

オブジェクト指向フレームワークの ホットスポット探索ナビゲーションツール

1 L - 1

瀬野尾 健 高木 浩則
日本電信電話株式会社 ソフトウェア研究所

1. はじめに

オブジェクト指向フレームワーク（以下、フレームワーク）を利用したビジネスアプリケーション開発手法が注目されている。フレームワークは協調稼働するように設計されたクラスの集合であり、デザインパターンによって拡張が容易になるような考慮がなされている。このため、フレームワークは従来のクラスライブラリと比較すると、より高度な再利用性と拡張性を提供することができる反面、その構造や振る舞いが複雑になるため、効果的に利用するには詳細な理解や知識を必要とする。

本研究では、フレームワークの利用プロセスを促進することを目的に、フレームワークの構築プロセスで得られる設計情報を統一的に操作・管理するための枠組みを提案し、これをベースにフレームワークの利用をガイドするためのナビゲーションツールの実現を目指している。今回、フレームワークのホットスポットの探索を目的としたナビゲーションツールを試作した。

2. フレームワーク利用プロセスの課題

フレームワークには機能的に標準化された部分（フローズンスポット）と柔軟性を持った部分（ホットスポット）が設定されている。フレームワークの利用プロセスにおいては、目的とするアプリケーションの機能要件に基づいてフレームワークのホットスポットを拡張することによって独自のクラスを実装する。上記を実施する上で、次のような課題を解決する必要がある。

- どのようにしてアプリケーションの機能要件を満足しそうなホットスポットを見つけ出すか。
- どのようにしてホットスポットをアプリケーションの機能要件に合わせて拡張するか。

上記の課題を解決するには、本質的に (1) フレームワークがサポートする業務や機能の範囲、および (2) フレームワークのアーキテクチャと含まれる各々のクラス、といったフレームワークの設計に関して異なる視点や詳細度での理解が不可欠であり、かつこれらの設計情報に効率よくアクセスできることが望ましい。

3. フレームワーク設計のメタ情報

2章で述べた課題を解決するための方法として、フレームワークの構築プロセスで得られる設計情報を活用することが有効である。しかし、その前提として (1) および (2) のようなフレームワークの設計情報を統一的に操作・管理するための枠組みを提供する必要がある。そこで本研究では、XML (eXtensible Markup Language) をベースにフレームワークの設計に対するメタ情報を記述するための言語を設計し、これを BDL (Business object framework Description Language) と呼ぶことにした。

3.1 基本的な枠組み

BDL で記述されるメタ情報は、フレームワークの設計情報とそれらの意味構造を反映している必要がある。BDL の基本的な枠組みを構成する記述要素について以下で説明する。

- 業務処理の内容 (Process, Task) — プロセスを構成するタスクは、その実行シナリオと実装クラスへの参照を持つ。
- 業務処理の実行順序 (Scenario, Action) — シナリオは順序づけされたアクションによって構成される。
- サブシステム (System, Subsystem) — お互いに関連の強いクラス群によって構成されるサブシステム。静的構造と動的挙動への参照を持つ。
- サブシステムの静的構造 (Structure) — サブシステムにおけるクラスモデル。設計パターンへの参照を持つ。
- サブシステムの動的挙動 (Behavior, Event) — サブシステムにおけるオブジェクトインタラクションモデル。順序づけされたイベントによって構成される。
- デザインパターン (Pattern) — フレームワークの設計に一貫して適用されたパターン。ホットスポットを適合させ

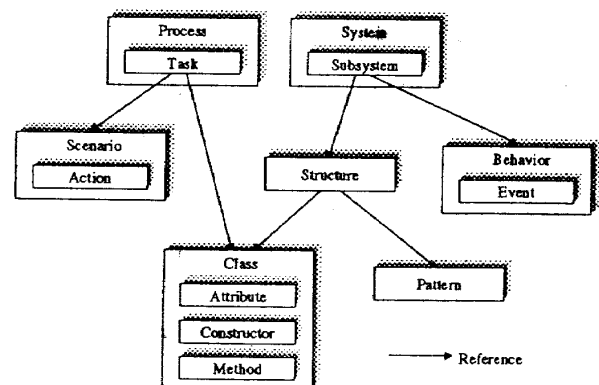


図1 BDLの基本的な枠組み

るための指針として必須である。

- クラス仕様 (Class, Attribute, Constructor, Method) サブシステムを構成する個々のクラスの仕様。メンバである属性、コンストラクタ、メソッドによって構成される。

3.2 メタ情報の記述

BDL では、基本的に各要素を階層的に構成することによってメタ情報を記述する。<Process>や<Task>のような BDL の基本的枠組みを構成する要素に関しては、すべて属性として id と href を持つ。属性 id では個々の要素の識別子を付与する。また、属性 href では詳細な設計情報を記述した HTML 文書の URL を指定する。さらに、<title>や<abstract>といった個々の設計情報の題目や要約を記述する要素を含む。これらは自然言語によって内容を記述する。ある要素から他の要素への参照を定義するには<reference>を利用する。具体的には、<reference>の属性 unit において参照先となる要素の識別子を指定し、<usedTask>や<respClass>のような参照の意味を表す要素の中にも含めることによって定義する。図 2 に XML をベースに記述した BDL の例を示す。

```
<Process id="P001" href="http://kiwi/p001.html">
  <title>Order Management</title>
  <abstract>This process covers...</abstract>
  <Task id="T001" href="http://kiwi/t001.html">
    <title>Create Order</title>
    <abstract>This task defines...</abstract>
    <usedTask>
      <reference unit="T021">...</reference>
    </usedTask>
    <respClass>
      <reference unit="C023">...</reference>
    </respClass>
    <descScenario>
      <reference unit="S012">...</reference>
    </descScenario>
    <isHotspot>Y</isHotspot>
  </Task>
  ...
</Process>
```

図 2 BDL の記述例

4. ナビゲーションツールの概要

4.1 プロトタイプ構成

プロトタイプは WWW (World Wide Web) を利用して実装した。WWW サーバには、設計情報を記述した HTML 文書、メタ情報を記述した BDL データ、および BDL 処理アプレットを格納する。WWW ブラウザは、ナビゲーション情報を表示するナビゲーションブラウザと詳細な設計情報を表示する設計情報ブラウザの 2 つで構成される。

最初にナビゲーションブラウザを起動すると、WWW サーバからアプレットがダウンロードされ、アプレットによって BDL データの読み込みと解析が行われる。ナビゲーションブ

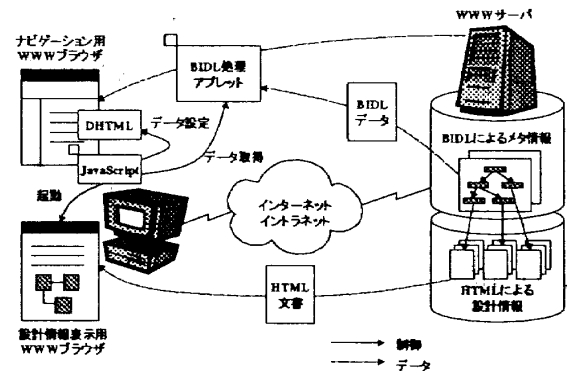


図 3 ナビゲーションツールの概要

ラウザでは、指定された識別子に対応する要素のデータをアプレットから取得し、HTML を動的に生成することによって表示を行う。また、URL が指定された場合は HTML 文書をロードして設計情報ブラウザに表示する。これらの処理は JavaScript によって実現されている。

4.2 機能とその特徴

試作したナビゲーションツールは、フレームワークがサポートするビジネスプロセスやタスクといった業務処理を参照できる経路とフレームワークを構成するサブシステムのようなアーキテクチャを参照できる経路の 2 つを提供する。ユーザはこれらの 2 つのルートを利用することによって、段階的にホットスポットを絞り込んでいくことができる。機能的な特徴としては以下の項目があげられる。

- フレームワークのホットスポットを見つけ出すまでの複数の到達可能な経路を提供する。
- ホットスポットを適合させるために必要な関連情報へ効率よくアクセスすることができる。
- フレームワークの異なる視点・詳細度における理解を促進する。

5. おわりに

本研究では、フレームワーク設計のメタ情報を記述するための言語を設計し、それを基にフレームワークのホットスポットを探索することを目的としたナビゲーションツールを試作した。

今後は、フレームワークのカスタマイズ方法をナビゲーションする機能を含めたフレームワークの利用プロセスを包括的に支援する機能の実現を検討していく予定である。

参考文献

- [1] 村田 真：XML 入門—HTML の限界を打ち破るインターネットの新技术—、日本経済新聞社 (1998)。
- [2] Pree, W.: Design Patterns for Object-Oriented Software Development, Addison Wesley (1994)。