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

Public Review Draft 01

7th August 2007

Specification URIs:

http://docs.oasis-open.org/pps/v1.0/pr01/pps-core-elements-1.0-pr01.doc
http://docs.oasis-open.org/pps/v1.0/pr01/pps-core-elements-1.0-pr01.html
http://docs.oasis-open.org/pps/v1.0/pr01/pps-core-elements-1.0-pr01.pdf

Previous Version:

N/A

Latest Version:

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

Latest Approved Version:

N/A

Technical Committee:

OASIS Production Planning and Scheduling TC

Chair(s):

Yasuyuki Nishioka, PSLX Forum / Hosei University

Editor(s):

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

Related work:

This specification is related to:

- Universal Business Language 2.0

Declared XML Namespace(s):

http://docs.oasis-open.org/pps/ns/core-elements

Abstract:

OASIS PPS (Production Planning and Scheduling) Standard deals with problems in all manufacturing companies who want to have a sophisticated information system for production planning and scheduling. PPS standard provides XML schema and communication protocols for information exchange among manufacturing application programs in the web-services environment. This document especially focuses on information model of core elements in the production planning and scheduling domain. Since the elements have been designed without specific contexts in planning and scheduling, they can be used in any specific type of messages as a building block depending on the context of application programs.

Status:

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

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

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

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

Notices

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", [insert specific trademarked names and abbreviations here] 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

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

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

This specification aims at production planning and scheduling for all kinds of products and services provided by manufacturing enterprises. Production scheduling explained in this specification can be divided into scheduling in the whole enterprise including some areas or sites and detail scheduling in the individual areas and sites.

This specification doesn't aim at optimization logic for solution, special knowledge of individual enterprises, concrete solution methods for production planning and scheduling, and planning for the total supply chain.

1.1 Terminology

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

1.2 Normative References

[RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, http://www.ietf.org/rfc/rfc2119.txt, IETF RFC 2119, March 1997.

[PPS02] PPS (Production Planning and Scheduling) Part 2: Transaction Messages, Version 1.0, Public Review Draft 01, http://www.oasis-open.org/committees/pps/

[PPS03] PPS (Production Planning and Scheduling) Part 3: Profile Specifications, Version 1.0, Public Review Draft 01, http://www.oasis-open.org/committees/pps/

1.3 Non-Normative References

[PSLXWP] PSLX Consortium, PSLX White Paper - APS Conceptual definition and implementation, http://www.pslx.org/

[PSLX001] PSLX Technical Standard, Version 2, Part 1: Enterprise Model (in Japanese), Recommendation of PSLX Forum, http://www.pslx.org/

[PSLX002] PSLX Technical Standard, Version 2, Part 2: Activity Model (in Japanese), Recommendation of PSLX Forum, http://www.pslx.org/

[PSLX003] PSLX Technical Standard, Version 2, Part 3: Object Model (in Japanese), Recommendation of PSLX Forum, http://www.pslx.org/

1.4 Conformance

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

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

1.5 Terms and definitions

plan

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

order

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

party

customer who is a sender of an order and has a demand to make a decision, or supplier thow 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 activity. The quantity or the quality of item is changed by means of the production activity.

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

resource

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

Example: equipment, machine, device, labor, tool

process

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

lot

instance of specific 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 quantity is equal to the quantity of item produced or consumed by the operation.

task

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

Notes: Task represents either the capacity value provided by resource at a specific time point or the 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 function in production activity.

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:element ref="Qty" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Char" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Duration" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="Time" 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.
- *Qty* element SHOULD represent the quantity of this element.
- *Char* element SHOULD represent the qualitative value of this element.
- *Duration* element SHOULD represent the net duration of this element.
- *Time* element SHOULD represent the time of this element.

2.2 List of primitive elements

This standard defines nine primitive elements: *Party*, *Plan*, *Order*, *Item*, *Resource*, *Process*, *Lot*, *Task*, and *Operation*. The type of this element 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 customer and supplier. Customer is an object that requests some products or services from the enterprise. Such requests are sent to a person in charge of production planning or scheduling. Supplier is an object providing some products or services to the enterprise. Supplier receives orders form the enterprise and provides additional item, resource or function to the enterprise.

2.2.2 Order element

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

Example: Ten of “A” products are requested.

```
<Order id=”Z01” item=”A”>
  <Qty value=”10”/>
</Order>
```

Example: Three labors in group “B” are requested.

```
<Order id=”Z02” resource=”groupB”>
  <Qty value=”3”/>
</Order>
```

Example: Switching operation is requested two times.

```
<Order id=”Z03” process=”change01”>
  <Qty value=”2”/>
</Order>
```

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

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

2.2.3 Item element

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

2.2.4 Resource element

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

2.2.5 Function element

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

2.2.6 Lot element

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

2.2.7 Task element

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

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

```
<Task id="T01">
  <Qty value="3" unit="human"/>
  <Duration value="P0Y0M2DT0H0M0S"/>
</Task>
```

2.2.8 Operation element

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

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="Duration" 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="value" type="xsd:double"/>
  <xsd:attribute name="disjunctive" type="xsd:boolean"/>
  <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.
- *value* attribute SHOULD represent the content corresponding to the qty element.
- *disjunctive* attribute SHOULD represent one of the following values. The default value of this attribute SHOULD be set “false".
  - If the attribute value is "true", it SHOULD be represented that this element is a disjunctive (OR) content under the parent element.

297 – If the attribute value is "false", it SHOULD be represented that this element is a conjunctive (AND)
298 content under the parent element.

299 • *party* attribute SHOULD represent an identifier of the party associated with this element.
300 • *plan* attribute SHOULD represent the identifier of the plan associated with this element.
301 • *order* attribute SHOULD represent the identifier of the order associated with this element.
302 • *item* attribute SHOULD represent the identifier of the item associated with this element.
303 • *resource* attribute SHOULD represent the identifier of the resource associated with this element.
304 • *process* attribute SHOULD represent the identifier of the process associated with this element.
305 • *lot* attribute SHOULD represent the identifier of the lot associated with this element.
306 • *task* attribute SHOULD represent the identifier of the task associated with this element.
307 • *operation* attribute SHOULD represent the identifier of the operation associated with this element.
308
309 • *Location* element SHOULD represent the location associated with this element.
310 • *Capacity* element SHOULD represent the capacity status of this element.
311 • *Progress* element SHOULD represent the progress of this element.
312 • *Spec* element SHOULD represent the specification of this element.
313 • *Start* element SHOULD represent the start event of this element.
314 • *End* element SHOULD represent the completion event of this element.
315 • *Event* element SHOULD represent the optional event under this element.
316 • *Price* element SHOULD represent the price of this element.
317 • *Cost* element SHOULD represent the cost of this element.
318 • *Priority* element SHOULD represent the priority of this element.
319 • *Display* element SHOULD represent how to display this element.
320 • *Description* element SHOULD represent the description of this element.
321 • *Author* element SHOULD represent the author of this element information.
322 • *Date* element SHOULD represent the date of this element information.
323 • *Qty* element SHOULD represent the quantity of this element.
324 • *Char* element SHOULD represent the qualitative value of this element.
325 • *Duration* element SHOULD represent the net duration of this element.
326 • *Time* element SHOULD represent the time of this element.
327

328 3.2 List of relational elements

329 This standard defines five relational elements: *Compose*, *Produce*, *Consume*, *Assign*, and *Relation*. The
330 type of this element SHOULD be represented with the following XML schema.
331

```
332 <xsd:element name="Compose" type="RelationalType"/>
333 <xsd:element name="Produce" type="RelationalType"/>
334 <xsd:element name="Consume" type="RelationalType"/>
335 <xsd:element name="Assign" type="RelationalType"/>
336 <xsd:element name="Relation" type="RelationalType"/>
```

337

3.2.1 Compose element

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

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

```
<Item id=”A”>
  <Compose type=”pps:child” item=”A1”/>
  <Compose type=”pps:child” item=”A2”/>
</Item>
```

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

```
<Item id=”B”>
  <Compose type=”pps:child” item=”C1” value=”2”/>
  <Compose type=”pps:child” item=”C2” value=”3”/>
</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=”pps:parent” item=”B1”><Qty value=”2”/></Compose>
  <Compose type=”pps:parent” item=”B2”><Qty value=”5”/></Compose>
</Item>
```

3.2.2 Produce element

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

3.2.3 Consume element

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

3.2.4 Assign element

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

3.2.5 Relation element

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

4 Specific Elements

4.1 Structure of specific element

Specific elements are defined to represent any properties. These elements MAY have multiple instances with its time stamp. The type of this element SHOULD be represented with the following XML schema and SHOULD fulfill the following constraints.

```
<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="Duration" 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="relative" type="xsd:boolean"/>
</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.
- *value* attribute SHOULD represent the content corresponding to the qty element.
- *relative* attribute SHOULD represent whether specifying spec is relative. The default value of this attribute SHOULD be “false”.

- *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.
- *Duration* element SHOULD represent the net duration of this element.
- *Time* element SHOULD represent the time of this element.

4.2 List of specific elements

For specific elements, this standard 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 specifying different names of the data. Change of the location can be represented time-dependently.

Example: Customer's address

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

4.2.2 Capacity element

Capacity element represents volume of capability of Resource, Item and Process. For Resource, it shows available summary of corresponding tasks. For Item, it shows the available summary of Lots. And for Process, it shows available rate 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">
  <Capacity><Location value="storage01"/><Qty value="150"/></Capacity>
</Item>
```

Example: Person X is available at A for 10 hours, and B for 8 hours.

```
<Resource id="Person X">
  <Capacity>
    <Location value="A"/>
    <Duration value="P0Y0M0DT10H0M0S"/>
  </Capacity>
  <Capacity>
    <Location value="B"/>
    <Duration value="P0Y0M0DT8H0M0S"/>
  </Capacity>
</Resource>
```

4.2.3 Progress element

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

4.2.4 Spec element

Spec element represents various specifications for primitive elements. The content can be represented with a pair of a spec name and a value. This element can also represent time-dependent change of the value. The value of the specification is represented with one data type of a numerical value, characters and date time. Spec elements with the same specification name under a common parent element 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="Duration" 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="exclusive" type="xsd:boolean"/>
</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.
- *value* attribute SHOULD represent the content corresponding to the qty element.
- *condition* attribute SHOULD represent the condition of this element.
- *exclusive* attribute SHOULD represent whether exclusive or not for the condition of this element. Default value is “false”.

- *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.
- *Duration* element SHOULD represent the net duration of this element.
- *Time* element SHOULD represent the time of this element.

5.2 List of eventual elements

This standard defines three eventual elements: *Start*, *End*, and *Event*. The *Start* and *End* is 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 order or operation. 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 order or operation. 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 attending with a customer, supplier, item, resource, function 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 income and spending. Price and cost of goods and services are the target of the elements. The type of this element SHOULD be represented with the following XML schema and SHOULD fulfill the following constraints.

```
<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="Duration" 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:decimal"/>
  <xsd:attribute name="condition" type="xsd:string"/>
  <xsd:attribute name="exclusive" type="xsd:boolean"/>
</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.
- *value* attribute SHOULD represent the content corresponding to the qty element.
- *condition* attribute SHOULD represent the condition of this element.
- *exclusive* attribute SHOULD represent whether exclusive or not for the condition of this element. Default value is “false”.

- *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.
- *Duration* element SHOULD represent the net duration of this element.
- *Time* element SHOULD represent the time of this element.

6.2 List of accounting elements

For accounting elements, *Price* element and *Cost* element are defined in this standard. 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. The currency unit can be set if necessary.

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. The currency unit can be set if necessary.

7 Administrative Elements

7.1 Structure of Administrative Elements

Administrative elements represent any administrative information, which is not the main body of domain objects but the information how to deal with the domain objects. 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="Duration" 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="value" type="xsd:string"/>
   <xsd:attribute name="condition" type="xsd:string"/>
   <xsd:attribute name="exclusive" type="xsd:boolean"/>
</xsd:complexType>
```

- *name* attribute SHOULD represent the name of this element.
- *type* attribute SHOULD represent the division of this element.
- *value* attribute SHOULD represent the content corresponding to the qty element.
- *condition* attribute SHOULD represent the condition of this element.
- *exclusive* attribute SHOULD represent whether exclusive or not for the condition of this element. Default value is “false”.

- *Qty* element SHOULD represent the quantity of this element.
- *Char* element SHOULD represent the qualitative value of this element.
- *Duration* element SHOULD represent the net duration of this element.
- *Time* element SHOULD represent the time of this element.

7.2 List of Administrative Elements

For administrative elements, *Priority*, *Display*, *Description*, *Author* and *Date* elements are defined in this standard. 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"/>
```

7.2.1 Priority element

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

7.2.2 Display element

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

7.2.3 Description element

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

7.2.4 Author element

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

7.2.5 Date element

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

8 Data Elements

8.1 Qty element

Qty element SHOULD represent quantity information. This element can be used to represent the quantitative numerical data. 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="value" type="xsd:decimal"/>
    <xsd:attribute name="condition" type="xsd:string"/>
    <xsd:attribute name="exclusive" type="xsd:boolean"/>
    <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.
- *value* attribute SHOULD represent the content corresponding to the qty element.
- *condition* attribute SHOULD represent the condition of this element.
- *exclusive* attribute SHOULD represent whether exclusive or not for the condition of this element. Default value is “false”.
- *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”/>
```

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="value" type="xsd:string"/>
    <xsd:attribute name="condition" type="xsd:string"/>
    <xsd:attribute name="exclusive" type="xsd:boolean"/>
    <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.
- *value* attribute SHOULD represent the content corresponding to the qty element.
- *condition* attribute SHOULD represent the condition of this element.
- *exclusive* attribute SHOULD represent whether exclusive or not for the condition of this element. Default value is “false”.
- *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 Duration element

Duration element SHOULD represent duration. Duration presented by this element is indicated with either continuous time scale such as day, hour, minute and second, or discrete time under a specific scale. The type of this element SHOULD be represented with the following XML schema and SHOULD fulfill the following constraints.

```
<xsd:element name="Duration">
  <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="value" type="xsd:duration"/>
    <xsd:attribute name="condition" type="xsd:string"/>
    <xsd:attribute name="exclusive" type="xsd:boolean"/>
    <xsd:attribute name="count" type="xsd:long"/>
    <xsd:attribute name="unit" type="xsd:string"/>
    <xsd:attribute name="base" type="xsd:duration"/>
  </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.
- *value* attribute SHOULD represent the content corresponding to the qty element.
- *condition* attribute SHOULD represent the condition of this element.
- *exclusive* attribute SHOULD represent whether exclusive or not for the condition of this element. Default value is “false”.
- *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: 3 hours and 10 minutes

```
<Duration value=”P0Y0M0DT3H10M0S”/>
```

Example: 3 weeks (discrete time scale)

```
<pps:duration count=”3” unit=”week” base=”P0Y0M7DT0H0M0S”/>
```

8.4 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="value" type="xsd:dateTime"/>
    <xsd:attribute name="condition" type="xsd:string"/>
    <xsd:attribute name="exclusive" type="xsd:boolean"/>
    <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.
- *value* attribute SHOULD represent the content corresponding to the qty element.
- *condition* attribute SHOULD represent the condition of this element.
- *exclusive* attribute SHOULD represent whether exclusive or not for the condition of this element. Default value is “false”.
- *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: noon on May 13th, 2005

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

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

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

A. Object Class diagram

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

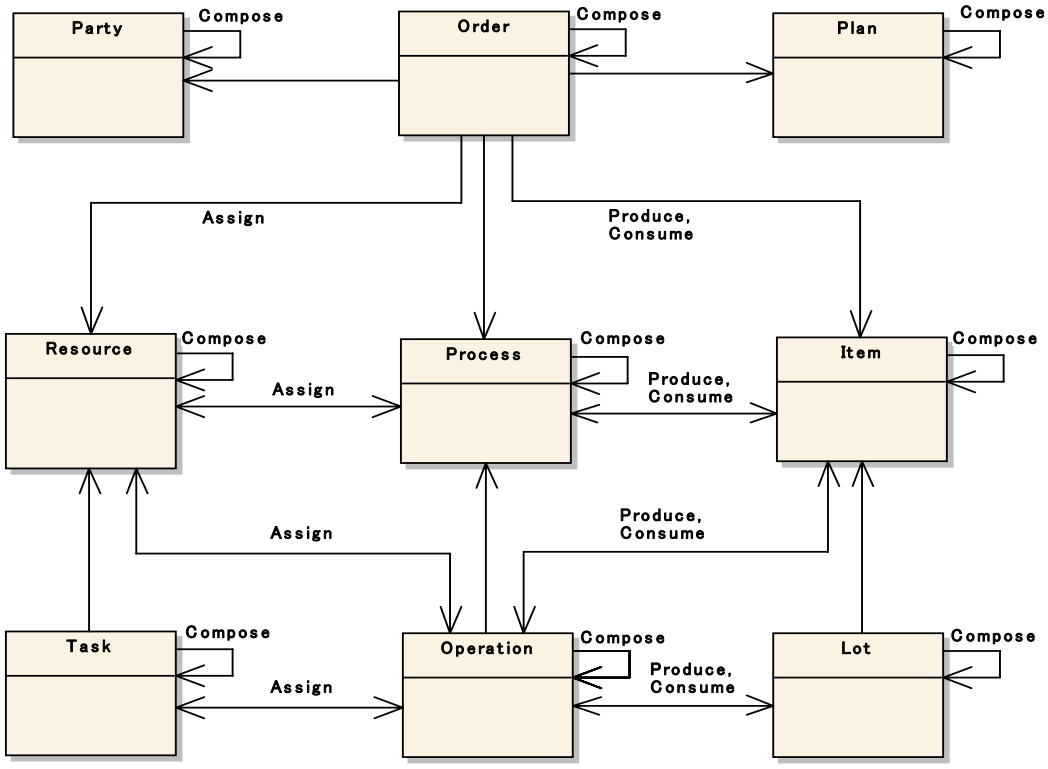
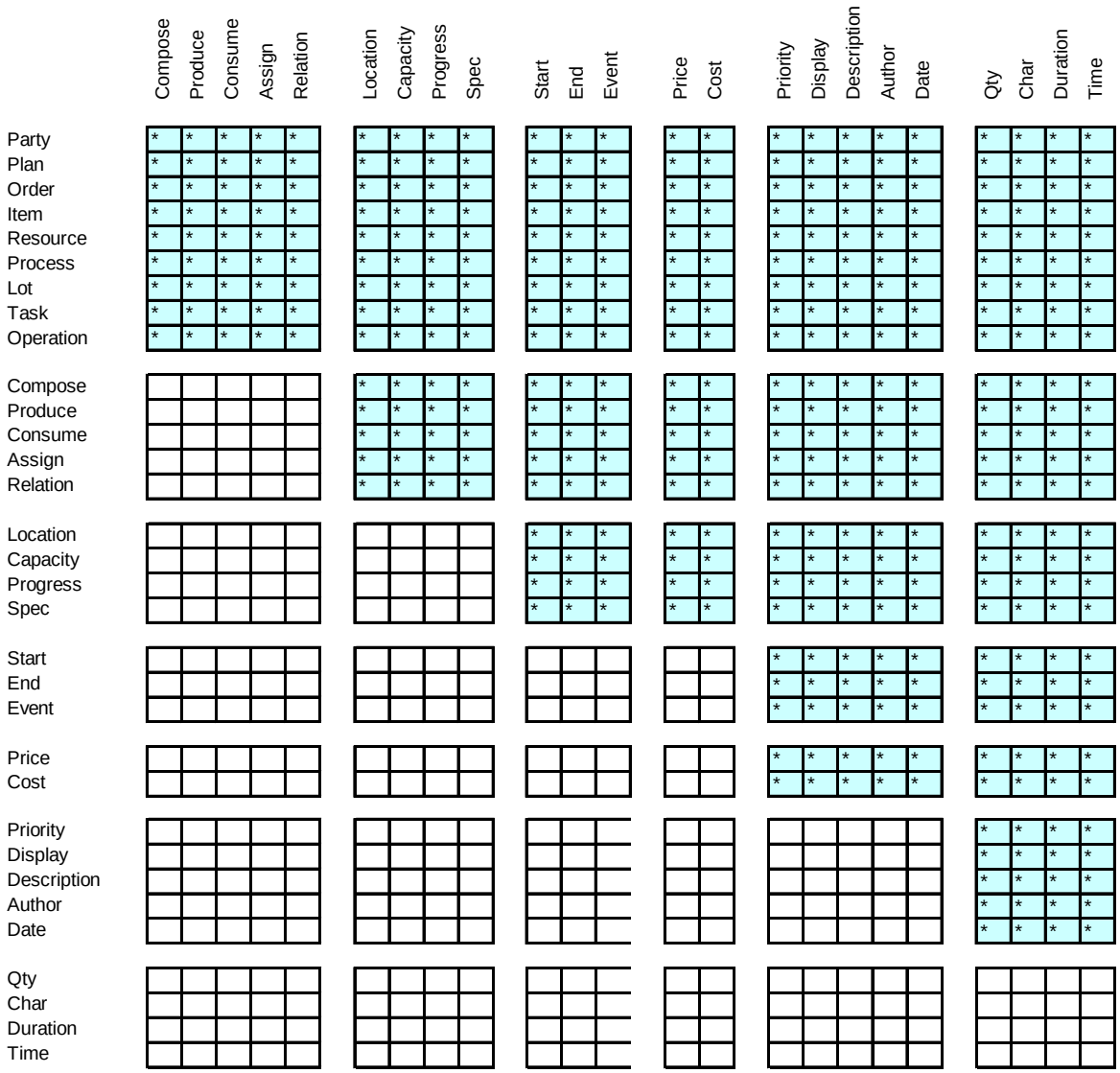


Figure A-1: Primitive objects for representing planning and scheduling problems

B. Cross reference of elements

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

Table B-1 Element and sub-element relations



The following table B-2 shows the correspondence between elements and attributes. The horizontal lines show element names and the vertical lines show attribute names. The characters in the table represent data types. The letters in the table are used as follows: “U” for identification character of element, “P” for an identification character of other elements, “S” for the character string, “F” for a decimal number, “N” for a integer number, “T” for date time, “D” for duration and “B” means that the value is “true” or “false”. Boldface means required information.

	id	key	name	parent	type	status	value	condition	relative	disjunctive	exclusive	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			B					P	P	P	P	P	P	P	P	P
Produce	U	N	S		S		S			B					P	P	P	P	P	P	P	P	P
Consume	U	N	S		S		S			B					P	P	P	P	P	P	P	P	P
Assign	U	N	S		S		S			B					P	P	P	P	P	P	P	P	P
Relation	U	N	S		S		S			B					P	P	P	P	P	P	P	P	P
Location	U	N	S		S	S	S		B														
Capacity	U	N	S		S	S	S		B														
Progress	U	N	S		S	S	S		B														
Spec	U	N	S		S	S	S		B														
Start	U	N	S		S	S	S	S			B												
End	U	N	S		S	S	S	S			B												
Event	U	N	S		S	S	S	S			B												
Price	U	N	S		S	S	S	S			B												
Cost	U	N	S		S	S	S	S			B												
Priority			S		S		S	S			B												
Display			S		S		S	S			B												
Description			S		S		S	S			B												
Author			S		S		S	S			B												
Date			S		S		S	S			B												
Qty			S		S	S	F	S			B	N	S	F									
Char			S		S	S	S	S			B	N	S	S									
Duration			S		S	S	D	S			B	N	S	D									
Time			S		S	S	T	S			B	N	S	T									

Table B-2 Element and attribute relations

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