

徳田英修

品田京平

三浦孝夫

法政大学工学部電気電子工学科

概要

デザインパターンを組み合わせ設計したソフトウェアのバージョンをデータベース上で体系的に管理し、またその管理したデザインパターンの利用を支援する環境を考える。デザインパターンが進化した時は差分をとり最新ファイルを管理する。しかしデザインパターンのバージョンの変化は、旧バージョンの単純な進化とは限らないので、その場合は差分はとらず、データベースに全てのファイルを保存する。新たなバージョンを登録した時、その元となったデザインパターンが更新されると新たなデザインパターン文書内にもそれに関するファイルが含まれているので時間的矛盾が発生する。本システムではこの差分と依存関係の差を見つけ出し、それを警告する環境を提供し、Makeのように論理的依存関係を時間順序で表現する。保存するデザインパターン文書はTEXなどマクロ機能のある書式で記述し、マクロをパラメタとした検索を実現する。

1 目的

デザインパターンを使用、登録するさい、それを支援するようなシステムを考える。デザインパターンは新たな経験を得てゆくことにより変化していくが、それら複雑に枝分かれしていくデザインパターンのバージョンの管理し、データベース上に置く事による検索システムを使った作業効率の向上、新たなデザインパターンバージョンの登録、管理の支援を目指す。デザインパターンが新たなアイデアによって進化を遂げた場合は最新ファイルと過去の状態に戻すパッチファイルによって管理する。しかし変化していくデザインパターンは単純な進化とは限らないので、その場合は差分をとらず、別のデザインパターンとして全てのファイルをデータベースに保存する。変化を遂げた各デザインパターンには、元バージョンに関連しているファイルが存在するため、そのファイルが更新された時は時間的矛盾が発生する事になる。本システムではMakeのようにデザインパターンバージョンの論理的依存関係を時間順序で表現することを目指す。データベースに保存したデザインパターンは、比較検討ができるようにデザインパターン文書をテンプレート化し、マクロ機能のある書式で書く事によりマクロをパラメタとした検索を実現する。本システムではデザインパターン文書のバージョン・サンプルコードの管理と、管理したデザインパターン利用の際の検索、保守・同期を支援する。この2点について述べる。

2 管理方法と妥当性

デザインパターンは目的や動機などを記録整理した文書と、パターンの実装方法を示したサンプルコードを一対として扱

Engineering Program Design in a MAKE style
Hidenobu Tokuda, Kyohei Shinada, Takao Miura
Hosei University, Dept. of Elec. and Elec. Eng.
Kajino-cho 3-7-2, Koganei, Tokyo, JAPAN

い、PostgreSQL を利用したデータベース上に実ファイルをそのまま入れる。ここで言うデザインパターン文書は内容をテンプレート化して統一し、TEXやXMLなどのようにマクロ機能のある書式で書いたものである。

2.1 デザインパターンのバージョン管理

デザインパターンとは、ソフトウェアを設計する際の設計経験を他の人が再利用できるように整理して記録したものである。デザインパターンを利用すれば、すでに成功している設計などの再利用が可能となる。これらのデザインパターンがアイデアや経験によって新たなデザインパターンとして変化していくこともあるだろう。本研究では変化していくデザインパターンバージョンとそれに付随するサンプルコードの体系的な管理を目指す。

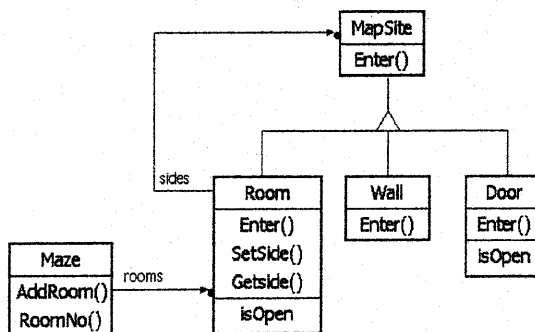


図 1: 迷路ゲーム (Gamma/オブジェクト指向における再利用のためのデザインパターン)

例えば図1の例はGammaらの「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」で紹介されていた、迷路ゲームのクラス間の関係を示したダイアグラムである。この例では単純に迷路に要素を追加するオペレーションを次々に呼び出し、その関係を与える方法をとっている。しかし柔軟性に向け、例えば迷路の構成を変更するためにはメンバ関数の再定義が必要であったりする。Gammaらは、この迷路の構成を再利用し、新しいクラスのオブジェクトを加えて新たな要素を盛り込んだ迷路ゲームを作る方法を様々なデザインパターンを適用して示している。例えばAbstract Factoryパターンではメンバ関数に部屋、壁、ドアを作成するオブジェクトをパラメタとして渡し、そのパラメタを変えることで各クラスを変更できるようにしている。Builderパターンでは部屋、ドア、壁を追加していくオペレーションを使い、新たに迷路を作成するオブジェクトに渡すようにしている。これにより継承を用いて迷路の一部を変更したりする事が可能となっている。このようにある問題を解決するのに、適用されるデザインパターンが一つとは限らない。この例では、いずれのパターンも迷路ゲームをつくるために利用したが、

Abstract Factory パターンと Builder パターンのどちらで作成したものが優れているとは一概にいけない。このため本システムでは Abstract Factory パターンで作成した迷路ゲームが Builder パターンより後に作成されたとしても、新しいバージョンとして登録するのではなく、違うものとして登録する。しかしこの例で元の迷路ゲームに「罠」という新たな要素を入れたとすると、それはあきらかにデザインパターンの進化であり、バージョンアップである。この場合は、最新ファイルと差分ファイルを管理していく事になる。当然この例のような場合、元の迷路ゲームから Builder パターンを用いて作成された迷路ゲームでも「罠」の要素を適用する可能性はあるわけである。新たなバージョンとして更新された時、本システムはこれに関連するデザインパターンに警告を与える。例えば、Builder パターンで作った迷路ゲームには元の迷路ゲームに関連するファイルが組み込まれており、元の迷路ゲームが更新された時、その関連するファイルが時間的矛盾をおびる事になる。本システムでは Make のようにこの差分と依存関係の差を見つけ出し、それを警告する環境を提供する。

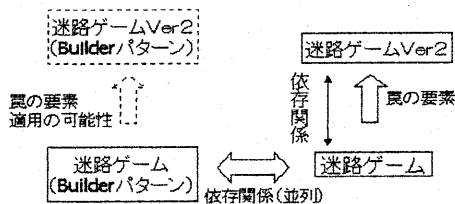


図 2: 論理的依存関係と時間順序

2.2 デザインパターン文書の管理と利用

デザインパターンのとらえ方や、整理、記録の仕方は人それぞれであり決まりがあるわけではない。そのため、いくつものデザインパターン候補があった場合それを比較し選択する基準がない。本研究では、数々のデザインパターンを比較検討できる環境をデザインパターン文書のテンプレート化とパラメタ化によって実現する。デザインパターン文書はテンプレート化し、同じ条件で比較できるようにする。全てのデザインパターンがテンプレート化されたことにより、使用目的や構成要素など各項目で比較ができる。

デザインパターンのパラメタ化は検索によって、各デザインパターンから必要な情報を取り出すためである。データベース上に登録する文書は、Tex や XML のようなマクロ作成機能を持つ書式で書き、あらかじめ設定しておいたマクロをデザインパターン文書の中から必要な情報を検索するために使用するパラメタとして利用する。

図3のようにデザインパターン文書のテンプレートの各タイトルにあたるものをパラメタ化することにより、検索で欲しい情報部分を抜き出す事ができる。本システムは検索する時、文書内を文字列検索をしてマクロ部分を探し出し、その部分を抜き出す。例えばパターン名から検索をかけるときは、デザインパターンテンプレート内のパターン名を表すマクロを探すのである。検索して抜き出したTEXファイルの一部分はMakeによって自動的にコンパイルされ、DVIファイルを生成する。ユーザーはこの方法で取り出したデザインパターン文書から必要な情報を選び比較することができる。例えば取り出したデザインパターンの使用目的部分を見比べ、どのパターンが現在必要かを判断する事ができるのである。なお、本稿ではTexにしぼって論じる。

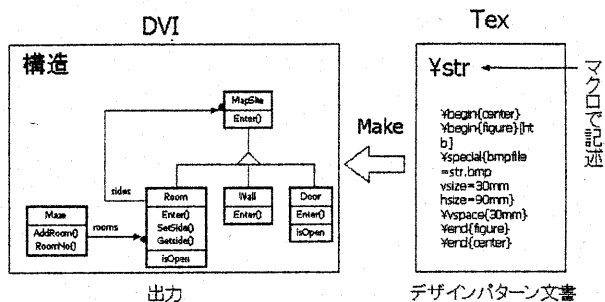


図 3: デザインパターン文書と検索

2.3 サンプルコードの管理と利用

データベース上にはデザインパターン文書の中のサンプルコードとは別に、プログラムの実ファイルを保存しておく。実際にサンプルコードを参照、利用する時はこの実ファイルを読み出して使うことになる。そのままコンパイルした場合、エラーが起きる可能性があり書き換えを必要とするところはコメントで注意を喚起するようにする。そのコメントには決められたキーワードを挿入する事に決め、本システムではそのキーワードを捨て、ユーザーに注意を促す事にする。ユーザーは注意を促すコメントを参考にその部分を書き直して、そのコードを利用する。エラーが起きる可能性のあるところは全てコメントが提示されているのでその部分にだけ注意を払えばいいことになる。実ファイルを利用しているため、直したプログラムはMakeを利用して同期させることができる。

3 システム化

データベース化にはデータベース用の標準言語であるSQLをサポートしているPostgreSQLを使用する。しかしデザインパターン文書がPostgreSQLの扱える8kbに納まるとは限らないので、データ格納形式をラージオブジェクトとする。ラージオブジェクト形式ではデータの大きさにほぼ制限がない。そのためデザインパターン文書内でも大きなデータとなる画像ファイルを扱う事が可能となる。

検索時の処理だが、本システムでは欲しい情報にあたるマクロを探し、その部分を抜き出すと用意されたTexの難型の\begin{document}と\end{document}の間に貼り付ける。こうしてできたTexファイルをMakeによって自動的にコンパイルし、ユーザーは欲しい情報をDVIoutで見ることが出来る。また取り出したサンプルコードを修正し作成したコードも同様にMakeを走らせて自動的に実行可能形式にする。Makeでコンパイルする時、他の依存関係にあるデザインパターンのファイルには時間的矛盾が発生する。この時に本システムでは依存関係にある各デザインパターンについて警告を与えることになる。

4 結び

データベース上でデザインパターンを扱い、変化していくデザインパターン文書の枝分かれと、依存関係にあるデザインパターンの管理を行なうシステムについて述べた。また、検索機能はデザインパターン文書にマクロ作成機能のある書式を使うことにより、マクロをパラメタとした検索とした。

参考文献

- [1] Erich Gamma, Richard Helm, Ralph Jonson, John Vlissides: 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 (1999)