



BORLAND® DEVELOPER CAMP

## Enterprise Core Objects ファーストチュートリアル

ボーランド株式会社  
QAグループ  
井上 勉

**Borland®**

Copyright (C) 2006, Borland Software Corporation. 本文書の一部または全部の転載を禁止します。



BORLAND® DEVELOPER CAMP  
第2回 ボーランド デベロッパー キャンプ

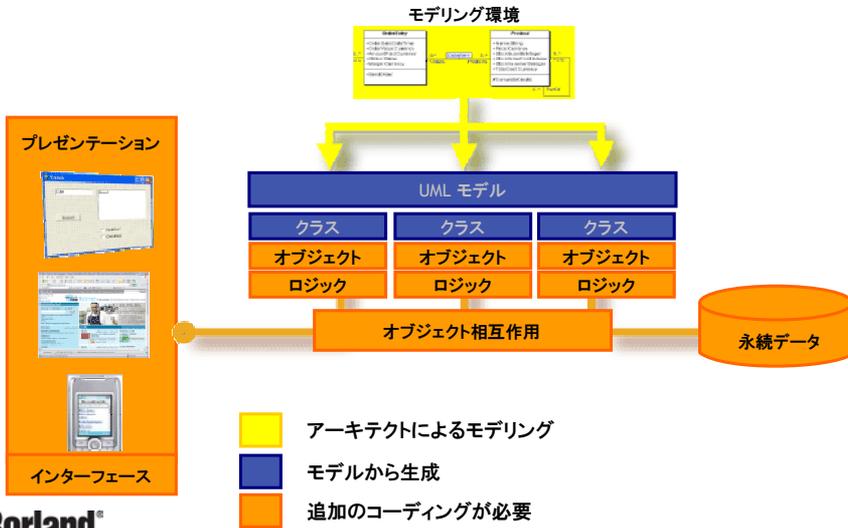
### ECOとは

- ECO = Enterprise Core Objects
  - モデルを実行するランタイムフレームワークを提供
    - モデルレベルで定義した情報を自動的に実行
    - 従来のアプローチのようにモデルを実装するコードを書く必要はない
- 開発時のECOのサポート
  - TogetherによるUMLモデリング
  - OCL式エディタ
- 実行時のECOのサポート
  - データバインディング、O/Rマッピング、UIインターフェース、ステートマシン
  - データの永続化



**Borland®**

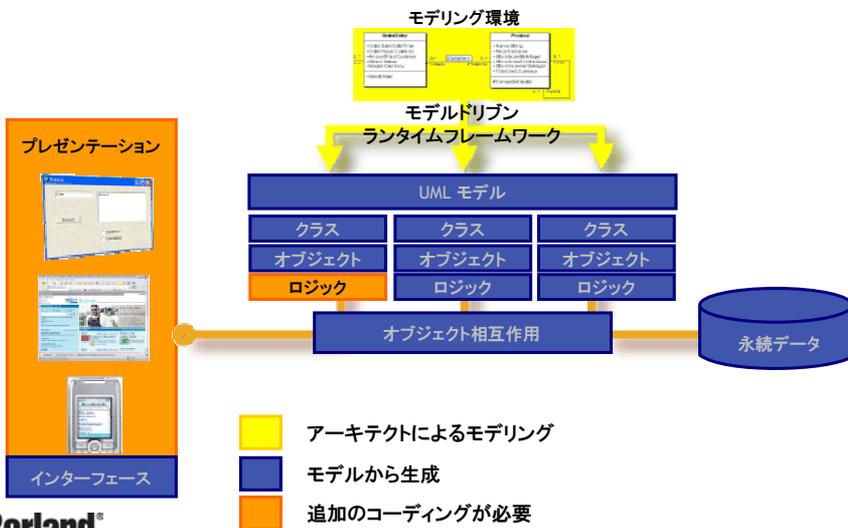
## 従来のモデル指向開発



Borland®

3

## ECOによるデザインドリブン開発



Borland®

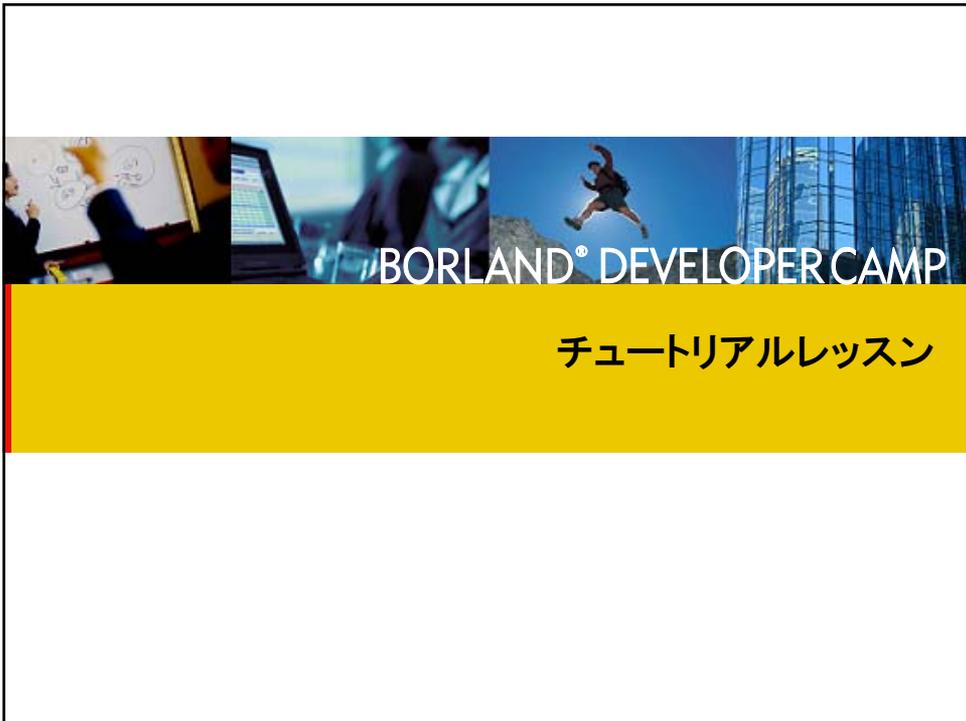
4

## ECOを使うことができる開発環境

	ECOの 基本機能	XMLファイル への永続化	DBへの 永続化	複数のDB への永続化	ステートマシ ンのサポート
Borland Developer Studio 2006 Architect (C# & Delphi.NET)	○	○	○	○	○
Borland Developer Studio 2006 Enterprise (C# & Delphi.NET)	○	○	○		
Borland Developer Studio 2006 Professional (C# & Delphi.NET)	○	○			
Turbo C# Professional & Explorer	○	○			
Turbo Delphi.NET Professional & Explorer	○	○			

## ECOのメリット

- **ビジネスアプリケーションの観点で**
  - 変更に強いアプリケーションの開発
    - モデルを変更すると直ちに実行可能なアプリケーションも修正される
    - ひとつのモデルをリッチクライアント、Webアプリケーション、Webサービスなど、さまざまな形態に展開可能
  - モデルによるアプリケーション開発の抽象化
    - モデルを実装するためのコードを一切書かなくてよいため、高い抽象度でアプリケーションを開発できる
- **オブジェクト指向学習の観点で**
  - すぐに動くモデルを作成できるので、オブジェクト指向設計の結果をただちに確認可能
  - Turbo ExplorerでもECOの基本機能を使用できます！



BORLAND® DEVELOPER CAMP  
第2回 ボーランド デベロッパー キャンプ

## 作成するクラス - 住所録のデータモデル

```
classDiagram
    class Contact {
        address: String
        email: String
        name: String
        phone: String
    }
    class Customer
    class Employee
    Contact <|-- Customer
    Contact <|-- Employee
```

The diagram illustrates a class hierarchy for a contact list. At the top is the **Contact** class, which defines the following attributes: `address: String`, `email: String`, `name: String`, and `phone: String`. Below it are two subclasses: **Customer** and **Employee**. Both **Customer** and **Employee** inherit from **Contact**, as indicated by the hollow-headed arrows pointing from the subclasses to the base class.

**Borland®**

8

Copyright © 2006, Borland Software Corporation. 本記事の一切は著作権が保護されています。



## ECO - 開発の手順

- ECOアプリケーションの新規作成
  - ECO WinFormsアプリケーション、ECO ASP.NET Webアプリケーション、ECO ASP.NET Webサービスなどを選択
- ECOモデルの作成
  - モデルビューツリーからパッケージを選択して、Togetherクラスダイアグラムビューを表示
- ユーザーインターフェースの作成
  - 設計時にユーザーインターフェースがECOクラスの情報を利用できるように、あらかじめ作成したモデルをコンパイルしておくこと
- データの永続化を設定
  - ECO Spaceに永続化を指定

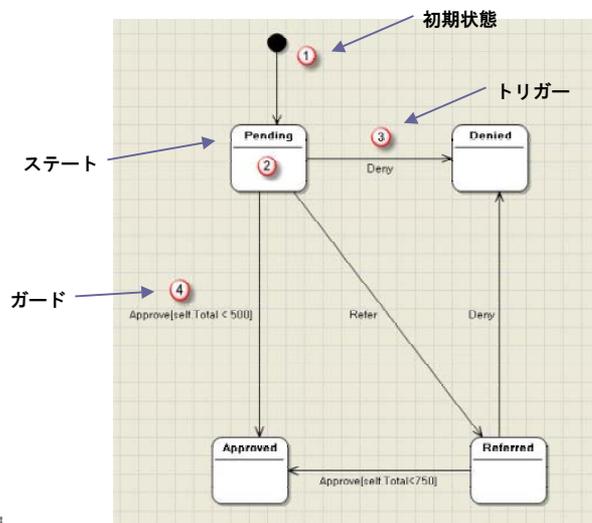


参考:ステートマシン

## ステートマシンとは？

- ステートマシン図
  - オブジェクトの状態(ステート)の遷移を表す図
- ステートマシンを理解するためのキーワード
  - イニシャルステート
    - 開始状態、すなわちオブジェクトの初期値
  - トリガー
    - 状態を遷移させるもの。メソッドとして定義する
  - ガード
    - ビジネスルール。状態遷移のルールを定義する
  - エフェクト
    - 状態遷移の結果発生すること

## ステートマシン図の例





## ECOステートマシン

- ECOフレームワークが実現してくれること
  - ステートマシン図で定義した、トリガー、ガード、エフェクトを実装コードを記述することなしに実行する
    - ステートマシン図およびオブジェクトインスペクタによる定義
    - OCL式エディタによる条件(ガード)の定義
  - ECOデバッガによるテスト
    - コードを記述することなく、インスタンスの生成、削除などをビジュアルに実行してテスト
    - ECO AutoFormにより、自動ビジュアルフォームで、ステートの遷移、オブジェクトの値変更などを行いテスト



Thank you