

SGMLに基づくデザインパターン利用支援システムの ユーザインターフェース

6AE-2

瀬川 淳一

大月 美佳

吉田 紀彦

牧之内 顕文

九州大学大学院システム情報科学研究科

1. はじめに

近年、オブジェクト指向が提供する再利用の枠組みを効率よく利用できる手法としてデザインパターンが注目されている。しかし現在のところデザインパターンの示す意味的構造を反映した構造的な記述形式と、それに基づくデザインパターン利用のためのシステムは存在していない。

我々はソフトウェア協同開発支援を目指して、オブジェクト指向ソフトウェア部品の統合的かつ分散的なリポジトリの構築を進めている^[2]。その一環としてデザインパターンを抽象度の高いソフトウェア部品としてとらえ、我々のシステムに統合しようと研究を進めている。^[3, 4]

その初段階として現在デザインパターン利用支援システムを作成している。この支援システムは、デザインパターンの説明文と意味的構造とをSGMLで記述しこれをもとにデザインパターンの管理、表示およびソースコード生成支援を目的としている。分散環境に対応できるように実装はJava言語で行い、システムの出力はHTMLやJavaのAppletで行う。ユーザはWebブラウザを通してシステムの出力を受け取る。

本稿ではSGMLによるデザインパターンの記述と支援システムのデザインパターン表示の説明を行う。

2. デザインパターン

デザインパターンとはオブジェクト指向ソフトウェアに繰り返し現れてくる特徴的な類型を抽出、解析し体系的に名前付けし、自然言語による説明とデザインパターンの構造を示す図を加えたものである。

説明における記述項目としては、名称、別名、目的、動機、適用可能性、構造、構成要素、強調関係、結果、実装、トレードオフ、コード例・使用例、関連パターンがある^[1]。

3. SGMLによるデザインパターンの記述

システムが目的を満足するために、デザインパターンを記述する際にはその説明文と構造を示す意味とを同時

に記述できなければならない。よって我々は意味構造を反映させて文書を構造化できる規格SGML^[5]を採用することにした。そして、説明文とパターンの意味的構造とを同時に記述できるようにSGMLを設計し、これをPIML (Pattern Information Markup Language) と名付けた。

3.1 基本的枠組

PIMLでは基本的に各記述項目ごとにブロック化し構造文書化する。目的、動機などのように内容が説明文の場合はそれに対応したタグを定義しそれで囲んでブロック化し、名称、別名などのように内容が数語程度のものはタグの属性値として記述する。

説明文は、そのテキストをそのまま対応するタグに囲まれた中に置くが、その際「関連パターン」などのテキスト中に参照がある場合には、リンクを「抽象アンカー」の形で埋め込む。これについては第4.1節で述べる。構造についてはパターン構造の示す意味をタグの階層構造で記述する。詳しくは第3.2節で述べる。

3.2 構造情報の記述

PIMLの構造記述は、表示時のOMT図の自動生成およびコード生成の元データとして用いる。

その内容は、関係定義群、役割定義群、配置情報の3つに分けられる(本研究ではデザインパターンでのクラスをシステムの中で果たす役割としてとらえ、実際のクラスと区別するために「役割(role)」と呼んでいる)。

関係定義群では、各役割間に成立する継承、参照などの関係を、関係の名称、関係の起点と終点で明示する。

役割定義群では、役割の名称、メソッド、メソッドの引数などがそれに対応するタグを階層的に組み込むことにより表現している。

配置情報にはOMT図での役割を示す「箱」の配置情報がはいる。具体的には図を行列と見たて図全体の行数・列数、各役割の箱が配置される行・列を記述する。

例として以下にIteratorパターンのPIMLを示す。

```
<pattern name="Iterator" alias="Cursor">
  <intent>
    目的の説明文
  </intent>
  ...
  <structure>
    <relations>
      <inheritance origin="ConcreteAggregate"
        target="Aggregate">
```

User Interface of Aiding System for Design Patterns Based on SGML

Jun'ichi Segawa, Mika Ohtsuki, Norihiko Yoshida, Akifumi Makinouchi

Kyushu University, Graduate School of Information Science and Electrical Engineering

```

...
</relations>
<roles>
  <role syslabel="ConcreteAggregate">
    <operations>
      ...
    </operations>
  </role>
</roles>
<layout rows="2" columns="3">
  <box name="Aggregate" row="1" column="1">
    ...
  </layout>
</structure>
</pattern>

```

4. デザインパターンの表示

システムはデザインパターンを表示する際、説明文をHTML、構造を示す OMT 図を Java の Applet でそれぞれ出力する。システムは CGI により起動され、ユーザから要求をうけた項目を出力する。

4.1 HTML の生成

記述項目の一覧が表示される（目次に相当する）ページは、該当項目のデータの存在を確認して、その項目をシステムに出力させるような CGI のタグを列挙する。

通常の記述項目のページは要求を受けた項目データを取りだし、これに HTML のヘッダ部分を付加して HTML を作成する。

目次のページと記述項目のページにはどちらもシステムを起動させる CGI タグが含まれることになる。しかし、このタグを PIML に直接埋め込むのは CGI の名前の変更等があった場合すべての PIML を変更しなければならないので不適切である。よって「抽象アンカー」という説明文の中では決して現れることのない識別子（現在は##）で囲まれた文字列を定義し、これを CGI を呼び出すタグへと置換する。

例えば PIML 中で

```

##p:Iteraotr## pattern
は HTML として出力されるとき
<a href="http://webserver/cgi-bin/
  getcomp.cgi?pattern&Iteraotr&index">
Iterator </a> pattern
へと変換される。

```

4.2 OMT 図の生成

OMT 図生成は次のような行程で行う。

- 1 名称・関係・「抽象」か否か、配置の情報を PIML の構造情報から抽出。
- 2 各箱の大きさをその内部に記入する文字列の大きさにより決定。
- 3 配置情報、箱の大きさ、あらかじめ設定されたマージン値より箱の最適な絶対座標の決定。
- 4 適切な折れ線形状の決定や線の重なり回避を行い、箱を矢印で連結

生成例として Iterator パターンの OMT 図を Java の Appletviewer で出力させたものを図 1 に示す。

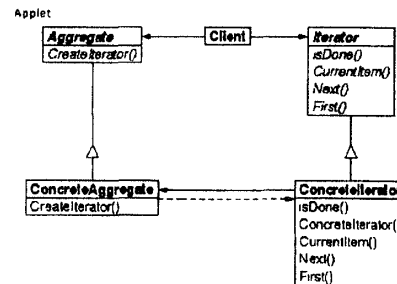


図 1: Iterator パターンの OMT 図の表示例

5. おわりに

今回は、デザインパターンの記述文法として設計した SGML である PIML と、これをシステムが読み込み表示データへと変換する過程を説明した。

これにより PIML で記述されたデザインパターンは説明文、OMT 図がシステムにより動的に生成される。

これまでの成果としてはデザインパターンの表示機構である説明文の HTML による出力、OMT 図の Applet による出力が完成した。現在はコード生成部におけるユーザインターフェースの実装を行なっている。

参考文献

- [1] E.Gamma, R.Helm, R.Johnson, and J.Vlissides. *Design Pattern: Elements of Reusable Object-Oriented Software*, Addison-Wesley (1995); 本位田他訳, オブジェクト指向における再利用のためのデザインパターン, ソフトバンク (1995)
- [2] 大月 美佳 他, “オブジェクト指向ソフトウェア開発支援のための分散部品リポジトリ”, ソフトウェア工学の基礎 II (ソフトウェア工学の基礎ワークショップ FOSE'95 論文集, 大蔭編) 近代科学社, pp.207-212 (1996)
- [3] 大月 美佳 他 “SGML によるデザインパターンの構造化文書化”, オブジェクト指向最前線 (情報処理学会オブジェクト指向シンポジウム OO'97 論文集, 岸・中所編) 朝倉書店, pp. 31-38 (1997)
- [4] Ohtsuki,M., Segawa,J., Yoshida,N. and Makinouchi,A. “Structured Document Framework for Design Patterns Based on SGML”, *Proc. COMPSAC '97*, to appear(1997) .
- [5] ISO: *Information Processing - Text and Office Systems - Standard Generalized Markup Language(SGML)*, ISO 8879(1986).