Web・DBアプリケーションの自動生成に関する一提案*

6Q−1

荻野 永策　小林 要†
金沢工業大学§

1. はじめに

本研究では，JAVAを用いたWeb・DBアプリケーションを生成するジェネレータを開発した．本ジェネレータは，（1）ソフトウェア開発において機能追加・削除を容易にする，（2）入力テキストを機能追加の単位で記述するという2点の特徴を持つ．本研究では，金沢工業大学内で実際に用いられるシステムを本ジェネレータにより開発し，ジェネレータの信頼性を確認した．

2. 機能追加型開発

機能追加型開発とはシステムの固定部分に変動部分を逐次追加する開発方法である．固定部分はシステムが動作するのに最低限必要な部分である．変動部分は機能追加する際の追加･削除の単位である．機能追加の概念を図1に示す．追加機能は，互いに共有する機能を持つことが多いため，共有機能ごとに複数機能をまとめた単位を追加単位とした．機能追加する場合にはさまざまな個所に変更が生じるが，本研究ではジェネレータを使用して機能追加の記述を局所化している．自動生成の方法[1]は，追加単位を記述した入力テキストから変動部分を生成し，固定部分に組み込む．各機能同士，各共有機能同士が依存している可能性があるため追加・削除の際には依存関係を管理する必要がある．機能追加・削除の場合の依存関係とは，①新しく追加する機能からすでに追加されている機能への依存，②先に追加した機能から後に追加した機能への依存，③先に追加した機能と後に追加した機能の相互依存の3つである．本研究では①のみの依存関係を持つ機能追加に限定した．②は機能Nの機能N+1に対する依存であるため，機能N+1が追加されていないとき，機能Nは動作しない．解決策としては，機能追加の順番を逆にすることで①と同様の依存関係になる．③は依存関係がループしているため機能追加を複雑にする．この結果，機能Nの機能N−aに対する依存のみとなる（0<a<N）．共有機能の依存関係も同様である．

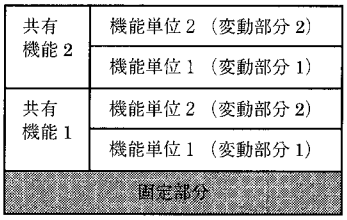

図1　機能追加型開発の概念

3. Web・DBアプリケーションの変動・固定部分

本研究では変動部分を以下のようにした．Web・DBアプリケーションが1要求に対し1応答という特徴を持つため機能単位をイベント単位とした．また，システム構成はアプレット・リスナ[2]・サーブレット[3]を用いる系を対象とした．追加単位としては，アプレットを共有機能に対応付け，1イベント・1サーブレットのイベント処理系を機能単位とした．Web・DBアプリケーションの共有機能と機能単位を図3に示す．複数の機能単位の組を追加単位とし，追加単位の数だけ入力テキストを記述する．

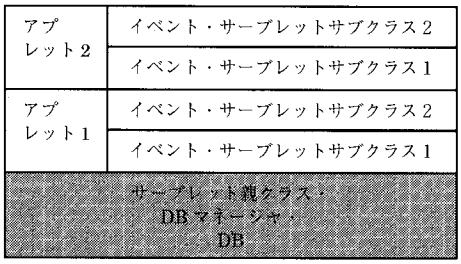

図3　固定部分・変動部分

*A consideration on a Web･DB application generator
† Eisaku Ogino , Kaname Kobayashi
§ Kanazawa Institute of Technology
7-1 Ohgigaoka Nonoichi Ishikawa 921-8501, Japan

次に固定部分の説明をする．サーブレット親クラスは，サブクラスを機能単位として扱いやすくするために抽象クラスとし，サブクラスに実装を持たせた．DB マネージャは DB 接続を行うクラスであるため，各サーブレットサブクラスの独立性を高めることができる．サーブレット親クラスと DB マネージャにより，各機能単位の独立性が向上する．

4. 入力テキストの形式

本研究では，入力テキストを木構造の形式に変換し，木構造レベルでの加工を行い，固定部分に変動部分を組み込み，自動生成している．記述方法は，SQL 文以外は木構造の形式で記述する．入力テキストの例を図 4 に示す．例では，追加単位が 1 つの場

```
画面(
  名前:Get,･･･
  内容:並び(
    既存リスナ付きテキストフィールド
      (名前:tf･･･,既存リスナ名:sample),
    テーブル(名前:table,･･･),
    ボタン(名前:sample･･･,
      リスナ ARG 群:リスナ ARG 群(･･･),
      動作:データベースアクセス(
        パラメータ:パラメータ群(
          パラメータ:変数(･･･)),
        変数取得先:変数取得先(
          日付文字列(･･･)),
        回答先:table,
        SQL 文:文並び(
          [SELECT 代表者名 FROM "代表者表";]
        )))))
```

図 4　入力テキストの例

合である．入力テキストでは，最初にアプレットの名前やサイズ等を記述する．次に，内容：並び（）内に使用する GUI コンポーネントを記述する．各 GUI コンポーネントは，名前や座標，サイズ等を記述する．機能追加の記述に関しては，ボタン単位で行う．ボタンを使用する場合は，ボタンに対するイベントを記述する．依存関係にある機能の追加は，依存先のイベントトリガ名等を記述する．図 4 では，ボタン sample で使用しているリスナ・サーブレットをテキストフィールド sampleTF で使用する場合の例である．

5. 信頼性と機能追加容易性

本ジェネレータでは①あるイベントが発生するとどのような SQL が発行されるか等を確認しやすくするためにイベント・サーブレットの対応確認テキストを，②依存関係にある GUI コンポーネントや機能を洗い出し，依存関係を確認するために依存関係確認テキストの 2 つをそれぞれ自動生成している．①，②を生成することで，依存関係やイベントとサーブレットの対応が確認しやすくなるため，信頼性と機能追加容易性の向上につながる．

6. おわりに

本ジェネレータの利点は，(a) 機能追加・削除が容易，(b) 固定部分は動作確認済み，(c) イベント・サーブレットの対応確認が容易，(d) 依存関係がループしている機能の排除により，プログラムの複雑化を防ぐの 4 つである．本ジェネレータの欠点は，（1）HTML が使用できない，（2）依存関係がループしている機能の追加ができないの 2 つである．（1）では，HTML はリスナを使用できないため，イベント単位での追加はできない．加えて，リンクなども絡んでくるため，依存関係を管理することが難しい．（2）では，機能同士が依存し合っているため，機能追加・削除の際に複雑になる．

本研究では本ジェネレータにより実際に本大学で使用される Web・DB アプリケーションを開発し，システム全体の 7 割をカバーした．3 割は HTML を使用したため自動生成後に手で開発したプログラムを追加した．今後の課題はジェネレータのカバー率を向上させることだが，依存関係がループしている機能の追加については制限があってもよいだろう．

参考文献

[1]：小林要：文法や加工手順を入力データとするテキスト加工システム RPS の開発，情報処理学会第 63 回全国大会，講演番号 6Q−02，2001 年

[2]：上田龍男：Java のからくり−オブジェクト指向入門−，(株) IDG コミュニケーションズ，(1999)

[3]：原田洋子：Java Servlet 最新サーバ・プログラミング，(株) 秀和システム，(1999)