6A-6

業務アプリケーションの超高速開発を可能にする 穴埋め方式によるフリーパッケージDE「DataExplorer」の提案

中原 正賢† 渡辺 透‡ 大江 信宏‡ 樋口 雅宏‡
株式会社メルコテクノ横浜‡

1. はじめに

近年、Web 業務システムは複雑に絡み合い肥大化してきた一方、開発にかけられる工数はますます短縮化の傾向にある。その為、コーディング量の削減による生産性向上の方策としては、業務特化型で高速な開発を支援するツールが注目されつつある。例えば IntraMart、GeneXus や Web Performer などであり、その殆どは、画面・プログラムソースジェネレータ類である。それらは DOA（データ指向アプローチ）による設計手法を採用し業務ロジックを設計して業務要件と実装要件をリポジトリに登録することで、Web システム（即ち、業務画面やプログラムソース）が自動的に生成される。小規模で簡単なシステムでは、有効性を実感でき高い生産性を得られるが、中大規模以上複雑なシステムになると、経験的にスクラッチ開発と比べ、ある程度の生産性向上には寄与するが、抜本的な改善が難しいという課題があった。

その解決策として、従来の様な開発ツール類ではない最終成果物（システム）そのものの雛形である自己成長型フリーパッケージを提案する。本稿では、既に 7 年以上実績を積んできた「コンテナー&穴埋め」方式による自己成長型フリーパッケージ「DataExplorer」（以降”DE”と略す）の詳細と実際の効果について報告する。

2. 近年の高速開発ツールの課題

前述の高速開発を支援するツールの概念図を図 1 に示す。

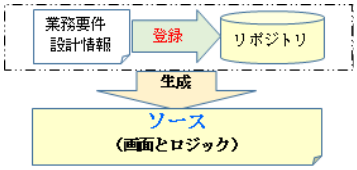

図 1 近年の高速開発ツールの概念図

殆どは、そのルーツが、画面自動生成ツールであり、画面作成の生産性向上が中心となっており、画面やその画面にある項目例えば TextBox や DropdownList の属性（プロパティ）又はシンプルな振る舞い（必須チェックや不正チェックなど）を、リポジトリに簡単な値登録で対応可能であるが、より複雑なビジネスロジックはそのツールの独自言語でプロシージャをコーディングしリポジトリに登録する。尚、業務プロセス管理（BPM）機能は標準装備していないのが一般的である。

その為、確かに、例えば得意先マスター管理などビジネスロジックが比較的少量で簡単な場合にコーディング量をかなり軽減でき、高い生産性が得られる。しかしながら、業務プロセス管理（BPM）をはじめ、複雑で大量なビジネスロジックが要求された中大規模システムの場合、リポジトリに簡単なパラメータ（値）登録では済まず、相応なビジネスルール（プロシージャ）のコーディングが、そのツール上で相変わらず必要である為。スクラッチ開発より 20～30%程度の生産性向上しか得られず、抜本的な改善が難しいという課題があった。

3. 分析と解決策

前章で述べた課題の根源を分析すると、ルーツ由縁のその設計思想にある。つまり、画面(UI)作成以外の必要なビジネスロジックはツールが提供した独自言語や API を駆使してコーディングすることになっている。但し、これではビジネスロジックコーディングの戦場が基層フレームワーク（JAVAEEx や.MS .Net）からそのツールに移っただけのことで、幾らか容易さや効率性が向上できたとしても、似たような画面や機能例えば、販売管理や生産管理それぞれの入力機能、データ参照・検索・表示機能、集計機能など基礎的な機能を毎回個々に登録し作っていたら、重複労働による無駄が多く、生産性と品質の大幅な向上が困難である。解決策としてツール類ではないパッケージ類即ち、「自己成長型パッケージ」DE を開発し生産性と品質の抜本的な向上を図った。

4. ＤＥの詳細

4.1 考え方 開発全般に渡って無駄を排除すること。即ち、(1) 上層における機能面では、普遍的な画面や基礎的な機能の重複作成を排除。(2) 下層における業務要件ロジックのコーディングを排除しノープログラミング（パラメータ定義）化。(3) 定義すると即座に動作確認できること。尚、(1) 項の画面や機能とは、分野問わず例えば、ユーザ認証・セキュリィ、入出力画面、BPM、印刷、メール送受信、ファイルや画像のアップ（ダウン）ロード、マスターなどデータ参照・検索・集計、他システムとの連携、様々なユーザ補助機能などがある。

4.2 実現方式 前述の考え方を実現した DE「コンテナー&穴埋め」方式の概念と構成を図 2 に示す。

図 2 において、構成順①～⑥に沿って下記に説明する。

① 業務要件・設計情報： まず、画面要素やレイアウトを含め業務要件やビジネスロジックなど設計情報を定められた書式で所定箇所「穴」に定義（②リポジトリ登録）しておく。尚、登録の際、所定パラメータ定義以外に任意@Script 記述が可能。@Script とは、④エンジンのコア機能（150 以上の@関数例えば、@Mid(・・・)、@Print(・・・)、@Export(・・・)、四則演算、論理演算や if 文評価など）で提供されている。

② リポジトリ（８００穴）： 定義の集合体のことで一般企業の活動における業務管理やデータ活用に必要な属性（プロパティ）、振舞い（メソッド）や画面要素、レイアウトを含め全てのビジネスロジックを、ジャンルごとにグループ分類し、それぞれパラメータ項目として洗い出し整理することができる。本方式では、それらのすべてを**マトリックス表**で表現しデータベースに格納する。具体的に、１８のテーブル（ジャンルごと）に分類し、全体で約８００のフィールド（パラメータ項目）に分けた。尚、個々のフィールドはパラメータ定義項又は「**穴**」と呼び、登録を「**穴埋め**」と言い、その集合を定義体又はリポジトリと呼ぶ。

Proposal of business application development system DE, that enables ultra high-speed development,characterized by filling system.

† Shoken NAKAHARA
‡ MelcoTechno Yokohama Co. Ltd.
‡ ‡ Toru WATANABE,Nobuhiro OHE,Masahiro HIGUCHI
325 Kamichyoya ,Kamakura City, Kanagawa, Japan 247-8520

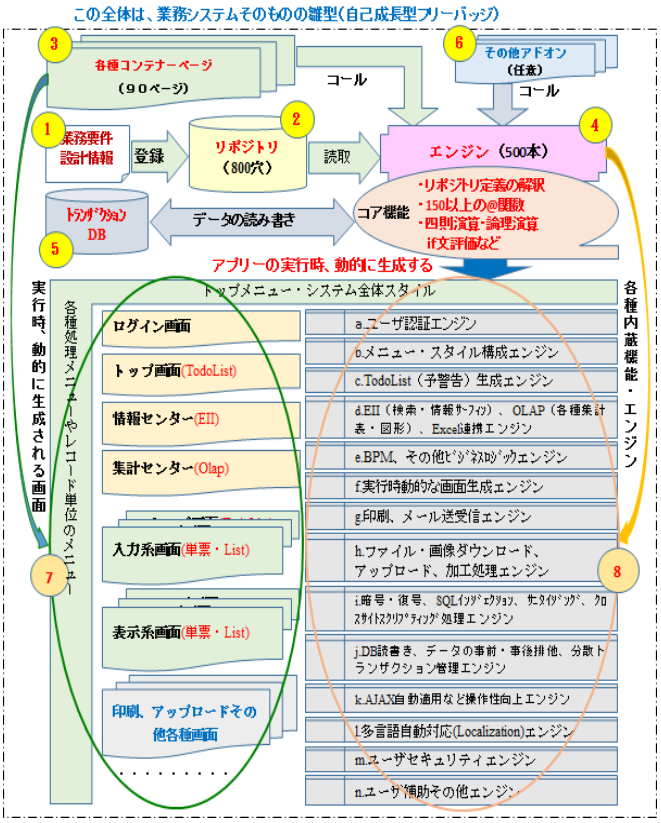

図２　DE「コンテナー&穴埋め」方式の概念図

③ 各種コンテナーページ（９０ページ）： コンテンツの受け皿のことでデータを入力または表示する為の UI ページのことであるが、**一般のページやテンプレートページと違って**、所定の幾つか「エンジン」をコールする箇所又は普遍的な基礎機能ロジックを除けば殆ど「空」である由縁、コンテナーページと呼ばれ、実際のコンテンツは、そのコンテナーページがコールされロードされる時に、④エンジンが②リポジトリから当該定義情報を取出し解釈しながら生成し動的に図２の⑦が示す諸対象ページを形成する。通常は、入力や表示など用途別のコンテナーページがあり、全部で約 **90** ページある。一例を取上げると、検索結果を表示するコンテナーページの一つ Y40List.aspx は、図３が示す通り、基礎的な普遍機能である ⓐ列固定 ⓑ一括データ確定 ⓒダウンロード ⓓリフレッシュ機能など以外にその中身が殆ど「空」であるが、

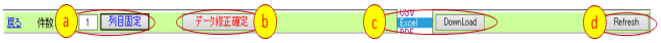

図３　List コンテナーY40List.aspx ページ

それがオープンすると、諸エンジンの一つである⑧内の f.**実行時動的に画面生成エンジン**がコールされ、②リポジトリの当該定義を解釈しながら⑤DB から対象データを取得し、図４が示すページを動的に形成し即座に表示する。

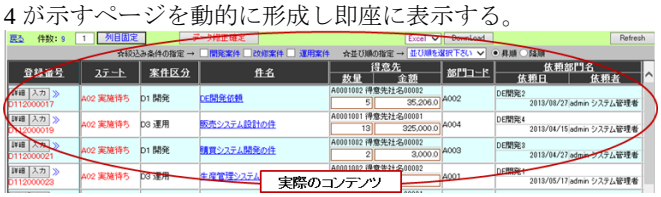

図４　動的に形成され表示された検索結果ページ

④ エンジン： ②リポジトリ（800 穴）に登録されていた定義情報の解釈、コア機能即ち独自言語である@Script（**150** 以上の@関数、四則演算、論理演算や if 文評価など）の提供と実行時動的なコンテンツ生成とロジックの実行。尚、図２内の⑧が示す様な種類の機能・エンジン(**500 本**)があり、実際のロジックは②リポジトリの定義情報によって可変される。

⑤ トランザクション DB： 対象データベースのこと。

⑥ その他アドオン： 「コンテナー & 穴埋め」方式で対応出来ない場合に開発者が自由に開発した内外部のページやライブラリー（ロジック）の任意取入れ。

4.3 近年の高速開発ツールとの相違点

近年の高速開発ツールの最終開発成果物は、ソース（画面・ロジック）であるのに対して図２が示した通り DE は、**ソースを吐き出さず、業務システムそのものの雛型であり**、前述の「穴埋め」を行うと即座に結果が動的に生成され確認できる為、「**穴埋め⇔即座確認**」を繰返すと業務システムがどんどん成長しやがて完成する。その由縁又**@Script** による高い自由度から、**自己成長型フリーパッケージ**と言う。

5. 効果

5.1 生産性の向上　一般的に前述の様な **90** 以上の普遍的な画面と **500** 以上の普遍的基礎的な機能が業務システム全体の約 **80%**以上占めている為、DE はそれら個々の開発を不要にしたことで（必要時の可変が可能）、生産性の大幅な向上を実現している。DE と近年の高速開発ツールそれぞれがスクラッチ開発より生産性の向上率を表１に示す。

スクラッチ開発より生産性の向上率対比表	小規模システム	中大規模システム
DataExplorer(DE)	90%以上	80%以上
近年の高速開発ツール	約 80%	20～30%

表１　DE と近年の高速開発ツールの生産性向上率

又、「**穴埋め⇔即座確認**」繰返しの特質により、アジャイル開発に最適であることは、長年の実績にて実証されている。

5.2 品質の向上　個々の画面や機能（ロジック）の作成と開発が 80%以上削減され手によるコーディング量が激減された為、品質の向上と維持に大きく寄与できる。尚、試験については、従来「単体試験⇒結合試験」スタイルが不要となり、開発の土台（環境）は、開発対象である「業務システム」そのものなので、最初から結合されていて試験というよりも動作確認（「穴埋め⇔即座確認」）になる。従って、従来設計、試験やドキュメントに費やされる労力と時間も大幅に軽減できる。

5.3 二つの三位一体　一つ目の「BPM & EII & Olap」三位一体によってエンドユーザが高価な Olap などツールを別途購入する必要がなく同一システムでシームレスな情報サーフィンと様々必要な集計・データ分析が、簡単に出来る。二つ目の「開発 & 稼働 & 保守」三位一体によって、エンドユーザ自ら、同一システムでマスターメンテナンスと同じ様な感覚である程度のシステム改修・改善・拡張が、簡単に出来る。

5.4 標準性・一様性　DE は設計者や開発者に依存せず、システムスタイル、構成、画+面・機能やオペレーションなどは、業務パッケージの様な高い標準性・一様性が実現されている。

6. まとめ

DE は、**個々のプログラムの様なミクロ粒度レベルの事項**を隠蔽し、業務システムの上層における基礎的又は普遍的な画面や機能の重複作成を不要にしたと同時に、開発対象である業務システムそのものなのでエンドユーザと同じ**視点（同じ土台）**で開発できることによって前述の効果をもたらし業務システム開発の考え方や在り方への変革に貢献できる。