

Pharmanage™ V 顧客固有の業務プロセスのシステム化を容易にする医薬向けMESの開発

Considerations in the development of Pharmanage V, a MES for pharmaceuticals that facilitates systematization of the user's manufacturing processes

染谷 知則
Tomonori Someya

松井 夏美
Natsumi Matsui

小山 英聡
Hidefusa Koyama

キーワード

最適化, Pharmanage, MES, 医薬, GMP, 製造, カスタマイズ, 在庫管理

近年、新薬の開発が盛んに行われ、医薬品製造においては市場への速やかな安定供給に向け、業務プロセスに適合するMES (Manufacturing Execution System:製造実行システム)の導入が求められている。このため、顧客固有の業務プロセスに柔軟に適合するための仕組みを備え、システム導入を容易にした新たな医薬向けMESパッケージソフトウェア Pharmanage Vを開発した。

In recent years the rapid pace of new drug development has spurred demand for a manufacturing execution system (MES) for pharmaceutical manufacturing that is adaptable to manufacturing processes in order to ensure prompt and stable supply to the market. Therefore, we have developed a new MES for pharmaceuticals, Pharmanage V, which has a mechanism for adapting flexibly to the user's manufacturing processes, facilitating systematization.

1. はじめに

Pharmanageはアズビルが提供している医薬向けMESパッケージソフトウェアである。

MESは、製造現場における様々な業務とリソースを人や製造機器と連携して管理・実行するシステムであり、品質や生産性の向上を目的に導入される。医薬向けMESは、一般的なMESの要件に加え、GMP (Good Manufacturing Practice:医薬品及び医薬部外品の製造管理及び品質管理の基準に関する省令)の順守を目的とし、誤作業防止や品質保証などの要件を満たした製造を実現する。

MESの導入では、医薬向けMESの要件を満たした上で、顧客固有の業務プロセスにシステムを適合することが求められる。

Pharmanageにおいて業務プロセスは、例えば造粒工程や打錠工程など一連の製造を表すシステムの管理単位である。業務プロセス単位で指示に基づいた製造が行われ、実績が記録される。

Pharmanage Vでは、使用する設備・機器・生産品目とその配合・作業手順などを定義することによって様々な業務プロセスを表現できる。これに加えて以下の特長を持たせることで、システムが顧客固有の業務プロセスに適合できるよう

にしている。

(1) 機能拡張・カスタマイズが容易に行えるシステム

システムがPharmanageの定義変更だけでは顧客固有の業務プロセスに適合できない場合、顧客の業務プロセスに対応したプログラムの変更が必要になる。このような場合においても、品質・コスト・納期への影響が少なく、機能の拡張やカスタマイズが容易に行えるシステムとする。

(2) 業務プロセス構築・変更の作業性が高いシステム

システム導入後であっても、生産品目の追加や工程・設備の改善により、業務プロセスの追加や変更が発生する。この場合、製造業務を実施しながら製造計画に間に合うように定義の構築や変更を行わなければならない。このため、業務プロセスの構築や変更の作業性が高い(編集中でも定義の整合性が保持される)システムとする。

本稿では、これらの特長を実現するために構築した仕組みを説明する。

2. 開発方針

Pharmanage Vの特長を実現するための開発方針を記述する。

2.1 機能拡張・カスタマイズが容易に行えるシステム

機能の拡張やカスタマイズは、追加・変更対象となる箇所が明確で、変更量が少なく、影響範囲が狭いことにより容易となる。

このため、Pharmanage Vでは、業務プロセスを構成する個々の業務を部品として構築・組替えができる「業務実行フレームワーク」を開発した。さらに、業務に共通するリソース操作の機能を業務部品と分離した上で、その変更が各種業務部品に波及しないようにする「伝票モデルリソース管理」も開発した。

2.2 業務プロセス構築・変更の作業性が高いシステム

Pharmanageでは、マスターデータと呼ばれる業務定義によって業務プロセスの構築や変更を行う。

業務プロセスの構築や変更の作業性は、実施中の製造業務に影響を与えない変更作業や、特定の設備や業務とそれに関連する各種マスターデータ式の並行した変更作業を実現することにより高めることができる。

このためPharmanage Vでは、変更用のマスターデータを複数管理し、実施中の製造業務に影響を与えないようにする「差分モデルマスターデータ管理」を開発した。

3. 業務実行フレームワーク

Pharmanage Vでは、人との主な連携手段としてブラウザで動作するインタラクティブなUI (User Interface)を使う。UIは図1に示すとおり、業務やリソースを表示・検索するための一覧表と選択行に関する各種業務のデータ入力・処理実行のための操作領域で構成される。

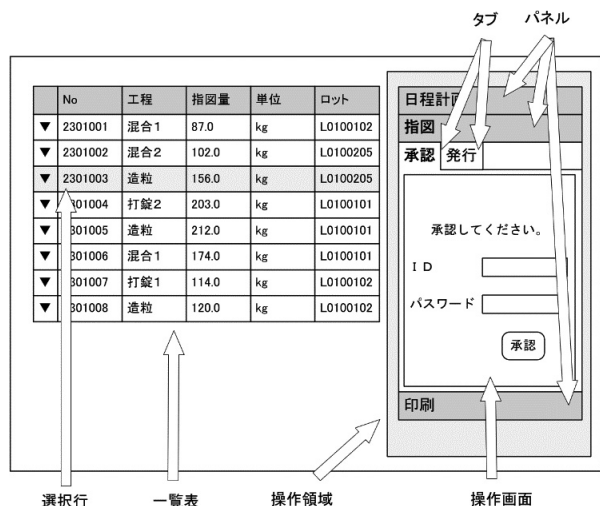


図1 画面構成

操作領域に表示される操作画面は業務の種類ごとに要件が異なり顧客ごとの違いも大きい。このため操作画面の実装は、開発方針に則り、機能拡張やカスタマイズが容易な形とする必要がある。

本開発では、操作画面を個々の業務と捉えて部品として構築するために、AO (Application Object) と名付けたフレームワークを開発した。AOは業務を表現する独立性の

高いオブジェクトの実行環境であり、業務の実装をオブジェクトの定義に集約することで独立性と拡張性を確保するフレームワークである。

3.1 言語の特長

AOは業務の実装（操作画面など）をオブジェクトの定義に集約する。オブジェクトの定義はAO専用のオブジェクト指向言語で記述される。AO専用言語は課題解決のために変更容易性を高めるべく、以下の特長を持つ言語として開発した。

(1) 迅速な開発・修正

動的にロードされるスクリプト言語であり、コンパイルを必要としないため、開発・修正後すぐに確認することができる。

(2) 理解しやすい記述

オブジェクトの振る舞いは状態遷移モデルで定義するため、処理の抜け漏れに気が付きやすく処理の構造が理解しやすい(3.2.3参照)。

(3) リレーショナルデータベース(RDB)との親和性

図2に示すとおり、データソースとしてRDBのテーブルを定義することで、レコードの情報に基づいたオブジェクトを検索・生成できる。このオブジェクトは基になったレコードにデータを反映することもできる。

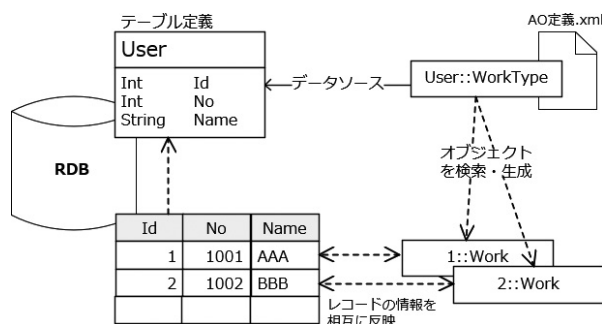


図2 RDBとオブジェクトの関係

(4) 言語自体の拡張性

オブジェクトの定義はAO専用言語によるオブジェクト指向における型 (AO型情報) の記述である。AO型情報は働き異なる複数の種類の「データソース」や「属性」「振る舞い」によって構成される。これらの要素は図3に示すとおり、必要に応じて追加することができる構造となっており、要素を追加することで言語自体を拡張することができる。例えばRDBのほかにCSVファイルやAOのオブジェクトを対象とする「データソース」を拡張可能な構造によって実現している。

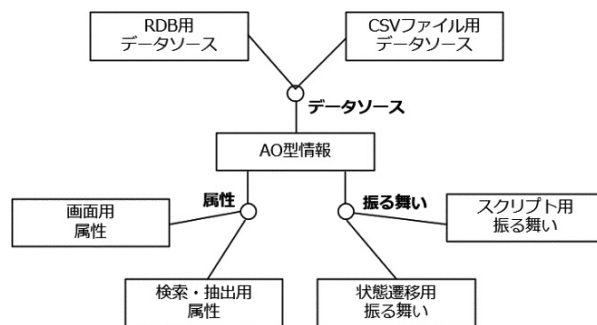


図3 拡張可能な構造

3.2 操作画面の実現

画面を実現する場合、一般に以下のような実装が必要となる。

- (1) フロントエンドの実装
(表示制御、入力検証、イベント処理など)
- (2) インターフェースの実装
(通信オブジェクト・通信処理・APIなど)
- (3) バックエンドの実装
(トランザクション制御・RDBアクセスなど)

これに対して、AOにおける実装はオブジェクトの定義に集約されており、これらの実装は不要である。オブジェクトの定義はAOの仕組みの上で実体化され、クライアントではフロントエンドとして、サーバーではバックエンドとして動作することで操作画面を実現する。

3.2.1 メニュー表示

一覧表で特定の行を選択すると、選択行に関連する各種業務を操作するための操作メニューが操作領域に表示される。この処理は図4に示すとおり、以下のように実現している。

- (1) AOは一覧表の種類と選択行を特定するキー情報からメニュー用のオブジェクトをサーバー上に生成する。
- (2) メニュー用のオブジェクトは属性として個々の操作画面のリンク情報を持っており（リンク属性）、業務名と業務用のオブジェクト定義名のほか、業務の分類や操作権限などが定義されている。
- (3) 生成されたオブジェクトはクライアントに返送され、属性の業務の分類と業務名を基にパネルとタブからなる操作メニューとして表示される。

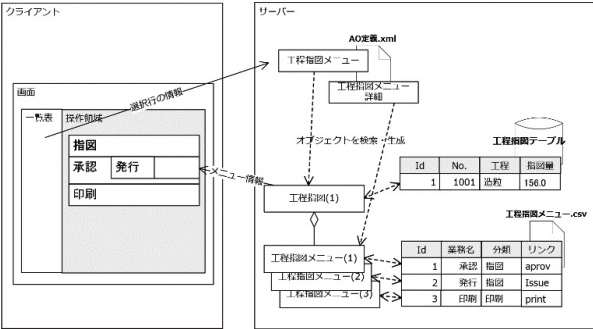


図4 メニューの表示

3.2.2 操作画面表示

操作メニューでタブを選択すると、対応する業務の操作画面が表示される。この処理は図5に示すとおり、以下のように実現している。

- (1) AOはタブに紐づいたオブジェクト定義名と選択行を特定するキー情報から操作画面用のオブジェクトをサーバー上に生成する。
- (2) 操作画面用のオブジェクトは属性として画面部品情報を持っており（画面部品属性）、画面部品の種類（テキストボックス・ドロップダウンリスト・ボタンなど）やレイアウト情報、表示条件、編集条件などが定義されている。
- (3) 生成されたオブジェクトはクライアントに返送され、レイアウト情報に従って画面部品が配置された操作画面としてタブの中に表示される。

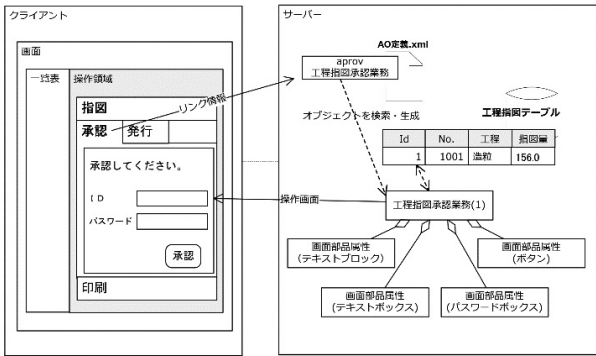


図5 操作画面の表示

3.2.3 イベント処理

操作画面でボタンを押すなどの確定操作を行うと、処理が実行され、結果が表示されるなど操作画面が変化する。この処理は図6に示すとおり、以下のように実現している。

- (1) AOは入力されたデータや押されたボタンのイベント名を内包する操作画面用のオブジェクトをサーバーに送信する。
- (2) サーバー上では操作画面用のオブジェクトがイベントを受け取り、状態遷移モデルに従って状態遷移とそれに伴う処理(Exit process・Entry process)を実行する。
- (3) 状態や処理が扱う入出力は属性であるため、透過的にデータソースであるRDBへの読み書きも行われる。
- (4) 状態遷移が終わったオブジェクトはクライアントに返送される。返送された操作画面用のオブジェクトは画面部品の表示判定などが変化しているため、表示される操作画面も変化する。

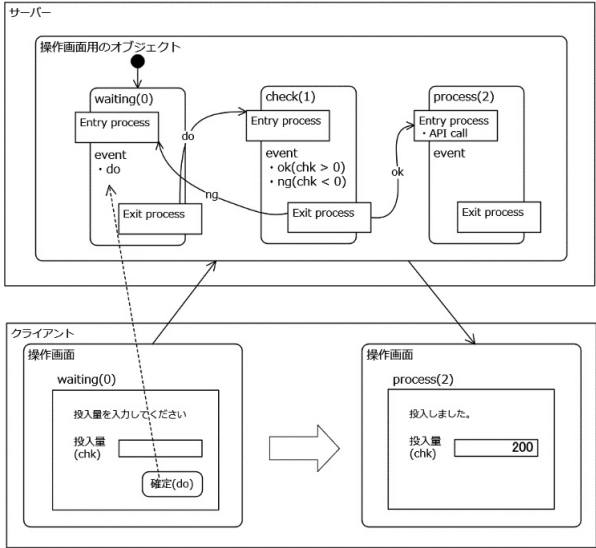


図6 状態遷移と画面の例

4. 伝票モデルリソース管理

Pharmanageは、医薬品製造で使用する原材料・容器・装置・機器などのリソースを扱い、品目・ロット・保管場所などの単位で管理される管理対象の現在の在庫量と、検査結果や有効期限などの現在の状態を管理する。

Pharmanage Vにおけるリソース管理の開発では、開発方針に則り以下の要件を定めた。

- (1) リソース操作の機能を業務部品と分離する。
- (2) リソース操作時に管理単位ごとの集計を行わない(参照時に集計する)。
- (3) データベースの構造を変えずに管理単位の追加や変更ができる。

このため、リソース操作の機能を業務部品と分離し、現在の在庫量ではなく受払量を伝票形式でモデル化した仕組みを開発した。

4.1 伝票モデル

現在の在庫量をデータベースに記録する方式(実体モデル)の場合、図7に示すとおり、同じ管理対象の操作は同じ場所に読み書きすることになる。

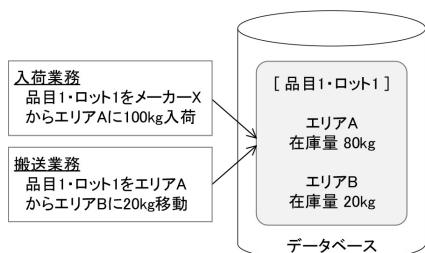


図7 実体モデルにおける現在データの記録

それに対して伝票モデルは、図8に示すとおり、発生した受払量をそのままデータベースに記録し、現在の在庫量は集計により動的に作られる。このため、在庫量を記録する場所が必要なく、管理単位を追加・変更できる。

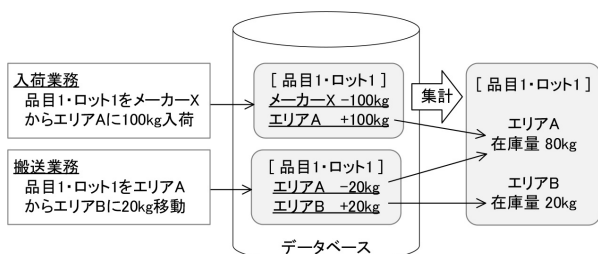


図8 伝票モデルにおける受払データの記録と集計

4.1.1 伝票モデルのデータ形式

伝票は 以下のように5W1Hのデータを保持している。

- (1) When:いつ … 操作日時, 計上日時
- (2) Where:どこで … 操作端末, 計上エリア
- (3) Who:だれが … 操作者, 計上者
- (4) What:なにを … 受払データ(予定/実績)
- (5) Why:なぜ … 指図キー, 操作理由
- (6) How:どのように… 伝票種別, 計上種別

なお、「How:どのように」の処理ごとに部品(API)化し、必要に応じて処理を追加できる仕組みにしている。

4.1.2 伝票モデルのデータ構造

伝票モデルのデータは、図9に示すとおり、5W1Hの中で「What:なにを」以外の基本データと、複数の「What:なにを」を表現する受払データから構成される。ここでは伝票モデルの核となる受払データについて詳細を記述する。

個々の受払データは、「なにを」「どこから」「どのくらい」移動するかを表すデータである。「なにを」は在庫の品目・ロット・現品などのキー項目、「どこから」は様々なレベルの保管場所を表す複数の項目(以降、管理項目)、「どのくらい」は受払量である。複数の管理項目にどの保管場所の組み合わせが記録されるかは、その組み合わせを分類する定義(以降、管理区分)で決まる。

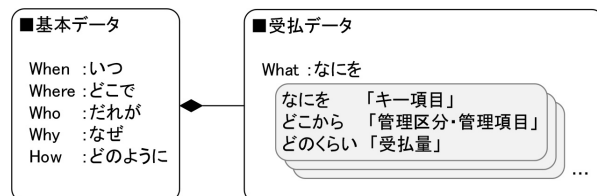


図9 基本データと受払データ

例えば、以下の2つの伝票が発行されたとする。

入荷:品目1・ロット1の原料をメーカーXからエリアAに20kg在庫生成(入荷指図α)

搬送:入荷した原料をエリアAからエリアBに10kgエリア変更(搬送指図β)

管理区分にERP(Enterprise Resource Planning)とエリアを割り当てると、受払データは表1のように記録される。

表1 受払データ/在庫生成とエリア変更

基本データ		受払データ					
指図キー	伝票種別	予定/実績	受払種別	受払量	在庫キー項目	管理区分	管理項目
入荷指図α	入荷	実績	払	-20kg	品目1	ロット1	ERP
		実績	受	+20kg	品目1	ロット1	エリア
搬送指図β	搬送	実績	払	-10kg	品目1	ロット1	エリア
		実績	受	+10kg	品目1	ロット1	エリア

この受払データを管理項目で集計すると、管理項目ごとの在庫量を集計することができる(この集計を在庫集計と呼ぶ)。表1の例では、原料(品目1・ロット1)がエリアAに10kg、エリアBに10kg集計される。

また、指図キーと伝票種別で集計すると、指図ごとの実績量を集計することができる(この集計を指図集計と呼ぶ)。

このように集計方法を変えると、在庫と指図の両方の情報を得られる(なお、メーカーXの原料-20kgは、ERPでメーカーXに原料20kg計上したものと捉えることができる)。

4.1.3 伝票モデルの特長

伝票モデルは、物品の入荷・出荷・生産・消費・移動のような物理的な操作を表現できるほか、予定から実績への変更、伝票の取り消しのような論理的な位置付けの変更も記録することができる。

ここでは、論理的な位置付けの変更の1つである予定から実績への変更(以降、実績振替)について記述する。

実績振替は、予定を表す受払データを実績に変更する処理である。

例えば、入荷指図αに対して以下の3つの伝票が発行され

たとする。

入荷予定：品目1・ロット1の原料をメーカーXからエリアAに20kg予定在庫生成

計上振替：入荷予定の計上を予定から実績に振替

在庫振替：入荷予定の在庫を予定から実績に振替

管理区分にERPとエリアを割り当てると、受払データは表2のように記録される。

表2 受払データ／予定在庫生成と実績振替

基本データ		受払データ						
指図キー	伝票種別	予定/実績	受払種別	受払量	在庫キー項目		管理区分	管理項目
入荷指図a	入荷予定	予定	払	-20k	品目1	ロット1	ERP	メーカーX
		予定	受	+20k	品目1	ロット1	エリア	エリアA
入荷指図a	計上振替	予定	払	+20k	品目1	ロット1	ERP	メーカーX
		実績	受	-20k	品目1	ロット1	ERP	メーカーX
入荷指図a	在庫振替	予定	払	-20k	品目1	ロット1	エリア	エリアA
		実績	受	+20k	品目1	ロット1	エリア	エリアA

計上振替は、入荷予定の払と同量の実績受と、払の打ち消し行を記録することで予定(払)を実績に振替えている。

在庫振替は、入荷予定の受と同量の実績受と、受の打ち消し行を記録することで予定(受)を実績に振替えている。

なお、計上振替で作られた実績は、ERP送信用のデータとして利用される。

また、計上振替と在庫振替が分かれていることにより、それぞれ別のタイミングで実施することができる。

5. 差分モデルマスターデータ管理

Pharmanage Vにおけるマスターデータ管理の開発では、開発方針に則り以下の要件を定めた。

(1) 分岐やマージによって複数のマスターデータの変更差分を管理する。

(2) マスターデータの変更が製造に適用されるタイミングを制御できる。

(3) マスターデータの作成や検証を実システム以外で行える。

このため、マスターデータの差分を管理することによって、編集用のマスターデータを複数持ち、製造に適用されるタイミングを制御できる仕組み(マスターデータリビジョン管理)と、マスターデータの作成や検証のための別システムとの間で変更を共有するための仕組み(システム間同期)を開発した。

5.1 マスターデータリビジョン管理

図10に示すとおり、リビジョンとは、リビジョン番号で識別される、ある時点での全マスターデータのことである。

リビジョンは継承元になるマスターデータからの差分を管理しており、リビジョンの発行によって確定される。また、同じリビジョンを継承元として複数のリビジョンを作成し、分岐することもできる。

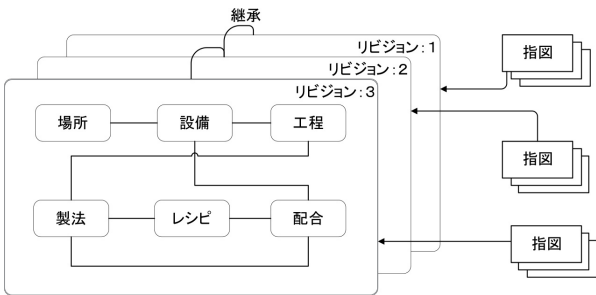


図10 リビジョン管理

5.1.1 リビジョンの作成

リビジョンは、特定のリビジョン(継承元)のコピーとして作成される。リビジョンは継承先を複数持つことができるため、リビジョン全体はツリー構造となる。

マスターデータの追加・変更は、リビジョンが継承されている場合、ツリー構造をたどって未発行かつ未変更の継承先リビジョンのマスターデータにも反映される。

5.1.2 リビジョンの発行

リビジョンは開始日時を持つ。発行され、開始日時が過ぎると、図11に示すとおり、現行リビジョンとして製造に使用できるようになる。

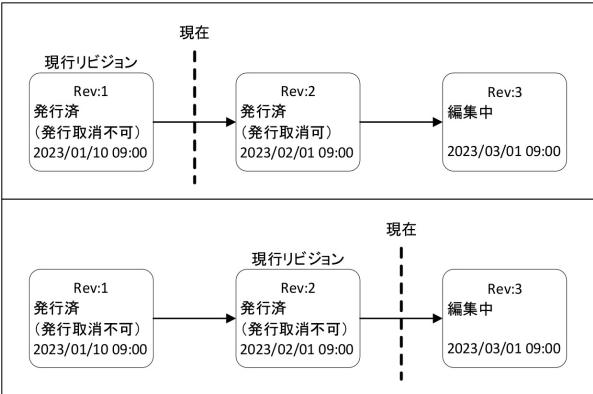


図11 リビジョンの発行と開始日時

さらに、リビジョンは図12に示すとおり、発行済と編集集中のステータスをもつ。編集集中の場合、マスターデータの追加・変更ができる。マスターデータの追加・変更が終了したら発行を行い、リビジョンのステータスを発行済にする。また現在日時が開始日時前ならば発行取消を行うことができる。

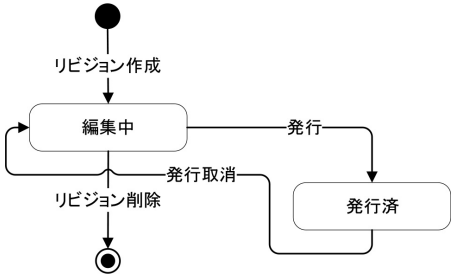


図12 ステータス遷移

製造指図を作成する際に使用されるリビジョンは、発行済かつ最も近い過去の開始日時のリビジョンである。以後、指図の記録と紐づくマスターデータは作成時のリビジョンで固定され、マスターデータの変更は指図の記録に影響を与えない。

5.2 システム間同期

Pharmanage Vは、図13に示すとおり、実システム以外にマスター検証用システムとマスター定義用システムを用意している。

これらは、マスター定義用システムでマスターデータを定義し、マスター検証用システムでマスターデータが正しく動作するかを検証後に実システムに取り込むことを想定している。これにより、マスターデータの間違っただけによる実システムのトラブル発生（製造停止）を防ぐことができる。

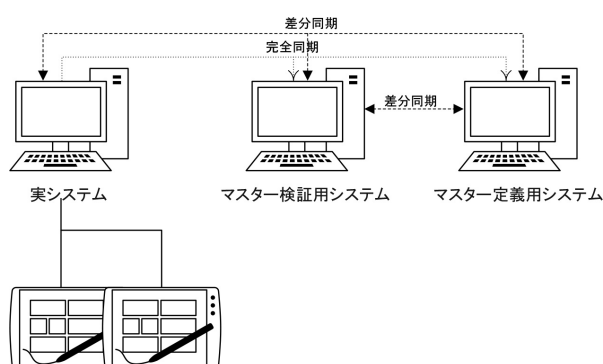


図13 システム構成

これらシステム間のマスターデータのやり取りには差分同期と完全同期がある。

5.2.1 差分同期

指定されたリビジョンが管理するマスターデータの変更差分を、ファイルを介して、別システムの編集時のリビジョンに反映させる機能である。

この機能により、同時に複数のリビジョンを検証し、任意のタイミングで実システムに取り込みマージすることができる。

5.2.2 完全同期

指定されたリビジョンのすべてのマスターデータを、ファイルを介して、別システムの新しいリビジョンとして作成する機能である。

完全同期で作成されたリビジョンは、継承元のリビジョンを持たないため、全データがこのリビジョンで作成されたデータとして扱われる。

6. おわりに

今回開発した、機能拡張やカスタマイズを容易にする仕組みと、システム上の業務プロセス構築や変更の作業性を高くする仕組みによって、顧客固有の業務プロセスを変更することなくPharmanage Vを導入することができる。

今後は、各種作業をタスクで表現したTo-Do管理ベースのインターフェースを追加することで、人・設備・指図それぞれの視点で作業負荷の見える化を進めたい。

さらに、システムで蓄積されたデータをAIで活用し、製造実績の品質基準からの逸脱予兆検知や業務プロセスの改善提案を実現することで、医薬品製造の安定供給における課題解決に貢献していきたい。

<商標>

Pharmanageはアズビル株式会社の商標です。

<著者所属>

染谷 知則	アズビル株式会社 アドバンスオートメーションカンパニー IAS開発部
松井 夏美	アズビル株式会社 アドバンスオートメーションカンパニー IAS開発部
小山 英聡	アズビル株式会社 アドバンスオートメーションカンパニー IAS開発部