



# PSLX ホワイトペーパー

## APS の基本アーキテクチャーと システム実装技術

2004年12月

PSLX コンソーシアム

#### 著作権

本書についての著作権は、PSLX コンソーシアムにあります。本書は自由に複製および配布することができます。ただし、本書の利用者は、本書の内容に関し、改変を行うことは出来ません。

#### 免責

PSLX コンソーシアムは、本書および本書により作成した成果物の使用に関し、本書およびその成果物の利用者が被る損害については、第三者の知的所有権の侵害に対し当該第三者からの請求により利用者が被った損害を含め、一切の責任を負わないものとします。

本書により作成した成果物を稼働させる責任は、すべて成果物の作成者が負うものとし、PSLX コンソーシアムは一切の責任を負わないものとします。

#### 最新版

本書の最新版は、以下のホームページで配布しています。

<http://www.pslx.org/>

#### 不具合、要望、感想など

本書の内容に関するご要望およびご感想を、以下のアドレスまでメールでお知らせ下さい。

E-mail: [admin@pslx.org](mailto:admin@pslx.org)

技術専門委員会メンバー (および貢献者)

氏名	所属企業
伊藤 昭仁	株式会社シムトップス
古賀 敏生	株式会社ロジックスジャパン
小島 浩	新日鉄ソリューションズ株式会社
児玉 公信	株式会社エクサ
佐々木 宏明	横河電機株式会社
高橋 邦芳	アスプローバ株式会社
手島 歩三	MASP アソシエーション
中山 健	株式会社日立東日本ソリューションズ
西岡 靖之*	法政大学
前田 智彦	富士通株式会社
南口 雅也	エムツーエムインコーポレーテッド
山崎 雅史	株式会社アイザック
山田 太郎	ネクステック株式会社
松川 信也	株式会社日立製作所

\* 委員長



PSLX ホワイトペーパー (p.iv)

目次

序章	ホワイトペーパーの目的	1
第一章	APS の概念と現状技術	3
	APS が必要となる背景	3
	APS の特徴	5
	APS の現状技術	8
第二章	情報システムアーキテクチャー	11
	APS のための情報システム	11
	意思決定の特徴	14
	プランニングとスケジューリング	16
	意思決定の階層と APS	18
	既存の情報システムとの対応	21
	製造業の分類とビジネス戦略	25
第三章	業務アクティビティ	29
	業務アクティビティとは	29
	業務アクティビティモデル	30
	管理区分による分類	32
	機能区分による分類	34
	業務アクティビティ間のデータ交換	35
第四章	システム実装の基本仕様	39
	データ交換の方式	39
	PSLX オントロロジー	40
	オントロロジーの分類と構造	41
	APS ドメインモデル	46
	実装のための標準スキーマ	49
	RDB 標準スキーマ	50
	XML 標準スキーマ	53
第五章	仕様の利用形態とメリット	57
	仕様のステークホルダー	57
	ビジネスプロセスの再設計	59
	システムの特徴分類と整理	61
	業務データの整理と整合性管理	62
	実装 RDB スキーマ設計支援	64
	システムインテグレーション	66

第六章 仕様準拠と拡張方法 .....	69
標準スキーマの個別拡張方法.....	69
PSLX 仕様拡張の基本方針 .....	72
仕様の標準化プロセス .....	73
仕様準拠の判定手順.....	74
参考文献 .....	77

# 序章

## ホワイトペーパーの目的

PSLX コンソーシアムでは、2003 年に技術仕様書を勧告し、APS (先進的計画&スケジューリング)の独自の定義とその普及を行ってきた。我々が提案する APS の概念とその必要性は、ここにきて徐々に実際の製造業にも浸透し、具体的なシステムの実装事例も見られるようになった。一方で、よりシステム実装の立場からすると、従来の仕様における曖昧さや、不完全さなども徐々に明らかになり、より具体的かつ有用性の高い仕様が求められている。PSLX コンソーシアムでは、このような要望に答えるため、2004 年度より、新たな仕様の策定に着手した。

このホワイトペーパーでは、現在策定中である新しいバージョンの PSLX 技術仕様書の概要とその骨格となるアイデアを紹介する。現在の技術仕様書は、我々の活動の範囲の明確化と問題点の整理、そして具体的なソリューションのための技術的かつ基本的なスタンスを確定させることに役立った。これに対し、新バージョンの仕様書は、実際のシステム開発のための具体的な手順を提供し、開発工数を飛躍的に削減させ、同時に我々の定義する APS の最大のテーマであるアプリケーション間の相互接続性を確保するための実用的ガイドとなることを目指している。

また、同時に、最終的な APS の推進者で、かつそこから最大の恩恵を受けなければならない製造業側の立場からすると、業務や情報システムの設計を、コンサルティング企業や SI 企業に丸投げするのではなく、自らがリーダーシップをとって APS の具体的な方針と要望を提示し、開発の内容をその都度チェックするためのツールとして活用されることを目指している。なぜなら、業務の詳細は、システムのユーザである製造業が熟知しており、彼らが主役となって業務改革とシステム開発を平行的に進めることが、APS を機能させ飛躍的な効果を上げるための唯一の方法だからである。

このホワイトペーパーで紹介する仕様の内容は、まだ仕様として正式に確定したものではない。正式に確定する前に、このようにして内容を公開するのは、多くの皆様に、この仕様策定活動そのものに広く関わっていただき、皆様がすでに持っている卓越した経験と、より多くの知恵を、新バージョンの仕様書の一部として組み込みたいからである。その意味で、このホワイトペーパーは、現時点での我々の基本アイデアを紹介し、理解を深めていただくこととともに、関連するあるいは相反する多くの関係者からの意見を集約するという目的も同時にもっている。

2004 年 12 月

西岡 靖之

PSLX 技術専門委員会委員長



# 第一章

## APS の概念と現状技術

---

### APS が必要となる背景

---

はじめに、我々の提案する APS がなぜ必要なのかについて、その背景を確認する。まず、現在の製造業をとりまく要因として以下の 3 点をあげる。

( 1 ) 不確実性の増大、需要変動の激化

需要予測が当たらないのは、もはや常識となっており、需要と供給とのバランスを保つことは非常にむずかしい。これには、製品が多様化し、個々の製品がターゲットとする市場セグメントがますます狭まっていることも影響している。また、ある日突然、技術的あるいは社会的要因で、製品の販売数量が激変する場合がある。

( 2 ) 製品ライフサイクル短期化

市場は目新しさを求めるため、常に新製品をタイムリーに投入し続けな

ければ、一定の販売量を確保できない状況になりつつある。また、技術進歩により、最新の技術動向にキャッチアップした製品をタイムリーに市場に投入することが戦略的に重要となっている。製品ライフサイクルが短期化するとともに、そのサイクルを予測することも困難となっている。

### (3) サプライチェーン、EDI の変化

日本的な系列企業による安定的な取引関係が徐々に崩れ、グローバルな視野から最適調達を指向する企業が増加している。企業間の取引は固定的な従来の EDI からインターネット技術を活用したよりフレキシブルなものとなり、企業内部の計画情報等を的確に提供し共有できるかどうか、強いサプライチェーンへの参加条件となりつつある。

一方で、現在の製造業の情報システムが、これらの変化に対応する能力があるかどうかを検証してみたい。現在の製造業の基幹情報システムは、以下の3点において欠陥がある。

#### (1) 既存 ERP の機能不全

現在の製造業の多くは、1980年代の CIM の時代に汎用機をベースとした基幹システムを構築し、さらに一部の企業は過去10年間に ERP システムを導入している。いずれの場合にも、そこに実装されている計画機能は、1970年代に開発された MRP (資材所要量計算) 方式をベースとしており、これが緻密でフレキシブルな計画変更に対応できず、現実に即したアクションとの大きな乖離が起こる要因となっている。

#### (2) クライアント・サーバ方式の弊害

汎用機におけるデータベースやクライアント・サーバ方式のデータベースは、いずれもデータ構造の設計には非常に緻密さが要求され、あらかじめ設計したスキーマに適合した情報しか受け付けない。しかし、生産現場で必要とされ、計画として考慮すべき情報には例外事項が山ほどあり、実はこれらがもっとも重要な情報であることが多い。これらがすべてシステム外の人間系で処理されており、部門間で共有することができない。

#### (3) 持続的な改善と情報システムの関係

ビジネスプロセスあるいはエンジニアリングプロセスは、常に改善、改革を繰り返し、企業全体として進化し続けなければならない。一方、情報システムは、よほどの工夫をしない限り、業務の持続的改善のように頻繁には変化できない。情報システムの更新には莫大な費用と期間とリスクが伴い、業務改善、改革の足かせとなる場合すらある。また、システムの更新にともない、データの移管がスムーズに行われず、貴重な過去の資産であるデータを捨てざるを得ない場合もある。

製造業をとりまくこれらの現状と、現実の情報システムの能力の限界を踏まえ、APS の新しい概念と、Web 技術に代表されるより具体的で新しい情報処理技術を活用し、新たなソリューションを提供しようというのが、PSLX コンソーシアムの目的である。

---

## APS の特徴

---

APS とは、プランニングやスケジューリングなどの組織の意思決定の要素を統合させ、さらに各部門が組織間や企業間の枠を超えて同期をとりあいながら自律的に全体最適を志向するしくみ( PSLX 仕様書 2003 年版, PSLX-05 「PSLX 共通用語辞書」より) のことである。APS は、もともと米国にて 1990 年代に提案されたコンセプトであり、拡張された詳細スケジューリング技術によって、企業間のサプライチェーンマネジメントを含む、製造業の全体最適化を指向したものであった。当時議論されていた APS は、現在は、既存の ERP の計画エンジンとして部分的に活用されている。

PSLX が提案する APS は、従来の APS のコンセプトをさらに発展させ、ERP における計画機能全体を、そっくり置き換えるものとして位置づけている。具体的には、以下の特徴をもっている。

### (1) 概念データモデルによる情報管理

従来の情報システムは、ビジネスプロセスとデータモデルとが密接に関係し合っており、ビジネスの変化に対応して情報システムを更新する場合に、データモデルの書き換え作業が大きなネックとなる。企業に蓄積

されたさまざまな情報は、企業の資産であり、ビジネスプロセスが変わっても継承される。PSLX が提唱する APS は、その背後に概念データモデルを持ち、データ中心の概念のもと、価値のある情報あるいは知識を中心に据えたシステム構築を行う。

#### ( 2 ) ビジネスモデル変更に対する拡張性

現在の製造業が直面している不確実性の極めて高い市場環境において、製品の良し悪しのみではなく、ビジネスモデルの優位性が企業の優劣を決定する大きな要因となっている。そして、ビジネスモデルの根幹をなすのは、企業の組織的な意思決定のしくみである。APS は、個々のビジネスモデルを具現化するための基幹システムとなる。計画とスケジューリングという意思決定を中心とした APS は、基幹システムとして企業が独自に設計し拡張することができる。

#### ( 3 ) 設計チームと製造チームの密な連携

製品のライフサイクルが短くなり、新規製品リリースや設計変更が頻繁になるにつれ、生産現場でも、常に新しい部品、新しい製造手順、新しいラインの設置などの非定常的な作業がますます増える。これは、計画やスケジューリングにとっては、意思決定のためのマスタ情報が常に変更になることを意味しており、データの管理が極めて煩雑となる。APS では、BOM 情報や製造手順書などを、製造部門と設計部門とで連携して管理し、正式な設計変更情報以外の非定型な情報の多くも情報システムの一部として管理できる。

#### ( 4 ) リアルタイムアカウンティングと KPI

生産現場のさまざまな作業は、すべてコストと直結しており、それらが企業全体の収益にどれだけ貢献しているかを常に管理することは、非常に重要であると同時に、非常に困難な課題の一つでもある。APS では、コスト情報を、生産現場レベルの個々の作業のスケジュールおよびその実績として把握し、それを企業の戦略的意思決定において意味のある形に変換するしくみを持っている。APS は、企業が独自にデザインした KPI ( 戦略的経営指標 ) に対し、リアルタイムにそのための基礎情報を提供することができる。

#### ( 5 ) 実行可能で信頼できるマスタースケジュール

従来の多くの ERP が扱う MPS (マスタースケジュール) は、計画の粒度が粗く、さらに現実の工場の実情を反映したものではなかった。これに対して、APS では、マスタースケジュールを、その製造業における計画とスケジューリングの中核として位置づけ、すべてのアクションをこれに同期させる。特に、製造現場の詳細スケジューリングと、マスタースケジュールとの双方向で密接な連携によってこれが可能となる。

#### (6) 詳細スケジュールベースの企業間連携

サプライチェーンマネジメントの対象は、中長期的な計画のレベルでの企業間連携と、日々のオペレーションの同期化という極めて短期的な詳細スケジューリングのレベルでの企業間連携がある。詳細スケジューリングレベルの連携は、企業内でも確実に実現することが難しく、更に、企業間でこれを情報システムとして実現している例は非常に少ない。多くのサプライチェーンのための情報システムは、計画レベルの粒度の粗い情報交換である。PSLX が提案する APS は、詳細スケジューリングと企業の基幹システムが連携し、必要に応じて企業間で詳細スケジュール情報を共有することで、このような連携が可能となる。

#### (7) 人間中心の自「働」化のための支援

トヨタ生産方式(リーン生産方式)の特徴の一つに、人間中心の自「働」化がある。これは、自ら考え自ら働き、問題点を主体的に解決するしくみである。情報システムにおける自「働」化とは、計算機からの出力に人間が合わせるのではなく、人間を含む工場全体や個々の生産現場の流れを目で見えるようにし、問題点の発見を容易にし、そして臨機応変にシステムそのものを改善することである。APS における意思決定のプロセスやロジックは、情報システム内部で固定化されることなく、常に人間の創意工夫をベースに継続的に進化する。

#### (8) 製造業(ユーザ)主体のシステム化

現在、情報システムは、製造業のビジネスモデルを実現するための骨格であり、製造業の競争的優位性の源泉でもある。これは、情報システムを提供する SI 企業や IT パッケージベンダーなしには実現しえないが、あくまでも主体となってシステム構築を行うのは製造業の側である。今までは、情報システムプロバイダーが提案する情報システムは、その内容や機能を詳細に理解することが難しく、最後になって意図しなかった

システムができあがる場合も多々あった。PSLX で提案する APS は、情報システムアーキテクチャーが明確であり、製造業の側は、独自のビジネスに特有の機能をそこに付加することに専念することができる。

---

## APS の現状技術

---

すでに、いくつかの製造業の情報システムでは、APS に関連した多くの技術が開発され利用されている。ここで、現時点での APS 関連技術について再確認する。以下に、主要なものを列挙し、簡単な解説を行う。

### (1) 作業中心 BOM データ管理

従来の BOM は、親品目と子品目との関係構造を記述する。これは、MRP (資材所要量計算) において、親品目である製品の必要数量から、子品目である部品や資材の必要数量を計算するために利用される。一方、親品目を生産するための作業は、別途ルーティング表で記述され、別に管理されている。これは、資源の負荷計算やスケジューリングに利用される。APS では、これらの独立した 2 つの情報を作業をキーに関係づけ、親品目と子品目の間に作業を含む形式の BOM を持つ。

### (2) 生産現場の詳細なモデリング

生産現場で実際利用する実行スケジュールを作成するためには、それぞれの現場固有の制約をスケジューリングシステムに設定する必要がある。従来の詳細スケジューリングでは、先行制約、資源制約など、非常に単純化されたものしか存在しなかった。APS では、これらに加え、資材や部品制約、品目切替制約、作業やツールなどの副資源制約、同時実行制約など、生産現場のさまざまな現実的な制約を設定することが可能である。

### (3) 有限能力 & 有限資材スケジューリング

従来の有限能力スケジューリングが、設備能力のみを考慮してスケジュールを作成していたのに対し、APS のスケジューリングロジックの最大の特徴は、資材在庫の制約も考慮している点といえる。つまり、有限能力 & 有限資材スケジューリングである。APS が行うこのスケジューリン

グロジックでは、作業をある時間帯に配置するにあたり、必要設備の能力をチェックすると同時に、必要資材の有無をチェックする。

#### (4) ボトルネック指向スケジューリング

特定の設備がボトルネックとなって工場全体の生産性を大きく左右するような場合、APS では、その設備の稼働率を最大にするスケジュールを作成することができる。具体的には、まずボトルネック設備のみについて、その稼働率が最大となるスケジュールを設定する。ボトルネックより上流工程では、タイムバッファを設定した上でバックワードスケジューリングを行い、下流工程ではフォワードスケジューリングを行う。

#### (5) MPS 詳細シミュレーション

MPS (マスタースケジュール) は、販売部門と製造部門との連携のための重要な情報である。APS では、MPS として設定された製品ごとの基準日程が、実際の生産現場の状況を考慮した上で、現実的に実現可能かどうかを、詳細スケジューリング機能を用いて事前に検証する。ここでは、詳細スケジューリングをシミュレーターとして用いている。また、必要に応じて、そこでの結果をそのまま、実行スケジュールとして共有することもできる。

#### (6) ダイナミック・フルペギング

従来の MRP では、個々の生産現場の個々の生産ロットが、最終的にどの顧客オーダーに対応するかを確認することができない。一方、製番方式では、個々の生産ロットが最終オーダーに対応しているが、ロットまとめなど効率的な生産には不向きである。APS のダイナミック・フルペギングでは、両者の特徴をとりこむことで、生産現場の各ロットや生産指示が、最終オーダーと常に対応づけて管理でき、必要に応じてその対応関係を付け替えることが可能である。

#### (7) メタヒューリスティックによる最適化

製造業における全体最適を指向した計画を作成するために、APS では、GA (遺伝的アルゴリズム) やタブーサーチなどのメタヒューリスティックによる最適化計算を行う。これらの最適化ロジックは、製造業の計画やスケジューリング問題のように、非常に複雑で厳密な最適解が見つからないような状況でも、比較的短い時間で、厳密な最適解ではないが、

人間が独自に行う計算よりははるかに勝る解を提示してくれる。



## 第二章

# 情報システムアーキテクチャー

---

### APS のための情報システム

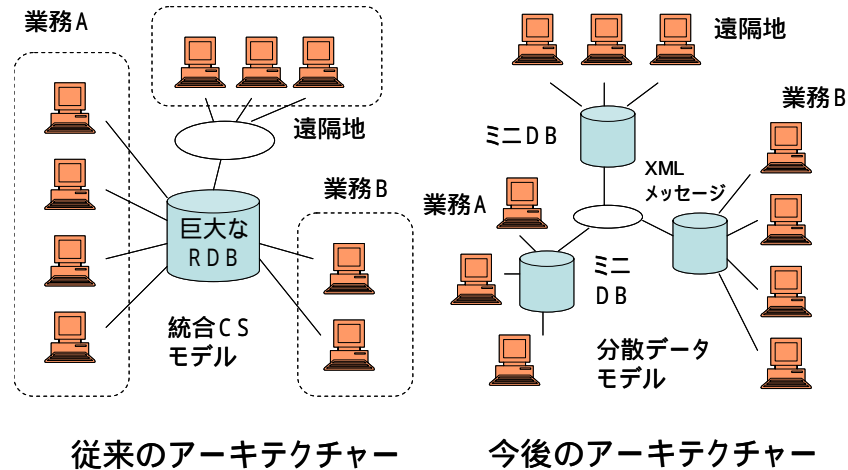
---

前章で、現在の情報システムとしての主流であるクライアント・サーバ方式の弊害について述べた。事前に設計されたデータスキーマに適合しないデータはすべて欠落し、集中管理であるので、スキーマの変更は多大な影響をあたえるため非常に困難な作業となっているのだ。個々の生産現場の実情にあったきめ細かな情報システムとするためには、そこで必要となるデータの受け皿が必要であり、さらにそれは、常に改善、改良を繰り返し持続的な進化が可能なものでなければならない。

PSLX が提案する APS のための情報システムアーキテクチャーは、従来のクライアント・サーバ方式の利点を生かしつつ、さらに個別の業務単位に独自のローカルな設計を可能とする自律分散型のしくみである。具体的には、図 1 にあるように、ミニ DB を企業内部に多数配置し、ローカルに必要なデータおよび業務処理はローカルで実行し、必要に応じて

個々の情報システム間で連携をとるといふしくみである。個々のデータは、基本的にそのデータが生成した場所で保管され管理される。

図1  
情報システム  
の基本形態



ただし、この場合、当然のことであるが、各独立した個々の情報システム間の処理の連携や、データの整合性が問題となる。独自に設計された2つの異なるデータスキーマ間で情報交換を行うためには、それらのデータの本来の意味のレベルまで立ち返って議論しなければならない。このために重要となるのが概念データモデルである。概念データモデルとは、情報システムの実装形態に依存せずに企業活動（ビジネスプロセス）そのものを純粋に表現するために必要な情報モデルである。これは、ビジネスプロセスの各要素と関係、そして各要素に対する入力、出力、資源、あるいはコントロールなどを表現するために必要となる。

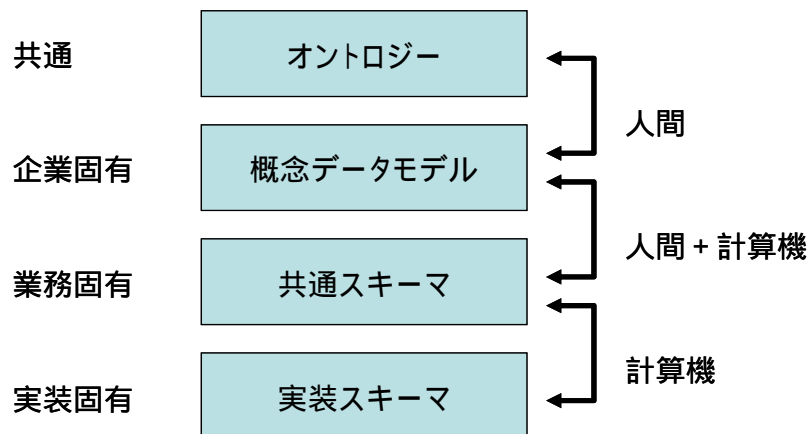
概念データモデルの具体的な例として、後ほど4章で説明するAPSドメインモデルがある。実際には、これをそれぞれの企業の実情に合わせて部分修正したものが、その企業に対する企業固有の概念データモデルとなる。概念データモデルは、企業固有のビジネスの実体を表すものとして、少なくとも企業の数だけ個別に存在し、それらの企業の業務が変わるにつれてその内容も変化していく。なお、概念データモデルは、あくまでデータおよびその構造を表現したものであり、ロジックや処理手順に関する情報は含まれない。

PSLXの概念データモデルは、その背後にPSLXオントロジーを持つ。オントロジーとは、さまざまなデータの意味を規定するための基本的な

概念とその構造の記述である。企業ごとに異なる概念データモデルは、APS ドメインモデルをベースとして各企業で独自に修正可能であり、また、APS ドメインモデルを参照せずに概念データモデルを構築することも可能である。このような場合に、PSLX オントロジーは、それぞれの個々の概念データモデルの解釈の枠組みを提供する。オントロジーの存在があって、はじめて「データ」を「情報」として活用することが可能となる。

実際にそれぞれの企業において情報システムを構築する場合には、概念データモデルを、さらに個々の情報システムの実装形態に合わせて最適化された実装スキーマが必要となる。実装 RDB スキーマや、プログラムの内部のデータ構造などがこれに相当する。しかし、図 1 で示したような、異なるシステム間での連携が必要となるような場合には、概念データモデルから、ダイレクトに実装スキーマを生成してはならない。なぜなら、概念データモデルと実装スキーマ間のデータ変換は、基本的に人間が介在した処理が必要となり、システム間のデータ交換のたびに業務担当者が変換作業（伝票の転記など）を行わなければならないからである。

図 2  
情報連携のためのレイヤ構造



異なるシステム間での情報連携を実現させるためには、見知らぬ連携相手との間で、計算機が理解できる形式の共通のデータスキーマが必要となる。これを、共通スキーマと呼ぶ。図 2 は、情報連携を実現させるために最低限必要となるデータモデルの階層関係を示している。図 2 において、共通スキーマとは、異なる計算機プログラムが、基本的に人間系での処理を介在させずに情報交換を行うための階層であり、業務ごとに

用意された共通のフレームである。共通スキーマは、各業務での実装スキーマよりも前に定義されていなければならない。そして、各実装を担当するシステム設計者は、この共通スキーマを、概念データモデルあるいはオントロジーによって正しく解釈したうえで計算機内に取り込み、異なる実装システム間での情報交換が可能となるようにする。

---

## 意思決定の特徴

---

人間が行う意思決定は非常に高度で広範囲である。ここでは、意思決定全般というよりは、より具体的に、企業の業務としての具体的なアクションをとるために必要な計画やスケジュールに関連するさまざまな情報を決定すること、と定義する。計画やスケジュールを直接生成する行為のみでなく、そのために必要となる情報の生成や管理といった間接業務も含む。それでは、まず、このような定義を前提として、APS を実現するために意思決定が持つべき要件について以下にまとめる。

- (1) 正しい情報、タイムリーな情報にもとづいた意思決定でなければならない。
- (2) 最終的な意思決定は現場で行えるしくみでなければならない。
- (3) 意思決定にもとづく行動はフォローし、フィードバックしなければならない。
- (4) 意思決定結果は共有し協調するためのものでなければならない。
- (5) 意思決定のしくみはつねにシンプルでビジブルでなければならない。

PSLX では、意思決定を、具体的なアクションに関する計画あるいはスケジュールリングを中心として考える。ここでは、具体的なアクションは、あらかじめ明らかである場合と、そのアクションの内容そのものを構成するパラメータも意思決定の対象となっている場合がある。APS を意思決定システムとして認識し、それを独自にデザインするためには、ある程度この意思決定の特徴に関する知識があることが望ましい。企業に存在するさまざまな意思決定は、以下のような視点から特徴づけることができる。

### (1) フィードバックとフィードフォワード

具体的なアクションに関する意思決定として、最も一般的なものがフィードバック方式である。これは、過去に行った意思決定で目指していた状況と、意思決定の内容を実行して得られた状況を比較し、そこから得られた情報を次ぎの意思決定の内容に反映させるというものである。自分の意思決定の結果を、将来の意思決定で再利用するという意味で、フィードバック方式と呼ばれる。これは、より物理的な制御の世界で広く用いられている。

一方、未来の状況が比較的予測可能である場合には、フィードフォワード方式が有効となる。フィードフォワード方式は、計画あるいはスケジューリングのしくみをもち、将来の一定期間にわたってのアクションや予測される事象をあらかじめ設定し、さまざまな制約や因果関係から起こりうる状況を事前に予測する。フィードバック方式では、変化が起こった後の対応となるために、その間のロスが発生することになるが、フィードフォワード方式では、あらかじめ変化を予測し、目標と現実とのギャップを最小にすることが可能となる。

### (2) 新規計画と再計画

将来の一定期間の業務におけるアクションを計画する場合、それらは新規計画と再計画の2つのタイプに分類できる。新規計画では、対象期間のアクションに関する意思決定を、ゼロの状態から行う。一方、いったん決定した計画は、それが将来のアクションに関するものであるため、時間の経過とともにより新しい状況が明らかになり、また想定していた状況とは異なる状況となった場合に、修正される場合がある。これを再計画と呼ぶ。

現実の業務では、常に現在から将来にわたってのアクションが設定されており、それらが常に最新の状況に応じて更新されている。つまり、実際の多くの意思決定は、新規計画ではなく再計画に属する。形式的には新規計画であるが、実際にはインフォーマルに設定していた計画に対する再計画として位置づけられる場合も多い。

再計画の場合、計画を行う対象期間に、過去に作成した計画の一部が登

場する。これらの過去（前回）の計画の要素には、現実的に変更することができないもの、変更しないことが望ましいもの、そして変更が自由にできるものなどが混在している。一般に、現在時刻に近いほど変更が困難であり、将来にいくほど変更が容易となる。

### （３）計画期間と計画サイクル

企業が行う意思決定をビジネスモデルとして記述する場合に、非常に重要なパラメータとして計画期間（ホライゾン）と計画サイクルがある。計画期間とは、計画の対象となる期間であり、その期間に含まれるアクションが意思決定の対象となる。計画期間は、現在時刻からみた相対的な開始時刻と終了時刻（または期間）によって定義できる。たとえば、翌週の月ようから一週間、といった表現となる。

また、計画サイクルとは、新規計画または再計画を行う頻度である。計画サイクルは固定である場合と、必要に応じて行われる不定期の場合と、両者の組み合わせの場合がある。固定サイクルの場合には、その周期および具体的な暦上でのポイントを指定する。たとえば、毎週水曜、毎月25日、などである。一方、計画サイクルが不定期の場合には、計画を行う条件を記述する。たとえば、特別な量の注文がきた場合、在庫量が一定量を下回った場合などがあげられる。

---

## プランニングとスケジューリング

---

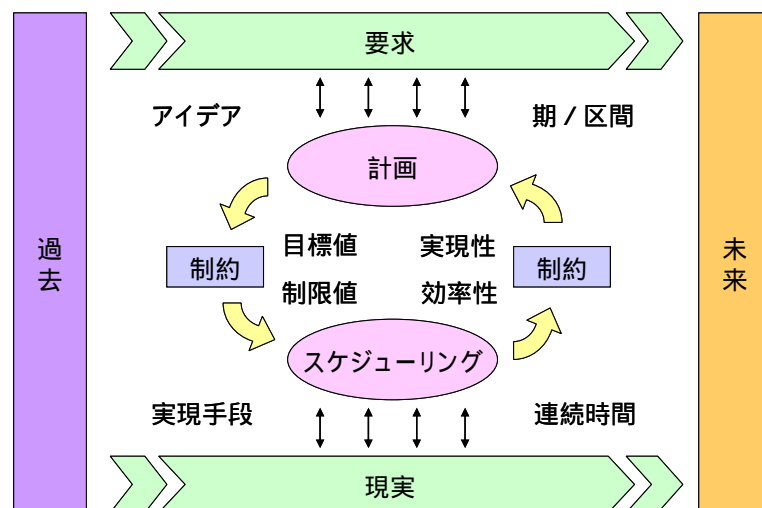
ここまで、計画とスケジューリングという2つの用語を、特にその意思決定の特徴の違いについての明確な定義なしに用いてきた。ここで、計画（プランニング）とスケジューリングの違いを定義する。まず、一般的な概念として、計画という用語は、ある目的を達成するための手段を明らかにし、その実行可能性を確保することであるといえる。このために、計画では、さまざまなパラメータを決定する。

これに対して、一般に、スケジューリングという用語は、具体的なアクションを、具体的な資源のもとで実行するための具体的な手順に関する意思決定である。ここでは、さまざまな物理的な制約が関与し、実際に

存在する各資源に対して，実時間軸上で，それらのアクションに関係するさまざまなパラメータを決定する．

意思決定の構造として見た場合，計画（プランニング）は常にスケジューリングの上位に位置づけられる．つまり，階層関係でいえば，計画の結果を受けてスケジューリングが実行される．スケジューリングにとっての制約や目標がプランニングによって設定される場合が多い．また，一方で，計画は，スケジューリングによって実行可能性が確認できた時点ではじめて，最終的に有効な計画案を得ることができる．この意味では，計画とスケジューリングはお互いに依存関係にある．図3に計画とスケジューリングの関係を示す．

図3  
計画とスケジューリングの  
関係



計画とスケジューリングの違いは，より技術的に言うと，主に扱うデータモデルの種類からくる．計画とスケジューリングとは，意思決定パラメータとして扱うデータモデルが異なり，それらが時間の概念とどのように関わっているかが，ここでは非常に重要となる．

まず，計画において，その中心的な意思決定パラメータは，一定の期間における何らかの値となる．たとえば，今月の生産数量，来月の部門売り上げ，来週の残業時間，などである．ここで重要なことは，計画では時間そのものを決定するのではなく，ある期間に対応する値を決定することが中心的な作業となるという点である．

これに対して、スケジューリングでは、その意思決定パラメータは、さまざまなアクションを実行する時刻そのものが中心となる。ある加工作業の開始時刻、あるいは終了時刻、倉庫からの出庫時刻、出荷時刻、などである。複数の作業の順序もスケジューリングの重要なテーマであるが、これは、相対的な時刻の決定であるといえる。

計画とスケジューリングの関係としてまず指摘できるのは、一つの計画に対応する複数のスケジュールが存在し得るのに対し、一つのスケジュールに対応する計画はただ一つである、という非対称性である。スケジューリングのほうが解のパラエティは爆発的に多い。一般に、計画を最初に作った後にスケジューリングを行うのは、この性質が大きく関係している。

また、計画における意思決定パラメータがもつ時間幅が小さくなるほど、内容は緻密になり、スケジューリング問題に近い精密な解を導くことができる。一方で、パラメータの時間幅が大きいと、よりラフで全体を見通した計画が策定できる。つまり、計画における単位期間の決定は、計画とスケジューリングをより効果的に連携させための大きなキーとなっているという点も指摘できる。

---

## 意思決定の階層と APS

---

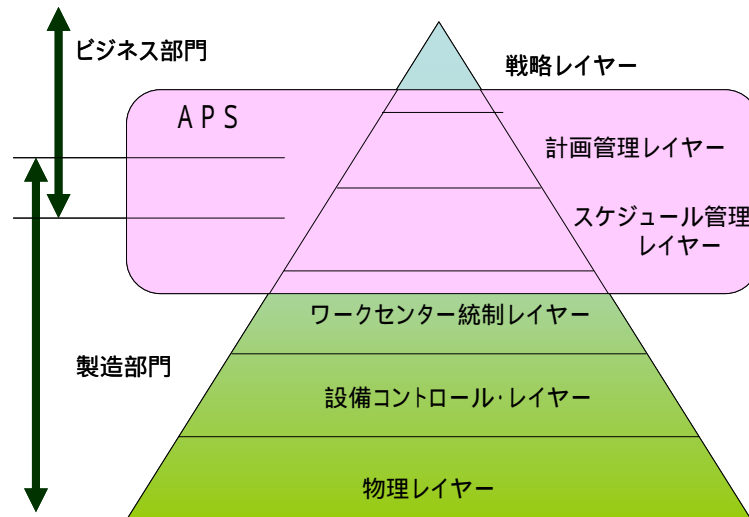
実際の製造業の意思決定のモデルとして、APS がどのように位置づけられ、具体的にどのような形で実装されるかについて説明する。まず、企業全体の視点から、そこに存在するあらゆる意思決定について、意思決定の6つの階層を導入して整理する。図4に、意思決定のレイヤとして、物理レイヤ、設備コントロール・レイヤ、ワークセンター統制レイヤ、スケジュール管理レイヤ、計画管理レイヤ、そして戦略レイヤを示した。

このうち、APS は、スケジュール管理レイヤ、計画管理レイヤ、そしてそれら2つのレイヤと接する上下のレイヤの一部が対応する。この部分は、図4でも示しているように、製造部門とビジネス部門がオーバーラップしている領域であり、さらに、下位のスケジュール管理レイヤの部分



は、場合によっては異なる工場または異なるエリアに分散されて配置されている場合もある。

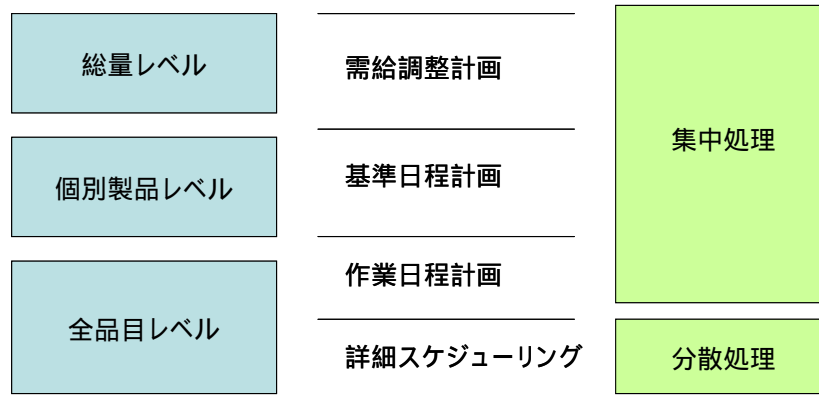
図4  
企業における  
意思決定のレ  
イヤ構造



APS が担当する意思決定の領域をさらに別の視点から分類すると図5のようになる。ここで左側の3つのレベルは、意思決定が対象とする情報の粒度の違いを表している。まず、最上位の総量レベルでは、さまざまな種類の製品を、そのグループやカテゴリごとにまとめた上で、それらのグループまたはカテゴリごとの総量が意思決定の対象となる。これに対して、個別製品レベルでは、異なる製品をまとめることは行わず、それぞれの品目ごとの情報が意思決定の対象となる。そして、最下位の全品目レベルは、製品のみでなく、その製品の構成要素となっている半製品、部品、資材などすべてが意思決定の対象となる。

図5の右側にある視点は、意思決定の形式が中央集中的であるか、自律分散的であるかという違いからくる。一般的な製造業では、資材所要量計算(MRP)として、全品目レベルまでを集中処理で行っている場合が多いため、図5のような形態となっている。これはあくまで例であり、必要条件ではない。一般的には、全品目レベルにおいて、生産現場ごとにさらに詳細な制約情報や、稼動状況などを加味した意思決定が行われる。

図5  
APS 内の意思  
決定の階層構  
造



以上の2つの視点にしたがって APS における意思決定を階層化すると、以下のような4種類の意思決定問題が存在することがわかる。これらの各意思決定は、通常は個別に行われるが、統合的に解決することも可能である。

#### (1) 需給調整計画

需給調整計画では、製品ファミリーのレベルでの生産数量の決定を行う。また、同時に、生産のための資源に関しては、工場全体あるいは特定のエリア単位で、その能力を決定する。これらの意思決定は、比較的中長期的な計画期間に対して行う。生産のための資源については、必要に応じて能力の補強も可能である。この意思決定は、財務的な視点からの投資計画と密接に関係しており、企業の全体的な収益の視点からの最適化がなされる。

#### (2) 基準日程計画

基準日程計画では、具体的な最終製品ごとに生産数量が決定される。計画の対象期間は短期から中期となる。具体的な最終製品の数量は、実際に受注した顧客オーダーや、需要予測に基づく見込みオーダーなどの情報をもとに設定される。生産のための資源に関しては、需給調整計画と同様に工場全体や特定のエリアとなるが、能力値は意思決定パラメータではなく制約となる。製造部門と販売部門がここで設定される具体的なスケジュールのもとで合意し、同時に、資源や資材について有限キャパシティーのもとでの実行可能性がチェックされる。

### (3) 作業日程計画

作業日程計画では、基準日程計画にて提示されている最終製品の生産量と完成日時に対し、それを実際に生産するために必要となる作業を確定し、それらの作業が、利用可能な有限の資源がもつ時間軸上に割当てられる。ここでは、意思決定の中心的な議論が作業となり、作業を介して必要な資源の工数や、中間組立品、部品、そして資材などの数量の要求が決定される。これは、いわゆる MRP (資材所要量計算) や CRP (能力所要量計算) の考え方を含む。

### (4) 詳細スケジューリング

最後に、詳細スケジューリングは、基本的な考え方は作業日程計画と同様であるが、ここでは、各生産現場の個別の制約や要望を組み込んだ意思決定となっている点、あるいは、意思決定のアウトプットであるスケジュールが、作業日程計画と比較してデータの粒度がより細かく、原則として業務の各アクションに直接対応したレベルとなっている点などが特徴となる。詳細スケジューリングのアウトプットは、直近になって、作業指示として現場の作業者に送られる。

---

## 既存の情報システムとの対応

---

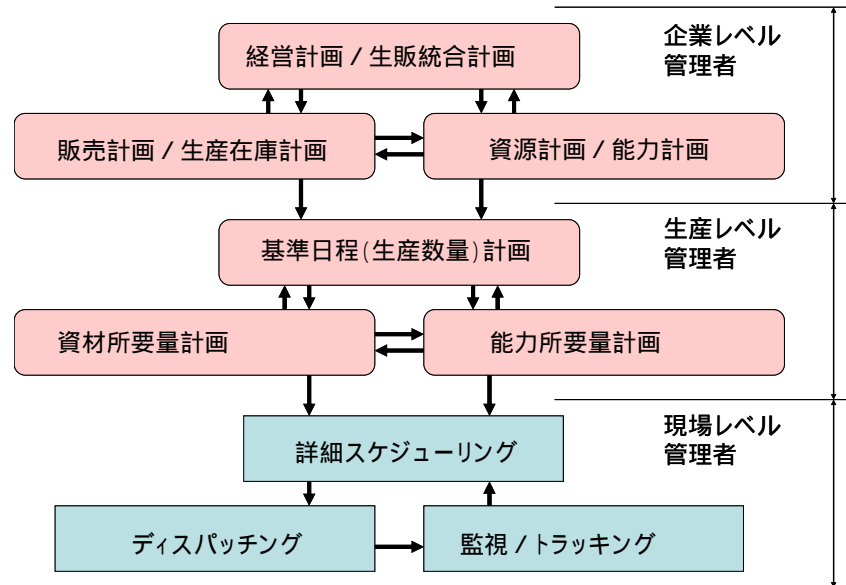
すでに多くの製造業では、その基幹システムとして、MRP II や ERP が実装され稼働している。これらレガシーシステムと、PSLX が定義する APS との関係を以下に確認する。まず、図 6 に、典型的な製造業における基幹情報システムの構成を示す。

図 6 では、まず全体を、企業レベル管理者、生産レベル管理者、そして現場レベル管理者の 3 つの階層に分けている。これは、各意思決定において、意思決定に関与する人間の視点あるいは目的がそれぞれ異なる点を強調するものである。企業レベルの管理者は、企業全体の収益の視点から意思決定を行い、生産レベルの管理者は、稼働率、リードタイム、在庫量などの工場全体のパラメータに注目しながら、最終的には工場全体のスループットを最大化する。一方、現場レベルの管理者は、上位レベルで作成した計画どおりには現実には動かないことを知っており、より

安全で確実に効率的な生産の方法を常に模索する。MES（製造実行システム）はこのレベルが対象となる。

以下に、個々の意思決定モジュールについて簡単に説明する。

図6  
製造業における既存の情報システム



(1) 経営計画 / 製販統合計画

企業の役員レベルが参画し、より戦略的あるいは将来の収益を直接左右する意思決定を行う。具体的には、製品開発計画、工場のライン増設、人員能力増減、サプライヤーとのアライアンスなどがある。これらはすべて、より下位の意思決定の前提条件となる。

(2) 販売計画 / 生産在庫計画

需要と供給のバランスをとる上で、ここでは需要側に関する意思決定を行う。顧客からの受注を単に予測するのではなく、需要を作り出すためのアクションや具体的な数字が議論される。また、将来的な需要変動に対応した製品在庫数の決定や、在庫移管計画なども含まれる。

(3) 資源計画 / 能力計画

需要と供給のバランスをとる上で、供給側の最上位の意思決定を行う。ここでは、工場全体の能力値を、中長期的な視点から見直し、必要能力の増強や、不要な設備の廃棄などを行う。また、作業者による必要な労

働力の確保も、ここでの重要な項目である。

#### (4) 基準日程計画

ここでは、最終製品レベルで MPS(マスタースケジュール)を作成する。資源の視点からラフカット能力計画によるチェックが行われる。生成された情報は、販売部門と製造部門との間で共有され、納期回答などにも利用される。

#### (5) 資材所要量計画

MRP ロジックにより、最終製品レベルの計画情報を、BOM 情報を用いて順次展開し、必要となる資材の調達や、それぞれの部品レベルで生産開始の時期と数量を計算する。計画対象は比較的短期から中期となり、必要に応じて、不定期に部分的な再計画を行う。

#### (6) 能力所要量計画

能力所要量計画 (CRP) では、各ワークセンターについての負荷の山積み山崩しを行いながら、最終的に、各ワークセンター与えられた有限の生産能力以下に負荷が収まるように生産の開始時期を調整する。品目ごとのルーティング情報が必要となる。

#### (7) 詳細スケジューリング

個々の生産現場において、上位の意思決定モジュールによって決定された生産計画で、該当部分の当日から数日分を対象に、その現場に存在する実際の設備を用いて、具体的にどのようなルートで、どのような順序で作業を行うかを決定する。

#### (8) ディスパッチング

個々の作業オーダを、担当の作業員、担当の設備に対して発行する。作業オーダは、作業が終了する都度発行する場合と、たとえばそのシフト内で行う分をまとめて発行する場合などがある。作業員は、作業オーダを受けて、必要工具など具体的な段取り等を行う。

#### (9) 監視/トラッキング

生産を実行した結果は、常に監視し記録する。監視し記録されたデータは、後になって品質解析やトレーシングに利用される。また、この情報

は、詳細スケジューリングの入力情報となり、次の詳細スケジュールのサイクルにおいて、実績が消しこまれる。

なお、図6において、矩形の角が丸い意思決定モジュールは「計画」、つまり一定幅に区切られた期間（タイムバケットまたはタイムフェーズ）に対する意思決定であることを示し、そうでないものは「スケジューリング」、つまり、連続する時間軸上での意思決定であることを示している。

ここまで説明したような既存の情報システムのモジュール構成は、以下の3点で問題がある。まず、現場レベルの管理者と生産レベルの管理者の間で、情報が一方であり、上位の計画に生産現場の実際の状況が反映されないこと。そして、第二は、企業レベル管理者および生産レベル管理者それぞれについて、製品や部品などの生産数量に関する意思決定と、工場や設備の能力に関する意思決定が独立しているため、処理効率が非常に悪いこと。そして、第三は、製造技術部門や製品設計部門、さらにはキャッシュフローの最適化を指向した財務部門などとの連携をとるしくみが組み込まれていないという点である。

PSLXでは、従来の情報システムに代わる新しいAPSのためのシステム構成として、図7のようなフレームワークを提案している。ここで、特に中心に位置づけられている需給調整計画、基準日程計画、作業日程計画、そして詳細スケジューリングの部分が、APSシステムの核となっている。これらは、すでに述べた意思決定の階層にそのまま対応するものである。

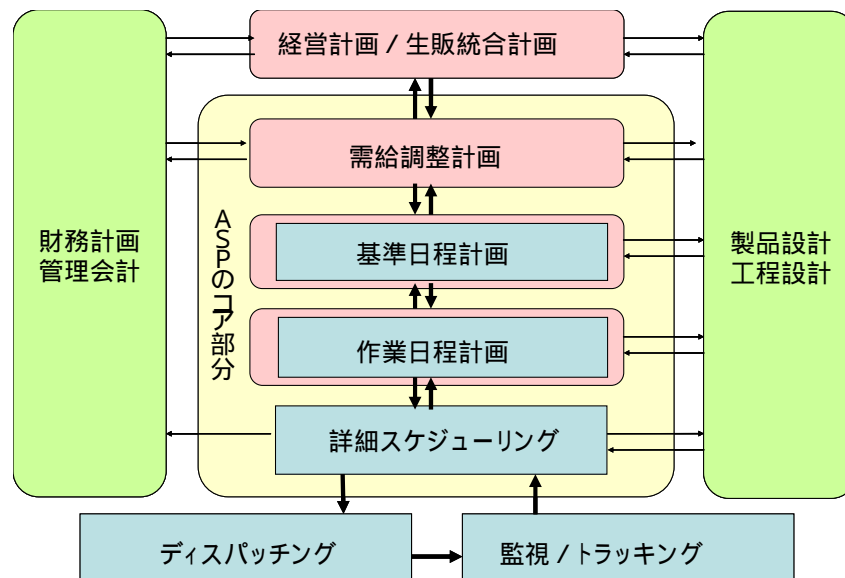
図6と図7とを比較して分かることとして、まず、基準日程計画および作業日程計画が、計画の要素とスケジューリングの要素を併せ持っているという点がある。つまり、必要に応じて、基準日程計画、あるいは作業日程計画は、従来のタイムバケットをベースとした処理ではなく、詳細スケジューリングと同様の連続時間軸上での意思決定を行えることを示唆している。これは、PSLXが提供する計画とスケジューリングの統合技術によって可能となる。

また、図7では、図6では2つに分かれていたモジュールが一つとなっている。まず、「販売計画/生産在庫計画」と「資源計画/能力計画」の

2つのモジュールは統合され「需給調整計画」となった。また、「資材所要量計画」と「能力所要量計画」も、「作業日程計画」として統合されている。これらは、ともに PSLX の概念データモデルにおいて、製品や部品や資材に関する概念と、設備や作業場などの資源の概念が、作業という概念を中心に明確に関係づけられていることで実現できる。

さらに図7では、APS が、財務計画や管理会計のモジュール、製品設計や工程設計のモジュールとも柔軟に情報交換ができるしくみであることを示している。

図7  
APS による新しいアーキテクチャー



## 製造業の分類とビジネス戦略

PSLX が提唱する情報システムアーキテクチャーを個々の企業に適用するにあたって、それぞれの企業が属している製造業の生産方法の分類、およびビジネス戦略上の分類を正しく認識した上で、具体的な情報システムの実装にむけて設計を進めるべきである。APS による意思決定のしくみは、それぞれのビジネスプロセスの集合体であり、さらに各ビジネスプロセスは、その企業の属しているカテゴリーによってその展開方法もビジネス的な意味も異なってくる。

まず、生産方法の分類として、生産計画あるいはスケジューリングといった意思決定のしくみの視点から、以下の3つの区分が重要となる。

#### (1) フロー型生産

これには、化学プラントなどの連続型の生産と、離散型生産における繰返し生産が対応する。いずれの場合も、生産数量は常に平準化されており、一定の割合で継続的に生産が行われる。在庫は基本的にパイプライン型の在庫が中心となり、中間での仕掛は原則としてない。この形態の生産では、ラインバランスやオーダ管理が重要となり、プロセス内のスケジューリングはほとんど必要ない。

#### (2) 断続型生産

生産の繰返し性はあるが、製造品目の量のばらつきが大きい。ほとんどの製造業は、このタイプに属する。離散型の生産では多品種小ロット生産が、プロセス系ではバッチプロセスによる生産がこれに対応する。スケジューリングでは、品種の切替えによるロスの最小化や、ボトルネックの管理などが重要となる。また、上位の計画では、多様で変化する需要に最小の設備投資で対応するための、需給バランス調整が重要となる。

#### (3) プロジェクト型生産

一回の生産を行うにあたり、資材や部品などの手配のみならず、生産設備や作業者などの資源も合わせて調達し、生産が終わったらすべて開放する方式。繰返し性はない。設計プロセスも含む場合が多い。造船や航空機などの大規模構造物である製品などが該当する。工場建設もこれに含まれる。プロジェクトスケジューリングの手法が多く利用される。

これらの3種類の分類は、その製造業が扱う製品や市場の特性によってある程度決められており、選択の余地がない。これに対して、以下にあげるビジネス戦略上の分類は、まさにその企業がどのようなビジネス戦略の上で意思決定を行っていくかについての分類であり、個々の製造業が主体的に選択すべきものである。

#### (1) 見込生産 (Make To Stock)

これは、顧客からの受注に先立って見込で生産を行い、完成品在庫とし



てストックしたものの中から，受注に応じて出荷していく形式である．納入リードタイムは最短となるが，在庫保管費用が高いという欠点がある．一方，まとめて作ることができるので，製造コストは安くなる．長期にわたって製品が劣化あるいは陳腐化せず，需要が安定している場合には，この形態が望ましい．

#### ( 2 ) 受注生産 ( Make To Order )

この方式では，顧客からの受注があるまで生産を開始しない．受注後にはじめて生産を開始し，製品が完成し次第出荷する．納入リードタイムは極めて長い，在庫が基本的にゼロとなる．資材はあらかじめ調達する場合と，受注後に行う場合とがある．多品種で，繰返し性の少ない場合にはこの形態が望ましい．

#### ( 3 ) 受注設計生産 ( Design To Order )

あらかじめ製品として定まった形をもたず，要素技術のみを提示し，その都度顧客の要望に合わせて製品を設計し製造する方式である．製品の大きな雛形やサンプルをベースにカスタマイズする．納入リードタイムは長く，不確定である場合が多い．標準的な部分のみを先行して製造しておくことは可能である．

#### ( 4 ) 受注即応生産 ( Finish To Order )

受注生産型と見込生産型を混合した方式である．あらかじめ見込生産として受注前に生産を開始するが，BOMのあるレベルにおいて生産を中断し，それ以降は受注生産方式とする．完成品の在庫を持たないため，在庫量を低く抑えたままで，納入リードタイムを短縮できる．製品のバラエティが生産工程の最終段階において形成されるタイプの BOM をもつ製品に有効である．



PSLX ホワイトペーパー (p.28)

## 第三章

# 業務アクティビティ

---

### 業務アクティビティとは

---

ここで定義する業務アクティビティとは、生産現場の視点からみた場合の必要不可欠な業務の単位のことである。一般に多くの基幹情報システムのアーキテクチャーは、企業全体がもつ機能をトップダウンに分類し、それぞれの機能単位に情報システムとしてモジュール化する傾向がある。これに対して、PSLX コンソーシアムでは、生産現場の視点から見たボトムアップのアプローチを重視する。

生産現場に存在し、計画およびスケジューリングに関連するあらゆる情報は、一つあるいは複数の業務アクティビティによって処理されなければならない。言い換えると、ここで定義される業務アクティビティは、最終的に生産現場で具体化されるあらゆる情報のうち、少なくとも具体的なアクションを決定するために必要な情報の生成、修正、管理等を的確に行うためのものである。

業務アクティビティの粒度は、業務の最小単位であり、UML で一般に用いられているアクティビティの粒度よりも粗い。また、業務アクティビティは階層をもたずフラットである。各アクティビティ間は、理論上は重複がないように定義されている。ただし、現実問題との対応関係の中で、実際のある業務が複数の業務アクティビティにマッピング可能である場合はある。

各業務アクティビティは、一つまたは複数のユースケースを持つ。ここでユースケースとは、業務のある目的を達成するためのアクションの系列であり、現実世界と情報システムとのインタラクションの方式を定めたものである。ここでのアクションには、物理的な操作と情報処理の操作の両方を含む。

各ユースケースを実行する上で、異なる業務アクティビティとの連携が必要となる場合には、関連する業務アクティビティのそれぞれのユースケースに対してコラボレーションを別途定義する必要がある。業務モジュール間での情報の典型的な伝達パターンは、コラボレーションとして、PSLX 仕様として別途規約として定められる。

---

## 業務アクティビティモデル

---

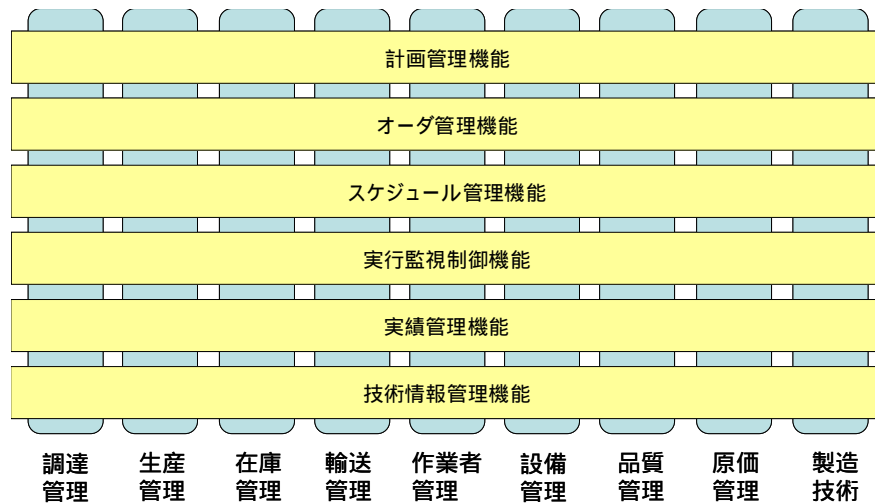
業務アクティビティを定義するうえで、以下の2つの視点を導入する。まず、第一が、業務の管理区分の視点である。業務の管理区分としては、調達管理、生産管理、在庫管理、輸送管理、作業管理、設備管理、品質管理、原価管理、そして製造技術の9つがある。これらの管理区分は、一般に、それぞれ独自の目的を持ち、独自の情報を持ち、ある程度独立して業務プロセスを実行できるため、企業内の個別の組織や担当者に対して、各管理区分のアイテムを割当てることが可能である。

第二の視点としてあげられるのが、機能区分の視点である。機能区分とは、管理区分の視点とは独立して定義可能なもので、業務アクティビティが計画やスケジューリングといった意思決定にどのように関わってい

るかという位置づけを示すものである。PSLXでは、機能区分として、計画管理機能、オーダ管理機能、スケジュール管理機能、実行監視制御機能、実績管理機能、技術情報管理機能の6つを定義している。

基本的に、第一の視点である管理区分の各項目は、ここで示したすべての機能区分の項目をその内部に含んでいるとみなすので、以下の図8のように、縦方向に業務の管理区分、横方向に機能区分をもつ9×6のマトリクスが描ける。業務アクティビティとは、このマトリクス上のそれぞれの交差部分に相当する粒度をもつ業務の総称のことである。

図8  
業務アクティビティの粒度



PSLX コンソーシアムでは、この各業務アクティビティに対して、業種ごとに標準的なユースケースを仕様として複数設定している。また同時に、業務アクティビティごとに必要となるデータクラスを、業務区分や機能区分ごとに業務アクティビティ間の基本的なコラボレーションを仕様として定めている。ここで定義されるユースケース、データクラス、そしてコラボレーション情報は、規約として強制されたものではないが、システム開発における基本設計や機能の検証等に非常に有用な情報となり得る。

## 管理区分による分類

以下に管理区分それぞれについて、その内容を簡単に説明する。

### (1) 調達管理

調達管理は、資材のサプライヤーや協力工場など、その企業の管理システムから見た外部で行われる生産活動を管理する。地理的に工場の内部にあっても、外部企業がその生産プロセスを受託している場合は、ここで管理する。また、資材のみではなく、設備や工具などの生産システムの一部の調達も、ここで管理する。

### (2) 生産管理

資材や部品を、加工や組立てによりその形状や性質を変化させる行為を管理する。設備や作業者などの生産資源が、さまざまな生産オーダに対して、効率的に活用されるための管理を行う。作業のディスパッチング管理、生産のスピードの管理、ボトルネック管理、部品や仕掛品の管理、かんばんや工程内在庫の管理などが対象となる。

### (3) 在庫管理

生産対象となる資材や部品や仕掛品、あるいは生産の結果得られた中間製品や最終製品など、サプライヤーからモノが到着した瞬間から、顧客へ到着する瞬間まで、生産対象となるモノのすべての状態を管理する。特に、資材在庫、完成品在庫、そして中間仕掛品在庫などについて、その保管場所や設備におけるモノの状態管理や出入りの管理を行い、適正な在庫量を保持する。

### (4) 輸送管理

生産を行う上で、工程間あるいはエリア間のモノの移動を管理する。輸送手段、輸送経路、輸送のための資源の確保、輸送ロットサイズの管理などが含まれる。企業間における製品や資材の輸送や配送が、その企業の管理下にある場合には、それらの管理も含む。必要なモノを必要な時に必要な場所に必要量提供することが目的となる。

### (5) 作業管理

生産現場におけるさまざまな生産活動に直接的または間接的に関わる作業者の管理を行う。作業者の勤怠管理、個別の技能やスキルの管理、グルーピング、元単位管理、作業工数管理、トレーニング、エリア配置、シフト管理などがある。作業者が常に最大のパフォーマンスを発揮できるようにすることが目的となる。

#### (6) 設備管理

生産に関連するあらゆる設備について、その能力を常に設計値どおりに維持できるように、状態のモニタリング、トラッキング、使用履歴管理、故障の事前予測、現象の解析、ユーティリティの管理、サプライ品の管理、定期点検、定期部品交換などを行う。また、故障や異常が発生した場合に、早急に事態をもとの状態に復旧させる。

#### (7) 品質管理

製品の品質を常にあらかじめ設計された一定の値に保つために、最終製品のみならず、生産プロセスのあらゆる過程において品質チェックを行う。また、生産対象となる資材や部品や製品や半製品のみならず、生産の方法そのもののチェックも合わせて行う。また、QC 活動などで統計的解析による問題点の発見と、品質向上のための活動を行う。

#### (8) 原価管理

工場の各設備や人的資源に対して、適切な投資が行われ、あらかじめ予想したパフォーマンスをあげているかを、コストとスループットの視点から管理する。また、工場で生産された製品の価格が適切に設定されているかどうかを、市場での価値とコストとの関係によりチェックし、戦略的な価格設定のための基礎データを提供する。

#### (9) 製造技術

設計された製品に対応する生産プロセスと構成設備を設計し、実際に生産現場に生産ラインを設置する。高度なエンジニアリング技術と知識をもとに、量産前の生産準備のフェーズにおける試作ラインの設置と、量産ラインの設置およびその後の設備の改良や改善を行う。そして、日々の生産に必要な各技術情報の設定や、マニュアル等の整備を行う。

## 機能区分による分類

機能区分による分類は、管理区分の各要素間の水平的な視点を提供する。管理区分による分類としてあげられている各管理業務は、それぞれが、その内部に以下にあげるような機能区分をもっている。

### (1) 計画管理機能

それぞれの管理業務の中長期的な活動は、ある計画期間において、日や週などを単位とする期間ごとの設定値あるいは目標値によって管理される。計画管理機能では、これらの値の設定、変更、予実管理、過去データの管理を行う。また、全体最適化を指向して、関連部門との連携、調整などを行う。

### (2) オーダ管理機能

各管理区分の要素において、具体的なアクションを要求するオーダを管理する。オーダは外部から与えられる場合、外部に送られる場合、そして内部でより具体化される場合がある。各オーダは、確度や精度が時間とともに変化する。これらのオーダの生成、変更修正などの管理、関係づけ管理、および状態管理を行う。

### (3) スケジュール管理機能

業務に関係するすべてのアクションは、最終的に必要とする資源の確保や関連する制約の解消、そして他の業務に関係するアクションとの整合性を確認した上で実行に移される。スケジュール管理機能は、生産現場のスケジュールリングを行い、さらに、その結果得られたスケジュールの内容を管理する。

### (4) 実行監視制御機能

生産現場における作業を、あらかじめ作成したスケジュールをもとに、さまざまな現実的な状況を加味しながら実行する。また、実行状況を常に監視し、異常に対して、あらかじめ設定された手順にそって即時に対応するとともに、必要に応じてスケジュール変更を要求する。また、その状況を的確に関連部門に報告する。



#### (5) 実績管理機能

生産現場で行われたさまざまな作業に対し、その結果を記録し時系列情報として保存する。ここで得られた情報は、その利用形態に合わせて再構成され、さまざまな視点から集約され解析される。異常が発生した場合の原因追求や、対応策を決定する際の基礎データや、品質を保证するための理論的な根拠としても利用される。

#### (6) 技術情報管理機能

各業務において技術的な情報として利用されている指標やパラメータ、あるいは設計文書や図面などを、その更新履歴も合わせて管理する。また、これらの技術情報のうち、各業務部門間で関連するものについて、その整合性の管理や、変更に対する利害調整、そして情報のセキュリティ管理もここで行う。

---

## 業務アクティビティ間のデータ交換

---

現在、PSLX 仕様では、9つの業務区分、6つの機能区分で、 $9 \times 6 = 54$ 個の業務アクティビティが存在する。データ交換のための仕様を考える際に、これら54個について、あらゆる組み合わせは、 $54 \times 53$ 通りが存在する。しかし、PSLX 仕様では、データ交換は、同一の管理区分あるいは同一の機能区分のみに限定している。つまり、管理区分間でのデータ交換は $9 \times 8 (\times 6)$ とあり、機能区分間では、 $6 \times 5 (\times 9)$ とありとなる。ただし、現実的には、ビジネスプロセス上、データ交換が存在しないペアが多く、定義すべき内容はさらに少なくなる。

業務アクティビティ間のデータ交換は、各業務アクティビティのコラボレーション仕様の延長として定義される。業務アクティビティは、同一の管理区分あるいは同一の機能区分の業務アクティビティとコラボレーション可能である。なお、ひとつのコラボレーションは垂直型（同一管理区分内コラボレーション）と水平型（同一機能コラボレーション）のいずれかに属することになり、それ以外は存在しない。

一般に、UMLのコラボレーションの記述では、複数のトランザクション

を含む場合が多いが、PSLXでは、個々のトランザクションごとに分割して仕様として定義している。さらに各トランザクションは、以下の3つの単純なパターンに限定し、より複雑なコラボレーションはこれらの単純なパターンの組合せによって記述される。

#### (1) 業務の実行

ある業務アクティビティが別の業務アクティビティのユースケースを実行する。実行はユースケースの先頭に対して起動を行う場合と、ステップの途中で中断された処理を再開させる場合がある。送信側は、具体的な業務の内容をメッセージとして伝え、受信側は該当する処理を行った結果を報告する。

#### (2) 情報の取り出し(プル)

ある業務アクティビティが別の業務アクティビティが管理している情報を問い合わせる。情報は、相手側の業務アクティビティによって永続的に保持されているものである場合、あるいはそれらの永続化データをベースに加工されたものである場合、あるいはさらに他の業務エージェントとのコラボレーションによって得られるタイプのものである場合がある。

#### (3) 情報の設定(プッシュ)

ある業務アクティビティが別の業務アクティビティに対して情報をインプットする。受信側の業務アクティビティは、インプットされる情報の内容と送信者を、あらかじめ想定し受信できるしくみを用意していなければならない。送信側は、受信側がメッセージの内容とコンテキストを理解できるような識別情報を付加する必要がある。

上記の3つのパターンはあくまでも業務アクティビティのレベルのパターンであり、より下位の通信レベルでのパターンとは異なる。たとえば、上記のプッシュ型のコラボレーションは、通信プロトコル上ではプル型で実装可能である。なお、PSLXでは、より下位レベルの仕様である通信プロトコルや、通信セキュリティに関する仕様は、現時点では規定していない。

PSLX仕様では、このように、単純化された3つのタイプのコラボレーションによって業務アクティビティ間の関係を規定しており、より高次元

でのワークフローやビジネスプロセスは標準化の対象外となっている。  
これらの情報は、PSLX 仕様をベースとしたベストプラクティスとして、  
仕様とは別に別途公開していく予定である。



PSLX ホワイトペーパー (p.38)

## 第四章

# システム実装の基本仕様

---

### データ交換の方式

---

業務アクティビティ間のデータ交換は、人間がそこに直接介在する場合と、計算機がそれを代行して行う場合とがある。帳票などを用いた情報伝達は前者であり、通常の情報システムの設計が対象とするのは後者である。ここでは後者についてのみ議論することにする。これはさらに、大きく以下の2つの方式に分類できる。

#### (1) データ蓄積方式

第一の方式は、データ蓄積方式である。これはデータベース(DB)を介してデータ交換を行う方式である。データベースとしては、リレーショナルデータベース(RDB)が業務アクティビティ内の情報処理においてもっとも活用されるが、これを異なる業務アクティビティ間のデータ交換にも利用しようというものである。PSLXでは、データ蓄積方式によるデータ交換のための仕様として、RDB標準スキーマを規定している。

## (2) メッセージング方式

データ交換の第二の方式は、メッセージング方式である。これは、2つの業務アクティビティそれぞれに存在する2つの業務プログラムが、ダイレクトにデータ交換を行う方式である。この場合、業務プログラム間で交換されるメッセージの内容は、基本的にその都度生成され、データ交換が終了した時点で消滅する。PSLXでは、メッセージング方式のための標準仕様として、XML標準スキーマを規定している。

一般に、異なる2つの業務アクティビティがデータ交換を行う場合、まず、双方であらかじめ合意された手順によるデータ交換を行わなければならない。これを通信のための業務プロトコルと呼ぶ。データ交換が正常に行われなかった場合の例外処理の方法などもここで規定される。そして、さらに重要な点として、いかにして交換されたデータの内容を一方が意図した意味で他方に理解してもらうかという問題に対処する必要がある。

PSLXで規定しているRDB標準スキーマもXML標準スキーマも、ともに特定のデータ構造(スキーマ)が規定されているが、これらのデータ構造(スキーマ)と、対応する意味との関係を、だれもが正しく理解できるように、まず、最初に、できるだけ厳密かつ分かりやすい形で定義することから始めなければならない。

---

## PSLX オントロロジー

---

オントロロジーとは、対象とする問題に関連するあらゆる情報の意味を表現するためのもっとも根源的な要素のことである。PSLXでは、その対象とする問題が製造業における生産計画およびスケジューリング問題であるため、PSLXオントロロジーは、生産計画およびスケジューリングに関するあらゆる情報の意味を表現するための統一的な基礎を提供する。

PSLXオントロロジーは、独自の構造を持っており、これが対象とする生産計画およびスケジューリングの問題特有の構造に対応している。個々の

オントロジーの意味は、その構造によって意味が制約されている。個々のオントロジーに付けられた語は、突き詰めれば単なるラベルであり、用語の字面よりも、むしろその構造的な位置関係のほうが重要となっている。

PSLX 仕様では、UML のクラス構造図によってオントロジーの構造を表現するが、オントロジーはオブジェクト指向モデリングでいうクラスではない。各オントロジーは、表現する意味の特徴から、分類可能ないくつかのタイプ（ステレオタイプ）によって便宜的に記述されている。それらのタイプは、各オントロジーの意味を理解するための助けとなる。なお、オントロジーには、属性もメソッドも存在しない。

生産計画およびスケジューリングに関係するあらゆる情報は、ここで規定されるオントロジーとの関係を定義することで、その意味を明らかにすることができる。業務アクティビティで扱う情報を表現するためのあらゆるオブジェクトは、必ず直接または間接的に PSLX オントロジーと関連をもっていなければならない。PSLX 仕様では、これらの関係もまた、UML のクラス構造図によって記述する。

---

## オントロジーの分類と構造

---

PSLX オントロジーは、以下に示すような共通の性質を持ったいくつかのグループに分類することができる。これらの分類は、オントロジーによって説明されるより具体的なクラスのタイプ（ステレオタイプ）として利用される。

### (1) エンジニアリング要素 (Engineering)

これは、設計部門や技術部門において、長い経験と多くの知識の集大成として、あらかじめ設定されている情報であり、企業の製造技術あるいはノウハウそのものに相当する。この情報は、実際の業務の中で繰返し参照され、必要に応じて更新される。オントロジーとして、作用 (Function)、性能 (Capability)、品目 (Item)、そして事象 (Event) の4つが存在する。

### (2) スケジューリング要素 (Scheduling)

作業を具体的なアクションとして実際に行う場合に必要となる情報であり、ここでは、時間と空間との対応関係を規定する。つまり、エンジニアリング要素を、現実世界の中で個々に具体化したものがこれに相当する。スケジューリングの過程においては、仮想的な将来の状況を表現することも可能である。オントロロジーとして、作業 (Operation)、タスク (Task)、ロット (Lot)、そしてアクション (Action) の4つが存在する。

### (3) 時系列要素 (Temporal)

実際の物理的な現実として、モノがどのような挙動をしており、どのような性質や特徴を持っているかを、時間軸上でのなんらかの変化から表現するための要素である。これらは、通常、物理的なモノと、実際の作業との関係の中で、その具体的な適用結果として生成されるものである。オントロロジーとして、能力 (Capacity)、在庫 (Inventory)、変化 (Change) の3つが存在する。

### (4) 物理的要素 (Physical)

物理的要素とは、生産現場に物理的に存在するモノ、あるいは物理的に存在するとみなせるモノと、一対一に対応する情報である。ある空間を占有する識別可能なモノをこれによって個々に表現する。ここで定義されるモノは時制をもたず、それらは付随する時系列要素によって表現される。オントロロジーとしては、資源 (Resource) が存在する。

### (5) オーダ要素 (Order)

生産現場において行われるさまざまな作業やアクションは、すべてオーダに基づいて生成され実行される。オーダはさらに別のオーダを生成することもある。企業内の主体的な変化の源泉は、すべてオーダによってもたらされる。オーダ要素は、業務の各要素で相手に対して具体的なアクションを要求するための媒体となる。オントロロジーとしては、作業オーダ (Work order)、ロットオーダ (Lot order)、タスクオーダ (Task order) の3つがある。

### (6) 計画集計要素 (Plan)

さまざまなオーダの連鎖による企業内の活動が、企業全体としてあるい



はあるまとまった部門内で整合性がとれており、かつ中長期的に見てもバランスがとれているかどうかは、計画集計要素によって判断する。計画集計要素は、ある一定期間について、関連するさまざまな情報を集計した情報である。これらに対して、制約や評価指標を設定することで、全体最適化が可能となる。オントロジーとしては、生産計画 (Production plan)、能力計画 (Capacity plan)、在庫計画 (Inventory plan) の3つがある。

#### (7) 主体者要素 (Party)

主体者要素とは、それ自体が主体的に意思決定を行うことができ、意思決定において最終的な評価指標を共有することができる範囲を表す。また、これは、企業活動における各資源の最終的な所有者を表現し、そこから生み出される収益の帰属先を表現するためにも利用できる。オントロジーとしては、メーカ (Maker)、サプライヤー (Supplier)、顧客 (Customer) の3つがある。

#### (8) 時間的要素 (Time)

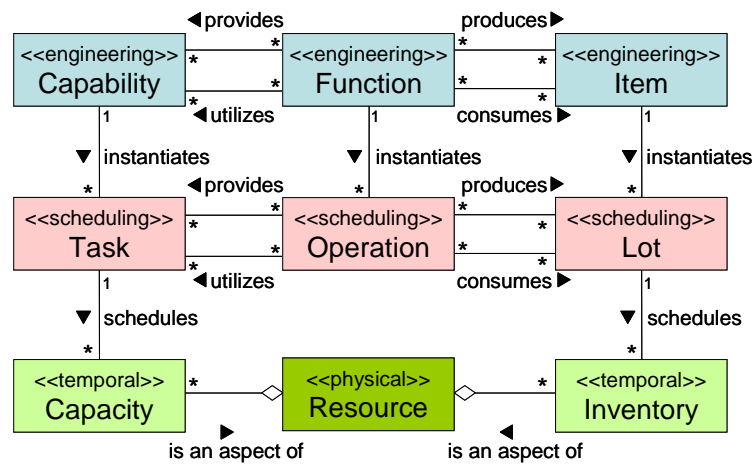
時間の存在は、哲学的オントロジーでも議論の中心となるものであるが、ここでは時間というものがすでに存在しているという前提で、その表現のための要素を提供する。具体的には、時刻 (Time point)、期間 (Time period)、時間 (Time span) の3つのオントロジーが存在する。

#### (9) 空間的要素 (Space)

空間的要素も、時間的要素とともに、最も重要な概念の一つである。空間的要素は、管理対象に対して空間的な視点から粒度を設定する際に有効な概念となる。PSLX オントロジーでは、位置 (Position)、範囲 (Region)、距離 (Distance) の3つのオントロジーが存在する。

現在、PSLX オントロジーは、以上の分類のもと、27個が定義されている。これらのオントロジーは、お互いの関係構造が設定されており、これらの位置関係によって、その意味が規定されている。以下の図9～図11に、それらの関係の一部を示し、それぞれの位置がもっている特別な意味を簡単に説明する。

図 9  
 オントロジー  
 (その1)



まず, 図9にある9つのオントロジーは, もっとも基本的なものであり, エンジニアリング要素の側面, スケジューリング要素の側面, 時系列要素の側面, そして物理的要素の側面の間の位置関係を見ることができる. エンジニアリング要素をもとに, スケジューリング要素が生成され, さらに物理的要素をスケジューリング要素との接点に時系列要素が存在している.

また, 図9における右の列は, すべて, 生産対象となるモノの側面の表現であり, 生産によって生成され消費されることが示されている. これに対して, 左の列は, 生産を行うための資源としてのモノの側面であるが, これらは生産後に消滅しない. 逆にこれらは, ある時刻や期間を超えて融通することができないため, それぞれの時刻や期間内で, 資源がもつ能力を利用したり, 逆に提供したりする.

図 10  
オントロジー  
(その 2)

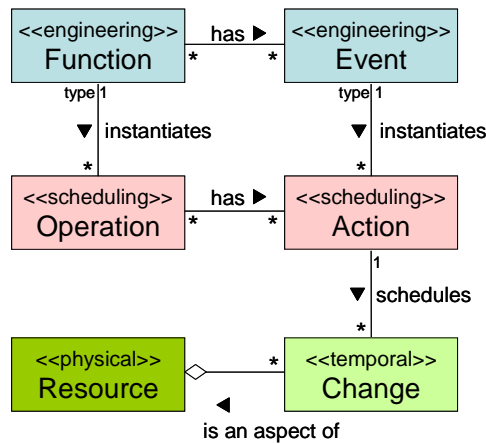


図 10 は，図 9 を補足するものであり，列がひとつ追加されている．追加された列は，エンジニアリング要素，スケジューリング要素，時系列要素それぞれについて，もっともプリミティブなものであり，図 9 の右列，左列で表現できないものはすべてこの 3 つのオントロジーで表現される．たとえばアクション (Action) によって，ある作業 (Operation) の内部を，複数のパートに詳細化することができる．

図 11  
オントロジー  
(その 3)

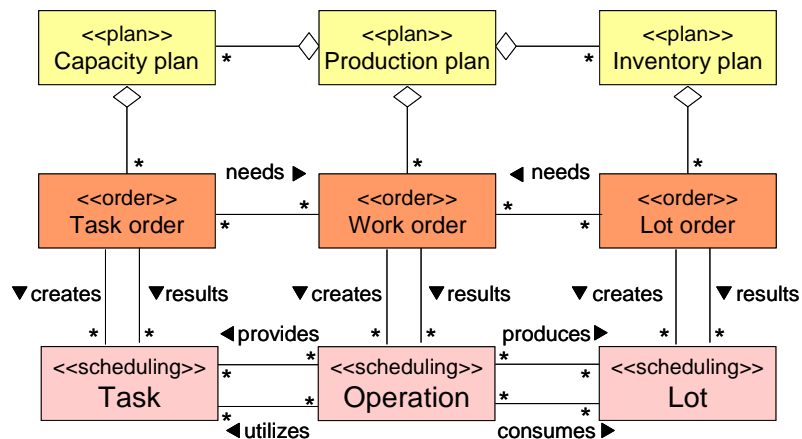


図 11 は，スケジューリング要素の 3 つのオントロジーに対して，オーダ要素，計画集計要素がどのような位置関係にあるかを示している．オーダ要素は，スケジューリング要素を生成し，計画集計要素は，オーダ要素の集約となっている．また，図 9 に対応して，右列が消費可能資源を，左列が消費不可能資源に関する情報であることを示している．なお，図

10 の右列に対するオーダ要素は存在せず、それらの具体的な指示は作業情報の一部として行われる。

---

## APS ドメインモデル

---

オントロジーは、生産計画およびスケジューリングの問題に対して普遍的に存在するものであり、PSLX 以外の他の標準との間でも互換性がある。しかし、これは抽象度が高すぎて、そのままの形で一般的に利用することは困難である。したがって、より抽象レベルが低く、一般的な生産管理の世界で普段利用している用語に対応したレベルのオブジェクトが必要となる。

各企業固有の業務を、情報システムの実装形態に依存しない形で表現したものが概念データモデルであり、そこにはさまざまな業務用語とその関係構造が定義されている。そして、この節で説明する APS ドメインモデルは、概念データモデルの標準形(参照モデル)である。PSLX が提供する APS ドメインモデルを利用することで、各企業独自の概念データモデルを比較的簡単に定義することが可能となる。

APS ドメインモデルもまた、UML のクラス構造図によって記述される。APS ドメインモデルに登場する各クラスの定義は、すべて PSLX オントロジーと関係づけられたクラス構造図の中で行われなければならない。クラスの記述方法としては、以下のいずれかとし、これ以外の定義はできない。

- (1) オントロジーに該当するクラス
- (2) 上記(1)のサブクラス
- (3) 上記(1)の集約クラス
- (4) 上記(1)~(3)と特定の関係をもつクラス

なお、上記の(1)~(4)のクラス間で、さらに特定の関係を追加することができる。ここで特定の関係とは、2つのクラス間の数的関係および役割関係のことであり、これらの情報はともに必須事項となってい

る。なお，APS ドメインモデルおよび概念データモデルは属性を持たない。

APS ドメインモデルの一部を、以下の図 12～図 15 に示す。まず，図 12 では，作業オントロジーに対応するクラスと，そのサブクラスが示されている。製造作業（Manufacturing），段取作業（Setup），検査作業（Inspection），輸送作業（Transport），管理作業（Administration），設計作業（Product Design），工程設置（Plant engineering），保守作業（Maintenance），出荷作業（Ship goods），入荷作業（Receive goods），出庫作業（Issue inventory），入庫作業（Store inventory）などがある。ステレオタイプに\*マークがあるクラスは、オントロジー対応クラスである。

図 12  
作業のサブクラス

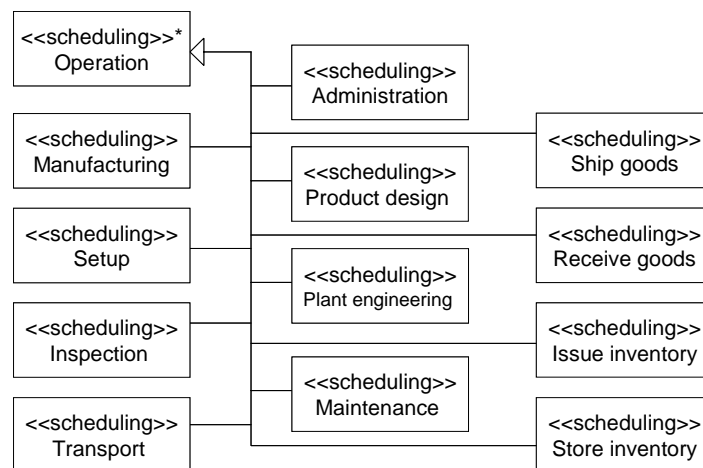
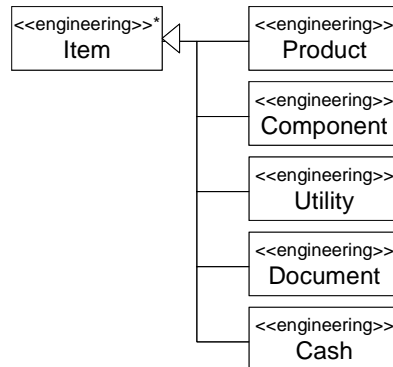


図 13 では，品目（Item）オントロジーに対応するクラスと，そのサブクラスが示されている。これらは，エンジニアリング要素に属しており，実在し個々に異なるさまざまなモノに対して，その共通的な特徴を概念的に示す抽象クラスである。製品（Product），構成品（Component），ユーティリティ（Utility），文書情報（Document），キャッシュ（Cash）がある。

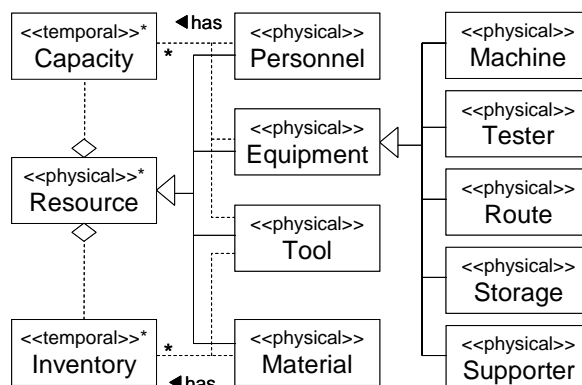
図 13  
品目のサブクラス



続く図 14 では、資源 (Resource) オントロジーに対応するクラスと、そのサブクラスが示されている。まず、資源クラスのサブクラスとして、作業員 (Personnel)、設備 (Equipment)、ツール (Tool)、そして資材 (Material) が示されており、さらに設備 (Equipment) のサブクラスとして、製造設備 (Machine)、検査設備 (Tester)、輸送設備 (Route)、保管設備 (Storage)、保守設備 (Supporter) が設定されている。

資源のそれぞれのサブクラスには、消費可能資源と、消費不可能資源、そして特殊なケースとしてその両方の性格をもつ資源がある。図 14 において、消費可能資源は在庫 (Inventory) との関連を持ち、消費不可能資源は能力 (Capacity) との関係を持たせることで、これらのサブクラスの特徴に関する識別が表現されている。

図 14  
資源のサブクラス

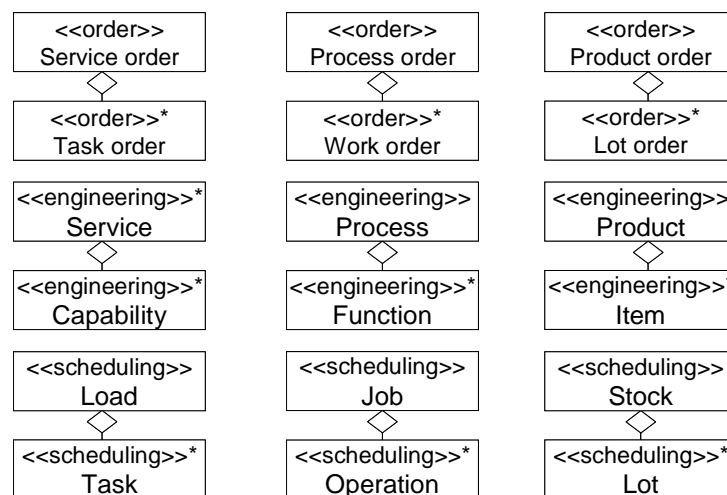


オントロジーに対応するクラスの多くは、図 15 に示すように、それぞれ集約クラスを持ち、これによって、階層的な意思決定における関係を表

現することができる。特に、計画レベルでのクラスと、詳細スケジューリングレベルでのクラスを別途設定し、それぞれの対応関係をとる場合に活用される。

まず、オーダ要素については、タスクオーダに対応するサービスオーダ (Service order)、作業オーダに対応するプロセスオーダ (Process order)、ロットオーダに対応する製品オーダ (Product order) がある。また、エンジニアリング要素では、性能 (Capability) の集約としてサービス (Service)、作用 (Function) の集約としてプロセス (Process)、品目の集約として製品 (Product) がある。そして最後に、スケジューリング要素について、タスク (Task) の集約として負荷 (Load)、作業 (Operation) の集約としてジョブ (Job)、ロット (Lot) の集約としてストック (Stock) がある。

図 15  
オントロジー  
クラスと集約  
クラス



PSLX 仕様で定義されたすべての APS ドメインモデルは、別途仕様書に、より詳細なクラス図およびその詳細説明がある。

## 実装のための標準スキーマ

APS ドメインモデルは、各業務アクティビティで利用されるさまざまな用語に対する意味の定義として重要な働きをする。一方で、計算機上で

の処理に適したスキーマは別途異なる形態として存在している。通常、これらの実装用スキーマは、システム個々に対してそれぞれ存在するため、これらの実装用スキーマの共通部分を抽出し、APS ドメインモデルに対応するように一般化した実装のための共通スキーマが必要となる。PSLX では、共通スキーマとして、RDB 標準スキーマと、XML 標準スキーマの 2 つの標準スキーマを規定している。

PSLX で規定する 2 つの標準スキーマは、APS ドメインモデルと同様に、PSLX オントロジーによって基本的な概念が規定されている。これは、APS ドメインモデルのスキーマをベースとして、以下の変換手順を繰り返すことによって生成できなければならない。言い換えると、2 つの標準スキーマは、PSLX の APS ドメインモデルによって説明可能な構造となっている必要がある。なお、PSLX の標準スキーマは、概念データモデルと異なり属性を持つ。

- ( 1 ) クラス名の変更
- ( 2 ) クラスの統合 ( 直接関係をもつ場合 )
- ( 3 ) 関係のクラス化
- ( 4 ) 関係の属性化
- ( 5 ) クラスの削除
- ( 6 ) クラスの追加
- ( 7 ) 関係の追加
- ( 8 ) 属性の追加

上記の各処理の系列は、上記の順でなければならない。また、処理系列は冗長であってはならない。つまり、同一の標準スキーマを生成するより短い処理系列が別途存在してはならない。なお、APS ドメインモデルの変換ルールは、別途仕様として詳細に規定されている。

---

## RDB 標準スキーマ

---

RDB 標準スキーマは、業務アクティビティ間で、蓄積型のデータ交換を行う際の共通スキーマである。RDB 標準スキーマの各クラスは、実際の



RDB 上で SQL 等により生成されるビューに相当する。RDB 標準スキーマの最大の特徴は、それぞれのクラスがお互いに明示的な関係を持たないことである。各クラスの関係は、実装 RDB 上のスキーマにおいて、PSLX オントロジーおよび APS ドメインモデルの構造にしたがって設定される。また、各クラスおよびクラスの属性は、その生成に過程を逆にたどることで、APS ドメインモデルとの関係を明らかにすることができる。

それぞれが異なる形態の実装 RDB では、共通のインタフェースとして、RDB 標準スキーマが設定するクラスに対応するビューを提供する必要がある。実装 RDB スキーマのどのクラス(テーブル)が、RDB 標準スキーマのどのクラス(ビュー)に対応しているか、あるいは、実装 RDB スキーマのどのクラス(テーブル)のどの属性(フィールド)が、RDB 標準スキーマのどのクラス(ビュー)のどの属性(フィールド)に対応しているかを、実装を担当するシステムエンジニアは、個々に設定しなければならない。

RDB 標準スキーマとして規定されているクラスは、以下のカテゴリーに分類できる。各クラスの詳細の定義は、別途 PSLX 仕様の中で定義されている。

#### (1) マスタクラス

マスタクラスに属するクラスは、APS ドメインモデルのクラスにそのまま相当するクラスで、かつ、複数の業務アクティビティから何度も参照される可能性がある。マスタクラスに属するものとしては、顧客マスタ、配送先マスタ、業者マスタ、能力マスタ、作業者マスタ、設備マスタ、工具マスタ、倉庫マスタ、作業マスタ、配送マスタ、プロセスマスタ、製品マスタ、資材マスタ、品目マスタ、作業区マスタ、工程マスタ、ルートマスタがある。

#### (2) 関係構造クラス

関係構造クラスは、APS ドメインモデルのクラス間の関係を記述するためのクラスである。関係をもつ2つのクラスの数的関係が、多対多の場合に、RDB の性格上この種のクラスが必要となる。関係構造クラスに属するものとしては、作業表(品目 作業)、資源表(品目 資源)、プロ

セス表 (製品 プロセス), 作業区表 (製品 エリア), 資源割当 (作業 資源), 負荷割当 (プロセス エリア), 先行関係 (作業 作業), 製品構成 (製品 資材), 作業構成 (作業 プロセス) がある。

### (3) オーダクラス

オーダクラスは, APS ドメインモデルにおいて, オーダ要素に分類されるクラスあるいはそのサブクラスに相当するクラスである。オーダクラスに属するものとしては, 顧客オーダ, 資材オーダ, 生産オーダ, 作業オーダ, 出荷オーダ, 入荷オーダ, 出庫オーダ, 入庫オーダがある。

### (4) 期間計画クラス

期間計画クラスは, 一定の時間幅をもつ期間における計画値や, 実行結果の集計値を表現するためのクラスである。期間計画クラスとしては, 製品ボリューム計画, 工場能力計画, 製品別生産数量計画, エリア別負荷計画, 期別需要量計画, 期別在庫量計画, 期別輸送量計画, 資材所要量計画がある。

### (5) スケジューリングクラス

スケジューリングクラスは, 時々刻々と変化する状況に対応してアクションをとるために必要な情報を表現するためのクラスである。これらのクラスは, すべて連続時間軸上の特定の時刻との関係を示すことで対応するデータの意味づけを行う。スケジューリングクラスに分類されるものとして, 工場カレンダー, 設備カレンダー, 作業者カレンダー, 資源能力データ, ロットデータ, タスクデータ, 製品在庫データ, 資材在庫データ, 資源負荷データ, 作業区負荷データ, 作業追跡データがある。

### (6) 処理実績クラス

処理実績クラスは, 現実の工場における現実の処理を行った結果を表すクラスである。これには, 物理的な状況をモニタリングした結果として得られた情報と, 具体的な情報処理を行った結果として確定 (コミット) した情報が含まれる。処理実績クラスには, オーダベギングデータ, ロットベギングデータ, 作業実績データ, 生産実績データ, 稼働実績データ, 入荷実績データ, 出荷実績データ, 入庫実績データ, 出庫実績データ, オーダ送信データ, オーダ受信データがある。

## XML 標準スキーマ

XML 標準スキーマは、業務アクティビティ間で、メッセージ型のデータ交換を行う際の標準スキーマである。メッセージ型のデータ交換の具体例としては、Web サービスを用いて Web アプリケーションサーバとクライアント業務プログラムが情報交換する例や、従来の固定長の EDI に変わって登場しつつある XML / EDI のメッセージコンテンツとして利用する例などが考えられる。

XML 標準スキーマに登場するクラスは、RDB 標準スキーマと異なり、クラス間の構造をもつ。ただし、XML が基本的に階層構造をもつ文書の記述方式として発展した経緯もあり、基本的にツリー構造である。ネットワーク構造を構成するリンクは、クラスの特定の属性値の参照先として記述され、XML の階層関係とは区別されている。属性値による参照先の ID は、あらかじめ単一キーによってインデックス化された特定のクラスの ID とする。ID を用いたクラス間の参照関係は常に一方向である。

業務アクティビティ間のトランザクションごとに、XML 標準スキーマのトップレベルのクラスが定義される。このスキーマの特徴として、同一のクラスが、複数の業務トランザクションクラスの下位に登場し、そのコンテキスト（業務トランザクションの種類）によって、そのクラスがもつ制約が異なる場合がある。トップレベルとなる業務アクティビティ間のトランザクションに対応したクラス数が非常に多いのに対して、下位レベルのクラスの数はいくつか少ない。

トップレベルのクラスを除く、第二レベル以下のクラスは、以下のように分類できる。なお、各分類にある個々のクラスの詳細仕様は、別途 XML 標準スキーマ仕様書において規定されている。

### (1) 基本要素

これは PSLX オントロジーに対応した要素である。これらは常に、メッセージの最上位に設定される。顧客 (customer)、サプライヤー (supplier)、品目 (item)、資源 (resource)、作用 (function)、作業 (operation)、オーダー (order)、ロット (lot)、タスク (task) の 9 個の要素が存在する。

#### ( 2 ) 管理要素

基本要素と同様に各メッセージの最上位に来る．基本要素を定義する上で共通的に必要となる基本要素であり，それ単独では重要な意味をもたない．スケール ( scale ) , 基準時刻 ( stone ) の 2 つが定義されている．

#### ( 3 ) 付帯情報

基本要素がもつ属性情報の中で，意思決定上あるいは管理上の理由から必要となるもの .備考 ( description ) ,優先度 ( priority ) 表示方法 ( display ) の 3 つが定義されている．

#### ( 4 ) 関係要素

これは，基本要素間の関係を表す要素である．生成する ( produce ) , 生成される ( produced ) ,消費する ( consume ) ,消費される ( consumed ) , 割当て ( assign ) ,割当てられる ( assigned ) ,先行する ( predecessor ) , 後続する ( successor ) , ペギング ( pegging ) , 部分 ( partof ) の 10 個の要素が定義されている．

#### ( 7 ) 属性要素

基本要素の属性を定義するための要素であり，時系列情報の設定を行うことができる．仕様 ( spec ) , 位置 ( location ) , 進捗 ( progress ) , 負荷 ( load ) , 在庫 ( stock ) , 能力 ( available ) , カレンダー ( calendar ) の 7 つの要素が定義されている．

#### ( 8 ) 事象要素

事象要素は，事象オントロジーに対応するクラスから派生した要素であり，作業に含まれる開始 ( start ) , 終了 ( end ) , 事象 ( event ) や，オーダに関するものとしてリリース ( release ) , 納期 ( duetime ) の合計 5 つの要素が定義されている．

#### ( 9 ) 基本データ要素

これは ,各要素の属性値を表すための基本データ要素である 数値 ( qty ) , 価格 ( price ) , 定性値 ( char ) , 時間 ( duration ) , 時刻 ( time ) の 5 つが定義されている．

( 1 0 ) 補助データ要素

基本データ要素で定義された値に対して，その範囲の設定など補助的な情報を表現するためのものである．最小値 ( min ) , 最大値 ( max ) , 最早値 ( earliest ) , 最遅値 ( latest ) , 最短値 ( shortest ) , 最長値 ( longest ) , 候補値 ( enumerate ) の 7 つが定義されている．

最上位のクラスは，業務アクティビティ間のデータ交換コラボレーションに対応して，別途 PSLX 仕様の中で規定されている．また，第一レベル以下のクラスで表現可能なあらゆる情報を含む，PSLX ( pslx ) というクラスが存在する．



PSLX ホワイトペーパー (p.56)

## 第五章

# 仕様の利用形態とメリット

---

### 仕様のステークホルダー

---

PSLX 仕様書は、APS を実際の製造業の管理方式として具体的に実装するためのツールであり、複数の利害関係者（ステークホルダー）の全体としての利得が向上するための規約でもある。PSLX 仕様のメリットとしては、以下の点があげられる。

- ( 1 ) APS 実現のためのグランドデザイン策定支援
- ( 2 ) システムのライフサイクル支援、段階的開発の支援
- ( 3 ) 異なるシステム間の相互接続性の確保の実現
- ( 4 ) システム拡張性、アドイン型アプリケーションの実現
- ( 5 ) ユーザ中心のシステム、成長可能なシステムの実現
- ( 6 ) データ中心の管理、データの保守、再設計支援

ここで、いま一度、各ステークホルダーにとっての仕様の意味とメリッ

トについて確認しておく。

#### ( 1 ) 製造業の経営者

製造業の経営者、あるいは経営企画担当者にとって、APS が具体的にどのような機能をもち、自社に対してどのようなメリットをもたらすかについて、単なるセールストークではなく、技術的な裏づけのある情報をもとに具体的な検討が可能となる。場合によって、企業の明暗を分けることにもなる大規模な情報システム投資にあたり、主体的に明確なビジョンをもって、企業のグランドデザインを書くことができる。

#### ( 2 ) 製造業の業務設計者

製造業の部門ごとの業務を掌握し、常に改善、改良を繰り返しながら今までビジネスプロセスを構築してきた業務設計者にとって、現状の業務そのものをより客観的な視点から整理し、従来は組織内の暗黙知として存在していたものを顕在化し、より具体的に議論ができる形式に変えることができる。特に、部門間での業務連携をデザインする際に、この情報はきわめて有効である。また、各業務で利用する情報について、それぞれの情報の管理主体を明らかにし、データ生成、修正、保管などのライフサイクル管理を、より効率的に行うことができる。

#### ( 3 ) 製造業の情報システム設計者

製造業の情報システム設計者にとって、まず、自社の業務の中で、どの部分が情報システムとして新たな開発が必要であり、どの部分はその優先度が低いかなどについて、ビジネスプロセス全体の流れとバランスを考慮した意思決定を行うことができる。また、システム開発にあたり、ベスト・オブ・ブリード(適材適所)の IT パッケージを選択し評価する場合に、パッケージがもつ機能と、自社が要求する機能との適合性の評価を、非常に詳細なレベルで行うことができる。

#### ( 4 ) IT / 業務コンサルタント

製造業に特化した業務コンサルタントや IT コンサルタントにとって、まず、クライアントである製造業との会話をスタートさせるための共通の土俵を獲得する。共通の用語の設定や、業務アクティビティとしての標準形を示すことで、その企業に特有の具体的な内容への議論が円滑に行えるようになる。また、複数企業を対象として業務診断をおこなう場合



や、ベンチマークを行う場合に、概念的かつ定性的な評価のみではなく、より内部構造に踏み込んだ定量的な解析も合わせて可能となる。

#### (5) システムインテグレータ

製造業がすでにもつレガシーシステムを含む複数の部分的な情報システムや IT パッケージを統合し、企業の業務の中で機能させることを生業とするシステムインテグレータにとって、異なるシステム間の相互接続のための作業を、飛躍的に削減させることが可能となる。システム間の相互接続性は、PSLX コンソーシアムの最大の技術課題であり、そのためのあらゆる工夫が仕様として提示されている。システムインテグレーションビジネスが、今までのような労働集約型から、知識集約型でより付加価値の高いものとなる。

#### (6) IT パッケージベンダー

最後に、各業務アプリケーションに特化し、その分野でのベスト・オブ・ブリードである IT パッケージベンダーにとっては、PSLX 仕様のインタフェースを追加するだけで、接続可能な他のアプリケーション数が飛躍的に増加し、ビジネスが急拡大する。従来のような、外部インタフェースを個別に作りこむ方式では、ビジネスとして成立しなかった中小製造業が、新たなマーケットとして追加される。

以下では、上記のステークホルダーが、本仕様を、具体的にどのような手順で利用するかについて、いくつかの代表的な例を挙げて説明する。

---

## ビジネスプロセスの再設計

---

PSLX 仕様の中で、製造業の計画およびスケジューリングに関する意思決定の階層構造モデルを参照し、現状の基幹業務における意思決定の流れを根本から再検討する場合について考える。ここでは、需給調整のための基本戦略にしたがい、その企業にあったビジネスプロセスの再設計と、それに伴う意思決定モジュールの再配置を行うものとする。これには、PSLX 仕様とは別に提供されるベストプラクティスが参考となる場合があるかもしれない。

特に、計画の粒度と、計画期間および計画サイクルといった時間軸との対応関係を、いかに設定するかは極めて重要である。そして、計画の粒度が細くなるにつれて、いかにしてスケジューリング情報と連携をとるかについての基本設計もしておかなければならない。これは、必ずしも情報システム上でのフォーマルなデータ交換のみを指すのではなく、むしろ人間系での情報処理を中心として、いかに情報システムとの間で精度の高い情報をやりとりするかについて議論しておかなければならない。

設計されたビジネスプロセスは、具体的な業務アクティビティの中のコースケースあるいはコラボレーションとして明確に記述する。そして、これらの業務アクティビティをビジネスプロセスという形でつなぎ、企業全体としての一連の流れを設計する。ただし、この際、現状のモデル (AS-IS) と、目指すべきモデル (TO-BE) とを別途描き、その中間地点にある現実的で短期間に実装可能なビジネスプロセスの案を、当面のシステム開発の目的とすべきである。そして、将来にわたって、段階的にシステムを拡充させていくという段階的な拡張のシナリオも合わせて作成する。

PSLX が提唱する APS の大きな特徴として、スケジューリングレベルの業務アプリケーション間連携があげられる。これは、生産現場のあらゆるアクションを統括するスケジューラーが、上位のビジネス部門で認識しているより粒度の荒いスケジュールや、下位の装置レベルのよりきめ細かなスケジュールと連携して意思決定をおこなうしくみを含む。特に、ビジネス部門で行う集中処理と、各工場の生産現場で行う分散処理との連携のしくみは、それぞれの企業の業態や管理レベルによってバラエティが多く、多くの専門知識を必要とする部分である。

このような、ビジネスプロセスと、製造プロセスあるいはエンジニアリングプロセスとが密接に関係している領域においては、それらをあえて分離して設計するのではなく、一体として検討しなければならない。PSLX の APS ドメインモデルを利用して、ビジネスプロセス上で扱う情報のコンテンツを規定することは、すなわち物理的な製造プロセスや制約を規定することと同義である。したがって、PSLX のよりアドバンスな

適用として、この作業を進めることで、従来は非常に複雑でシステム化が困難であったこの複合領域を、IT の適用対象として取り込むことができる。

---

## システムの特徴分類と整理

---

製造業の情報システムの複雑さは、その企業の情報システム部門が単独で扱えるレベルを超えており、さらに情報システム開発を専業とする外部の企業であっても、一社ですべてをまかなうことは不可能となりつつある。必然的に、ベスト・オブ・ブリード（適材適所）なソフトウェアを必要に応じて組み合わせて利用することになる。この際に問題となるのが、ソフトウェアの詳細の仕様あるいは機能が、ライセンスを購入し、時間と費用をかけて導入した後でなければ検証できない場合が多いことである。ソフトウェアの機能を具体的かつ客観的に記述することはきわめて難しい。

PSLX の業務アクティビティの仕様を利用することで、ソフトウェアシステムの特徴分類と整理（プロファイリング）が可能となる。これは、製造業側が、IT パッケージベンダーあるいは SI 企業に対して要求仕様を作成する際、あるいは逆に、IT パッケージベンダーが自社の製品のプロモーションを行う場合や、SI 企業がクライアントに対してシステム提案を行う場合などに非常に有効な表現手段となる。具体的には、以下のような手順となる。

まず、PSLX で定義されたすべての業務アクティビティの中で、該当するものをマークする。通常、マークされたエリアは、同一の業務区分、あるいは同一の機能区分として連続している。業務アクティビティとして、このような連続性のない独立した複数の範囲をカバーするソフトウェアは、連続した領域となるように分割して議論するか、そうでなければここでは議論の対象としない。

続いて、該当する各業務アクティビティに対し、そこで仕様としてすでに登録されているユースケースのうち、対応しているものをチェックす

る。もし、表現したい内容に対応するユースケースが存在しない場合には、別途、新たなユースケースとして同様のフォーマットで記述する(必要に応じて、あらたな PSLX 仕様として登録申請することも可能である)。また、コラボレーションについても、同様に対応するものをチェックする。

もし、コラボレーションに関係している業務アクティビティのすべてがシステムの対象としてマークされておらず、その一部を他の業務アプリケーションに依存している場合には、コラボレーション内部にあるシステムの境界におけるインタフェース仕様も、機能の一部として記述される必要がある。また、同一の業務アクティビティが、分散環境において物理的に複数の異なるシステムによって実現される可能性がある場合も同様に、異なるシステム間のインタフェース仕様が重要となる。

---

## 業務データの整理と整合性管理

---

多くの IT プロジェクトにおいて、個別要望に応じてソフトウェアをカスタマイズする作業以上に、労力と期間を必要とするのが、システムの実行に必要な業務データの整備である。PSLX 仕様を用いて、ソフトウェアの機能が具体的な業務アクティビティとして明らかになる過程において、そのシステムを実行するために必要となる具体的なデータ項目とその所管部門が明らかになる。したがって、実際のシステム開発に先立って、必要となるデータ生成および加工のための準備が進められる。

新しいビジネスプロセスを、企業の戦略的な強みとして実装するにあたり、あらかじめ必要なデータが継続的に低コストで得られるかどうかの見極めは非常に重要な要素である。また、新規にまったく新しいシステムを一式導入する場合でも、そこで処理されるデータは、現状のビジネスで用いられ、現実にもそこで動いているデータが形式を変えて対応せざるを得ない。したがって、既存データの移行計画が、システム開発を進める上で極めて重要な要素となる。これらは、人間活動を含んだ情報システムを考えた場合、すべてが業務データを中心として成り立っているという開発思想にもとづいている。

レガシーシステムの解析(リバースエンジニアリング)を例にとり,PSLX仕様の利用例を説明しよう。まず,システムの特徴分類と整理の節で示した方法で,レガシーシステムの機能と具体的な振る舞いを,業務アクティビティおよびそこに含まれるユースケース,コラボレーション等で詳細に記述する。続いて,それぞれのユースケースあるいはコラボレーションにおける情報交換において必要となる情報を,APS ドメインモデルを用いて記述する。ここでは,データの具体的な中身ではなく,形式,つまりクラスのみを明らかにする。

通常のリバースエンジニアリングによって,既存のレガシーシステムのRDBスキーマの内容を個々に解析することは非常に期間と労力を費やす過酷な作業である。多くの場合,それらのスキーマはエレガントな設計にはなっておらず,システム改変の経緯から意味のないフィールドが数多く歪な形状をしているため,あまり生産的な作業ではない。PSLXでは,レガシーシステムがもつ外部仕様から意味を抽出し,同等な機能を実現するための概念データモデルを,よりエレガントな形で得ることができる。

業務に即した概念データモデルができた後は,対応する共通データスキーマを設定し,そこに実際のデータを充填するだけである。一定期間は二重入力することも必要かも知れないし,何らかのデータ移管ツールが別途ベンダーから提供される場合もあるだろう。場合によっては独自に開発してもよい。多くのシステム開発では,新しいシステムを新規に作るほうが向かいやすいが,実際には現状のシステムからその新しいシステムへの移管のプロセスをいかに行うかに費やす労力のほうが極めて大きい場合がしばしばある。

PSLX仕様は,業務データの整合性の管理という点でも,非常に有効なツールとなる。たとえば,BOM(部品表)の管理がそのよい例である。通常,BOMには,設計BOM,製造BOM,販売BOM,そして製品のライフサイクル管理に用いられるサービスBOMなどがある。これらの情報は,それぞれが異なるRDB上に分散して存在するケースが多いが,そこに存在するデータの一部は,技術的に同じものを表現しており,整合性が維持されていない場合がある。

業務データを分散的に配置し、自律的な管理を行う場合に、このような整合性の管理は難しく、これが理由でやむなく集中処理型のクライアント・サーバ方式をとっている場合が少なくない。PSLX仕様は、このような問題を解決するために、分散した各RDBの各テーブルまたはフィールドの関係を、APSドメインモデルあるいはオントロジーのレベルまでさかのぼって対応づける。そして、あらかじめ、それらのデータ間の整合性の維持のためのルールを定義しておく。

たとえば、同一のデータが2箇所に書かれており、ともに独自に書き換え可能である場合、これらのシステム間の連携の前提として、値が異なる場合にどちらを優先させるか、ネゴシエーションのプロセスをどうするか、あるいはどのようなタイミングで同期を取るかなどを別途記述する。これらの付加的な情報は、PSLX仕様によって、表面的に異なる実装上のデータ内容が、意味的に同じであることを明示的に記述可能であることではじめて実現できる。

---

## 実装 RDB スキーマ設計支援

---

PSLXが提唱する情報システムアーキテクチャーは、集中処理型というよりは、むしろ自律分散型である。データを一箇所に集中して管理する方法は、汎用機の時代から、巨大なデータサーバ・マシンによるクライアント・サーバ方式となった現時点まで、比較的長い歴史がある。一方、ミニDBによる自律分散型のアーキテクチャーは、比較的新しい概念であり、まだ全体として主流にはなっていない。しかし、この方式の最大の強みは、例外の多いきめ細かなデータを正規に管理できる点と、データスキーマの部分的な改変を、ビジネスプロセスへの影響を最小限にとどめながら比較的自由に行うことができる点にある。

しかし、一方で、従来ならばデータベース設計は特殊技術として一部のシステムエンジニアのみが習得すればよかったのに対して、この技術をより多くの関係者が習得する必要がでてきた。個々のRDBシステム固有の実装技術はともかくとして、個別の業務アクティビティがもつべき個

別の情報を過不足なく表現でき、しかも業務アクティビティの具体的なロジックから見てもある程度効率的といえるスキーマを設計することは非常に大変な作業である。

PSLX 仕様として提供される RDB 標準スキーマは、個々の企業の個々の業務に合った個別の実装 RDB スキーマを設計するためのテンプレートである。情報システムの設計者は、以下のような非常に簡易な手順によって、非常に高度なデータベース設計者と同等レベルの実装 RDB スキーマを設計可能となる。

まず、業務アクティビティに含まれる各ユースケースが必要とする RDB 標準スキーマのクラスを仕様にしながらって選択する。また、仕様に含まれないユースケースがある場合には、そこで必要となる情報を最小数のクラスによってカバーできるように別途クラスを選択する。そして、これらをすべて実装 RDB のビューとして定義する。

RDB 標準スキーマのクラス(ビュー)には、最低限の属性(フィールド)のみが仕様としてあらかじめ設定されている。これに対して、実際には、企業の業務固有の膨大な種類の情報が存在している。この企業の業務固有の情報は、まず、APS ドメインモデル上でその位置づけを明確にした後に、RDB 標準スキーマ上のクラスの属性として再定義する。これにより、その情報の意味が明確となり、また複数の RDB 標準スキーマのクラスへ同一の情報を設定することも可能となる。

続いての処理は、新規にデータベースを設計する場合と、既存の RDB を活用する場合とで展開が異なる。はじめに、まったく新規に設計する場合には、まず、RDB 標準スキーマとして選択されたクラス、および追加されたクラスに関するすべての APS ドメインモデルのクラスを列挙する。そしてこの APS ドメインモデルに相当するテーブルを実装 RDB スキーマとして設定する。さらに、クラス間の数的関係のみを、必要に応じて関係を表すテーブルも設定する。ただし、RDB 標準スキーマの関係構造クラスに分類されるものがある場合には、そのまま対応する関係を定義する。

このようにして、新規に開発する場合には、実装 RDB スキーマは、APS

ドメインモデルをベースに、スキーマ変換処理を行うことで生成する。さらにテーブルの連結構造をシンプルにして検索効率を向上させるために、必要に応じて属性(フィールド)の移動や複製を行うことも可能である。そして最後に、各属性(フィールド)のデータタイプを設定する。完成した実装 RDB スキーマは、SQL 文によって、RDB 標準スキーマを動的に生成できるようにしなければならない。なお、ベースとするスキーマを APS ドメインモデルとせず、オントロジーから直接生成することも可能である。

一方、すでに実装 RDB スキーマが存在する場合には、現状の実装 RDB スキーマをベースとし、RDB 標準スキーマをそのビューとして関係づける必要がある。具体的な手順としては、まず実在する実装 RDB スキーマを APS ドメインモデル上にマッピングする。そして、その上で、新規開発の場合と同様に、選択された RDB 標準スキーマのクラスに対応する APS ドメインモデルを加えたものを列挙し、既存の実装 RDB スキーマに存在しないデータを明らかにする。もしデータ項目が、既存のクラスの一属性である場合には、該当する属性を追加し、新たなクラスの一部である場合には、そのクラスを実装 RDB スキーマに新規に追加する。

現実には、最終的な実装 RDB スキーマが得られるまでに、さらに緻密で細かな処理を必要とするが、ここでは詳細の説明は省略する。いずれにしても、基本的なアイデアとして共通しているのは、RDB 標準スキーマを、実装 RDB へ外部からアクセスするための共通なビューとして位置づけている点である。実装 RDB 内部では、比較的自由にスキーマを設計してよい反面、共通仕様となっている RDB 標準スキーマを、ビューとして実装 RDB のテーブルと責任をもって関係づけなければならない。

---

## システムインテグレーション

---

システムインテグレーションという作業は、情報システム構築における最終的な目標であると同時に、最大の難関でもある。これは、単にプログラム A からプログラム B へ情報を正しく送信できることのみが目的ではない。プログラム A とプログラム B とが、共通の DB の共通のデータ



にアクセスできることも、全体の目的のほんの一部である。システムインテグレーションという場合には、少なくとも、異なる2つのシステムを、ビジネスプロセスとして、一体となつてある共通の取り決めのもとで全体として機能させることが重要となる。

ただし、現実には、最初の2つですら実現するのは難しい。プログラムAがプログラムBと情報交換する場合、両者が同じ設計者のもと、あらかじめ情報交換できるように設計されていた場合には問題はない。しかし、たとえば、それぞれが異なる設計者のもと、あらかじめ相手の存在を知らない状況でシステムがすでに実装されている場合にはどうだろうか。また、2つの異なるプログラムが共通して一つのDBにアクセスする場合についても、そのDBのスキーマの詳細が不明である場合や、アクセスしたものの、そのデータ項目の正しい意味が不明である場合など、現実には難しい問題が多い。

まず、情報の意味の問題は、極めて奥の深い問題であるが、PSLXではこの分野のオントロジーを導入することでこの問題に対処している。特に、生産現場で使用される用語は、業種、業態の違い、企業の文化によって異なり、そして企業内部でも、担当者ごとでも異なる場合がある。それぞれの用語を、オントロジーを用いて表現しなおすことで、意味のあいまい性が排除できる。また、オントロジーの代わりに、APSドメインモデルにある各クラスの名称を標準用語として辞書用に利用するのも、PSLX仕様の効果的な利用方法である。

その時点では未知のシステムとの間で、将来起こるであろう情報交換のためのしくみを実装することは基本的に不可能である。このような場合には、あらかじめ情報交換のためのスキーマを定義しておき、すでに存在するシステムも、これから新たに開発するシステムも、すべてこの共通のスキーマに合わせたインタフェースを用意する、というアプローチをとる必要がある。PSLXのRDB標準スキーマも、XML標準スキーマも、いずれもこれに該当する。これらの標準スキーマの仕様に準拠したプログラムは、基本的にカスタマイズなしで、複数のシステム間の情報交換が可能となる。

情報システムを開発する場合、あるいはITパッケージを開発する場合、

外部システムとのインタフェース設計を行う必要がある。この際に、PSLX 仕様を利用して、その基本要件を洗い出すことができる。まず、業務アクティビティによってそのシステムの範囲と内容を明らかにし、その上で、外部とのコラボレーションの形態をまず宣言する。続いて、そのコラボレーションに必要なインタフェースの形態を、メッセージ型である XML による方式と、データ蓄積型である RDB による方式から選択し、それぞれ該当する標準スキーマを選択する。そして、実装したその標準スキーマのクラス名を、ファサード（インタフェースの形態）として公開し、外部からの未知のアクセスに備える。

個々の業務アクティビティのユースケースに対応した業務ソフトウェアに対して、自分からアクセスする機能のみをもったシステム（つまり外部からのアクセスを受けない）は、RDB 標準スキーマ、あるいは XML 標準スキーマ上のデータの存在を前提として自分自身のロジックを組むだけでよくなる。このしくみを利用すると、データベースを持たない、計算アルゴリズムに特化したソフトウェアや、特殊な管理ノウハウをブラックボックス化したソフトウェアなどが容易に開発可能となる。逆に、製造業の側からは、それらの多様な外部知識を容易に業務の一部に組み込むことが可能となる。

## 第六章

# 仕様準拠と拡張方法

---

### 標準スキーマの個別拡張方法

---

情報システムの個別の実装にあたり、特に共通スキーマである2つの標準スキーマの仕様が、実装にもっとも近い仕様となっている。多様な個別の企業の複雑な情報構造を、これらの標準スキーマがすべて包括することは不可能である。基本的に、PSLX で規定している標準スキーマは、標準として設定されていない項目に対して、その他の情報という形で、仕様の範囲内で自由に追加することが可能であるが、この部分の内容は、データ交換相手に正確に伝わる保障はない。

そこで、PSLX 仕様は、すでに規定された標準スキーマに関する規約をベースに、個々の業種や業務に特化した仕様を拡張仕様として規定することを許している。たとえば、自動車業界用の標準スキーマや、化学プラント向けの標準スキーマなどである。これらの個別拡張仕様は、その内容に対して、PSLX コンソーシアムは特別な関与はせず、認証等のサービ

スも行わない。拡張仕様の提案者は、PSLX コンソーシアムのメンバーである必要があるが、拡張仕様の内容は審査されず、申請のための必要事項を満たしていれば自動的に登録される。

PSLX 拡張仕様の策定は、PSLX コンソーシアムのプロジェクト、PSLX コンソーシアムとリエゾンをもつ他の標準化団体、あるいは PSLX コンソーシアムの会員企業が推進する。拡張仕様の知的所有権は、原則として PSLX 仕様の扱いに順ずるものとする。つまり、原則としてメンバー全員に無償で公開される。

RDB 標準スキーマの拡張仕様は、具体的には以下の 3 種類のパターンがある。

#### (1) トップレベルのクラス (ビュー) の追加

RDB 標準スキーマのクラスは個々に独立しており、すべてがトップレベルである。これらは実装 RDB のビューに相当する。業務アクティビティの各ユースケースで必要な情報は、すべてこのクラスに存在しなければならない。できれば、ユースケースにおけるひとつのアクションで必要となる情報は、ひとつのクラスであることが望ましい。したがって、もし複数のクラスの情報を同時に考慮する必要があり不便である場合には、別途そのユースケース用に新しくクラスを定義することができる。

#### (2) 既存のトップレベルクラスへの属性の追加

業務アクティビティのユースケースで利用する情報の一部が、そのユースケースの他の情報が存在する RDB 標準スキーマのクラスに存在しない場合、仮に別の RDB 標準スキーマのクラスに存在していたとしても、業務アクティビティの立場から、同一のクラスに情報が存在することが望ましいと考えられる場合には、そのクラスに対応する新たな属性を追加することができる。属性があらたに追加されたクラスは、PSLX の標準クラスとは別のクラス名で管理される。

#### (3) 属性のデータタイプに関する制約の追加

PSLX 仕様における RDB 標準スキーマに属する各クラスの属性は、データタイプが設定されていない。つまり、仕様としては、あらゆるタイプのデータタイプを受け付ける。しかし、実際には、あらかじめデータタ

イブを規定しておかないと、実装RDBへの接続ができない。このため、拡張仕様として、あらかじめ各属性に対するデータタイプを設定するか、データタイプに対する制約を規定することができる。

XML 標準スキーマの拡張仕様は、具体的には以下の3種類のパターンがある。

(1) トップレベルのクラス(タグ要素)の追加

XML 標準スキーマのトップレベルのクラスは、業務アクティビティ間で交換される XML メッセージにおけるトップレベルの要素タグに相当する。これは、業務アクティビティ間のコラボレーションにおいて必要となるデータ交換トランザクションをクラス化したものである。したがって、新しいコラボレーションを定義した場合、あるいは既存のコラボレーションの内容をカスタマイズした場合などに、合わせてメッセージの内容にあたる XML 標準スキーマのトップレベルのクラスを定義しなければならない場合がある。

(2) 下位のクラスの属性値のボキャブラリーの追加

PSLX 標準 XML の仕様としては、各クラスの属性が取り得る値の範囲が、特に文字列の場合について一部分のみが規定されている。ここで設定される文字列は PSLX 仕様における APS ドメインモデルと対応づけて定義されるものであり、業種や業態等の違いによって APS ドメインモデルを拡張した場合には、それに応じて属性値としてのボキャブラリーが増加する。したがって、拡張仕様として、これらの追加されたボキャブラリーとそれに対する業務ロジック上の制約等をあわせて規定することができる。

(3) 下位のクラスの構造や属性値に対する制約の追加

XML 標準スキーマは、UML および XML スキーマによって記述されている。XML スキーマによる記述では、あるクラス(要素)が複数の上位クラス(要素)を持つ場合に、上位クラスが何であるかによって異なる制約を与えることが可能である。たとえば、オーダというクラス(要素)が「新規顧客オーダ」という上位クラスの下位で定義されている場合と、「作業オーダ指示」という上位クラスの下位で定義されている場合とでは、その内容が異なる。このような下位クラスの構造や属性値に関する

制約を、トップレベルの既存のクラスあるいは新規クラスに対して定義することができる。

---

## PSLX 仕様拡張の基本方針

---

個別に拡張された仕様は、正式な PSLX 仕様とは区別して管理されるが、それらの情報は、将来、新しい PSLX 仕様として組み込まれる可能性が高い。また、個別拡張として許されていなかった部分に対しても、今後、必要に応じて PSLX 仕様は定期的に見直される予定である。

製造業には、多種多様な業種があり、さまざまなタイプの企業がさまざまなビジネスモデルのもとでビジネスを行っている。そもそも、そこで行われている情報処理は、とうてい標準という一語で対処することは不可能である。また、標準化が、逆に個別のシステムの成長や発展を阻害し、健全な競争にもブレーキをかける危険性も認識しておかなければならない。

システム実装のための基本仕様を作成するにあたり、PSLX コンソーシアムでは、以下のような基本姿勢をとる。これらは、今後の仕様の改定時にも引き続き継承される考え方である。

- (1) 業務アクティビティ間の情報交換に必要最低限の内容を規定する。
- (2) 業務アクティビティ内の業務の詳細(内部仕様)は規定しない。
- (3) 標準項目に該当しない内容も仕様の一部としてラフな記述を可能とする。
- (4) 個別の業種やビジネスにあわせた仕様拡張を可能とする。
- (5) 通信などの共通な部分は他団体の仕様を採用しバインド方式のみ規定する。

## 仕様の標準化プロセス

PSLX コンソーシアムでは、APS 実現のための具体的なさまざまな仕様を、コンソーシアム勧告という形で公開する。仕様の勧告にいたるステップは、おおまかには以下の手順をとる。

- ( 1 ) 技術委員会における標準化項目の承認
- ( 2 ) 必要に応じて分科会を新規に設定
- ( 3 ) 分科会にて仕様の内容を議論しドラフト作成
- ( 4 ) ドラフトを技術委員会にて審議し承認
- ( 5 ) ドラフトの勧告案としての公開
- ( 6 ) 一般からの意見受付と部分的修正
- ( 7 ) 勧告案を理事会にて審議し承認
- ( 8 ) 最終仕様を勧告仕様として登録し一般公開

これらのプロセスは、すべてWeb上で公開されており、技術委員会、分科会ともあらゆる関係者が議論に参加可能である。また、勧告仕様となる前の勧告案は、PSLX コンソーシアムの会員はもちろんのこと、それ以外の多くの関係企業、関係団体等にPRし、仕様の内容についてのコメントを積極的に収集する。最終的に勧告仕様として公開された仕様は、さらに配布可能な形で整備し、セミナーやチュートリアル等によって、より多くの関係者に対して宣伝普及活動を行う。

仕様の策定にあたっては、特に他の標準化団体との連携が重要である。たとえば、現在 PSLX コンソーシアムは、XML 標準スキーマに関する仕様の策定を、OASIS の生産計画スケジューリング技術委員会とのリエゾンの中で行っている。このケースでは、まず仕様詳細決定を OASIS にて行い、そこでの内容を PSLX コンソーシアムに持ち帰り改めて審議するという方法をとっている。また、今後、国際標準化へのステップとして、IEC / TC64A / JWG15 等との連携も行っていく。

## 仕様準拠の判定手順

情報システムや IT パッケージなどのソフトウェア製品が、PSLX 仕様に準拠しているかどうかの判定方法は、それぞれの仕様書に詳細が記述される予定である。仕様準拠の判定は、各情報システムの所有者あるいは関係者が独自に行う場合と、PSLX コンソーシアムの技術委員会が行う場合の 2 種類がある。PSLX コンソーシアムの会員であるかどうかに関わらず、情報システムの所有者あるいは関係者は、独自に仕様準拠の判定を行うことができるが、PSLX 仕様準拠を対外的にアナウンスする場合には、PSLX コンソーシアムが定める特定の手続きをとる必要がある。

以下に PSLX 仕様に対する準拠判定のためのカテゴリーを説明する。これらの各カテゴリー別に、PSLX 準拠を宣言することができる。

### (1) 情報システムアーキテクチャー審査

ここでは、それぞれの企業の情報システムアーキテクチャーが、PSLX が定義する APS のための情報システムアーキテクチャーに準拠しており、そこで APS が、意思決定のしくみとしてきちんと機能しているかどうかを仕様と照らし合わせて判断する。準拠判定のための詳細は、APS システム認定審査仕様として定められている。

### (2) 業務アクティビティ情報統合審査

ここでは、生産現場における業務アクティビティが正しく定義されており、そこで必要となるデータの内容、およびデータ交換のトランザクションが明確に定義されているかどうかを審査する。ここでは、人間系の処理の中で扱っている情報もすべて対象となる。準拠判定のための詳細は、業務アクティビティ仕様に規定されている。

### (3) 業務データのオントロジー適合審査

生産現場での業務アクティビティ、あるいはより上位の管理部門において利用されている生産計画やスケジューリングに関するあらゆる情報が、APS ドメインモデルあるいはオントロジーによってその意味が正しく定義されているかを判定する。準拠判定のための詳細は、PSLX オントロジー仕様あるいは APS ドメインモデル仕様に規定されている。



( 4 ) RDB 標準スキーマ接続性審査

実装 RDB スキーマが、RDB 標準スキーマに含まれる特定のクラスに対応するビューを正しく提供しているかを判定する。また、業務アプリケーションが、RDB 標準スキーマの特定のクラスに対応するビューを通してデータにアクセスできることを判定する。準拠判定のための詳細は、RDB 標準スキーマ仕様に規定されている。

( 5 ) XML 標準スキーマ接続性審査

業務アプリケーションが、XML 標準スキーマにクラスとして定義された特定のメッセージを正しく送信、または受信でき、その内容を正しく理解した上で、あらかじめ設定した通信相手とのコラボレーションを実現できるかどうかを判定する。準拠判定のための詳細は、XML 標準スキーマ仕様に規定されている。



PSLX ホワイトペーパー (p.76)

## 参考文献

- (1) PSLX Technical Specification 00, Guidance, PSLX Consortium, 2003, <http://www.PSLX.org/en/>
- (2) PSLX Technical Specification 01, Grand Design for Manufacturing Industry, PSLX Consortium, 2003, <http://www.PSLX.org/en/>
- (3) PSLX Technical Specification 02, APS Agent Model, PSLX Consortium, 2003, <http://www.PSLX.org/en/>
- (4) PSLX Technical Specification 03, PSLX Domain Objects, PSLX Consortium, 2003, <http://www.PSLX.org/en/>
- (5) PSLX Technical Specification 04, XML Specification for PSLX, PSLX Consortium, 2003, <http://www.PSLX.org/en/>
- (6) PSLX 技術仕様書 05, PSLX用語辞書, PSLX Consortium, 2003, <http://www.PSLX.org/en/>
- (7) PPS (Production Planning and Scheduling) Part 1: Core Elements Working Draft 01, OASIS Production Planning and Scheduling TC, 2004, <http://www.oaSIs-open.org/>
- (8) MESX ホワイトペーパー, MESX ジョイントワーキンググループ, 2004
- (9) ANSI/ISA-95.00.01-2000 - Enterprise-Control System

- Integration Part 1: Models and Terminology, 2000,  
<http://www.isa.org/>
- (10) ANSI/ISA-95.00.02-2001 - Enterprise-Control System  
Integration Part 2: Object Model Attributes, 2001,  
<http://www.isa.org/>
- (11) ISA-95 Enterprise-Control System Integration Part3 (draft 19),  
Activity Models of Manufacturing Operations Management,  
2004, <http://www.isa.org/>
- (12) APICS Dictionary 11<sup>th</sup> edition, APICS, 2004,  
<http://www.apics.org/>
- (13) MESA White paper number 2: MES Functionalities & MRP to  
MES Data Flow Possibilities, MESA International, 1997,  
<http://www.mesa.org/>
- (14) Supply-Chain Operations Reference Model ver.6.1, Supply  
Chain Council, <http://www.supply-chain.org/>
- (15) Open Application Group Integration Specification Release.8.0,  
Open Application Group, Inc.,  
<http://www.openapplications.org/>
- (16) Business To Manufacturing Markup Language (B2MML)  
version 02, World Batch Forum,  
<http://www.wbf.org/>
- (17) UN/CEFACT, UN/CEFACT Modeling Methodology (UMM),  
CEFACT/TMWG/N093, 2003, <http://www.unece.org/cefact/>
- (18) A Reference Model For Computer Integrated Manufacturing  
(CIM), ISA, 1989, [http://www.pera.net/Pera/  
PurdueReferenceModel/ReferenceModel.html](http://www.pera.net/Pera/PurdueReferenceModel/ReferenceModel.html)
- (19) Process Specification Language: ISO 18629 Part 1, ISO, 2004,  
<http://www.iso.org/>
- (20) GERAM: Generalised Enterprise Reference Architecture and  
Methodology, IFIP-IFAC Task Force, 1999,  
<http://www.cIT.gu.edu.au/~bernus/taskforce/geram/verSIons/geram1-6-3/v1.6.3.html>