

製造業情報連携プラットフォーム 実装の手引き

2014年〇月

NPO法人ものづくりAPS推進機構

目次

はじめに	1
本書の利用方法.....	1
システム環境	1
連携システムのアーキテクチャー	2
連携方法の種類.....	2
連携アーキテクチャー.....	2
連携コントローラー	3
管理データの形式.....	4
連携方法 1（テキスト形式）	6
テキスト連携の概要	6
システム環境の設定	6
プログラムの実装.....	6
コントローラーの利用.....	6
実装例	6
連携方法 2（RDB形式）	7
連携方法 3（WEB形式）	8
連携コントローラー仕様.....	9
ユースケース	9
データ項目	9
動作環境.....	10
起動と終了.....	10
操作方法.....	10
エラーメッセージ	11
スキーマ拡張方法.....	12
業務オブジェクトの項目定義.....	12
業務オブジェクトの項目の追加と削除	12
業務オブジェクトの分割と統合	12
業務オブジェクトの追加と削除.....	13
概念スキーマと拡張物理スキーマの関係.....	13

はじめに

本書の利用方法

本書は、P S L Xバージョン3仕様書にしたがって、実際に業務アプリケーション・プログラムを連携させる情報システムを設計、構築するための手引書です。本手引書は、P S L Xバージョン3仕様書に準拠していますが、本手引書で示す方法が、唯一のP S L Xバージョン3仕様書に準拠した実装方法ではありません。

本手引書の利用者は、情報システムの構築を専門とする技術者です。ただし、XML関連技術や、データベース設計などの高度な専門技術や知識は前提としません。一般的な業務アプリケーション・プログラムを開発あるいは設計した経験のある技術者であれば理解可能な記述となっています。

システム環境

業務アプリケーション・プログラムが動作する環境は、Windows7または8のオペレーションシステム上とし、オンプレミス型のソフトウェアとして稼働するものとします。ただし、これは、これはWindows以外のOSであることを否定するものではなく、またサーバー上で稼働するWebアプリであることを否定するものではありません。こうした動作環境でも情報連携システムを構築することは可能ですが、本手引書では、Windows OSを前提とした解説となっています。

一方、連携のための連携オブジェクトや連携データがおかれる場所は、ネットワーク上のあらゆる場所が考えられます。本手引書では、それぞれの業務アプリケーション・プログラムが共通してアクセス可能なファイルサーバー、RDB、そしてWebサーバーの3種類を想定します。こうした外部のリソースには、連携データとは別に、連携のためのコントローラーがインストールさせる場合があります。

連携システムのアーキテクチャー

連携方法の種類

連携方法を、連携データの物理的な形式で分類すると、テキスト形式、RDB形式、そしてWEB形式の3種類があります。

分類	保存場所	接続方法	データ型式
テキスト形式	ファイルサーバー	F i l e	C S V
RDB形式	RDBサーバー	S Q L	テーブル
WEB形式	WEBサーバー	R E S T	J S O N

(1) テキスト形式

テキスト形式は、・・・

(2) RDB形式

RDB形式は、・・・

(3) WEB形式

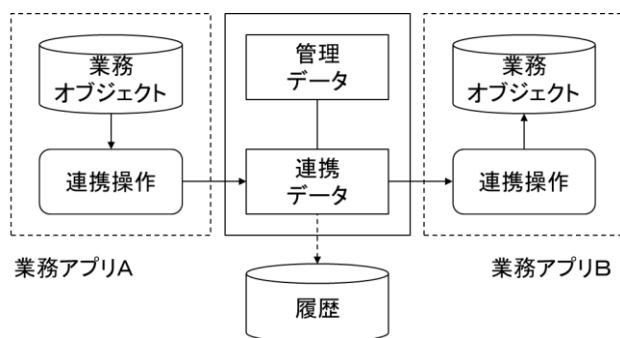
WEB形式は、・・・

連携アーキテクチャー

業務アプリケーションAから業務アプリケーションBへ、連携データを介して業務が連携する場合、一般的には、以下の図のように、業務アプリケーションAの業務オブジェクトと、業務アプリケーションBの業務オブジェクトとは別に、独自の連携データが設定され、連携操作方法に関する管理データとともに送付されます。連携データは、履歴として保存される場合もありますが、連携が完了したら基本的にその役割を終えて削除されます。このタイプの連携を、メッセージ型の連携と呼びます。

メッセージ型の連携は、OASIS-PPS仕様など、XML記述をベースとした業

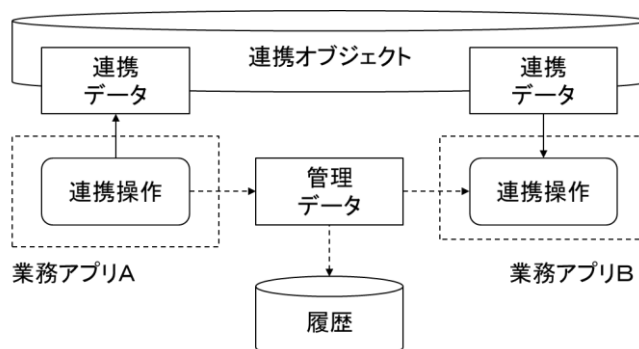
務アプリケーション連携方式が提案されています。



一方で、連携データを一時的なものではなく、永続化することを前提とした連携の方式では、以下の図のように、共通の連携オブジェクトに対応する連携データについて操作します。業務アプリケーションAから業務アプリケーションBへ連携データを送信するのではなく、業務アプリケーションAが連携データを書き、業務アプリケーションBが読むという形式をとります。

この形式の場合、業務アプリケーションAが登録した連携データを、業務アプリケーションBが取り出すために、連携データのIDや連携操作の内容などを、管理データとして業務アプリケーションAから業務アプリケーションBへ伝えます。これは、一時的な連携データであっても、データベースなどのストレージに蓄積される方式であり、ストレージ型の連携と呼びます。

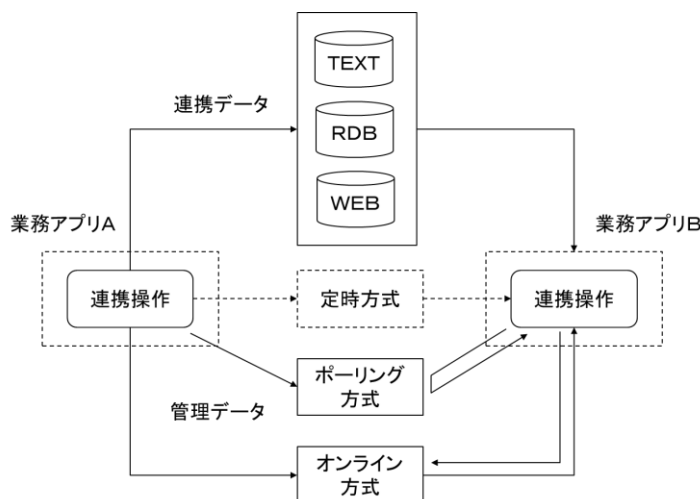
ストレージ型の連携は、業務アプリケーションにとって連携操作が比較的实施しやすいため、本手引書では、このタイプの連携を行なうことにします。



連携コントローラー

管理データは、たとえば、業務アプリケーションAが、“業務アプリケーションBに対して、ある連携データを共通領域に登録した”という事実を業務アプリケーションBに通知するために利用します。これを管理するのが連携コン

トローラーです。連携コントローラーが、連携データの送受信に関する情報を受信側の業務アプリケーションに通知する方法は、以下の3種類あります。



(1) 定時方式

定時方式は、業務アプリケーション間で、あらかじめ連携データの更新時刻を決定しておき、その時刻にあわせて連携データの送信側はデータを登録します。一方で、連携データの受信側は、その時刻から一定時間経過後に、データを受信します。したがって、実際の連携情報の送受信時には、管理データのやり取りはありません。

(2) ポーリング方式

ポーリング方式は、連携データの受信側が、連携コントローラーに対して、自分宛ての連携データあるいは問い合わせ情報がないかを、定期的に確認にいきます。ポーリングのサイクルを短くすることによって、連携データを送信する時間をリアルタイムに近いものにすることができます。

(3) オンライン方式

オンライン方式は、連携コントローラーが、データ連携を行なう業務アプリケーションとの間であらかじめ双方向の通信手段を確立しておき、連携データの送信側である業務アプリケーションが連携データを送信後、その事実を受信側の業務アプリケーションにリアルタイムで通知します。

管理データの形式

連携コントローラーは、業務アプリケーションからの連携データの転送依頼

に対応して、管理データを作成し保存および相手の業務アプリケーションに通知します。管理データは、連携操作1つに対して1つ生成され、以下の構造となります。

項目名	説明	必須
管理 I D		
送信者 I D		
データ I D		
サーバー名		
保存形式		
照会形式		
標準スキーマ名		
業務オブジェクト名		
業務データキー		
受信者または受信グループ		
受信者の権限		
送信日時		
保管期限		
ステータス		

管理データは、管理 I D でユニークであると同時に、送信者 I D とデータ I D の組でユニークとなっていなければなりません。

連携方法 1 (テキスト形式)

テキスト連携の概要

システム環境の設定

プログラムの実装

コントローラーの利用

実装例

連携方法 2 (RDB形式)

連携方法 3 (WEB形式)

連携コントローラー仕様

ユースケース

業務アプリケーションのIDと権限を管理する

業務アプリケーションが現在オンラインとなっているか把握している

業務アプリケーションから送信された連携データのログを蓄積する

送信先をグループとした場合に、グループに登録している業務アプリに配信してくれる。

連携データが保管されているサーバーおよびアクセス方法（物理アドレス）を管理する。

連携データの保管のデータを一定期間で削除する（データそのものを削除するかは別の話）

自分宛ての連携データがあるかどうかを教えてくれる

送信した連携データを、相手アプリがもっていったかどうかを知る

送信した連携データを、誰がもっていったかを知る。

自分が起動（READY）したこと、シャットダウンすることを通知する。

送信者が送った連携データについて、コントローラーに正常に到達、相手業務アプリに正常に到達、相手業務アプリからの返信をコントローラーが受信、返信内容を送信者が正常に受信、といったステータスを管理する。

通信エラーとなった場合に、その事実を記録し相手に伝える。

返信の期限を設定し、それを超えた場合はタイムアウトとして終了する。

データ項目

(1) 管理データ

(2) 業務アプリデータ

(3) メソッド

動作環境

連携コントローラーには、WEB版とLAN版の2種類があります。WEB版はポーリング方式のみ、LAN版はポーリング方式、オンライン方式の2種類の連携が可能です。

(1) WEB版

WEB版は、WWWサーバー上のPHPプログラムによって実行されます。データベースはMySQLを利用します。業務アプリケーション・プログラムが異なる動作環境、異なるサイト等に分散されている場合に有効です。

(2) LAN版

LAN版は、ローカルサーバー上で、Windowsオペレーションシステムにおけるサービスとして実行されます。ファイアウォールなどが介在するネットワーク上のセグメント内部でのより密なデータ連携を行なう場合に有効です。

起動と終了

システム動作環境

インストール

起動方法

終了方法

操作方法

業務アプリケーションの登録

インタフェース・プロファイルの登録

連携ログの管理

システム環境の定義

エラーメッセージ

スキーマ拡張方法

業務オブジェクトの項目定義

業務オブジェクトおよび業務データには、その特徴を表すための項目が定義されています。この項目は、概念モデルと実装モデル（物理モデル）の2つのレベルがあります。概念レベルでは、その項目が表す意味を指定し、実装モデルではその形式を合わせて定義します。概念モデルではデータ型は存在しないのに対して、実装モデルではデータ型を定義します。

たとえば、作業者の氏名を、氏名として1つの文字列で扱うか、姓と名に分けて2つの文字列とするかは、実装モデル（物理モデル）での定義であり、概念モデル上はこれを意識しません。

業務オブジェクトの項目定義では、概念モデル上での項目の追加または削除を行なう場合と、実装モデル上での変更の場合に分けて議論してください。概念モデル上では、項目の修正はできません。

業務オブジェクトの項目の追加と削除

業務オブジェクトは、業務データを識別するための主キーとなる項目があらかじめ定義されています。主キーは複数の項目の組合せである場合もあります。概念モデル上で項目を追加することで、主キーの候補が増える場合があります。主キーを構成する項目は削除できません。

項目を追加し、既存の主キーを拡張する場合があります。たとえば、“図面番号”が主キーとなっている業務オブジェクトがあるとします。これに対し、“改訂番号”という項目を追加し、“図面番号”と“改訂番号”の組合せを主キーとすることで、ことなる改訂番号をもった図面番号のデータを扱うことが可能となります。

業務オブジェクトの分割と統合

業務オブジェクトの項目の値の範囲によって、業務オブジェクトを分割する

ことができます。たとえば、“品目”という業務オブジェクトを、そのカテゴリによって“製品”と“部品”という2つの業務オブジェクトに分割することができます。また、“得意先”と“仕入先”という2つの業務オブジェクトを“取引先”という1つの業務オブジェクトに統合することも可能です。

業務オブジェクトを統合する場合には、統合後の業務オブジェクトの項目は、統合前のそれぞれの項目を引き継ぐ形となります。したがって、一方にしか存在しなかった項目は、統合後に、業務データの種類によって、値をもたない項目となる場合があります。業務オブジェクトの統合では、こうした項目はできるだけ削除してください。

業務オブジェクトの追加と削除

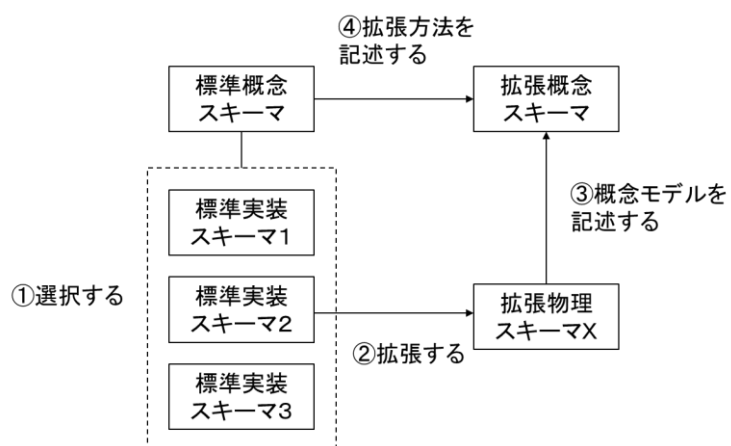
業務オブジェクトを追加する場合には、必ず既存の業務オブジェクトを参照する項目を設定するか、既存の業務オブジェクトの項目によって参照されるようにしてください。これらは外部参照キーとなります。

業務オブジェクトを新規に定義する場合には、その業務オブジェクトが表現する対象の意味を説明するとともに、その業務オブジェクトが属する業務情報、そしてその業務オブジェクトを利用する業務アクティビティを例示してください。

また、業務アプリケーションにとって、必要のない業務オブジェクトは、極力対象とする連携スキーマから除外してください。

概念スキーマと拡張物理スキーマの関係

連携システムの実装にあたっては、あらかじめ連携する業務アプリケーション間で、共通の連携スキーマを定義しておく必要があります。本手引書では、PSLX3プラットフォームで定義された概念スキーマに対応する物理スキーマを利用することを推奨していますが、その内容を拡張した拡張物理スキーマを別途定義することも可能です。



新たに物理スキーマを定義する場合には、それに対応する拡張概念スキーマを定義し、標準概念スキーマからの拡張方法を記述できなければなりません。具体的には、拡張概念スキーマは、標準概念スキーマに対して、追加した業務オブジェクト、追加した項目が明示されていなければなりません。また、主キーや外部参照キーを記述し、追加または変更された箇所が明示されていなければなりません。