

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, <i>Key words for use in RFCs to Indicate Requirement Levels</i> , 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>

1.5 Terms and definitions

Plan

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

Order

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

Party

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

Item

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

Resource

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

Process

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

Lot

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

Task

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

Operation

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

2 Primitive Elements

2.1 Structure of primitive elements

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.

```
<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>
```

- *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.

- *item* attribute SHOULD represent the identifier of the item associated with this element.
- *resource* attribute SHOULD represent the identifier of the resource associated with this element.
- *process* attribute SHOULD represent the identifier of the process associated with this element.
- *lot* attribute SHOULD represent the identifier of the lot associated with this element.
- *task* attribute SHOULD represent the identifier of the task associated with this element.
- *operation* attribute SHOULD represent the identifier of the operation associated with this element.
- *Compose* element SHOULD represent the element corresponding to part of this element.
- *Produce* element SHOULD represent the relation that this element produces.
- *Consume* element SHOULD represent the relation that this element consumes.
- *Assign* element SHOULD represent the relation that this element uses.
- *Relation* element SHOULD represent the relation to other primitive elements.
- *Location* element SHOULD represent the location where this element exists.
- *Capacity* element SHOULD represent the capacity status of this element.
- *Progress* element SHOULD represent the progress of this element.
- *Spec* element SHOULD represent the specification of this element.
- *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.
- *Date* element SHOULD represent the date of this element information.

2.2 List of primitive elements

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

```
<xsd:element name="Party" type="PrimitiveType"/>
<xsd:element name="Plan" type="PrimitiveType"/>
<xsd:element name="Order" type="PrimitiveType"/>
<xsd:element name="Item" type="PrimitiveType"/>
<xsd:element name="Resource" type="PrimitiveType"/>
<xsd:element name="Process" type="PrimitiveType"/>
<xsd:element name="Lot" type="PrimitiveType"/>
<xsd:element name="Task" type="PrimitiveType"/>
<xsd:element name="Operation" type="PrimitiveType"/>
```

2.2.1 Party element

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

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

2.2.2 Plan element

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

2.2.3 Order element

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

Example: Ten "A" products are requested.

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

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

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

Example: Switching operation is requested two times.

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

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

```
<Order id="Z00">
  <Compose order="Z01"/>
  <Compose order="Z02"/>
  <Price value="3000" unit="yen"/>
</Order>
<Order id="Z01" item="A">
  <Spec type="quantity"><Qty value="10"/></Spec>
</Order>
<Order id="Z02" item="B">
  <Spec type="quantity"><Qty value="5"/></Spec>
</Order>
```

2.2.4 Item element

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

2.2.5 Resource element

Resource element represents a resource, which is an object enabling production, transportation, storage, inspection and other various services. As resource can produce tasks to execute operations, it is 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>`

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.

3 Relational Elements

3.1 Structure of relational elements

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.

```
<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>
```

- *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. This element is a disjunctive (OR) content under the parent element, if the attribute value is "*disjunctive*".
- *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.

- *item* attribute SHOULD represent the identifier of the item associated with this element.
- *resource* attribute SHOULD represent the identifier of the resource associated with this element.
- *process* attribute SHOULD represent the identifier of the process associated with this element.
- *lot* attribute SHOULD represent the identifier of the lot associated with this element.
- *task* attribute SHOULD represent the identifier of the task associated with this element.
- *operation* attribute SHOULD represent the identifier of the operation associated with this element.
- *Location* element SHOULD represent the location associated with this element.
- *Capacity* element SHOULD represent the capacity status of this element.
- *Progress* element SHOULD represent the progress of this element.
- *Spec* element SHOULD represent the specification of this element.
- *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.
- *Date* element SHOULD represent the date of this element information.
- *Qty* element SHOULD represent the quantity of this element.
- *Char* element SHOULD represent the qualitative value of this element.
- *Time* element SHOULD represent the time of this element.

3.2 List of relational elements

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

```
<xsd:element name="Compose" type="RelationalType"/>
<xsd:element name="Produce" type="RelationalType"/>
<xsd:element name="Consume" type="RelationalType"/>
<xsd:element name="Assign" type="RelationalType"/>
<xsd:element name="Relation" type="RelationalType"/>
```

3.2.1 Compose element

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

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

```

<Item id="A">
  <Compose type="child" item="A1"/>
  <Compose type="child" item="A2"/>
</Item>

```

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

```

<Item id="B">
  <Compose type="child" item="C1"><Qty value="2"/></Compose>
  <Compose type="child" item="C2"><Qty value="3"/></Compose>
</Item>

```

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

```

<Item id="C1">
  <Compose type="parent" item="B1"><Qty value="2"/></Compose>
  <Compose type="parent" item="B2"><Qty value="5"/></Compose>
</Item>

```

3.2.2 Produce element

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

3.2.3 Consume element

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

3.2.4 Assign element

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

3.2.5 Relation element

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

4 Specific Elements

4.1 Structure of specific element

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.

```
<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>
```

- *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*”.
- *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.

- *Date* element SHOULD represent the date of this element information.
- *Qty* element SHOULD represent the quantity of this element.
- *Char* element SHOULD represent the qualitative value of this element.
- *Time* element SHOULD represent the time of this element.

4.2 List of specific elements

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

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

4.2.1 Location element

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

Example: Customer's address

```
<Party id="ABC Inc.">
  <Location type="address"><Char value="123 ABC street"/></Location>
  <Location type="city"><Char value="Cambridge"/></Location>
  <Location type="state"><Char value="MA"/></Location>
  <Location type="code"><Char value="02139"/></Location>
  <Location type="country"><Char value="USA"/></Location>
</Party>
```

4.2.2 Capacity element

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

Example: Inventory level of "material01"

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

Example: Temporal change of the material

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

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

```
<Item id="material01">
  <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.

5 Eventual Elements

5.1 Structure of eventual element

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.

```
<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>
```

- *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. The condition of this element is exclusive, if the value is "exclusive".
- *condition* attribute SHOULD represent the condition of this element.
- *value* attribute SHOULD represent the qualitative value 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.
- *Date* element SHOULD represent the date of this element information.
- *Qty* element SHOULD represent the quantity of this element.
- *Char* element SHOULD represent the qualitative value of this element.
- *Time* element SHOULD represent the time of this element.

5.2 List of eventual elements

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

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

5.2.1 Start element

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

5.2.2 End element

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

5.2.3 Event element

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

6 Accounting Elements

6.1 Structure of Accounting element

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

```
<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>
```

- *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. The condition of this element is exclusive, if the value is "exclusive".
- *condition* attribute SHOULD represent the condition of this element.
- *value* attribute SHOULD represent the qualitative value 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.
- *Date* element SHOULD represent the date of this element information.
- *Qty* element SHOULD represent the quantitative value of this element.
- *Char* element SHOULD represent the qualitative value of this element.
- *Time* element SHOULD represent the temporal value of this element.

6.2 List of accounting elements

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

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

6.2.1 Price element

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

6.2.2 Cost element

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

7 Administrative Elements

7.1 Structure of Administrative Elements

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.

```
<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>
```

- *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. The condition of this element is exclusive, if the value is "exclusive".
- *condition* attribute SHOULD represent the condition of this element.
- *value* attribute SHOULD represent the qualitative value of this element.
- *Qty* element SHOULD represent the quantitative value of this element.
- *Char* element SHOULD represent the qualitative value of this element.
- *Time* element SHOULD represent the temporal value of this element.

7.2 List of Administrative Elements

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.

```
<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"/>
```

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.

8 Data Elements

8.1 Qty element

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.

```
<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>
```

- *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. The condition of this element is exclusive, if the value is “exclusive”.
- *condition* attribute SHOULD represent the condition of this element.
- *value* attribute SHOULD represent the content corresponding to the qty element.
- *count* attribute SHOULD represent the countable value of this element.
- *unit* attribute SHOULD represent the type of currency unit data of this element.
- *base* attribute SHOULD represent the base data of this element. The value of the “value” attribute is divided with this value.

Example: 1/3 meters

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

Example: 3 weeks (discrete time scale)

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

8.2 Char element

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.


```

<xsd:element name="Char">
  <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:string"/>
    <xsd:attribute name="count" type="xsd:long"/>
    <xsd:attribute name="unit" type="xsd:string"/>
    <xsd:attribute name="base" type="xsd:string"/>
  </xsd:complexType>
</xsd:element>

```

- *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. The condition of this element is exclusive, if the value is "exclusive".
- *condition* attribute SHOULD represent the condition of this element.
- *value* attribute SHOULD represent the content corresponding to the qty element.
- *count* attribute SHOULD represent the countable value of this element.
- *unit* attribute SHOULD represent the type of currency unit data of this element.
- *base* attribute SHOULD represent the base data of this element. The value of the "value" attribute is divided with this value.

8.3 Time element

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

```

<xsd:element name="Time">
  <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:dateTime"/>
    <xsd:attribute name="count" type="xsd:long"/>
    <xsd:attribute name="unit" type="xsd:string"/>
    <xsd:attribute name="base" type="xsd:dateTime"/>
  </xsd:complexType>
</xsd:element>

```

- *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. The condition of this element is exclusive, if the value is "exclusive".
- *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

A. Object Class diagram

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.

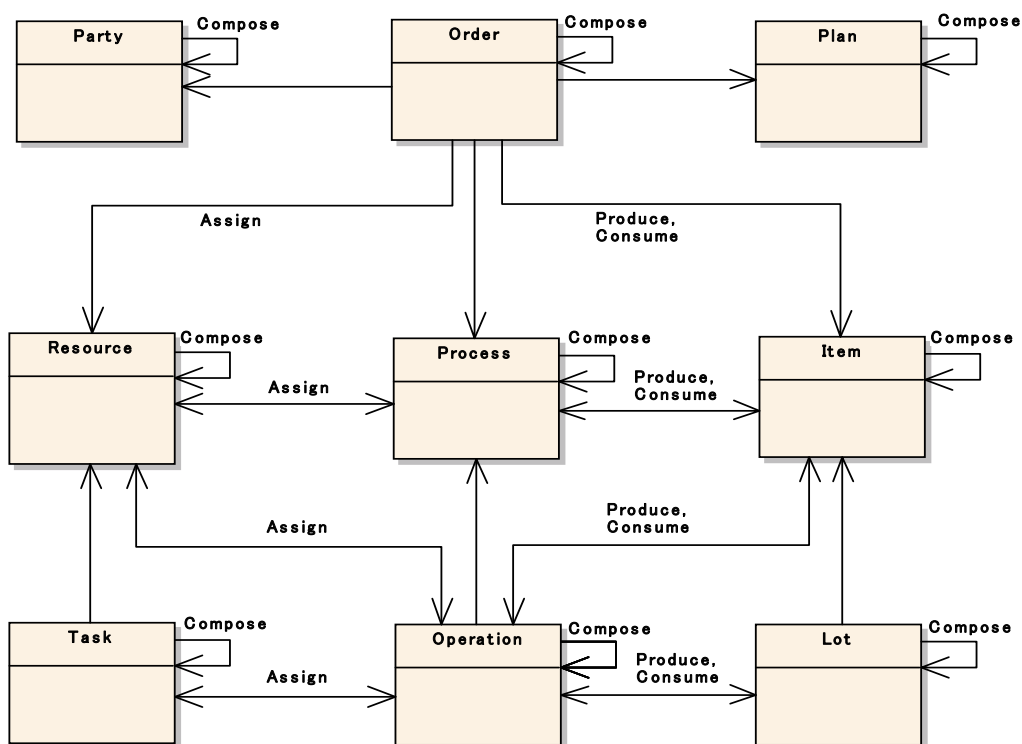


Figure A-1: Primitive objects for representing planning and scheduling problems

B. Cross reference of elements

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.

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																						

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

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	S	D		N	S	D							
Char			S		S	S	S	S	S			N	S	S							
Time			S		S	S	S	S	S	T		N	S	T							

C. Acknowledgements

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

Participants:

Shinya Matsukawa, Hitachi
Tomohiko Maeda, Fujitsu
Masahiro Mizutani, Unisys Corporation
Akihiro Kawauchi, Individual Member
Yuto Banba, PSLX Forum
Osamu Sugi, PSLX Forum
Hideichi Okamune, PSLX Forum
Hiroshi Kojima, PSLX Forum
Ken Nakayama, Hitachi
Yukio Hamaguchi, Hitachi
Tomoichi Sato, Individual
Hiroaki Sasaki, Individual

D. Revision History

Revision	Date	Editor	Changes Made
01		Y.Nishioka	