



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

Public Review Draft 02

24 Oct 2009

Specification URIs:

<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>

Previous Version:

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

Latest Version:

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

42 **1.5 Terms and definitions**

43 **Plan**

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

47 **Order**

48 Unit of requirement describing concrete item, resource or operation in a specific place at a
49 specific time. This can also represent the results to the requirement.

50 **Party**

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

53 **Item**

54 Object to be produced or consumed by production activities. The quantity or the quality of item is
55 changed during the production activity. Examples include product, parts, module, unit, work in
56 process and materials.

57 **Resource**

58 Object that can provide essential function for production activities. The capacity of function is
59 used during production activity, and is available again after finishing the production. Examples
60 include equipment, machine, device, labor and tool.

61 **Process**

62 Segment of production activities indicating a certain production line or method. This takes
63 duration from start time to end time, and gives added value to the producing item. One process
64 may have two or more than two processes detailed in the lower levels.

65 **Lot**

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

69 **Task**

70 Unit of necessity to execute a specific operation at a specific time, indicating the volume of used
71 capability provided by the applicable resource. This can represent both capacity value provided
72 by resource at a specific time point, and aggregated total value of capacity provided by resource
73 during specific duration.

74 **Operation**

75 Actual processing element to be executed by a specific task, and to produce or consume a
76 specific lot. It is a concrete instance of particular processes in production activities.

77

2 Primitive Elements

78

2.1 Structure of primitive elements

79

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

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

```

<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>

```

121

122

123

124

125

126

127

128

129

130

- *id* attribute SHOULD represent an identifier of this element.
- *key* attribute SHOULD represent a key used in the local applications.
- *name* attribute SHOULD represent the name of this element.
- *parent* attribute SHOULD represent the identifier of the inherited element of this element.
- *type* attribute SHOULD represent the division of this element.
- *status* attribute SHOULD represent the status of this element.
- *party* attribute SHOULD represent an identifier of the party associated with this element.
- *plan* attribute SHOULD represent the identifier of the plan associated with this element.
- *order* attribute SHOULD represent the identifier of the order associated with this element.

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

157 2.2 List of primitive elements

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

160

```

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

```

170

171 2.2.1 Party element

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

174 scheduling. Supplier is an object providing some products or services to the enterprise. Supplier
175 receives orders from the enterprise, and provides corresponding items, resources or processes for the
176 enterprise.

177 2.2.2 Plan element

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

181 2.2.3 Order element

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

185

186 Example: Ten "A" products are requested.

```
187 <Order id="Z01" item="A">  
188 <Spec type="quantity"><Qty value="10"/></Spec>  
189 </Order>
```

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

```
191 <Order id="Z02" resource="groupB">  
192 <Spec type="quantity"><Qty value="3"/></Spec>  
193 </Order>
```

194 Example: Switching operation is requested two times.

```
195 <Order id="Z03" process="change01">  
196 <Spec type="quantity"><Qty value="2"/></Spec>  
197 </Order>
```

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

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

210

211 2.2.4 Item element

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

215 2.2.5 Resource element

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

219 **2.2.6 Process element**

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

223 **2.2.7 Lot element**

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

227 **2.2.8 Task element**

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

231

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

```
233 <Task id="T01">  
234 <Capacity type="human"><Qty value="3"/></Capacity>  
235 <Capacity type="duration"><Qty value="2" unit="day" /></Capacity>  
236 </Task>
```

237

238 **2.2.9 Operation element**

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

242

3 Relational Elements

243

3.1 Structure of relational elements

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

247

```
248 <xsd:complexType name="RelationalType">
249   <xsd:sequence>
250     <xsd:element ref="Location" minOccurs="0" maxOccurs="unbounded"/>
251     <xsd:element ref="Capacity" minOccurs="0" maxOccurs="unbounded"/>
252     <xsd:element ref="Progress" minOccurs="0" maxOccurs="unbounded"/>
253     <xsd:element ref="Spec" minOccurs="0" maxOccurs="unbounded"/>
254     <xsd:element ref="Start" minOccurs="0" maxOccurs="unbounded"/>
255     <xsd:element ref="End" minOccurs="0" maxOccurs="unbounded"/>
256     <xsd:element ref="Event" minOccurs="0" maxOccurs="unbounded"/>
257     <xsd:element ref="Price" minOccurs="0" maxOccurs="unbounded"/>
258     <xsd:element ref="Cost" minOccurs="0" maxOccurs="unbounded"/>
259     <xsd:element ref="Priority" minOccurs="0" maxOccurs="unbounded"/>
260     <xsd:element ref="Display" minOccurs="0" maxOccurs="unbounded"/>
261     <xsd:element ref="Description" minOccurs="0" maxOccurs="unbounded"/>
262     <xsd:element ref="Author" minOccurs="0" maxOccurs="unbounded"/>
263     <xsd:element ref="Date" minOccurs="0" maxOccurs="unbounded"/>
264     <xsd:element ref="Qty" minOccurs="0" maxOccurs="unbounded"/>
265     <xsd:element ref="Char" minOccurs="0" maxOccurs="unbounded"/>
266     <xsd:element ref="Time" minOccurs="0" maxOccurs="unbounded"/>
267   </xsd:sequence>
268   <xsd:attribute name="id" type="xsd:string"/>
269   <xsd:attribute name="key" type="xsd:long"/>
270   <xsd:attribute name="name" type="xsd:string"/>
271   <xsd:attribute name="type" type="xsd:string"/>
272   <xsd:attribute name="status" type="xsd:string"/>
273   <xsd:attribute name="apply" type="xsd:string"/>
274   <xsd:attribute name="party" type="xsd:string"/>
275   <xsd:attribute name="plan" type="xsd:string"/>
276   <xsd:attribute name="order" type="xsd:string"/>
277   <xsd:attribute name="item" type="xsd:string"/>
278   <xsd:attribute name="resource" type="xsd:string"/>
279   <xsd:attribute name="process" type="xsd:string"/>
280   <xsd:attribute name="lot" type="xsd:string"/>
281   <xsd:attribute name="task" type="xsd:string"/>
282   <xsd:attribute name="operation" type="xsd:string"/>
283 </xsd:complexType>
```

284

- 285 • *id* attribute SHOULD represent an identifier of this element.
- 286 • *key* attribute SHOULD represent a key used in the local applications.
- 287 • *name* attribute SHOULD represent the name of this element.
- 288 • *type* attribute SHOULD represent the division of this element.
- 289 • *status* attribute SHOULD represent the status of this element.
- 290 • *apply* attribute SHOULD represent application type of this element. This element is a disjunctive
291 (OR) content under the parent element, if the attribute value is "*disjunctive*".
- 292 • *party* attribute SHOULD represent an identifier of the party associated with this element.
- 293 • *plan* attribute SHOULD represent the identifier of the plan associated with this element.
- 294 • *order* attribute SHOULD represent the identifier of the order associated with this element.

- 295 • *item* attribute SHOULD represent the identifier of the item associated with this element.
- 296 • *resource* attribute SHOULD represent the identifier of the resource associated with this element.
- 297 • *process* attribute SHOULD represent the identifier of the process associated with this element.
- 298 • *lot* attribute SHOULD represent the identifier of the lot associated with this element.
- 299 • *task* attribute SHOULD represent the identifier of the task associated with this element.
- 300 • *operation* attribute SHOULD represent the identifier of the operation associated with this element.
- 301
- 302 • *Location* element SHOULD represent the location associated with this element.
- 303 • *Capacity* element SHOULD represent the capacity status of this element.
- 304 • *Progress* element SHOULD represent the progress of this element.
- 305 • *Spec* element SHOULD represent the specification of this element.
- 306 • *Start* element SHOULD represent the start event of this element.
- 307 • *End* element SHOULD represent the completion event of this element.
- 308 • *Event* element SHOULD represent the optional event under this element.
- 309 • *Price* element SHOULD represent the price of this element.
- 310 • *Cost* element SHOULD represent the cost of this element.
- 311 • *Priority* element SHOULD represent the priority of this element.
- 312 • *Display* element SHOULD represent how to display this element.
- 313 • *Description* element SHOULD represent the description of this element.
- 314 • *Author* element SHOULD represent the author of this element information.
- 315 • *Date* element SHOULD represent the date of this element information.
- 316 • *Qty* element SHOULD represent the quantity of this element.
- 317 • *Char* element SHOULD represent the qualitative value of this element.
- 318 • *Time* element SHOULD represent the time of this element.
- 319

320 **3.2 List of relational elements**

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

324

```

325 <xsd:element name="Compose" type="RelationalType"/>
326 <xsd:element name="Produce" type="RelationalType"/>
327 <xsd:element name="Consume" type="RelationalType"/>
328 <xsd:element name="Assign" type="RelationalType"/>
329 <xsd:element name="Relation" type="RelationalType"/>
  
```

330

331 **3.2.1 Compose element**

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

335

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

```
337 <Item id="A">
338   <Compose type="child" item="A1"/>
339   <Compose type="child" item="A2"/>
340 </Item>
```

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

```
342 <Item id="B">
343   <Compose type="child" item="C1"><Qty value="2"/></Compose>
344   <Compose type="child" item="C2"><Qty value="3"/></Compose>
345 </Item>
```

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

```
347 <Item id="C1">
348   <Compose type="parent" item="B1"><Qty value="2"/></Compose>
349   <Compose type="parent" item="B2"><Qty value="5"/></Compose>
350 </Item>
```

351

352 3.2.2 Produce element

353 *Produce* 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 produced by the process or operation
355 respectively, or how many items or lots are produced by the process or the operation respectively.

356 3.2.3 Consume element

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

360 3.2.4 Assign element

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

364 3.2.5 Relation element

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

368

4 Specific Elements

369

4.1 Structure of specific element

370

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.

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

```
<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>
```

398

399

- *id* attribute SHOULD represent an identifier of this element.
- *key* attribute SHOULD represent a key used in the local applications.
- *name* attribute SHOULD represent the name of this element.
- *type* attribute SHOULD represent the division of this element.
- *status* attribute SHOULD represent the status of this element.
- *apply* attribute SHOULD represent application type of this element. Specification of the element is relative, if the value is “*relative*”.

406

407

408

409

410

411

412

413

414

415

- *Start* element SHOULD represent the start event of this element.
- *End* element SHOULD represent the completion event of this element.
- *Event* element SHOULD represent the optional event under this element.
- *Price* element SHOULD represent the price of this element.
- *Cost* element SHOULD represent the cost of this element.
- *Priority* element SHOULD represent the priority of this element.
- *Display* element SHOULD represent how to display this element.
- *Description* element SHOULD represent the description of this element.
- *Author* element SHOULD represent the author of this element information.

- 416 • *Date* element SHOULD represent the date of this element information.
- 417 • *Qty* element SHOULD represent the quantity of this element.
- 418 • *Char* element SHOULD represent the qualitative value of this element.
- 419 • *Time* element SHOULD represent the time of this element.

420

421 4.2 List of specific elements

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

424

```
425 <xsd:element name="Location" type="SpecificType"/>
426 <xsd:element name="Capacity" type="SpecificType"/>
427 <xsd:element name="Progress" type="SpecificType"/>
428 <xsd:element name="Spec" type="SpecificType"/>
```

429

430 4.2.1 Location element

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

434

435 Example: Customer's address

```
436 <Party id="ABC Inc.">
437 <Location type="address"><Char value="123 ABC street"/></Location>
438 <Location type="city"><Char value="Cambridge"/></Location>
439 <Location type="state"><Char value="MA"/></Location>
440 <Location type="code"><Char value="02139"/></Location>
441 <Location type="country"><Char value="USA"/></Location>
442 </Party>
```

443

444 4.2.2 Capacity element

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

449

450 Example: Inventory level of "material01"

```
451 <Item id="material01">
452 <Capacity><Qty value="150"/></Capacity>
453 </Item>
```

454 Example: Temporal change of the material

```
455 <Item id="material01">
456 <Capacity><Qty value="150"><Time value="2005-04-10T00:00:00"/></Capacity>
457 <Capacity><Qty value="200"><Time value="2005-04-17T00:00:00"/></Capacity>
458 </Item>
```

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

```
460 <Item id="material01">
461 <Location value="storage01"/>
```

```
462 <Capacity><Qty value="150"/></Capacity>  
463 </Item>
```

464

465 **4.2.3 Progress element**

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

469 **4.2.4 Spec element**

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

475

5 Eventual Elements

476

5.1 Structure of eventual element

477

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.

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

```
<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>
```

501

502

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

503

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

504

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

505

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

506

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

507

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

508

509

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

510

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

511

512

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

513

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

514

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

515

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

516

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

517

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

518

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

519

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

520

521 **5.2 List of eventual elements**

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

525

```
526 <xsd:element name="Start" type="EventualType"/>  
527 <xsd:element name="End" type="EventualType"/>  
528 <xsd:element name="Event" type="EventualType"/>
```

529

530 **5.2.1 Start element**

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

533 **5.2.2 End element**

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

536 **5.2.3 Event element**

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

540

541

6 Accounting Elements

542

6.1 Structure of Accounting element

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

547

548

```
<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>
```

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

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

570

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

571

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

572

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

573

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

574

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

575

576

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

577

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

578

579

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

580

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

581

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

582

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

583

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

584

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

585

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

586

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

587

588 **6.2 List of accounting elements**

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

591

```
592 <xsd:element name="Price" type="AccountingType"/>  
593 <xsd:element name="Cost" type="AccountingType"/>
```

594

595 **6.2.1 Price element**

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

598 **6.2.2 Cost element**

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

601

602

7 Administrative Elements

603

7.1 Structure of Administrative Elements

604

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

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

```
<xsd:complexType name="AdministrativeType">
  <xsd:sequence>
    <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="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>
```

621

622

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

623

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

624

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

625

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

626

627

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

628

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

629

630

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

631

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

632

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

633

634

7.2 List of Administrative Elements

635

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

636

637

638

639

640

641

642

```
<xsd:element name="Priority" type="AdministrativeType"/>
<xsd:element name="Display" type="AdministrativeType"/>
<xsd:element name="Description" type="AdministrativeType"/>
<xsd:element name="Author" type="AdministrativeType"/>
<xsd:element name="Date" type="AdministrativeType"/>
```

643

644 **7.2.1 Priority element**

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

647 **7.2.2 Display element**

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

650 **7.2.3 Description element**

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

653 **7.2.4 Author element**

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

656 **7.2.5 Date element**

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

659

8 Data Elements

660

8.1 Qty element

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

665

666
667
668
669
670
671
672
673
674
675
676
677
678

```
<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>
```

679

- 680 • *name* attribute SHOULD represent the name of this element.
- 681 • *type* attribute SHOULD represent the division of this element.
- 682 • *status* attribute SHOULD represent the status of this element.
- 683 • *apply* attribute SHOULD represent application type of this element. The condition of this element is
684 exclusive, if the value is "exclusive".
- 685 • *condition* attribute SHOULD represent the condition of this element.
- 686 • *value* attribute SHOULD represent the content corresponding to the qty element.
- 687 • *count* attribute SHOULD represent the countable value of this element.
- 688 • *unit* attribute SHOULD represent the type of currency unit data of this element.
- 689 • *base* attribute SHOULD represent the base data of this element. The value of the "value" attribute is
690 divided with this value.

691

692 Example: 1/3 meters

693

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

694 Example: 3 weeks (discrete time scale)

695

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

696

8.2 Char element

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

701

```

702 <xsd:element name="Char">
703   <xsd:complexType>
704     <xsd:attribute name="name" type="xsd:string"/>
705     <xsd:attribute name="type" type="xsd:string"/>
706     <xsd:attribute name="status" type="xsd:string"/>
707     <xsd:attribute name="apply" type="xsd:string"/>
708     <xsd:attribute name="condition" type="xsd:string"/>
709     <xsd:attribute name="value" type="xsd:string"/>
710     <xsd:attribute name="count" type="xsd:long"/>
711     <xsd:attribute name="unit" type="xsd:string"/>
712     <xsd:attribute name="base" type="xsd:string"/>
713   </xsd:complexType>
714 </xsd:element>

```

- 715
- 716 • *name* attribute SHOULD represent the name of this element.
 - 717 • *type* attribute SHOULD represent the division of this element.
 - 718 • *status* attribute SHOULD represent the status of this element.
 - 719 • *apply* attribute SHOULD represent application type of this element. The condition of this element is
720 exclusive, if the value is “exclusive”.
 - 721 • *condition* attribute SHOULD represent the condition of this element.
 - 722 • *value* attribute SHOULD represent the content corresponding to the qty element.
 - 723 • *count* attribute SHOULD represent the countable value of this element.
 - 724 • *unit* attribute SHOULD represent the type of currency unit data of this element.
 - 725 • *base* attribute SHOULD represent the base data of this element. The value of the “value” attribute is
726 divided with this value.
- 727

728 8.3 Time element

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

```

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

```

- 746
- 747 • *name* attribute SHOULD represent the name of this element.
 - 748 • *type* attribute SHOULD represent the division of this element.
 - 749 • *status* attribute SHOULD represent the status of this element.
 - 750 • *apply* attribute SHOULD represent application type of this element. The condition of this element is
751 exclusive, if the value is “exclusive”.
 - 752 • *condition* attribute SHOULD represent the condition of this element.

- 753 • *value* attribute SHOULD represent the content corresponding to the qty element.
- 754 • *count* attribute SHOULD represent the countable value of this element.
- 755 • *unit* attribute SHOULD represent the type of currency unit data of this element.
- 756 • *base* attribute SHOULD represent the base data of this element. The value of the “value” attribute is
757 divided with this value.

758

759 Example: noon on May 13th, 2005

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

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

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

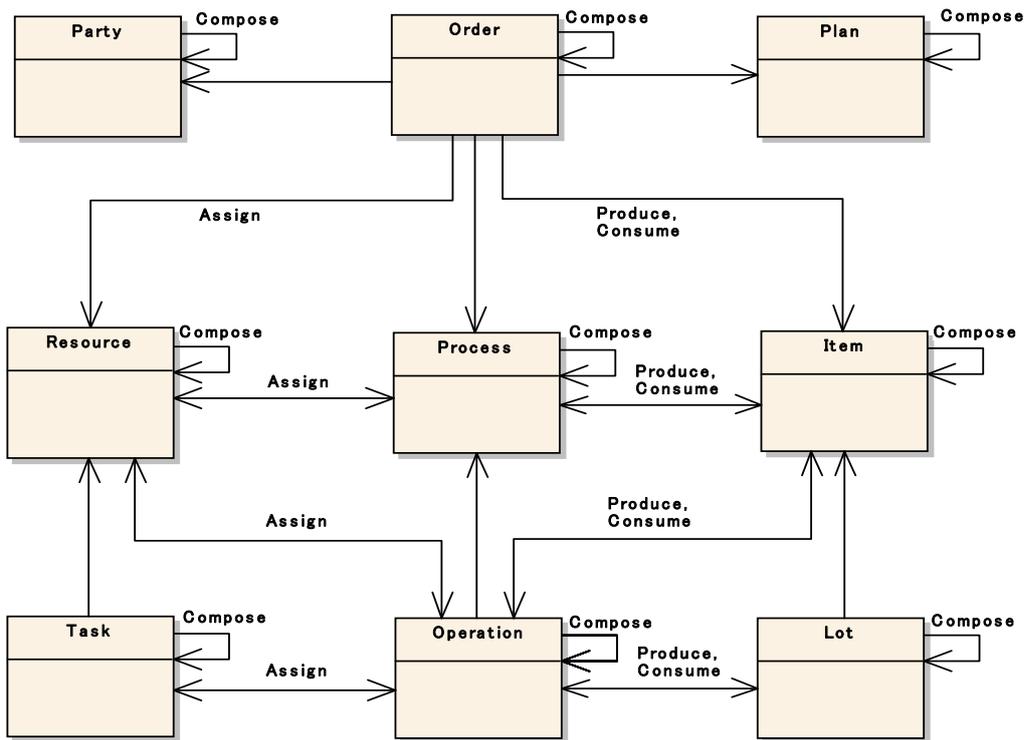
763

764

A. Object Class diagram

765
766
767
768
769
770

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



771
772
773

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

774

B. Cross reference of elements

775
776
777

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

778

779 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																						

780

781

782

783

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

790
 791 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									

792

793 **C. Acknowledgements**

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

796 **Participants:**

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

809 **D. Revision History**

810

Revision	Date	Editor	Changes Made
01		Y.Nishioka	

811

812