

PSLX仕様の概要と スケジューラへの組み込み

2001年10月3日

西岡靖之

法政大学工学部

もくじ

- はじめに
- PSLXの基本要素
- さまざまな関係の記述
- 既存スケジューラへの組み込み

広義のPSLXと狭義のPSLX

- 広義のPSLX

- APSを中核とした製造業の新しいアーキテクチャであり、個別の計画機能が柔軟に連携することで市場環境に即応する全体最適を実現するしくみを指す。

- 狭義のPSLX

- XMLをベースとした生産計画・スケジューリング関連情報の交換方式および記述方式であり、複数の業務システム間でデータの受け渡しを行う場合の標準仕様。

PSLX仕様(狭義)の内容

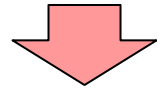
- 対象とする生産計画・スケジューリング問題の内容を表現するためのクラス群
- それらのクラス(またはクラス間の関係)を定義するためのXMLタグ仕様
- XMLデータをアプリケーション間で送受信する場合の手順と付帯情報
- XMLデータ内容の意味的整合性の検査条件とエラー時の処理ルール

スケジューラがPSLX対応になると、

- スケジュールした内容を複数の場所で見ることができる。
- 複数の工程（拠点）の計画を連携することができる。
- MESに対して統一形式のインタフェース開発ができる。
- 上位計画との連携をダイナミックにおこなうことができる。

この発表の目的

- PSLXによる記述の雰囲気を知ってもらう。
- PSLXの記述の柔軟性を知ってもらう。
- PSLXの設計思想を理解してもらう。
- PSLXの可能性をイメージしてもらう。



~~PSLXを実際に記述できるようになってもらう。~~

PSLXで記述できるものとできないもののおおよその境界を知ってもらう。

PSLXの基本要素

設計思想

- できるだけ少ないタグで表現する
- 表面的な現象を追わずに背景の構造をとらえる
- パーフェクトを目指さずに合格ラインのクリアを目指す。
- 国際標準や先行する便利な標準には従う。
- インプリメントを仕様策定と完全に平行して行う。
- 意見が分かれた場合には最終ユーザの利益最大化を志向する。

準備運動(その1)

- 以下のものはすべて「品目(アイテム)」と呼ぶことにします。

- 製品、完成品、
- 仕掛品、中間品
- 原材料、部品、資材

準備運動(その1)

- 以下のものはすべて「資源(リソース)」と呼ぶことにします。

- 機械、設備、装置
- 人、作業者、ロボット
- ジグ、工具

準備運動(その1)

- 工場の内外にあって、生産計画・スケジューリングに関係ある“もの”は、すべて「品目」か「資源」のどちらかに分類してください。

ヒント

作業を行うことで“なくなる”または“生まれる”ものは品目。作業中だけ拘束され、終了後にもとの状態にもどるものは資源。

以下のものは品目？資源？

- 作業場所
- 工場、ライン
- 設計図、操作マニュアル
- 切りくず、端材
- 電気、水道
- 臨時アルバイト
- 運搬車両、コンテナ
- 梱包材、納品書
- ロット、バッチ
- 予算、収益

ポイント(その1)

- PSLXでは、顧客およびサプライヤー以外の“もの”は、「品目」か「資源」のいずれかとして記述されます。

判断が難しい場合

- 状況によって変わる場合：機械はそれによって生産する場合には「資源」だが、それが組み立てられる時には品目として扱われる。
- 視点によって変わる場合：水道やガスなどは安定供給されるとみなせば「資源」だが、物理的には消費されているので「品目」。

PSLX記述例

品目の定義

```
<item name=“製品A”>  
  <!-- 製品Aに関する情報 -->  
</item>
```

資源の定義

```
<resource name=“機械A”>  
  <!-- 機械Aに関する情報 -->  
</resource>
```

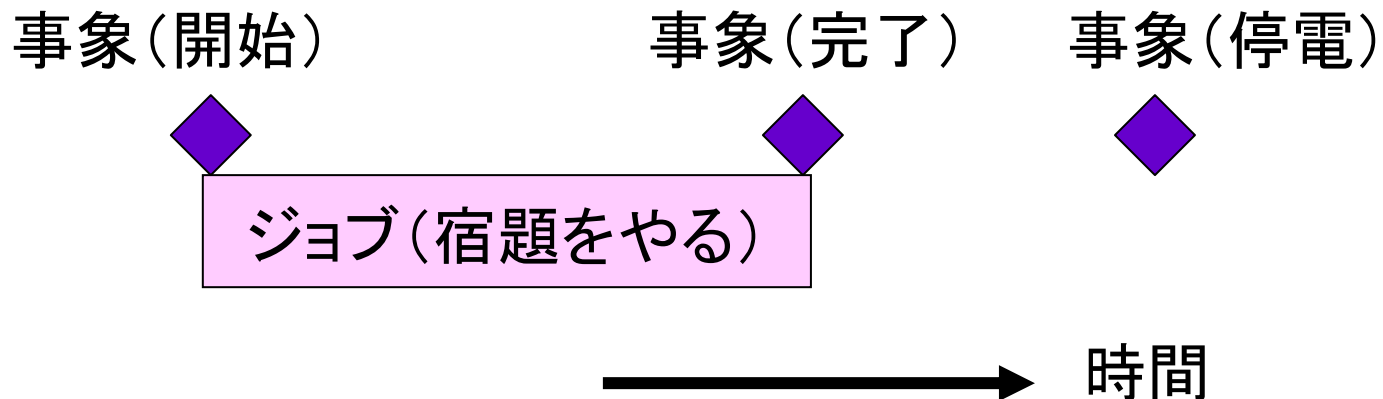
準備運動(その2)

- 以下のものはすべてジョブと呼ぶことにします。

- タスク
- 作業、オペレーション
- プロセス、(工程)
- アクティビティ

ジョブと事象

- 「ジョブ」は、時間幅をもっています。
- 「事象」は、時間幅をもっていません。
- 事象はジョブの一部である場合や、単独で存在する場合があります。



ポイント(その2)

プランニング の対象は「品目」や「資源」が持つ属性や要素となります。

- 品目の数量、品目の仕様、品目そのものの生成、削除、資源の能力、資源の配置、など

スケジューリング の対象は「ジョブ」または「事象」の属性や要素となります。

- ジョブの開始時刻、終了時刻、ジョブが使用する資源、消費する品目、など

PSLX記述例

ジョブの定義

```
<job name=“作業A”>  
  <!-- 作業Aに関する情報 -->  
</job>
```

事象の定義

```
<event name=“事象A”>  
  <time>2001/10/3 13:30:00</time>  
</event>
```

PSLX記述例

ジョブの定義(開始時刻、終了時刻)

```
<job name="作業A">  
  <!-- 作業Aに関する情報 -->  
  <start>  
    <time>2001/10/03 14:00:00</time>  
  </start>  
  <end>  
    <time> 2001/10/03 14:10:00</time>  
  </end>  
</job>
```

準備運動(その3)

- 以下のものはすべてオーダーと呼ぶことにします。

– 確定オーダー	出荷指示
– 内示、見込みオーダー	生産指示
– 製造オーダー	入庫・出庫指示
– 資材要求	残業指示
– カンバン	設備手配

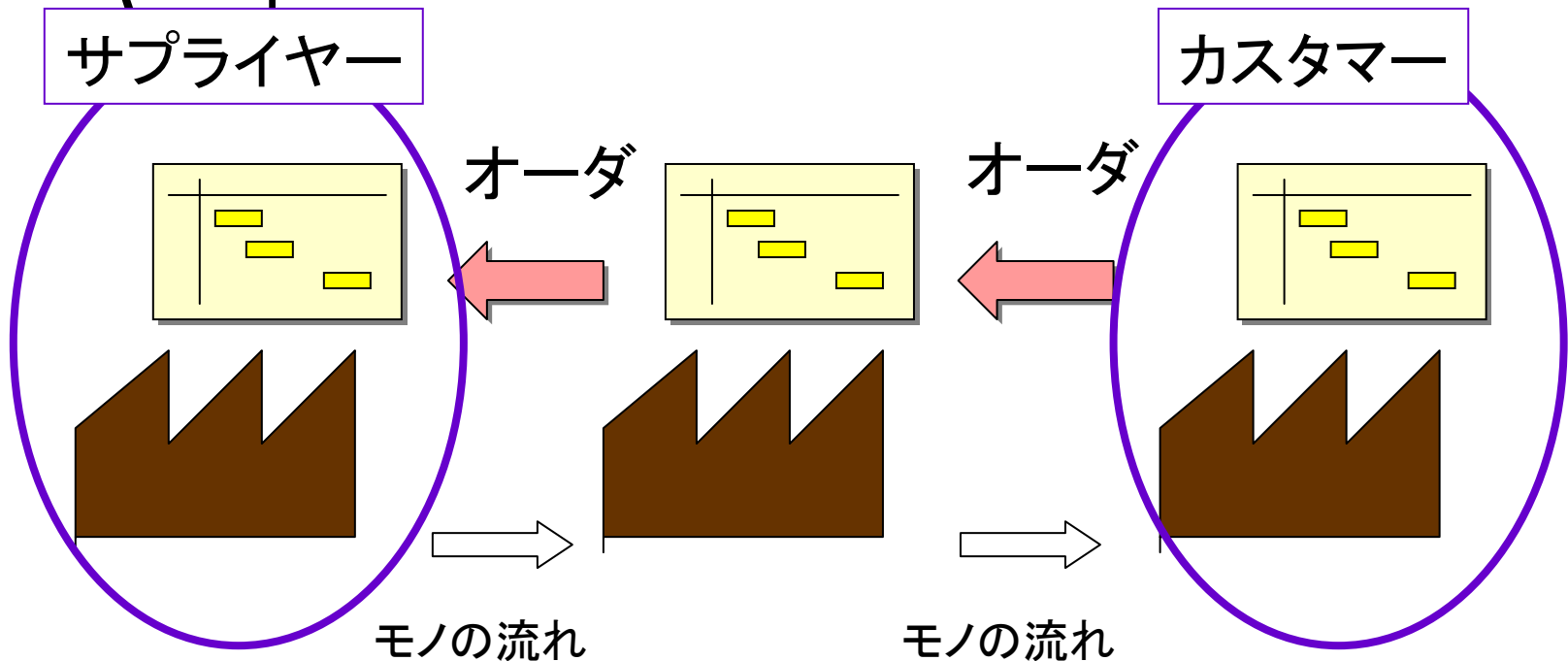
ポイント(その3)

- オーダは、いろいろなところに存在する。プログラミングやスケジューリングを要求するものがオーダである。

- 品目に対するオーダ
- 資源に対するオーダ
- ジョブに対するオーダ
- 事象に対するオーダ

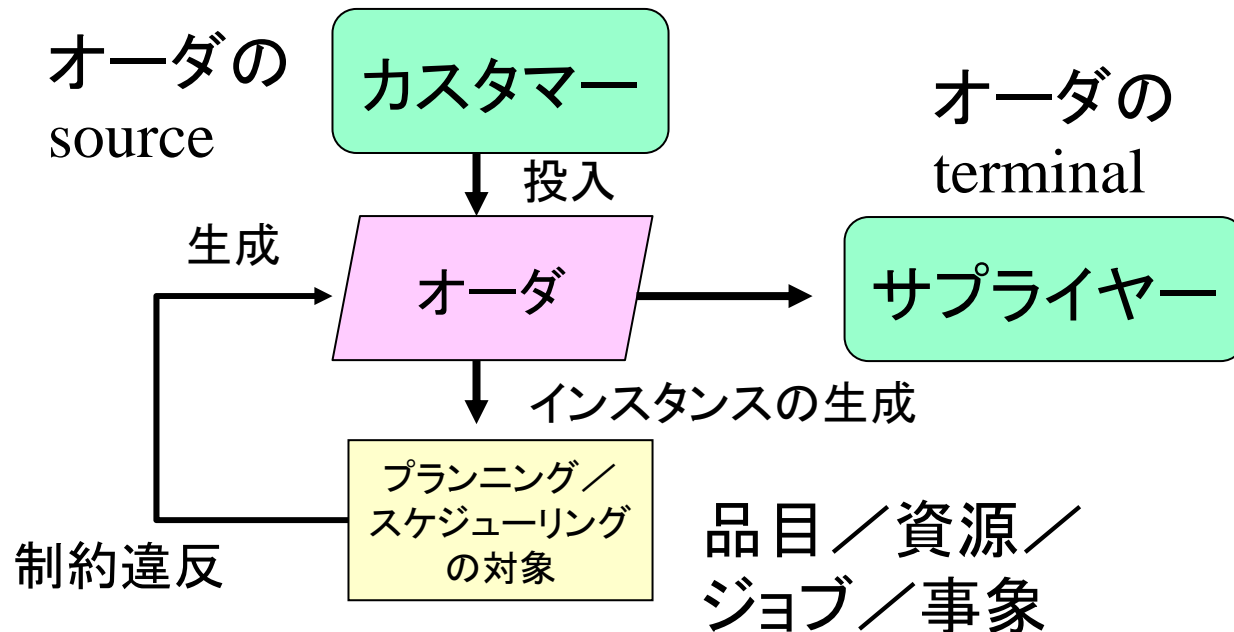
カスタマーとサプライヤー

- オーダを外部で生成するのが「カスタマー」
- オーダを外部で受けるのが「サプライヤー」



ポイント(その4)

- オーダーは最初にカスタマーから投入された後、何度かプランニング／スケジューリング問題の内部で生成され変換された後、最終的にはサプライヤーに送られる。



PSLX記述例

オーダーの定義

```
<order name="注文A">  
  <!-- 注文Aに関する情報 -->  
  <customer ref="A商事"/>  
  <time>  
    <latest>2001/10/10 12:00:00</latest>  
  </time>  
  <item ref="製品A"/>  
  <qty>100</qty>  
  <location ref="送先A"/>  
</order>
```

顧客名

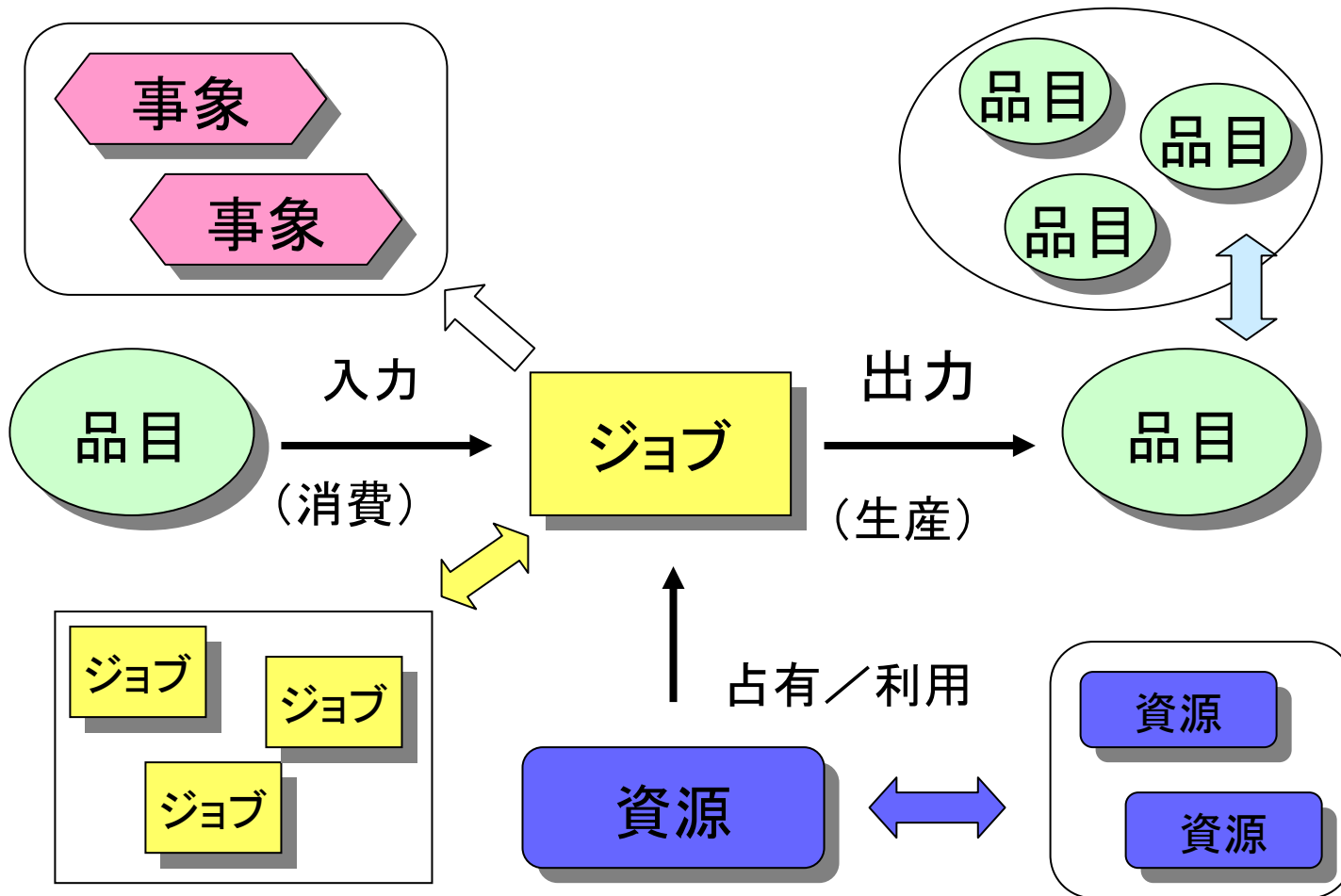
納期

品目

数量

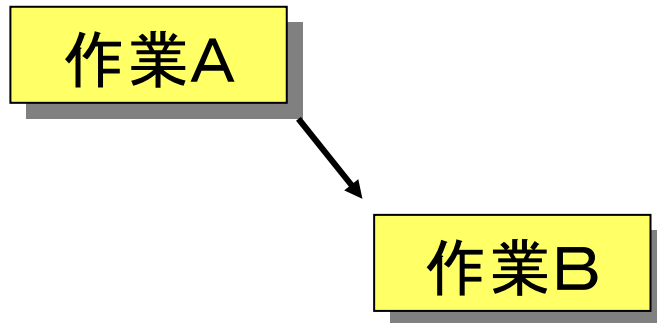
送り先

ここまでのまとめ



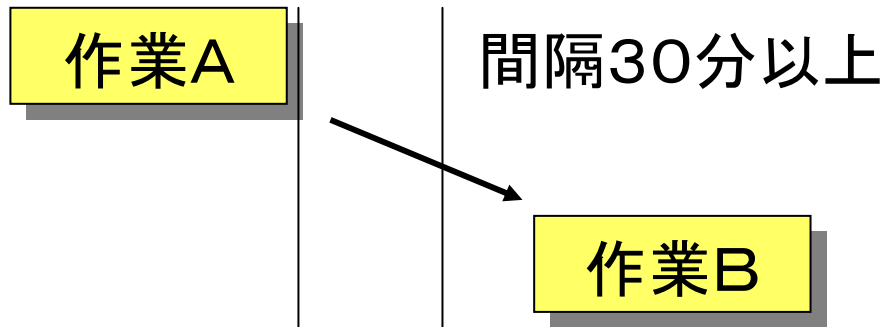
さまざまな関係の記述

ジョブとジョブ(先行関係)



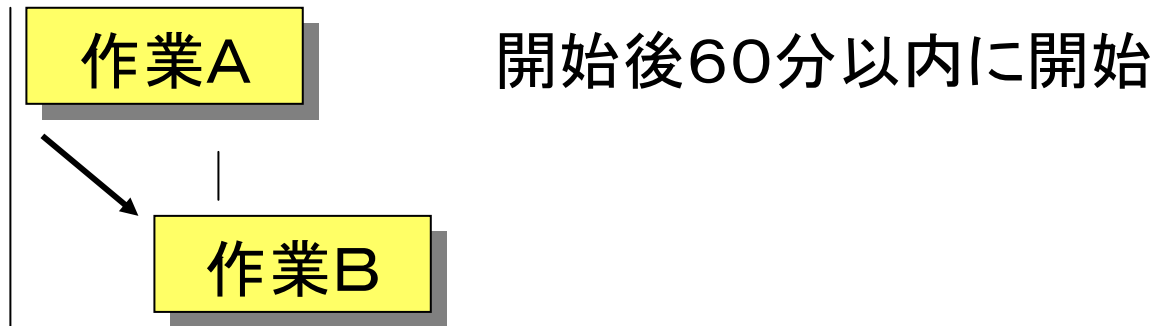
```
<job name="作業A">
  <successor ref="作業B"/>
</job>
<job name="作業B">
  <predecessor ref="作業A"/>
</job>
```

ジョブとジョブ（先行関係：応用1）



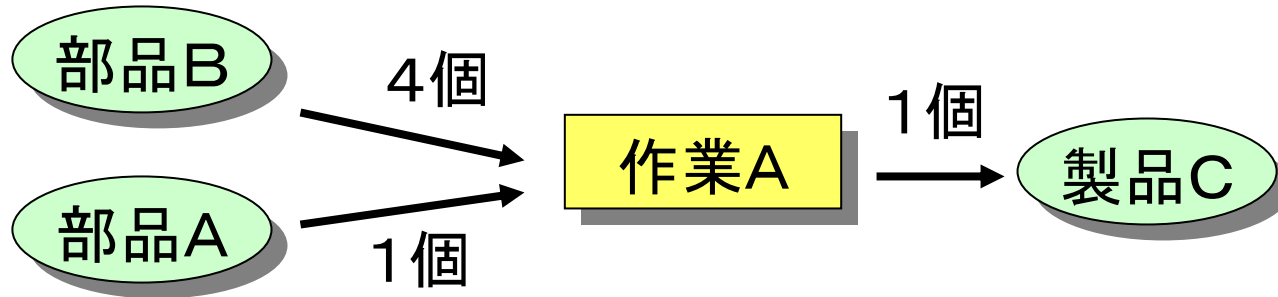
```
<job name="作業A">  
  <successor ref="作業B">  
    <duration>  
      <shortest>00:30:00</shortest>  
    </duration>  
  </successor>  
</job>
```

ジョブとジョブ（先行関係：応用2）



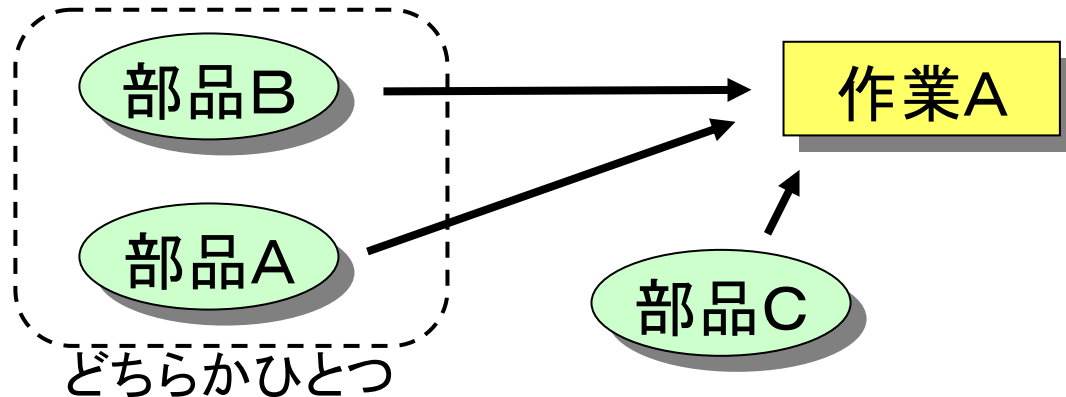
```
<job name="作業A">  
  <successor ref="作業B" type="ss">  
    <duration>  
      <longest>01:00:00</longest>  
    </duration>  
  </successor>  
</job>
```

ジョブと品目（生産／消費関係）



```
<job name="作業A">
  <consume><item ref="部品A" /></consume>
  <consume>
    <item ref="部品B" /><qty>4</qty>
  </consume>
  <produce><item ref="製品C" /></produce>
</job>
```

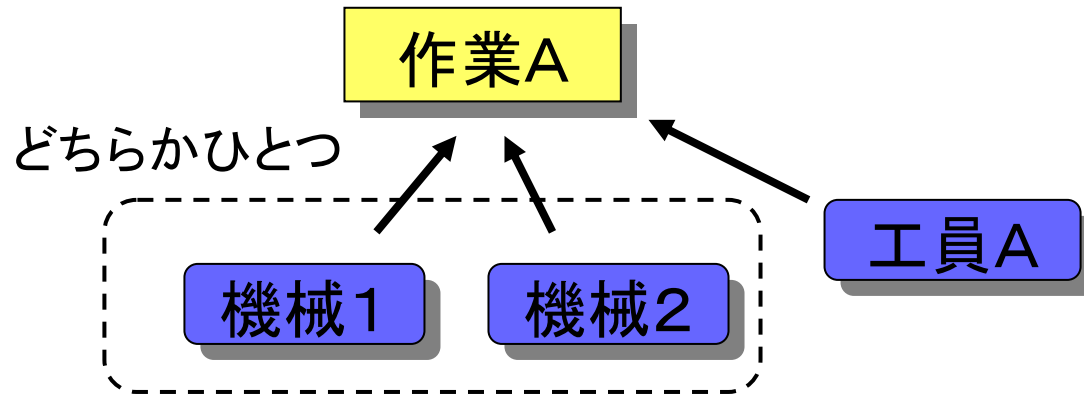
ジョブと品目（生産／消費関係：応用）



```
<job name="作業A">  
  <consume nth="1"><item ref="部品A"/></consume>  
  <consume nth="1"><item ref="部品B"/></consume>  
  <consume nth="2"><item ref="部品C"/></consume >  
</job>
```

nthの値が同じものはひとつにまとめられる

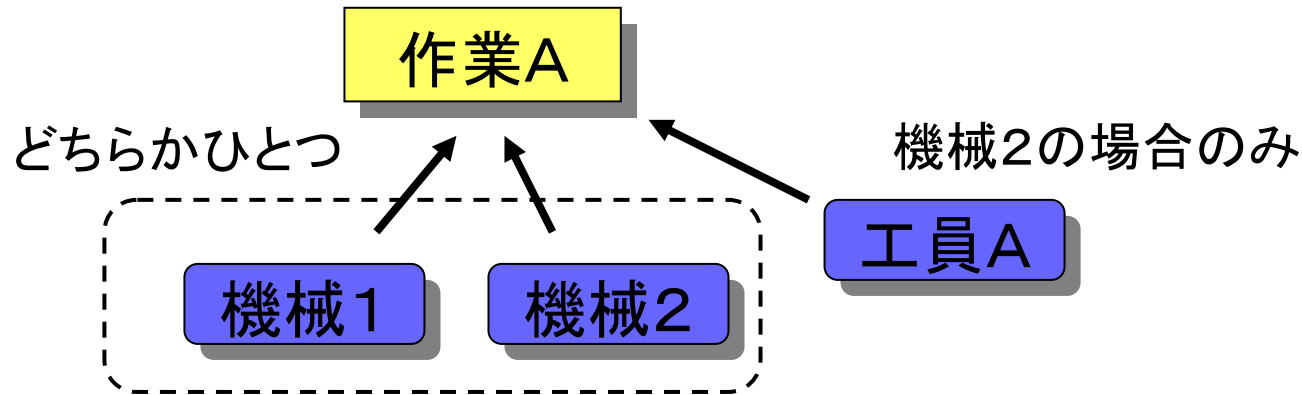
ジョブと資源（割当関係）



```
<job name="作業A">  
  <assign nth="1"><item ref="機械1" /></assign>  
  <assign nth="1"><item ref="機械2" /></assign>  
  <assign nth="2"><item ref="工員A" /></assign>  
</job>
```

nthの値が同じものはひとつにまとめられる

ジョブと資源（割当関係：応用）



```
<job name="作業A">  
  <assign nth="1" sel="1" ><item ref="機械1" /></assign>  
  <assign nth="1" sel="2" ><item ref="機械2" /></assign>  
  <assign nth="2" sel="2" ><item ref="工員A" /></assign>  
</job>
```

selはjob内で同時にひとつしか存在できない

在庫量の推移

製品A



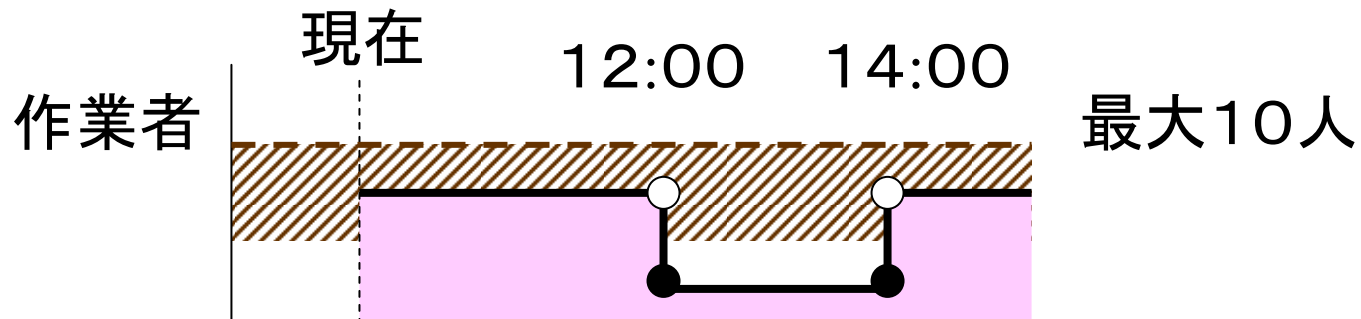
```
<item name="製品A">
  <stock><start><time ref="now"></start>
  <qty>50</qty></stock>
  <stock>
    <start><time>12:00:00</time></start>
    <qty>100</qty>
  </stock>
</item>
```

負荷量の推移



```
<resource name="作業者">  
  <load><start><time ref="now"></start>  
  <qty>8</qty></load>  
  <load><start><time>12:00:00</time></start>  
  <qty>2</qty></load>  
  <load><end><time>14:00:00</time></end>  
  <qty>8</qty></load>  
</resource>
```

負荷量の制約



```
<resource name="作業者">  
  <load>  
    <start><time ref="init"></start>  
    <qty><max>10</max></qty>  
  </load>  
</resource>
```

親子の関係

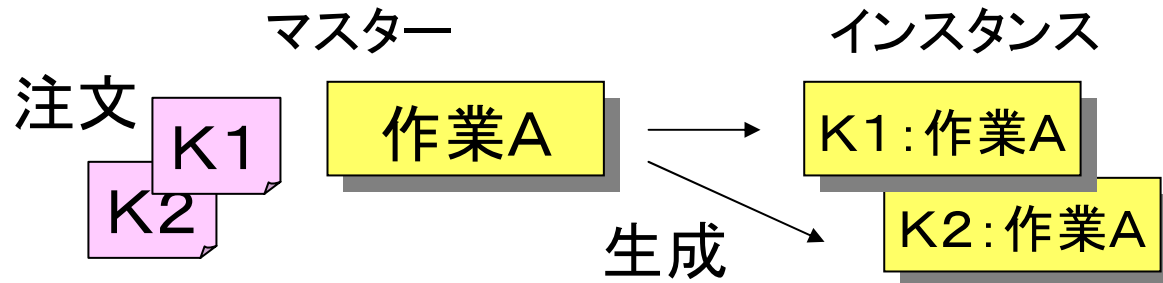
作業A

作業B

ジョブC

```
<job name="ジョブC" />  
<job name="作業A">  
  <parents ref="ジョブC">  
</job>  
<job name="作業B">  
  <parents ref="ジョブC">  
</job>
```

マスターとインスタンス



```
<job name="作業A"/>
<job name="K1: 作業A" master="作業A">
  <order ref="K1"/>
</job>
<job name="K2: 作業A" master="作業A">
  <order ref="K2"/>
</job>
```

既存スケジューラへの組み込み

既存スケジューラをPSLX対応にする

手順1: 既存のデータ構造とPSLXのスキーマとのマッピングを行う。

手順2: マッピングできない部分についての対応を検討する。

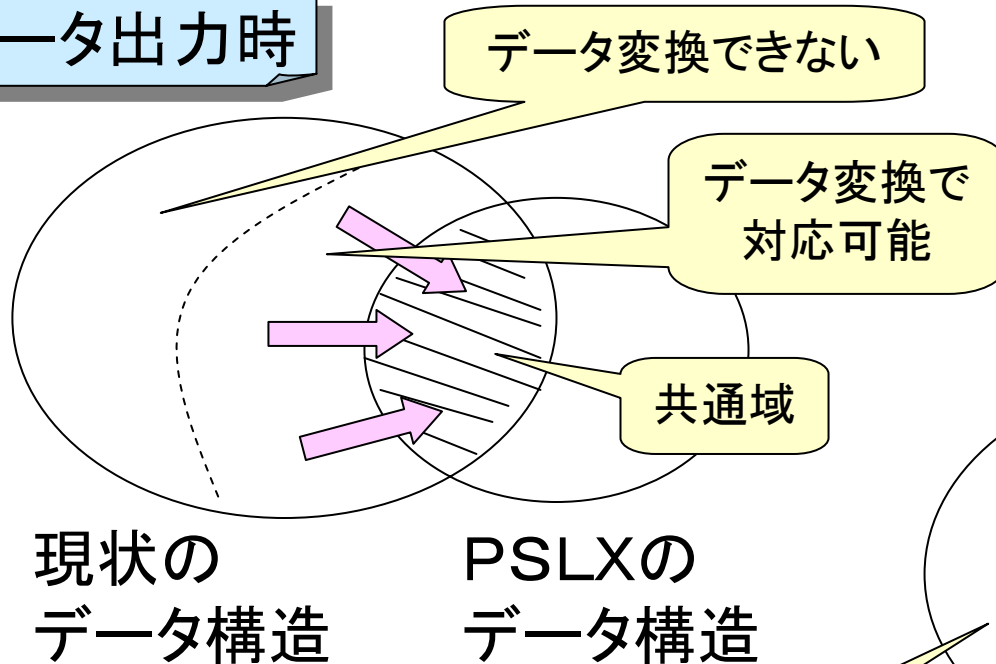
手順3: 必要となる追加情報の入力IFのデザインを行う。

手順4: 外部からの遠隔命令を内部のメソッドとの対応を整理する。

手順5: 実際の開発(プログラミング)

マッピングについて

データ出力時



現状の
データ構造

PSLXの
データ構造

データ変換できない
→ 想像して作る!

データ変換で
対応可能

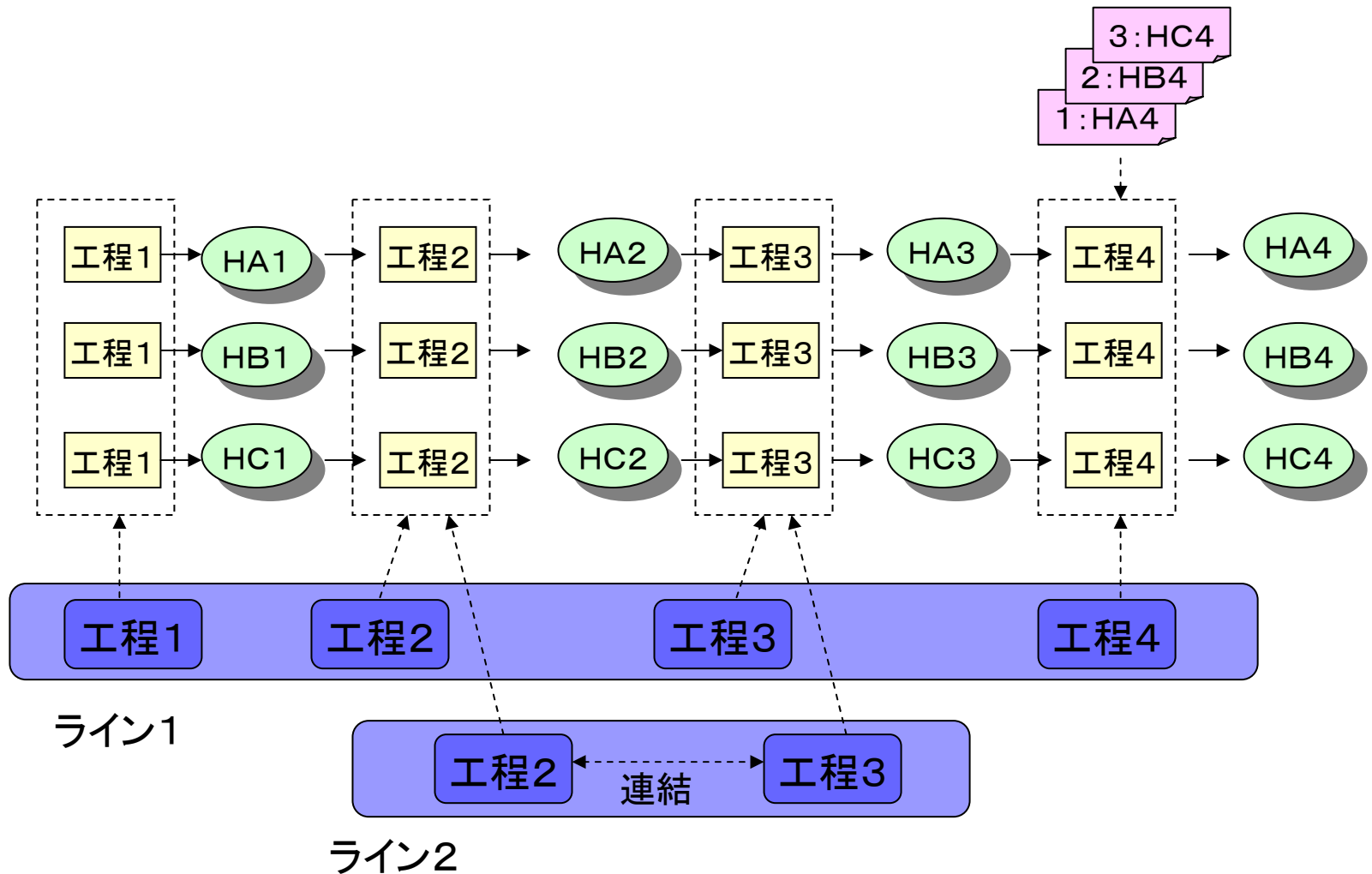
共通域

現状の
データ構造

PSLXの
データ構造

データ入力時

例1: ACCROADの場合



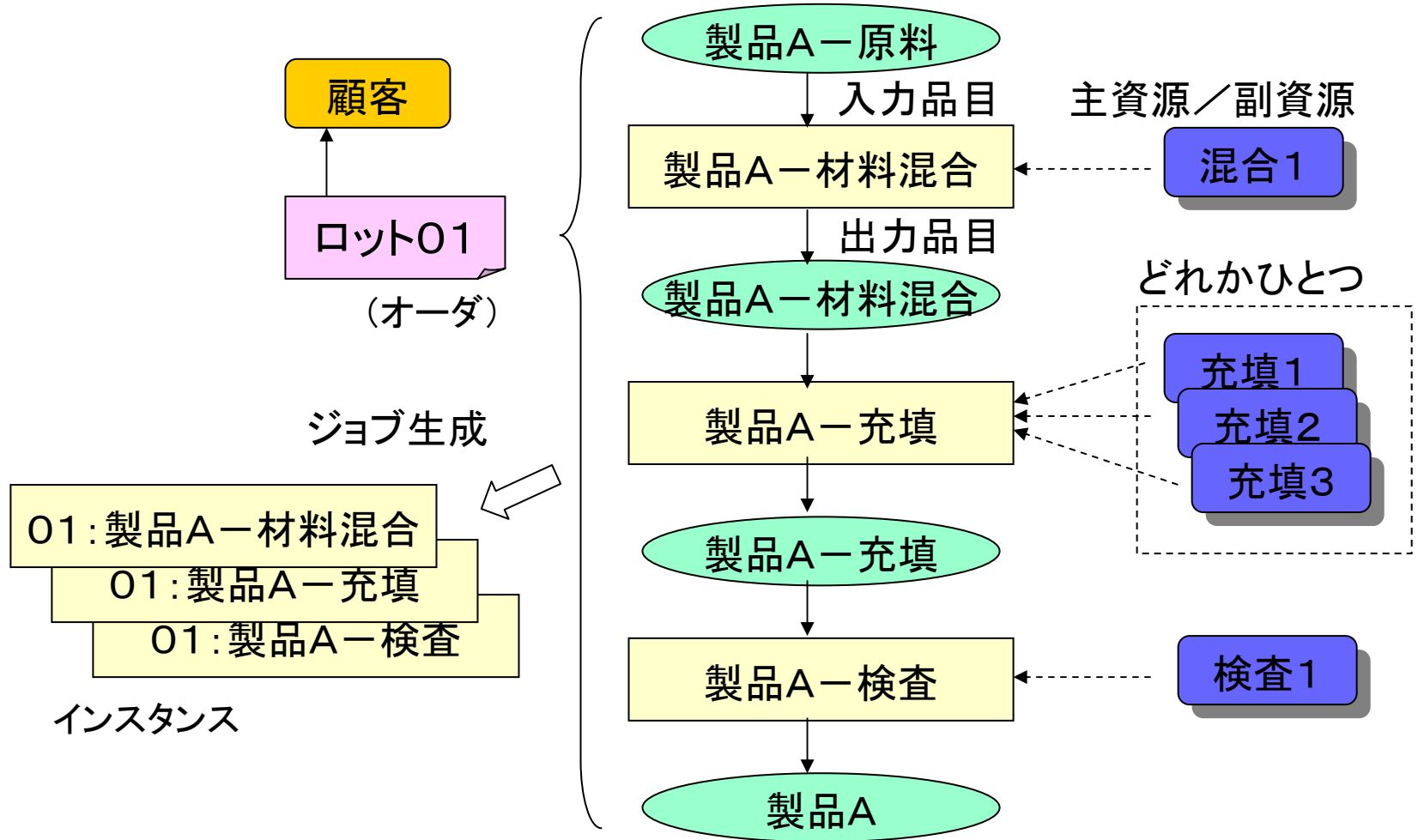
データベース記述サンプル

工程
品目
ライン
品名
時間当生産能力
ロットサイズ
優先順位
段取G

生産能力マスタ

```
<resource name="ライン:工程">  
  <switch>*段取G*</switch>  
</resource>  
<item name="品目">  
  <assign>  
    <resource ref="ライン:工程"/>  
    <priority>優先順位</priority>  
  </assign>  
  <display><name>品名</name></display>  
  <qty><min>ロットサイズ</min></qty>  
</item>  
<job name="ライン:工程:品目">  
  <duration><calc>時間当生産能力</calc>  
  </duration>  
</job>
```

例2: ASPROVAの場合



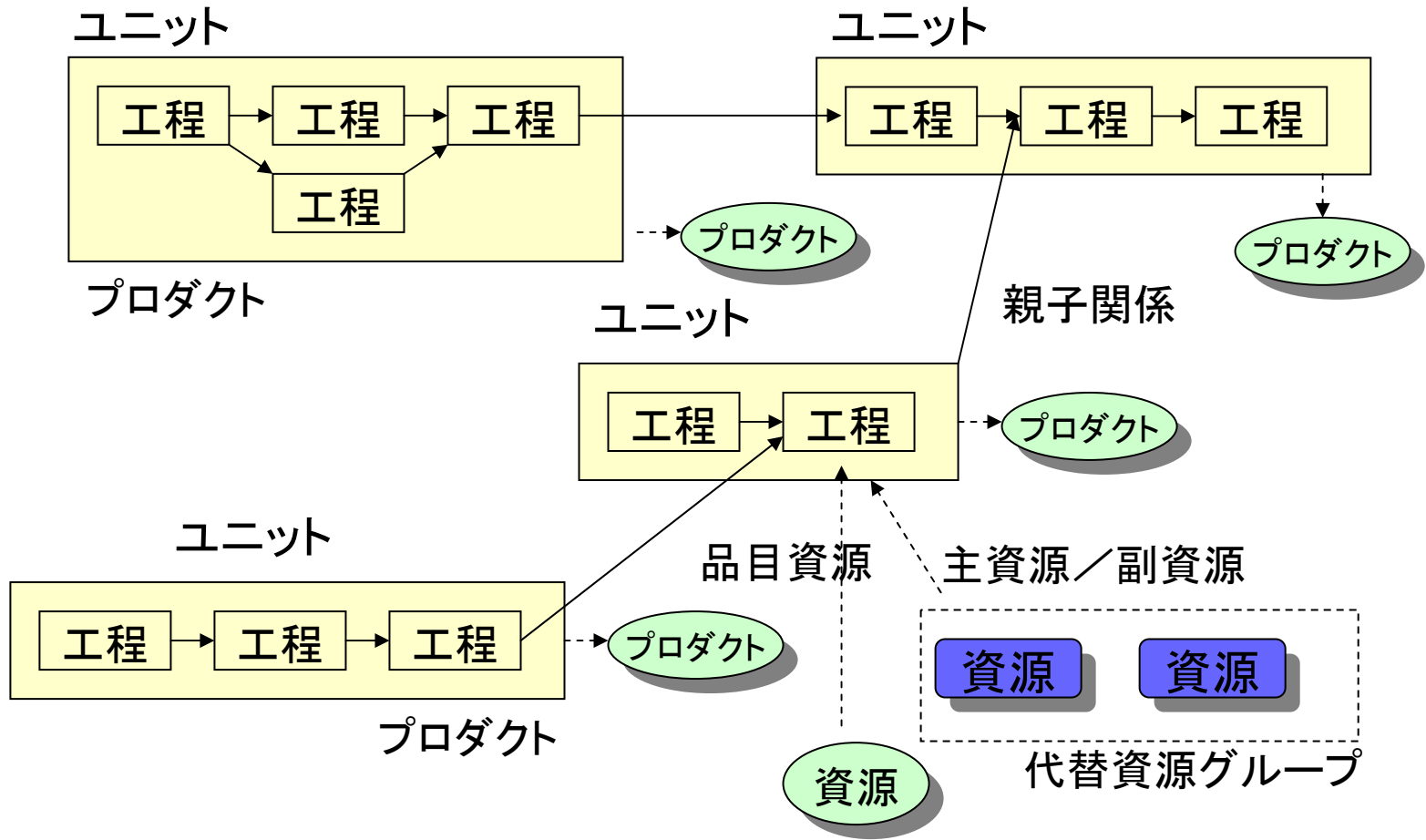
データベース記述サンプル

完成品目
工程順
工程コード
入力品目
出力品目
必要量
重なり方法
重なりMIN

部品表テーブル

```
<job name="工程コード">  
  <consume ref="入力品目">  
    <qty>必要量</qty></consume>  
  <produce ref="出力品目"/>  
  <predecessor ref="前工程コード"  
    type="重なり方法">  
    <duration><min>重なりMIN</min>  
  </duration>  
</predecessor>  
</job>  
<item name="出力品目">  
  <parents ref="完成品目"/>  
</item>
```

例3: DIRECTOR5の場合



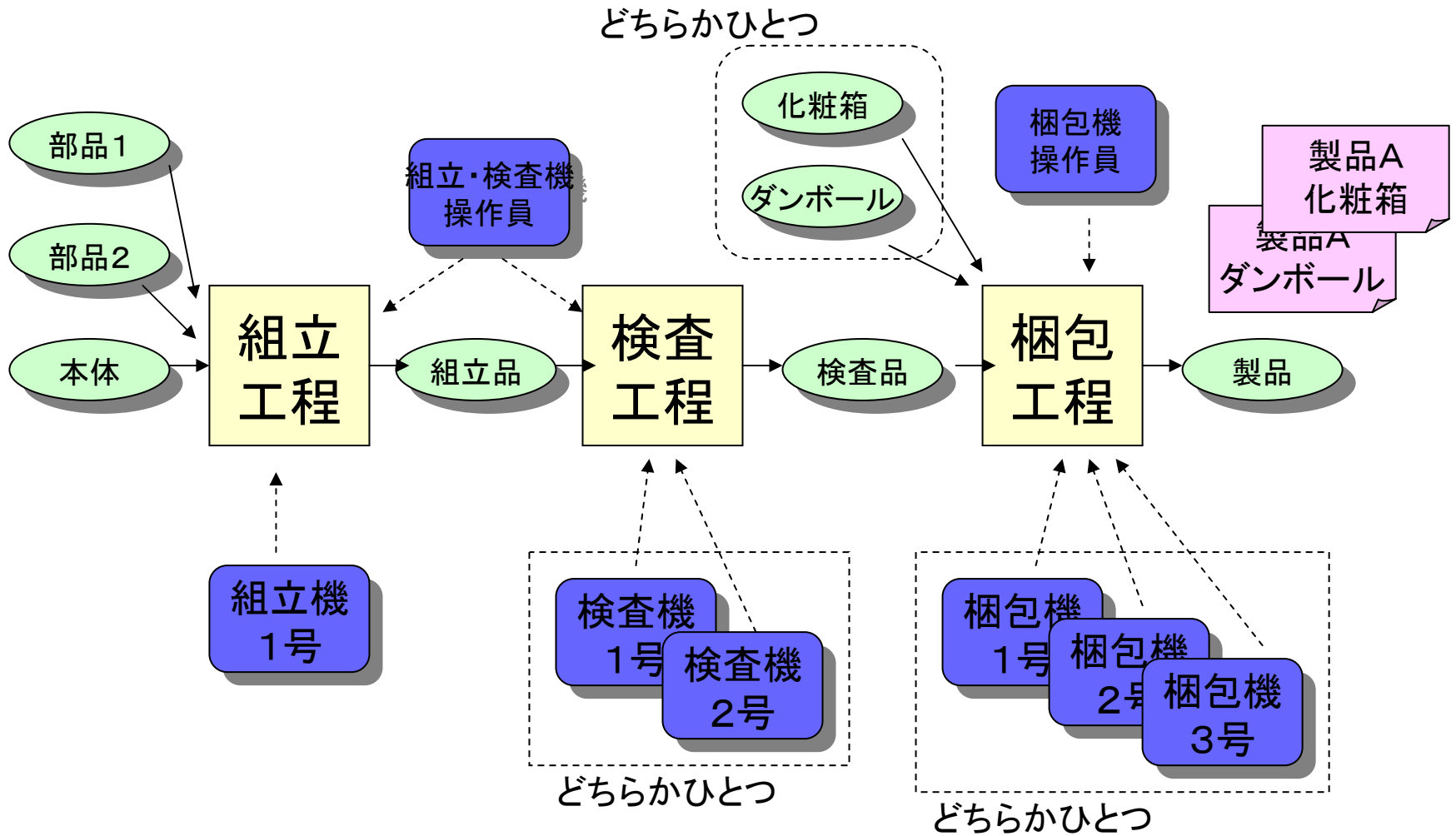
データベース記述サンプル

ユニットID
最上位ユニットID
プロダクトID
オーダーNo
開始可能日
納期
ユニット数量
オーダー数量

UNITラン(オーダー)
テーブル

```
<order name="オーダーNo">  
  <time><latest>納期</latest>  
  <earliest>開始可能日</earliest>  
</time>  
  <item ref="プロダクトID"/>  
  <qty>オーダー数量</qty>  
</order>  
<job name="ユニットID">  
  <parents ref="最上位ユニットID"/>  
</job>  
<item name="プロダクトID">  
  <qty><min>ユニット数量</min></qty>  
</item>
```

例4: JoySchedulerの場合



データベース記述サンプル

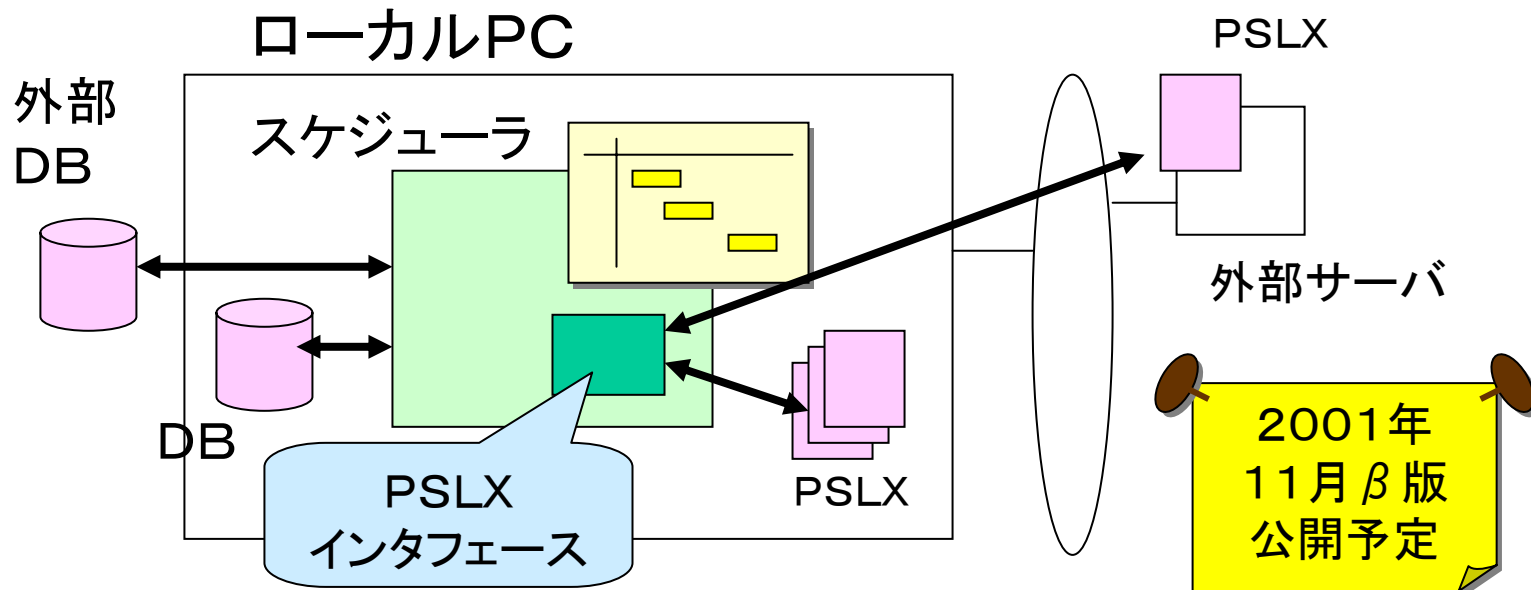
品目コード
品目名
単価
単位品目数
在庫増方法
仕様コード1
安全在庫
まとめ期間

品目マスタ

```
<item name="品目コード">  
  <display><name>品目名</name></display>  
  <priority>品目優先度</priority>  
  <price>単価</price>  
  <qty><min>単位品目数</min></qty>  
  <spec name="仕様コード1"/>  
  <stock>  
    <inc>*在庫増減方法*</inc>  
    <qty><min>安全在庫</min></qty>  
  </stock>  
</item>  
<job>  
  <item ref="品目コード"/>  
  <duration>まとめ期間</duration>  
</job>
```

PSLXインタフェース

PSLXインタフェースは、アプリケーション開発者がXMLの詳細を知らなくてもPSLXを組み込み可能なように、機能の詳細を隠蔽したCOMのクラス。VB, C++, JAVA等から利用できる。



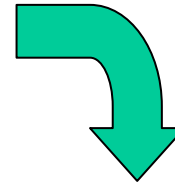
関数名サンプル

関数名	説明
InitProblem	データの初期化を行う
WriteData	設定したデータをXML出力する
ReadData	XMLデータを内部に読込む
CreateOrder	オーダーを一つ生成する
CreateJob	ジョブを一つ生成する
GetNextOrder	次のオーダーをさがす
GetNextJob	次のジョブをさがす
SetDescription	要素の記述内容を設定する
GetDescription	要素の記述内容を取り出す

プログラム (VB: イメージ)

```
Sub 練習問題()  
  InitProblem  
  CreateProblem “練習問題”  
  CreateOrder “注文A”  
  CreateOrder “注文B”  
  WriteData “サンプル”  
End Sub
```

XML生成
と保存



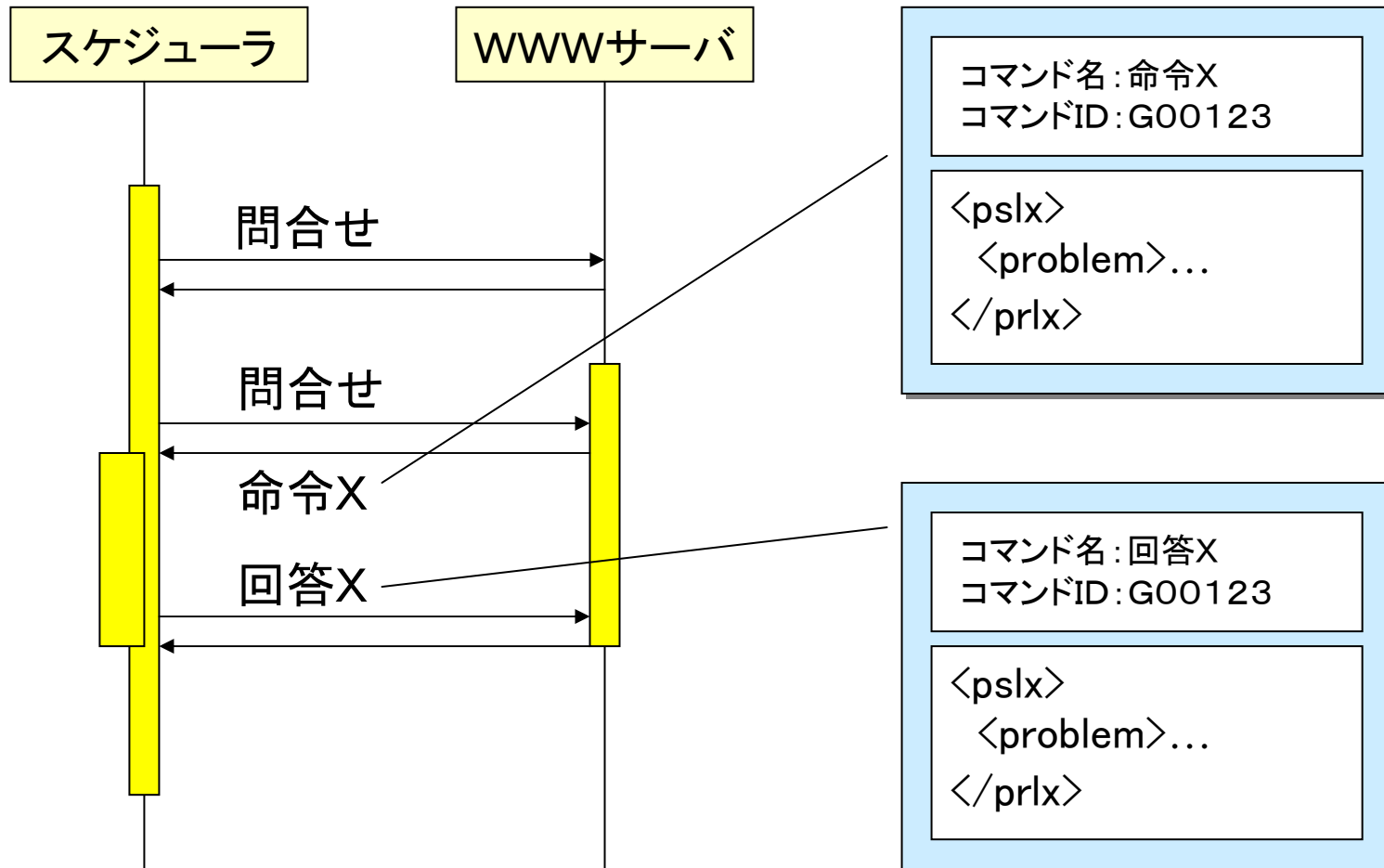
```
<?xml version="1.0" ?>  
<pslx ver="1.0">  
  <problem name="練習問題"/>  
  <order name="注文A"/>  
  <order name="注文B"/>  
</pslx>
```

サンプル.xml

どのようなデータのやりとりがあるか

- スケジュール結果を送る(スケジューラ →)
- 問合せに対しスケジュールの一部を送る(スケジューラ →)
- 注文(内示、確定)を送る(スケジューラ ←)
- マスター情報を送る(スケジューラ ←)
- 実績情報を送る(スケジューラ ←)
- 制約を追加／変更する(スケジューラ ←)

アプリケーション間のやりとり



おしまい

PSLX仕様は、以下のアドレスにて公開しています。
http://www.pslx.org/xml_specification/