



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

Committee Specification 01

11 April 2008

Specification URIs:

<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>

Previous Version:

<http://docs.oasis-open.org/pps/v1.0/pr02/pps-core-elements-1.0-pr02.doc>
<http://docs.oasis-open.org/pps/v1.0/pr02/pps-core-elements-1.0-pr02.html>
<http://docs.oasis-open.org/pps/v1.0/pr02/pps-core-elements-1.0-pr02.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/pps/ns>

Abstract:

OASIS PPS (Production Planning and Scheduling) Standard deals with problems in all manufacturing companies who want to have a sophisticated information system for production planning and scheduling. PPS standard provides XML schema and communication protocols for information exchange among manufacturing application programs in the web-services environment. This document especially focuses on information model of core elements in the production planning and scheduling domain. Since the elements have been designed without specific 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. 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	Conformance.....	6
1.5	Terms and definitions.....	7
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	Order element.....	10
2.2.3	Item element.....	10
2.2.4	Resource element.....	10
2.2.5	Function element.....	10
2.2.6	Lot element.....	10
2.2.7	Task element.....	11
2.2.8	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
A.	Object Class diagram	27
B.	Cross reference of elements	28
C.	Acknowledgements	30
D.	Revision History.....	31

1 Introduction

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

If information is exchanged between some applications related to Production Planning and Scheduling, 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 or sites and detail scheduling in the individual areas and sites.

This specification doesn't aim at 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] PPS (Production Planning and Scheduling) Part 2: Transaction Messages, Version 1.0, Public Review Draft 01, <http://www.oasis-open.org/committees/pps/>
- [PPS03] PPS (Production Planning and Scheduling) Part 3: Profile Specifications, Version 1.0, Public Review Draft 01, <http://www.oasis-open.org/committees/pps/>

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 Conformance

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

<http://docs.oasis-open.org/pps/v1.0/pps-core-elements.xsd>

43 1.5 Terms and definitions

44 **plan**

45 representation unit for intensive information of related orders corresponding to a specific period
46 on a discrete time scale, or calculated information on the schedule under the related orders. It
47 may become actual results of the progress according to the timing of the action, whether it is in
48 the past or future.

49 **order**

50 unit of requirement describing concrete item, resource or operation in a specific place at a
51 specific time. It can also represent the results to the request.

52 **party**

53 customer who is a sender of an order and has a demand to make a decision, or supplier thow is a
54 receiver in case that a decision-maker sends the demand that can't be handled inside.

55 **item**

56 object to be produced or consumed by production activity. The quantity or the quality of item is
57 changed by means of the production activity.

58 Example : product, parts, module, unit, work in process, materials

59 **resource**

60 object that can provide essential function for production activity. The capacity of function is used
61 during production activity but is available again after finishing production.

62 Example: equipment, machine, device, labor, tool

63 **process**

64 element of production activity indicating a concrete production method. It has duration and gives
65 the added value to a produced item. One function may have two or more functions in a more
66 detail unit inside.

67 **lot**

68 instance of specific item that exists in a specific place at a specific time. Generally the specific
69 time corresponds to start or end of an operation, and the specific quantity is equal to the quantity
70 of item produced or consumed by the operation.

71 **task**

72 unit of necessity to execute a specific function at a specific time, indicating the volume of used
73 capability provided by the applicable resource.

74 Notes: Task represents either the capacity value provided by resource at a specific time point or
75 the aggregated total value of capacity provided by resource during specific duration.

76 **operation**

77 actual processing element to be executed by a specific task and to produce or consume a
78 specific lot. It is a concrete instance of function in production activity.

79

2 Primitive Elements

80

2.1 Structure of primitive elements

81 Primitive elements are the minimum series of element that corresponds to the most basic domain objects.
82 The type of this element SHOULD be represented with the following XML schema and SHOULD fulfill the
83 following constraints.

84

```

85 <xsd:complexType name="PrimitiveType">
86   <xsd:sequence>
87     <xsd:element ref="Compose" minOccurs="0" maxOccurs="unbounded"/>
88     <xsd:element ref="Produce" minOccurs="0" maxOccurs="unbounded"/>
89     <xsd:element ref="Consume" minOccurs="0" maxOccurs="unbounded"/>
90     <xsd:element ref="Assign" minOccurs="0" maxOccurs="unbounded"/>
91     <xsd:element ref="Relation" minOccurs="0" maxOccurs="unbounded"/>
92     <xsd:element ref="Location" minOccurs="0" maxOccurs="unbounded"/>
93     <xsd:element ref="Capacity" minOccurs="0" maxOccurs="unbounded"/>
94     <xsd:element ref="Progress" minOccurs="0" maxOccurs="unbounded"/>
95     <xsd:element ref="Spec" minOccurs="0" maxOccurs="unbounded"/>
96     <xsd:element ref="Start" minOccurs="0" maxOccurs="unbounded"/>
97     <xsd:element ref="End" minOccurs="0" maxOccurs="unbounded"/>
98     <xsd:element ref="Event" minOccurs="0" maxOccurs="unbounded"/>
99     <xsd:element ref="Price" minOccurs="0" maxOccurs="unbounded"/>
100    <xsd:element ref="Cost" minOccurs="0" maxOccurs="unbounded"/>
101    <xsd:element ref="Priority" minOccurs="0" maxOccurs="unbounded"/>
102    <xsd:element ref="Display" minOccurs="0" maxOccurs="unbounded"/>
103    <xsd:element ref="Description" minOccurs="0" maxOccurs="unbounded"/>
104    <xsd:element ref="Author" minOccurs="0" maxOccurs="unbounded"/>
105    <xsd:element ref="Date" minOccurs="0" maxOccurs="unbounded"/>
106  </xsd:sequence>
107  <xsd:attribute name="id" type="xsd:string" use="required"/>
108  <xsd:attribute name="key" type="xsd:long"/>
109  <xsd:attribute name="name" type="xsd:string"/>
110  <xsd:attribute name="parent" type="xsd:string"/>
111  <xsd:attribute name="type" type="xsd:string"/>
112  <xsd:attribute name="status" type="xsd:string"/>
113  <xsd:attribute name="party" type="xsd:string"/>
114  <xsd:attribute name="plan" type="xsd:string"/>
115  <xsd:attribute name="order" type="xsd:string"/>
116  <xsd:attribute name="item" type="xsd:string"/>
117  <xsd:attribute name="resource" type="xsd:string"/>
118  <xsd:attribute name="process" type="xsd:string"/>
119  <xsd:attribute name="lot" type="xsd:string"/>
120  <xsd:attribute name="task" type="xsd:string"/>
121  <xsd:attribute name="operation" type="xsd:string"/>
122 </xsd:complexType>

```

123

- 124 • *id* attribute SHOULD represent an identifier of this element.
- 125 • *key* attribute SHOULD represent a key used in the local applications.
- 126 • *name* attribute SHOULD represent the name of this element.
- 127 • *parent* attribute SHOULD represent the identifier of the inherited element of this element.
- 128 • *type* attribute SHOULD represent the division of this element.
- 129 • *status* attribute SHOULD represent the status of this element.
- 130 • *party* attribute SHOULD represent an identifier of the party associated with this element.
- 131 • *plan* attribute SHOULD represent the identifier of the plan associated with this element.
- 132 • *order* attribute SHOULD represent the identifier of the order associated with this element.

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

159 2.2 List of primitive elements

160 This standard defines nine primitive elements: *Party*, *Plan*, *Order*, *Item*, *Resource*, *Process*, *Lot*, *Task*,
 161 and *Operation*. The type of this element SHOULD be represented with the following XML schema.

162

```

163 <xsd:element name="Party" type="PrimitiveType"/>
164 <xsd:element name="Plan" type="PrimitiveType"/>
165 <xsd:element name="Order" type="PrimitiveType"/>
166 <xsd:element name="Item" type="PrimitiveType"/>
167 <xsd:element name="Resource" type="PrimitiveType"/>
168 <xsd:element name="Process" type="PrimitiveType"/>
169 <xsd:element name="Lot" type="PrimitiveType"/>
170 <xsd:element name="Task" type="PrimitiveType"/>
171 <xsd:element name="Operation" type="PrimitiveType"/>

```

172

173 2.2.1 Party element

174 *Party* element represents customer and supplier. Customer is an object that requests some products or
 175 services from the enterprise. Such requests are sent to a person in charge of production planning or

176 scheduling. Supplier is an object providing some products or services to the enterprise. Supplier
177 receives orders form the enterprise and provides additional item, resource or function to the enterprise.

178 2.2.2 Order element

179 *Order* element represents an object of information produced to request some products or services. Order
180 is a source to finally dispatch a schedule to the plant floor. Orders can be divided into an item order, a
181 resource order and a function order according to the type of request.

182

183 Example: Ten of "A" products are requested.

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

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

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

191 Example: Switching operation is requested two times.

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

195 Example: An Order, which consist of ten "A" products and five "B" products, is total 3,000 yen.

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

207

208 2.2.3 Item element

209 *Item* element represents a product, component, parts, work in process (WIP), raw material and other
210 items. Item is produced by any function, and after that, it is consumed by another function.

211 2.2.4 Resource element

212 *Resource* element represents a resource. Resource is an object enabling production, transportation,
213 storage, inspection and other various services. Resource is assigned to an operation after considering its
214 capacity.

215 2.2.5 Function element

216 *Function* element represents a function. Function is a unit of activities in production process, and
217 produces and consumes items by being executed for certain duration.

218 2.2.6 Lot element

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

222 2.2.7 Task element

223 *Task* element represents a task. Task is an object showing the usage of a specific resource for a specific
224 period. Schedule requests a task for each resource assigned to execute the operation.

225

226 Example: Task corresponding to the quantity of work that 3 labors work for 2 days

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

233

234 2.2.8 Operation element

235 *Operation* element represents an activity to dispatch. Operation makes a function executed at a specific
236 place on a plant floor for a specific time. Operation is associated with a specific lot or task by executing
237 the activity.

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* SHOULD represent that 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 standard defines five relational elements: *Compose*, *Produce*, *Consume*, *Assign*, and *Relation*. The
 318 type of this element SHOULD be represented with the following XML schema.

319

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

325

326 3.2.1 Compose element

327 *Compose* element defines a hierarchical relation between two same primitive elements. This element
 328 can represent that the object referred to in this element composes or be composed by the upper level
 329 element.

330

331 Example: Product “A” group includes product “A1” and product “A2”.

```
332 <Item id="A">
```

```
333 <Compose type="pps:child" item="A1"/>
334 <Compose type="pps:child" item="A2"/>
335 </Item>
```

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

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

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

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

346

347 3.2.2 Produce element

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

351 3.2.3 Consume element

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

355 3.2.4 Assign element

356 *Assign* element defines a relation between function and resource, or a relation between operation and
357 task. This element can show the quantity of the resource or task used by the function or operation
358 respectively, or how many resources or tasks are used by the function or operation respectively.

359 3.2.5 Relation element

360 *Relation* element can show that one element in Primitive elements has a specific relation to other
361 elements. This element can additionally define relational classes between primitive elements. The type
362 of this element SHOULD be represented with the following XML schema and SHOULD fulfill the following
363 constraints.

364

4 Specific Elements

365

4.1 Structure of specific element

366

Specific elements are defined to represent any properties. These elements MAY have multiple instances with its 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

```
<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>
```

393

394

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

395

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

396

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

397

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

398

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

399

- *apply* attribute SHOULD represent that the specifying spec is relative, if the value is “*relative*” ..

400

401

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

402

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

403

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

404

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

405

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

406

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

407

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

408

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

409

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

410

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

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

415 4.2 List of specific elements

416 For specific elements, this standard has four elements: *Location*, *Capacity*, *Progress*, and *Spec*. The type
417 of this element SHOULD be represented with the following XML schema.

418

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

423

424 4.2.1 Location element

425 *Location* element represents a location. When the expression of location has structure, multiple values
426 can be set by specifying different names of the data. Change of the location can be represented time-
427 dependently.

428

429 Example: Customer's address

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

437

438 4.2.2 Capacity element

439 *Capacity* element represents volume of capability of Resource, Item and Process. For Resource, it shows
440 available summary of corresponding tasks. For Item, it shows the available summary of Lots. And for
441 Process, it shows available rate of production. All of this information is represented in a time horizon.

442

443 Example: Inventory level of "material01"

```
444 <Item id="material01">
445 <Capacity><Qty value="150"/></Capacity>
446 </Item>
```

447 Example: Temporal change of the material

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

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

```
453 <Item id="material01">
454 <Location value="storage01"/>
455 <Capacity><Qty value="150"/></Capacity>
456 </Item>
```

457

458 **4.2.3 Progress element**

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

462 **4.2.4 Spec element**

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

468

5 Eventual Elements

469

5.1 Structure of eventual element

470

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.

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

```
<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>
```

494

495

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

496

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

497

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

498

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

499

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

500

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

501

502

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

503

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

504

505

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

506

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

507

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

508

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

509

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

510

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

511

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

512

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

513

514 **5.2 List of eventual elements**

515 This standard defines three eventual elements: *Start*, *End*, and *Event*. The *Start* and *End* is special cases
516 of *Event* element. The type of this element SHOULD be represented with the following XML schema.

517

```
518 <xsd:element name="Start" type="EventualType"/>  
519 <xsd:element name="End" type="EventualType"/>  
520 <xsd:element name="Event" type="EventualType"/>
```

521

522 **5.2.1 Start element**

523 *Start* element represents a start event of order or operation. In case of order, this element represents an
524 event at the earliest start time of corresponding operations.

525 **5.2.2 End element**

526 *End* element represents an end event of order or operation. In case of order, this element represents an
527 event at the latest end time of corresponding operations.

528 **5.2.3 Event element**

529 *Event* element represents an event attending with a customer, supplier, item, resource, function or
530 operation. Event brings any action or any status change at a specific time point. In general, the status
531 value of item or resource changes discontinuously before the event.

532

533

6 Accounting Elements

534

6.1 Structure of Accounting element

535

Accounting element represents any accounting information such as income and spending. Price and cost of goods and services are the target of the elements. The type of this element SHOULD be represented with the following XML schema and SHOULD fulfill the following constraints.

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

```
<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="apply" type="xsd:string"/>
  <xsd:attribute name="condition" type="xsd:string"/>
  <xsd:attribute name="value" type="xsd:decimal"/>
</xsd:complexType>
```

559

560

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

561

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

562

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

563

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

564

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

565

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

566

567

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

568

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

569

570

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

571

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

572

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

573

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

574

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

575

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

576

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

577

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

578

579 **6.2 List of accounting elements**

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

582

```
583 <xsd:element name="Price" type="AccountingType"/>  
584 <xsd:element name="Cost" type="AccountingType"/>
```

585

586 **6.2.1 Price element**

587 *Price* element represents a price. This element can be used to represent price information of primitive
588 element and some properties. The currency unit can be set if necessary.

589 **6.2.2 Cost element**

590 *Cost* element represents a cost. This element can be used to represent cost information of primitive
591 element and some properties. The currency unit can be set if necessary.

592

593

7 Administrative Elements

594

7.1 Structure of Administrative Elements

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

598

```
599 <xsd:complexType name="AdministrativeType">  
600 <xsd:sequence>  
601 <xsd:element ref="Qty" minOccurs="0" maxOccurs="unbounded"/>  
602 <xsd:element ref="Char" minOccurs="0" maxOccurs="unbounded"/>  
603 <xsd:element ref="Time" minOccurs="0" maxOccurs="unbounded"/>  
604 </xsd:sequence>  
605 <xsd:attribute name="name" type="xsd:string"/>  
606 <xsd:attribute name="type" type="xsd:string"/>  
607 <xsd:attribute name="status" type="xsd:string"/>  
608 <xsd:attribute name="apply" type="xsd:string"/>  
609 <xsd:attribute name="condition" type="xsd:string"/>  
610 <xsd:attribute name="value" type="xsd:string"/>  
611 </xsd:complexType>
```

612

- 613 • *name* attribute SHOULD represent the name of this element.
- 614 • *type* attribute SHOULD represent the division of this element.
- 615 • *status* attribute SHOULD represent the status of this element.
- 616 • *apply* attribute SHOULD represent that the condition of this element is exclusive, if the value is
617 “exclusive”.
- 618 • *condition* attribute SHOULD represent the condition of this element.
- 619 • *value* attribute SHOULD represent the content corresponding to the qty element.
- 620
- 621 • *Qty* element SHOULD represent the quantitative value of this element.
- 622 • *Char* element SHOULD represent the qualitative value of this element.
- 623 • *Time* element SHOULD represent the temporal value of this element.

624

7.2 List of Administrative Elements

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

628

```
629 <xsd:element name="Priority" type="AdministrativeType"/>  
630 <xsd:element name="Display" type="AdministrativeType"/>  
631 <xsd:element name="Description" type="AdministrativeType"/>  
632 <xsd:element name="Author" type="AdministrativeType"/>  
633 <xsd:element name="Date" type="AdministrativeType"/>
```

634

635 **7.2.1 Priority element**

636 *Priority* element represents the priority of primitive elements or relational elements. This information is
637 used to make a decision for planning or scheduling.

638 **7.2.2 Display element**

639 *Display* element is an element to set how to display primitive elements. This element can specify colors or
640 display locations on the screen.

641 **7.2.3 Description element**

642 *Description* element is an element to set an optional comment to some elements of primitive elements.
643 The comment data type is a character string.

644 **7.2.4 Author element**

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

647 **7.2.5 Date element**

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

650

8 Data Elements

651

8.1 Qty element

652

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.

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

```
<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>
```

670

671

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

672

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

673

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

674

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

675

676

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

677

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

678

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

679

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

680

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

681

682

683 Example: 1/3 meters

684

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

685

Example: 3 weeks (discrete time scale)

686

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

687

688

8.2 Char element

689

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.

690

691

692

```

693 <xsd:element name="Char">
694 <xsd:complexType>
695 <xsd:attribute name="name" type="xsd:string"/>
696 <xsd:attribute name="type" type="xsd:string"/>
697 <xsd:attribute name="status" type="xsd:string"/>
698 <xsd:attribute name="apply" type="xsd:string"/>
699 <xsd:attribute name="condition" type="xsd:string"/>
700 <xsd:attribute name="value" type="xsd:string"/>
701 <xsd:attribute name="count" type="xsd:long"/>
702 <xsd:attribute name="unit" type="xsd:string"/>
703 <xsd:attribute name="base" type="xsd:string"/>
704 </xsd:complexType>
705 </xsd:element>

```

- 706
- 707 • *name* attribute SHOULD represent the name of this element.
 - 708 • *type* attribute SHOULD represent the division of this element.
 - 709 • *status* attribute SHOULD represent the status of this element.
 - 710 • *apply* attribute SHOULD represent that the condition of this element is exclusive, if the value is
711 “exclusive”.
 - 712 • *condition* attribute SHOULD represent the condition of this element.
 - 713 • *value* attribute SHOULD represent the content corresponding to the qty element.
 - 714 • *count* attribute SHOULD represent the countable value of this element.
 - 715 • *unit* attribute SHOULD represent the type of currency unit data of this element.
 - 716 • *base* attribute SHOULD represent the base data of this element. The value of the “value” attribute is
717 divided with this value.

718

719 8.3 Time element

720 *Time* element SHOULD represent a specific time. Time is represented by a continuous time scale, or a
721 specific discrete time scale. The type of this element SHOULD be represented with the following XML
722 schema and SHOULD fulfill the following constraints.

723

```

724 <xsd:element name="Time">
725 <xsd:complexType>
726 <xsd:attribute name="name" type="xsd:string"/>
727 <xsd:attribute name="type" type="xsd:string"/>
728 <xsd:attribute name="status" type="xsd:string"/>
729 <xsd:attribute name="apply" type="xsd:string"/>
730 <xsd:attribute name="condition" type="xsd:string"/>
731 <xsd:attribute name="value" type="xsd:dateTime"/>
732 <xsd:attribute name="count" type="xsd:long"/>
733 <xsd:attribute name="unit" type="xsd:string"/>
734 <xsd:attribute name="base" type="xsd:dateTime"/>
735 </xsd:complexType>
736 </xsd:element>

```

- 737
- 738 • *name* attribute SHOULD represent the name of this element.
 - 739 • *type* attribute SHOULD represent the division of this element.
 - 740 • *status* attribute SHOULD represent the status of this element.
 - 741 • *apply* attribute SHOULD represent that the condition of this element is exclusive, if the value is
742 “exclusive”.
 - 743 • *condition* attribute SHOULD represent the condition of this element.

- 744 • *value* attribute SHOULD represent the content corresponding to the qty element.
- 745 • *count* attribute SHOULD represent the countable value of this element.
- 746 • *unit* attribute SHOULD represent the type of currency unit data of this element.
- 747 • *base* attribute SHOULD represent the base data of this element. The value of the “value” attribute is
- 748 divided with this value.

749

750 Example: noon on May 13th, 2005

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

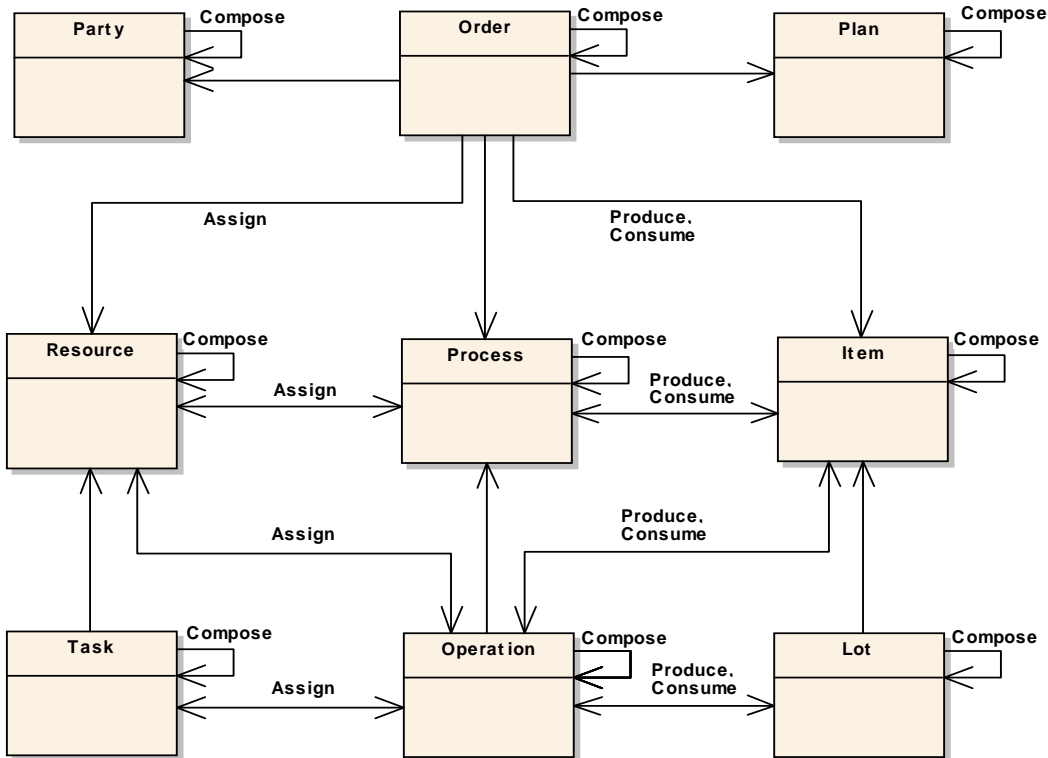
752 Example: 2 months later since the present month (May, 2005) (discrete time scale)

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

754

755 **A. Object Class diagram**

756 Figure A-1 shows the structure of primitive objects in this specification with UML class diagram. Each
757 object corresponds to each XML element. In this figure, arrows represent the source and destination
758 between the referring objects. When an arrow has role names, it corresponds to an independent XML
759 element associating the two objects. This figure doesn't include all the information of XML schema but the
760 information on primitive elements.
761



762
763 Figure A-1: Primitive objects for representing planning and scheduling problems
764

765 **B. Cross reference of elements**

766 The below table B-1 shows the relations between elements. The horizontal lines represent parent
 767 elements and the vertical lines represent child elements. Symbol * in the table means 0 or more than 0
 768 element.

769
 770 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																						

771
 772
 773
 774

775 The following table B-2 shows the correspondence between elements and attributes. The horizontal lines
 776 show element names and the vertical lines show attribute names. The characters in the table represent
 777 data types. The letters in the table are used as follows: "U" for identification character of element, "P"
 778 for an identification character of other elements, "S" for the character string, "D" for a decimal number, "N"
 779 for an integer number and "T" for date time. Boldface means required information.

780

781 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	T	T	N	S	T									

782

783

C. Acknowledgements

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

786 **Participants:**

787 Shinya Matsukawa, Hitachi
788 Tomohiko Maeda, Fujitsu
789 Masahiro Mizutani, Unisys Corporation
790 Akihiro Kawauchi, Individual Member
791 Yuto Banba, PSLX Forum
792 Osamu Sugi, PSLX Forum
793 Hideichi Okamune, PSLX Forum
794 Hiroshi Kojima, PSLX Forum
795 Ken Nakayama, Hitachi
796 Yukio Hamaguchi, Hitachi
797 Tomoichi Sato, Individual
798 Hiroaki Sasaki, Individual

799

D. Revision History

800

Revision	Date	Editor	Changes Made

801

802