

現行システム可視化における リバースエンジニアリング手法の適用

長年にわたり改修を重ね複雑化・肥大化したシステムを刷新したいものの、現行システムの構造が把握できないために、業務課題の整理やシステム機能との関連付けができないというケースが少なくない。そこで本稿では、ソフトウェアツールによりソースコードから設計情報を抽出することで現行システムを可視化する、リバースエンジニアリングの手法を紹介する。

現行システム可視化の意義

システムの刷新は、まずシステム企画から始まる。システム企画とは、現行の業務課題を明らかにし、それをシステムによってどう解決するかを決めることである。しかし、企業活動の大部分がシステム化され、たとえば与信額や発注量の算出のような業務上の計算ルールや、出荷指示や実績報告といった部門間の連携はほぼシステム処理に埋め込まれている環境下では、現行の業務課題を現場インタビューのようなやり方で整理しようとしてもうまくいかないことが多い。業務フローの広範な領域がシステムでカバーされているため、その実態を管理者のみならず現場担当者すら熟知しているとは言えず、何が問題なのかを指摘できないからである。現行の業務課題そのものが定義できなければ、課題を解決すべきシステム施策も的外れなものになる。

そこで、システム企画の段階で現行システムの構造を調査し、その中に織り込まれた業務活動やルールを可視化することによって業務課題を明らかにするアプローチが有効になる。すなわち、現行システム可視化は現行業務課題の定義の一環なのである。

調査によって発見されるものは課題ばかりではない。一般に日本企業の基幹系システムは既成のパッケージ製品とは異なり、現場ユーザーの要望にきめ細かく対応することで成長してきた。そのため、現行システムを調査する過程で、普段は意識することのない業務ノウハウや知恵が、システム機能という形で発掘される。次期システムを構築する際には、ビジネス価値をもたらすこれらの機能を正しく継承することが必要である。

現行システム可視化における問題点

業務課題を明らかにするための現行システム可視化とは、単にプログラムやデータの物理的な構造・配置を記述するだけでなく、それらが業務的に意味のある機能・情報にどう結び付いているかを明らかにすることである。

しかし、これは容易な作業ではない。企業のシステム環境は、長年にわたる改修や新規システムの追加構築を経てきた。そのため、どの部門のどの業務がどのシステム機能に依存しているのかという関係が非常に複雑化している。また、改修や追加構築の過程で設計書がメンテナンスされず、現行システムに関する知識が特定の個人に蓄積される傾向にあ

野村総合研究所
システムコンサルティング事業本部
システムデザインコンサルティング部
主任システムコンサルタント
山本俊樹（やまもととしき）
専門はレガシーモダナイゼーション、オフショア開発など



る。実際、システム再構築のプロジェクトでは、現行システムの可視化といえは保守・運用担当者へのヒアリングに頼るのが常であった。ところが、いわゆる2007年問題により、頼みの綱である知識をもった人が退職などで保守・運用の現場を離れることが多くなっている。残された手段は、プログラム・データの業務的な意味を人手で地道に読み解いてゆくことである。しかし、一定の業務知識を備え、かつ多様な技術要素・開発方式に通じたスキルのあるSE（システムエンジニア）を確保するには大きなコストを要する。

有効な可視化ツールの活用

現行システムの可視化は、次期システム構築のための事前作業と位置付けられる。にもかかわらず、上で述べたようにかなりの時間とコストを費やす、作業担当者にもプロジェクトマネージャーにも頭の痛い作業なのである。そこで、この作業をツールによって効率化することが必要になってくる。

プログラムのソースコードから抽象度の高い論理的な設計情報を解読する（とくにソフトウェアツールを利用して自動的に生成する）手法は、一般にリバースエンジニアリングと呼ばれる。リバースエンジニアリングの考え方やツールは、これまで開発工程におけるデバッグや、保守・運用工程における影響調査、そして大型汎用コンピュータからオープン系へのプラットフォーム移行に適用されるケー

スが多く、システム企画への適用はほぼ未開拓の分野であった。野村総合研究所（NRI）では、システム企画における現行システム可視化にリバースエンジニアリング手法を適用する取り組みを行っている。

現行システム可視化には、業務活動の順序や制御条件を明らかにする「ビジネスプロセス可視化」、データ項目の意味構造やライフサイクルを明らかにする「データ管理構造可視化」、業務ルールをデシジョンテーブル（条件と動作の組み合わせ表）や計算式の形で表現する「ビジネスルール可視化」という3つの領域がある。

ビジネスプロセス可視化の方法

歴史のある大型汎用コンピュータシステムで典型的に採用されてきたアプリケーション構造は、業務機能の大部分をバッチ処理側で実装し、オンライン系には伝票データ作成のような比較的単純な機能のみ実装するというものである。そのため、ビジネスプロセス可視化のカギを握るのが、バッチ処理のシステムフロー作成である。システムフローとはプログラムの実行順序に沿ってプログラムとデータのIn/Out関係を表現した流れ図である。

システムフロー作成は、以下の3つのステップを経て行われる。

①物理システムフローの作成

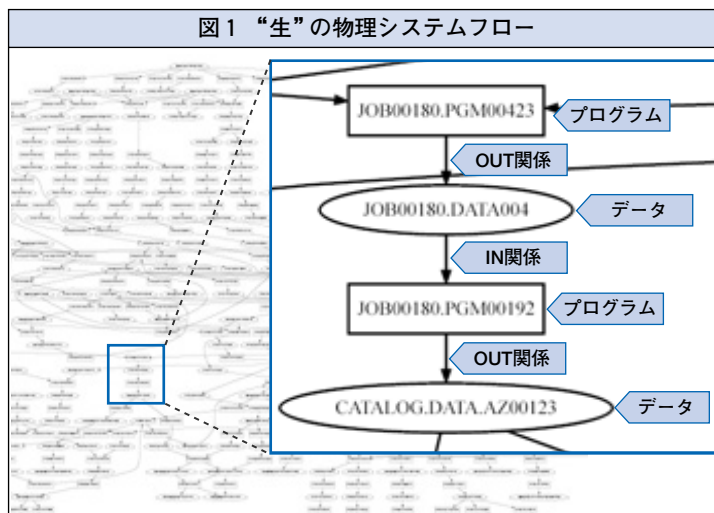
ジョブ運用ツールの定義情報からプログラムの実行順序の情報を抽出し、またJCL（ジ

ジョブ制御言語)やCOBOL(一般業務処理用のプログラミング言語)のソースコードから、プログラムとデータのIn/Out関係の情報を抽出・解析し、物理的な“生”のシステムフローを描画する(図1参照)。このシステムフローは非常に大規模なものになるため、実際にはより詳細なバッチ処理の実行サイクル(日次・週次・月次・年次)で図を分割する。

②物理-論理変換

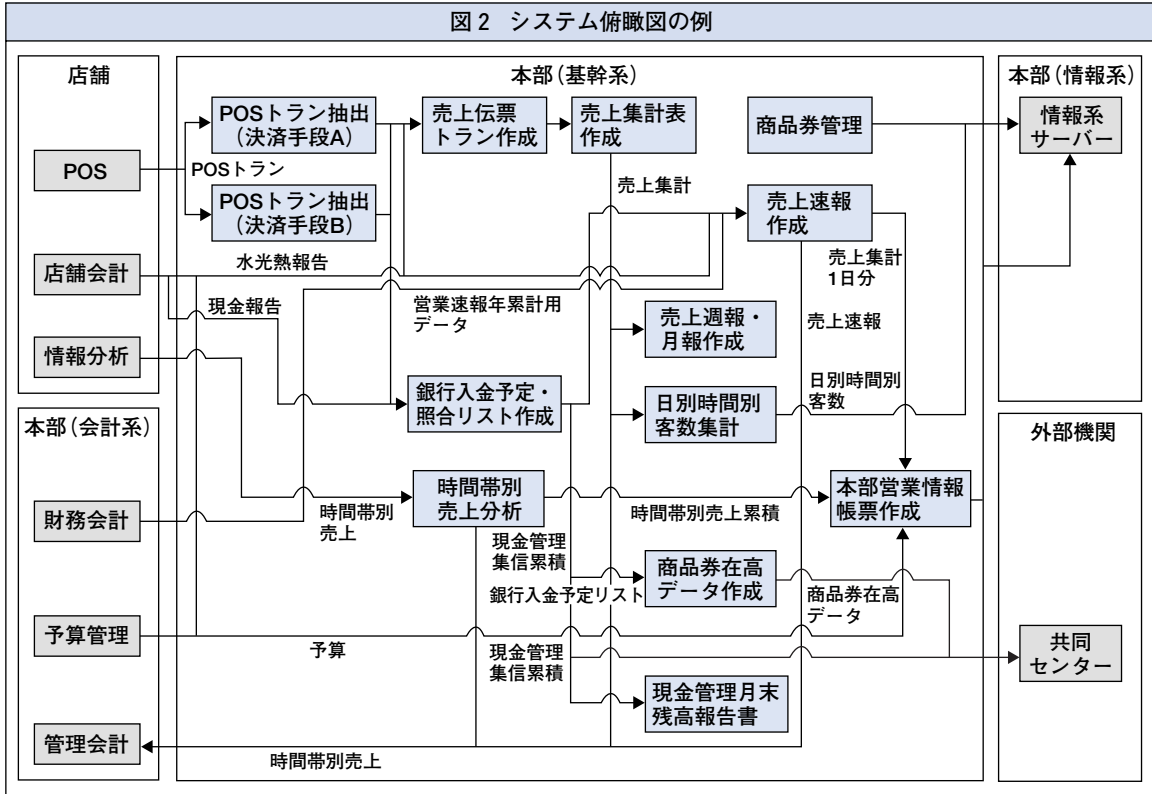
物理システムフローは業務的な意味を表現していないため、これを論理的なフローに変換する必要がある。まず文字コード変換、バックアップといったテクニカルな処理を削除する。こうした処理は市販の共通ユーティリティによって実装されることが多く、プログラム名などの特徴により比較的容易に特定、削除することができる。次に、プログラムとデータに日本語の論理名を付ける。この命名作業、とくにデータの命名作業では資料調査やヒアリング調査が必要になる。とはいっても、設計書がほとんど揃っていないシステムであっても、プログラム管理台帳は日々の保守・運用の観点から整備されていることが通例なので、ここからプログラムIDに対応する論理名を取得することができる。またプログラムソースのコメントから有用な情報が抽出できることも多い。

図1 “生”の物理システムフロー



③論理レベル抽象化

論理システムフローにより、局所的・断片的にはプログラムやデータの意味が読み取れるようになる。しかし、それらが組み合わさってどのような業務機能を果たしているのかは定かではない。そこで、プログラム-データの系列を業務機能の単位でグループ化し、抽象度の高い俯瞰図の形でフローをまとめあげる必要がある。グループ化には高度な意味判断が求められるため、大部分は人手による作業となる。判断の材料として他の可視化領域(データ管理構造可視化、ビジネスルール可視化)の分析結果を活用することも必要である。具体的にはデータレイアウトの類似性や変数項目の編集過程などを参照する。グループ化された業務機能には簡潔な機能名を付ける。最後に業務機能グループを組織や外部システムが配置された台紙に並べ、抽象化されたシステム俯瞰図を仕上げる(図2参照)。



リバースエンジニアリングの有効性

現行システム可視化の成果物は、ビジネスプロセスを表したシステムフローだけではない。以下のように、他の可視化領域においてもリバースエンジニアリング手法の適用は有効である。

データ管理構造可視化に関しては、データベース定義やCOPY登録集の解析により、システム内の膨大なファイルやテーブルのデータレイアウトを分類・整理し、論理データモデルを作成することができる。そしてビジネスプロセス可視化の分析結果と合わせ、業務

機能×概念データのマトリックスを作成する。

ビジネスルール可視化に関しては、変数の演算・操作過程をプログラム横断でトレースするスライシング手法を適用し、業務概念に対応するデータ項目の編集過程を計算式の形で表現する。またデータ項目を判断条件に利用している箇所を特定し、デシジョンテーブルを整理することも可能である。

これまで、ソフトウェアツールによるリバースエンジニアリング手法は、おもにシステム構築の下流過程で使用されてきた。今後は、より上流のシステム企画においてもその適用が進むものと思われる。