

Java 言語へのソフトウェアコンバージョンに関する研究

2W-5

池田 孝雄

小林 要

金沢工業大学

1 はじめに

インターネットの利用拡大に伴い、既存のソフトウェア資源をネットワークアプリケーション用に移行する機会が増えている。

本研究では、C++や Delphi で作られたスタンドアロンで動作するプログラムを Java プログラムへ移行するコンバージョンシステムを提案する。本コンバージョンシステムでは、加工データをデータベースに蓄積し、ユーザーの要求に応じた構造の Java プログラムへ加工する。

スタンドアロンのプログラムを Java にコンバージョンすることにより、スタンドアロンのプログラムの場合には、限られた人々しか使用できなかったサービスを、web 上からアクセスが可能なアプレットなどを利用することにより、不特定多数の人々にそのサービスを提供できる。また、Java はプラットフォームに独立なため、サービスできる範囲をさらに拡大できる。

例えば、図書検索システムの例では、その図書館を利用する人しか使えなかった図書検索サービスを、web 上に公開することにより、不特定多数の人々に遠く離れた場所から検索サービスを利用してもらうことができる。

C++から Java へコンバージョンする既存のツールとしては、c2j [1] が存在する。c2j は、単に 1 対 1 の対応で変換しているため、入力された C++プログラムに何の機能も加えずに Java プログラムを出力している。

本研究で提案するコンバージョンシステムは、コー

ドだけを言語変換するばかりでなく、ユーザーの要求に応じて、様々な機能を追加した Java プログラムを生成するアップグレードコンバージョンを目的としている。

2 コンバージョンの分類と方法

コンバージョンの目的分類としては、最もニーズが高いと思われる 4 つが考えられる。

1. スタンドアロン型 Java のアプリケーション

基本的なコードの部分は、c2j のように変換する。GUI、DB、標準ライブラリは、あらかじめ対象となる各部品に対応関係を決める。同じ機能が無い場合は、別の機能で代用させる方式をとるか、削除する。GUI については、AWT [5] と Swing [2] のどちらも変換可能な場合は、より高度な GUI にするため Swing に変換するのが望ましい。

2. クライアントアプリケーション/サーバーアプリケーションシステム

クライアントとサーバーに機能を分ける。通常は、GUI はクライアント側、DB はサーバー側に持たせる。アルゴリズムの部分は、構造を解析して、サーバー側に隠蔽したいアルゴリズム部分を分ける。アルゴリズム部分は、なるべくクライアント側に持たせる部分を多くすることにより、サーバー側の負担を軽減することができる。分離後は、アプリケーションの場合には、RMI [6] により双方を接続する。GUI については、アプリケーションの場合、Swing を利用できる。サーバー側をマルチユーザー対応型に変えるには、スレッドを生成して、処理を対応付ける分配アルゴリズムを追加する。

3. クライアントアプレット/サーバーアプリケーション

最初に上記1のコンバージョンを行う。GUIはアプレットに含める。ただし、Swingがアプレットでは使用できないため、AWTに変換する。AWTにない機能については、代用の別機能を当てる。アプレットとサーバー間をソケットで結ぶよう構造を分離する。

4. クライアントアプレット/サーバプレットサーバー

上記3の加工後、サーバーアプリをサーバプレットに加工する。

その他、組み合わせによって多くのコンバージョン方法が存在する。

3 コンバージョンシステム

ソフトウェアの加工ツールとしては、RPS [3] [4]を用いる。ユーザーの要求に応じて、様々なコンバージョンを加工ルールを与えることによって、実現することができる。加工ルールは、できる限り再利用が可能なものにする。そのため、C++などのプログラムを解析したあとに、対象となるプログラミング言語の共通部分を抽象化し、共通の構造に変換することにより、抽象化後のコンバージョンの道筋は一本化できる。

本研究で提案するコンバージョンシステムの加工手順は、以下の順番で行う。

1. C++やDelphiのプログラムを抽象化する。
2. 抽象化した構造を共通の構造に加工する。
3. 共通化した構造を解析して、GUI、DB、標準ライブラリ、アルゴリズムの各部分に切り分ける。
4. スタンドアロンのJavaにコンバージョンする。
5. クライアントアプリケーションとサーバーアプリケーションに分ける。
6. 4の後に、クライアントをアプレットにコンバージョンする。
7. 6の後に、サーバーをサーバプレットにコンバージョンする。

これより、4つの構造を持つJavaプログラムを同時に生成できる。

本システムでは、加工ルールを洗練、追加できるので、いろいろなバージョンのJavaにも対処することができる。また、共通構造からの加工ルールを作れば、Java以外のプログラム言語へも移行ができる。加工ルールを洗練することによって、より良いコンバージョンシステムに進化させることができる。

4 おわりに

本研究では、ユーザーの要求によって様々な構造のJavaプログラムを出力するソフトウェアコンバージョンシステムを提案した。コンバージョンするプロセスで共通化する加工を施すことにより、加工ルールを効率良く準備できる。また、Java以外の言語への移行も可能な拡張性の高いコンバージョンシステムにすることができる。

参考文献

- [1] CHRIS LAFFRA 著, 長尾 高弘訳: advanced JAVA, 株式会社ロングテール (1997)
- [2] 植田 龍男: Java のからくり, IDG コミュニケーションズ (1999)
- [3] 小林 要: RPS 機能マニュアル, 金沢工業大学 (1996)
- [4] 小林 要, 木村 高久: 個人言語によるプログラム作成システム, 情報処理学会ソフトウェア工学研究会報告書 pp.81 (1993)
- [5] Paul T'yma, Gabriel Torok, Troy Downin 著, 吉川 明広訳: Java パワープログラミング, SOFTBANK (1996)
- [6] Elliotte Rusty Harold 著, 戸松 豊和 監訳, 株式会社 ユニテック 訳: JAVA ネットワークプログラミング, オライリー・ジャパン (1997)