



オブジェクト指向2003シンポジウム・デモンストレーション資料

# ソフトウェアパターン関連ツール・ デモンストレーション

久保 淳人 鷺崎 弘宜

早稲田大学理工学部

( IPSJ/SIGSE パターンワーキンググループ・研究タスク )

{ a.kubo, washi } @fuka.info.waseda.ac.jp

<http://patterns-wg.fuka.info.waseda.ac.jp/>

資料作成協力: 小林隆志さん(東京工業大学) 他  
パターンワーキンググループ研究タスクの皆様

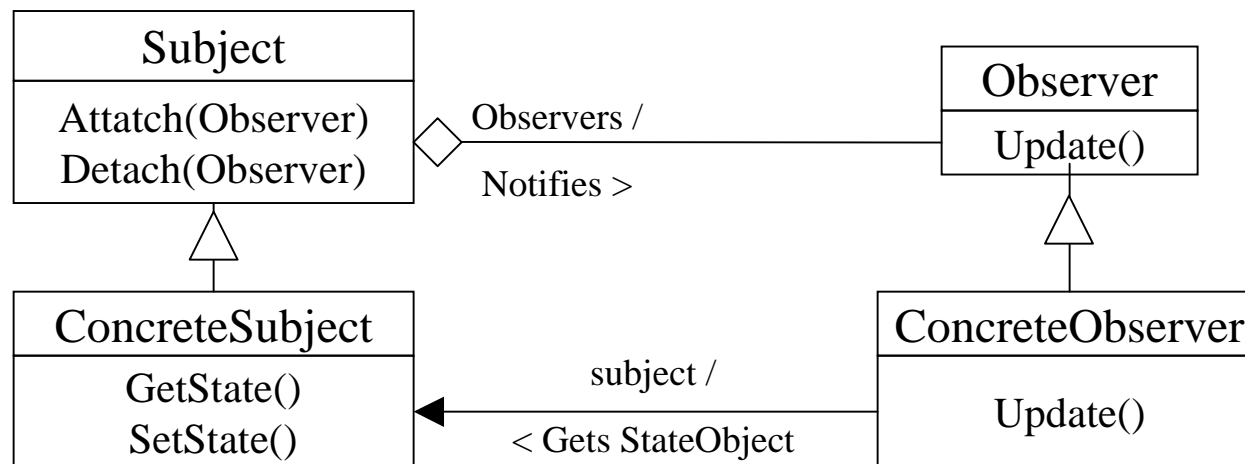
# 三 デモ・シナリオ

- 時刻を表示するプログラムの開発
  - Java言語
  - GUI
    - アナログ時計で表示
    - デジタル時計で表示
  - 将来、表示方法は変更されるかもしれない

時刻を管理する部分と表示を管理する部分を分離した、疎結合なシステムが良い

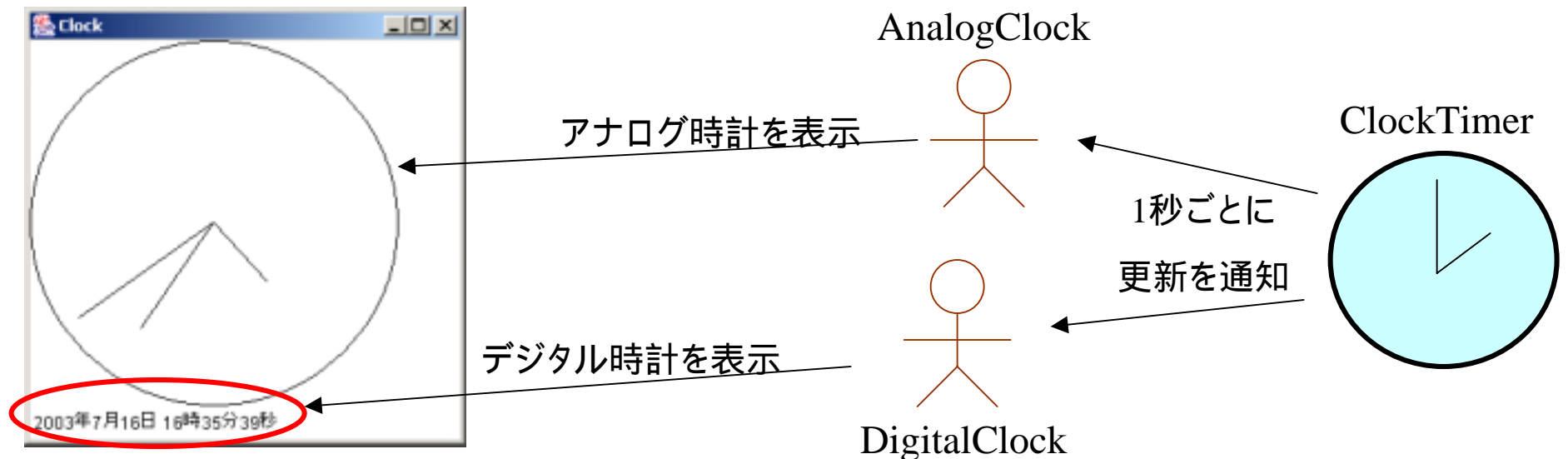
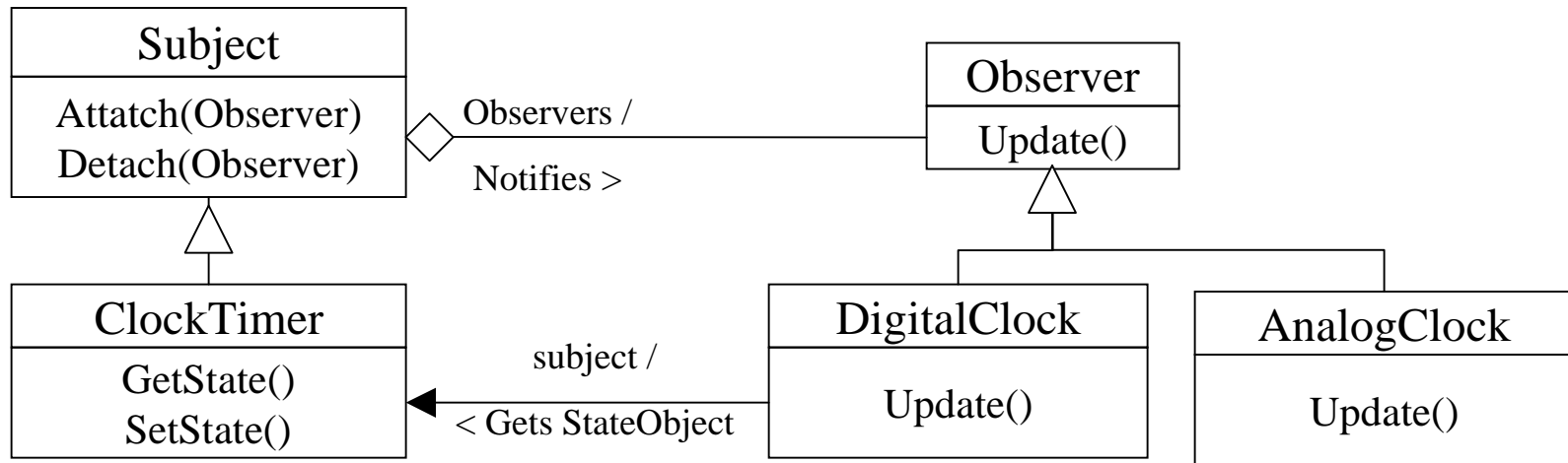
Observerパターンを適用

# Observerパターン



- あるオブジェクトが状態を変えたとき、自動的に他のオブジェクトに通知するように、オブジェクト間に依存関係を定義する
- 状態変化を通知するオブジェクトと、状態変化の通知を受けるオブジェクトを疎に結合したいとき有効

# 時刻表示プログラム

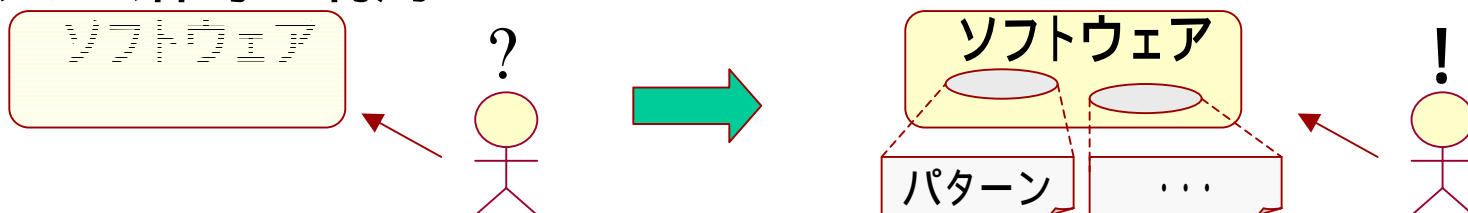


# 三 パターンを用いた開発の問題点

- 同じようなコードやモデルを何度も作成
  - 人手で書くのは非効率  
機械的に処理するツール
- パターン本来の意図と異なる実装をしてしまう可能性

# 有用なパターン関連機能

- デザインパターン適用機能
  - 既存のモデル・プログラムソースコードへデザインパターンを自動的に適用する
  - プログラム設計や開発で利用
  - 分類
    - (1) モデル・ソースコードの雛形を新規生成
    - (2) デザインパターンの雛型を、クラス・メソッドに対して役割を決めて新規生成
    - (3) 既存モデル・コード中の特定個所を、リファクタリング操作によりデザインパターンの雛型で置換
- デザインパターン検出機能
  - 既存のモデル・ソースコードについて、既知のデザインパターンが適用されているかどうかを検出し、理解を支援する
  - プログラム保守で利用



# 紹介するツール

ソフトウェアパターン(主にGoFデザインパターン)の検出や適用といったパターン関連機能を持つツールを幾つか紹介する

- (1) デザインパターンの雛形となるモデル・ソースコードを新規生成
  - UML Studio
  - ModelMaker
- (2) クラス・メソッドに対して役割を決めてデザインパターンの雛形を新規生成
  - Pattern Support for Eclipse (PSE)
  - CodePro Studio
- (3) 既存モデル・コード中の特定個所を、リファクタリング操作によりデザインパターンの雛型で置換
  - PTIDEJ



# (1) デザインパターンの雛形となるモデル・ソースコードを新規生成

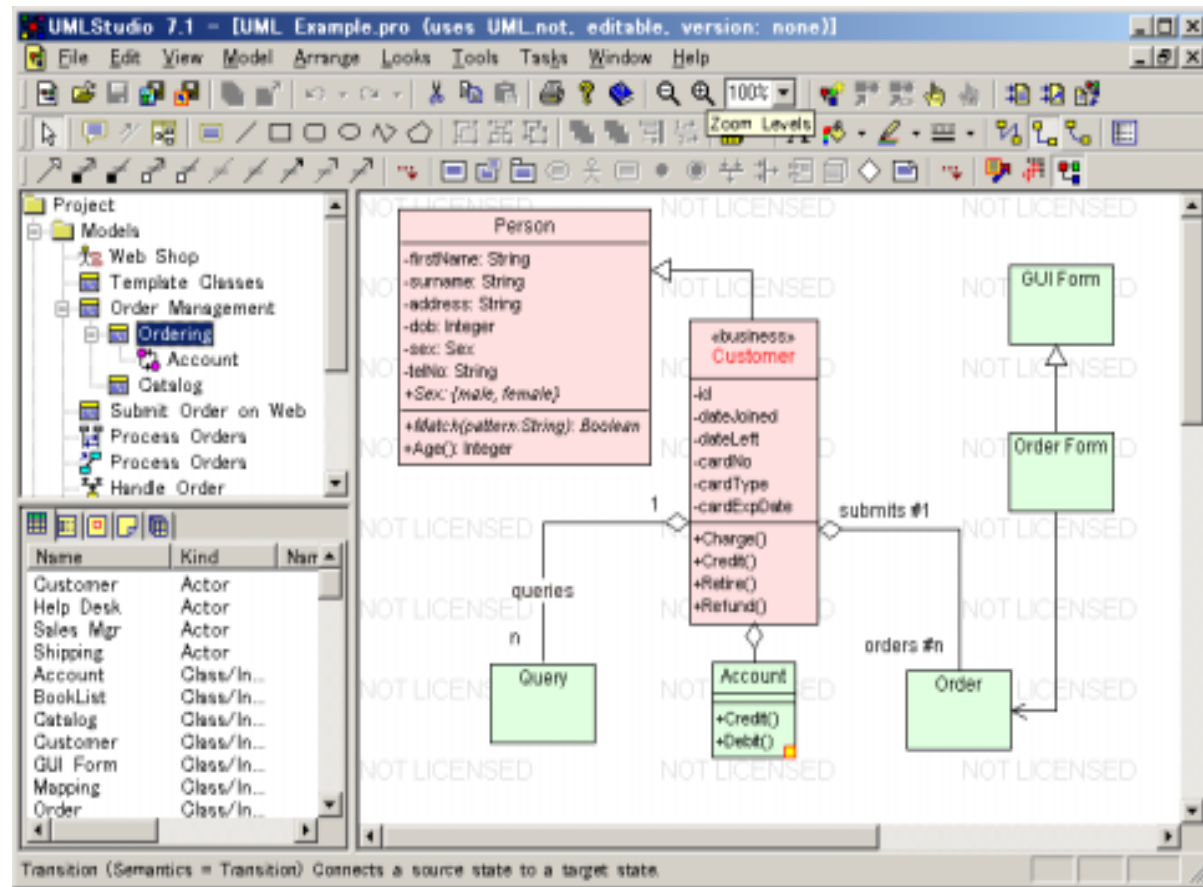


# UML Studio

Javaソースコードからモデルのリバーサ作成が可能な  
UMLモデリングツール

## 機能

- モデリング機能
- Javaソースコードを解析してモデルをリバーサ作成





# UML Studioのパターン関連機能

## ● パターン関連機能

- 繰り返し出現するモデル図の一部分をデザインパターンとして登録し再利用する
- 扱うパターン: GoFのデザインパターンの一部

## ● 特徴

- モデル要素の表記法を利用者が定義することができる  
各種の表記法に対して柔軟に対応

## ● 対応環境

- 32bit Microsoft Windows環境

## ● 入手方法

- 開発元: PragSoft, <http://www.pragsoft.com/>

## ● ライセンス

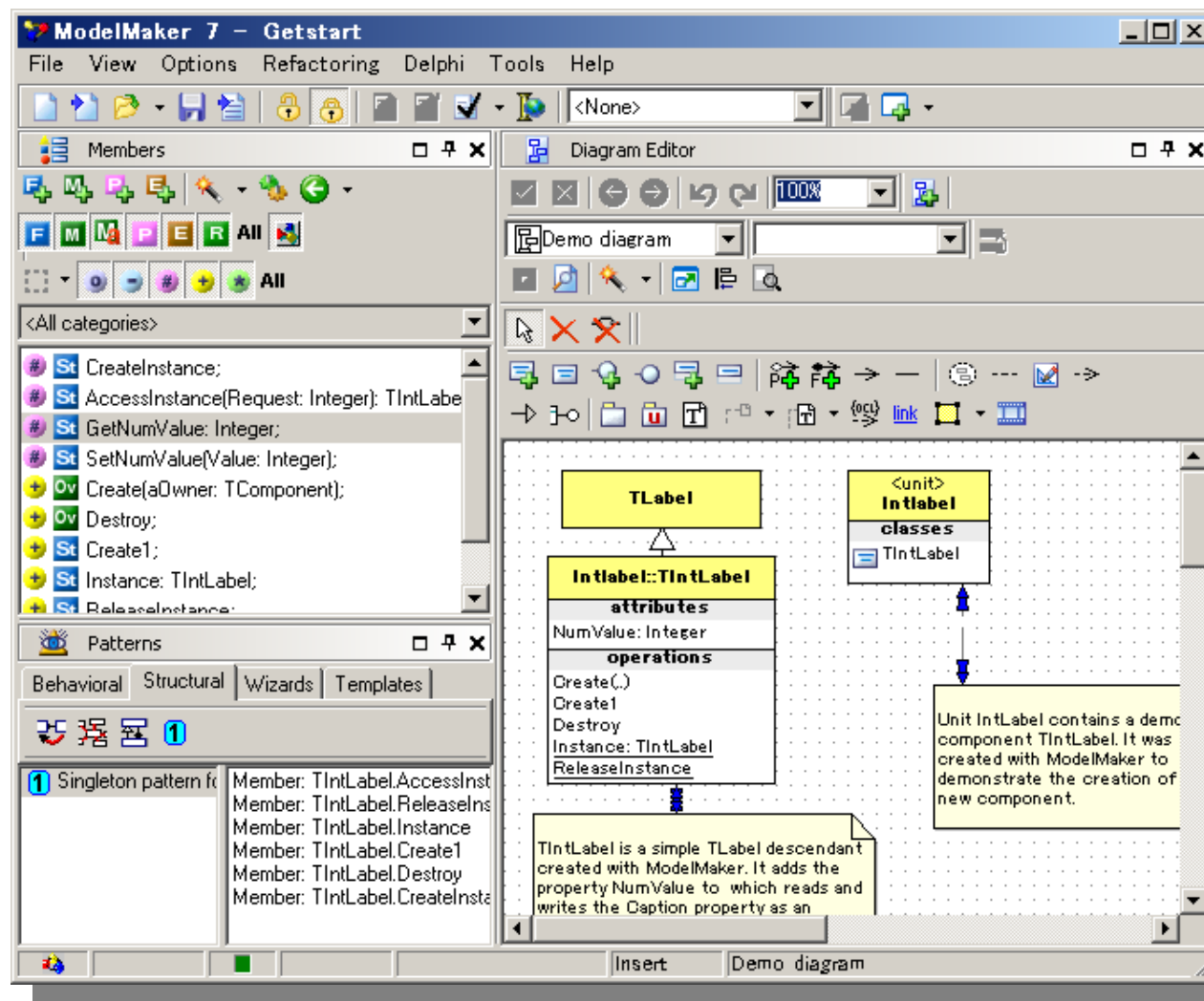
- 商用、体験版あり

# ModelMaker

## Borland Delphiに特化したUMLモデリングツール

### 機能

- リファクタリング支援
- 既存のDelphiソースコードからのUMLモデルのリバース作成





# ModelMakerのパターン関連機能

## ● パターン関連機能:

- デザインパターンをコードテンプレート的一种としてサポート
- ダイアログからパターンを選択して、雛型コードの作成
- 扱うパターン: GoFデザインパターンの一部

## ● 動作環境

- Microsoft Windows環境

- 生成したソースコードのコンパイルにはBorland Delphiが必要

## ● 入手方法

- 開発元: ModelMaker Tools BV,

<http://www.modelmakertools.com/mm.htm>

## ● ライセンス:

- 商用、体験版あり



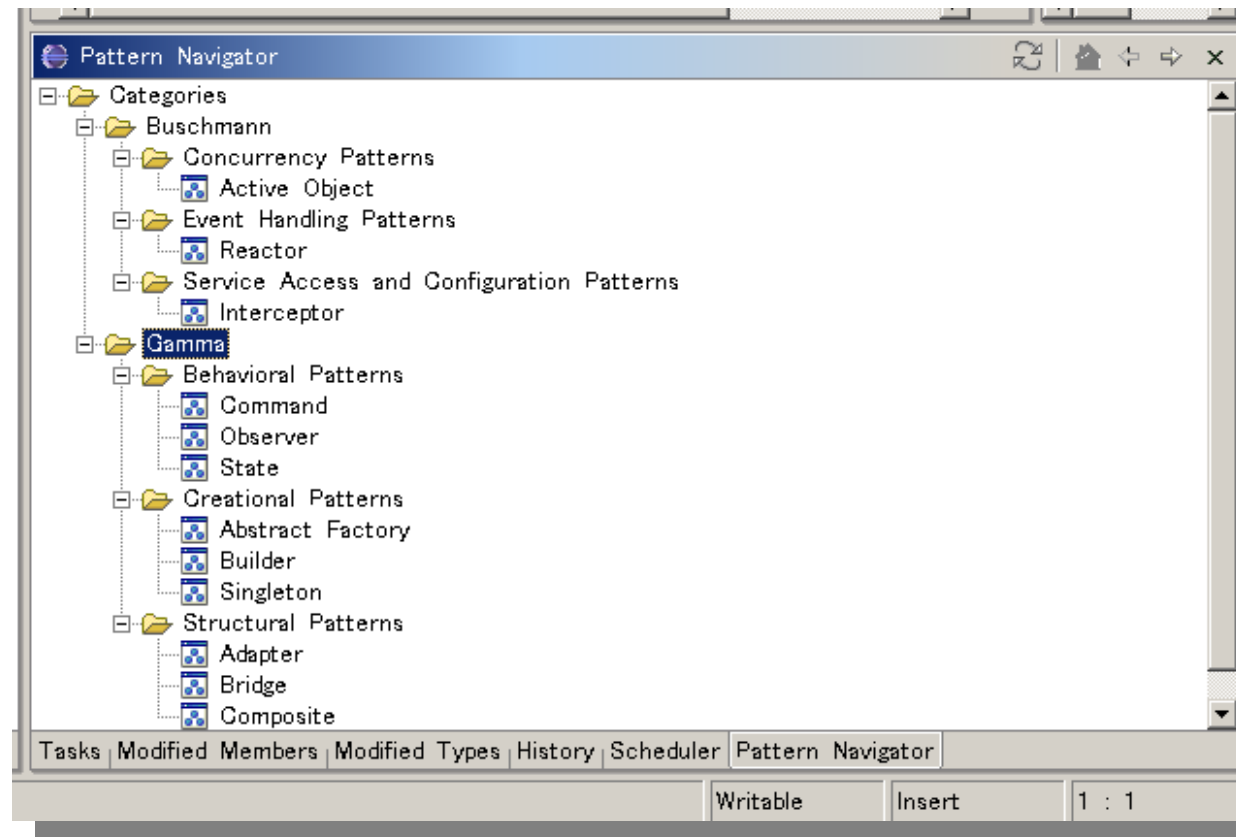
## (2) クラス・メソッドに対して役割を決めて デザインパターンの雛形を新規生成

# Pattern Support for Eclipse (PSE)

パターン関連機能を提供するEclipse Plug-in

## ● パターン関連機能:

- ウィザードからデザインパターンを選択・適用
- 既存のソースコードにデザインパターンを適用可能
- メソッドの実装も生成
- 扱うパターン: GoFのデザインパターン、Buschmannらによるデザインパターン



# PSEの入手方法

## ● 対応環境

- Eclipseが動作する環境

## ● 入手方法

- 開発者: Irmgard Mayr-Kern, Andreas Wöckl (Johannes Kepler Universität Linz),

<http://www.swe.uni-linz.ac.at/people/sametinge/research/pse.html>

## ● ライセンス

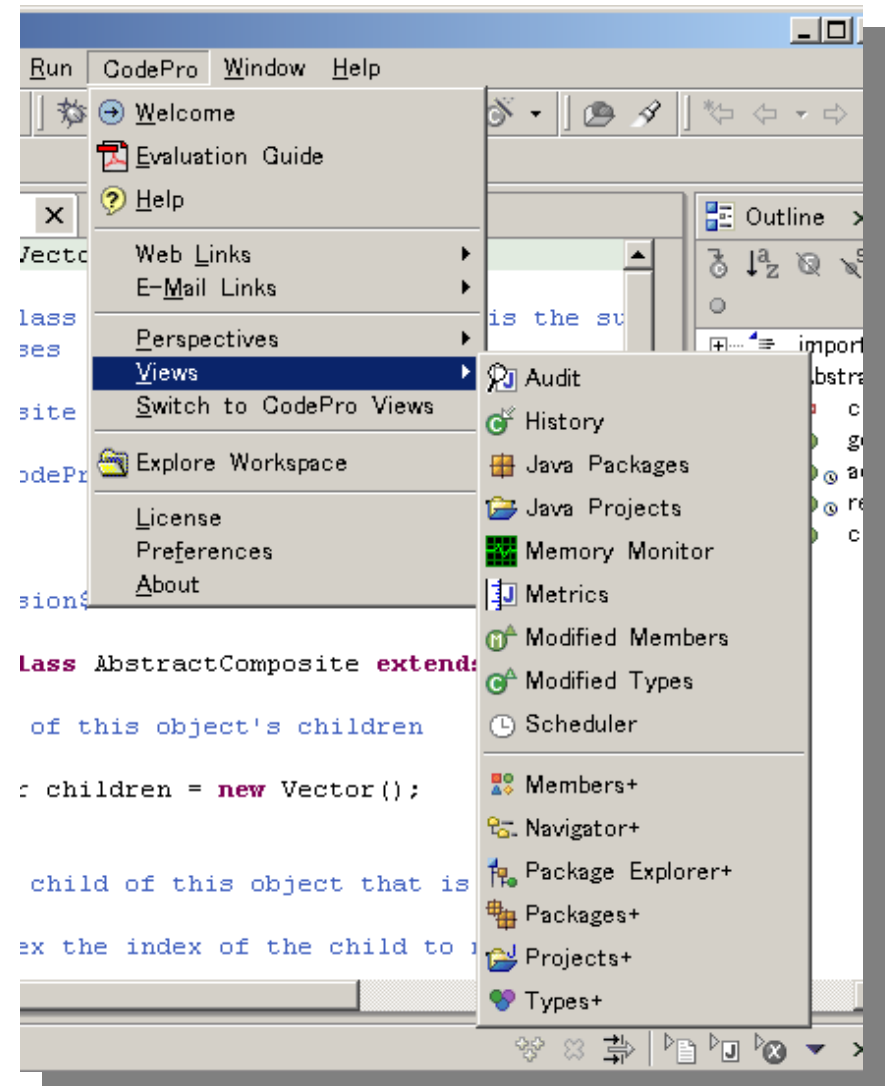
- フリー

# CodePro Studio

コード改善支援、ビルド管理等の機能を持つEclipse Plug-in

## 機能

- ソースコードの改善支援機能
- ビルド管理機能
- メトリクス測定機能
- Javadocの自動修復機能







# CodePro Studioのパターン関連機能

- パターン関連機能
  - ウィザードを用いたデザインパターンの適用
    - デザインパターンを構成するロール等のオプション指定が可能
  - 雛型コードの生成
  - 扱うパターン: GoFデザインパターン、Buschmanらのデザインパターン

## ● 動作環境

- Eclipseが動作する環境

## ● 入手方法

- 開発元: Instantiations Inc.,

<http://www.instantiations.com/copepro/>

## ● ライセンス

- 商用、体験版あり



**(3)リファクタリング操作でパターンを適用する**

**パターン検出機能**



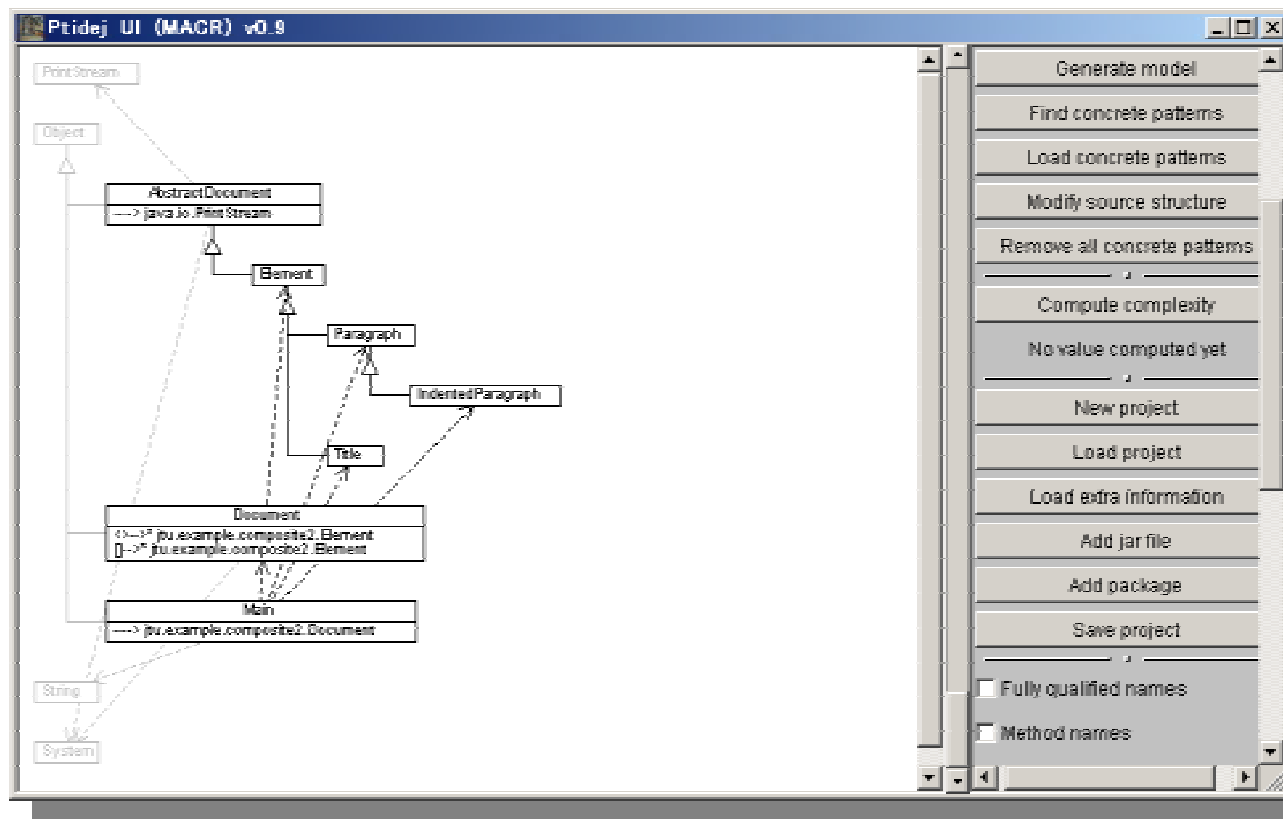
## パターン検出機能を持つソフトウェア変更支援ツール

### ● 機能

- Javaソースコード・クラスファイル解析機能
- 簡易メトリクス測定機能

### ● パターン関連機能

- デザインパターンの検出
- ソースコードの自動リファクタリング
- 扱うパターン: GoFデザインパターン





# PTIDEJのパターン関連機能

- メタモデル

- デザインパターンを抽象的なモデルとして定義したもの
- 文脈に依存しない

```
anInterface = new Interface("Observer");  
abMethod = new Method("Update");  
anInterface.addElement(abMethod);
```

```
ContainerAggregation anAssoc = new  
    ContainerAggregation("observers", anInterface, 2);
```

# PTIDEJの特徴

## ● 想定される利用状況

- 現在のクラス設計が妥当なものか調べたい

Javaクラスファイルやソースコードを解析してUMLクラス図を表示、存在するデザインパターンを検出

既知のデザインパターンに部分的に合致する場合は、完全に合致するようにソースコードのリファクタリングを行う

## ● 対応環境

- Java2 SDKが動作する環境

## ● 入手方法

- 開発者: Yann-Gael Gueheneuc (University of Montreal),

<http://www.yann-gael.gueheneuc.net/Work/Research/Pt ide j/>

## ● ライセンス

- フリー

# まとめ

- パターン関連ツールの機能
  - パターン適用機能
  - パターン検出機能
- 今回紹介したツールは全て、フリーまたは体験版をWebから取得可能



ありがとうございました

# ソフトウェアパターンとは

- 「ソフトウェア開発において繰り返し発生する問題に対する、再利用可能な解法」
  - 経験豊富なプログラマや設計者は、設計上の問題が過去に経験したものと似たものであれば、これを素早く解決することができる。
  - このノウハウを形式的に表現したもの
- ソフトウェアパターンを利用した開発では、既存の設計知識を再利用して開発することで、ソフトウェア設計の改善や迅速な設計・開発が期待できる。