UML Modeling与 Together For VS.NET

? ? Gordon Li
Borland大中华首席技术官

ULL NAME
TLE POSITION

Borland
Excellence Endures

Agenda

- ? ? ? ? ? Modeling
- Best Practices
- UML? ?
- UML Diagrams and Extensions
- Design pattern
- ? ? Borland Together For .NET? ? Modeling? ? ? ? ?

Borland®

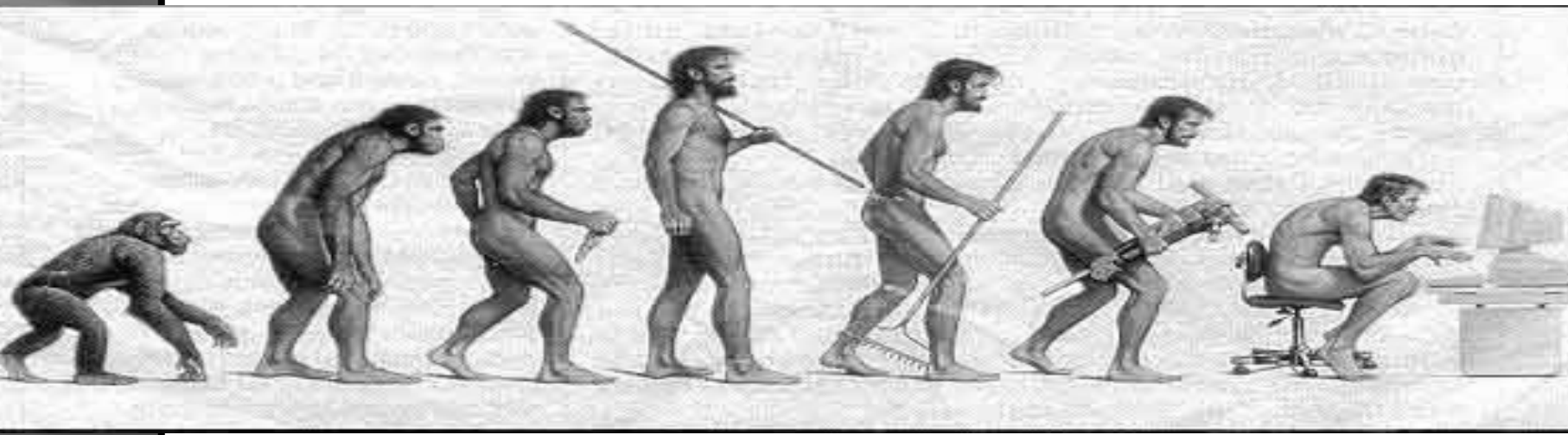
为什么需要Modeling

其实我们一直在做Modeling!

Borland®

为什么需要Modeling

其实我们一直在做Modeling!



Modeling Cycle

Modeling Process

Modeling Development

Modeling Code

Borland®

为什么需要Modeling

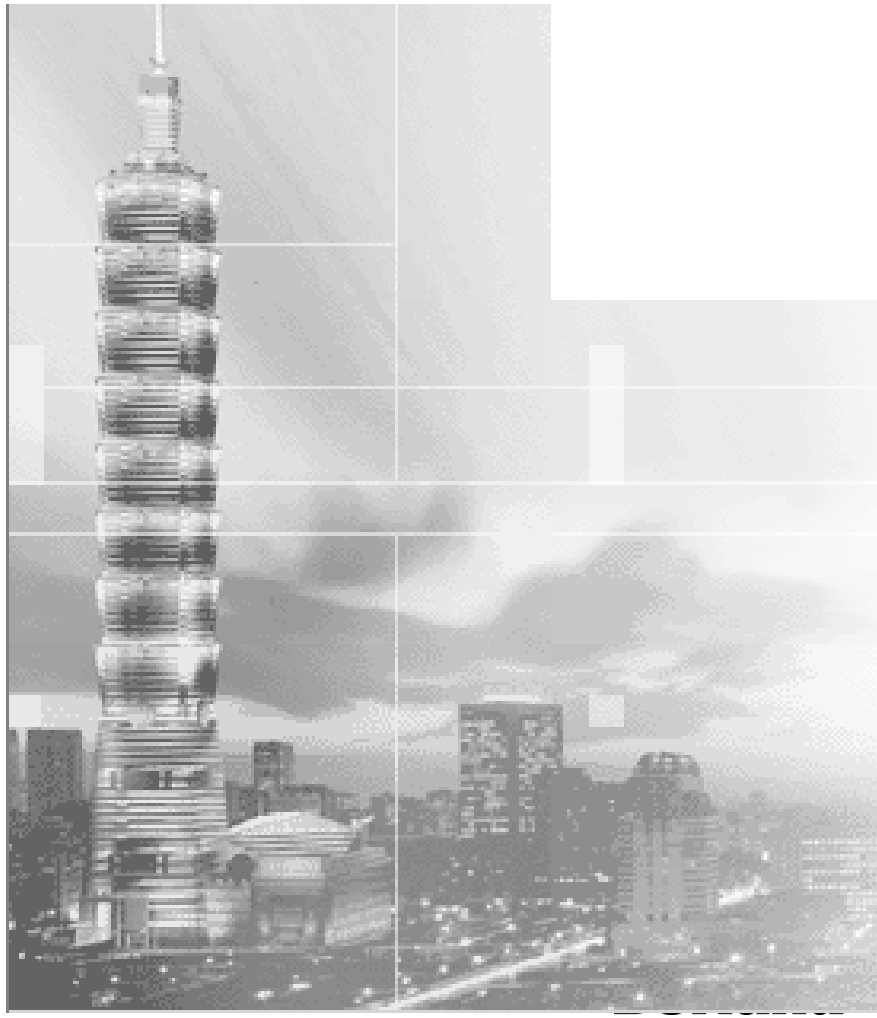
- 问一个简单的问题, 您了解你的产品/项目吗?
  - Microsoft Duwamish范例

Borland®

为什么需要Modeling

Simple vs. Complex
One man vs. Team work
Non-Critical vs. Critical

VS.



Model & Blueprint

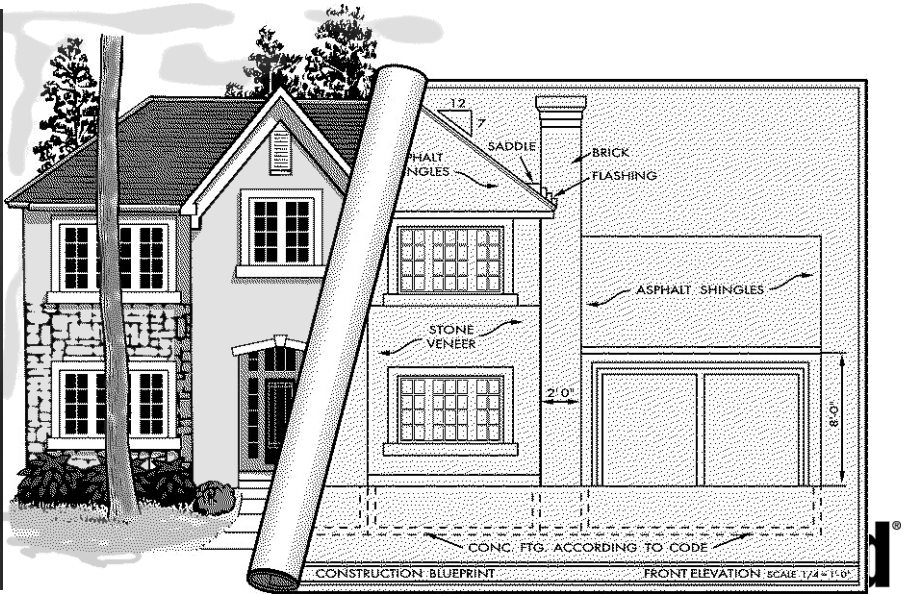
模型与蓝图是所有参与者的沟通根据

Business Model

Requirement Model

Analysis Model —Platform Independent Model

Design Model —Blueprint for implementation



Best Practices

From IBM Rational - RUP

Develop Iteratively
Manage Requirements
Use Component Architectures
Model Visually (UML)
Continuously Verify Quality
Manage Change

Component architectures is evolved into Service Oriented architecture.

Borland®

Once Upon A Time ...

Borland®

If you only knew the power of the dark side.

® ™ &© ? ? ? ? ? ? ? ? ? ?

Borland®

Preparation and Recommendation

- Mindset
  - Not religion, but rigorous
  - Incremental adoption
- Training
  - Learning & Training
  - Apprenticeship
  - Mentors
- Tools are required
- Agile Process
- Start Small

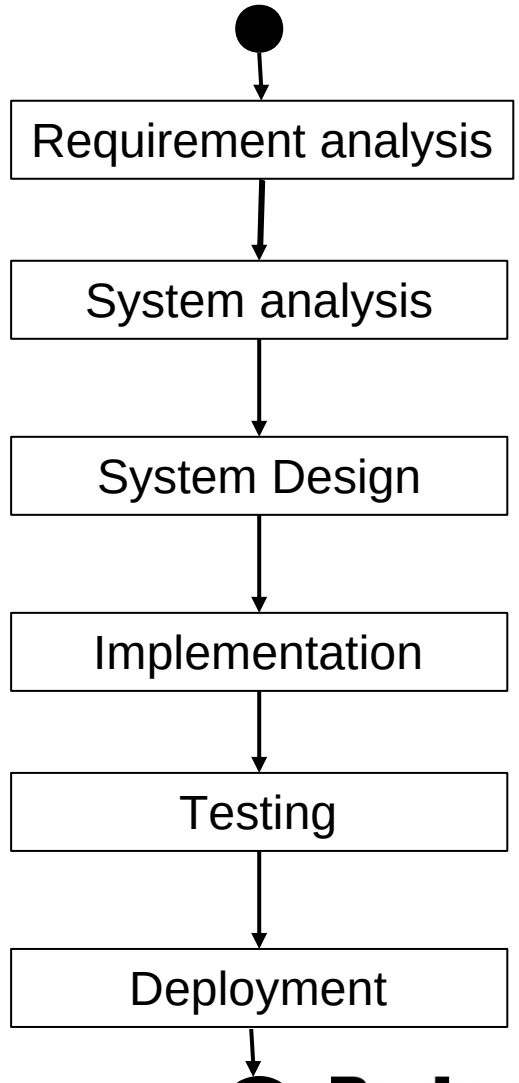
® ™ &© ? ? ? ? ? ? ? ? ? ?

Borland®

The Process

- Business Model
- Requirement Model
- User Experience Model
  - look-Feel & Interaction
- Analysis Model
- Design Model
  - Architecture design
  - Data Model
- Implementation Model
- Test Model
- Deployment Model

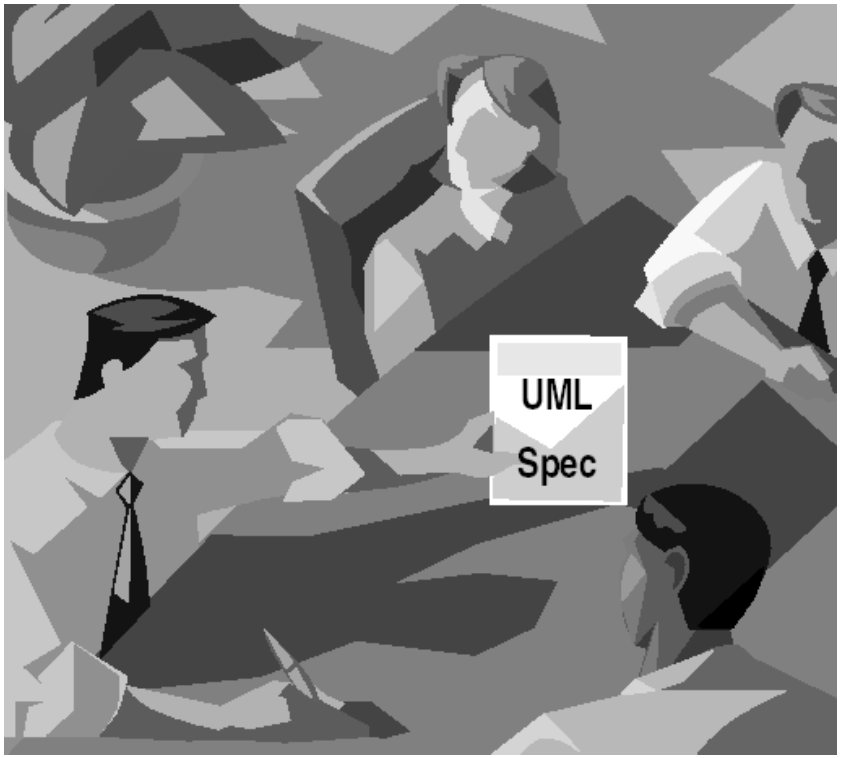
From IBM Rational - RUP



Borland®

UML: Unified Modeling Language

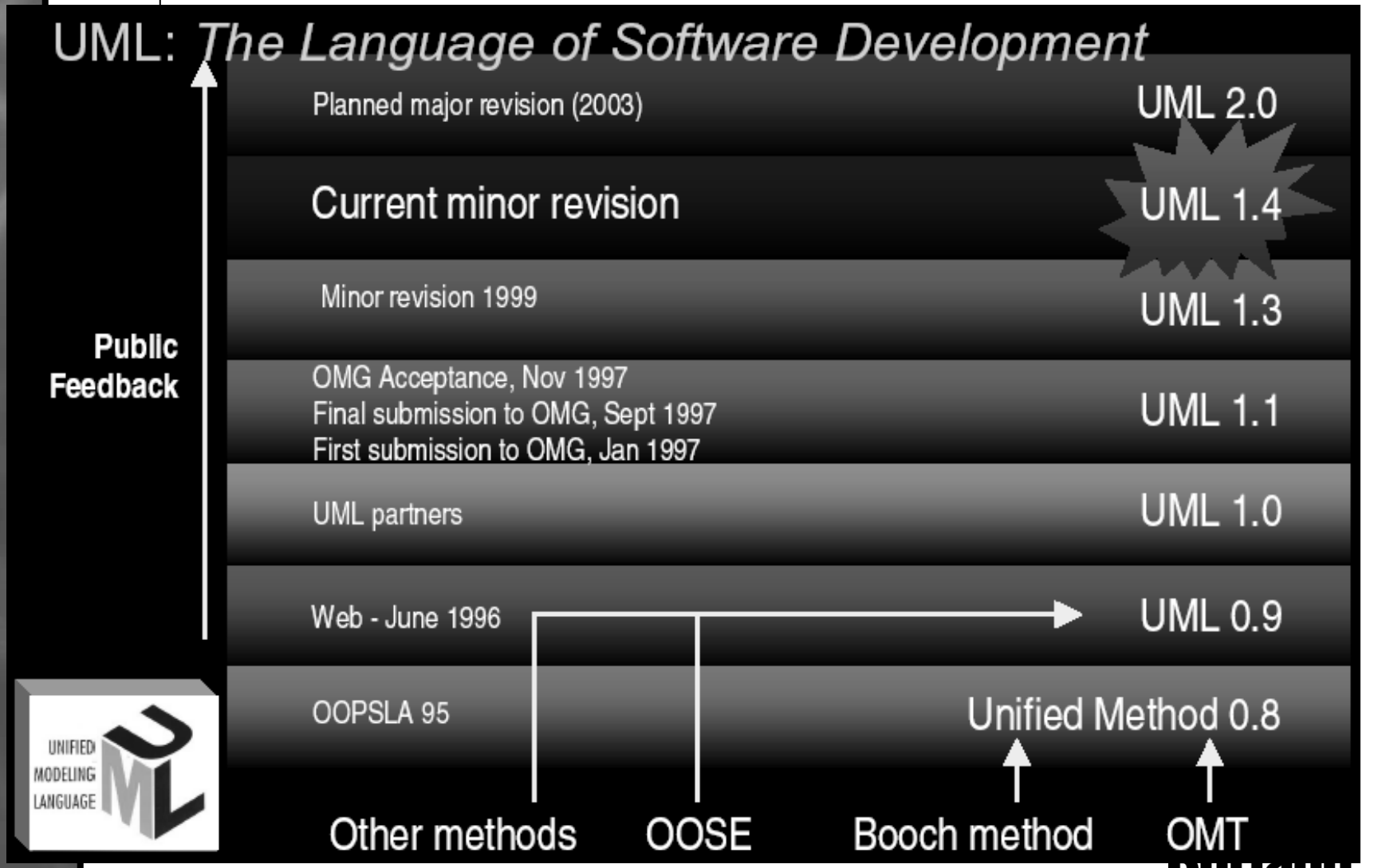
The UML is the standard language for visualizing specifying, constructing, and documenting the artifacts of a software-intensive system



From IBM Rational - RUP

Borland®

UML? ? ? ?


UML: The Language of Software Development
Planned major revision (2003)
UML 2.0
Current minor revision
UML 1.4
Minor revision 1999
UML 1.3
Public Feedback
OMG Acceptance, Nov 1997
Final submission to OMG, Sept 1997
First submission to OMG, Jan 1997
UML 1.1
UML partners
UML 1.0
Web - June 1996
UML 0.9
OOPSLA 95
Unified Method 0.8
UNIFIED MODELING LANGUAGE
Other methods
OOSE
Booch method
OMT

The Value of the UML

Is a standard

Supports the entire software development lifecycle

Supports diverse applications areas

Is based on experience and needs of the user community

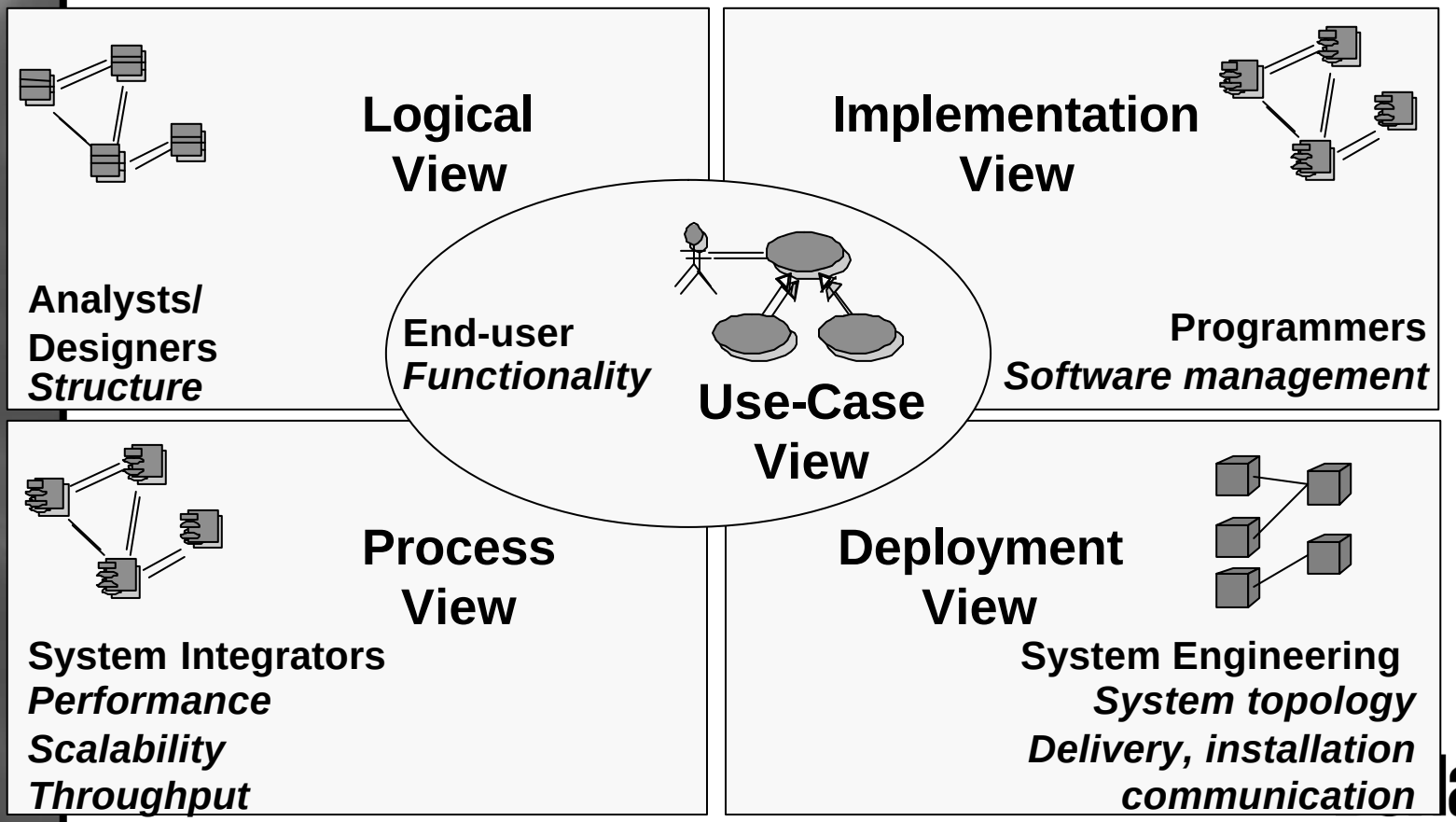
Supported by many tools

Borland®

Model, View, Diagram

Model contains views for different purposes.

View contains multiple diagrams.



UML 12 Diagrams

Behavior :

- Use Case
- Activity
- Sequence
- Collaboration
- State Chart

Structural:

- Class
- Component
- Deployment
- Object

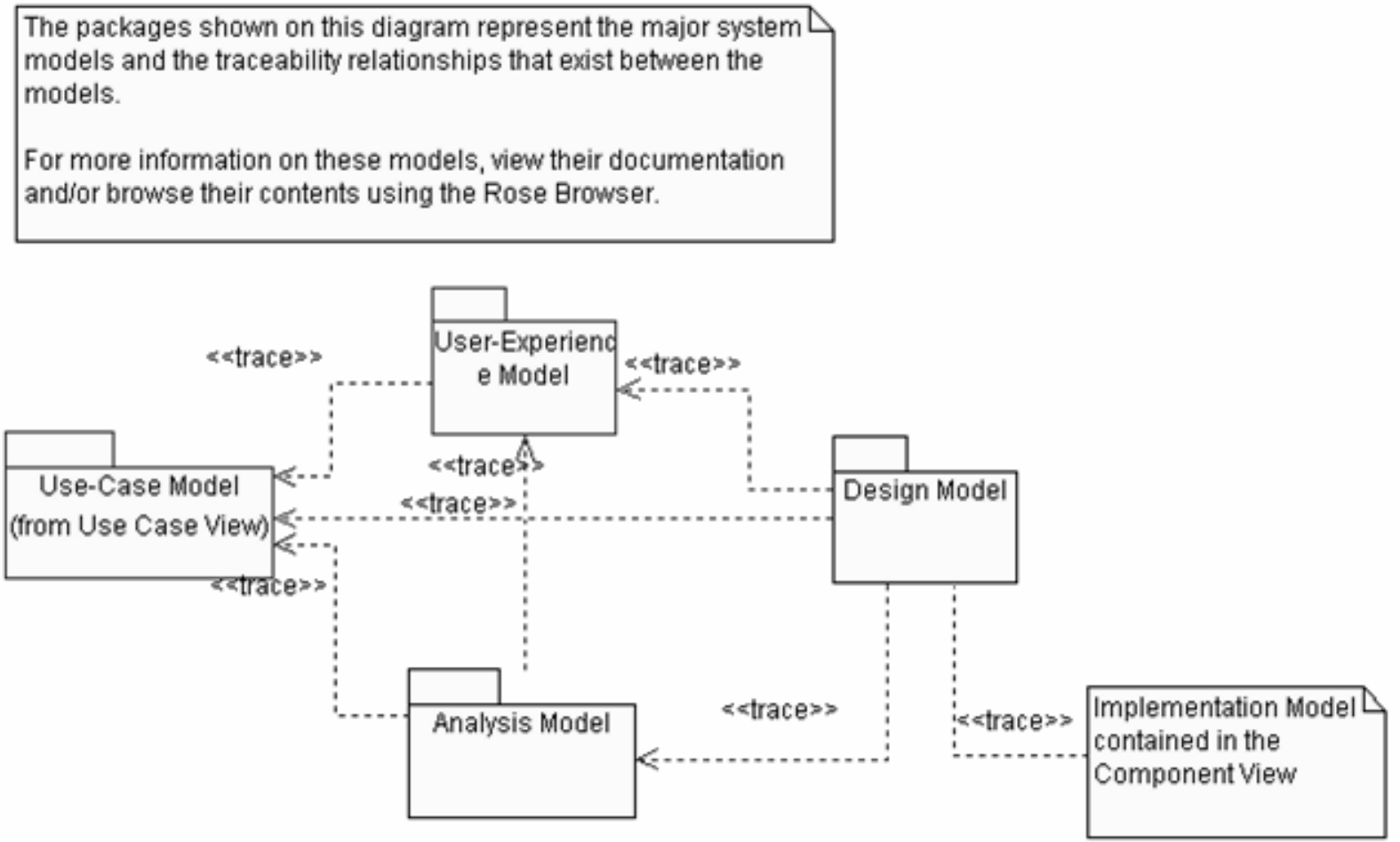
Model Management:

- Packages (class diagram contains packages)
- Subsystems (class diagram contains subsystems)
- Models (class diagram contains models)

Borland®

Model Diagram

使用Model Diagram**描述所要执行的**Modeling**工作**



需求模型： Use-Case Model

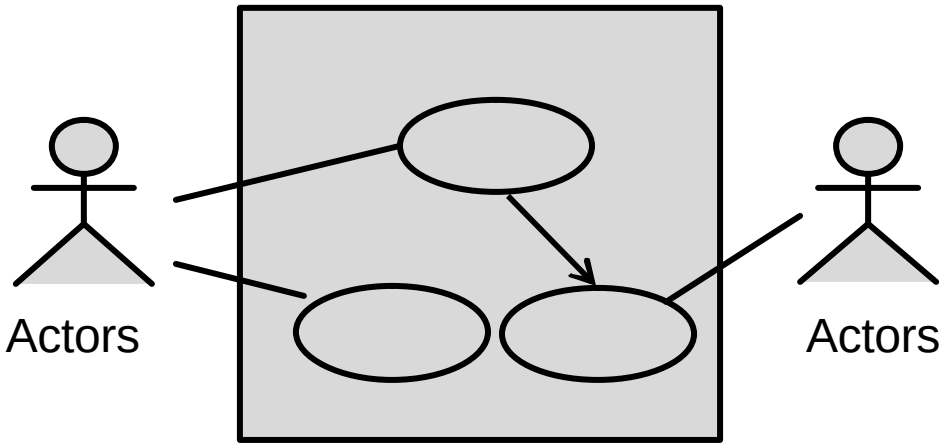
Use case model**描述系统功能，是与客户沟通的工具**

Use Cases: **系统行为或功能**

Actors: **系统的各种使用者**

Use Cases Diagrams

Use-Case Reports or Use-Case Specifications



Use-Case Diagram

Borland®

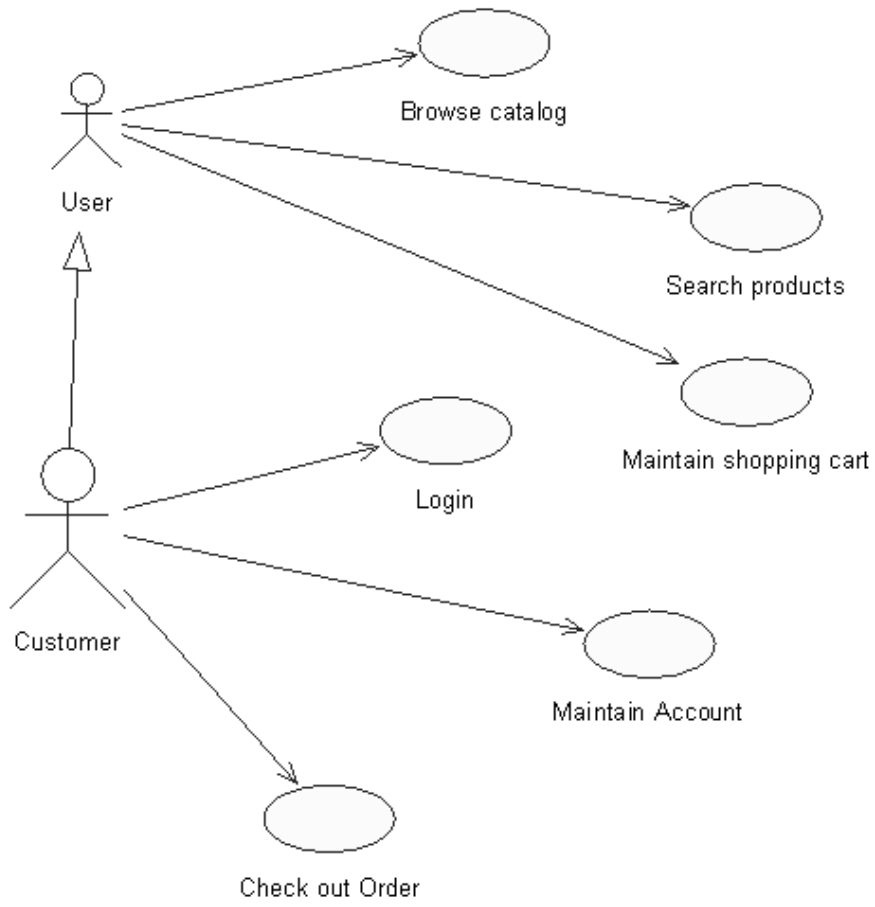
网络购物系统的需求模型

Use cases:

- Browse catalog
- Search products
- Maintain shopping cart
- Logon
- Maintain account
- Check out order

Actors

- User
- Customer



Borland

Use case report

This is requirement model version.

1. Name: Browse Product Catalog
2. Flow of Events
   2.1 Basic flow: **显示所有的次类别及促销产品**
   1.**使用者开启首页**
   2.**系统显示所有的次类别, 与当时的促销产品**
   3.**使用者选取次类别**
   4.**系统显示该类别的全部产品简介**
   5.**使用者选择产品**
   6.**系统显示该产品的信息**

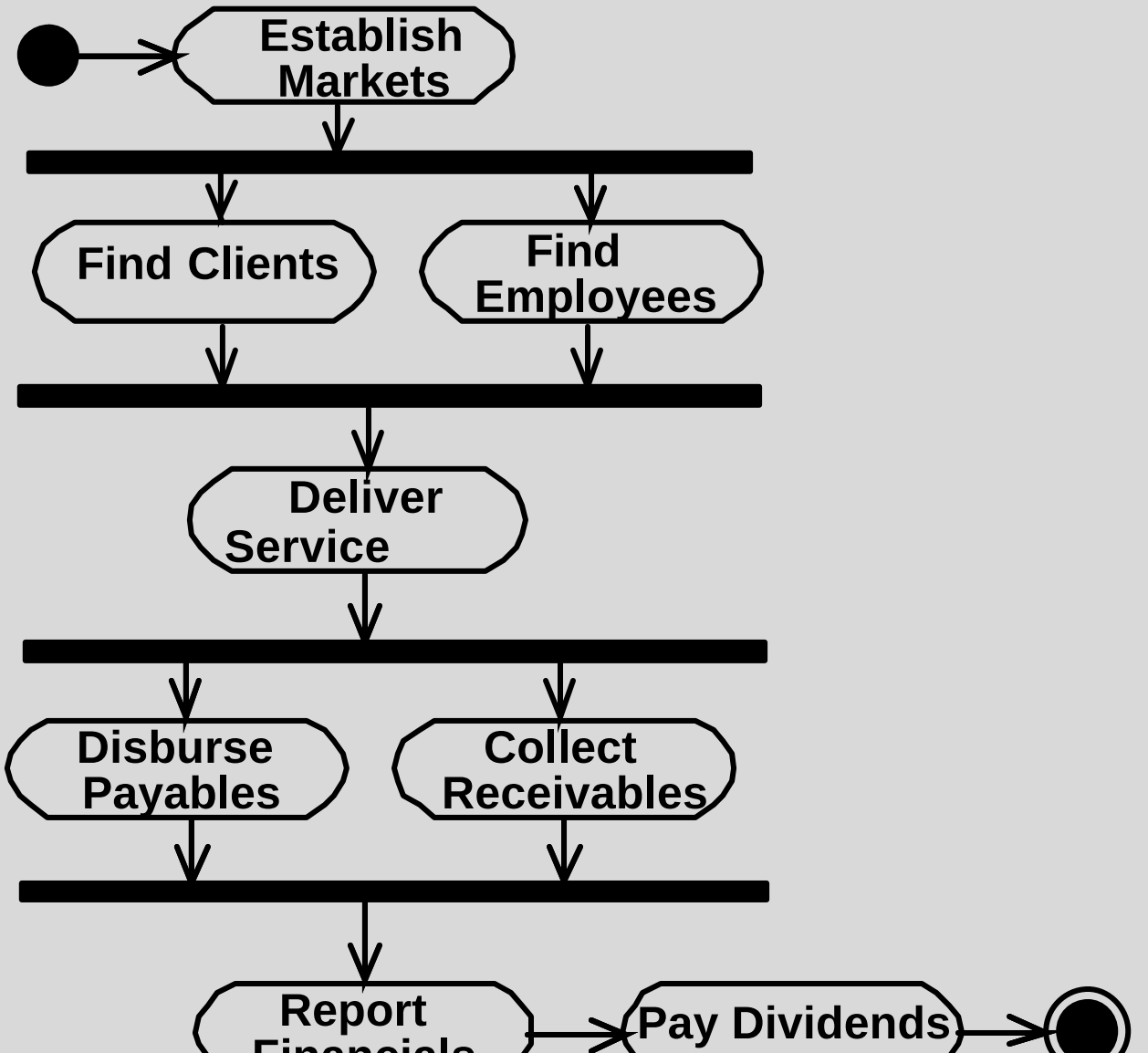
Borland®

Activity diagram

描述Business process**或**use case**的操作流程**

Borland

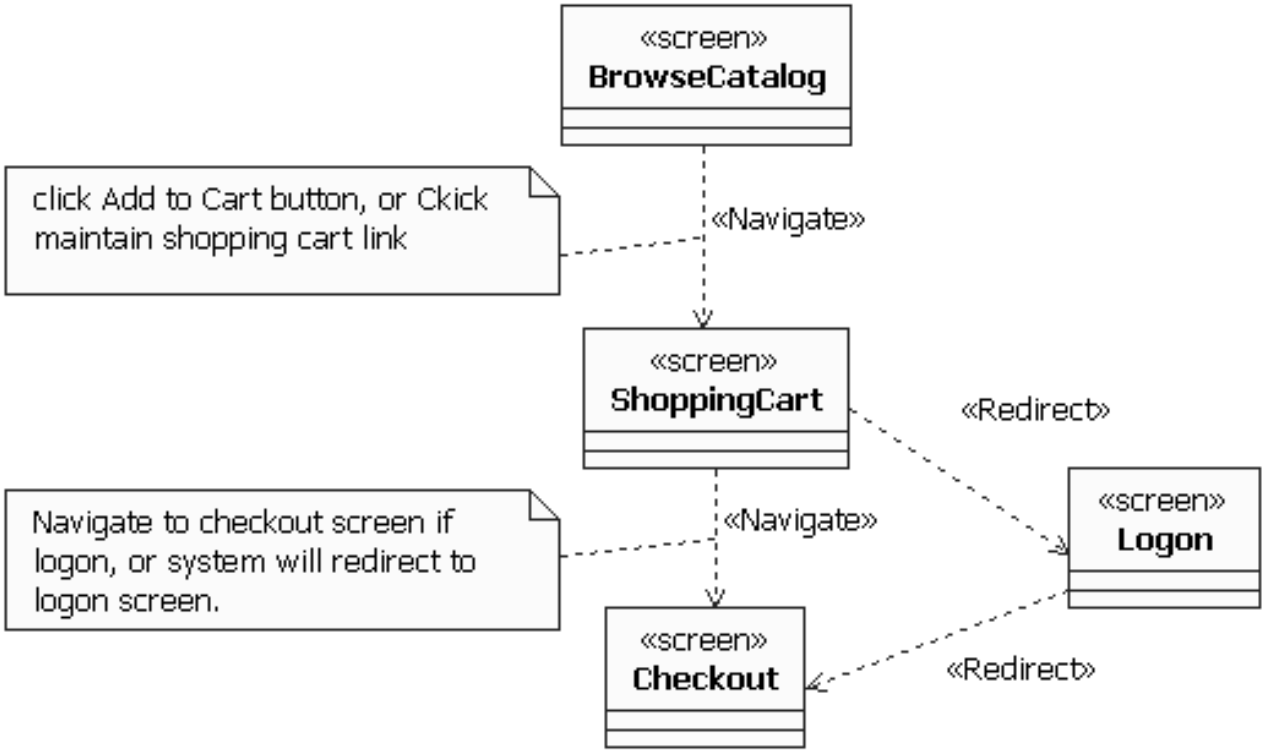
Business Modeling



User Experience Model

UI Prototype

Navigate flow: use UI class with screen stereotype



borland®

More Detailed Use Case

This is User Experience Model version, Optional.

1. Name: Browse Product Catalog
2. Flow of Events

2.1 Basic flow: ? ? ? ? ? ? ? ? ? ? ? ? ?

1. ? ? ? ? ? ? ?
2. ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?
3. ? ? ? ? ? ? ? ? ? ? pick of today? ? ? ? ? ? ?
4. ? ? ? ? ? ? ? ?
5. ?
6. ? ? ? ? ? ? ?
7. ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? , ? ? ? AddToCart?

Borland®

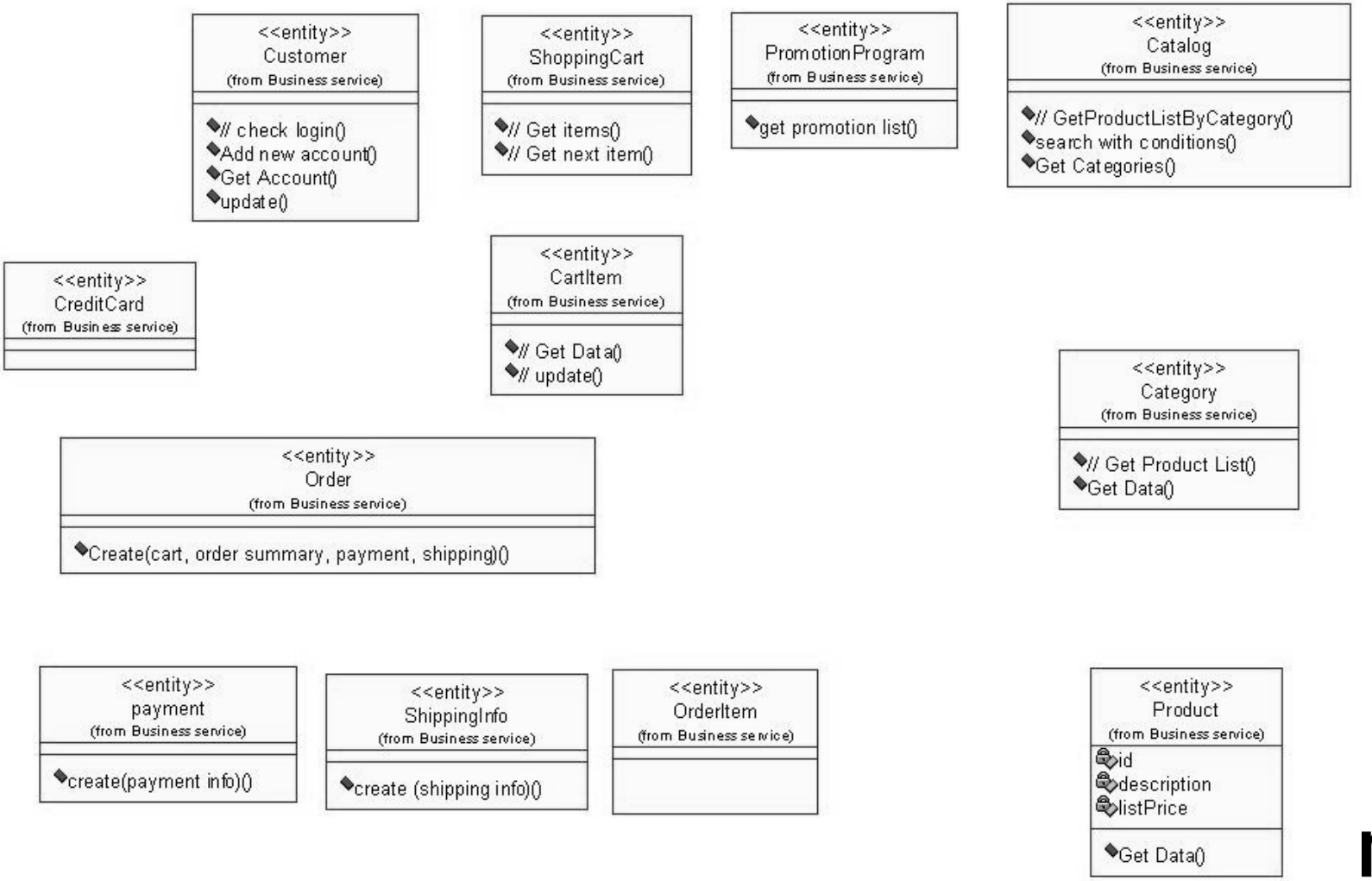
Analysis model

- Platform Independent Model
- **找出完成**use cases **所需要的类别**
  - Analysis classes
- **规划**Classes**的**relationship
- **指派**use-case behaviors**给适当的类别**
  - ？ ？ Operations
- **找出每个**class**的**attributes

Borland®

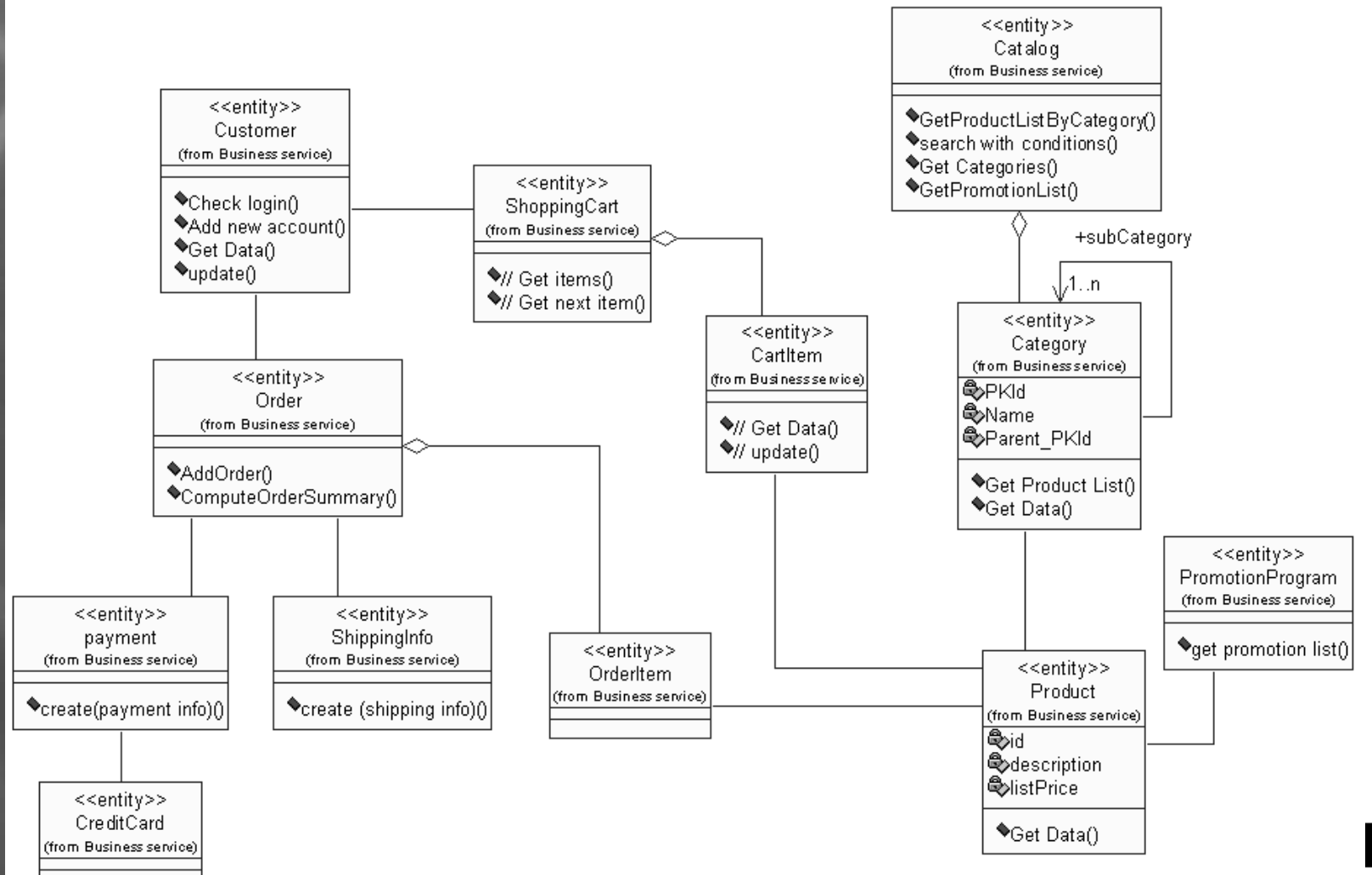
Analysis Classes

- Class name, operations, attributes



Class diagram

Find Relationships, be care of being too complex



Architecture design

Architecture design**处理较高层次的设计组件，包括**package, Subsystem, Interface**。**

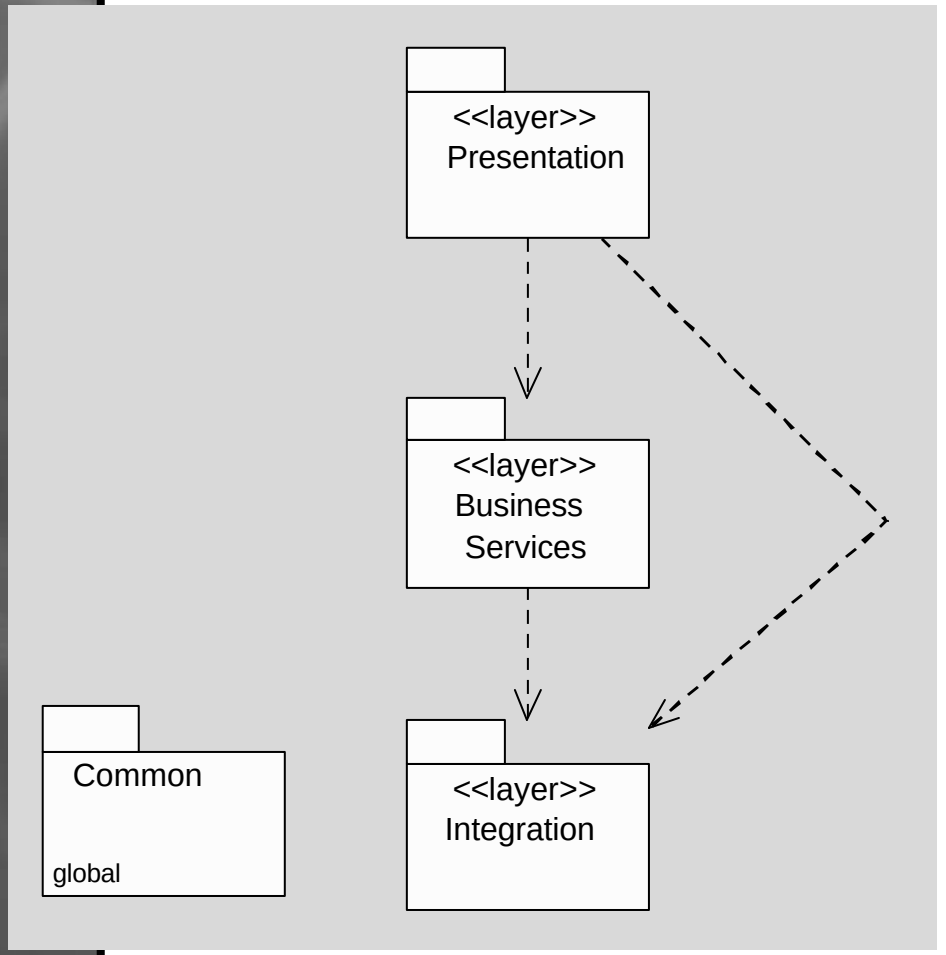
使用Package**组织设计组件。**

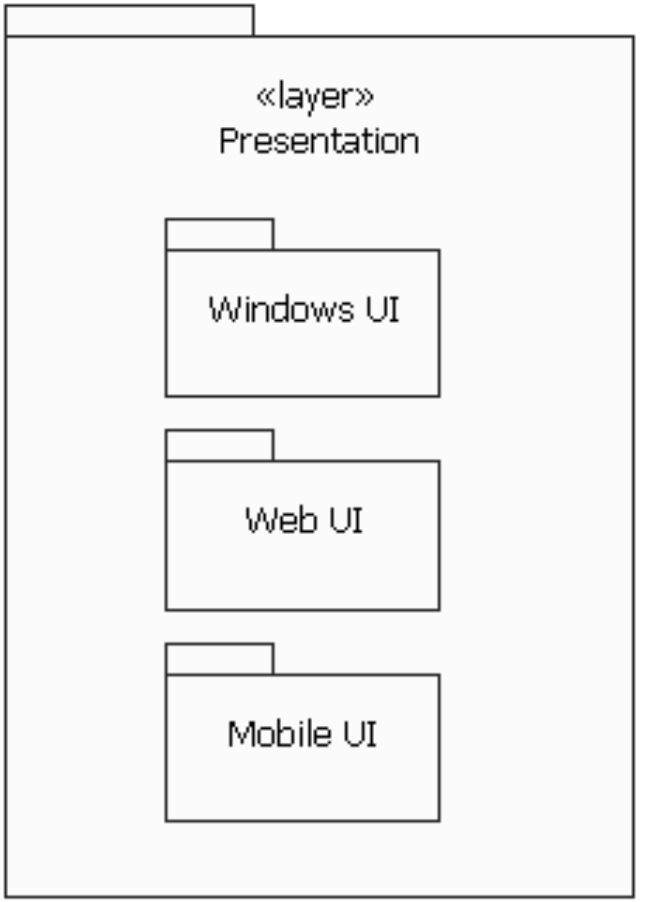
Subsystem**包含多个**classes, **透过**Interfaces**展现功能。**

Borland®

Package diagram

Package diagram is a class diagram which contains packages.

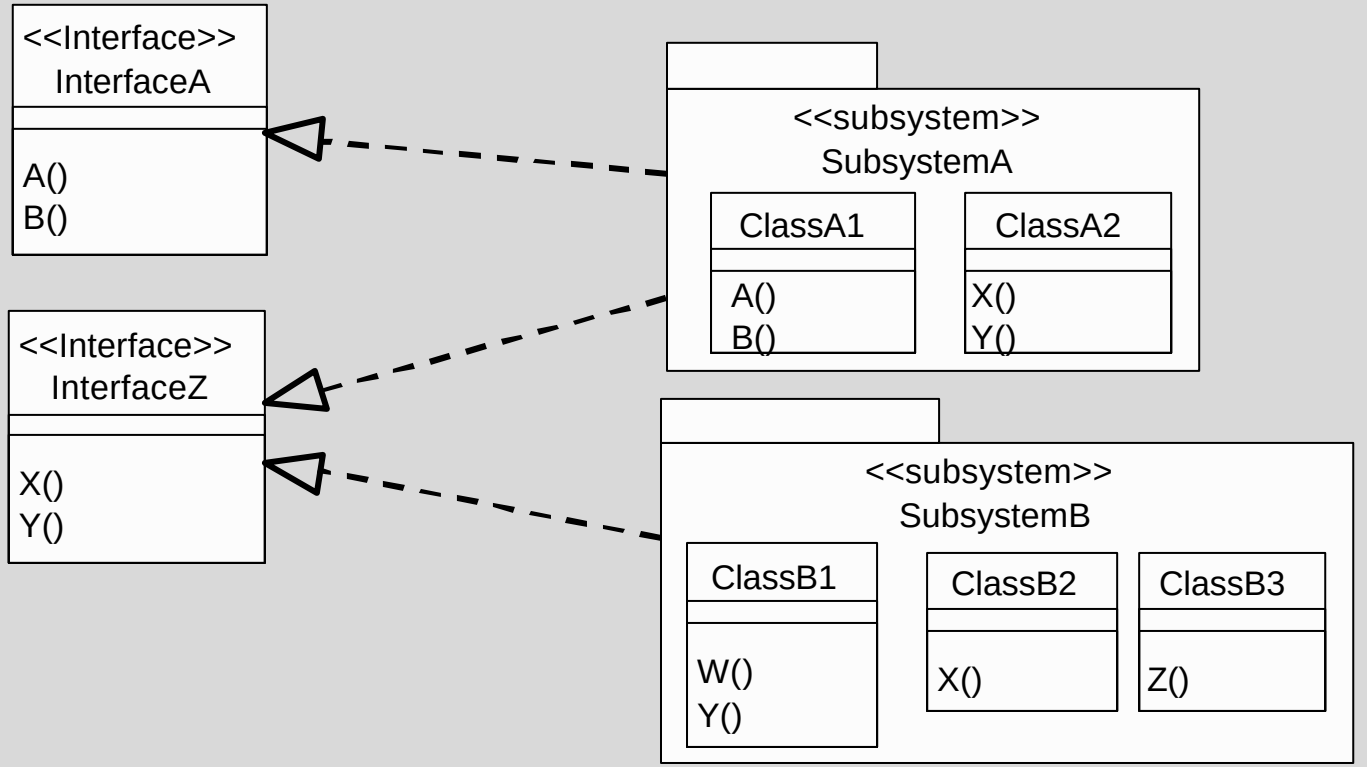




Subsystem diagram

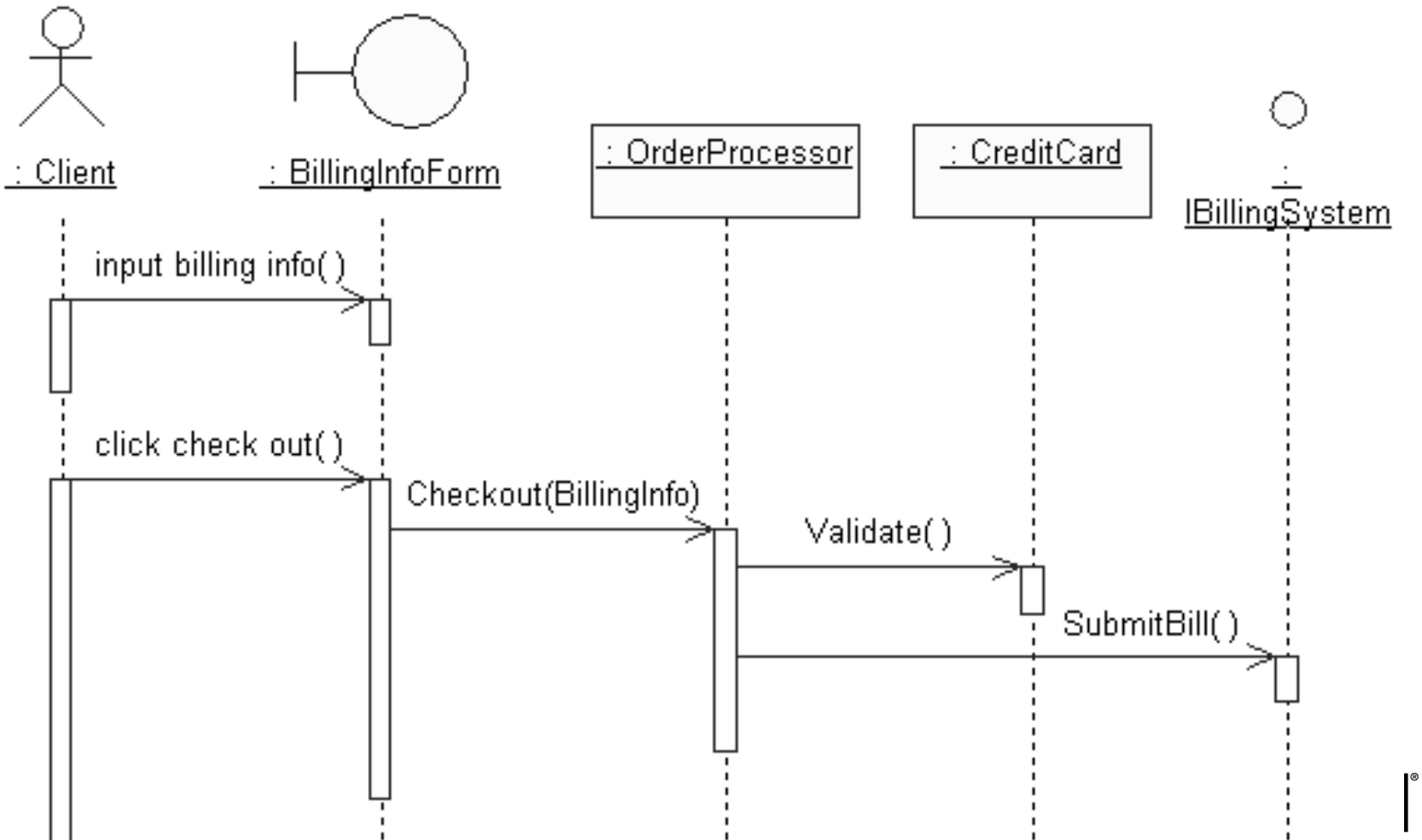
Subsystem**可以**realize**两个以上的接口**

一个接口可以被二个以上的子系统realize



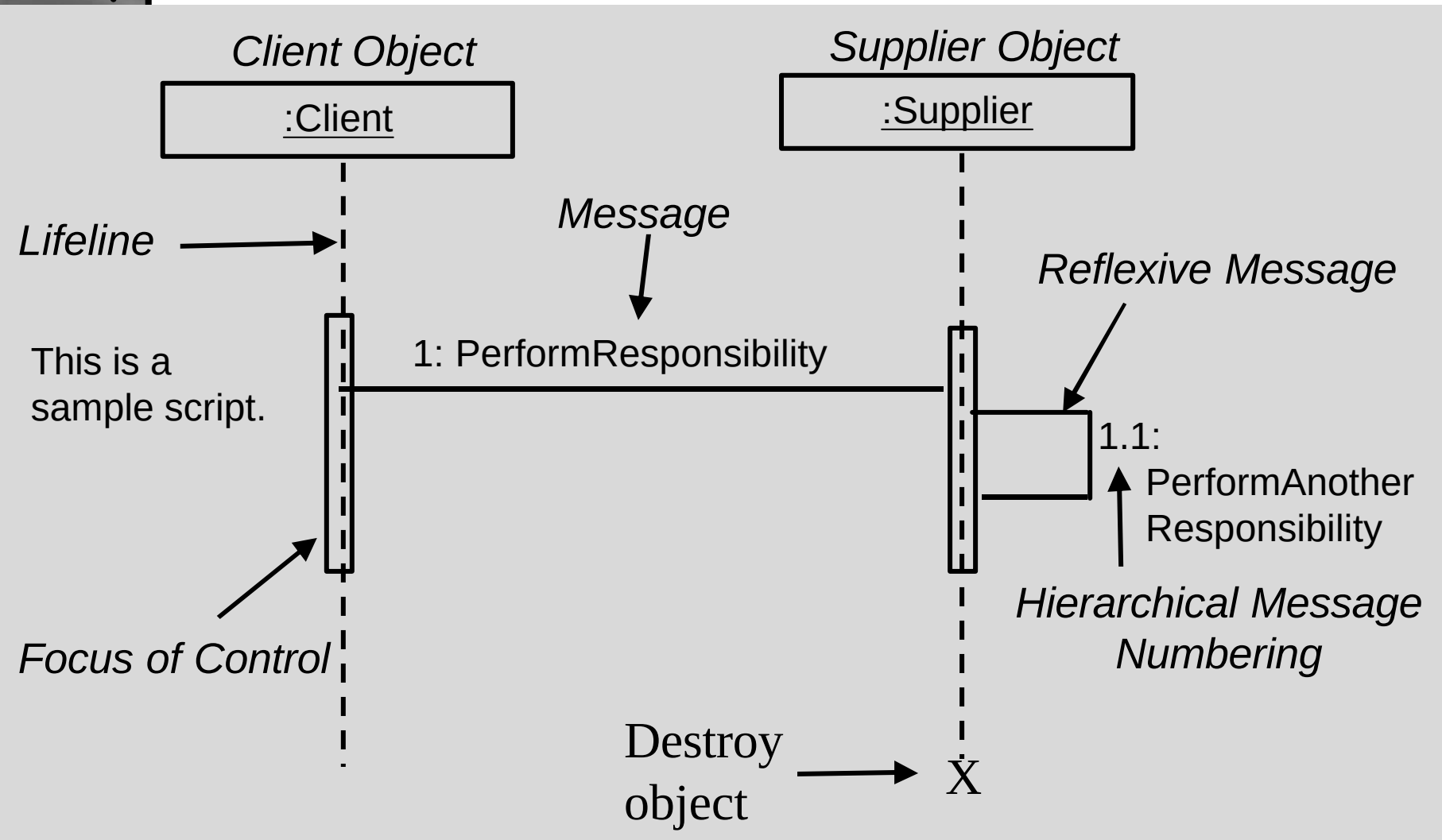
Use Case Design

Walk thru flow of events in use case by sequence diagram



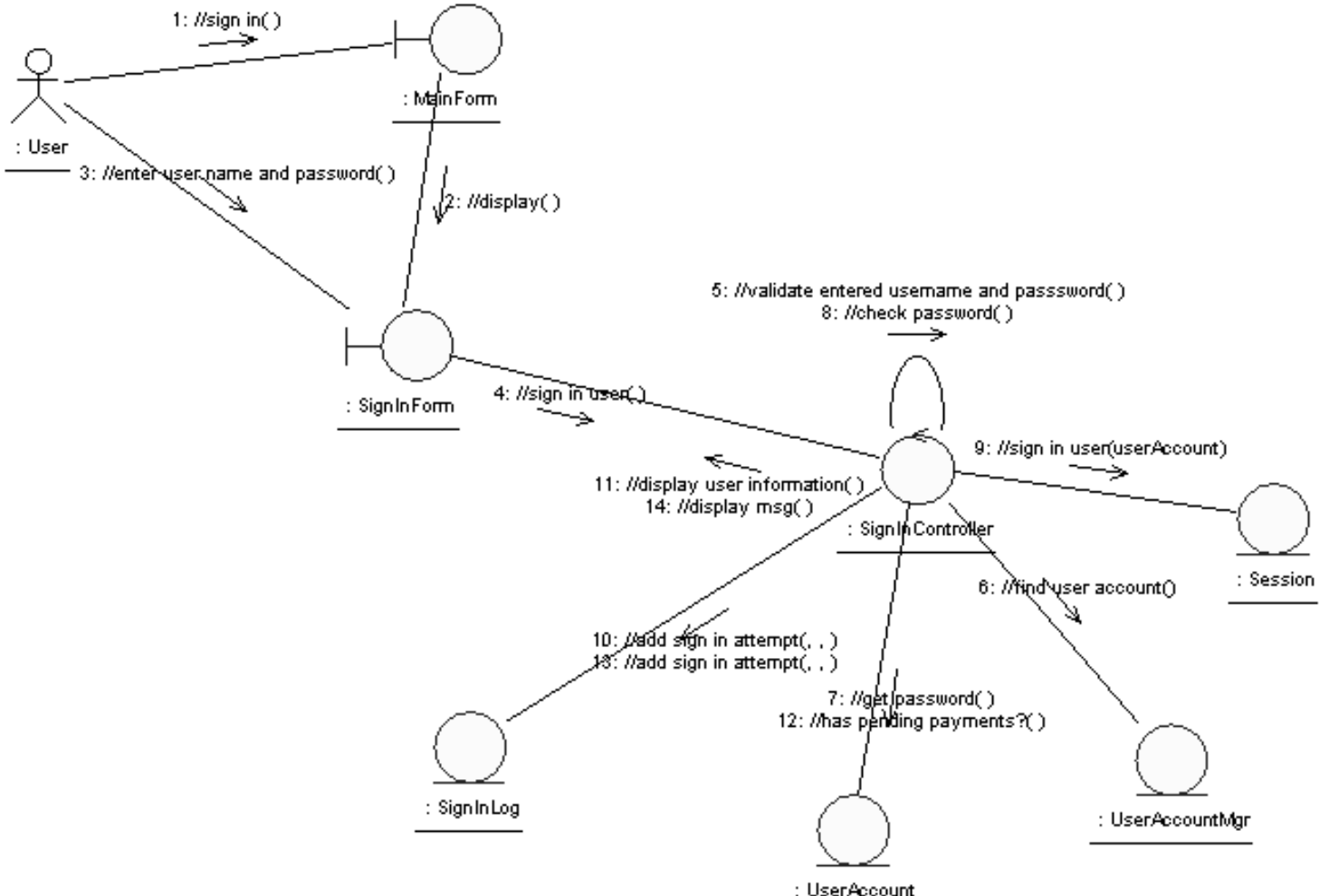
Sequence Diagrams

A sequence diagram displays object interactions arranged in a time sequence



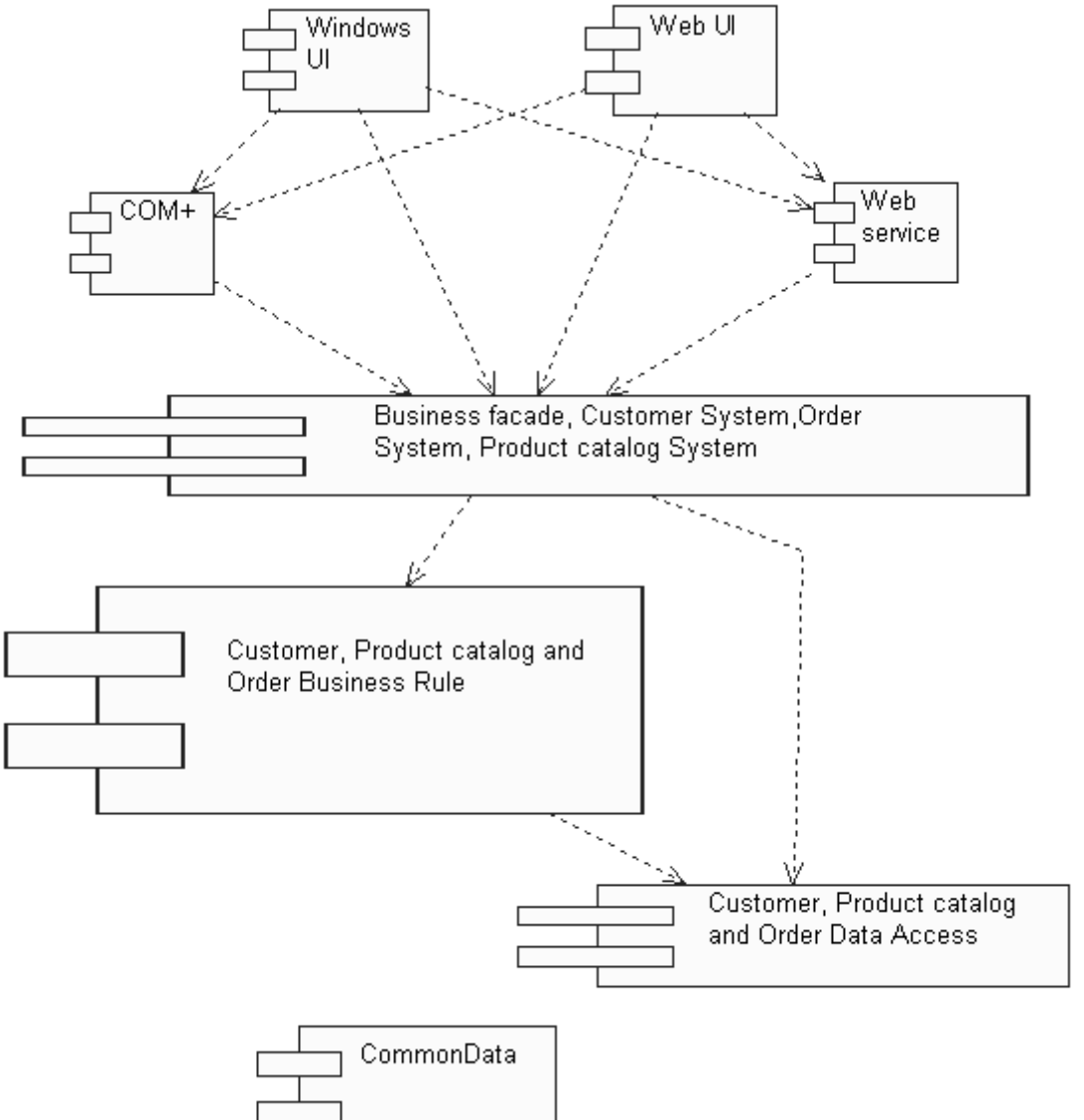
Collaboration Diagram

A collaboration diagram displays object interactions organized around objects and their links to one another

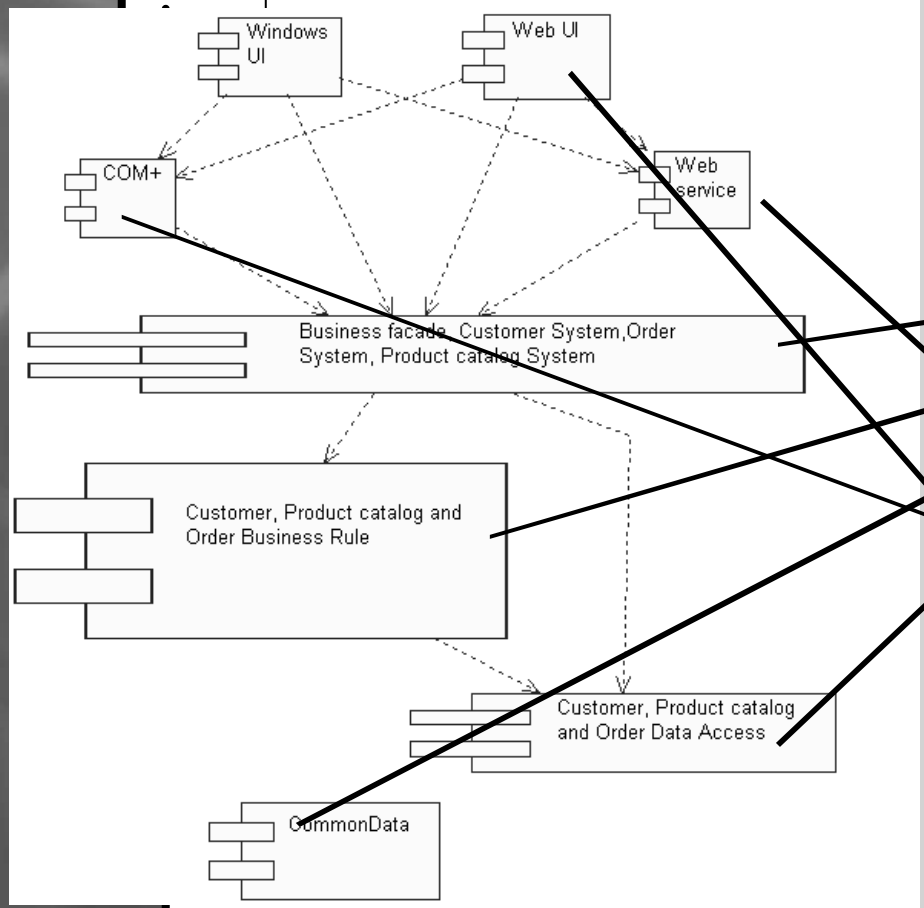


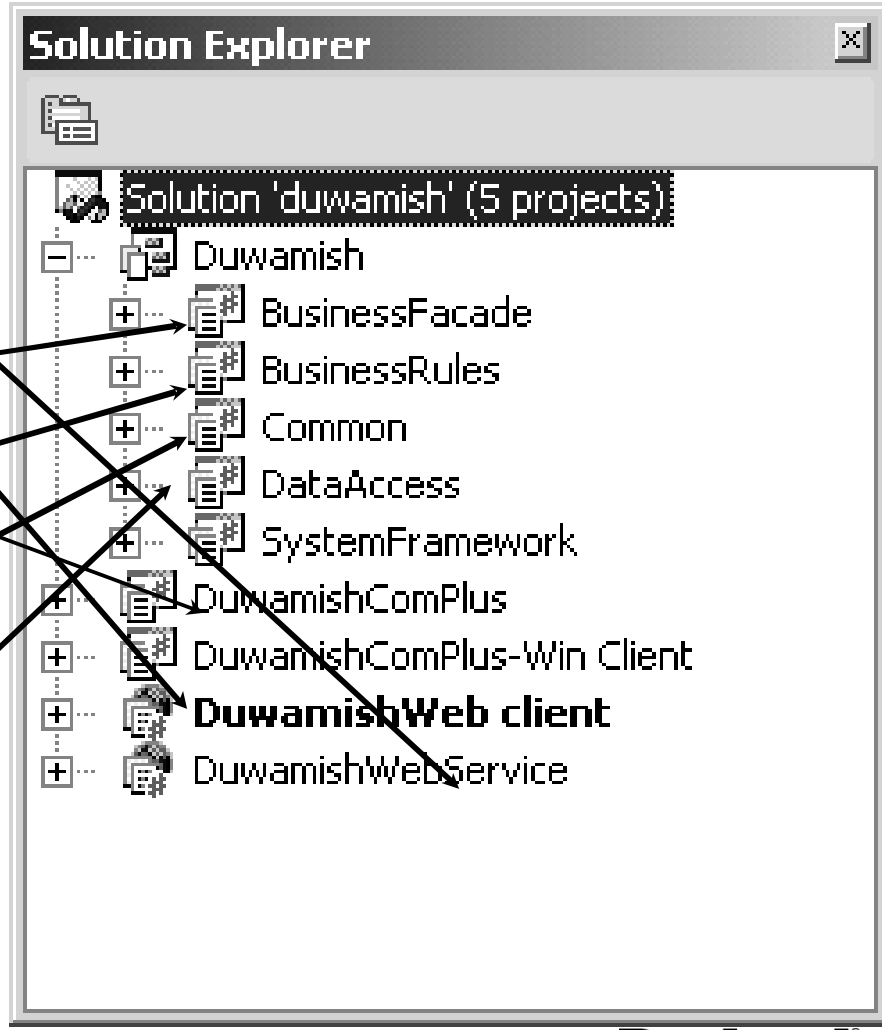
Implementation Model and Component Diagram

For small system, we use
Business fa çade component
Business rule component, a
Data access component.



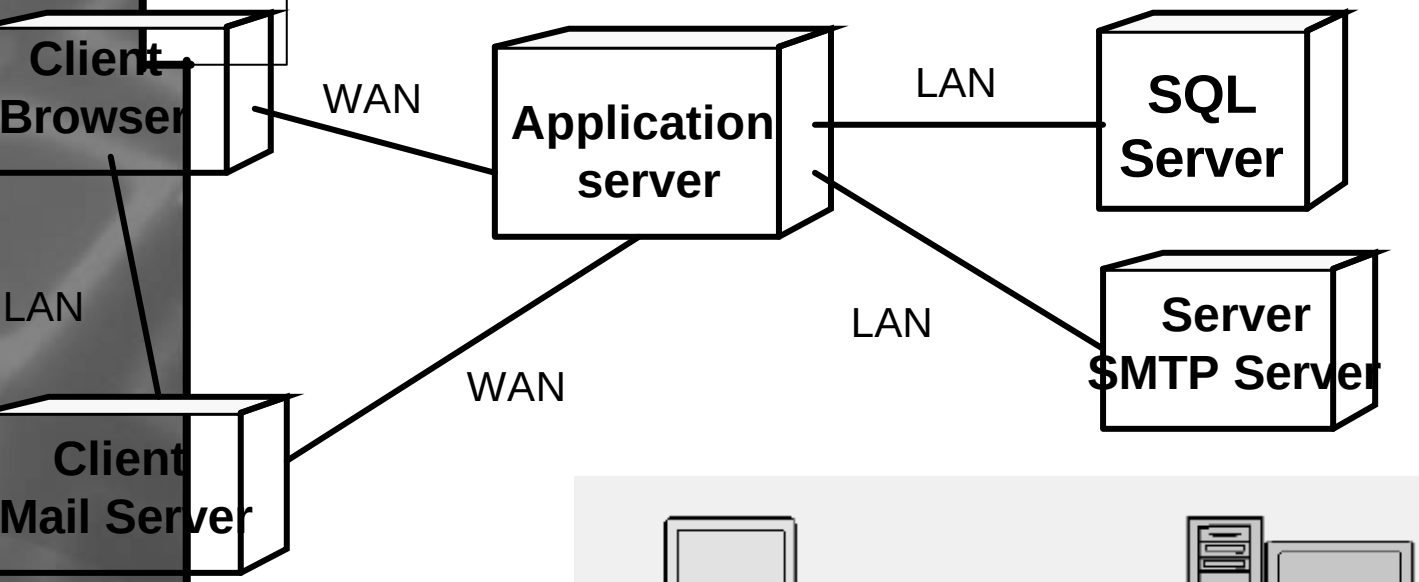
组件设计与VS.NET项目规划

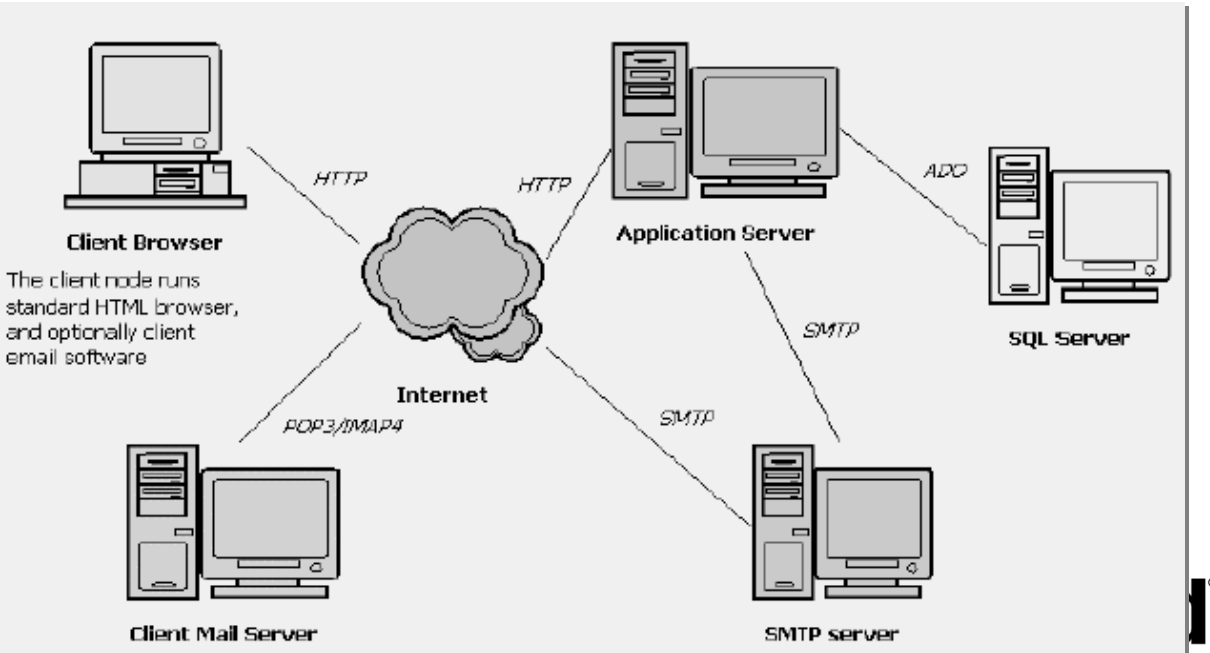




Borland®

Deployment Diagram





What we learn about UML Diagrams ?

Behavior :

- ✓ Use Case
- ✓ Activity
- ✓ Sequence
- ✓ Collaboration
- ☐ State Chart

Structural:

- Class
- Component
- Deployment
- Object

Model Management:

- Packages (class diagram contains packages)
- Subsystems (class diagram contains subsystems)
- Models (class diagram contains models)

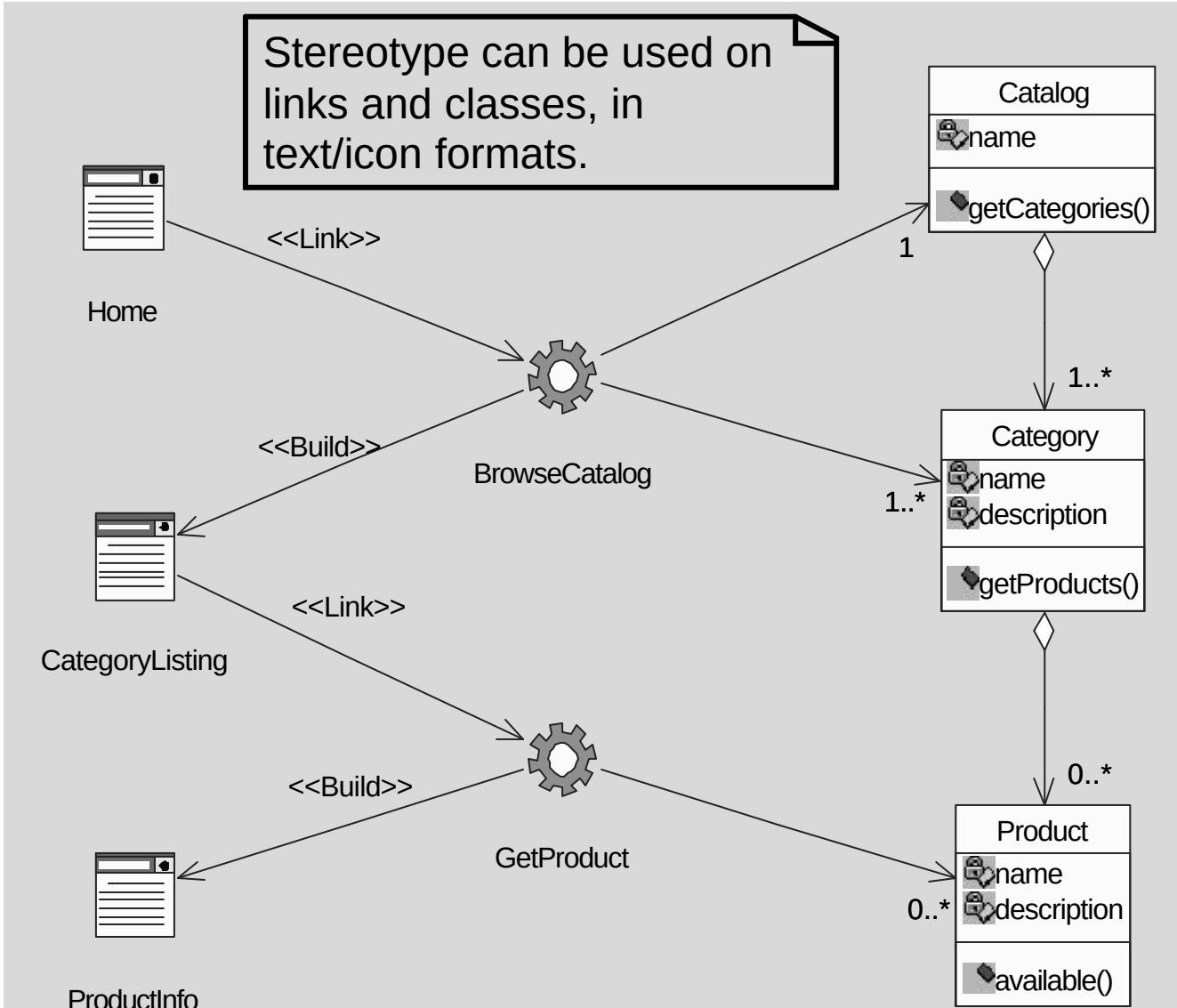
Borland®

扩充UML的描述能力

Stereotype

Note

Tag



Design Pattern

A pattern is a solution to a problem in a context, it documents in an abstract and compact form

- the problem
- the context in which it occurs
- a good solution
- and it embodies wisdom about how the solution addresses the problem.

Different levels

- basic building blocks
- design patterns
- architecture patterns

Borland®

Façade Design Pattern

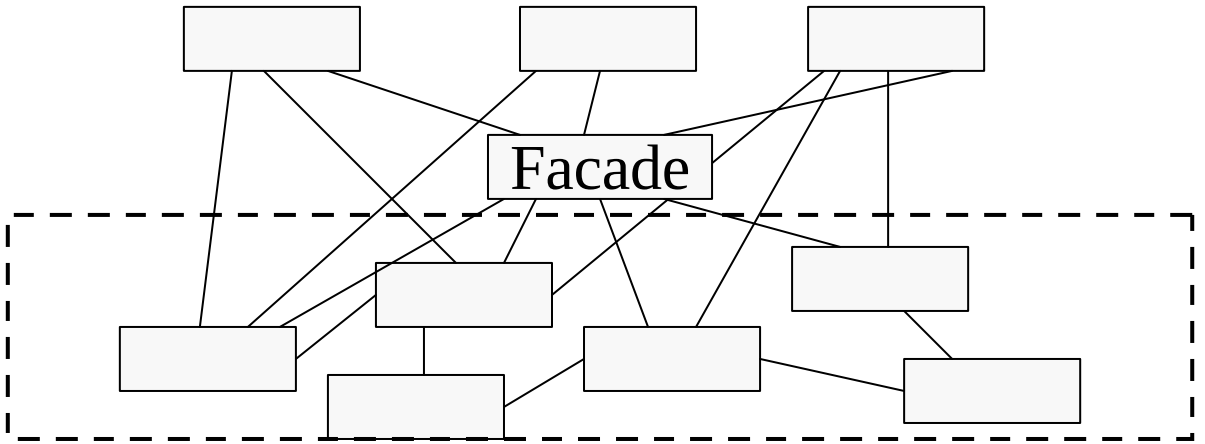
Façade: defines a clean, high-level interface to a subsystem.

Context: building easy-to-use and maintain subsystems

Problem: Each class in the subsystem provides part of the subsystem's functionality, clients has to know the inside, changes to the subsystem may require changes to the clients.

Solution: Add an interface class (the façade class) that knows the structure of the subsystem and forwards requests…

Consequences: no or less dependency of client from structure of subsystem, ideal for layered subsystems



Borland®

What is Model Driven Architecture?

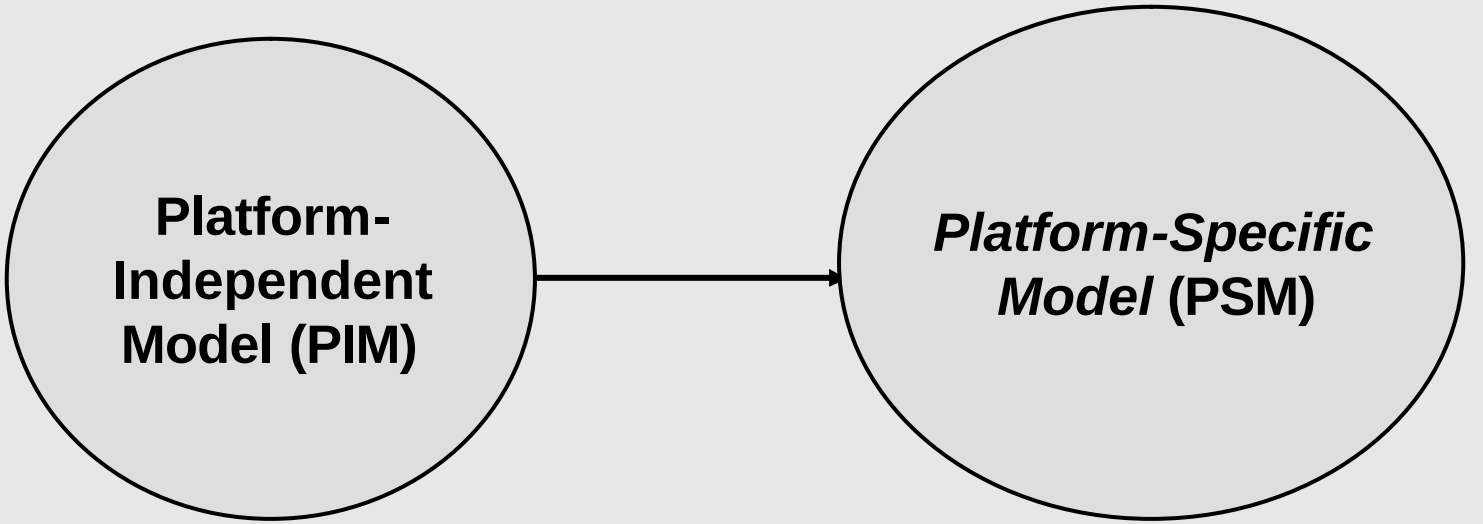
A New Way to Specify and Build Systems

- **2001年由 OMG 制定的新开发架构**
- **以 UML 塑模为基础**
- **支持完整开发周期** : analysis, design, implementation, deployment, maintenance, evolution & integration with later systems
- **内建协同运作性及跨平台性**
- **降低开发初期成本及提高** ROI

- **可套用至你所使用的任何环境** :
  - Programming language
  - Operating system
  - Network
  - Middleware

Borland®

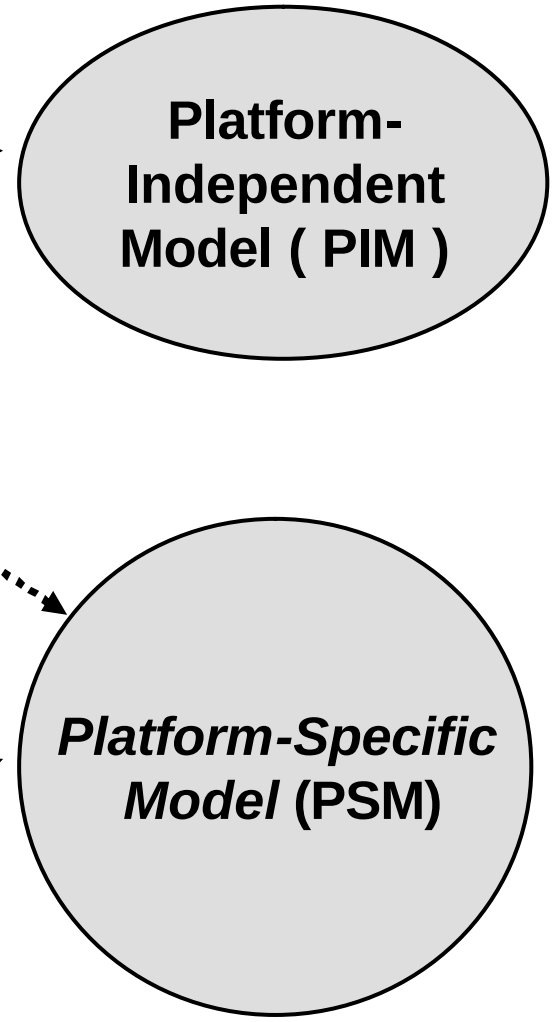
Model Driven Architecture



MDA tool applies an standard mapping to generate *Platform-Specific Model* (PSM) from the PIM. Code is partially automatic, partially hand-written.

开发流程

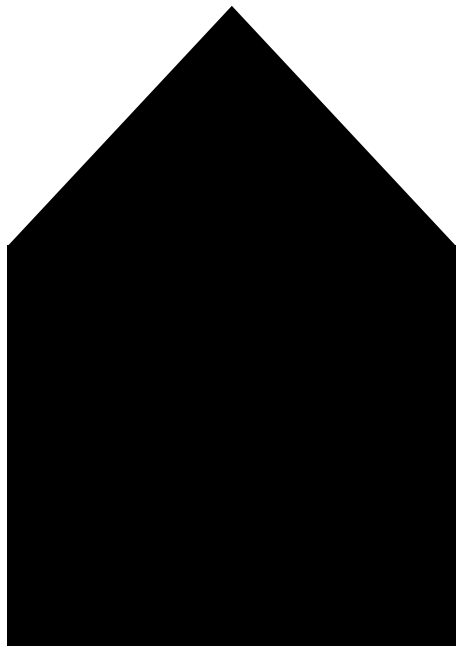
- Analysis Model
- Design Model
  - Architecture design
  - Data Model
  - User Experience Model
    - look-Feel & Interaction
- Implementation Model

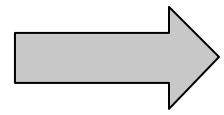


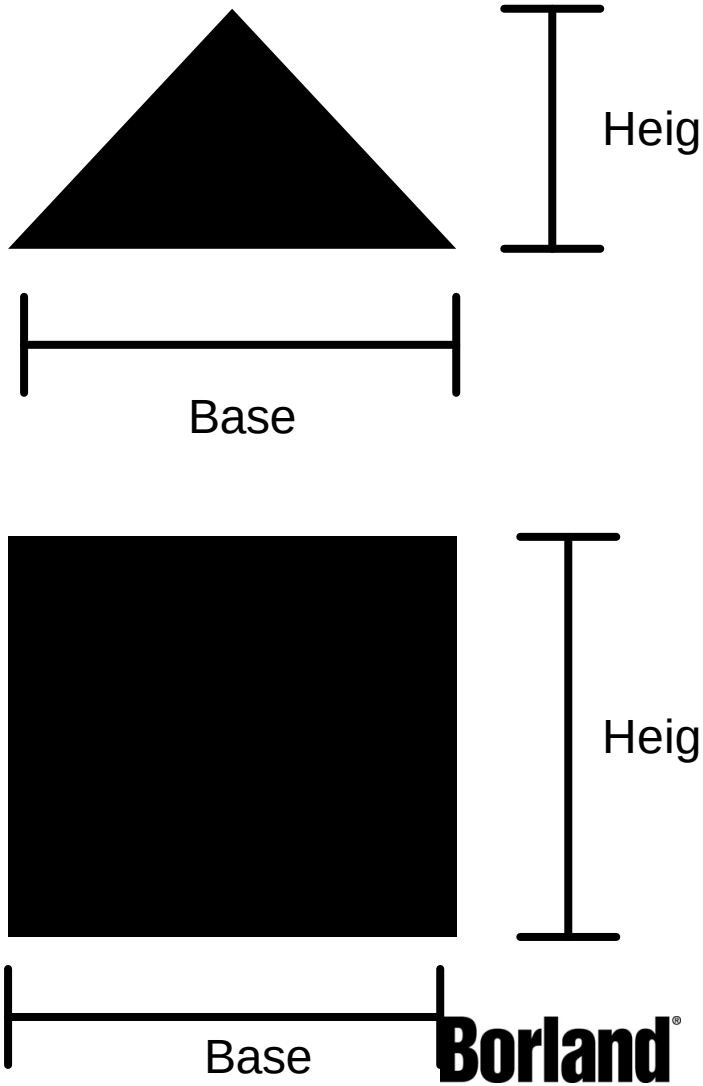
Borland®

展示实例

面积计算

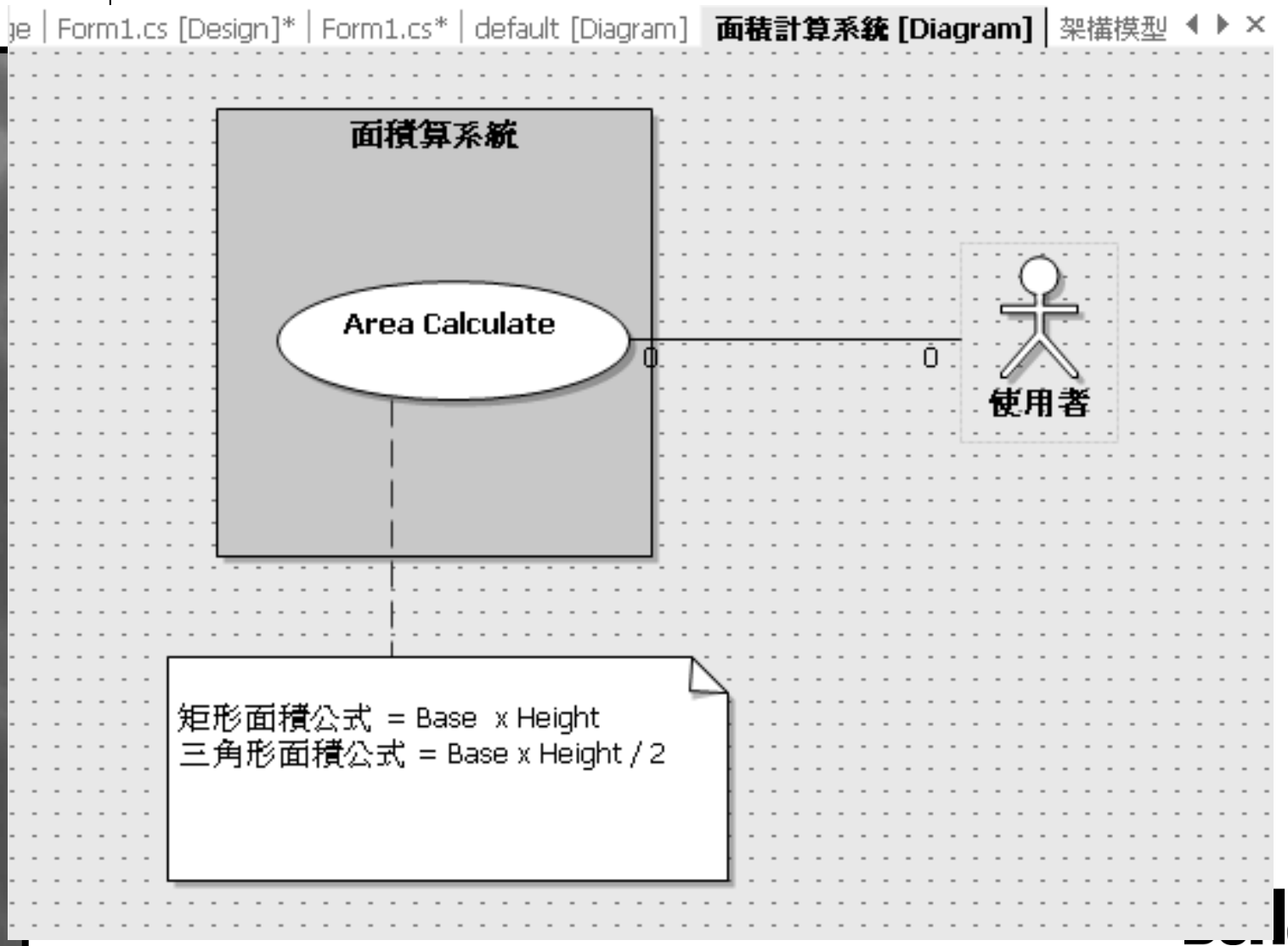




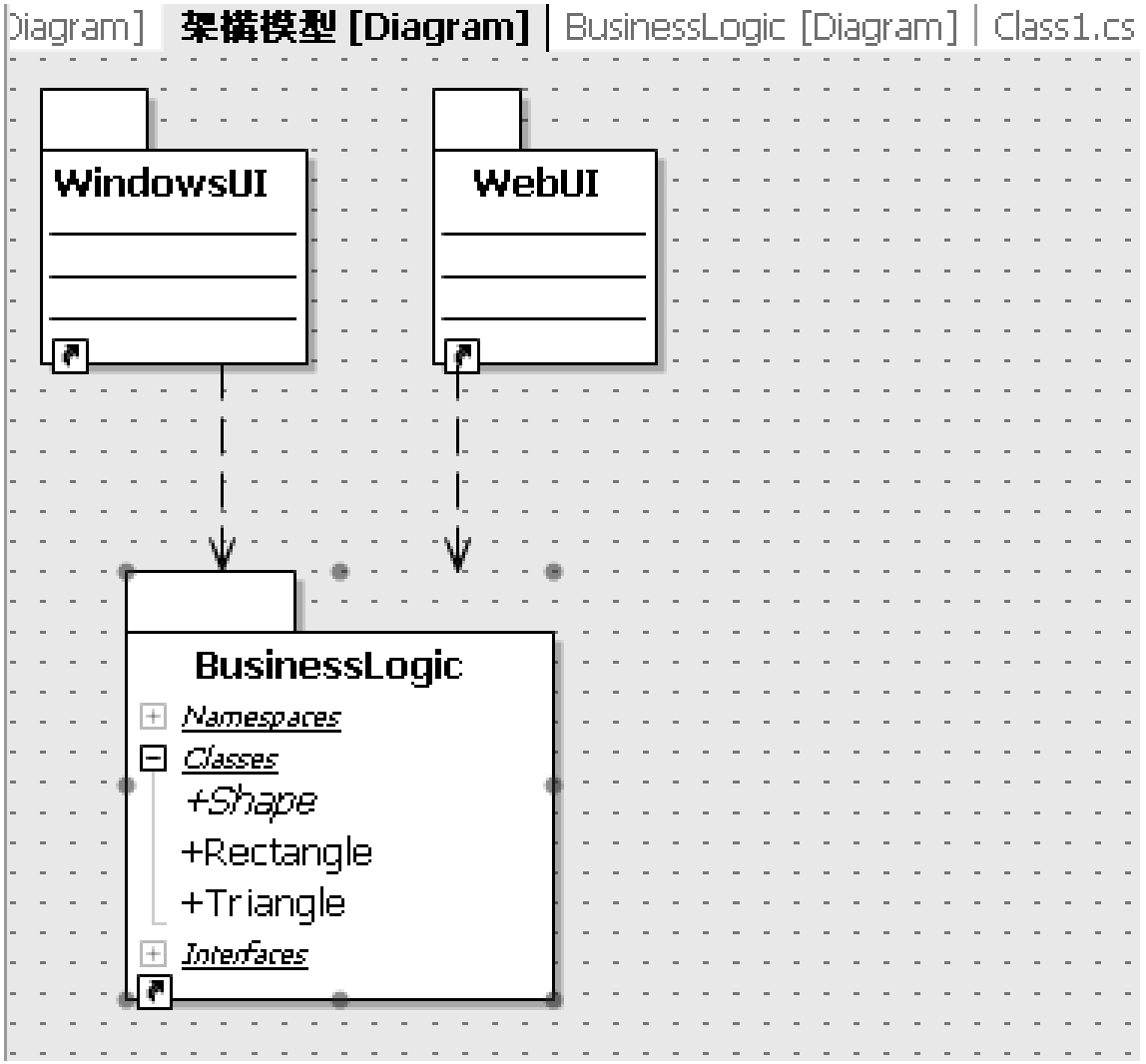


Borland®

Use Case Model

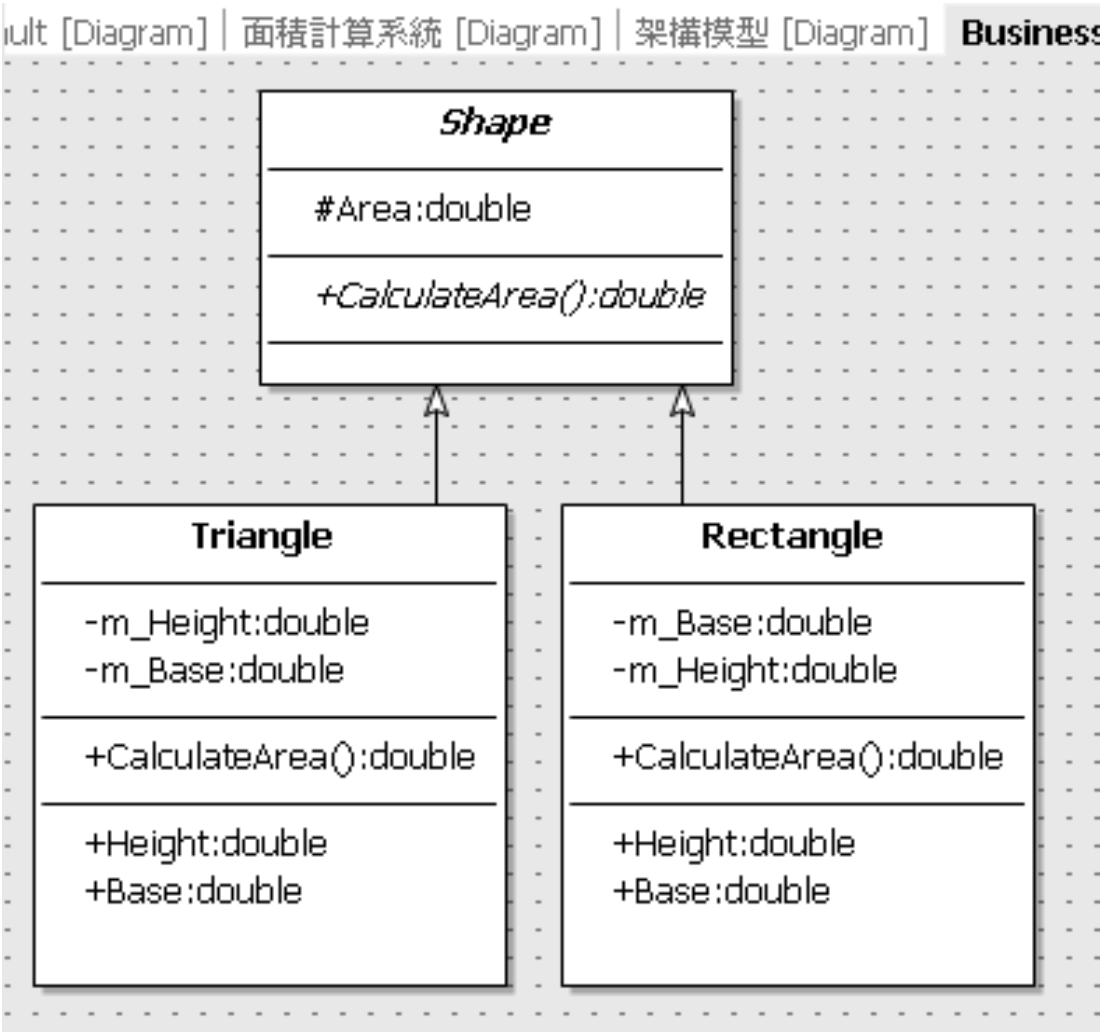


架构设计



Borland®

系统设计



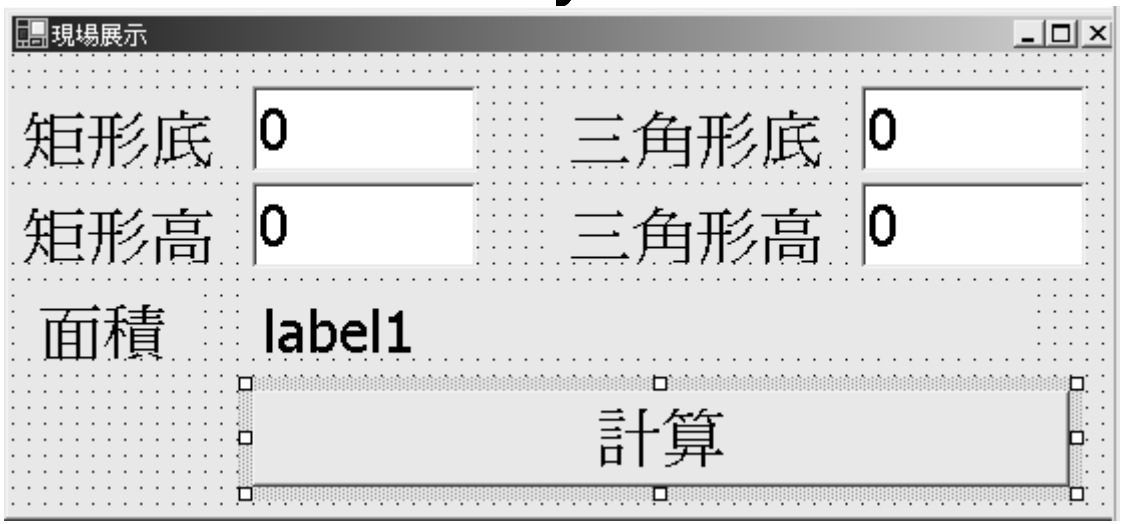
Borland®

Business Logic

```
namespace BusinessLogic
{
  public class Triangle: Shape
{
      public override double CalculateArea()
      {
         this.Area = this.m_Base  * this.m_Height / 2;
         return (this.Area);
      }
     private double m_Height;
     private double m_Base;
     public double Height {
       get {
          return (m_Height);
       }
       set {
          m_Height = value;
       }
```

Borland®

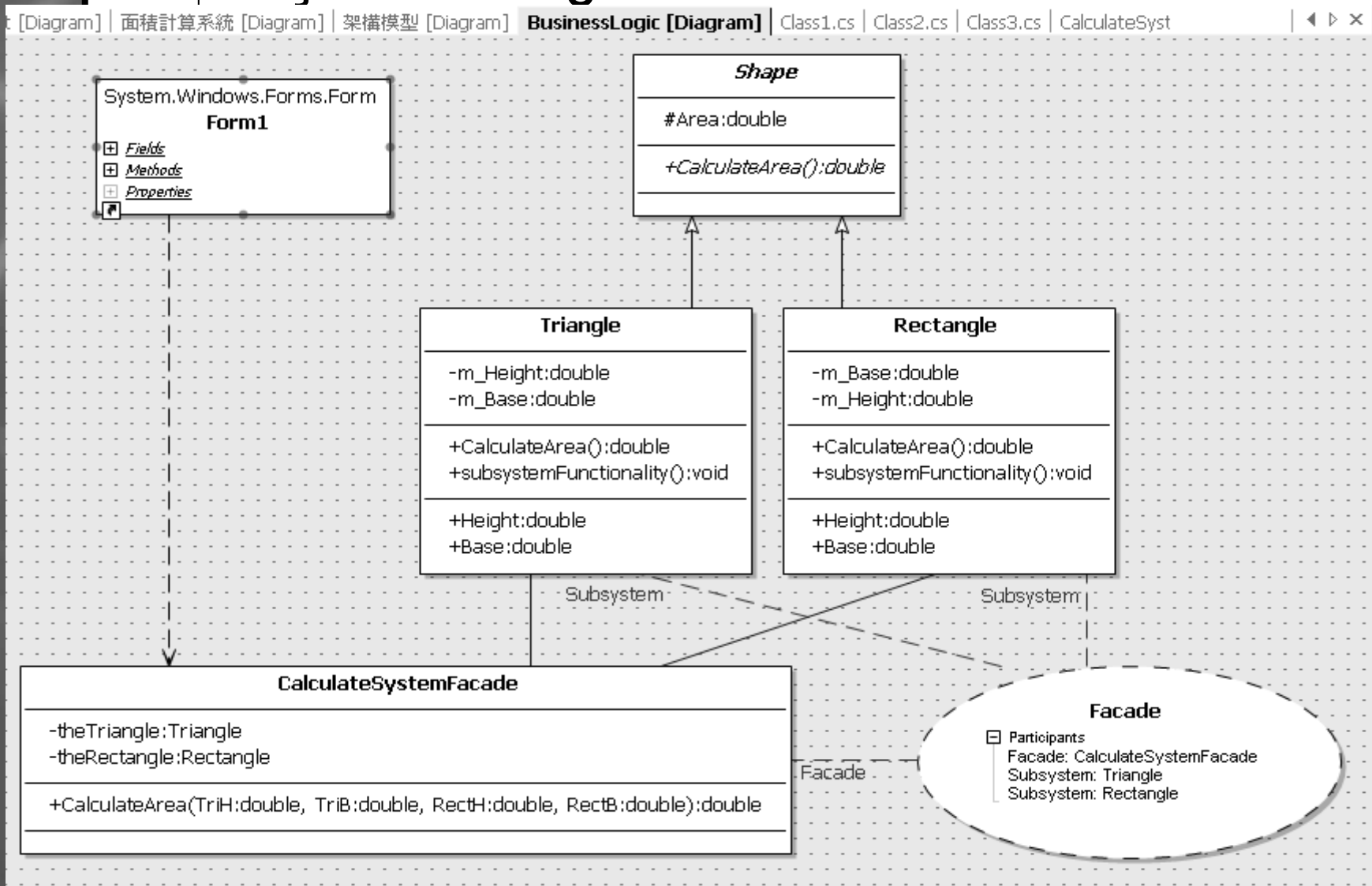
Presentation Layer



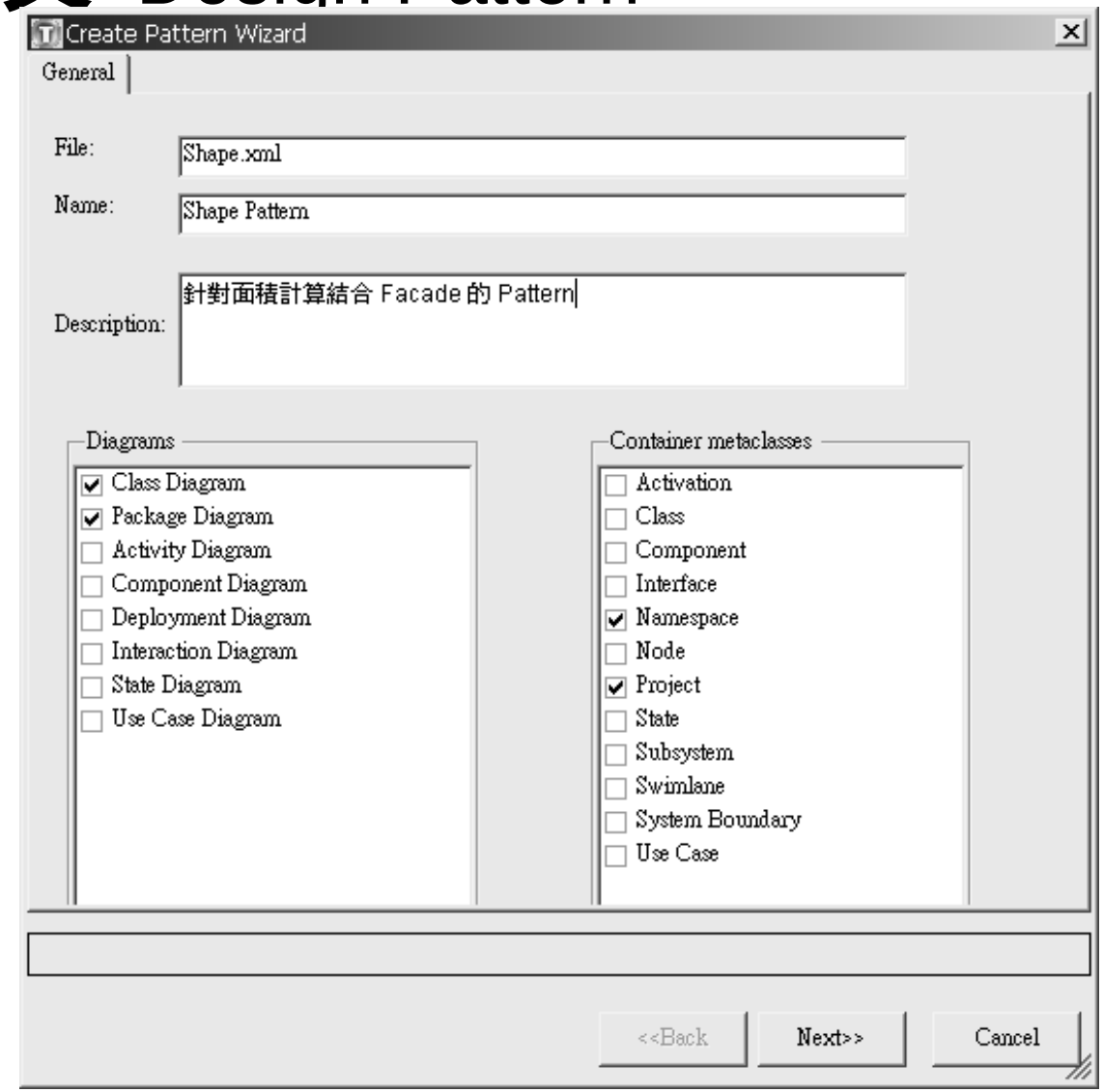

```
theRectangle.Base = RectB;
theRectangle.Height = RectH;
theTriangle.Base = TriB;
theTriangle.Height = TriH;
result = theTriangle.CalculateArea()+ theRectangle.CalculateArea();
```

Borland®

Façade Design Pattern

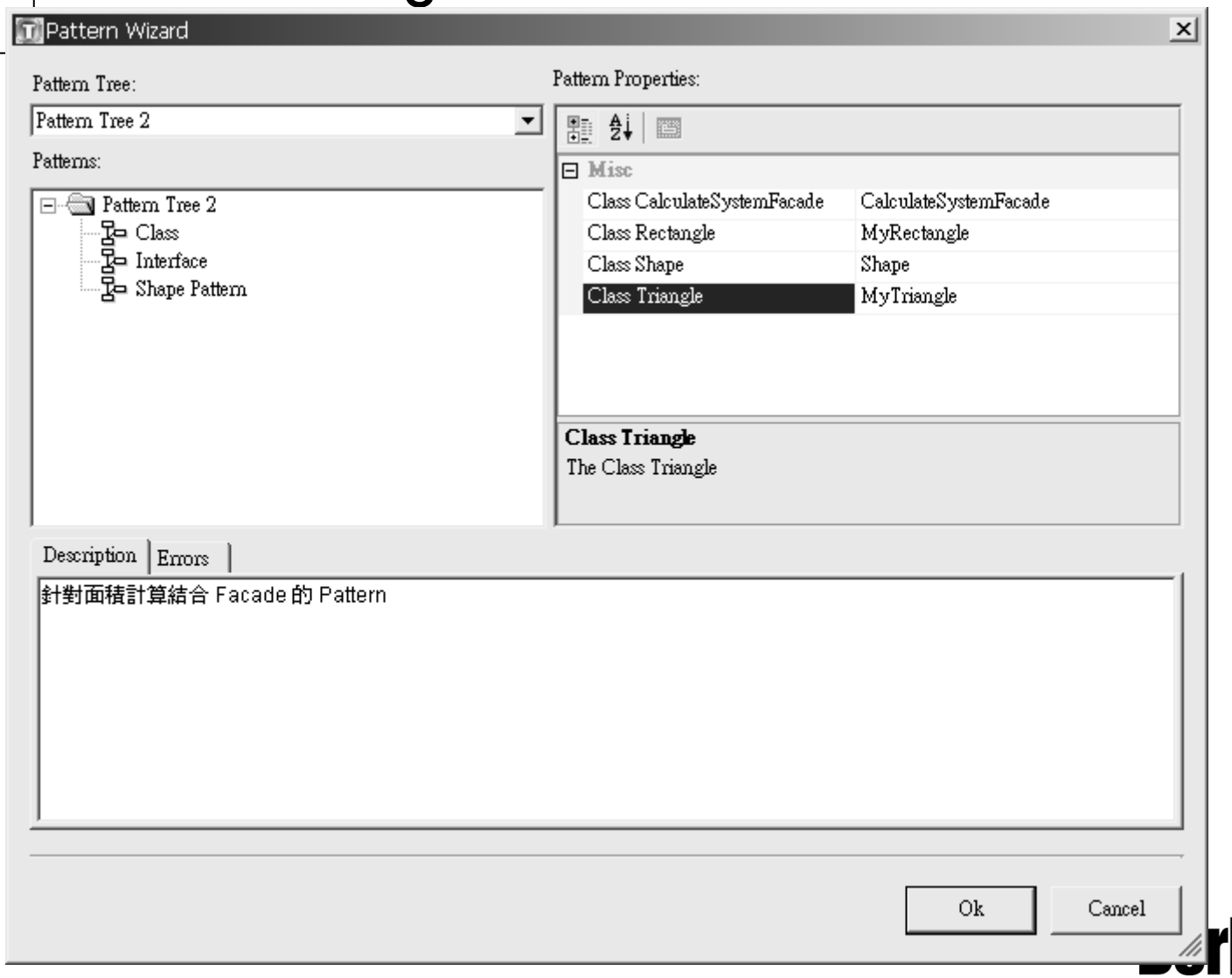


定义 Design Pattern



Borland®

Reuse Design Pattern



Together Edition
for Microsoft Visual Studio .NET

UML Diagram **绘制功能**

LiveSource™ **技术实时同步** Model **与程序代码**

深度整合 Microsoft Visual Studio .NET **自动化文件产生**

支援 Pattern

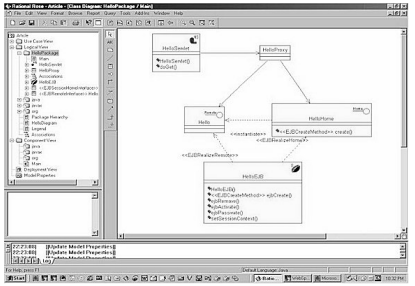
支持 XMI **汇入与汇出**

Borland®

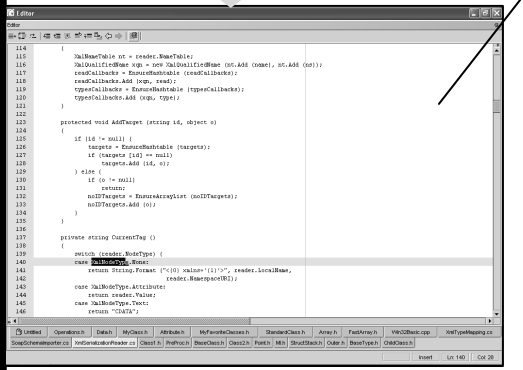
LiveSource™技术

Reverse/Forward Engineer

IBM/Rational

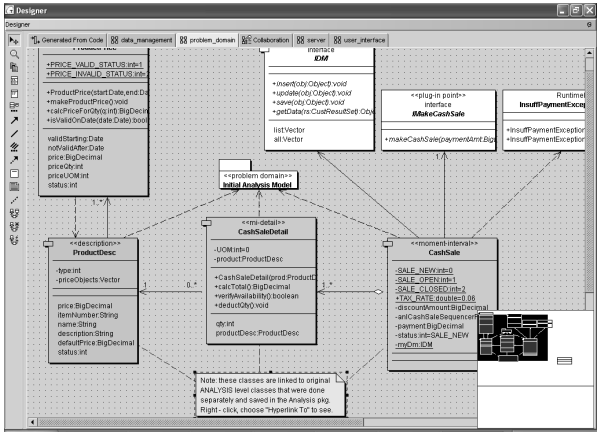


Binary Repository

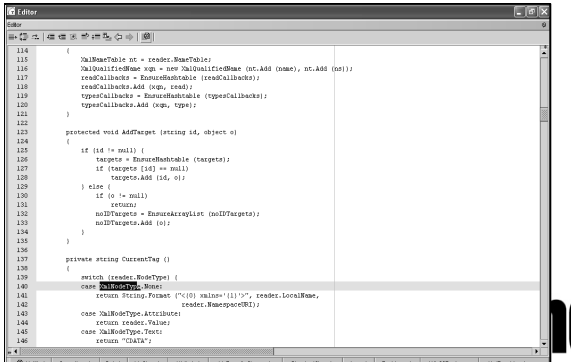


? ? ?

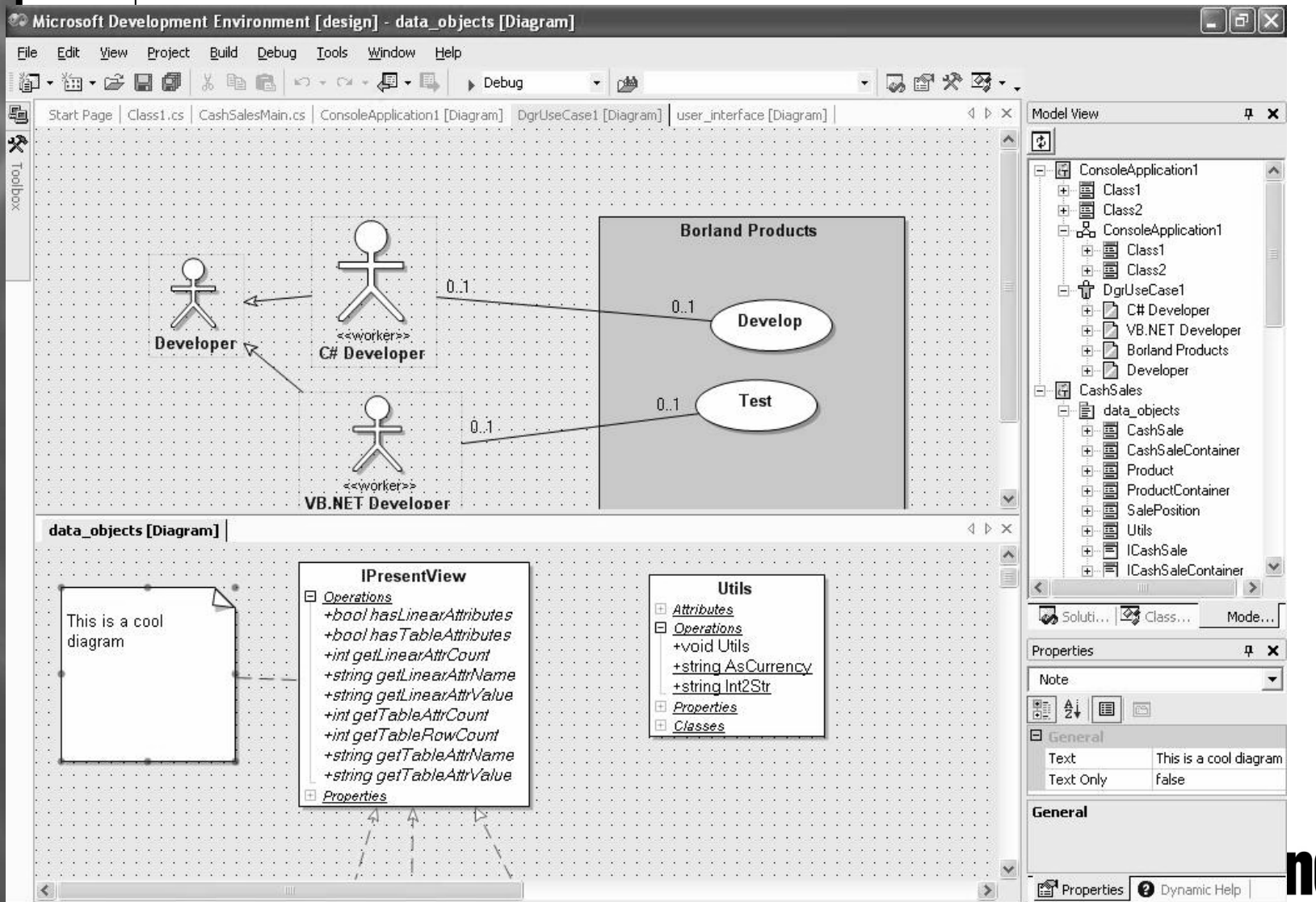
Borland Together



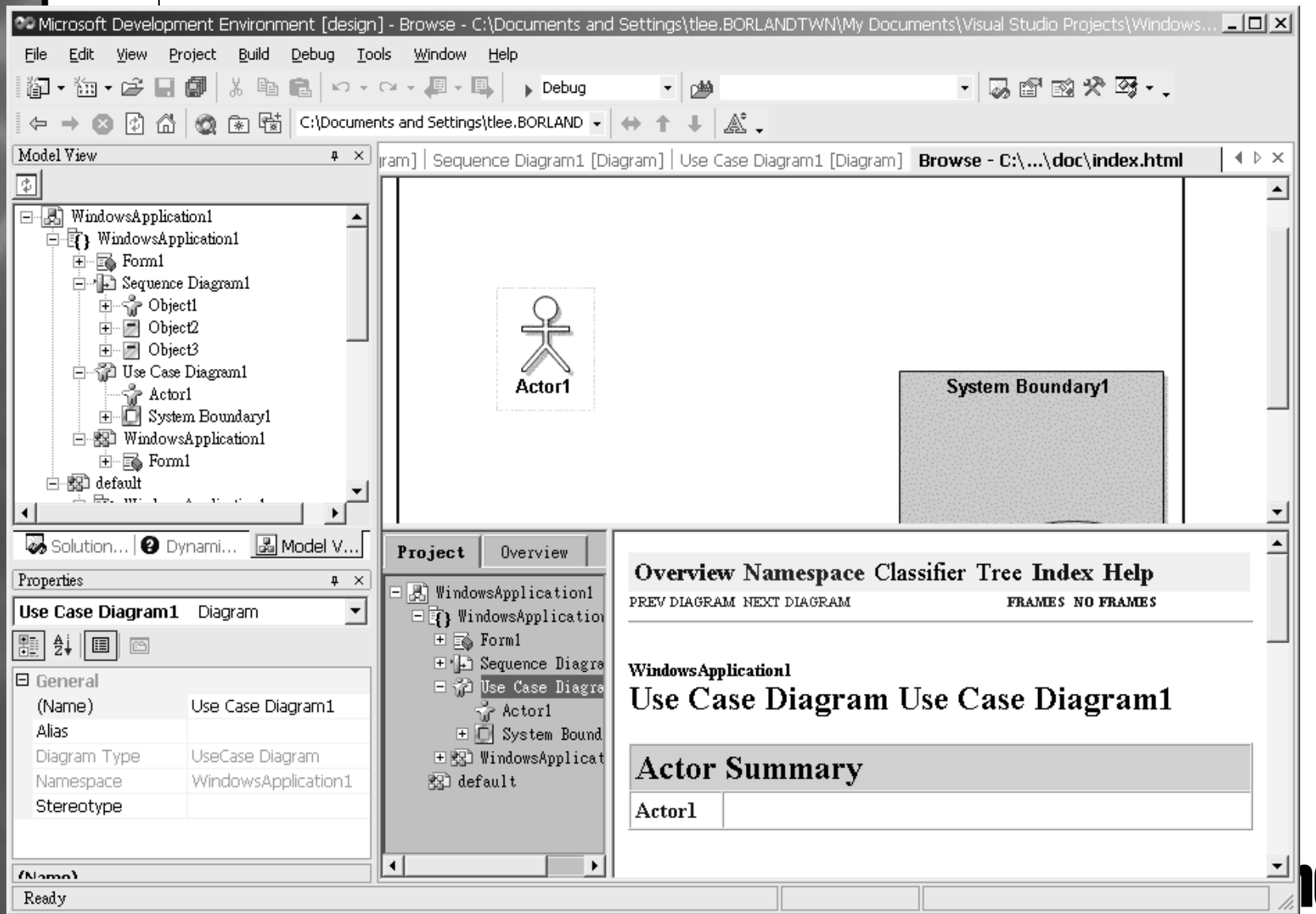
? ? ? ?



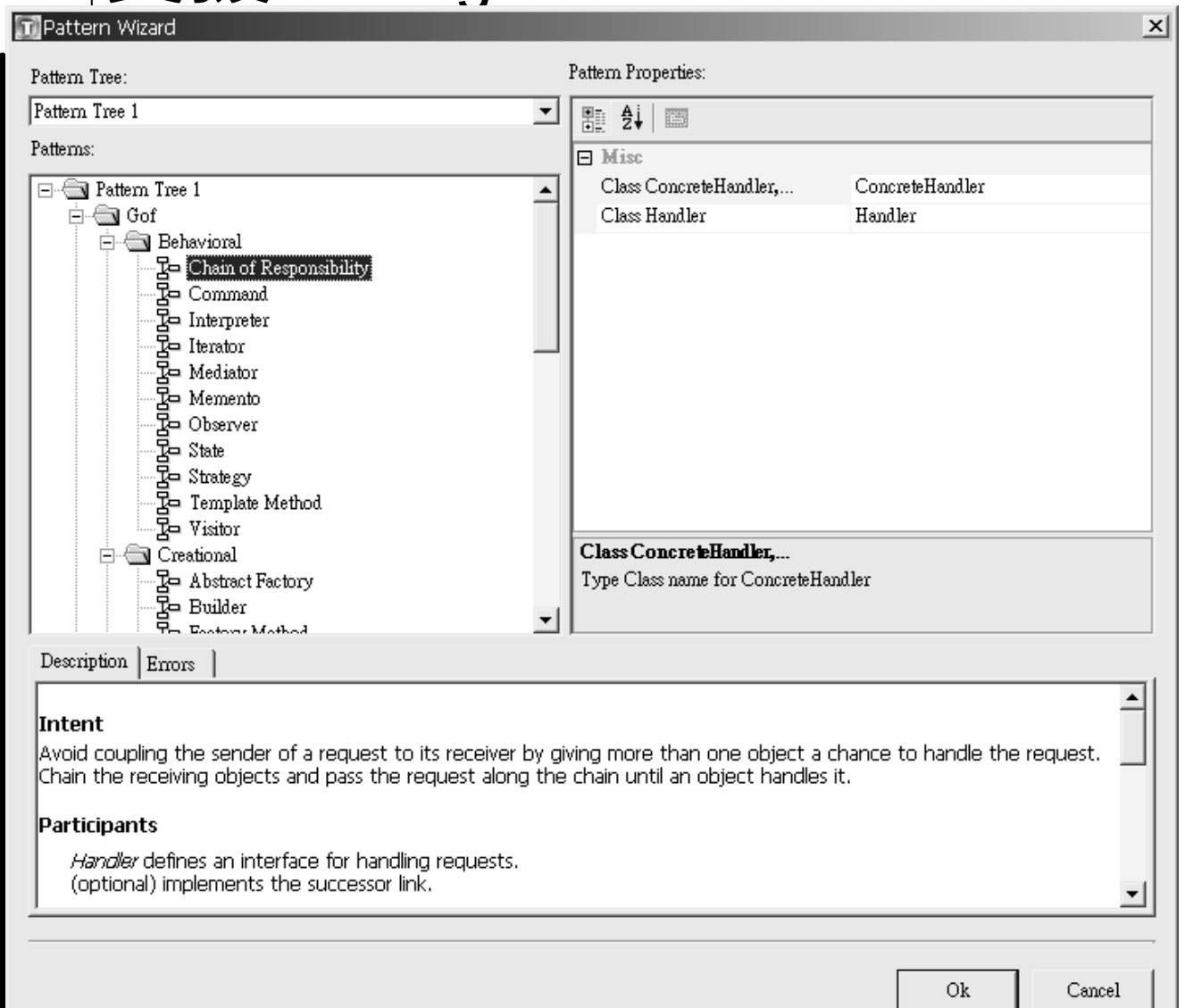
深度整合 Microsoft Visual Studio .NET



自动化文件产生


Microsoft Development Environment [design] - Browse - C:\Documents and Settings\tlee.BORLANDTWN\My Documents\Visual Studio Projects\Windows...
File Edit View Project Build Debug Tools Window Help
Debug
C:\Documents and Settings\tlee.BORLAND
Model View
WindowsApplication1
WindowsApplication1
Form1
Sequence Diagram1
Object1
Object2
Object3
Use Case Diagram1
Actor1
System Boundary1
WindowsApplication1
Form1
default
Solution... Dynami... Model V...
Properties
Use Case Diagram1 Diagram
General
(Name) Use Case Diagram1
Alias
Diagram Type UseCase Diagram
Namespace WindowsApplication1
Stereotype
Sequence Diagram1 [Diagram]
Use Case Diagram1 [Diagram]
Browse - C:\...\doc\index.html
Actor1
System Boundary1
Project Overview
WindowsApplication1
Form1
Sequence Diagra
Use Case Diagra
Actor1
System Bound
WindowsApplicat
default
Overview Namespace Classifier Tree Index Help
PREV DIAGRAM NEXT DIAGRAM
FRAMES NO FRAMES
WindowsApplication1
Use Case Diagram Use Case Diagram1
Actor Summary
Actor1
Ready

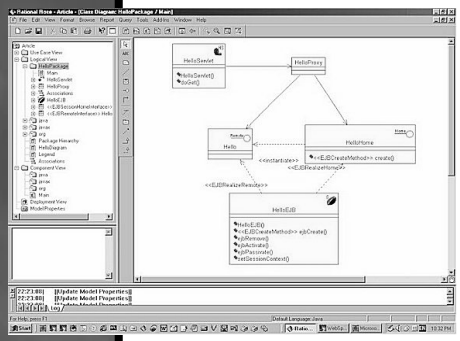
支援 Design Pattern



Borland®

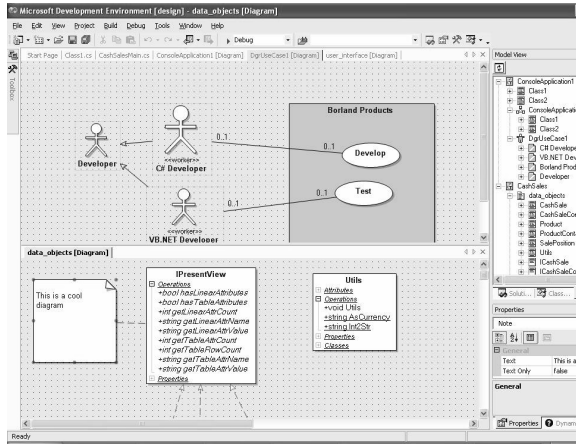
支持 XMI 汇入与汇出

IBM/Rational



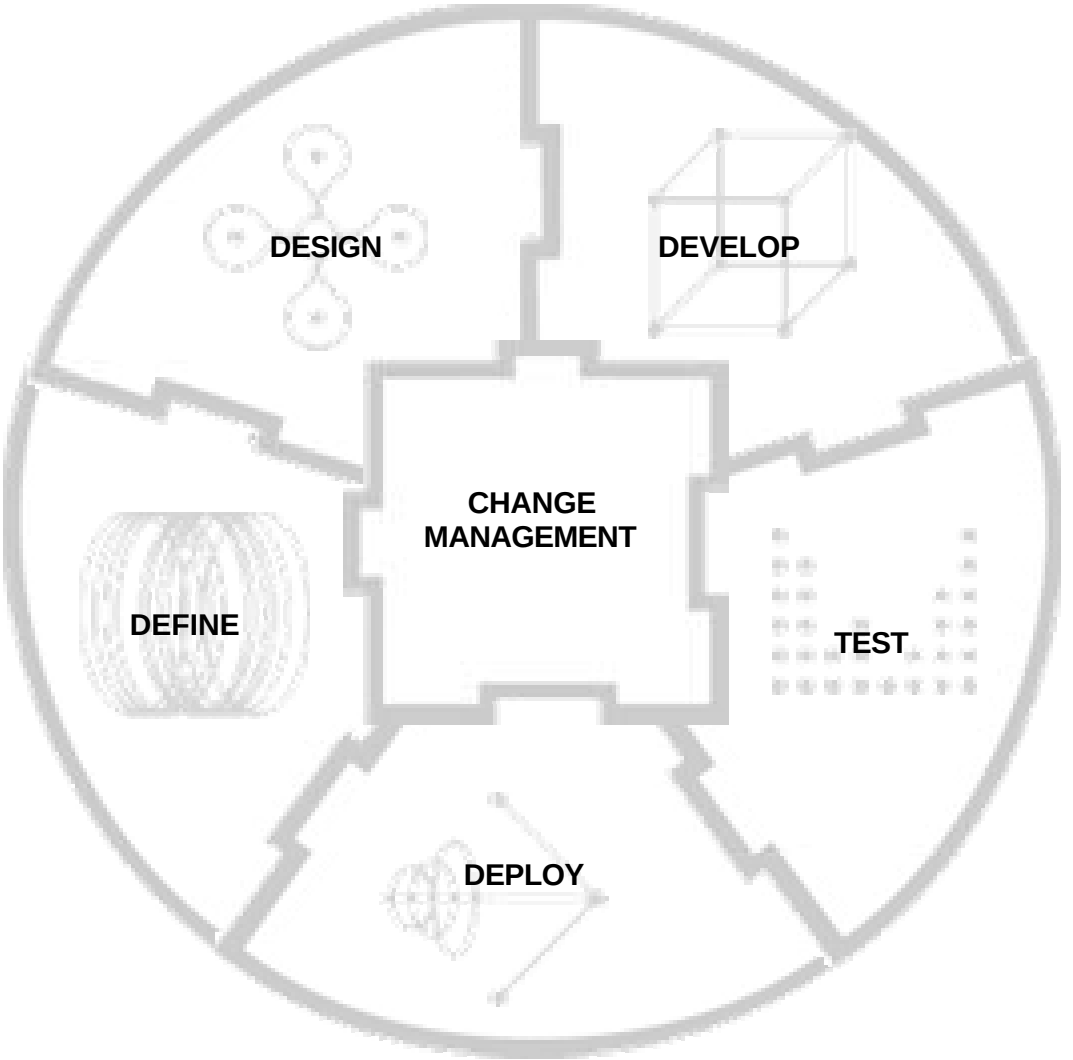
XMI？？？？

Borland Together



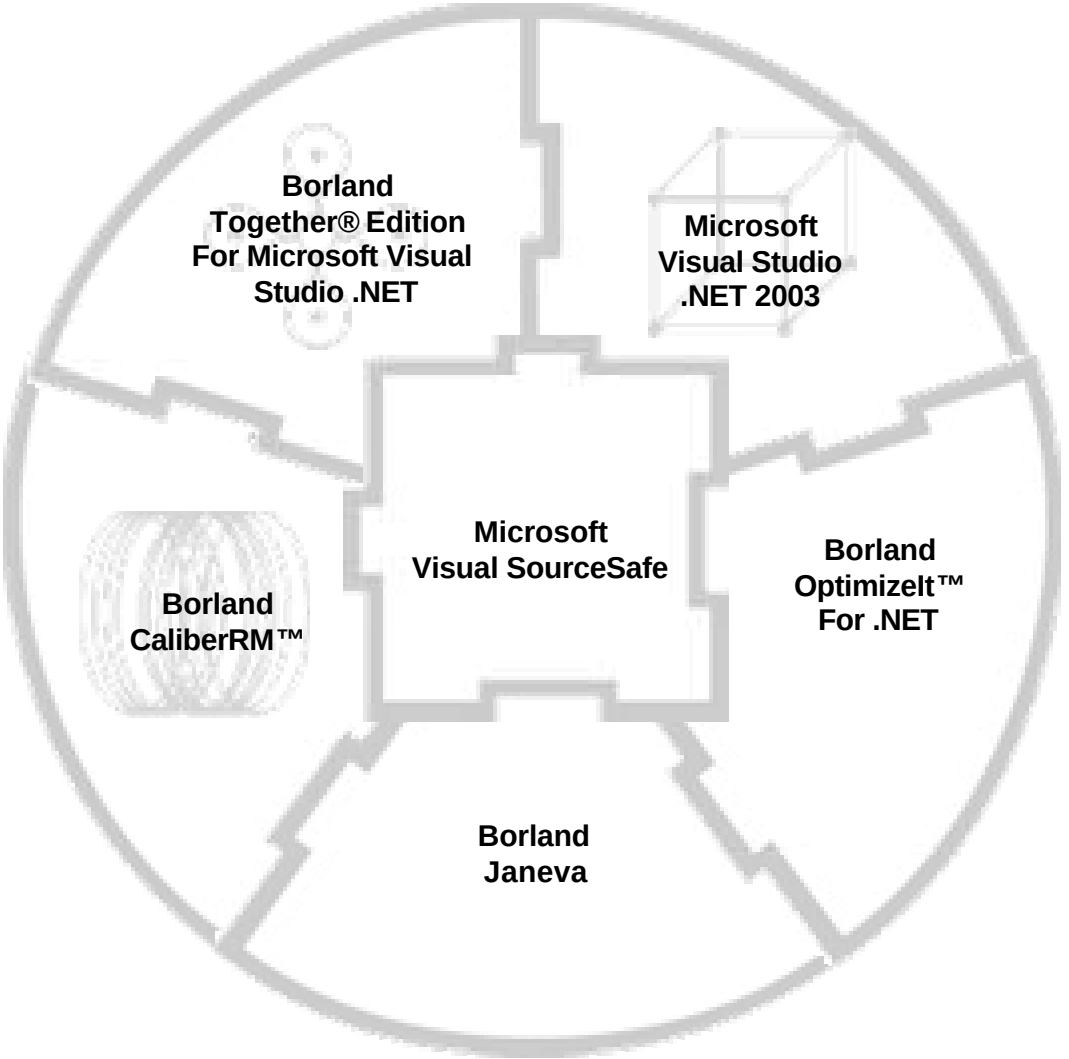
Borland®

Borland ALM ? ? ? ? - .NET



Borland®

Borland ALM ? ? ? ? - .NET



Borland®

总结

Learn UML

Follow a process

Try a modeling tools

Find supports

- Time,
- Budget,
- Mentoring

May The Power

Be With You !!

Resources

需搭配 Microsoft Visual Studio .NET 2003

试用版下载网址

http://www.borland.com/products/downloads/download_together.html

Borland®

Q&A

Borland®