

B-022

データディクショナリ共用による効率的な多言語プログラム自動合成手法

The Efficient Method for Automatic Synthesis of Multiple-language Programs
by Sharing Data Dictionary

山村 喜恒[†] 鹿島 啓吾[†] 西本 幸治[†] 山口 潔[†] 秋元 敏夫[†]
Yoshihisa Yamamura, Keigo Kashima, Kouji Nishimoto, Kiyoshi Yamaguchi, Toshio Akimoto

1. はじめに

近年、業務システムの基盤技術としてインターネットを利用した Web アプリケーション開発がトレンドとなっている。Web アプリケーション開発言語としては、Java™ が採用される事例が増える傾向にあるが、特に既存システムからの移行などでは、オンラインは Java™、バッチは COBOL のように既存システムのコード流用のメリットを生かした構成をとることも多い。本稿では、このように複数のプログラミング言語から構成されるシステムを効率よく製造する生産技術について紹介する。

2. Web システム開発とプログラミング言語

Web アプリケーションを構築する場合、Perl や C による CGI や、Servlet による構築など様々な手法があるが、一般的に CGI の場合はプロセス起動によるパフォーマンスの悪さやセッション管理ができないなどの問題点が指摘され、基幹業務アプリケーション開発には向かないとされている。

このような状況を踏まえ、弊社生産技術部門では、J2EE(Java 2 Platform, Enterprise Edition)アーキテクチャによるアプリケーション開発パターンを導入し、特に大規模や高信頼性を要求される Web システムは、JSP/Servlet + JavaBeans、または JSP/Servlet + EJB(Enterprise JavaBeans)による構成を推奨している。このため、Web アプリケーションは Java™ で開発するケースが多い。

しかしながら、前述したように、既存システムのコードを流用できる場合や、大量データの処理においては、まだまだ COBOL に一日の長がある。特にイントラネットで使用する Web アプリケーションは、24 時間オンライン稼働を要求されるわけでもないため、既存システム構築のノウハウ、資産、技術者を活かし、オンラインは Java™、バッチは COBOL のように 適材適所に配置する構成も有効と言える。以下、本稿ではこのような形態のシステムを多言語複合システムと呼ぶ。

3. 多言語複合システム開発の生産技術

3.1 コンポーネント指向

弊社生産技術部門では生産性向上技術として、プログラムで使用するデータ項目をデータディクショナリに統合管理し、プログラムの基本構造(スケルトン)を有するテンプレートをベースにプログラムソースを自動生成するというアプローチを取ってきた。すなわち、スケルトンの単位を最適化し、プログラムの自動生成率を向上することで、生産性向上に結びつける設計方針である。

スケルトンの単位としては、アプリケーションの 3 層構造、すなわち、プレゼンテーション(P: Presentation)層、ビジネスロジック(F: Function)層、データベースアクセス(D: Data Access)層のコンポーネントからなる枠組みを基本と考える。各コンポーネントは、それぞれ上位のプログラムから共用され、再利用されることを目的としている。

3.2 多言語複合システムへの応用

ここで、多言語複合システム開発におけるコンポーネント指向のあり方を考える。ユーザインタフェースを既定する P 層(バッチプログラムの場合は、バッチメインプログラム)は、オンラインプログラムとバッチプログラム間でシステムの利用形態が異なるため、設計情報を共有することは難しい。しかし、業務処理の中心を担う F 層や、データベースアクセスを行なう D 層については、コンポーネントの単位を最適化することで、“完全に”共有できる機能が少なからず存在する。つまり、プログラミング言語が異なっても、システム全体で設計情報を相互共有可能な範囲は十分大きいといえる。

本手法を適用するには、ビジネスロジックやアクセスパターンの洗い出し、洗練化による集約が重要になってくる。これは、システム全体を通じて各処理層を管理するクラスマスタを置き、無駄な重複開発を排除することで実現する考えである。

[†]株式会社 日立製作所 情報・通信グループ生産技術本部,
Hitachi,Ltd. Information & Telecommunication Systems.

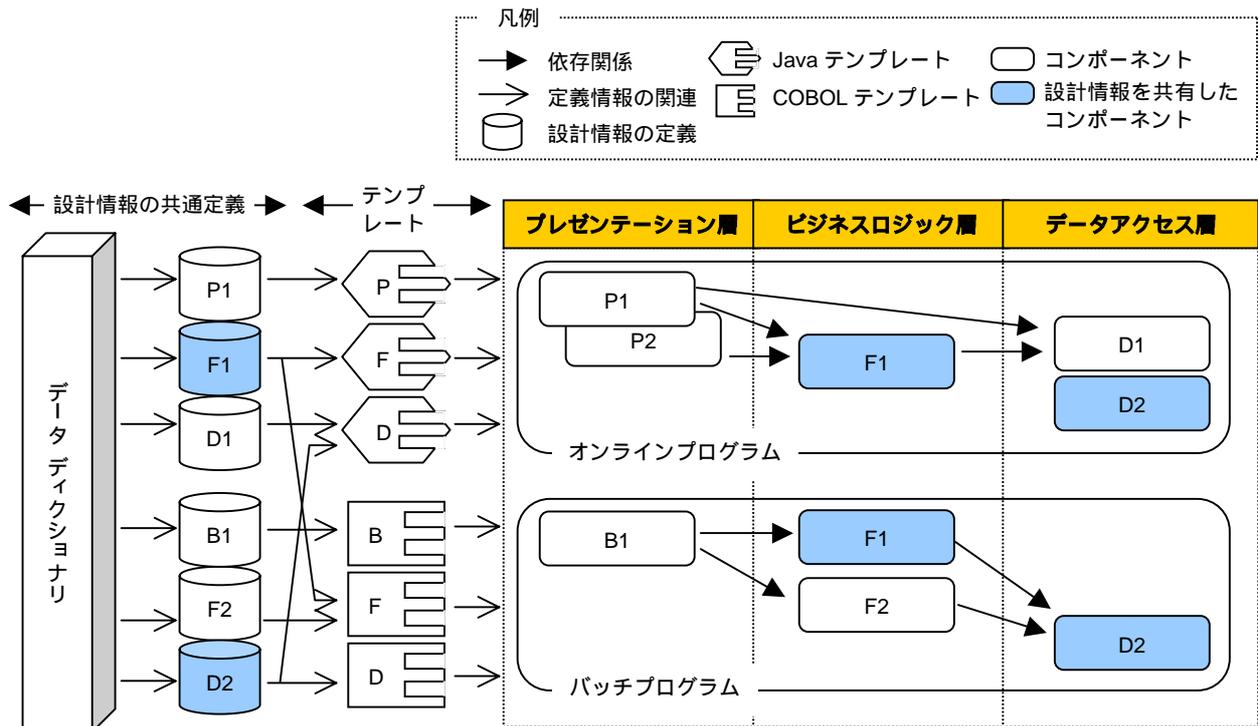


図1 データディクショナリによる多言語複合システム開発

3.3 . 開発手法

多言語複合システムの具体的な開発手法としては、弊社で従来から培ってきたデータディクショナリによるプログラムの自動生成技術を応用する。まず、ビジネスロジックやアクセスパターンから洗い出した設計情報を自社製開発支援ツール SEWB+ † の定義情報として登録する。次に、この定義情報と、各プログラミング言語の構造に応じたテンプレートとを組み合わせ、半完成ソースを自動生成し、業務処理の必要性に応じて追加コーディングを行なう。プログラミング言語間の差異、つまり、インタフェース定義やプログラム構造などの違いはテンプレートで吸収する。このようにして作成した SEWB+ の定義情報は、オンラインプログラム内はもちろん、バッチプログラムも含めてシステム全体で再利用し、生産性の向上に結びつける。

また、既存システムの COBOL 資産が生かせる場合は、SEWB+ のリバース機能を使い、プログラムソースから設計情報へのフィードバックを行なう。

このように、多言語複合システムの開発においても、各プログラミング言語のコードの粒度や構造を合わせることで、複数の言語、すなわち、COBOL と Java™ の差異を可能な限り埋め、システム全体の生産性向上を狙うことが可能であると結論付ける。

4 . まとめと今後の課題

今回、データディクショナリの共有化による多言語複合システムの生産技術を発表することにより、システム全体としての最適な開発スタイル、及び生産技術の向上施策を提唱できたと評価する。

現在のところ、本開発スタイルを実施した事例は存在しないが、Java™ による Web アプリケーション開発や COBOL による C/S システム開発事例は多数存在するため、本提案を適用した場合の生産性向上効果の評価を実施する予定である。

参考文献

- [1] 斎藤岳 他：DOA を取り入れたコンポーネント指向開発手法、FIT2003.
- [2] 宮崎肇之 他：アプリケーション開発スタイルの標準化、情報処理学会第 63 回大会,2001.

† SEWB+ : Software Engineering Workbench + 基幹システムに要求される高品質、高生産性を実現するための機能を提供する CASE ツール