

適応型システムのソフトウェアアーキテクチャと支援サービス

5 Q-5

野中 誠 木村 泰己 長崎 等 東 基衛

早稲田大学大学院理工学研究科

1 はじめに

筆者らは、システムが利用者の要求や周囲の環境に適応できる仕組みを備えた、分散・適応型情報システム(Distributed and Adaptive Information Systems: DAISY)の研究を行っている[1]。本研究ではその一環として、オブジェクト指向技術による適応型システムの構築技術に関する研究を行っている。

運用システムの適応には、業務適応と利用者適応との2種類がある。前者は組織及び業務の要求に基づいて運用システムが変更されるプロセスである。後者は一つの運用システムを利用する複数の利用者それぞれに対して、運用システムの振る舞いが変更されるプロセスである。本論文では、これらの適応を一部実現するソフトウェアアーキテクチャおよび適応支援サービスを提案する。また適応型システムの例を一部実装し、これを示す。

2 適応型システムのソフトウェアアーキテクチャ

以下に、業務適応の実現に焦点を当てた適応型システムのソフトウェアアーキテクチャを示す。これは以下の要求項目を実現するものである。

- 運用システムへの新規タスクの動的な導入
- タスク実施プロセスの変更

2.1 概要

適応型システムのソフトウェアアーキテクチャとは、適応型システムを構築する際に基本となるソフトウェア構造である。図1にそのオブジェクトモデルをUML(Unified Modeling Language)を用いて示す。これは適応型システムを構成するオブジェクト及びその相互の関係を抽象的に表したものである。

実際の適応は、後述の適応支援サービスと連携することによって行われる。

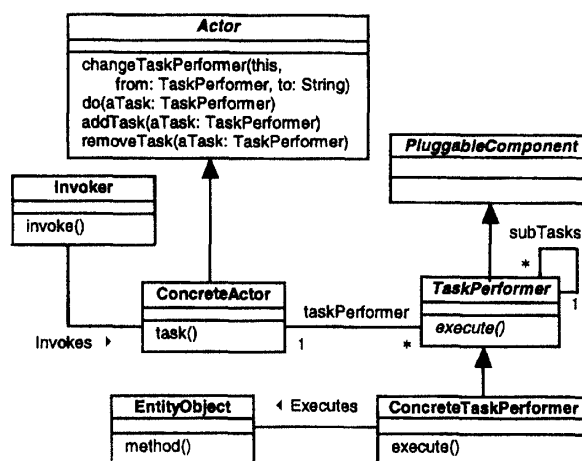


図1 適応型システムのソフトウェアアーキテクチャ

2.2 構成要素

以下に、ソフトウェアアーキテクチャの主な構成要素について説明する。

Actor: タスク実施プロセスの変更及び新規タスクの追加を行うメソッドを提供する抽象クラスである。メソッド `changeTaskPerformer` によりタスク実施プロセスが変更される。またメソッド `do` により、新規に追加されたタスクが実施される。

ConcreteActor: 利用者が実施できるタスクをメソッドとして提供するクラスである。オブジェクト `ConcreteActor` の生成時に特定のオブジェクト `TaskPerformer` が割り当てられる。メソッド `task` が割り当てられたオブジェクト `TaskPlayer` にメッセージ `execute` を送ることで、そのタスクが実施される。

TaskPerformer: オブジェクト `ConcreteActor` のメソッド `task` に割り当て可能なオブジェクトを型チェックにより制限するクラスである。また、自己再帰的な関係によりサブタスクを含むことができる。

ConcreteTaskPerformer: クラス `ConcreteActor` のタスク実施プロセスを一つのオブジェクトとしてカプセル化したクラスである。タスク実施プロセスは `ConcreteTaskPerformer` のメソッド `execute` 内に記述される。

PluggableComponent: 動的に交換可能なオブジェクトのルートクラスである。

3 適応支援サービス

適応支援サービスは、ソフトウェアアーキテクチャに基づいた運用システムの適応を支援する。

3.1 サービスの種類

Attachment Making Service

オブジェクト ConcreteActor のメソッド task によって実行される特定のオブジェクト ConcreteTask Performer から別のオブジェクトへと、クラス名を引数にして動的に切り替える。

Instance Creating Service

運用システムの運用後に新規タスクを導入する際には、クラス ConcreteTask Performer のオブジェクトを生成しなければならない。これを行うのがこのサービスであり、クラス名を引数にオブジェクトを生成する。

3.2 適応時のインタラクション

新規タスク実現要求に対する適応

- (1) Instance Creating Service によって、新規タスクを表すクラス ConcreteTask Performer のオブジェクトが開発者により生成される。
- (2) オブジェクト ConcreteActor のスーパークラスのメソッド addTask が利用者の操作によって呼ばれ、新規タスクのオブジェクト ConcreteTask Performer への参照が task Performer の集合に追加される。
- (3) オブジェクト Invoker によりオブジェクト Actor のメソッド do が呼ばれると、追加されたオブジェクト ConcreteTask Performer によりタスクが実施される。

タスク実施プロセスの変更要求に対する適応

- (1) オブジェクト ConcreteActor がスーパークラスのメソッド changeTask Performer によって交換元と交換先のオブジェクト ConcreteTask Performer を指定し、Attachment Making Service に交換を要求する。
- (2) Attachment Making Service によってオブジェクト ConcreteTask Performer への参照が別のオブジェクトへの参照に交換される。
- (3) オブジェクト Invoker によりオブジェクト Actor のメソッド task が呼ばれると、交換されたオブジェクト ConcreteTask Performer により新たなタスク実施プロセスが実行される。

4 適用例

ソフトウェアアーキテクチャに基づいた運用システムと、適応支援サービスに基づいた適応サーバの例を java により実装した。図 2 は、複数利用者の所在及び状態を表示するという運用システムの画面例である。このシステムにおいて、例えば所在・状態指定プロセスの変更といった要求が生じうる。

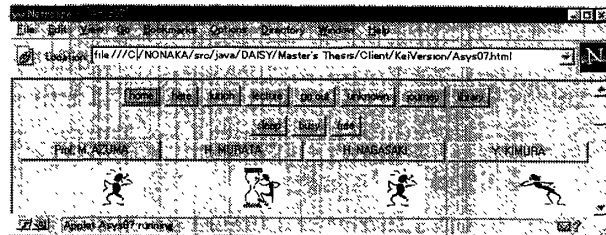


図2 運用システムの画面例

この運用システムのオブジェクトモデルを図 3 に示す。クラス DefaultStatusChanger のオブジェクトからクラス FrameStatusChanger へと task Performer の参照を動的に切り替えることで、上記の要求のようなタスク実施プロセスの変更が行える。

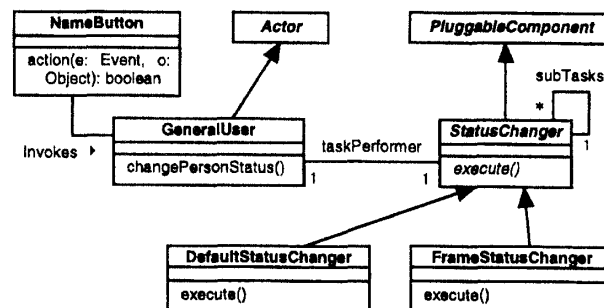


図3 図2のシステムのオブジェクトモデル（一部）

5 考察

本論文で示したソフトウェアアーキテクチャ及び適応支援サービスは適用領域に依存しないため、様々な適用領域で共通利用できる。また適応を実現する機構と、適用領域を表すオブジェクトが分離しているため、運用システムのモデルが複雑にならずに済む。ただしイベント駆動型ユーザインタフェースの性質上、Actor がタスク完了までのイベント制御を行うことは困難であるため洗練が必要である。

参考文献

- [1] 東基衛他, 分散・適応型システム実現のフレームワークと目標, 情報処理学会第 53 回全国大会, 1996.