the content corresponding to the qty element.

683

- *count* attribute SHOULD represent the countable value of this element.

684

- *unit* attribute SHOULD represent the type of currency unit data of this element.

685

- *base* attribute SHOULD represent the base data of this element. The value of the “value” attribute is divided with this value.

686

687

688 Example: 1/3 meters

689

```
<Qty value="1" unit="m" base="3"/>
```

690

Example: 3 weeks (discrete time scale)

691

```
<Qty count="3" unit="week" />
```

692

8.2 Char element

694

Char element SHOULD represent character data. This element can be used to represent a qualitative value of specification or a value of location. The type of this element SHOULD be represented with the following XML schema and SHOULD fulfill the following constraints.

695

696

697

698

```
<xsd:element name="Char">
```

```

699 <xsd:complexType>
700 <xsd:attribute name="name" type="xsd:string"/>
701 <xsd:attribute name="type" type="xsd:string"/>
702 <xsd:attribute name="status" type="xsd:string"/>
703 <xsd:attribute name="apply" type="xsd:string"/>
704 <xsd:attribute name="condition" type="xsd:string"/>
705 <xsd:attribute name="value" type="xsd:string"/>
706 <xsd:attribute name="count" type="xsd:long"/>
707 <xsd:attribute name="unit" type="xsd:string"/>
708 <xsd:attribute name="base" type="xsd:string"/>
709 </xsd:complexType>
710 </xsd:element>

```

- 711
- 712 • *name* attribute SHOULD represent the name of this element.
 - 713 • *type* attribute SHOULD represent the division of this element.
 - 714 • *status* attribute SHOULD represent the status of this element.
 - 715 • *apply* attribute SHOULD represent application type of this element. The condition of this element is
 - 716 exclusive, if the value is “exclusive”.
 - 717 • *condition* attribute SHOULD represent the condition of this element.
 - 718 • *value* attribute SHOULD represent the content corresponding to the qty element.
 - 719 • *count* attribute SHOULD represent the countable value of this element.
 - 720 • *unit* attribute SHOULD represent the type of currency unit data of this element.
 - 721 • *base* attribute SHOULD represent the base data of this element. The value of the “value” attribute is
 - 722 divided with this value.

723

724 8.3 Time element

725 *Time* element SHOULD represent a specific time. Time is represented by a continuous time scale, or a

726 specific discrete time scale. The type of this element SHOULD be represented with the following XML

727 schema and SHOULD fulfill the following constraints.

728

```

729 <xsd:element name="Time">
730 <xsd:complexType>
731 <xsd:attribute name="name" type="xsd:string"/>
732 <xsd:attribute name="type" type="xsd:string"/>
733 <xsd:attribute name="status" type="xsd:string"/>
734 <xsd:attribute name="apply" type="xsd:string"/>
735 <xsd:attribute name="condition" type="xsd:string"/>
736 <xsd:attribute name="value" type="xsd:dateTime"/>
737 <xsd:attribute name="count" type="xsd:long"/>
738 <xsd:attribute name="unit" type="xsd:string"/>
739 <xsd:attribute name="base" type="xsd:dateTime"/>
740 </xsd:complexType>
741 </xsd:element>

```

- 742
- 743 • *name* attribute SHOULD represent the name of this element.
 - 744 • *type* attribute SHOULD represent the division of this element.
 - 745 • *status* attribute SHOULD represent the status of this element.
 - 746 • *apply* attribute SHOULD represent application type of this element. The condition of this element is
 - 747 exclusive, if the value is “exclusive”.
 - 748 • *condition* attribute SHOULD represent the condition of this element.
 - 749 • *value* attribute SHOULD represent the content corresponding to the qty element.

- 750 • *count* attribute SHOULD represent the countable value of this element.
- 751 • *unit* attribute SHOULD represent the type of currency unit data of this element.
- 752 • *base* attribute SHOULD represent the base data of this element. The value of the “value” attribute is
753 divided with this value.

754

755 Example: noon on May 13th, 2005

756 `<Time value="2005-05-13T12:00:00"/>`

757 Example: 2 months later since the present month (May, 2005) (discrete time scale)

758 `<Time count="2" unit="month" base="2005-05-01T00:00:00"/>`

759

760 9 Conformance

761 A document or part of document confirms OASIS PPS Core Elements if all elements in the artifact are
762 consistent with the normative text of this specification, and the document can be processed properly with
763 the XML schema that can be downloaded from the following URI.

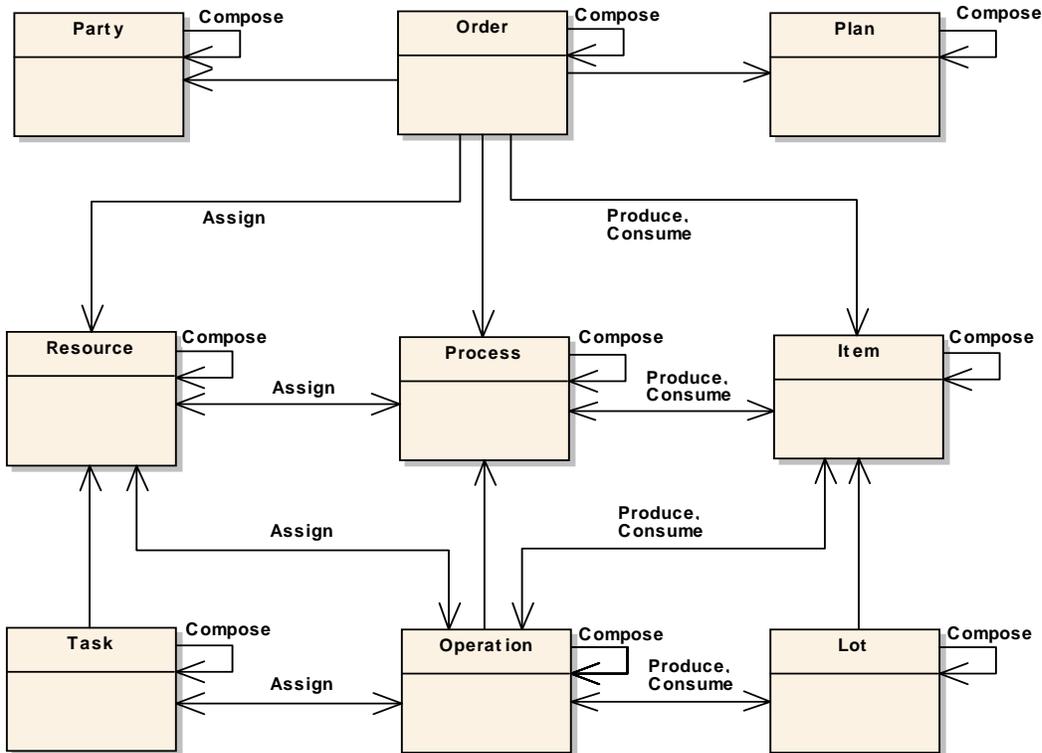
764

765 <http://docs.oasis-open.org/pps/v1.0/pps-schema-1.0.xsd>

766

767 **A. Object Class diagram**

768 Figure A-1 shows the structure of primitive objects in this specification with a UML class diagram. Each
769 object corresponds to each XML element. In this figure, arrows represent relative information between the
770 source and destination objects. When an arrow has role names, it corresponds to an independent XML
771 element in the specification. This figure doesn't include all the information of XML schema but the partial
772 information of the primitive elements.
773



774
775 Figure A-1: Primitive objects for representing planning and scheduling problems
776

777

B. Cross reference of elements

778 Table B-1 shows the relations between elements. The row headers represent parent elements and the
 779 column headers represent child elements. Symbol * in the table means 0 or more than 0 element can be
 780 described.

781

782 Table B-1 Element and sub-element relations

	Compose	Produce	Consume	Assign	Relation	Location	Capacity	Progress	Spec	Start	End	Event	Price	Cost	Priority	Display	Description	Author	Date	Qty	Char	Time
Party	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*			
Plan	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*			
Order	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*			
Item	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*			
Resource	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*			
Process	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*			
Lot	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*			
Task	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*			
Operation	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*			
Compose						*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Produce						*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Consume						*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Assign						*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Relation						*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Location										*	*	*	*	*	*	*	*	*	*	*	*	*
Capacity										*	*	*	*	*	*	*	*	*	*	*	*	*
Progress										*	*	*	*	*	*	*	*	*	*	*	*	*
Spec										*	*	*	*	*	*	*	*	*	*	*	*	*
Start															*	*	*	*	*	*	*	*
End															*	*	*	*	*	*	*	*
Event															*	*	*	*	*	*	*	*
Price															*	*	*	*	*	*	*	*
Cost															*	*	*	*	*	*	*	*
Priority																				*	*	*
Display																				*	*	*
Description																				*	*	*
Author																				*	*	*
Date																				*	*	*
Qty																						
Char																						
Time																						

783

784

785

786

787 The following table B-2 shows the correspondence between elements and attributes. The row headers
 788 show the element name, and the column headers show attribute the name. The characters in the table
 789 represent data types. The character in the table are used as follows: "U" denotes identification character
 790 of element, "P" denotes an identification character of referencing elements, "S" denotes the character
 791 string, "D" denotes a decimal number, "N" denotes an integer number and "T" for date time. Boldface
 792 means required information.

793

794 Table B-2 Element and attribute relations

	id	key	name	parent	type	status	apply	condition	value	count	unit	base	party	plan	order	item	resource	process	lot	task	operation
Party	U	N	S	P	S	S							P	P	P	P	P	P	P	P	P
Plan	U	N	S	P	S	S							P	P	P	P	P	P	P	P	P
Order	U	N	S	P	S	S							P	P	P	P	P	P	P	P	P
Item	U	N	S	P	S	S							P	P	P	P	P	P	P	P	P
Resource	U	N	S	P	S	S							P	P	P	P	P	P	P	P	P
Process	U	N	S	P	S	S							P	P	P	P	P	P	P	P	P
Lot	U	N	S	P	S	S							P	P	P	P	P	P	P	P	P
Task	U	N	S	P	S	S							P	P	P	P	P	P	P	P	P
Operation	U	N	S	P	S	S							P	P	P	P	P	P	P	P	P
Compose	U	N	S		S	S	S						P	P	P	P	P	P	P	P	P
Produce	U	N	S		S	S	S						P	P	P	P	P	P	P	P	P
Consume	U	N	S		S	S	S						P	P	P	P	P	P	P	P	P
Assign	U	N	S		S	S	S						P	P	P	P	P	P	P	P	P
Relation	U	N	S		S	S	S						P	P	P	P	P	P	P	P	P
Location	U	N	S		S	S	S														
Capacity	U	N	S		S	S	S														
Progress	U	N	S		S	S	S														
Spec	U	N	S		S	S	S														
Start	U	N	S		S	S	S	S	S												
End	U	N	S		S	S	S	S	S												
Event	U	N	S		S	S	S	S	S												
Price	U	N	S		S	S	S	S	S												
Cost	U	N	S		S	S	S	S	S												
Priority			S		S	S	S	S	S												
Display			S		S	S	S	S	S												
Description			S		S	S	S	S	S												
Author			S		S	S	S	S	S												
Date			S		S	S	S	S	S												
Qty			S		S	S	S	S	D	N	S	D									
Char			S		S	S	S	S	S	N	S	S									
Time			S		S	S	S	S	T	N	S	T									

795

796

C. Acknowledgements

797 The following individuals have participated in the creation of this specification and are gratefully
798 acknowledged:

799 **Participants:**

800 Shinya Matsukawa, Hitachi
801 Tomohiko Maeda, Fujitsu
802 Masahiro Mizutani, Unisys Corporation
803 Akihiro Kawauchi, Individual Member
804 Yuto Banba, PSLX Forum
805 Osamu Sugi, PSLX Forum
806 Hideichi Okamune, PSLX Forum
807 Hiroshi Kojima, PSLX Forum
808 Ken Nakayama, Hitachi
809 Yukio Hamaguchi, Hitachi
810 Tomoichi Sato, Individual
811 Hiroaki Sasaki, Individual

812 **D. Revision History**

813

Revision	Date	Editor	Changes Made
01		Y.Nishioka	

814

815