

オブサーバブル・モデルを実装したインタプリタに関する研究

土山 由浩

1. 背景と目的

オブジェクトの凝縮化を行う場合、オブジェクト間の矛盾性を無くすためにデザインパターンのObserver モデル[1]が使われている。

Observer モデルはobserver をsubject に登録し、subject が変更されるたびに observer に通知する機構である。例えば、図1のように一つのデータに対して複数の画面表示を行うときや、MVC モデルのModel と View 間で表示内容の更新を行うときなどに凝縮に使われている。

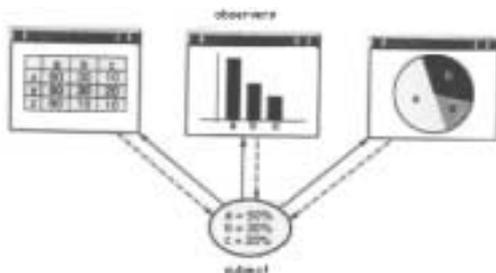


図1 Observer モデル

しかし、subject のデータが変更されるたびに subject - observer 間で表示内容の問合せ(ポーリング)を行わずにはならず、コミュニケーション量が增大するといった問題が生じる。

これらの問題を解決するために、observer 側の状態を監視するメソッドを返信し、subject 側に登録するための機構が考案されている[2]。しかしながら、具体的に実装されたものは現在見当たらない。本研究で試作システムを作成し、実装上の問題点を明らかにすることを目的とする。

2. 研究内容

本研究では、上記の問題を解決するためにメソッドを返信・登録するための機構を持ったインタプリタを開発した(図2参照)。

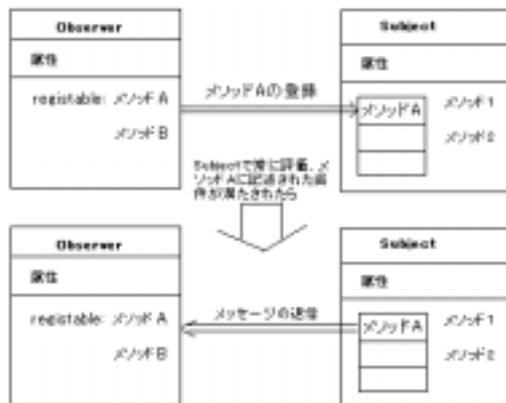


図2 試作システムの機構

メソッドにregistable タグを付加した場合、コンパイル時に宣言されていない属性が現れてもコンパイルエラーを起こさない。そして、実行時にメソッドの登録を行うと、宣言されていない属性と登録先の属性とでマッチングを行い、不一致の場合、実行エラーとなる。登録されたメソッドは登録先で evaluate ステートメントを実行するたびに実行し、任意にメッセージの返信を行うことができる。

3. 試作システムの機能

試作システムでは、オブジェクト指向インタプリタに次の機能を追加した。

- (1) メソッドを送るための機能
- (2) メソッドを登録するための機能
- (3) 実行時に登録可能か、変数のマッチングをする機能
- (4) 登録してあるメソッドを評価する機能

インタプリタはスタンドアローン環境としたが、将来的にコンピュータ間での凝縮化も視野に入れた機能となっている。

4. 検討

インタプリタの試作によって、次の様な問題点が明らかになった。

- (1) 登録先のメソッド中で使用する属性へのアクセス頻繁に属性にアクセスするメソッドを登録すると、オブジェクト間のトラフィックが逆増大してしまう可能性がある。登録するメソッドは、可能な限り登録先の属性にはアクセスしないほうがよい。
- (2) 登録元に対するセキュリティ保護

登録したメソッドは、登録先のすべての属性にアクセス可能である。このままでは登録先のデータを自由に更新したり、登録元に本来公開したくないデータまで見えてしまう。読み込み専用、読み書き可、読み書き不可などのアクセス制御が必要である。

5. 今後の課題

本研究でオブサーバブル・モデルを実装したインタプリタを開発し問題点を明らかにした。しかし、コンピュータ間でのオブサーバブル・モデルを考えれば、バイトコードを使用していないとメソッドの返信ができないという問題が残されている。

今回の試作システムで明らかになった問題点を考慮し、試作システムに改良を加える必要がある。

参考文献

- [1] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: "デザインパターン、ソフトバンク、1999年"
- [2] 大木幹雄: "凝縮化機構の実態認識と分析プロセスへの適用" 情報処理学会 ソフトウェア工学研究会 オブジェクト指向シンポジウム99 99巻 9号 pp.127-136,1999年
- [3] HE.エリクソン, M.ベンガー 著 杉本直男, 落合修 武田多美子 監訳: "UML ガイドブック、トッパン、1998年"