

Java + Caché

オブジェクト指向開発の事例

～ O/Rマッピングを排除し開発効率と安定性を向上させる ～

第31回 日本Mテクノロジー学会大会
MTA2004 京都

2004年8月22日
ネクストデザイン株式会社
村山 徹

はじめに

- M言語ではなく、JavaとCachéの事例
- Java
 - ビジネスアプリケーション分野の新規開発で最も多く使用されている言語
 - オブジェクト指向設計モデルの実装に適した言語
- UML
 - オブジェクト指向分析・設計モデルをビジュアルに表現するための標準言語(記法)
- データベース
 - Java、UMLは広く普及したが、データベースは依然としてRDBが多い
 - **オブジェクト指向モデルやJavaとRDBは相性が悪い**
(インピーダンス・ミスマッチ)
 - Caché(オブジェクトアクセス)との相性は良い

ネクストデザイン(有)のご紹介

<http://www.nextdesign.co.jp/>

- 特徴: オブジェクト指向技術とJavaに特化
- 業務内容:
 - 技術者向けセミナー
 - コンサルティング
 - システム開発
 - UMLツール「Compass」の開発・販売
 - Caché販売パートナー
- 所在地: 福岡市(ソフトウェア・リサーチ・パーク)

事例の概要

- ユースケース・モデリング・ツール「Compass」の開発
- 永続化エンジンにCachéを採用
- 開発技法・ツール
 - オブジェクト指向技術
 - 実装言語: Java (1.4.2_01)
 - GUIコンポーネント: SWT、JFace
 - Caché Javaバインディング (5.0.8)

開発の動機

- ソフトウェア開発プロジェクトの成功率アップ
 - 例: 26.7% (日経コンピュータ 2003.11調査結果)
- ソフトウェア開発技術の動向
 - 主流はオブジェクト指向技術
 - 複雑化したユーザー要件に対応できる開発技術
 - UML、JAVA、C#の普及
 - オープンソースの拡大

… 従来型よりも成功率は向上したか？

成功率低迷の原因

- 外部要因
 - ソフトウェア要件は従来よりも複雑化
 - 要件の変動
 - ユーザー都合、ビジネス環境
 - 短期開発
 - ...
- オブジェクト指向に関係する要因
 - ツール、プラットフォームの変化
 - 熟練技術者の不足
 - 間違った技術の使用
 - ...

当社の対策と提案

- Caché オブジェクト・アクセス
 - O/Rマッピングをなくし
 - 開発工数半減
 - 複雑さを減らすことで品質と安定性を向上
- Compass
 - ユースケースを活用し
 - オブジェクト指向開発プロセスを正しく使用する

開発時の基本方針

- オブジェクト指向技術を正しく適用し、その効果を最大限にするために



- リレーショナルデータベースを使用しない
O/Rマッピングによる負荷をなくし、
短期間で製品化する

O/Rマッピングとは

- 「オブジェクト・モデル」と「リレーショナル・モデル」のインピーダンス・ミスマッチを解決する手段
- オブジェクトを(複数の)テーブルに関連付ける
- オブジェクトをテーブル群に分解して保存する
- テーブル群を読んでオブジェクトを組立て、復元する
- 主キーを生成するオブジェクトやテーブルにアクセスするためのオブジェクトを追加する

これらは煩雑で、本来不要！

O/Rマッピングが引き起こすもの

- 本来不要なオブジェクトを追加しなければならない
- モデルが複雑になり、分かり難くなる
- 多くのSQLコードを書かなければならない
(非オブジェクト指向コードを書く必要がある)
- その結果、コード量が増える(1.5~2.0倍)
- そこにバグが多発する
- テーブルスキーマ変更時の変更箇所が多い

オブジェクト指向開発で失敗につながる最大の原因

- 工数オーバー
- 品質不安定

RDBを使わない永続化方式

■ 選択肢

- (1) オブジェクト指向データベースを使う
- (2) O/Rマッピングツール + RDBを使う
(例) Hibernate + PostgreSQL
- (3) J2EE/CMP を使う
- (4) JDOを使う
- (5) Caché オブジェクトアクセスを使う

■ Cachéを採用した理由

- 一般的なJavaの開発スタイルが使えること
- 実績
- 価格

その効果

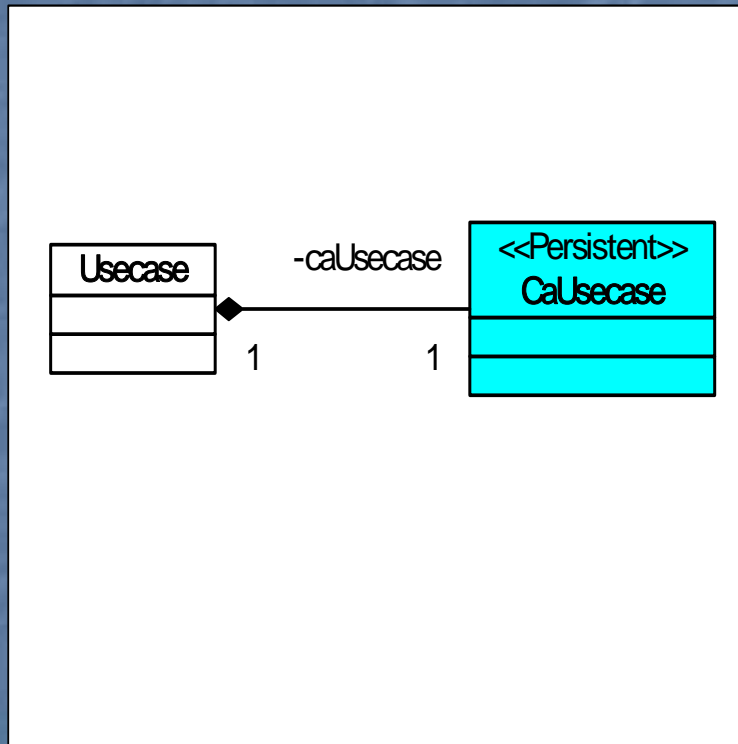
- モデルを自然な状態に保てた
 - 分かりやすいモデル(安定したモデル)
 - 変更、拡張に強いモデル
 - コード量が減少 (1/3 から 1/2)
 - バグが減少



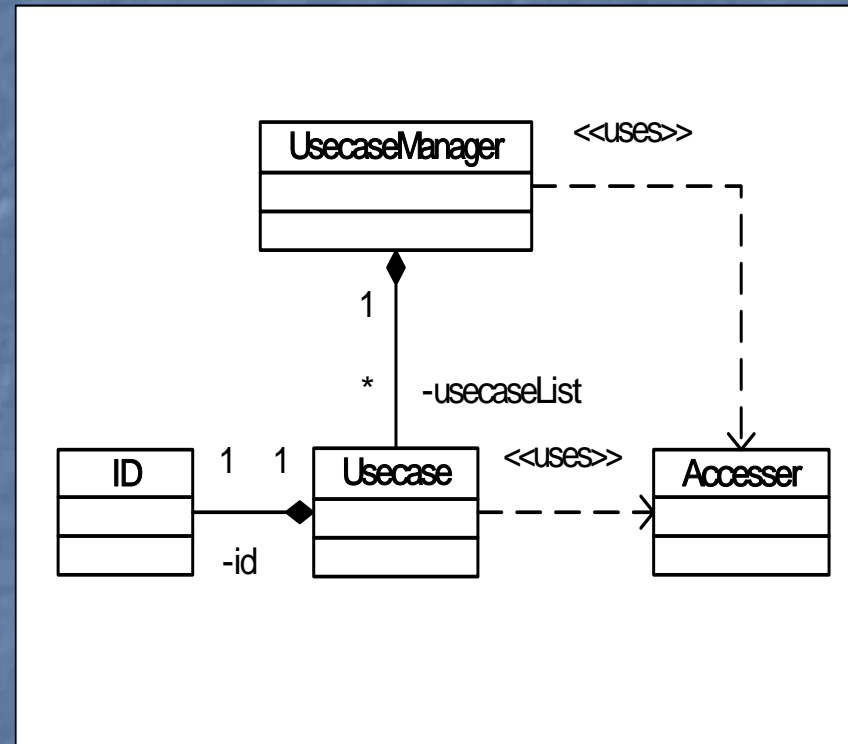
開発のスピードアップ
本来のドメインロジックに専念できた

設計モデルの比較

Cachéを使用する場合

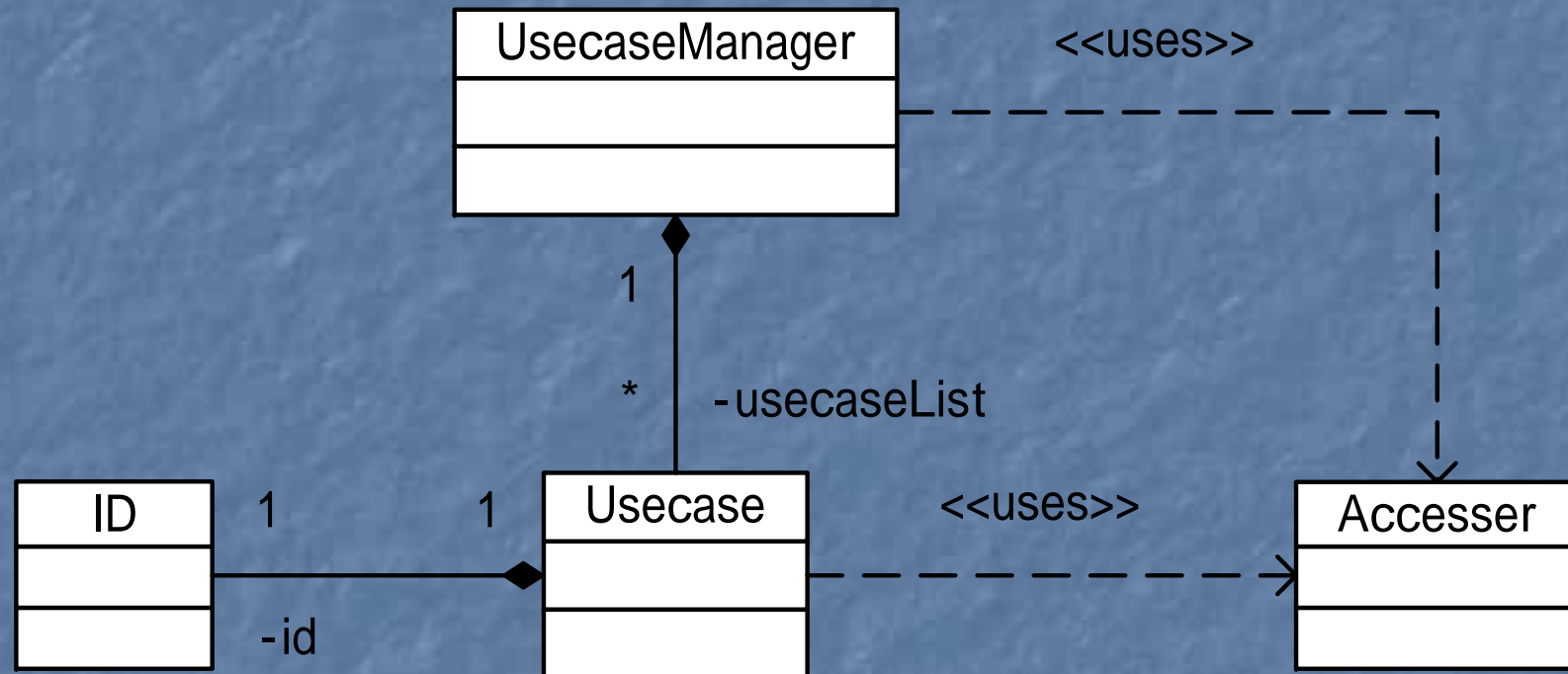


RDBを使用する場合



RDBを使用した設計モデル

主キーを生成する責務を持つオブジェクト
エクステントを管理するオブジェクト

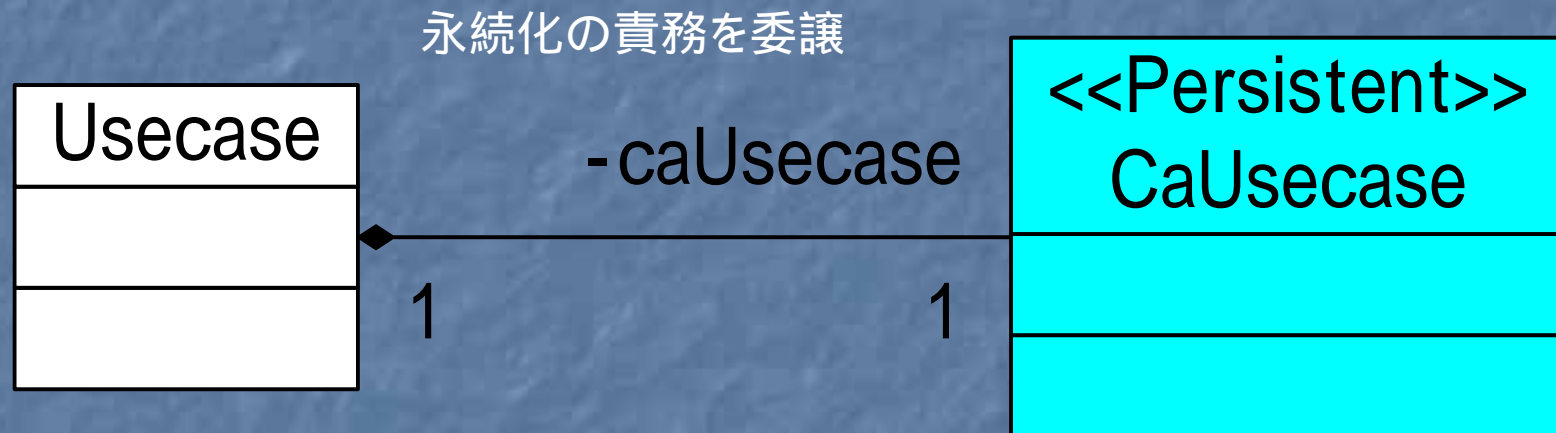


ドメインとしては
不要なオブジェクト

テーブルへのアクセスを
責務とするオブジェクト
SQLステートメントを含む

Cachéを使用した設計モデル

本事例では、継承ではなくコンポジションを使用した
強いカプセル化を行い、開発手順を含めた相互波及を最小限にした

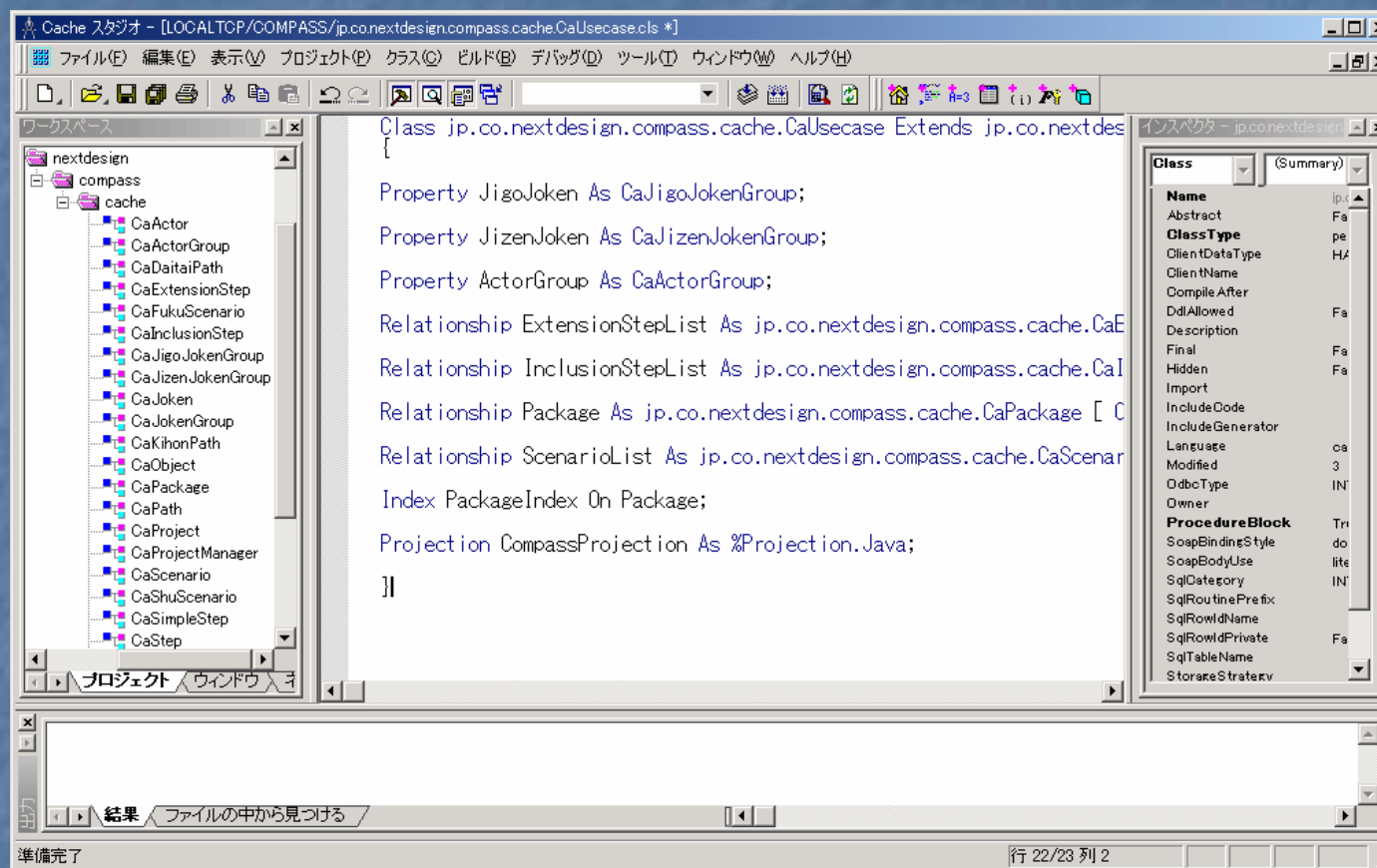


Cachéが自動生成するクラス
永続オブジェクト

開発手順(繰り返し型)

1. オブジェクトモデルを作成する(UML図)
2. Cachéスタジオで永続クラスを定義する
 - ・ 本事例では、オブジェクトの属性のみを定義した
3. CachéプロジェクトでJavaクラスを生成する
 - ・ 自分の開発フォルダーに出力する(Javaソースコード)
4. 他のJavaクラスを作成する
 - ・ テキストエディターやEclipseを使用する(Javaソースコード)
5. 全てをコンパイルする
 - ・ AntやEclipseを使用する
6. 実行テストする(1～6を繰り返す)

Cachéスタジオで永続クラスを定義



Cachéスタジオで永続化クラスを定義し、Javaクラスを生成する

Cachéプロジェクトでクラスを生成

```
jp
  co
    nextdesign
      compass
        cache <----- Casheプロジェクト先
        core      (生成後は、自作のJavaクラスと同じ)
          comparator
          state
          text
        studio
          core
          printing
          ui
        transaction
        util
```


Caché利用方法の選択

- 検討した利用方法

- (1) Cachéスタジオで全てのメソッドも定義する

- (2) Caché永続クラスを継承する

- (3) Caché永続クラスに(永続化のみを)委譲する

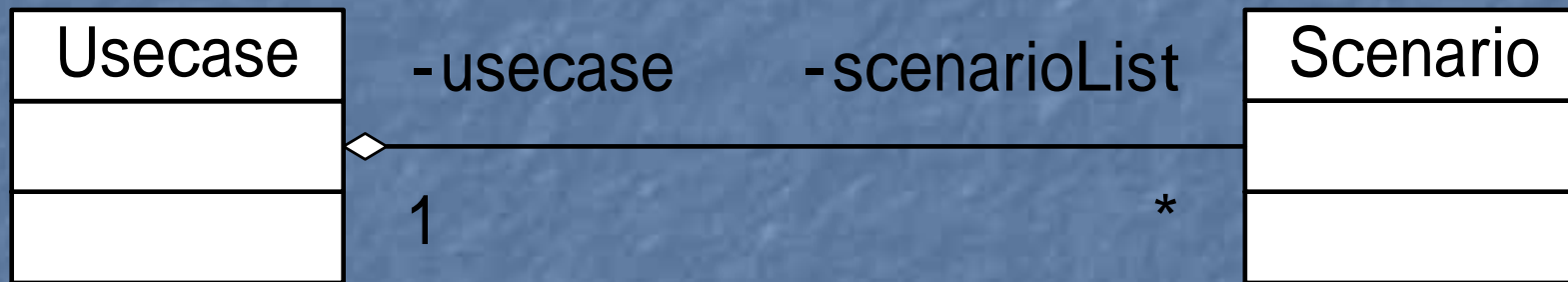
- (3)を選択した理由

- ・ 開発手順のCachéスタジオ依存を小さくする

- ・ Caché自動生成クラスのカプセル化を強める

関連(Relationship)のサポート

1対Nであればプログラマが特別なコードを書く必要がない



(Usecase側の定義)

```
Relationship ScenarioList As Scenario (JAVATYPE = "java.util.Map")
[ Cardinality = many, Inverse = Usecase ];
```

(Scenario側の定義)

```
Relationship Usecase As Usecase [ Cardinality = one, Inverse = ScenarioList ];
```

これらの定義だけで、関連を維持するコードがCaché生成クラスに実装される。

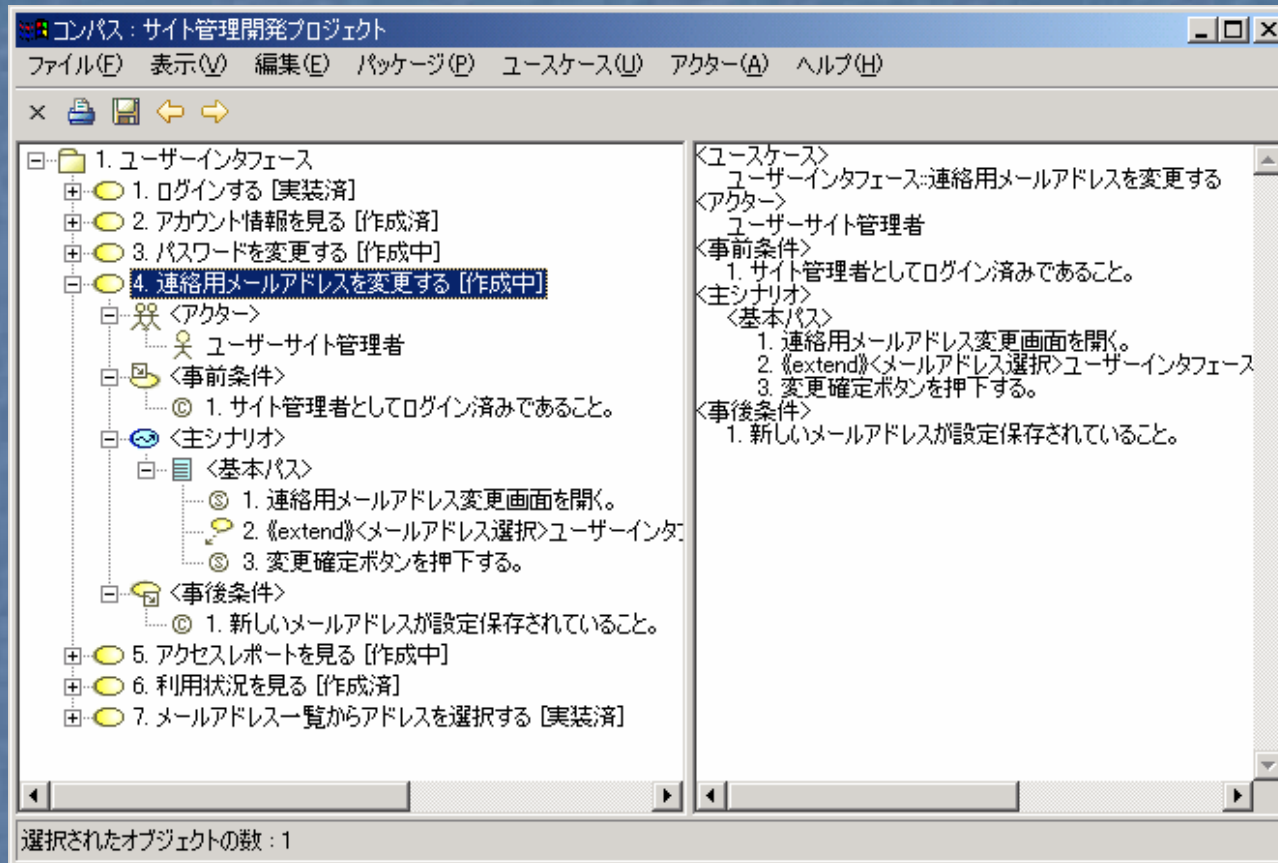
Compassの特徴

- UMLユースケース・モデリング・ツール
- スケジュール管理
- 工数見積り
- オブジェクト指向分析支援



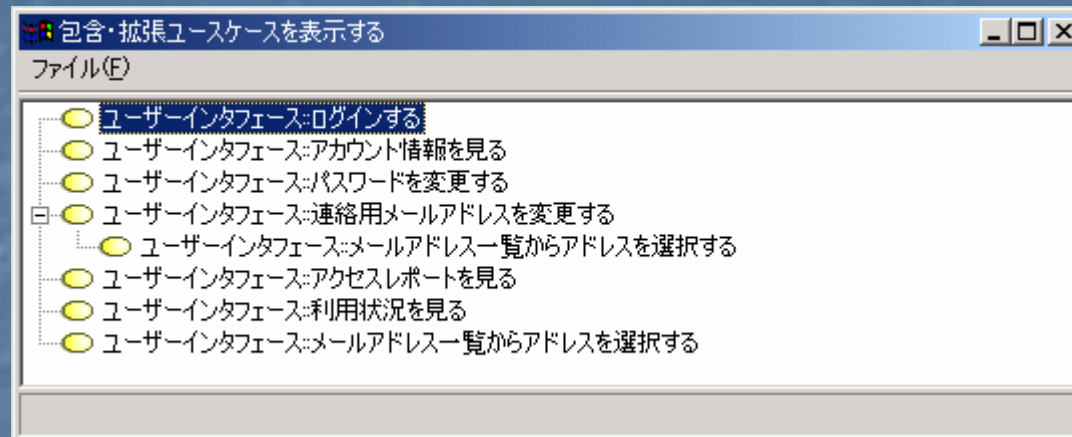
一貫してユースケースを活用できる
ユースケースは、オブジェクト指向開発の軸
「書いて保存した後は利用しない」を解消

Compass Studio

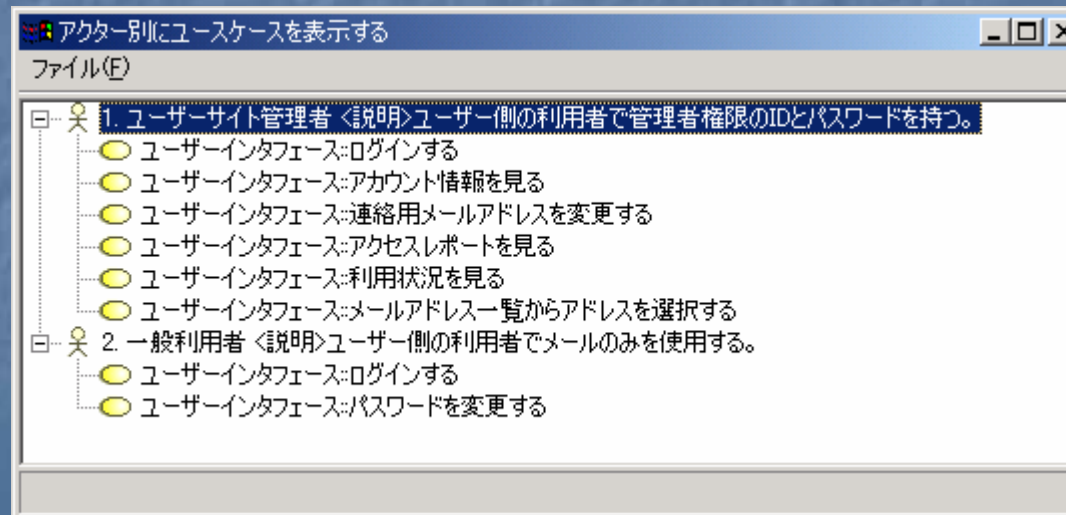


ユースケースの登録・表示・編集・削除
ステータス(進捗度)の登録・表示・編集など

拡張関連、包含関連を表示する



アクター別にユースケースを表示する



本事例の開発規模

- クラス数
 - 全280クラス
 - Cachéが生成した25クラス含む
- ステップ数
 - 全48,439行(うちコメント17,151行)
 - Cachéが生成した23,576行含む(うちコメント4,768行)
- 工数(実装)
 - 本事例では専任開発していなので、約5人月(初期のドメイン分析・設計は含まず)

本事例で再利用したもの

- MVCモデル
- Factoryパターン
- Singletonパターン
- Chain of Responsibilityパターン
- Stateパターン
- Commandパターン
- SWT/JFaceコンポーネント

ご質問など