



ホワイトペーパー 2

製造現場の情報連携を実現する PSLX プラットフォーム計画

2008 年 6 月

PSLX フォーラム

NPO 法人ものづくり APS 推進機構

目 次

第一部 マネジメント編

1.	はじめに.....	1
2.	工程スケジューラの現状.....	1
3.	製造現場マネジメントの課題.....	2
4.	製造現場は標準化できるか?.....	2
5.	ベストオブブリードの世界.....	3
6.	PSLXプラットフォームとは.....	3
7.	RDBの功罪.....	4
8.	分散データベースとメッセージング.....	4
9.	データベースの構造とメッセージの構造の違い.....	6

第二部 エンジニアリング編

10.	メッセージングに関する標準.....	6
11.	SOAとPSLX標準仕様.....	7
12.	メッセージの基本構造.....	7
13.	メッセージの伝達方法.....	8
14.	個別性と拡張性のための仕組み.....	9
15.	PSLXプラットフォームの基本仕様.....	10
16.	プラットフォームの管理.....	10

第三部 システム構築編

17.	PSLX対応アプリケーションとは.....	12
18.	アプリケーションプロファイル対応.....	13
19.	共通ソフトウェアコンポーネント.....	13
20.	サーバの実装方法.....	14
21.	PSLX対応アプリケーションの自作.....	14
22.	PSLXが生み出す世界.....	16
23.	おわりに.....	16

第一部 マネジメント編

1. はじめに

2008年1月に PSLX 技術仕様バージョン 2 が完成しました。しかし、これはあくまで、仕様書であり、これによって、ただちに製造業が変わるわけではありません。図に示すとおり、これはあくまで、製造業のあるべき情報化を行うのに必要な道具のうち、片側の車輪が完成したのであって、もう一方の車輪である共通的な道具や環境の整備をただちに行う必要があります。

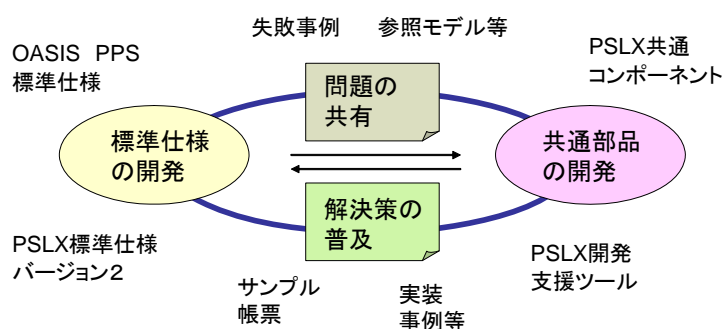


図1 プラットフォーム構築のための要素

本稿では、製造業の製造現場を中心とした情報化を実現するための知的情報プラットフォームを構築するために、PSLX フォーラムが、次に取り組むべき課題を明らかにし、その実現へむけての具体的な方策を提示します。

2. 工程スケジューラの現状

製造現場は、例外事項のかたまりのようなもので、工程進捗を統括する担当者は、なか長年の勘と経験でやりくりしながら現場をやりくりします。10年くらい前から、スケジューラのソフトウェアが実用的なレベルとなり、現場の管理の一部はスケジューラの支援が得られるようになってきましたが、まだそのカバーできる範囲は限られています。

工程スケジューラが製造現場で十分に活用されていない理由は、多くの場合、大きく2つの状況が考えられます。第一が、スケジュール作成時点での実際の生産ラインの状況（進捗）を、きちんと反映していない場合です。そして、第二が、マスタがきちんとメンテナンスされていない場合です。

この2つの問題はともに、スケジューラ単独では解決することができません。本当にスケジューラが機能し、製造現場が常に機動的に振舞うためには、複数のアプリケーションプ

プログラムが、お互いに持っている情報を交換しあうことで、情報連携を実現することが必須となっています。そして、これは PSLX の設立当初から依然として変らぬ目標でもあります。現在、その解決策に手がとどくところまでできています。PSLX プラットフォームは、これを安価で短期間に実現します。

3. 製造現場マネジメントの課題

工程スケジューラの主な役割は、生産のためのスケジュールを作成し、その進捗を管理することです。しかし、製造現場の活動には、生産以外にもさまざまな種類の活動が存在しており、それらが相互に密接に関係しています。生産管理以外の活動としては、以下の図のように、在庫管理、品質管理、保守管理があげられます。製造現場では、これらの個々の活動が全体として連携していなければなりません。

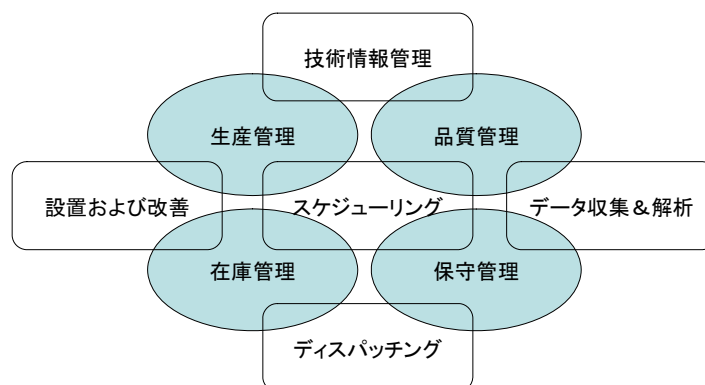


図2 製造現場のさまざまな活動

個々の活動に対する作業指示は、相互に関係しています。個々の活動で利用する技術情報も相互に関係しています。それぞれの活動が他の活動の前提条件や結果に大きく影響を与えます。製造現場は、常に動いており、そのあり様は、生き物のように変化し続けています。これを、従来からある一般的な IT システム構築のためのアプローチで対応するには、そもそも無理があると我々は考えています。

4. 製造現場は標準化できるか？

生産現場は、知識やノウハウが集積された場所であり、日々それらを拡大再生産することで付加価値を生み出しています。どの生産現場も、どれひとつ同じ形で表現することはできません。個々の部品や要素作業は標準化できても、その組合せとなるシステム全体を標準としてとらえることには無理があります。我々が標準として定義したいのは、生産現場そのものではなく、生産現場で交換される情報の表現方法です。

複数の工程間、あるいは部門間でそれぞれの活動を同期させ連携させる場合に、なんらか

の情報交換を行います。現場で飛び交う作業指示や帳票類は、情報交換の一形態です。製造現場のさまざまな情報機器や制御機器では、デジタル情報、アナログ情報が交換され、同時に、機器と作業者との間で情報交換が行われています。人間の意思決定により近い情報になるほど、その情報の意味を正しく定義することが必要となります。計算機でも人間でも、理解可能な XML 記述が注目される理由もここにあります。

5. ベストオブブリードの世界

これからの IT 業界は、これから徐々に知識集約型に移行し、ソフトウェアパッケージベンダーの役割がますます高まっていくと思われれます。各ベンダーは、より専門特化したソフトウェアを提供するようになり、ベストオブブリード（その分野で最高なソフトウェアを選択してインテグレーションする開発方法）が主流となっていくでしょう。インテグレーションのコストが安ければ、ユーザ企業は、ベストオブブリード型のシステム構築を望みます。

一方、パッケージベンダーは、ベストオブブリード型のビジネス環境で生きていくことで、経営資源を、自社の得意な分野に集中させることが可能となります。知識集約型の産業に移行するために必須となる研究開発により力を注ぐことが可能となります。インテグレーションや外部との連携は、業務アプリケーションに特化したパッケージベンダーにとっては、長い目で見た場合に、付加価値に結びつきません。

当然のことですが、パッケージベンダーにとって、ベストオブブリードの世界でビジネスを展開するためには、自社のパッケージを外部と容易に連携できるインタフェースを整えておかなければなりません。外部に公開するインタフェースはできるかぎり標準的にすることで潜在的なビジネスチャンスを広げ、内部に隠蔽するロジックはできるかぎり固有なものとする事で差別化を図るといふ戦略が求められます。

6. PSLXプラットフォームとは

通常、プラットフォームという用語は、IT の世界では、アプリケーションソフトウェアを動作させるための OS の種類や環境設定などを指します。しかし、すでに述べたように、人間系でのコミュニケーションを含む広い意味でのソフトウェアを実際に機能させるためには、計算機の OS といった従来の意味でのプラットフォームではなく、業務で利用されている情報を正しく解釈し相手に確実に伝えるための環境が必要となります。

具体的に言うと、最初にあげた例のように、工程スケジューラは、品目マスタがなければスケジュールを生成できません。工程スケジューラのためだけに毎回マスタをデータ入力することが非現実的な場合には、すでにその製造現場で利用されているマスタから必要な

情報を取り出すこととなります。その際、コード体系、データ項目、そして必要な情報の欠落など、情報の意味のレベルでのギャップをどうやって埋めるかが問題となります。

PSLX プラットフォームとは、このようなより高次のレベルでの製造現場の情報管理の基盤（インフラ）であるということが出来ます。そこでは、業務アプリケーションが情報連係を行う場合の情報の表現方法が、知識により近いレベルで標準化されており、その標準形式にもとづくさまざまな道具が提供されます。これによって、製造現場の情報システムのインテグレーションの工数は飛躍的に削減され、ベストオブブリードの世界が実現されます。

7. RDBの功罪

現在の基幹情報システムにとって、RDB 技術はなくてはならない存在であり、これを抜きにして情報システムは語れません。しかし、RDB がもつ機能が非常に魅力的であるがゆえに、これに対するさまざまな要求が爆発的に増え、実際に企業に導入される RDB システムは、非常に複雑で難解になり、一部の専門技術者でなければその内容をメンテナンスできなくなっています。

高度な信頼性と応答性が要求される全社的な基幹システムの場合と比較して、製造現場の場合には、必要な情報の構造が常に変化します。そして、それらの情報は、例外的なものが多く、あらかじめ設計されたスキーマに対応できず、RDB に登録できない情報が大半を占めています。結果的に、Excel 等の個人的な情報管理ツールでそれらの非定型な情報が管理され、効果的な情報共有や連携ができずにいます。

したがって、PSLX プラットフォームは、RDB の存在を前提としません。RDB は、プラットフォームに関与するひとつのアプリケーションソフトウェアとして存在し、そのデータ構造は完全に内部に隠蔽されます。つまり、RDB のデータスキーマは公開されず、代わりに、情報としてどのような構造を持っているのかが、標準的な表現形式によって論理的に記述されます。

8. 分散データベースとメッセージング

製造現場のさまざまな情報は、その情報が発生する場所、あるいはその情報を管理する場所に近いところに置くべきです。すべての情報を集中して配置する必要はありません。それぞれの情報は、意味をもっており、その所有者、管理責任者が存在します。たとえば、品質管理に関する情報は、品質管理部門が所有し、その部門の内部でデータ構造も含めて適宜更新されます。本社の情報システム部門に依頼する必要はありません。

その代わり、それぞれの情報を管理する部門は、外部からその情報の問い合わせに対し、責任をもって回答する必要があります。そして、さらに、その部門がどのような情報を管理しているのかを、あらかじめ外部に対して、周知しておく必要があります。言い換えれば、その部門が提供可能な情報サービスの種類と内容を、あらかじめ外部に公開しておかなければなりません。

業務を行う上で必要となるさまざまな情報は、その管理担当部門にその都度問い合わせを行い、その回答をもとに作業を進めます。必要に応じて、ローカルに情報を保持することも可能ですが、適宜その内容は更新しておかなければなりません。外部に対する問い合わせは、メッセージを用いて行います。これをメッセージングと呼びます。業務を行う主体がアプリケーションソフトウェアである場合、このメッセージングによって、情報の受け渡しが行われます。

プログラムは、RDBのスキーマ構造を知る必要はありません。その代わりに、メッセージングにおけるメッセージのフォーマットと送受信の手順をあらかじめ知っておく必要があります。これは、製造現場において、さまざまな帳票のフォーマットを知り、その情報を交換するためのワークフローを知っておく必要があるのと同じです。

第二部 エンジニアリング編

9. データベースの構造とメッセージの構造の違い

PSLX プラットフォームでは、データベース構造もメッセージ構造も重要です。データベースが不要であると言っているわけでは決してありません。情報交換する際に、データベースの構造を公開するのではなく、メッセージの構造を公開することでアプリケーション連携を実現するべきであるという点が重要です。

データベースの構造とメッセージの構造は、そもそも、なぜ異なっているのでしょうか？この違いは、データベースは情報を蓄積することが目的であり、メッセージは情報を伝達することが目的であることに起因します。情報を蓄積する場合に、その情報はさまざまな業務によって利用されます。一方、情報の伝達は、ある業務の中で行われており、その情報の利用形態あるいは情報の持つべき意味は特定できます。つまり、データベースの構造は、さまざまな利用形態を想定している分だけ、複雑にならざるを得ません。

一方、メッセージの構造は、データベースの構造と比較して、きわめてシンプルになりますが、その分、さまざまな種類のメッセージが必要となります。極端に言えば、業務の数だけ異なるメッセージ構造を準備しなければならないかもしれません。ただし、メッセージ構造（たとえば帳票のフォーマット）は、送信相手との間であらかじめ合意しておく必要があるため、業務を円滑に行うためには、その種類を無制限に増やすことはしないでしょう。社内の標準化活動がこのような抑制力として働きます。

10. メッセージングに関する標準

PSLX 標準仕様バージョン 2 では、このような製造現場で行われているメッセージングに関して、それらをより円滑に行うための標準技術が示されています。まず、メッセージの内容について議論するためには、2つのツール（道具）が必要となります。一つは、その業務とはどのような業務なのかをきちんと把握するための道具です。そして、もう一つは、そのメッセージに含まれる情報は、現実世界のどの対象（問題）を指しているのかをきちんと把握するための道具です。

最初の道具は、PSLX 標準仕様の中で、業務アクティビティモデル（パート 2）として定義されています。そして、二つ目の道具は、業務オブジェクトモデル（パート 3）として定義されています。これらの標準仕様はともに、情報交換を行う相手が、過去の長い経験の中で気心の知れた相手ではない場合や、環境の変化によって、既存の業務が大きく変わってしまった場合に、メッセージングを正しく行うために必須となる基本ルールあるいは辞

書となるものです。

1.1. SOAとPSLX標準仕様

現在、IT 業界では、SOA という言葉がほとんど無定義に使われています。おそらく、“サービスを指向したソフトウェアアーキテクチャ”とは、従来は機能やその実現手段を中心に議論してきたソフトウェアを、要求（つまり利用者にとっての価値）の視点から再構成しようということでしょう。目指す方向としては、PSLX と同じです。

PSLX 標準仕様の業務アクティビティモデルは、広い意味でサービスの定義といえます。図3に示すように、これらは階層を持っています。業務アクティビティモデルとしての PSLX 標準仕様では、図の業務機能ブロックと、業務アクティビティが定義されています。たとえば、業務機能ブロックの例として、“詳細スケジューリング”があり、その業務アクティビティの例として“作業指示情報の設定”が挙げられます。

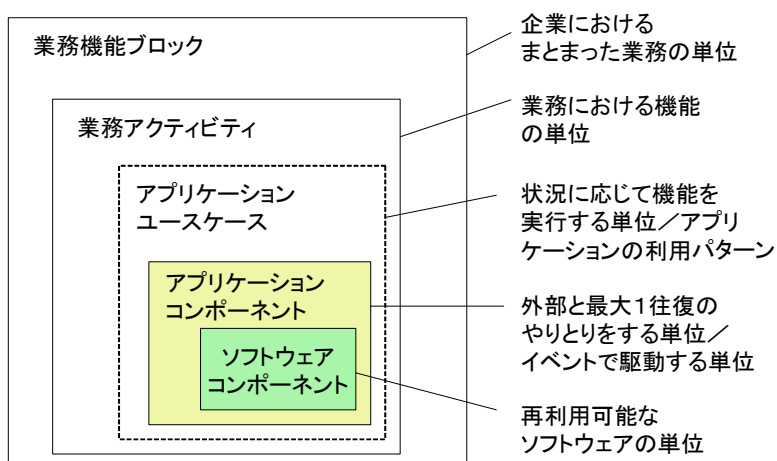


図3 サービスを指向したソフトウェアの定義

図3のように、ソフトウェアに関する議論は、この粒度よりもさらに細かいレベルとなります。つまり、各業務アクティビティにおいて、アプリケーションソフトウェアを利用する際の利用形態（アプリケーションユースケース）に応じて、アプリケーションの基本的な動作単位が“アプリケーションコンポーネント”となります。そして、アプリケーションコンポーネントを構成する再利用可能な共通的な部分を“ソフトウェアコンポーネント”として切り出すことができます。

1.2. メッセージの基本構造

複数のアプリケーションが連携する場合には、図3のアプリケーションコンポーネントの最小の単位は、外部と最大1往復のやりとりをする単位となります。つまり、メッセージ

交換の単位となります。PSLX プラットフォームで利用するメッセージは、OASIS の PPS 委員会が規定した標準仕様を用いて定義されます。各メッセージの構造は以下の図 4 のようになっています。これらのメッセージは、拡張可能な記述言語である XML によって記述されます。

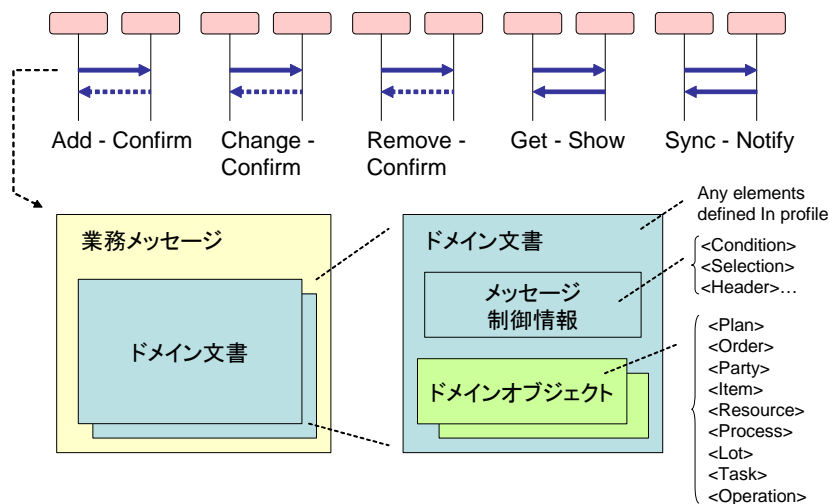


図 4 OASIS PPS メッセージの基本構造

図 4 において、ドメインオブジェクトとは、PSLX 標準仕様における業務オブジェクトモデルによって説明可能なものです。メッセージの中で、一つの業務オブジェクトは、9 つのプリミティブな XML 要素によって記述されます。

1.3. メッセージの伝達方法

業務メッセージを、相手のアプリケーションに送信し、その応答を受取るという基本的なメッセージングは、図 4 にあるように、いくつかのタイプに分類されます。図 4 において、Add は情報の追加を、Change は情報の修正を、Remove は情報の削除を依頼するものです。また、Get は情報の照会、Sync は事象の通知を依頼するものです。PPS 仕様では、これらの個々のタイプごとにアプリケーションが行うべき手順を規定しています。

中でも特に重要なメッセージングのタイプが Get-Show による照会です。これは、RDB における SQL 文に相当する仕組みであるからです。これは、図 5 のように、照会メッセージにおいて、まず Condition 要素によって対象オブジェクトを選択し、さらに Selection 要素によって対象オブジェクトがもつ属性（プロパティ）を指定します。ちょうど SQL 文の WHERE 句、SELECT 句に相当します。

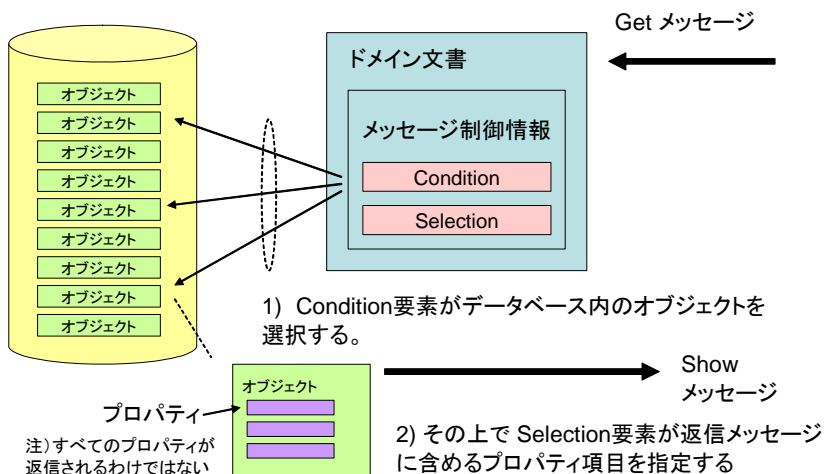


図5 メッセージングによる照会の仕組み

なお、複数のアプリケーション間で、メッセージングを実際に行う場合には、より下位レベルの通信方式について、規定する必要があります。PSLX 標準仕様では、その具体的な方式は規定していません。一方、PSLX プラットフォームでは、個々のアプリケーションが確実に繋がるために、具体的な通信方式を規定しています。これについては、後で説明します。

1 4. 個別性と拡張性のための仕組み

個々の業種や企業ごとに、業務メッセージの構造は異なっています。さらに、業務メッセージの構造は、ビジネス環境や生産方式の変遷によって徐々に進化しています。したがって、あらかじめ業務メッセージの構造を決定し、すべての関係者にその遵守を求めることには限界があります。OASIS PPS 仕様のパート3では、このような状況を想定して、アプリケーションプロファイルを用意し、当事者間で合意したアプリケーションプロファイルの上で、情報交換を保証する方式を規定しています。

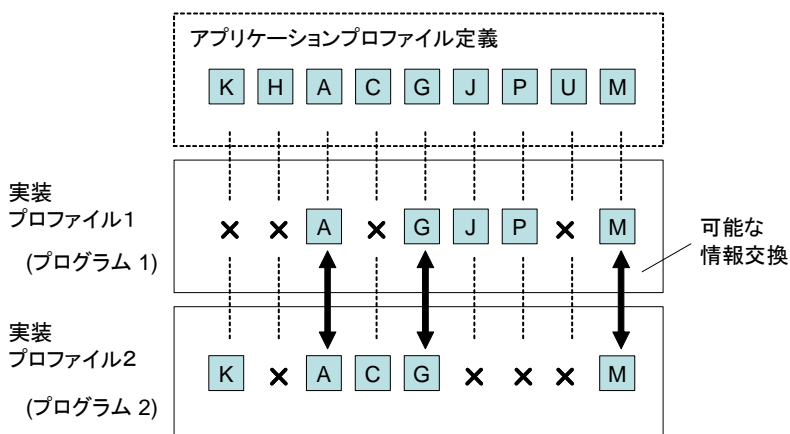


図6 プロファイルを用いた業務アプリケーションの能力定義

図6に示すように、各アプリケーションプログラムは、既存のアプリケーションプロファイルに対して、実行可能な項目（メッセージングのタイプや業務ドキュメントの種類など）を自己宣言します。システムインテグレーターは、その内容によって、複数のアプリケーション間の情報交換の可否を判断します。

アプリケーションプロファイルは、標準化団体が設定する場合、アプリケーションソフトウェアのベンダーが設定する場合、製造業自身が設定する場合などが考えられます。ベストオブブリード型のアプリケーションソフトウェアは、同時に複数のアプリケーションプロファイルに対して、自分のソフトウェアの能力定義を“実装プロファイル”として公開する必要があります。

15. PSLXプラットフォームの基本仕様

PSLXプラットフォームとは、製造現場を中核とした情報システムを構築する場合に、そこで利用されるアプリケーションソフトウェアが効果的に機能するための実行環境です。PSLXプラットフォームでは、各アプリケーションソフトウェアが、自分の得意とする機能をサービスとして提供することに特化でき、付帯的な機能を外部に委託することが可能となります。

既存のプラットフォームとしてのOSにたとえて言えば、MPUは印刷をする必要はなく、画面を制御する必要もありません。同様に、PSLXプラットフォームでは、工程スケジューラは、品目マスタのメンテナンスに多くの労力を費やす必要はありません。また、実績のタイムリーな取込のためにデータを加工し準備する必要もありません。

このような環境を実現するために、PSLXプラットフォームは、PSLX標準仕様およびOASIS PPS仕様と同時に、いくつかの共通的なソフトウェアコンポーネントと、システム実装のためのツール類を用意しています。また、実際にシステムを構築するために、より具体的な仕様として、データ通信の方式や、その管理方法なども合わせて規定しています。

16. プラットフォームの管理

PSLXプラットフォームのためのソフトウェア群は、ある意味で、ミドルウェアとすることもできます。ただし、一般的にはEAIソフトウェアのアプローチとは異なり、ミドルウェアによって分散したデータベース管理が、結果的に集中管理することはしません。プラットフォームを管理するためのソフトウェアは、それが存在しなくてもシステムが機能するような頑強さを持つようにし、原則としてオープンソースのソフトウェアとします。

PSLX プラットフォームを実現するための最低限のソフトウェアとしては、プラットフォーム上のアプリケーションの登録、アプリケーション間のメッセージの伝送とログ管理、業務オブジェクトの辞書管理、各業務オブジェクトの所有者（配置）の登録、マスタ情報の ID とバージョン管理、そしてアプリケーションプロファイル管理などが挙げられます。このうち、最も中核となる機能として、メッセージの伝送管理がありますが、これは SOAP やメッセージキューなど、オープンでかつ無償で公開されている信頼性の高いソフトウェアを用いて実装されます。

第三部 システム構築編

17. PSLX対応アプリケーションとは

アプリケーションソフトウェアを PSLX 対応とするには、具体的にどうすればよいのでしょうか？ PSLX プラットフォームで利用可能なアプリケーションであるためには、さまざまなアプローチが可能です。最も簡単なアプローチは、PSLX プラットフォームで標準的に用意されているアプリケーションプロファイルに対応して、特定の業務ドキュメントをファイル形式で出力あるいは入力できるようにすることです。これは、いわば CSV 形式のファイルを扱えるようにすることと似ています。

ただし、一般的に言えば、PSLX 対応アプリケーションは、アプリケーションプロファイルに基づいて、PPS 形式のメッセージを送信または受信できるソフトウェアを指します。これによって、特定の業務ドキュメントを、メッセージとして他のアプリケーションに送信し、それに対応した返信メッセージを解釈してアプリケーション自身の処理を行います。

以下の図 7 に、PSLX 対応アプリケーションの基本形を示します。従来のシステムとの最大の違いは RDB のスキーマを外部に公開せず、代わりに、対応可能な実装プロファイルを公開する点にあります。つまり、データベースの構造を隠蔽して、外部との情報交換はメッセージで行うことを原則とします。

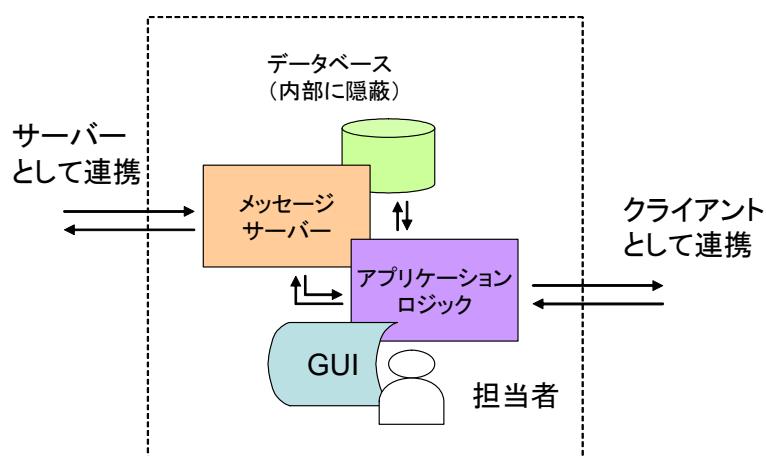


図 7 業務アプリケーションの基本形

各アプリケーションソフトウェアは、サーバとして振舞う場合と、クライアントとして振舞う場合があります。クライアントは、サーバに対して、最初にメッセージを送信し、何らかのサービス（追加、修正、削除、照会、同期）を依頼します。逆にサーバは、常に第

三者からのメッセージの受信に備えて待機し、メッセージを受取ったら、対応するサービスを実施します。

18. アプリケーションプロファイル対応

PSLX プラットフォーム対応アプリケーションは、さらに各業界団体や標準化団体が提供しているアプリケーションプロファイルに対応することで、それぞれのプロファイルへの準拠を PR することができます。また、基本的なマッピング部分を内部に構築し、必要に応じて、アドインソフトウェアの形で、製品販売後に提供されたプロファイルに対して、接続のためのコンポーネントを配布することも可能です。

最終的には、以下のように、ソフトウェアパッケージやカタログ等に表記されます。このように、同じ品目マスタでも、業界ごとに構造が異なる場合には、それぞれアプリケーションプロファイルが異なり、それぞれに対応してソフトウェアが設計されます。

<p>PSLX のマスタ連携機能</p> <p>電子機器業界用品目マスタ(003) 対応 自動車部品業界用品目マスタ(041) 対応</p> <p>PSLX トランザクション連携機能</p> <p>作業指示データ(038) 対応 作業実績データ(044) 対応 在庫実績データ(016) 対応</p>
--

図8 種類のことなるプロファイルへの対応

19. 共通ソフトウェアコンポーネント

OASIS PPS 標準仕様にもとづいた XML メッセージに対して、エンコード（生成）およびデコード（解読）を行うための基本的なプログラムは、どのアプリケーションプログラムでも共通で利用できますので、これらは共通ソフトウェアコンポーネントとして PSLX プラットフォーム参加者に提供されます。アプリケーションプログラムは、提供されたコンポーネントの API を利用して、アプリケーション自身の内部構造のデータを設定または取得することができます。

アプリケーション間のメッセージ交換のための機能も、共通コンポーネントあるいはミドルウェアとして提供されます。それぞれの製造現場の実装環境ごとに、メッセージ送受信の制御ソフトウェア（メッセージサーバ）が一つ必要であり、その形式に合わせて共通コンポーネントの形式を合わせます。たとえば、メッセージキュー方式として ActiveMQ を選択する場合には、それに対応したクライアント用コンポーネントが必要となり、SOAP 方式として、Axis2 を選択する場合には、またそれに対応したクライアント用コンポーネントが必要となります。

20. サーバの実装方法

PSLX プラットフォームでは、最低1つ、メッセージを仲介するブローカー（メッセージサーバ）が存在します。これによって、データベースを含むすべてのアプリケーションソフトウェアは、TCP/IP レベルで言うところのクライアントとして位置づけることが可能となり、常にメッセージを受信できるように待機させる必要はありません。その代わりに、アプリケーションレベルで言うところのサーバ（つまり追加、修正、削除、照会、同期の要求に応答する側）は、頻繁にメッセージサーバに自分へのサービス依頼があるかどうかをチェックし、ある場合には、直ちにサービスを開始しなければなりません。

したがって、アプリケーションレベルでのサーバを構築する場合には、下位のデータ通信のレベルで特別な機能を追加する必要はありません。アプリケーションのロジック上、メッセージを定周期で監視し、それに対応する処理を独自に記述してください。ただし、より高速なレスポンスが要求される場合には、メッセージサーバから、ダイレクトにアプリケーションレベルのサーバを呼び出すか、なんらかのトリガーを送ることでタイミングを待たずに処理させることも可能です。

21. PSLX対応アプリケーションの自作

PSLX プラットフォーム対応ツールキットあるいはコンポーネントを利用して、製造業の中で製造現場の情報化を推進するユーザ部門は、自分の手で、簡単なアプリケーションソフトウェアを自作することが可能です。以前と比較して、アプリケーションプログラム開発環境が飛躍的に進歩したために、ちょっとしたプログラミングを行うだけで、高度な業務処理を行うアプリケーションが簡単に作成できるようになりました。

たとえば、ガントチャートビューアのソフトウェアを用いて、Excel 上にある作業スケジュールの一覧表を、資源別のガントチャートにして表示したいとします。このためのプログラムは、共通コンポーネントソフトウェアと用いることで、図9のような数行のプログラムで実現可能となります。このプログラムは、ガントチャートビューアの入力ファイルを PSLX 標準仕様に従った XML 形式で出力します。

また、PSLX 標準仕様バージョン2では、パート6として RDB スキーマを定義しており、これを利用して、簡単な業務アプリケーションのデータベースを内部に構築することも可能です。


```

string nsPSLXSchemaName = "../../../data/pps-message-top-1.0.xsd";
string nsCoreSchemaName = "../../../data/pps-core-elements-1.0.xsd";
string nsTransactionSchemaName = "../../../data/pps-transaction-messages-1.0.xsd";
string excelFileName = Application.StartupPath + "/ganTest.xls";
string xmlFileName = Application.StartupPath + "/ganTest.xml";
string profileName = Application.StartupPath + "../../../data/profile-gantt.xml";

// Excel 上の列の定義
string[] operationScheduleFields = { // 作業指示フィールドの定義
    "operation-id",
    "equipment-id",
    "start-time",
    "end-time",
    "display-name",
    "display-name-top",
    "display-name-bottom",
    "display-color"
};

// Excel 用の変数
Excel.Application excel = new Excel.ApplicationClass();
excel.Workbooks.Open(excelFileName,
    Type.Missing, Type.Missing, Type.Missing, Type.Missing,
    Type.Missing, Type.Missing, Type.Missing, Type.Missing,
    Type.Missing, Type.Missing, Type.Missing, Type.Missing,
    Type.Missing, Type.Missing);
Excel.Worksheet sheet1 = (Excel.Worksheet)excel.Worksheets[1];

// PPS 用の変数
TransactionManager manager = new TransactionManager();
manager.nsCoreSchema = nsCoreSchemaName;
manager.nsTransactionSchema = nsTransactionSchemaName;
manager.nsPSLXSchema = nsPSLXSchemaName;
manager.Initialize(profileName);

// PPS 用の定義関数
Document operationScheduleDoc = manager.CreateDocument("OperationSchedule");
ProfileObject pObj = manager.GetProfileObject("OperationSchedule");
if (operationScheduleDoc == null || pObj == null) return;

for (int row = 2; ; ) //一行目はヘッダです。
{
    Excel.Range data = (Excel.Range)sheet1.Cells[row, 1];
    if ((string)data.Text == "") break;

    DomainObject obj = operationScheduleDoc.CreateObject();
    int cnt = 1;
    foreach (string field in operationScheduleFields)
    {
        data = (Excel.Range)sheet1.Cells[row, cnt++];
        obj.Set(field, data.Text);
    }
    row++;
}
manager.XmlWrite(xmlFileName);
MessageBox.Show("出力しました");

```

図9 ガントチャート表示のためのプログラム

2.2. PSLXが生み出す世界

製造業は、PSLX プラットフォームを採用することで、何がどのように変わるのでしょうか？ おそらく、従来から続いている生産の方法が、それによって大きく変わることはないでしょう。しかし、確実に言えることは、製造現場で行き交う情報が、整流化され、それによって、必要な情報が必要なときに必要なところに確実に届けられるようになります。現場と一体となって、あらゆる無駄をそぎ落とした、リーンな情報システムが出来上がります。

たとえば、生産管理で必要となるさまざまな形式の生産実績や稼働実績を専門に扱うソフトウェアが容易に実装できます。このソフトウェアは、さまざまな現場端末から、あるいは HMI や SCADA 等の機器から情報を集め、工程スケジューラ等のデータを提供することができるでしょう。また、実績だけでなく、スケジュールそのものを必要に応じて提供するスケジュール（作業指示）サーバも簡単に実装できるでしょう。

また、生産管理、品質管理、在庫管理、設計情報管理など、さまざまな業務アプリケーションが共通して利用する品目マスタを、一元的に管理メンテナンスする業務を独自に切り出し、それを行うアプリケーションソフトウェアが提供されることも期待できます。特に、個別受注生産のように、生産品目が、マスタとして定義できないような状況において、顧客独自の仕様や部品表をタイムリーに共有する仕組みを構築するには、PSLX プラットフォームは非常に適していると言えるでしょう。

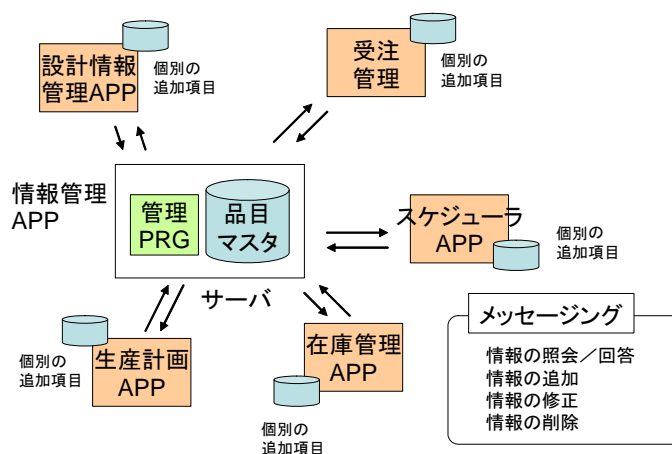


図 1.0 品目マスタを共有する PSLX プラットフォーム

2.3. おわりに

本稿で提案した PSLX プラットフォーム計画は、現時点（2008年5月）では、まだやるべき作業がいくつか存在していますが、今後1年以内に確実に実現できる内容としてま

とめています。ここで提案した仕組みは、ある意味では、現在の情報システム構築に対する新しいアプローチとして価値のあるものであり、ここでの成果は、他のいろいろな分野の情報システム構築に大きな影響を与えるでしょう。あわせて、製造現場が現在かかえている多くの問題を解決する可能性をも秘めています。多くの皆様の、ご支援とご協力のもとで、この計画をより具体化し、確実な成果につなげていきたいと思えます。

付 録

A. 利用可能ソフトウェア

- (1) コンポーネント
 - PPS メッセージ変換コンポーネント (C#対応)
 - PPS メッセージ変換コンポーネント (Java 対応)

- (2) サンプルプログラム
 - PSLX 標準 RDB 対応データサーバ
 - ActiveMQ テスト用モジュール
 - PSLX-Mule 接続キット

- (3) アプリケーション
 - ガントチャートビューア (Java アプレット)
 - RFID 対応在庫管理システム

B. 参考文献

- (1) PSLX 標準仕様バージョン 2 <http://www.pslx.org/>
 - [1] Part 1 : エンタープライズモデル
 - [2] Part 2 : 業務アクティビティモデル
 - [3] Part 3 : 業務オブジェクトモデル
 - [4] Part 4 : オントロジと情報モデル
 - [5] Part 5 : XML スキーマ
 - [6] Part 6 : RDB スキーマ

- (2) OASIS PPS 標準仕様 Version 1.0 (PPS : Production Planning and Scheduling)
http://www.oasis-open.org/committees/documents.php?wg_abbrev=pps
 - [7] Part 1 : Core Elements
 - [8] Part 2 : Transaction Messages
 - [9] Part 3 : Profile Specifications

このホワイトペーパーは以下のタスクフォースチームによってまとめられました。

西岡 靖之（法政大学デザイン工学部）

前田 智彦（富士通株式会社）

伊藤 昭仁（株式会社シムトップス）

野本 真輔（株式会社構造計画研究所）

PSLX フォーラムについて

PSLX フォーラムは、ものづくり APS 推進機構が運営している完全にオープンな技術集団であり、PSLX 標準仕様の策定や、計画スケジューリングを中心とした製造業の IT 化に関するさまざまな問題を議論するための組織です。ここで策定された標準仕様の一部は、すでに IEC 国際標準として採用されており、それ以外にも、国際標準化機関である ISO や、XML 標準化団体である OASIS などにおいて、製造業の IT 化を実現するための重要な国際標準として現在規格化が検討されています。わが国のものづくりの強みをさらに確かなものとするために、日本発の国際標準を今後も出し続けていく予定です。

ものづくり APS 推進機構について

NPO 法人ものづくり APS 推進機構は、わが国の製造業がもつ貴重な知的財産である現場中心の知識やノウハウを形式知化し、高度な計画スケジューリング統合技術によってダイナミックに変化するビジネス環境に同期しながら全体最適を実現するための情報技術インフラを確立するために、設立された非営利団体です。2001年に設立された PSLX フォーラムが改名し、2007年より新たにスタートしました。製造業の視点から、IT の活用を積極的に提案し、次世代の知的プラットフォームの構築を目指しています。

PSLXフォーラム ホワイトペーパー2
ー製造現場の情報連携を実現する
PSLXプラットフォーム計画ー

発行 平成20年(2008年)6月

発行者 PSLXフォーラム

NPO法人ものづくりAPS推進機構

東京都港区虎ノ門三丁目11番15号

財団法人製造科学技術センター内

電話 03-5472-2561