ARCHIVES of FOUNDRY ENGINEERING

Published quarterly as the organ of the Foundry Commission of the Polish Academy of Sciences

ISSN (1897-3310)
Volume 7
Issue 3/2007

147 – 152

28/3

XML Model of Planning System in Foundry

A. Stawowy*, J. Duda
Faculty of Management, AGH University of Science and Technology , Gramatyka 10, 30-067 Krakow, Poland
*e-mail: astawowy@zarz.agh.edu.pl

Received on 20.04.2007; Approved for printing on: 27.04.2007

Abstract

Contemporary enterprises (including those manufacturing iron castings) have at their disposal advanced computer resources for the management of production processes. The disadvantage of these solutions is an incoherent language for writing production planning and scheduling problems. The lack of the standard for data exchange and model description makes the work on designing, development and implementation difficult. Various dialects of XML language, among others for production planning purposes, which may change this situation have been developed for the last several years.
The paper describes a PSLX language, which can be viewed first of all as an ontology and communication protocol inside and outside of the Advanced Planning and Scheduling (APS) system, as well as an XML standard for production planning and scheduling. This language has been then used to develop a model of planning system in a foundry. The model consists of customer's order model, resources model and scheduling model.

Keywords: Application of information technology to the foundry industry, XML, Modelling

1. Introduction

A short-term planning problem in steelworks and foundries is especially complex, because production processes in such manufacturers are of a continuous-discrete type. Complex and comprehensive character of the modelled objects requires development of universal tools which would be useful for depicting wide range of manufacturing planning and control (MPC) systems. Another significant aim is adding to the research connected with standardisation of models and methods in industrial and business oriented applications. The lack of standardisation has adverse impact not only on solving individual problems and designing tools for real-world systems, but also significantly limits exchange of knowledge between research teams. Another important problem is the integration with the computer systems of MRPII/ERP type, and more widely, a language in which planning and scheduling problems are defined. Relational database tools are commonly used for this purpose but this solution is not enough for different reasons. In this case we propose to use the PSLX – Planning and Scheduling Language on XML specification, developed by Japanese PSLX Consortium [1].

This language provides the following:

- a set of XML model and terminology for common manufacturing and planning problems,
- standardized, inexpensive data input and output tools,
- low cost of integration, both among and within enterprises, through the reuse of common data structures.

This paper continues in Section 2 with an explanation of the PSLX schema. Section 3 gives the details on model of the system under consideration. The conclusions are drawn in Section 4.

2. XML specification of MPC elements

Simplified diagram of a manufacturing planning and control system [2] is shown in Figure 1. The system encompasses three activities (phases):

1. establishment of the overall direction for the firm,
2. detailed planning of material flows and capacity,
3. detailed shop scheduling and control.

MPC system is core element of various ERP class systems. The key component of modern MPC systems is advanced planning and scheduling system (APS) which is employed at *engine* and *back end* phases. APS is a technology that utilizes optimisation algorithms for solving models which simultaneously take into account all the materials and enterprise resources available, while reflecting all the constraints and business goals.

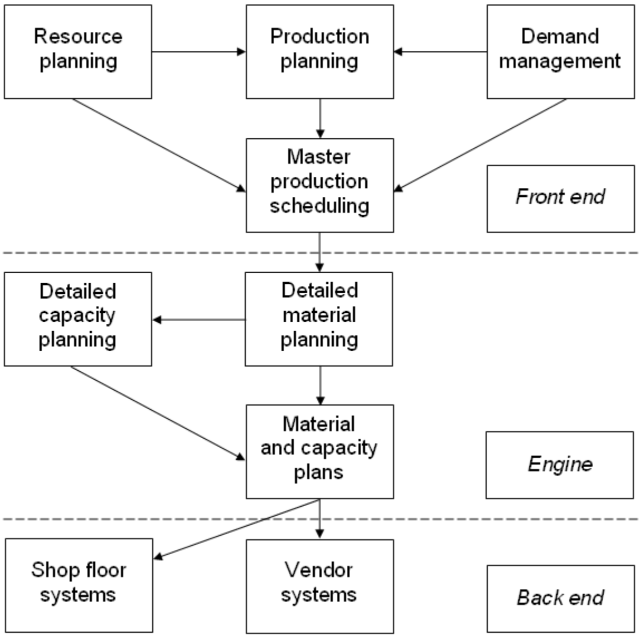


Fig.1. Manufacturing planning and control system

The most standardized XML tool for APS design and implementation is PSLX which covers all aspects of business and manufacturing activities in this area:

a) plan – represent detailed plan of material flows and capacity and has three ontologies: Production plan, Capacity plan, Inventory plan,
b) order – represents document which is a trigger that creates and executes every action and operation; this aspect has tree ontologies: Work order, Lot order, Task order,
c) physical – represents non tense real objects and has one - Resource ontology,
d) party – represents an owner of resources and has three ontologies: Maker, Supplier, Customer,
e) engineering – represents technical knowledge and has four ontologies: Function, Capability, Item, Event,
f) temporal – defines physical features change in time as the results of interaction between physical and scheduling aspects, and has three ontologies: Capacity, Inventory, Change,
g) time – defines time moment and duration, and has three ontologies: Time point, Time period, Time span,
h) space – defines space dimensions and has three ontologies: Position, Region, Distance,
i) scheduling – represents the relationship of actions in time and space, and has four ontologies: Operation, Task, Lot, Action.

2.1. Model of planning and scheduling system

Described above ontologies have relations with each other, representing their semantics depending on their positions. As the relations have multi-dimensional nature only the main semantics are shown in Figure 2.

The right column in Figure 2 symbolises something that is manufactured or consumed by production while, the left side represents something that is utilized by production process.

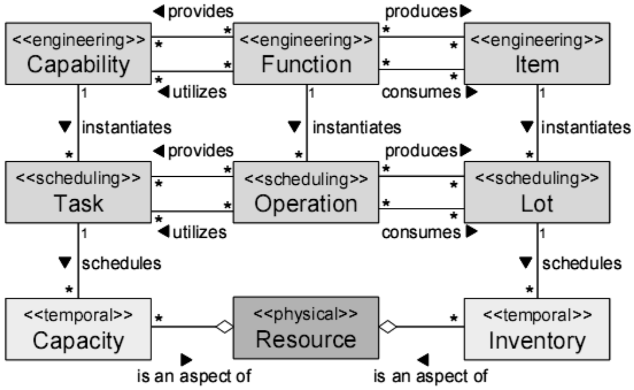


Fig.2. PSLX ontology

Although PSLX ontology has many advantages it is too abstractive for the average user so more detailed vocabularies are necessary. The next sections give the several abstract data models (set of objects and relations) for key APS aspects: plan, order, physical, and scheduling.

2.2. Model of plan and order

Order classes have various kinds of orders in business activities, corresponding to the order class and its sub-classes in the APS domain model. The following classes are the member of this type: customer order, purchase order, production order, work order, ship order, receive order, issue order, and store order.

Figure 3 represents relations of plan and order: objects in the aspect of order create an object in the scheduling aspect, while objects in the aspect of plan are aggregated from the objects of orders.

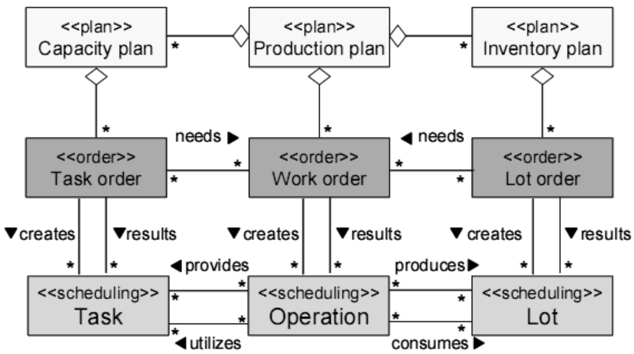


Fig. 3. Model of plans and orders

2.3. Model of physical system

Model of manufacturing system consists of various physical elements.

A class corresponding to Resource ontology and its sub-classes are illustrated in Figure 4. Resource class has the following sub-classes: Personnel, Equipment, Tool, and Material. Equipment further has five sub-classes as Machine, Tester, Route, Storage, and Supporter.

These objects have two different functional characteristics: consumable and non-consumable resources. Most of resources can be classified into one of those characteristics, while a few classes have both. Figure 4 represents these characteristics by connecting to either Capacity or Inventory: classes connected to Capacity are non-consumable, while classes connected to Inventory are consumable.

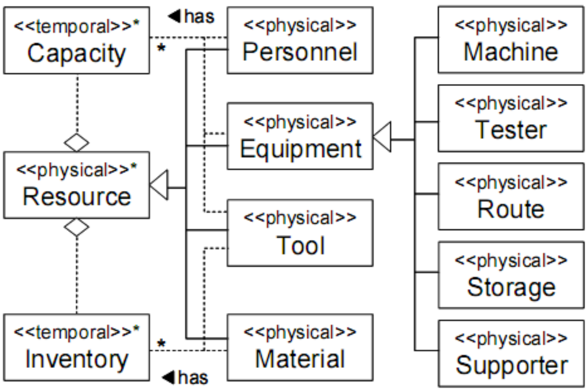


Fig. 4. Sub-classes of Resource ontology

2.4. Model of schedule

Scheduling classes deal with a certain actions on the time horizon, defining information associated to any kind of operations in plant floors. The main class that corresponds to the ontology is operation class; this class has the following sub-classes: Manufacturing, Setup, Inspection, Transport, Administration, Product design, Plant engineering, Maintenance, Ship good, Receive good, Issue inventory, and Store inventory.

2.5. XML common schema

There is need for more implementation-oriented model to realize interoperability of the business data in computer systems. PSLX offers two common schemas for this purpose: the PSLX-RDB and the PSLX-XML common schema.

Classes in the PSLX-XML common schema have hierarchical structure; the following are all classes and their categories:

a) primitive elements – represent main PSLX ontologies; there are: *customer*, *supplier*, *item*, *resource*, *function*, *operation*, *order*, *lot*, and *task*,
b) administrative elements – used as a supporting data among the primitive elements; there are: *scale*, *description*, *priority*, and *display*,
c) relational elements – represent relations between the primitive elements; there are: *compose*, *composed*, *produce*, *produced*, *consume*, *consumed*, *assign*, *assigned*, *predecessor*, *successor*, *pegging*, and *relation*,
d) property elements – used for any class that represents attribute of the primitive elements; there are: *spec*, *location*, *progress*, *load*, *stock*, *available*, and *calendar*,
e) temporal elements – describe time aspects of operation (*start*, *end*, *event*) or order (*release*, *due*),
f) basic data elements – define a value of parameters; there are: *qty*, *price*, *char*, *duration*, and *time*,
g) data auxiliary elements – specify a range of data; there are: *min*, *max*, *earliest*, *latest*, *shortest*, *longest*, *enumerate*.

It is worth to note that all mentioned above classes correspond to XML tags which are used in PSLX document representation.

2.5. PSLX messages

PSLX fully describes the form of transaction messages, which can represent messages between sender (e.g. customer) and receiver in the case of unidirectional communication or initiator (e.g. cooperative plant) and responder in the case of bidirectional communication. The structure of a message document is shown in Figure 5.

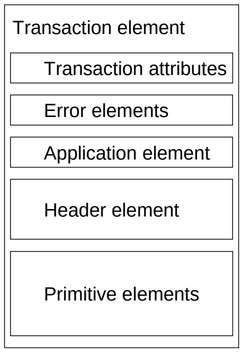
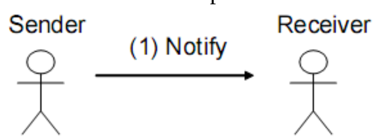


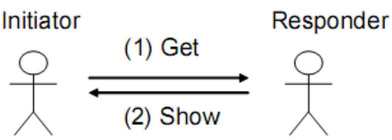
Fig. 5. Structure of a message document

Seven messages types are defined in PSLX: Notify, Get, Show, Add, Change, Remove and Confirm. In order to ask the confirmation, the messages may have an attribute of the following confirmation requests: Never, On Always, On Error. PSLX standard also defines four simple messaging patterns:

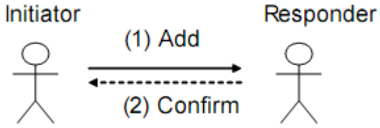
a) *Notify* – the sender sends a Notify message to the receiver; no obligation on the receiver to respond to the message,



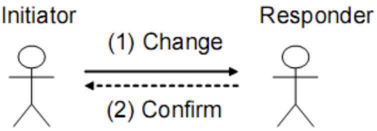
b) *Get-Show* – the sender sends a Get message to the responder in order to get some information; the responder tries to answer the request by sending Show message with corresponding information that is managed,



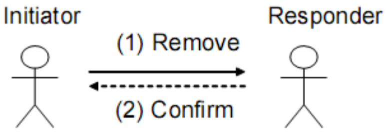
c) *Add* – the sender sends an Add message to request to add information to the database that is managed by the responder; the responder can send a Confirm message depending on the confirmation request,



d) *Change* – the sender sends a Change message to request some changes to the database that is managed by the responder; the responder can send a Confirm message depending on the confirmation request,



e) *Remove* – the sender sends a Remove message to delete particular information in the database managed by the responder; the responder can send a Confirm message depending on the confirmation request.



3. The XML example of foundry model

This Section provides example instances of PLSX documents being used in the order-to-schedule process. This example follows the general schema presented in Figure 6.

Fig. 6. PSLX general schema of APS system

The exemplar manufacturer, Polskie Odlewnie SA, is high automated, universal foundry that produces wide variety of castings in make-to-order system.

3.1. Customer order

The first document illustrates the ordering of car castings. The customer, Polskie Samochody SA, orders several different items from Polskie Odlewnie SA. The buyer knows the supplier's codes for the items and the price for each.

```
<CustomerOrder id="ZAM230/07" sender="PSSA"
action="notify">
  <Header count="2"/>
  <Property type="target" name="party" value="PSSA"/>
  <Item id="CHC123" name="Cylinder head cover no. 123">
    <Compose type="child" item="CoreCHC123">
    <Qty value="2"/></Compose>
    <Price unit="USD" value="52.10"/>
    <Qty value="5"/>
    <Time value="2007-02-14T14:00:00"/>
  </Item>
  <Item id="CHC124" name="Cylinder head cover no. 124">
    <Compose type="child" item="CoreCHC124">
    <Qty value="1"/></Compose>
    <Price unit="USD" value="48.50"/>
    <Qty value="10"/>
    <Time value="2007-02-14T14:00:00"/>
  </Item>
</CustomerOrder>

<Party id="PSSA" type="customer">
  <Char name="fullname" value="Polskie Samochody SA"/>
  <Location name="address"><Char value="Samochodowa 10"/>
    </Location>
  <Location name="city"><Char value="Tychy"/></Location>
  <Location name="code"><Char value="43-222"/></Location>
  <Location name="country"><Char value="PL"/></Location>
</Party>
```

Fig. 7. Customer order and Party example instance

The customer order presented in Figure 7 is a notification type message. This means the producer does not need to confirm the order. In this paper we use only the simplest Notify action, however, the remaining messaging patterns can be used as well. For example if a customer requires the confirmation of the order its message should contain the following properties in the <CustomerOrder> tag:

```
<CustomerOrder id="ZAM230/07" sender="PSSA"
action="Add" confirmation="Always">
```

3.2. Production order

After the customer's order is accepted for production, a production order has to be generated. It contains the operations necessary to produce desired items and the assignments of the main resources such as machines and materials. The next document (Figures 8) illustrates the production order generated on the basis of the customer order shown in Figure 7.

```
<ProductionOrder id="PROD211/07" action="Notify">
  <Header count="3">
    <Property type="target" name="order"
    value="ZAM230/07"/>
  </Header>
  <Operation id="MoldingCHC123">
    <Produce item="CHC123" value="5"/>
    <Consume item="CoreCHC123" value="2"/>
    <Assign resource="Machine1" type="machine"/>
    <Qty unit="min" value="8"/>
  </Operation>
  <Operation id="PouringCHC123">
    <Produce item="CHC123" value="5"/>
    <Consume item="iron 500">
    <Qty unit="kg" value="3"/>
    </Consume>
  </Operation>
  <Operation id="FinishingCHC123">
    <Produce item="CHC123" value="5"/>
    <Assign resource="Ocz2">
    <Qty unit="min" value="3"/>
    </Assign>
  </Operation>
</ProductionOrder>
```

Fig. 8. Work order example instance

The values for the production order have to be taken from operational sheets of the castings. The PSLX specification does not allow for direct technology writing. However, some of the core elements of the PSLX schema can be used in order to do so. Sample document presenting such technological record is shown in Figure 9.

```
<Item id="CHC123" name="Cylinder head cover no. 123">
  <Compose type="operation" operation="MoldingCHC123"
    resource="Machine1" disjunctive="yes">
      <Duration value="P0Y0M0DT0H3M10S"/>
  </Compose>
  <Compose type="operation" operation="MoldingCHC123"
    resource="Machine2" disjunctive="yes">
      <Duration value="P0Y0M0DT0H4M05S"/>
  </Compose>
  <Compose type="operation" operation="FinishingCHC123"
    resource="FMachine2" >
      <Duration value="P0Y0M0DT0H0M50S"/>
  </Compose>
  <Consume type="resource" operation="Pouring"
    resource="iron 500">
      <Qty unit="kg" value="3"/>
  </Consume>
  <Consume type="item" operation="MoldingCHC123"
    item="CoreCHC123" value="2"/>
</Item>
```

Fig. 9. Operations sheet example instance

The presented operation sheet lacks the succession of the operations. This can be done in the way shown in Figure 10.

```
<Operation id="FinishingCHC123">
  <Relation type="precedence" operation="Pouring">
    <Duration value="P0Y0M0DT3H0M0S"/>
  </Relation>
</Operation>
```

Fig. 10. Operations succession written in PSLX

3.3. Work order

Opposite to the production order, which is related to particular items ordered, a work order is related to each work centre (production) or manufacturing resource (maintenance). The work order must contain starting and ending times of all operations planned for a given date as well as item identification and the quantity to be produced. This is presented in Figure 11.

```
<WorkOrder id="M1/22/03/07" action="Notify">
  <Header count="3">
    <Property name="Date" value="2007-03-20"/>
    <Property name="Resource" value="Machine1"/>
  </Header>
  <Operation id="MoldingCHC123">
    <Produce item="CHC123" value="5"/>
    <Consume item="CoreCHC123" value="2"/>
    <Assign resource="Machine1"/>
    <Start value="2007-03-07T09:00:00"/>
    <End value="2007-03-07T10:30:00"/>
  </Operation>
  <Operation id="MoldingBR56">
    <Produce item="BR56" value="30"/>
    <Consume item="CoreBR56" value="3"/>
    <Assign resource="Machine1"/>
    <Start value="2007-03-07T10:30:00"/>
    <End value="2007-03-07T12:30:00"/>
  </Operation>
  <Operation id="MoldingRS254">
    <Produce item="RS254" value="100"/>
    <Assign resource="Machine1"/>
    <Start value="2007-03-07T12:30:00"/>
    <End value="2007-03-07T13:30:00"/>
  </Operation>
</WorkOrder>
```

Fig. 11. Work order example instance

Capacities and inventory levels

Work orders are created on the basis of the production orders, capacities of the resources and the inventory levels of materials, and items consumed during particular operations. The resources can represent machines or people and their capacities may very depending on date as well as location. For example Person A can be available at Location X for 4 hours and at Location Y for the remaining work time in a day.

Exemplar XML documents containing such data are shown in Figure 12.


```
<Resource id="Machine1">
  <Capacity>
    <Date value="2007-03-05T00:00:00"/>
    <Duration value="P0Y0M0DT8H0M0S"/>
  </Capacity>
  <Capacity>
    <Date value="2007-03-06T00:00:00"/>
    <Duration value="P0Y0M0DT7H30M0S"/>
  </Capacity>
</Resource>

<Item id="CRC123">
  <Capacity>
    <Qty value="50"/>
  </Capacity>
</Item>
```


Fig. 12. Capacities and inventory recordings in PSLX

4. Conclusions

Our previous work [3] presented the advantages of unified XML specification to information exchange in e-collaborative environment. Now we concentrate on APS system design and implementation. One of XML tools which can be applied in whole APS life-cycle is PSLX. Although another standards exist for depicting some manufacturing and planning aspects [4] the PSLX dialect seems to be the most comprehensive and suitable for APS design and implementation.

In this paper we present the model of manufacturing and planning system in an exemplar foundry. The model consists of customer's order model, resources model and scheduling model; all these sub-models are written in PSLX language.

Our researches confirm that PSLX is very good standard that could work as:

– reference architecture for APS systems design,
– XML schema for planning and scheduling data communications,
– common model and terminology for planning and scheduling problems.

The model presented here will be the basis for real-life APS system which we want to design for one of polish foundry.

Acknowledgements

This work is supported by the MN from the resources assigned for scientific researches in years 2005 – 2007

References

[1] PSLX white paper, www.pslx.org/en/publicationslx.html

[2] T.E. Vollmann, W.L. Berry, D.C. Whybark, Manufacturing planning and control systems, Irwin, Homewood, 1988.

[3] J. Duda, A. Stawowy, The use of XML standard for production planning in foundries, Archives of Foundry vol. 6, No. 19 (2006) 73–80 (in Polish).

[4] Universal Business Language v2.0, http://docs.oasis-open.org/ubl/os-UBL-2.0