

SAS ETL Serverを用いた 臨床試験データの集計解析用 データ加工業務での活用例

武田薬品工業株式会社 統計解析部 統計グループ 高浪 洋平
株式会社ファースト 西山 和志
住商情報システム株式会社 峯松 清志

目次

- データウェアハウスの変遷
- ETLの構成及び位置付け
- PH-DataWareとETLの比較・評価
- ETLの主な機能
- ETL活用事例
- 今後の展望

データウェアハウスの変遷

■ SAS 6.12

- SAS Warehouse Administrator 1.3
 - データウェアハウス構築ツール
- PH-DataWare
 - **Clinical Dataに特化していくつかの機能を追加**

データ加工システム
として導入

■ SAS 8.2

- SAS Warehouse Administrator 2.2
 - WA 1.3の後継システム
- SAS Drug Development
 - 医薬品開発に特化、FDA対応し、様々な機能を持つ

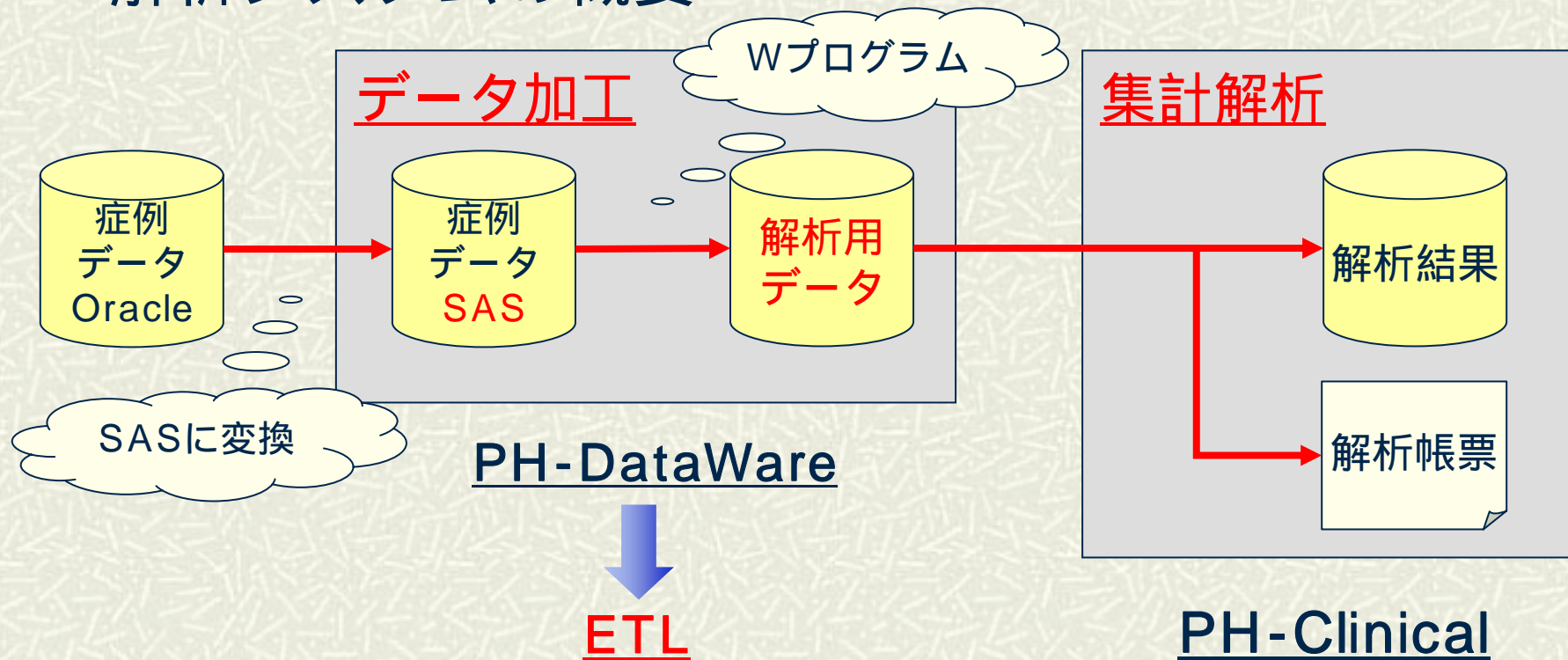
■ SAS 9.1

- SAS Warehouse Administrator 2.3
 - WA 2.2のメンテナンス版
- SAS ETL Server (JAVAベース)
 - **WAの後継ソフトウェア**
メタデータの管理・SASの実行・操作性など大幅に機能UP

新データ加工システム
として導入

ETLの位置付け

■ 解析システムの概要

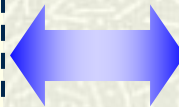
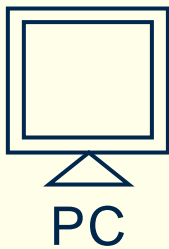


PH-DataWareに替わる新たなデータ加工システムとして
ETLの導入を検討

ETLの構成

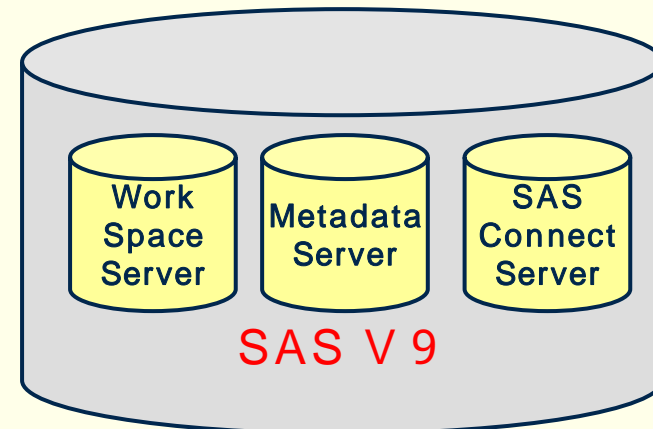
クライアント

- SAS Management Console (SMC)
 - メタデータの管理
- ETL Studio
 - プログラムの実行・ジョブの定義・ライブラリの定義etc
- Job Scheduler
 - ジョブの一括実行



サーバー

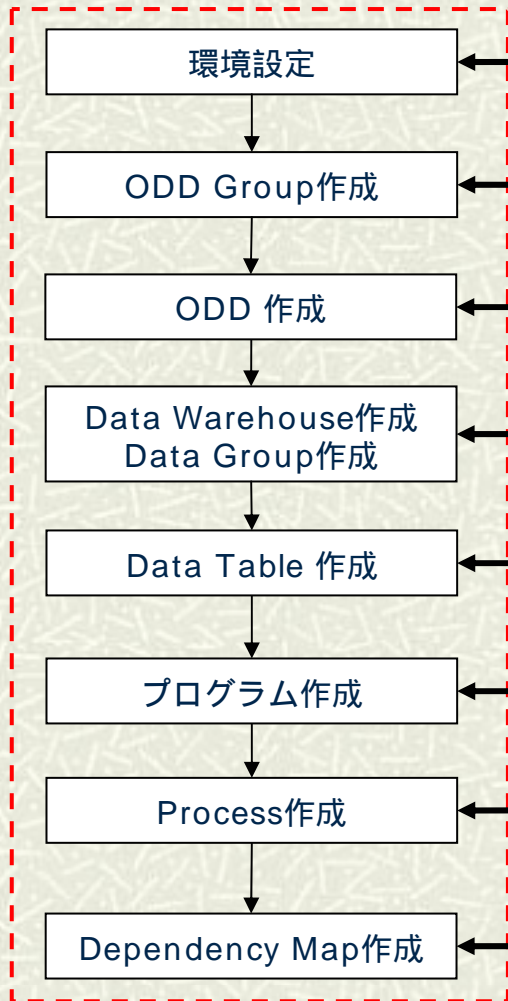
- SAS Workspace Server
- SAS Metadata Server
- SAS Connect Server
- SAS V 9



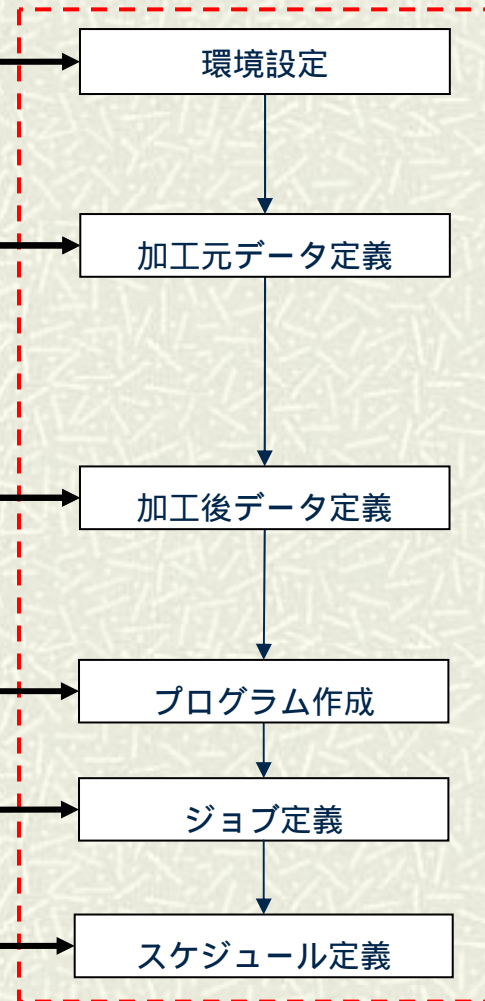
PH-DWとETLの比較・評価

各プロセスで
作業効率を
比較・評価

PH-DW



ETL



SAS Mangement Console(SMC)
・メタデータ管理

ETL Studio

- ・ライブラリ定義
- ・ジョブ定義
- ・プログラムの実行

Job Scheduler
・ジョブの一括実行

ETLの評価結果

■ ETLの主な評価結果

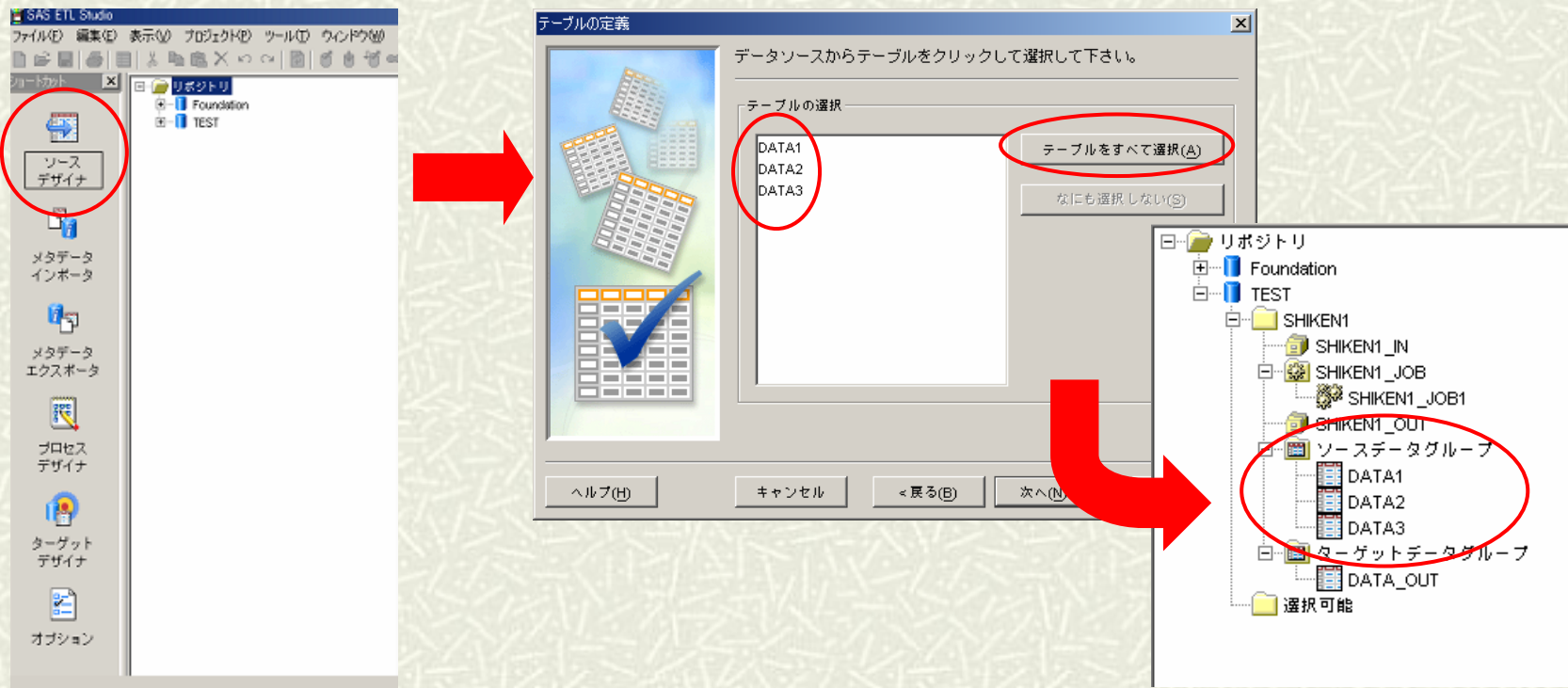
- 約50項目においてETLとPH-DWを比較した結果、ほぼ全ての項目でETLの評価が優れていたため、導入を決定した

ETL	PH-DataWare
SAS V9で動作	SAS V6で動作
プログラムの定義方法が簡単(SASファイル、TXTファイル等のプログラムが記述されたファイルを所定のフォルダに保存)	プログラムはBuildプロシジャで定義(SASカタログ形式で保存)
コピー&ペースト、ドラッグ&ドロップ、拡張エディタなど、GUI機能が豊富(主な機能のショートカットがメイン画面に表示されている)	コピー&ペースト、ドラッグ&ドロップはできない
データの一括定義が可能	データは1つ1つ定義(V6のデータセットのみ一括定義可能)
SMCでユーザー管理、権限の制約などを細かく指定できる	サーバーのユーザー定義で管理
チェックイン、チェックアウト、フェッチなどのデータの排他制御、同時アクセスが可能	データの排他制御はできない(同一試験では同時にログインできない)
コスト低、サポート、バグ対応が速やか	左記の点でETLに劣る

ETLの主な機能（1）

■ ソースデザイナー – 加工元データの定義

- 指定したライブラリのデータセットをETLに定義
- データセットの一括定義が可能



ETLの主な機能（2）

■ ターゲットデザイナー – 加工後データの定義

- 指定したライブラリにデータセットを定義
- 加工後データセットの変数名・ラベル・属性を指定
- エクセル・テキストファイル等からコピー＆ペースト可能

ターゲットテーブル デザイン

新しいテーブルの列の情報を確認して、テーブル列のインデックスを追加または変更します。

#	名前	説明	長さ	種類
1	ID	識別番号	8	数値
3	SEX	性別	8	数値
2	WEIGHT	体重	8	数値

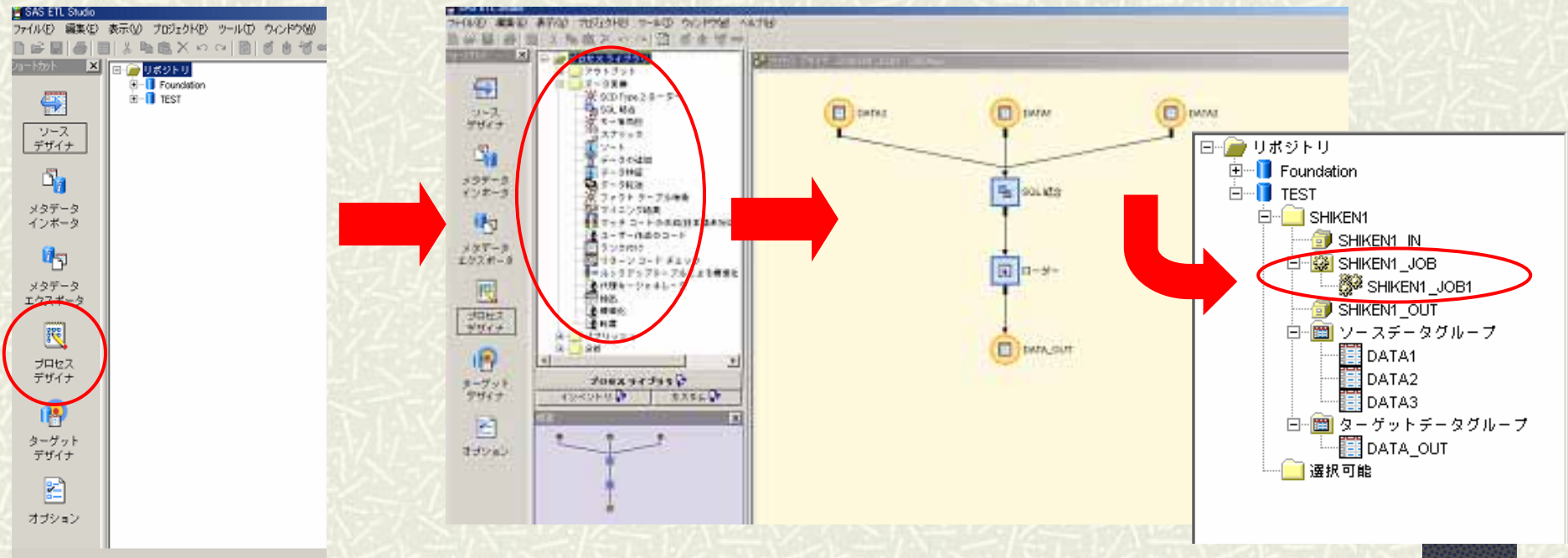
リポジトリ

- Foundation
- TEST
 - SHIKEN1
 - SHIKEN1_IN
 - SHIKEN1_JOB
 - SHIKEN1_JOB1
 - SHIKEN1_OUT
 - ソースデータグループ
 - DATA1
 - DATA2
 - DATA3
 - ターゲットデータグループ
 - DATA_OUT

ETLの主な機能（3）

■ プロセスデザイナー – ジョブの定義

- データの様々な変換プロセスを定義
- デフォルトで用意されている変換プロセス、ユーザーが作成したSASコードを定義
- データや変換プロセスをドラッグ&ドロップで定義



ETL活用事例

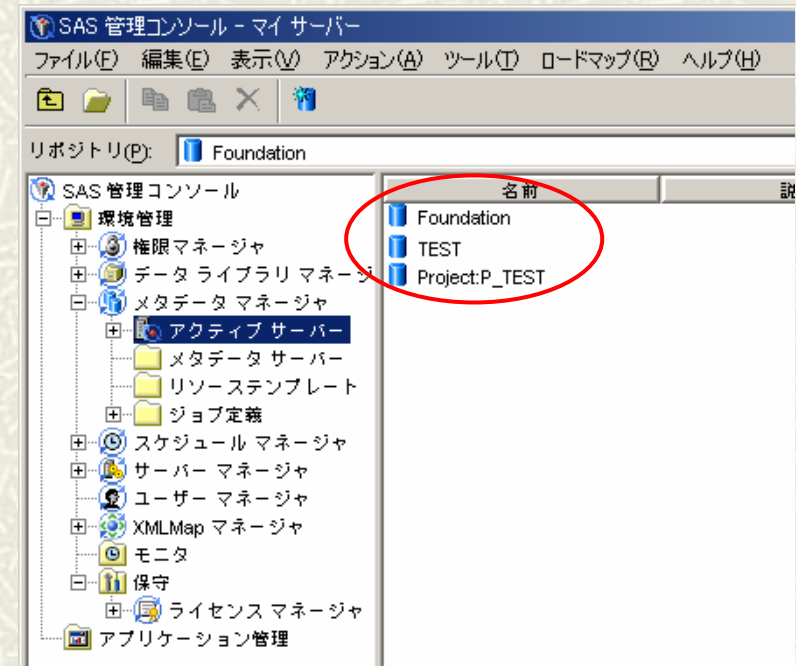
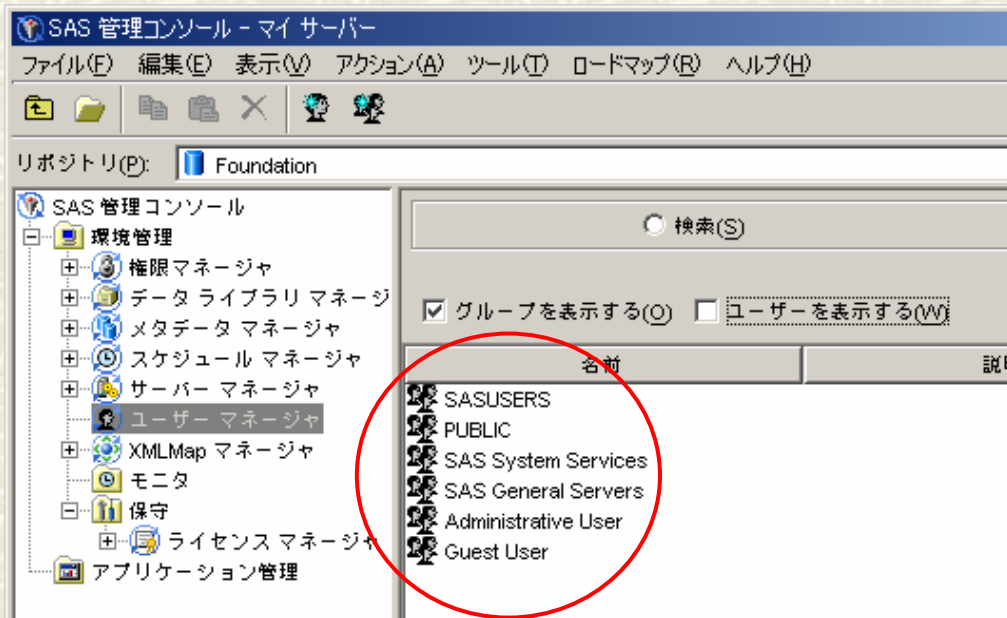
■ ETL導入後のデータ加工業務

- 環境設定
- データ定義・履歴管理
- プログラム作成
- ジョブ定義
- 一括実行

ETL活用事例(1) - 環境設定

■ SAS Management Console (SMC) - メタデータの管理

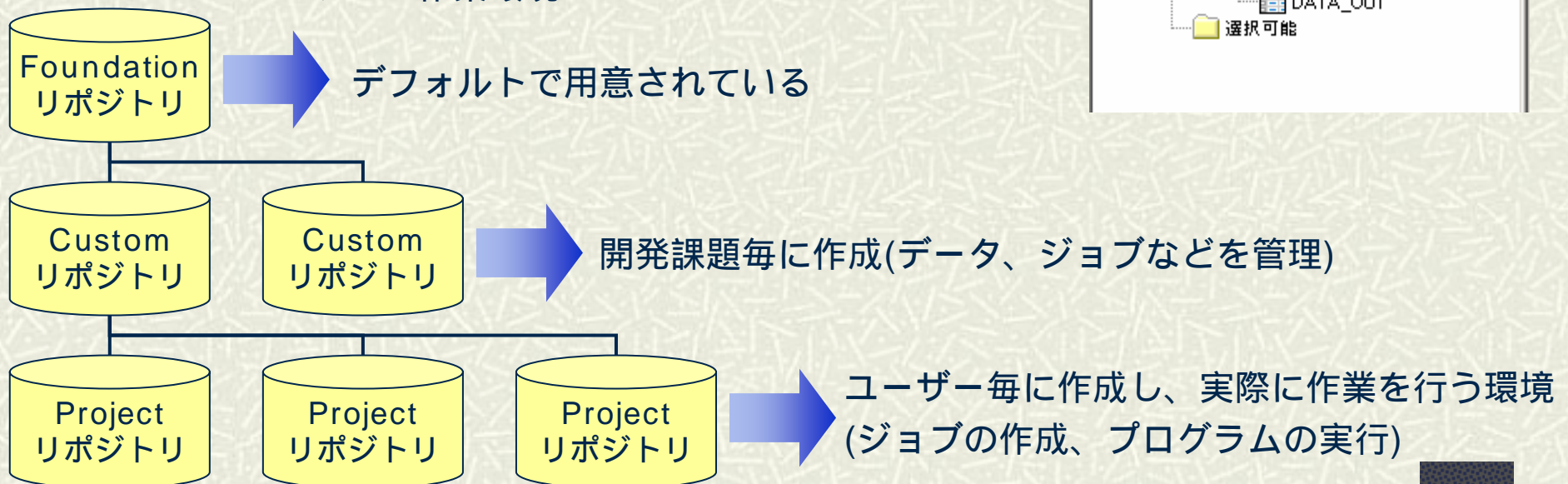
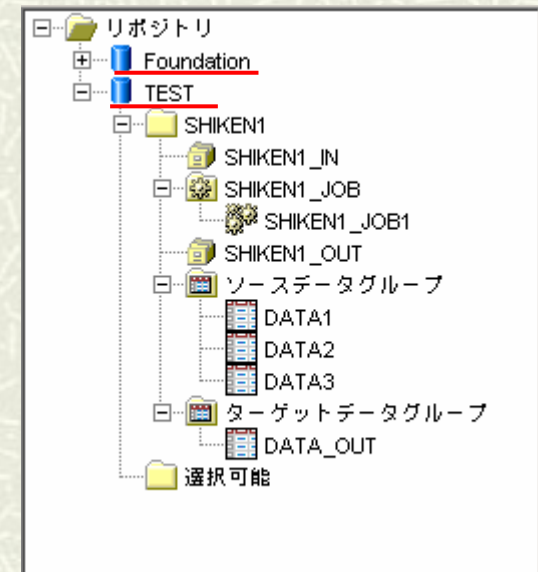
- ユーザー管理
- リポジトリ (メタデータ、ユーザーの作業領域等を定義するオブジェクト) の作成・管理



ETL活用事例（1） - 環境設定

■ リポジトリ管理機能（SMCで作成）

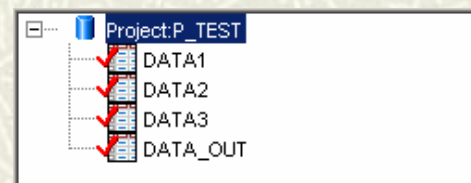
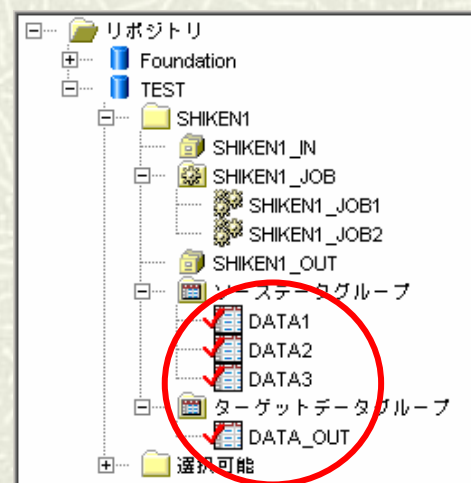
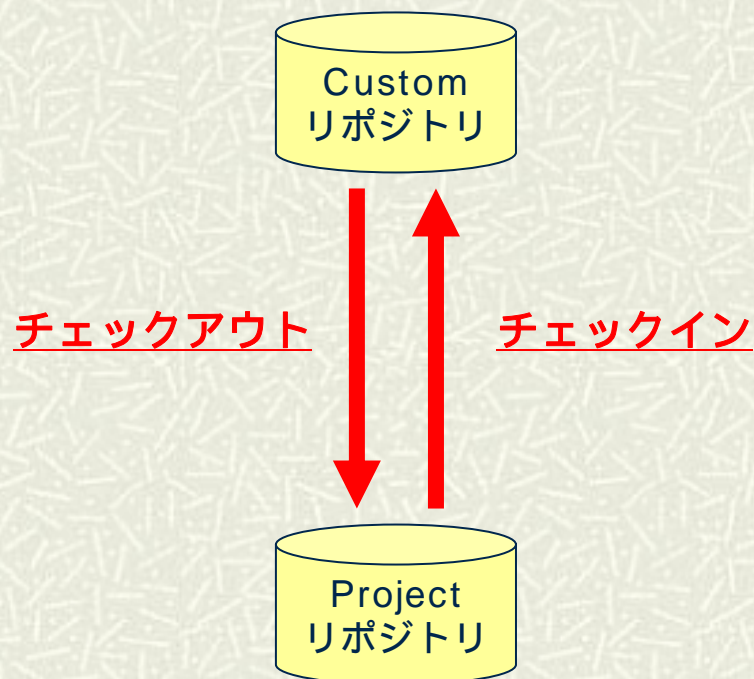
- Foundationリポジトリ
 - サーバーやメタデータを一括管理するオブジェクト
- Customリポジトリ
 - 開発課題毎のメタデータを管理するオブジェクト
- Projectリポジトリ
 - プログラム実行、変更履歴管理や排他制御等を行うユーザーの作業環境



ETL活用事例（2） - データ定義・履歴管理

■ チェックイン、チェックアウト機能

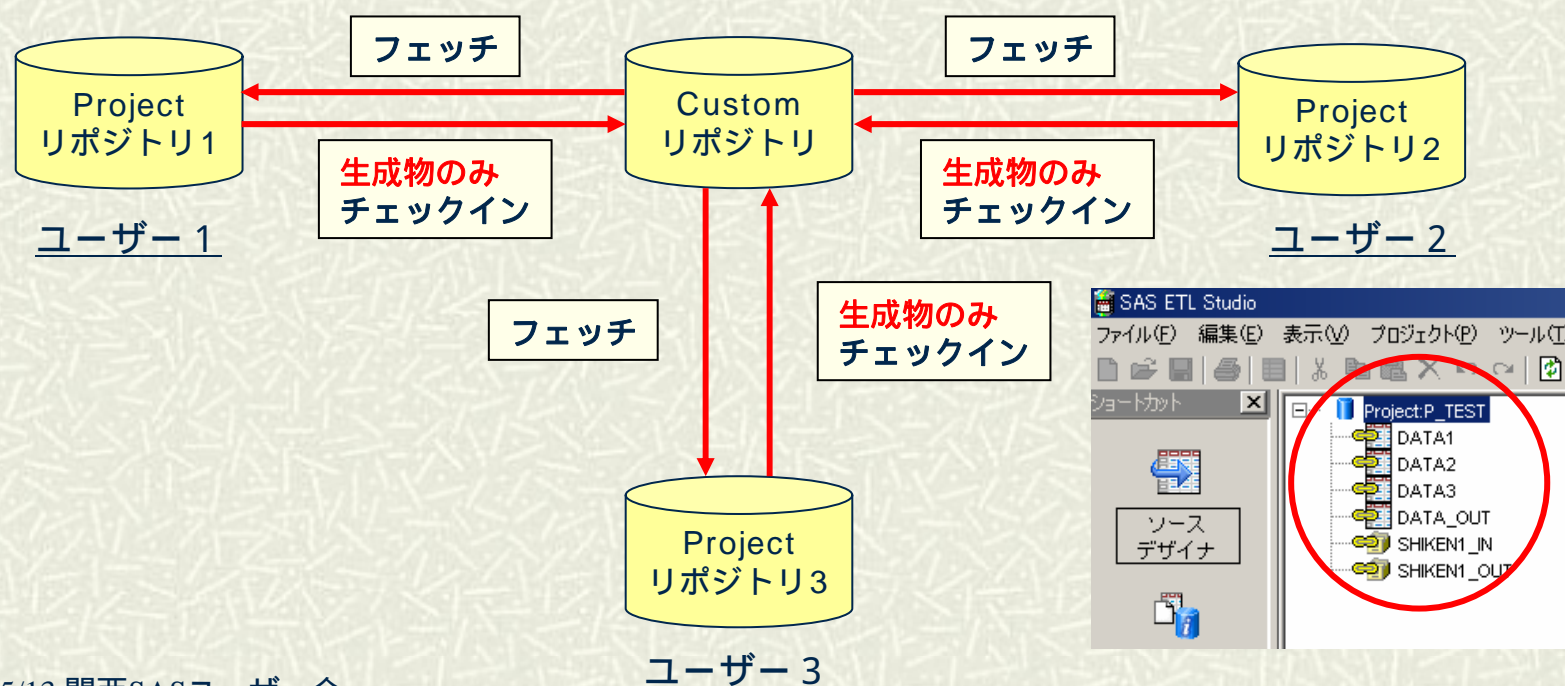
- Projectリポジトリでデータの定義、ジョブの定義、プログラムの実行等を行うため、ライブラリ、データ等を**排他制御**する
 - 下図のように、チェック(✓)マークが付いているオブジェクトは**他のユーザーがアクセスできない**
 - 作業後はCustomリポジトリにチェックインして排他制御を解除



ETL活用事例（2） - データ定義・履歴管理

■ フェッチ機能

- 排他制御はかけずにデータ加工が可能
 - オブジェクトをコピーしてProjectリポジトリへデータを取り込む
 - 変更はCustomリポジトリには反映されない
 - 作成されたオブジェクトのみ（加工後・加工元データ等）をチェックイン
- 複数ユーザーの同時アクセスが可能



ETL活用事例（3） - プログラムの作成

■ ユーザー作成コードとして定義

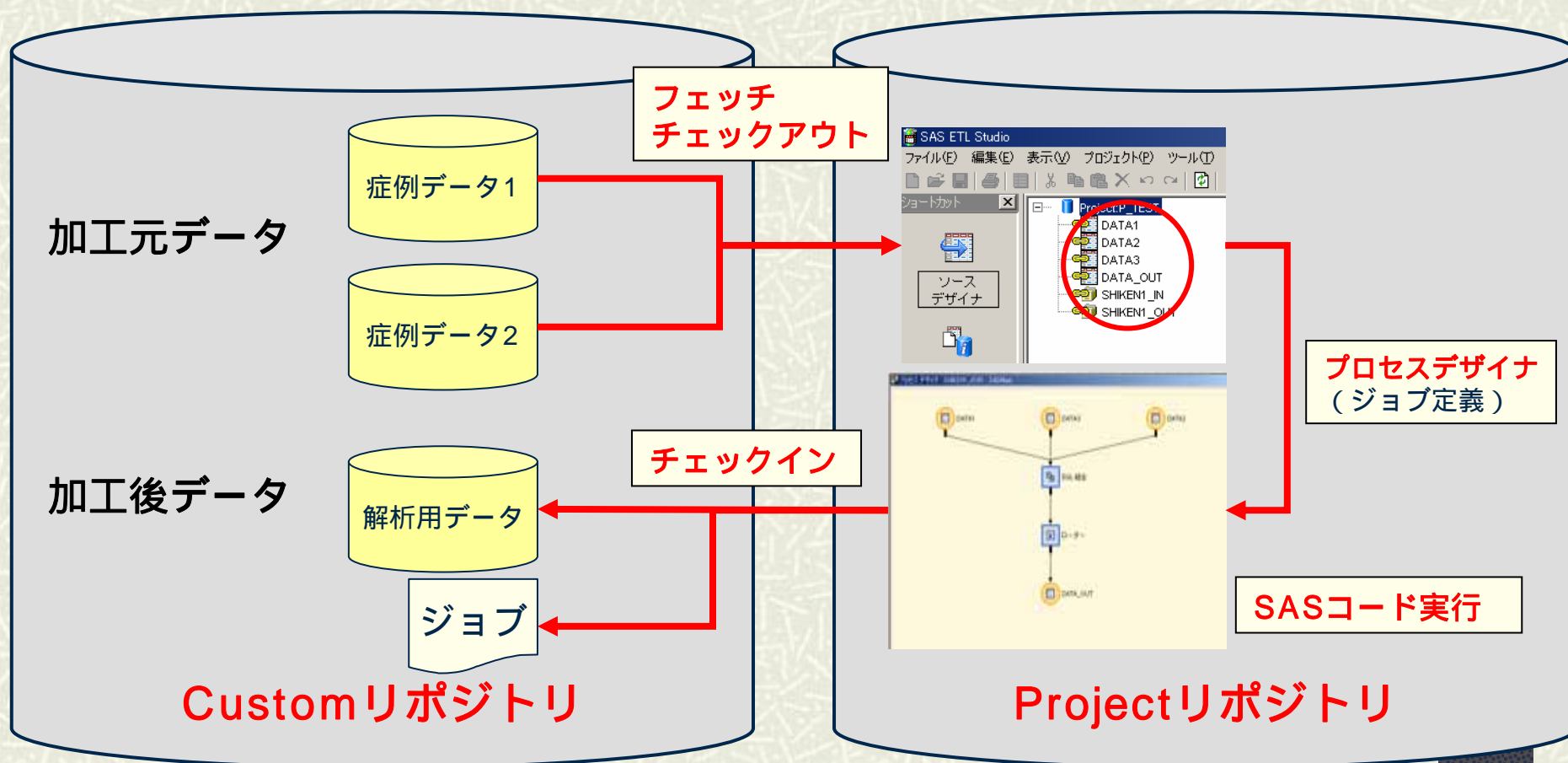
- プログラムは所定のフォルダに保存
- ETLへの定義は保存場所を指定するだけ
- プロセスデザイナーで直接ドラッグ&ドロップ

The screenshot displays the SAS Process Designer interface. On the left, the 'Process Library' (プロセスライブラリ) is visible, with 'User-created code' (ユーザー作成のコード) highlighted in red. The central workspace shows a process flow: DATA1 (source) -> 'ユーザー作成のコード' (User-created code) -> ロード (Load) -> DATA_OUT (target). The 'ユーザー作成のコード' process is circled in red. On the right, the 'Properties of User-created code' (ユーザー作成のコードのプロパティ) dialog box is open. The 'Code Generation' (コードの生成) section has 'User-created code' (ユーザー作成) selected. The 'Path' (パス) field is set to 'D:\ETL.txt' and is also circled in red. Other fields include 'Type' (種類) set to 'File' (ファイル) and 'Host' (ホスト) set to 'SASMain'.

ETL活用事例（４） - ジョブの定義

■ ジョブの定義-プロセスデザイナー

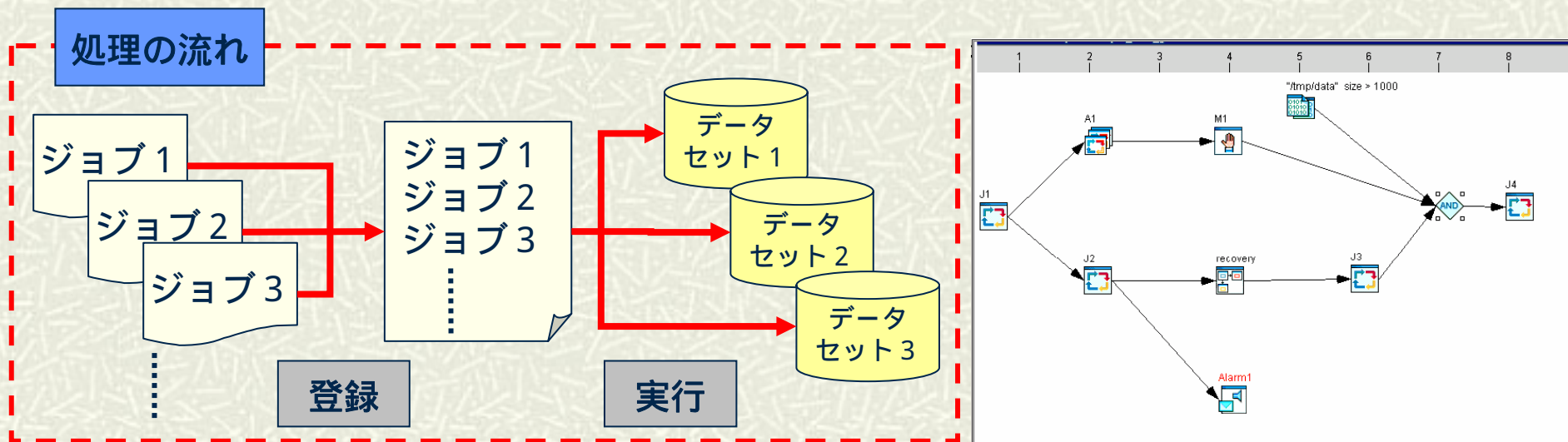
- フェッチ、チェックアウトしたデータに対する処理プロセスを定義



ETL活用事例（5） - 一括実行

■ **Job Scheduler**（本格的にはまだ稼働していませんが・・・）

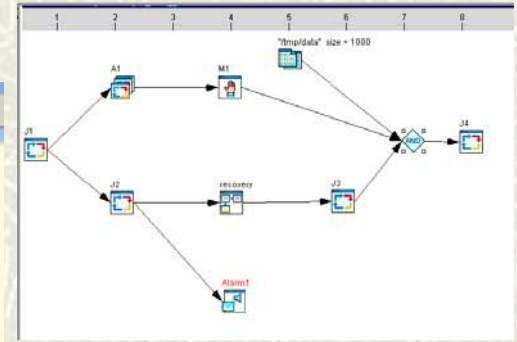
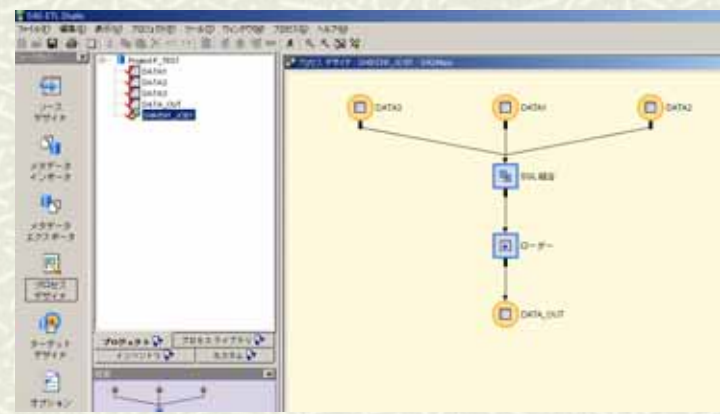
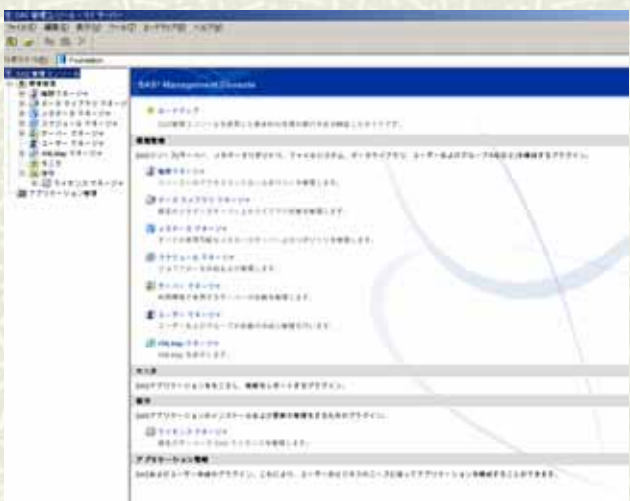
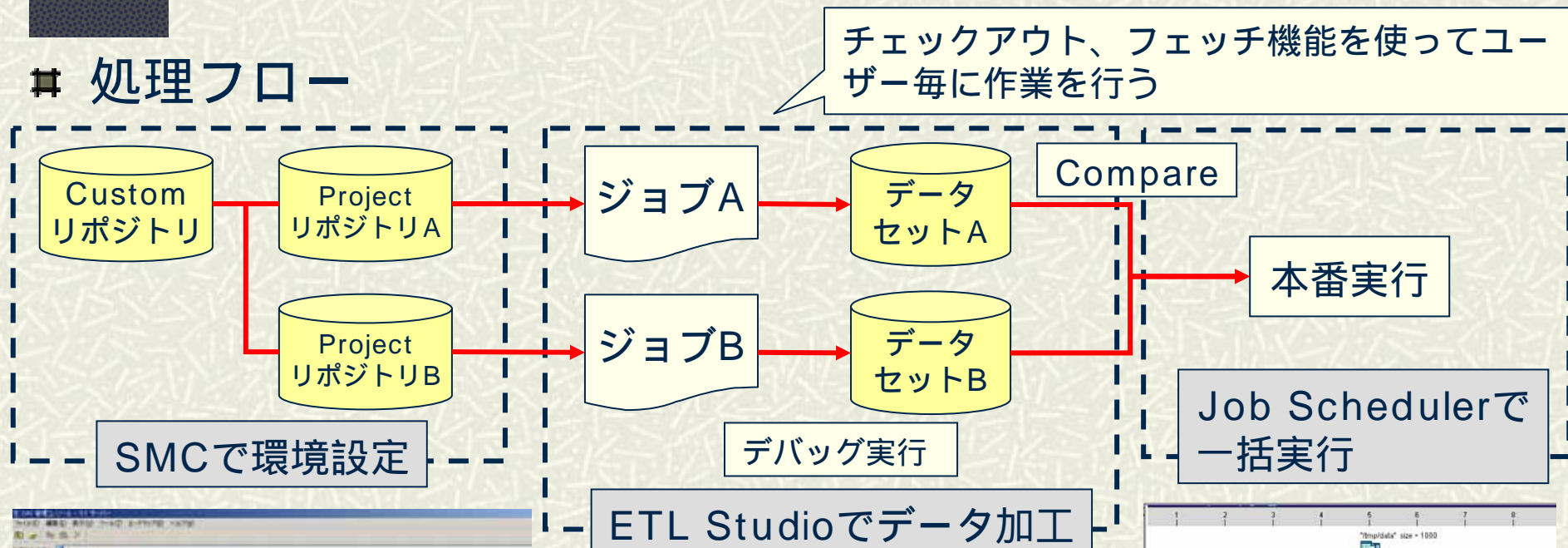
- ジョブの一括実行を行うツール
- ETLで定義したジョブを登録し、一括実行を行う



ローカルPC（クライアント側）で
Administratorの権限が必要！

ETL活用事例（6） - データ加工業務の流れ

■ 処理フロー



今後の展望

- # ユーザー権限の制約への対処
 - 同じユーザーでも作業によって必要な権限が異なる
- # プロセスライブラリの活用
 - 変換プロセス（デフォルトで用意されているマクロ）の活用
 - ユーザー作成コードを新たな変換プロセスとしてETLへ追加・汎用化
- # インパクトアナリシスの活用
 - テーブルレベル・変数レベルでの分析
- # Job Schedulerの本格稼働
 - 次期バージョンでは稼働しそう？