

# PSLX標準仕様による スケジューリングソフトの新たな展開

2001年11月27日

法政大学工学部 西岡靖之

nishioka@k.hosei.ac.jp

# もくじ

---

- PSLXの目的と新たなソリューション
- スケジューラーの位置付け
- 狭義のPSLX仕様策定の状況
- PSLXインタフェースによる実装方法
- プロトコルの定義とメッセージ交換
- プロトコルによるプロセスの標準化
- PSLXによるシステムアーキテクチャ

# APSの目的(=PSLXの目指すもの)

---

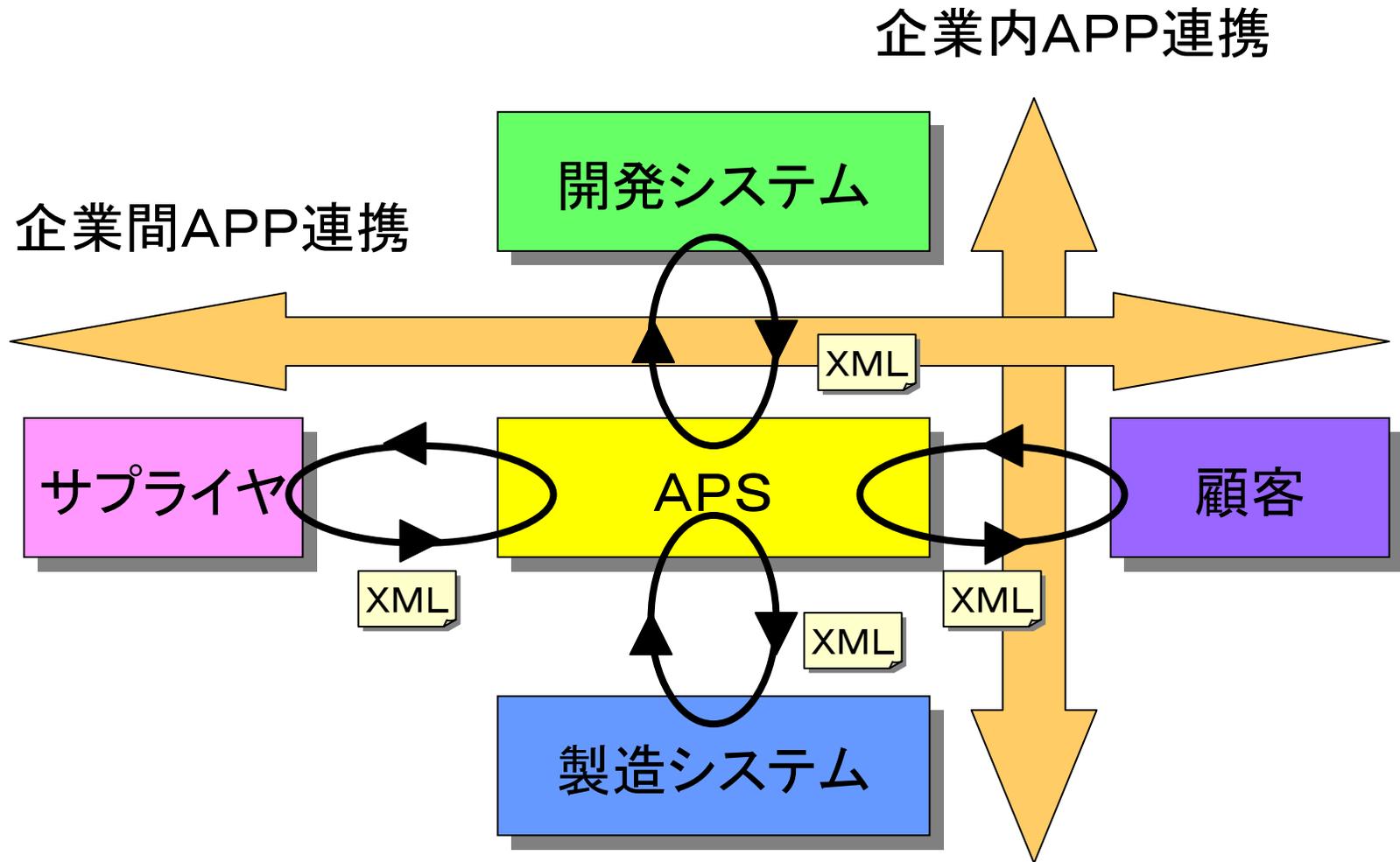
- 変化に俊敏に対応できるしくみをつくる
- リードタイムを極限まで短縮する
- 設計部門と製造部門をつなぐ
- 生産現場の活動を利益に結びつける
- サプライチェーンの全体最適に貢献する

# 新たなソリューションの提案

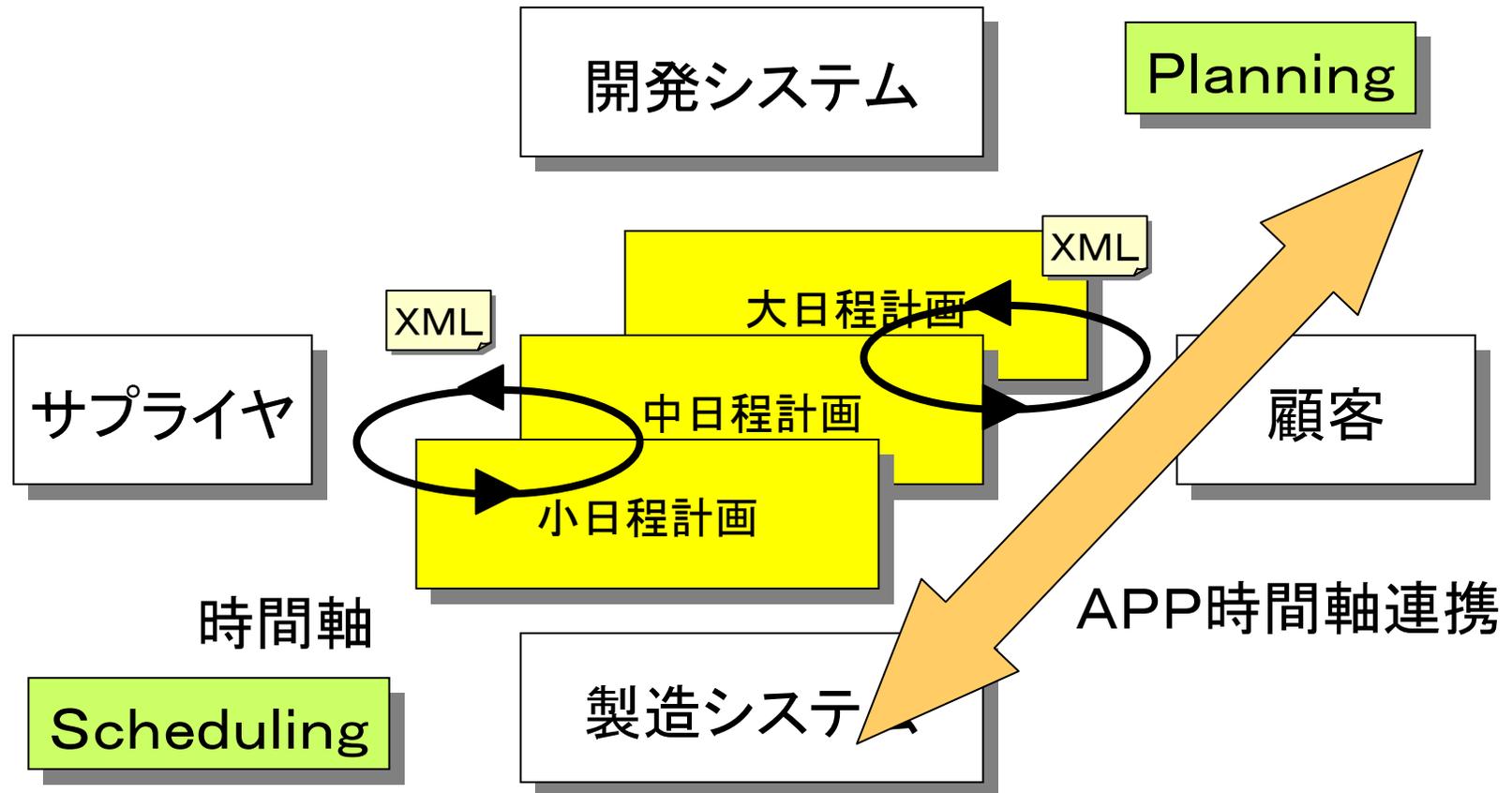
---

- 在庫ではなくフロー情報によるダイナミックな引き当て管理
- コンカレントな製品開発を可能にする設計情報と製造情報の連携
- 超プロジェクト型生産に対応したスムーズな新製品切替管理
- 計画情報の共有による企業間サプライチェーンの最適化
- スループットをベースとした収益に貢献する評価システム

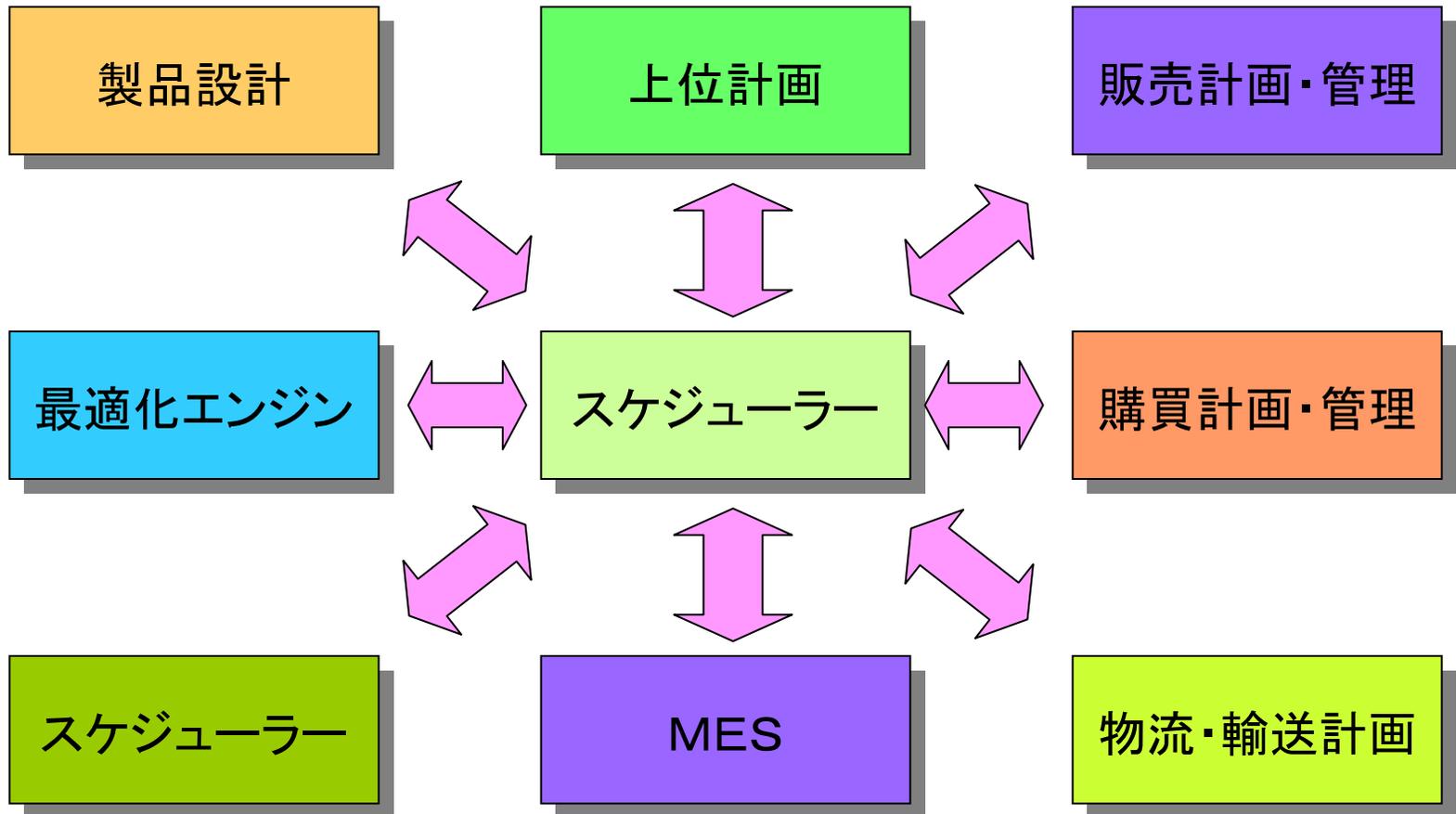
# APSを中心としたアプリケーション連携



# APSを中心としたアプリケーション連携



# 全体最適とスケジューリング問題



# スケジューラがPSLX対応になると、

---

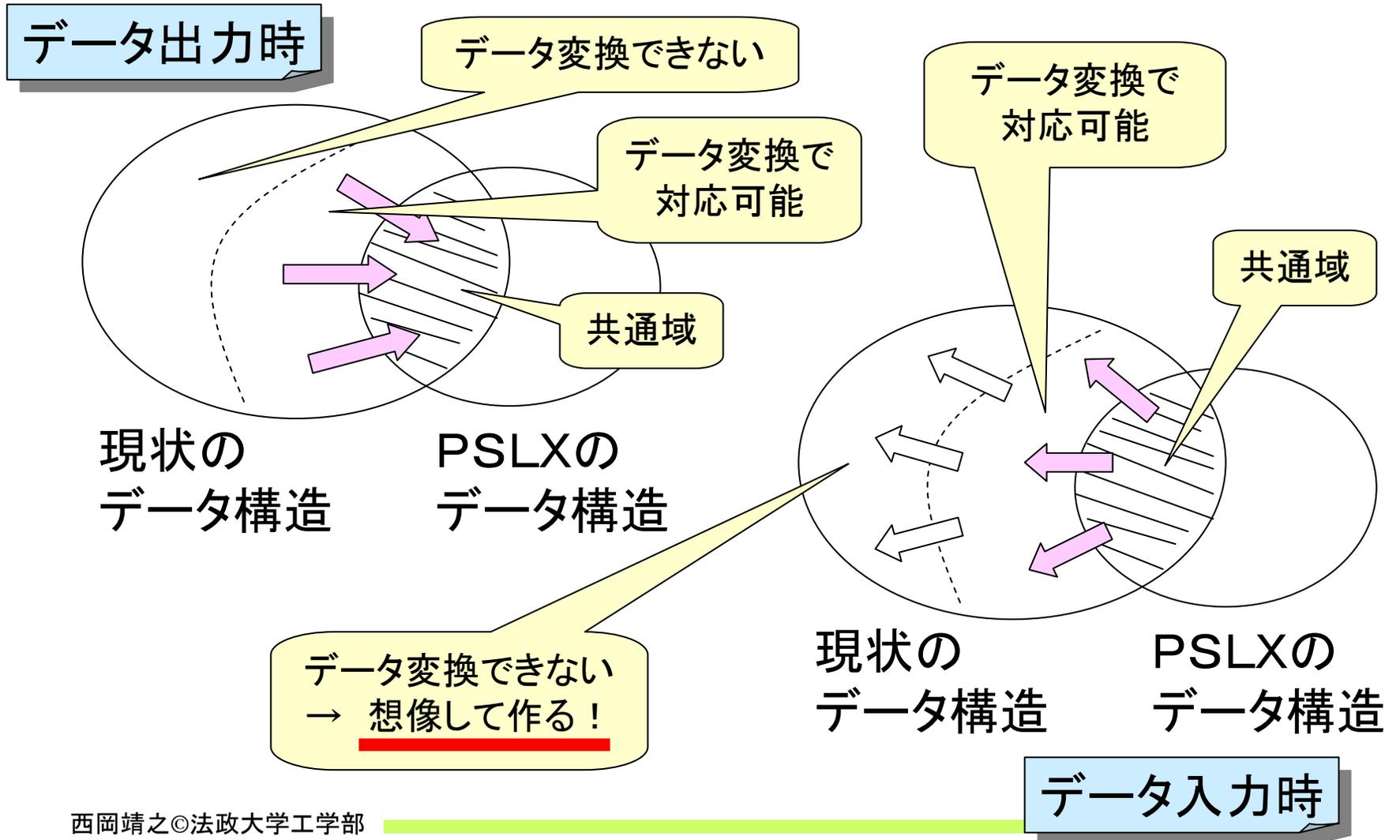
- スケジュールした内容を複数の場所で見ることができる。
- 複数の工程（拠点）の計画を連携することができる。
- MESに対して統一形式のインタフェース開発ができる。
- 上位計画との連携をダイナミックにおこなうことができる。
- スケジューラをコンポーネントとしてAPSを自由に組み立てられる。

# PSLX仕様(狭義)の内容

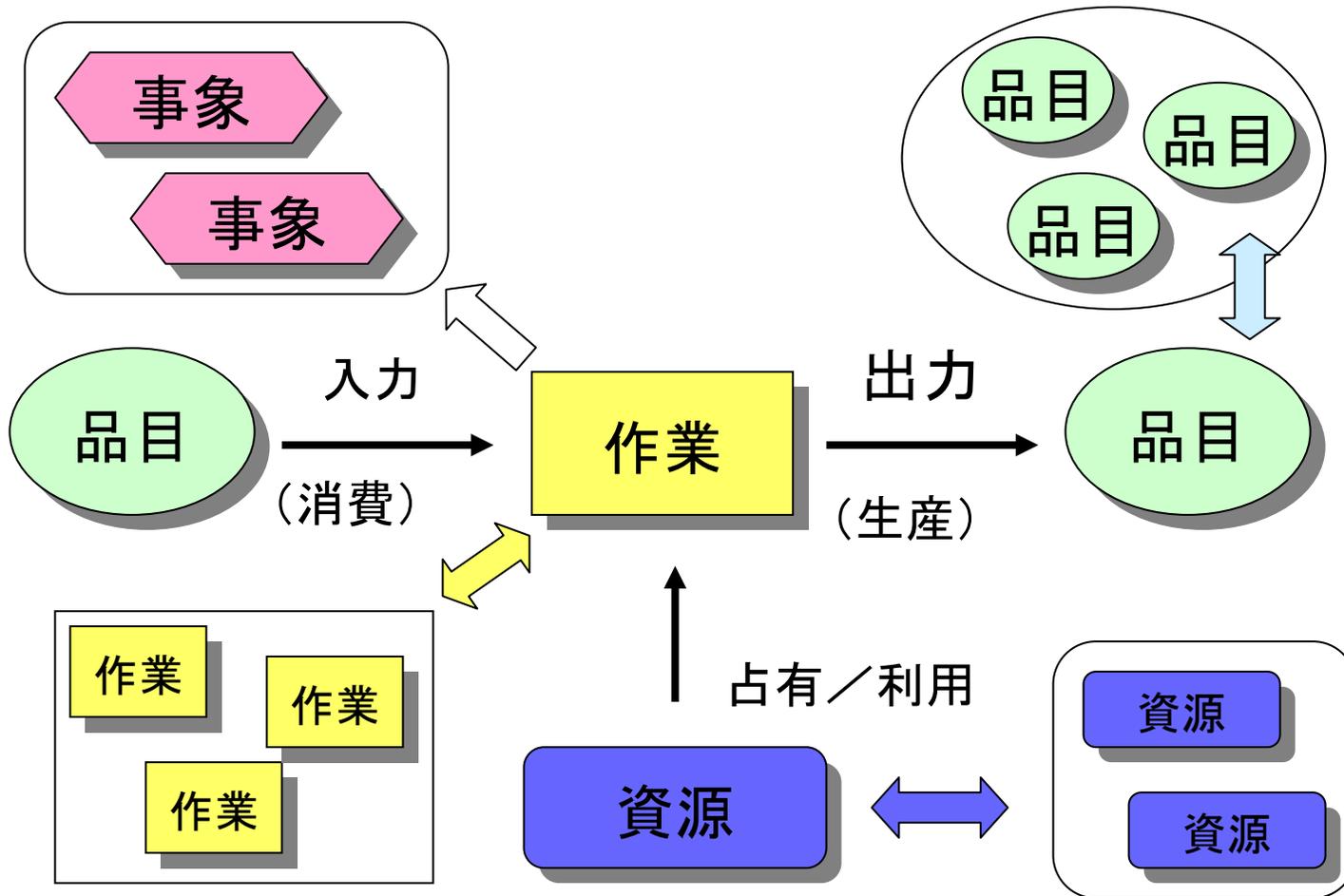
---

- 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

# 既存システムとのマッピング



# PSLXにおけるプリミティブなモデル



# プランニング上の制約

---

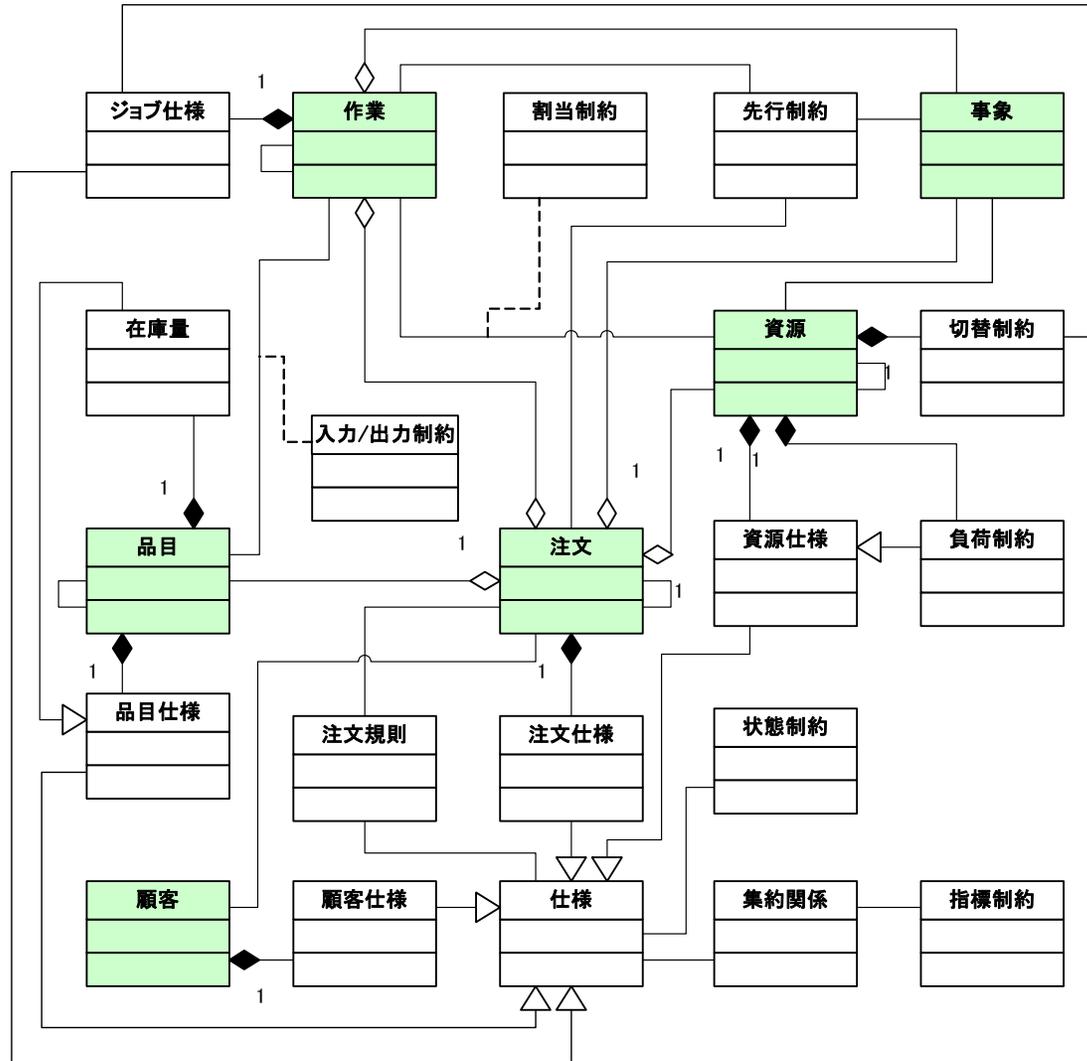
- 資源の割当制約
  - 作業と資源間の割当制約
- 入出力選択制約
  - 作業に対する入出力品目の制約
- オーダの生成／割当制約
  - 制約違反に対するオーダ充足制約
- 集計データの制約
  - 期間や項目ごとに集約されたデータの制約

# スケジューリング上の制約

---

- 先行制約
  - プロセスを構成する2つの作業間の時間的關係
- 切替制約
  - 同一資源で前後する2つの作業間の時間的關係
- 負荷制約
  - 同一時刻に行われる作業の負荷の合計の許容範囲
- 在庫制約
  - 原材料、中間品、製品それぞれの在庫量の制約
- 状態制約
  - 作業が資源を利用する場合の前提条件

# PSLXクラス構造図



# 例1: ACCROADの場合

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>  
<operation name="ライン:工程:品目">  
  <duration><calc>時間当生産能力</calc>  
  </duration>  
</operation>
```

# 例2: ASPROVAの場合

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

```
<operation 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>  
</operation>  
<item name="出力品目">  
  <parents ref="完成品目"/>  
</item>
```

# 例3: DIRECTOR5の場合

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>  
<operation name="ユニットID">  
  <parents ref="最上位ユニットID"/>  
</operation>  
<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

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>  
<operation>  
  <item ref="品目コード"/>  
  <duration>まとめ期間</duration>  
</operation>
```

# どのようなデータのやりとりがあるか

---

## マスター情報

- 品目情報
- 設備情報
- BOM情報
- 制約情報
- 評価指標情報

## トランザクション情報

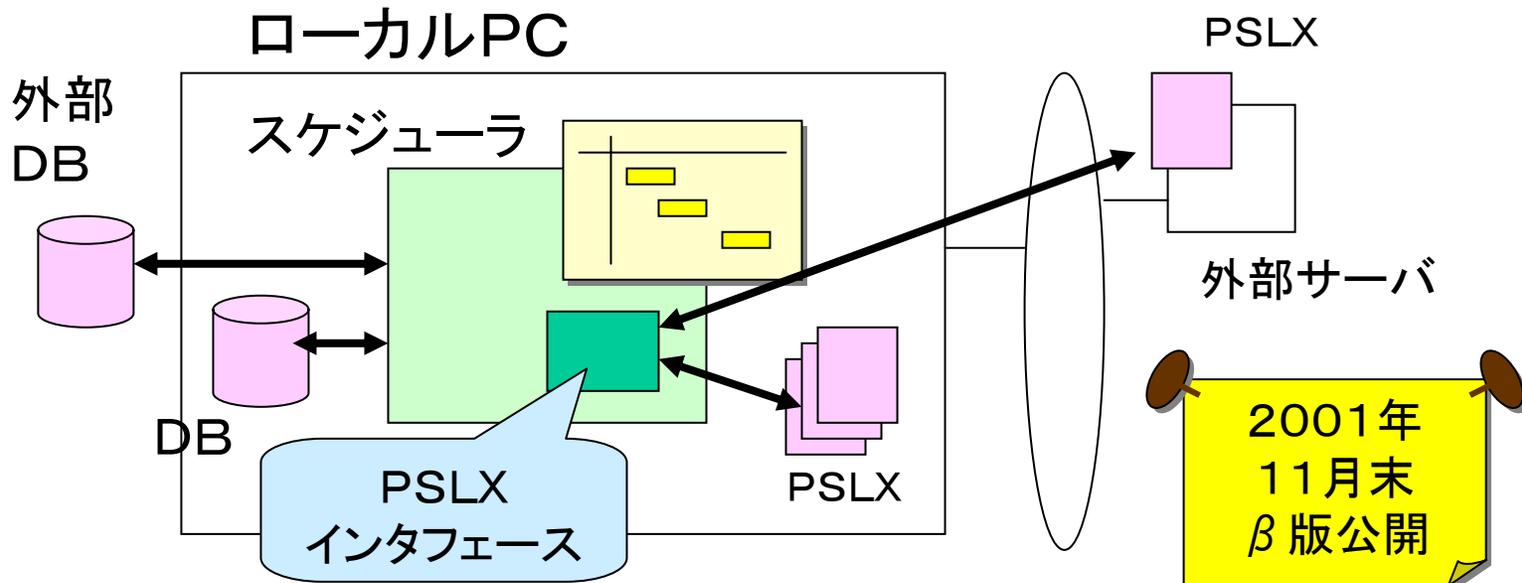
- オーダー情報
- 作業情報
- 在庫情報
- 負荷情報
- 生産座席情報
- 操業カレンダー情報

## 制御情報

- コマンドの実行
- 各種情報の問合せ
- アラーム情報
- エラー情報

# PSLXインターフェース

PSLXインターフェースは、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。



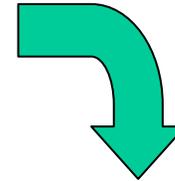
# 関数名サンプル

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

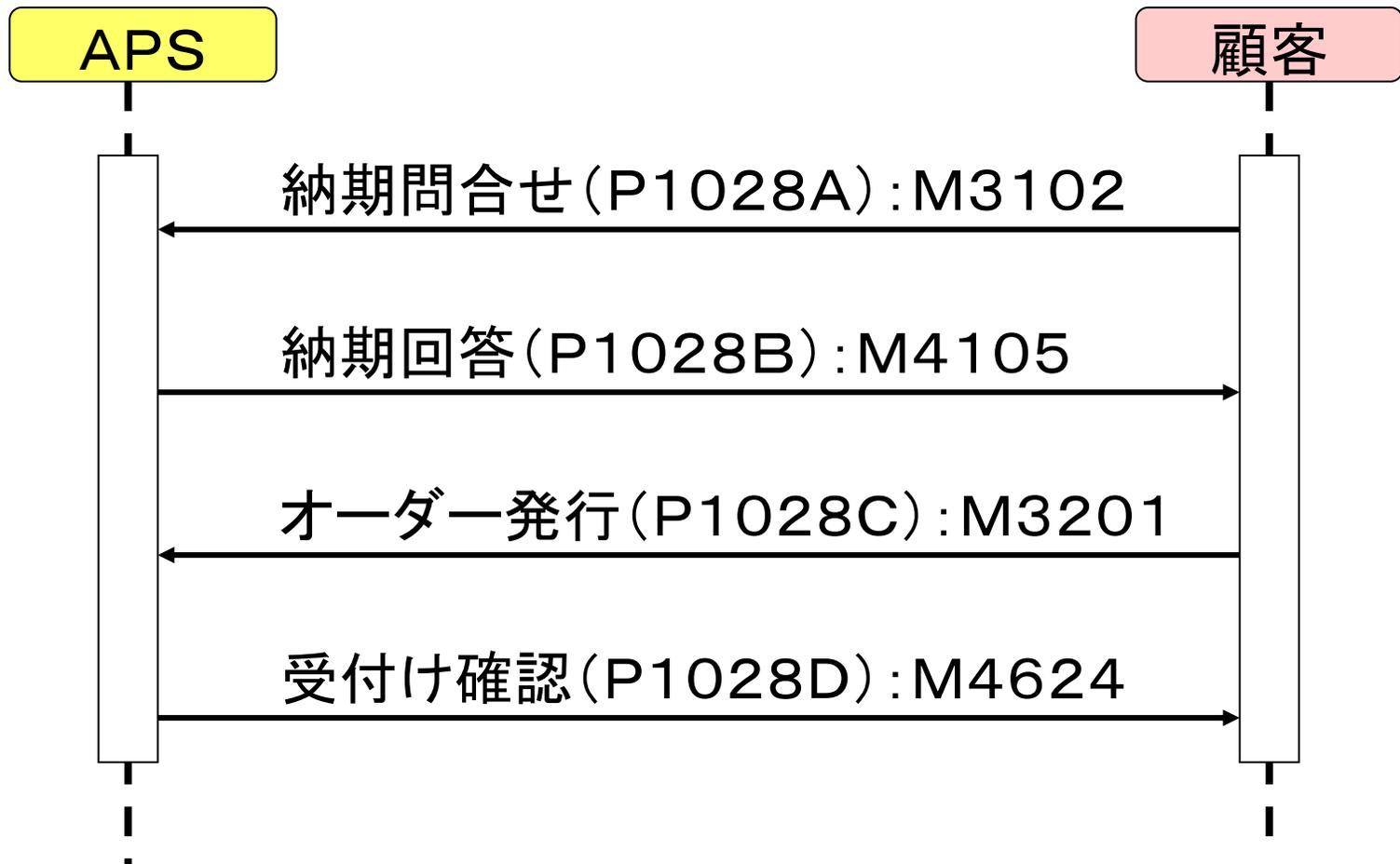


```
<?xml version="1.0" ?>  
<pslx ver="1.0">  
  <problem name="練習問題"/>  
  <order name="注文A"/>  
  <order name="注文B"/>  
</pslx>
```

サンプル.xml

# プロトコルIDとPSLXメッセージ

## オーダー受付け処理(P1028)



# SOAP+PSLXサンプル

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV=...>
```

```
<SOAP-ENV:Header>
```

```
<from>A222</from><to>C123</to>
```

```
<date>2001/11/14 12:20:00</date>
```

```
<protocol>P1028C</protocol>
```

```
<message-class>M3201</message-class>
```

プロトコルID

```
</SOAP-ENV:Header>
```

```
<SOAP-ENV:Body>
```

```
<pslx ver="1.0">
```

```
<problem name="Q0100A"/>
```

```
<order name="Z001">
```

```
<item name="PEW3TY"/><qty>20</qty>
```

```
</order>
```

```
<order>
```

```
<item name="TT9DRY"/><qty>100</qty>
```

```
</order>
```

```
</pslx>
```

PSLXデータ

```
</SOAP-ENV:Body>
```

```
</SOAP-ENV:Envelope>
```

# コンテキストによる情報の使い分け

## 計画の制約情報

```
<operation name="作業B013">  
  <start><time><earliest>2001/11/27 08:00:00</earliest></time></start>  
  <end><time><latest>2001/11/27 20:00:00</latest></time></end>  
</operation>
```

## 計画の結果情報

```
<operation name="作業B013">  
  <start><time>2001/11/27 09:10:00</time></start>  
  <end><time>2001/11/27 09:23:00</time></end>  
</operation>
```

## 製造の実績情報

```
<operation name="作業B013">  
  <start><time>2001/11/27 10:25:00</time></start>  
  <end><time>2001/11/27 10:46:00</time></end>  
</operation>
```

# PSLXによる問合せ (PSLX-Query)

## 指定名のオーダー内容の問合せ

```
<order name="X01">
```

```
  <time/> <item/>
```

```
</order>
```

空タグの情報を出力する

## 指定顧客のオーダーの問合せ

```
<order>
```

```
  <customer name="顧客A"/>
```

```
</order>
```

空タグがない場合には  
オーダー名のみ出力する

## 指定期間のオーダーの問合せ

```
<order>
```

```
  <time><earliest>2001/11/27 00:00:00</earliest>
```

```
  <latest>2001/11/28 00:00:00</latest></time>
```

```
</order>
```

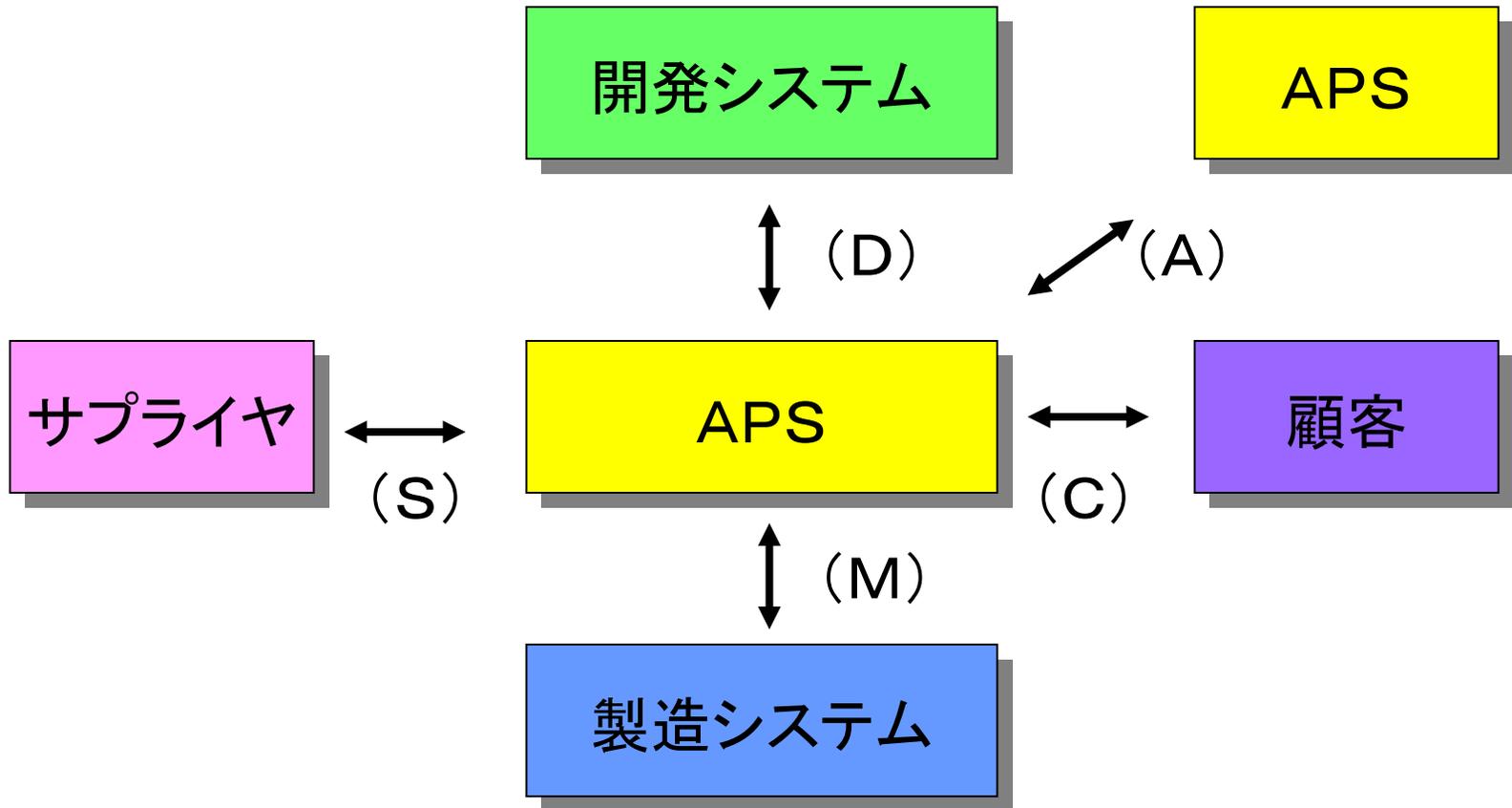
範囲指定したタグの内容は出力  
される

# システムの実装方法

---

- **WWWサーバー連携方式**
  - タイマー起動で定期的に特定のWWWサーバーにアクセスし、そこから命令を得て実行する。同時に複数のリクエストを受付けられない。比較的簡単に実装可能。
- **PSLXサーバー連携方式**
  - 専用のPSLXサーバーでリクエストを受付け、そのキューの中から順に命令を取り出して実行する。サーバーは独自プロセスなので、スケジューラOFF時も受け付けられる。
- **独自サーバー起動方式**
  - アプリケーション内部にサーバーを組み込み、直接リクエストを受け付ける。あらかじめ準備したコールバック関数に処理を追加する方式。完全にPUSH型となる。

# 相手によるプロトコルの分類



# メッセージの分類

---

01: 問合せ

02: 回答

03: オーダー入力

04: オーダー出力

05: オーダー修正入力

06: オーダー修正出力

07: オーダー削除入力

08: オーダー削除出力

09: マスタ入力

10: マスタ出力

11: マスタ修正入力

12: マスタ修正出力

13: マスタ削除入力

14: マスタ削除出力

15: トランザクション入力

16: トランザクション出力

17: 制御コマンド入力

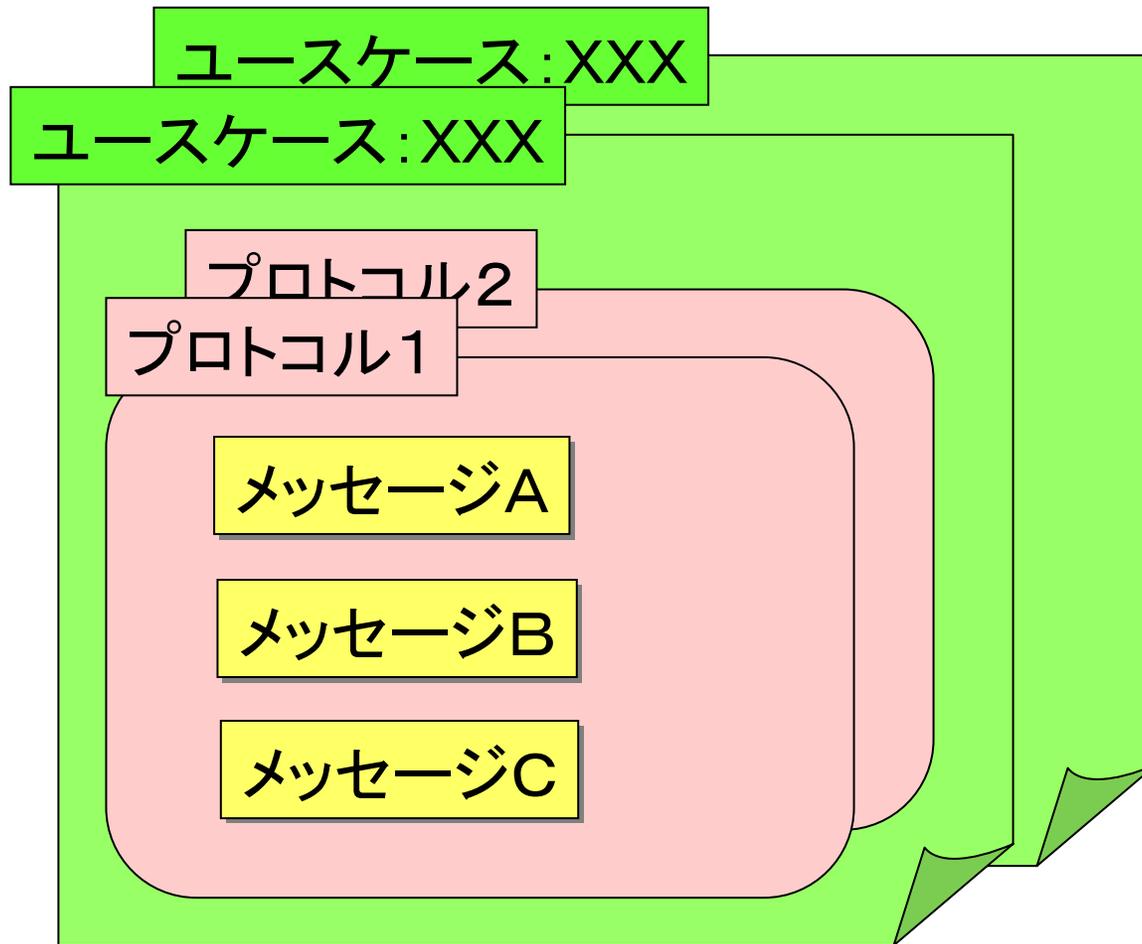
18: 制御コマンド出力

19: エラー入力

20: エラー出力

# PSLXを用いたシステムアーキテクチャ

---



# PSLX仕様書(3月公開予定)

## 第一部: 製造業のグランドデザイン(PSLX-01)

- 第1章 ビジネス環境と課題
- 第2章 APSの定義、製造業アーキテクチャー
- 第3章 対象とするビジネスモデルとシナリオ
- 第4章 核となる技術要素
- 第5章 必要とする新たなユースケース

## 第三部: XMLコアテクノロジー(PSLX-03)

- 第1章 プロトコルの定義
- 第2章 メッセージの一覧
- 第3章 PSLXの記述例
- 第4章 PSLX仕様詳細(解釈)
- 第5章 PSLXスキーマ定義

## 第二部: オブジェクトモデリング(PSLX-02)

- 第1章 APSの相対的位置付け
- 第2章 ユースケース記述
- 第3章 オブジェクトモデル
- 第4章 システムアーキテクチャ
- 第5章 インタフェース記述

## 第四部: 統一用語辞書(PSLX-00)

- 用語あいうえお順
- 用語索引

---

どうもありがとうございました