

製造業情報連携技術仕様書
(PSLXバージョン3)

2014年〇月

NPO法人ものづくりAPS推進機構

目次

はじめに	3
目的とスコープ	3
用語の定義	4
エンタープライズモデル	5
製造業の分類	5
業務モデルの定義	6
業務の階層と情報	7
業務アクティビティモデル	9
業務アクティビティの定義方法	9
販売管理・受注管理	9
生産計画・能力計画	10
業務オブジェクトモデル	11
業務オブジェクトモデルの定義方法	11
業務連携モデル	12
業務連携とデータ連携	12
業務データ連携のパターン	13
業務オブジェクトと業務データ	15
業務オブジェクトの標準化	16
インタフェース・プロファイル	16
連携エラーと取消	17

はじめに

目的とスコープ

この仕様書は、製造業あるいは製造に関係する企業や組織における情報システムを設計、構築、運用、保守するために必要となるモデルとその活用方法について規定しています。ここで情報システムとは、いわゆるコンピュータやデジタル情報機器によって加工、蓄積、伝達される狭義のICTシステムのみをさすのではなく、人と人とのコミュニケーション、高度な知識処理、組織的な慣習やルールといった非形式的なものも含まれます。

また、この仕様書を利用する企業や組織は、複雑で不確実な環境の中で、その情報システムを自ら再定義し、常に変化を繰り返すという進化型、あるいはイノベーション志向であること想定しています。つまり、そうした企業や組織の情報システムでは、そのビジネスの形態や業務モデルの再構築を繰り返す行うことにともない、常に設計、構築、運用、保守が繰り返されます。

こうした変化につよい柔軟な情報システムのためには、あらかじめそのビジネスカテゴリごとに大枠がデザインされたプラットフォームの上で、ビジネスを構成する各業務機能が個々に自律性を保ちながらも、相互に柔軟に連携できるしくみが必要となります。そして、それぞれの企業や組織の強みを発揮できる仕組みであると同時に、弱みをカバーできる平均レベルの機能は標準で用意されているような仕組みが望まれます。

この仕様書は、豊富な資本力とエンジニアリング力によって最新鋭の生産工場を建設する製造業にとって、あまり参考にはならないでしょう。また、あらかじめ安定的な需要が見込まれ、特定の製品を低コストで大量に生産する製造業も、参考になる内容ではない可能性があります。この仕様書は、むしろ、工場には多くの資本を投入できず、高度な自動化もあきらめ、人が中心となって日々変化する市場の要求に答えなければならない企業や組織を想定したものとなっています。

PSLX標準仕様バージョン1は、2003年6月にPSLXコンソーシアムから勧告されました。その後、2006年と2008年にバージョン2が

勧告され、この仕様書はそれに続くバージョン3となります。この10年間で我が国の製造業がおかれた環境は大きく変わりました。当時多かった“製造業の復権”や“生き残りをかけて”といった用語はもう聞こえません。すでに産業がサービス中心にシフトしてしまったともいえるでしょう。

バージョン3では、むしろ“開き直って”モノ作りの原点にもどり、モノと情報が一体となった、効率と柔軟性を併せ持った、日本人独特の創意工夫やカイゼン精神をふんだんに取り入れた情報システムを志向しました。国内でまだまだ頑張っている中小製造業、アジアをはじめとする海外で奮闘する現地工場、あたらしい付加価値をもとめたネットワーク型のファブレス企業や受託製造サービス企業など、あえてこれまでフォーカスが当たっていなかった組織形態をも念頭においた仕様となっています。

用語の定義

エンタープライズモデル

製造業の分類

製造業は、業種、規模、管理レベル、サプライチェーン上の位置付けなどにより、さまざまな業務形態をとっており、それらを1つの標準的な枠組みの中で議論することは不可能です。しかし、企業全体の視点から、すこしずつ粒度を細かくしていくと、個々の業務、あるいは個々の情報のレベルでは、共通する部分が多くみられます。

製造業を企業全体の視点からとらえた場合、この仕様書では、以下の2種類の分類軸を用いて整理することとします。第一の分類では、生産するモノの形態とそれともなう生産工程の形態からくる分類で、連続プロセス型、バッチプロセス型、そしてディスクリート型の3種類があります。

- (1) 連続プロセス型
- (2) バッチプロセス型
- (3) ディスクリート型

これに対して、第二の分類では、市場の要求、あるいは得意先の要求の多様性と不確実性の度合いによる分類として、繰り返し型、個別受注型、個別設計型の3種類があります。

- (1) 繰り返し型、
- (2) 個別受注型、
- (3) 個別設計型

以上の分類軸に従って、製造業のそれぞれの企業は、3×3の9種類の大枠のどれにもっとも近いかを議論した上で、この仕様書にある内容を、それぞれの企業にあった形にフィルタをかけて適用してください。

図 1 製造業の分類パターン

	繰り返し型	個別受注型	個別設計型
連続プロセス	①	—	—
バッチプロセス	②	③	—
ディスクリート	④	⑤	⑥

ただし、図に示すとおり、連続プロセスは、繰り返し生産を前提としたものであり、またバッチプロセスは、あらかじめ設定されたレンジの中での組合せで対応可能な個別受注型か、あるいは繰り返し生産となる場合が多いため、実質的には6パターンとなります。

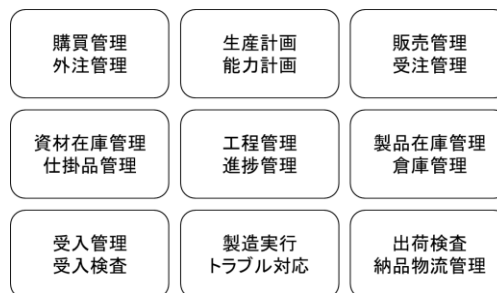
なお、新しい企業の形態として、次節で説明する企業内部の個別の業務機能をアウトソーシングした形のビジネスモデルも存在します。たとえば、設計機能のみに特化したファブレス企業、製造をサービスとして請け負うEMSなどの製造サービス企業などがこの代表例です。

業務モデルの定義

製造業は、それぞれ目的の異なる複数の業務カテゴリから構成されています。業務カテゴリとは、業務の分類名です。この仕様書では、サプライチェーンの視点とエンジニアリングチェーンの視点から、一般的で共通と思われるもののみを定義しています。ここであげたそれぞれの業務カテゴリは、次章で詳しく説明します。

(1) サプライチェーンの視点

サプライチェーンの視点では、製造業がもつ設計情報や製造技術をもとに、実際の得意先からの要求に対応して実際に製品を製造し納入します。また、そのために、あらかじめ計画を作成し、資材調達や在庫管理、工程管理などを実施します。



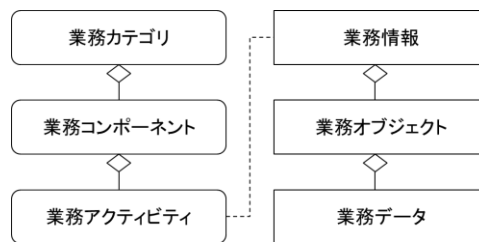
(2) エンジニアリングチェーンの視点

エンジニアリングの視点では、マーケットのニーズに対応して製品やサービスを設計し、製造業が生産可能なように、設計情報および製造技術として内部に取り込みます。生産のための設備や作業員などの能力および製造工程を実際に作り上げ、その能力を常に外部の環境にあわせて変更または維持します。



業務の階層と情報

この仕様書では、業務を定義するにあたって、3つの階層を設けています。まず、業務の最小単位は業務アクティビティです。業務アクティビティは、その成果物（アウトプット）として業務情報を定義できる単位となります。業務コンポーネントは、一般でいう“業務”に相当します。これは、業務担当者の視点から、関連する業務アクティビティをグループ化したものです。最後に、これらの業務コンポーネント、あるいは“業務”を、企業全体の視点からグループ化したものが業務カテゴリとなります。



一方、業務アクティビティの入力または出力となる業務情報は、1つまたは複数の業務オブジェクトによって構成されています。業務オブジェクトは、何らかの意味のある対象（オブジェクト）を表現したものであり、その具体的な内容を表す1つ以上の業務データによって構成されます。

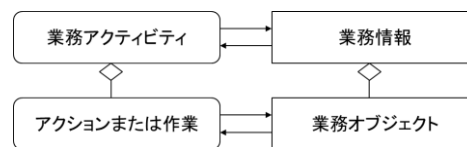
たとえば、N社の製品情報は、品目マスタと構成マスタによって構成されてお

り、品目マスタには、N340BとN339Aの2つの品目が設定されているとします。この場合に、製品情報は“業務情報”に、品目マスタと構成マスタは“業務オブジェクト”に、そしてN340BとN339Aは“業務データ”に対応します。

業務アクティビティモデル

業務アクティビティの定義方法

業務アクティビティは、その構成要素として、アクションまたは作業を持ちます。アクションまたは作業は、業務オブジェクトを入力または出力として定義します。つまり、アクションまたは作業は、1つまたは複数の業務オブジェクトを入力として、1つまたは複数の業務オブジェクトを出力とし、具体的には業務データを追加、修正、削除あるいは移動します。



業務アクティビティを記述する際には、構成するアクションまたは作業の他に、その開始条件と終了条件、トリガーとなるイベント類、インプットおよびアウトプットとしての業務情報、担当者および実施場所と実施頻度などが設定されます。ただし、この仕様書では、各業務アクティビティの特徴を規定する最低限の範囲として、アクションまたは作業の代表例と、それらに対応する業務オブジェクトを示すことに留めた。

販売管理・受注管理

(1) 業務アクティビティ名

業務アクティビティの説明

表 (アクションまたは作業、業務オブジェクト、操作区分: 参照、入力、追加、修正、削除など)

(2)

生産計画・能力計画

業務オブジェクトモデル

業務オブジェクトモデルの定義方法

業務オブジェクトは、情報システムの設計において用いられる概念データモデルに相当します。また、正規化されたER図（エンティティ・リレーションシップダイアグラム）のエンティティにも対応させることができるでしょう。ただし、リレーショナル・データベースのスキーマのように、システム実装上の物理的な制約や、効率などを考慮したモデルとは異なります。

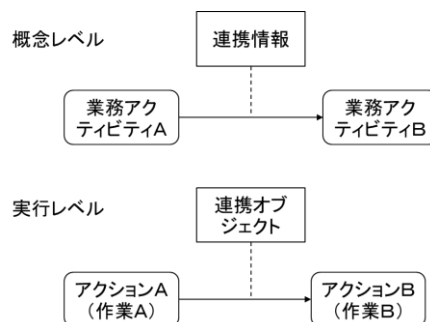
一方、UMLのクラス図のように、抽象化、一般化の関係を、モデル上で記述することはできません。

業務連携モデル

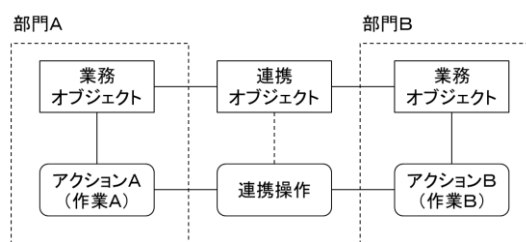
業務連携とデータ連携

企業や組織において、それぞれの業務が効果的に連携することで、得意先の要求への迅速な対応による収益の最大化や、予定外の問題に対する素早い対応による費用の最小化が可能となります。定型的な業務に加え、半定型的な業務、そして業務の形態が大きく変化する過程において、そうした業務間での連携を柔軟かつ効率的に行なうための情報システムとはどのようなアーキテクチャが必要なのでしょうか？

業務単独で考えた場合には、業務オブジェクトモデルの章において説明したように、連携する業務アクティビティおよびその構成要素となるアクションまたは作業の解層に対応して、業務情報と業務オブジェクトが定義可能です。これに対して、複数の業務間で連携するためには、それを仲介する連携情報と連携オブジェクトが必要となります。



ここで、連携オブジェクトは、以下の図のように、連携する2つの業務オブジェクト間での共通部分を抽出したものであるといえます。一般に、部門Aと部門Bとは、同じ対象をさす業務オブジェクトをそれぞれの業務のコンテキストに合わせて独自に定義しています。部門Aの担当者は、部門Bがもつ業務オブジェクトの内容をすべて知る必要はなく、部門Aと部門Bで共通する部分のみを認識すれば十分です。連携オブジェクトは、そうした共通部分を切り出したものと考えてください。



たとえば、受注伝票や出荷伝票などが連携オブジェクトに相当します。伝票それぞれは、連携データとなります。伝票は、ある業務から、別の業務に対して伝える必要がある情報を効率よく選択し配置したものであり、すべてのデータ項目に用途があります。

連携操作は、その操作対象を連携オブジェクトあるいは連携データのレベルに限定した場合、以下のように、連携データの追加、修正、削除、照会、そして通知などの処理となります。また、連携相手の状態の照会など、連携を行なうために必要な管理上の操作もあります。

連携操作の種類	説明
連携データの追加	
連携データの修正	
連携データの削除	
連携データの照会	
連携データの通知	
相手の状態の照会	

なお、情報システムを設計するうえで、情報連携とともに重要なキーワードに、情報共有があります。情報共有は、情報連携と異なり、情報を提供する側の業務が、その情報を利用する業務の時間や場所をあらかじめ特定できません。したがって、そうした情報オブジェクトは、データベースなどに蓄積保存され、必要な業務アクティビティがそれを必要なときに照会するという形式になります。主に、製品や工程などを表すマスタ情報は、こうした情報共有型のモデルが適しています。

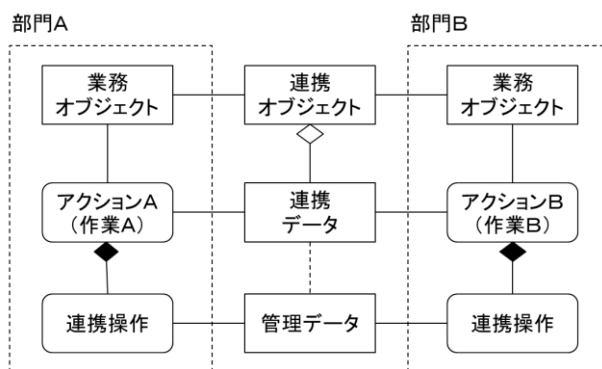
情報共有モデルは、情報連携モデルにおける連携先が、データベースなどの業務データの蓄積、検索、照会を行なう業務であるということもできます。

業務データ連携のパターン

この仕様書では、2つの業務が連携するにあたり、ITを活用した業務データ

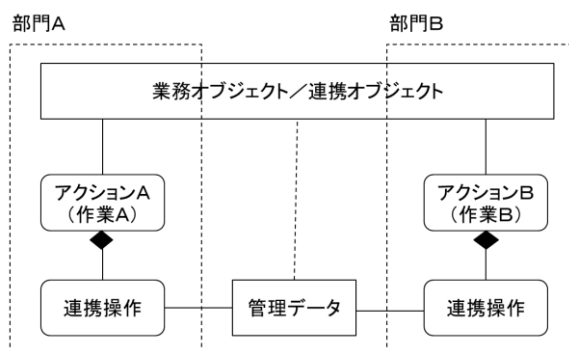
連携を行なうことを前提としていますが、それぞれに企業の実情に応じて、部分的に紙や人間系のコミュニケーションをベースとしてアナログ的なものも共存することが可能な形でモデル化します。

堅い情報システムの場合、あらかじめ業務データの流れに関する要件を定義し、その要件定義から逸れない範囲で、情報システムを設計します。ただし、こうした情報システムの場合には、最初に想定していなかった業務フローが後から追加された場合など、対応できません。したがって、そうした状況にも対応可能なように、業務データ連携のパターンはいわゆる“粗結合”とし、以下の図のように、連携オブジェクトと連携データ、そして実際に連携データを操作する連携操作から構成します。



この連携モデルの場合には、連携操作にあたる処理が既存のアクションに追加され、そこで業務オブジェクトと連携オブジェクト間のマッピングや、データの送受信にともなう処理を実施します。連携データは、実際に送信または受信するデータです。これに対して、管理データは、連携の方法に関する情報であり、具体的に送受信する連携データ以外の情報（操作の種類など）を設定し連携先に伝えます。

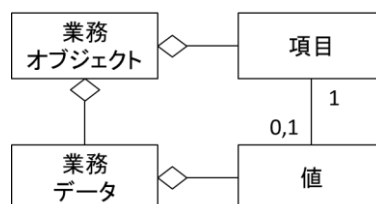
上記の連携パターンを実際に情報システムとして実装するには、比較的大がかりなしくみが必要となるため、簡易的な方法として、連携オブジェクトを介さずに直接業務オブジェクトを操作する方法もあり得ます。あらかじめ定義された業務オブジェクトの範囲内でのみ連携することで要求が満足する場合には、こうした方法が有効です。



これに対して、アクションAが、アクションBの連携操作を直接起動する場合は、アクションBの業務オブジェクトの内容を直接変更する場合など、2つの業務に対応したアクションを密結合する方法もありますが、この仕様書では、業務間の関係をより柔軟なものとするため、そうした方法は推奨していません。

業務オブジェクトと業務データ

業務オブジェクトは、複数の項目（オブジェクト指向では属性に相当します）を持っています。業務データは、対象となる業務オブジェクトの項目をすべて持っており、値が設定されています。



個々の業務オブジェクトにある業務データには、識別のための主キーが設定されていなければなりません。ここで主キーは、1つのデータ項目である場合と、複数のデータ項目の組合せで構成される場合があります。

複数の業務間で利用される連携データの場合、その識別のためには対応する業務データの主キーとは別に、連携データキーを設定しなければなりません。連携データキーは、連携データの送信者（作成者）が設定するユニークなIDと、送信者自身のIDの組合せとしてください。

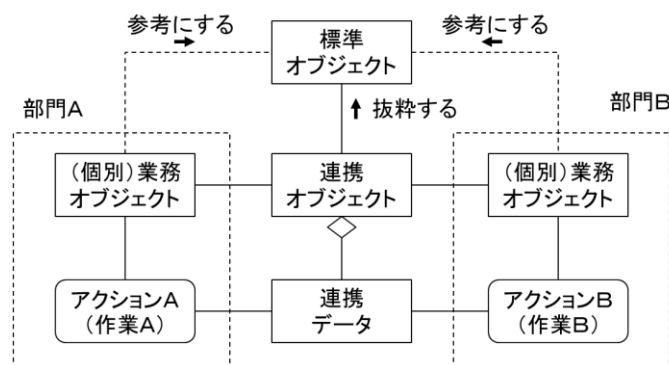
連携データをグローバルに識別可能とすることで、データ送受信時により低レベルでのエラー等による再送への対応や、データ連携における連携データと管理データとの対応づけなどが明確になります。

業務オブジェクトの標準化

業務データを連携するにあたり、あらかじめ連携する2つの業務が分かっている場合には、その2つの業務の間で共通となる連携データおよび連携オブジェクトを定義します。しかし、実際には、その後、2つの業務の一方の業務内容が変更になることや、それを実施するITシステムの仕様が変更になるなど、あらかじめ想定できない状況が発生します。

こうした想定外の状況に対応して、連携オブジェクトの仕様を変更すると、それに伴って、これまで古い仕様で連携できていたところで不具合が発生し、再度連携のための手直しが連鎖的に必要となってしまいます。

このような事態を避けるためには、個々の部門あるいは個々の企業が、それぞれ独自の業務オブジェクトを定義するにあたって、以下の図のように、標準業務オブジェクトを参考とし、できる限り差異が少なくなるような努力が求められます。こうすることで、業務オブジェクトの一部を抜粋した連携オブジェクトと、個別の業務オブジェクト間でのマッピングの手間を最小限にとどめることができます。



こうした考え方にに基づき、この仕様書では、標準オブジェクトとして利用可能な業務オブジェクトを定義しています。ただし、業種の違いや企業独自の環境によって、この仕様書で定義している業務オブジェクトを改変したものを、そうした限定された範囲の中での標準オブジェクトとすることは可能です。

インタフェース・プロファイル

連携オブジェクト、連携データのすべての項目に対して、どの項目が利用可能であるか、あるいは追加として設定する項目にはどのようなものがあるかなど、個々の業務オブジェクトを操作するアクションおよびアクティビティの

機能を定義したものをインタフェース・プロファイルと呼びます。

インタフェース・プロファイルは、業務または業務アプリケーション・プログラムが、他の業務または業務アプリケーション・プログラムと連携するために、どのような連携オブジェクトと連携データを利用するかを示したものです。また同時に、どのような連携操作を行なうことができるかを示します。

すべての業務または業務アプリケーション・プログラムが、インタフェース・プロファイルを明確に宣言することで、複数業務にまたがる情報システムのインテグレーションが容易となります。

ここで、連携オブジェクトは、標準オブジェクトから派生したものでなければなりません。標準オブジェクトとは、この仕様書の(3章)で規定する業務オブジェクトであるか、それらの業務オブジェクトとの関係を明示したものでなければなりません。業務オブジェクトとの関係とは、オリジナルの業務オブジェクトの項目の追加や削除、主キーの変更、業務オブジェクトの結合や分割などです。

もし、この仕様書で規定する業務オブジェクトではない標準オブジェクトを連携で利用する場合には、インタフェース・プロファイルとともに、標準オブジェクト定義書もあわせて用意し、連携する業務間であらかじめ合意する必要があります。

連携エラーと取消

連携操作をいくつかまとめて1つのトランザクションとして処理することは可能ですが、こうした連携操作によるアクションは、実行後に取り消すことはできません。また、もし連携操作がエラーとなった場合には、トランザクションの単位でもとの状態にもどしたうえで、エラーであることを依頼者側に通知しなければなりません。

なお、連携先の状態の照会が可能な場合には、過去の連携操作などのトランザクションを連携オブジェクトIDとしてその実行結果を照会することを可能としてください。