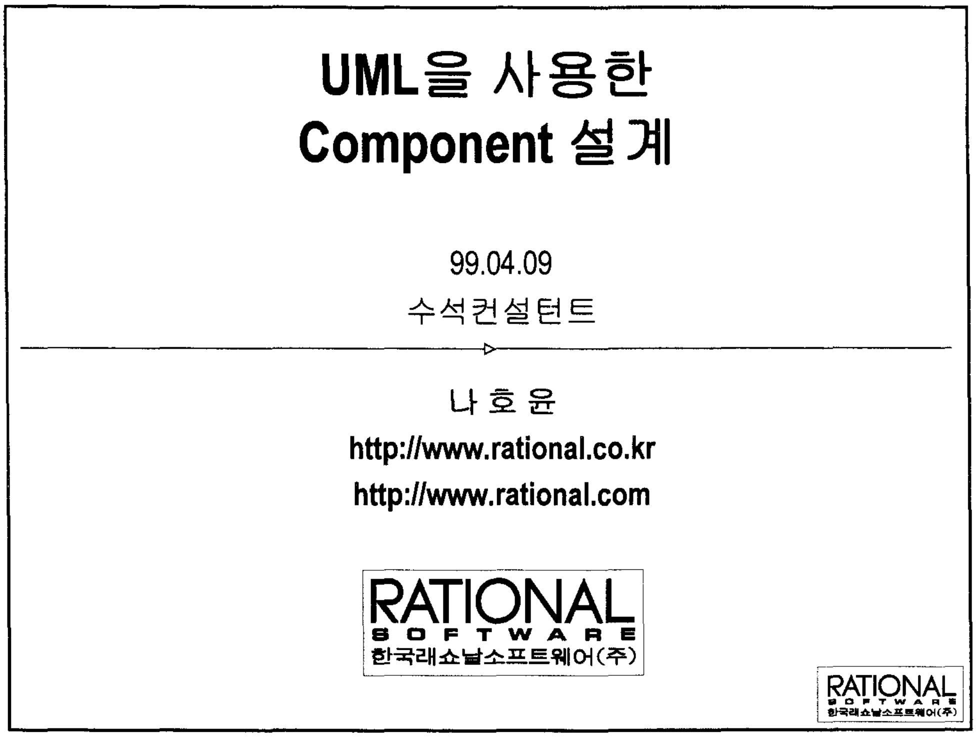
UML을 사용한 Component 설계

❷

1999. 4. 9

나 호 윤
Rational Software Corp.



UML을 사용한 Component 설계

99.04.09

수석컨설턴트

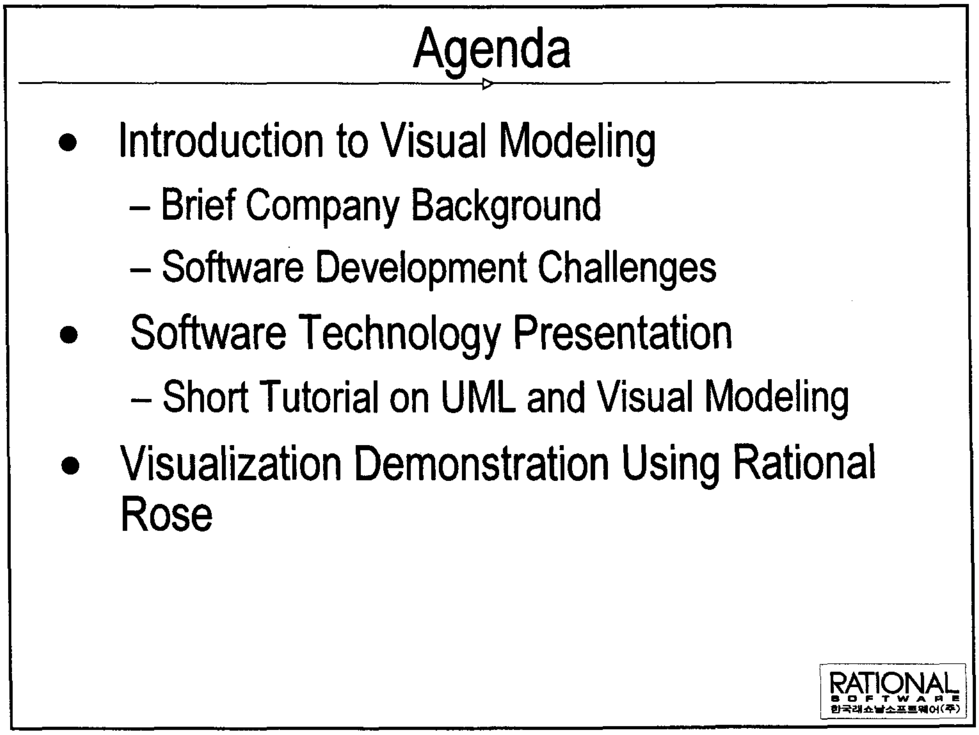
나호윤

http://www.rational.co.kr

http://www.rational.com

RATIONAL SOFTWARE

한국래쇼날소프트웨어(주)



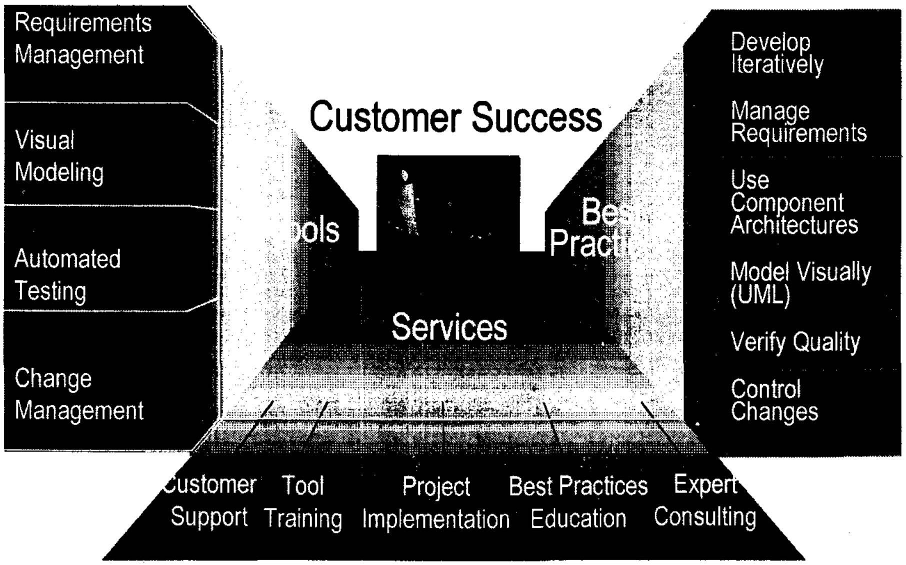
Agenda

- Introduction to Visual Modeling
  - Brief Company Background
  - Software Development Challenges
- Software Technology Presentation
  - Short Tutorial on UML and Visual Modeling
- Visualization Demonstration Using Rational Rose

Rational Software's Mission

To ensure the success of customers who depend on developing or deploying software

Rational's Unique Approach



Requirements Management

Visual Modeling

Automated Testing

Change Management

Develop Iteratively

Manage Requirements

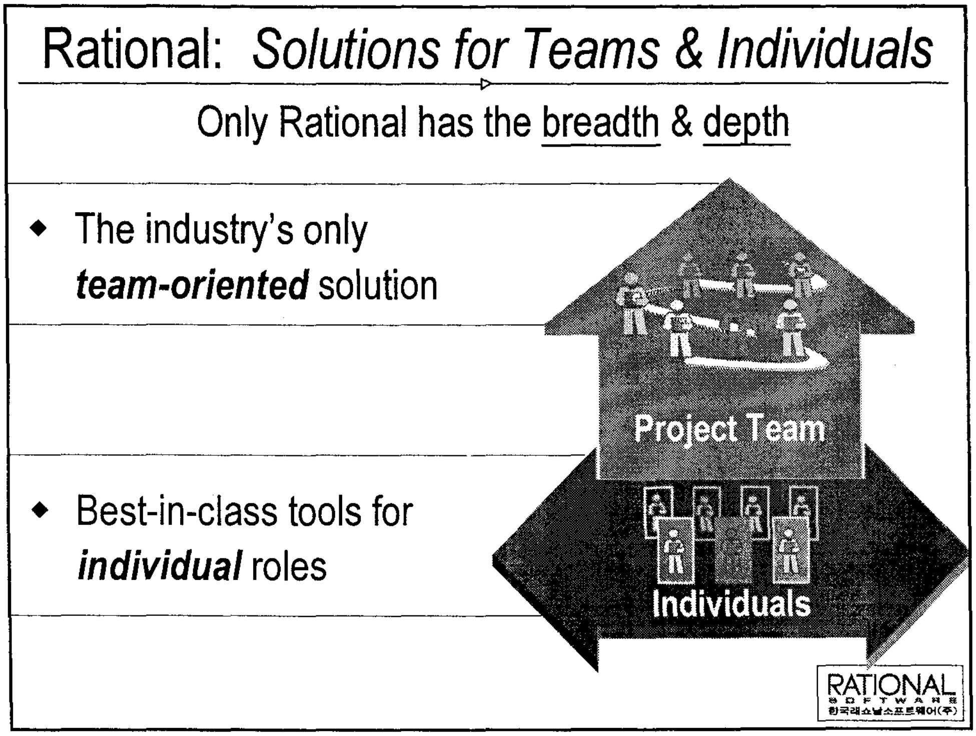
Use Component Architectures

Model Visually (UML)

Verify Quality

Control Changes

Customer Support | Tool Training | Project Implementation | Best Practices Education | Expert Consulting



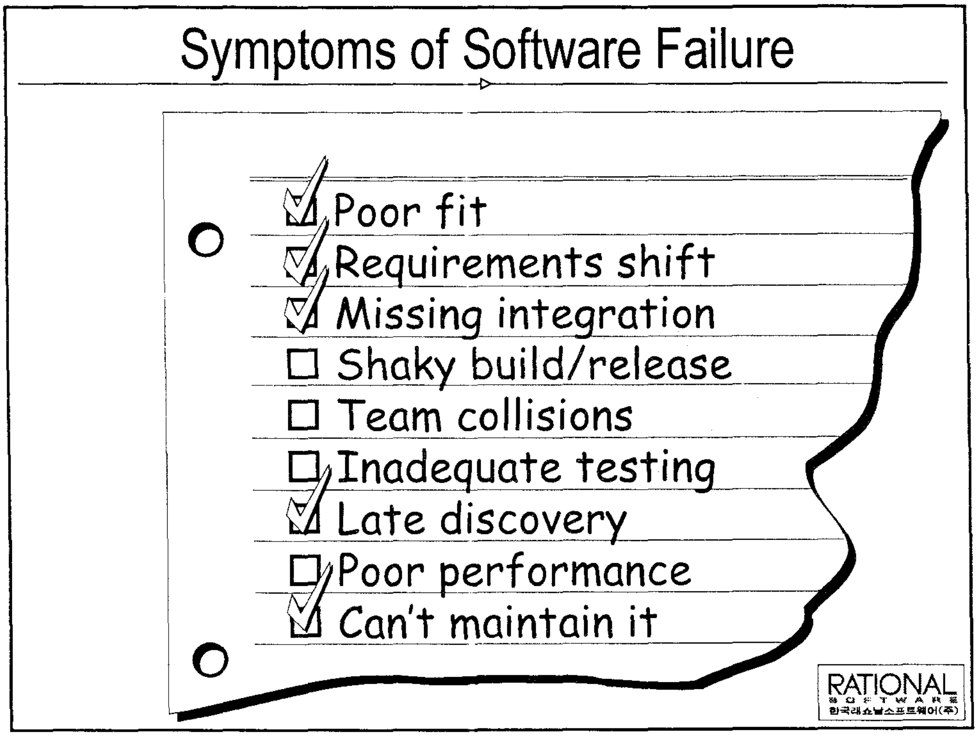
Rational: *Solutions for Teams & Individuals*

Only Rational has the breadth & depth

- The industry's only ***team-oriented*** solution
- Best-in-class tools for ***individual*** roles

Project Team
Individuals

RATIONAL SOFTWARE
한국래쇼날소프트웨어(주)



Symptoms of Software Failure

☑ Poor fit
☑ Requirements shift
☑ Missing integration
□ Shaky build/release
□ Team collisions
□ Inadequate testing
☑ Late discovery
□ Poor performance
☑ Can't maintain it

RATIONAL SOFTWARE
한국래쇼날소프트웨어(주)

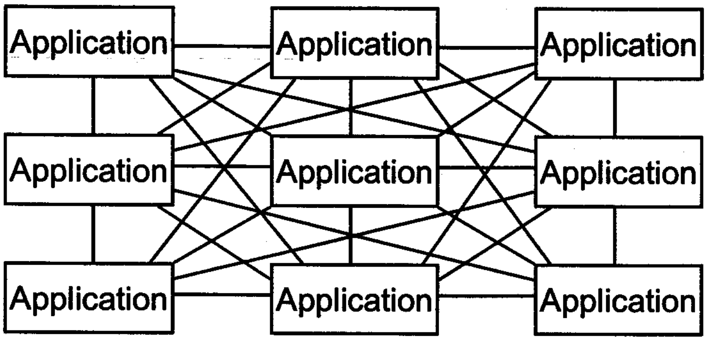
Rational: *Solutions for Teams & Individuals*

This presentation will discuss how we can help you develop quality products that meet your customers real needs.

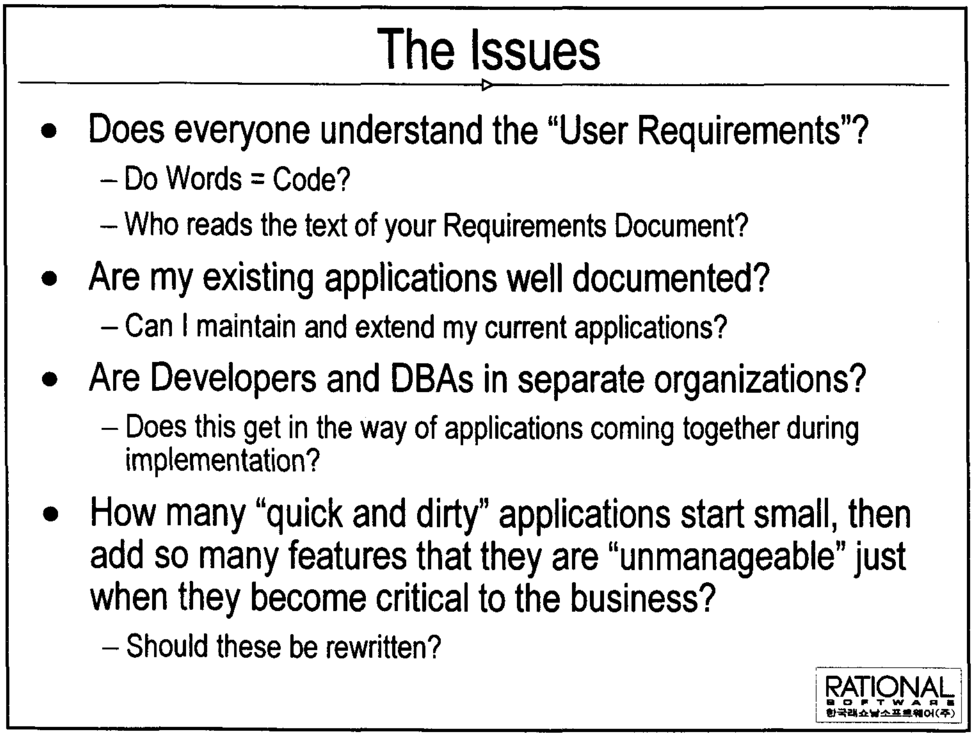
RATIONAL SOFTWARE 한국래쇼날소프트웨어(주)

The “Problem”

Current State: Unmanageable application interdependencies which are poorly documented

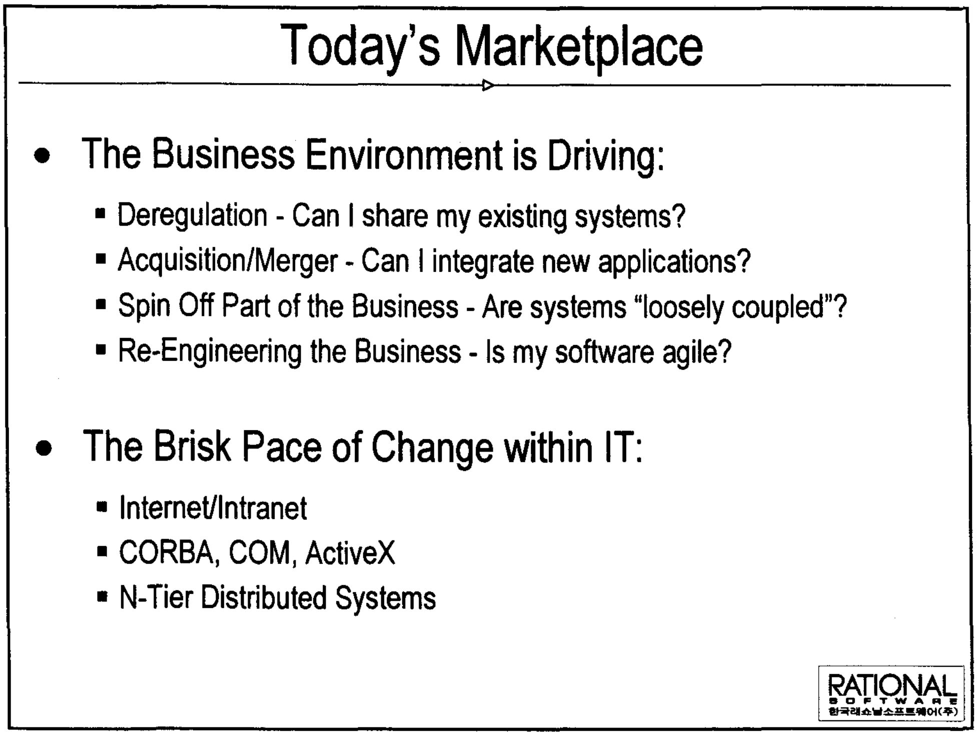


RATIONAL SOFTWARE 한국래쇼날소프트웨어(주)



The Issues

- Does everyone understand the “User Requirements”?
  - Do Words = Code?
  - Who reads the text of your Requirements Document?
- Are my existing applications well documented?
  - Can I maintain and extend my current applications?
- Are Developers and DBAs in separate organizations?
  - Does this get in the way of applications coming together during implementation?
- How many “quick and dirty” applications start small, then add so many features that they are “unmanageable” just when they become critical to the business?
  - Should these be rewritten?



Today’s Marketplace

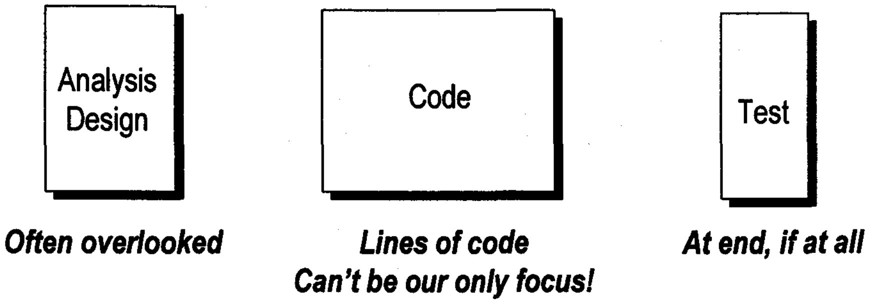
- The Business Environment is Driving:
  - ▪ Deregulation - Can I share my existing systems?
  - ▪ Acquisition/Merger - Can I integrate new applications?
  - ▪ Spin Off Part of the Business - Are systems “loosely coupled”?
  - ▪ Re-Engineering the Business - Is my software agile?
- The Brisk Pace of Change within IT:
  - ▪ Internet/Intranet
  - ▪ CORBA, COM, ActiveX
  - ▪ N-Tier Distributed Systems

IT Reality

We are forced into an environment of:

High Risk

Sacrifice Quality and Maintainability



Often overlooked

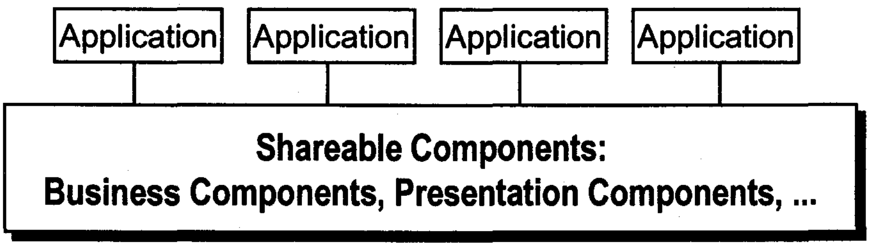
Lines of code
Can't be our only focus!

At end, if at all

The "Solution"

Visualization and Components

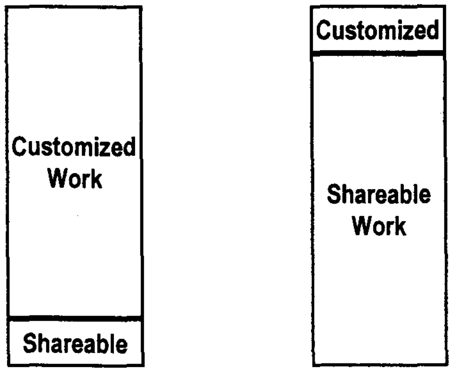
Desired State: No unmanageable interdependencies;
Well Documented and Maintainable



Shareable Components:
Business Components, Presentation Components, ...

The Drive Towards Less Customized Code

Application Infrastructure



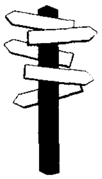
Current State

Desired State

RATIONAL SOFTWARE
한국래쇼날소프트웨어(주)

But, Getting to the Solution.......

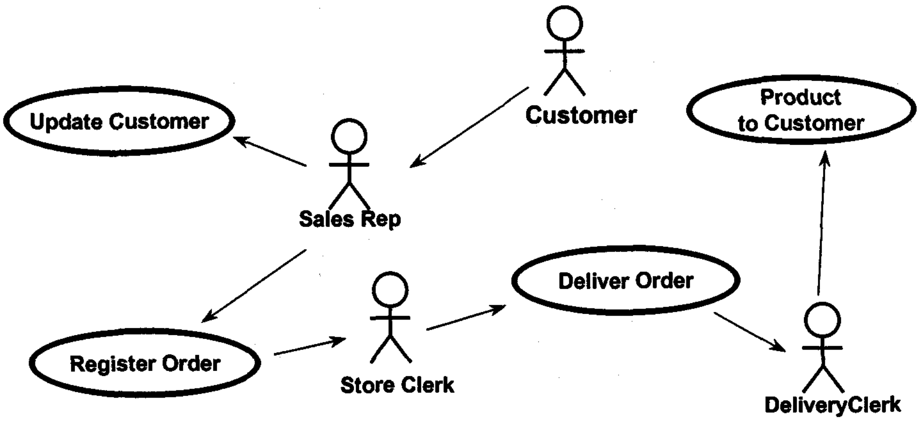
- Cannot get there overnight
  - Requires Process and Architecture
  - Requires Common Toolset to Automate Process and Insure Architecture Compliance
  - Visualization of Components
  - Cultural and Organizational Change



- Yet, we cannot afford to "Not Start"
  - There is a way to begin
  - Visualization is an excellent starting point

RATIONAL SOFTWARE
한국래쇼날소프트웨어(주)

Visualization - Everyone sees the big picture



Update Customer

Customer

Product to Customer

Sales Rep

Deliver Order

Register Order

Store Clerk

DeliveryClerk

Visualization - Details for Development

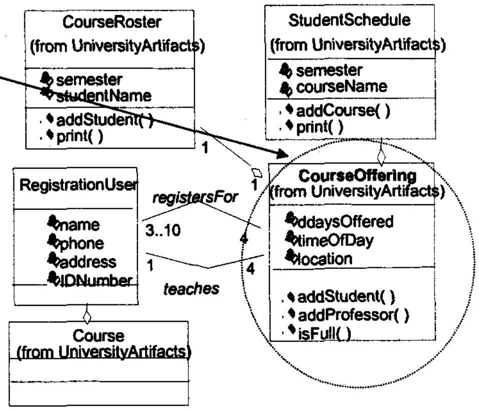
```
public class CourseOffering {

    private int m_ddaysOffered;
    private int m_timeOfDay;
    private int m_location;

    public CourseRoster m_CourseRoster;
    public StudentSchedule m_StudentSchedule;
    public Vector m_registersFor = new Vector();
    public RegistrationUser m_teaches;

    public void addStudent() {     }
    public void addProfessor() {     }
    public return isFull() { }

}
```



CourseRoster
(from UniversityArtifacts)
semester
studentName
addStudent( )
print( )

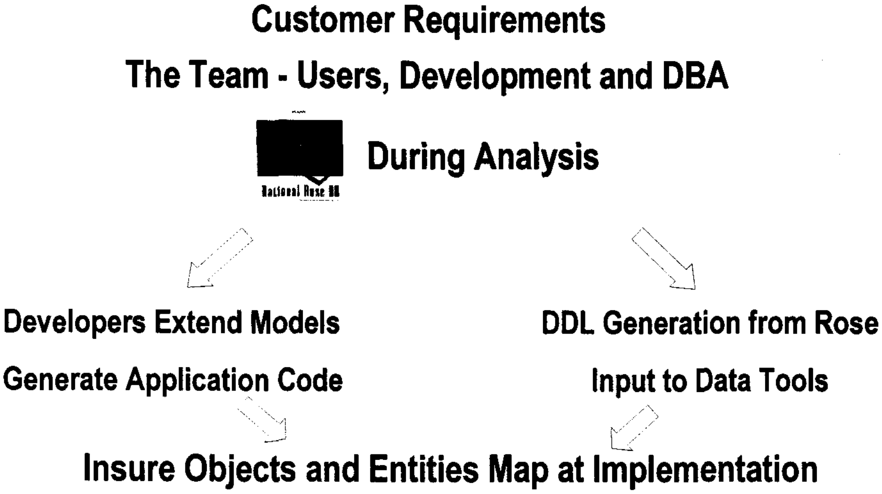
StudentSchedule
(from UniversityArtifacts)
semester
courseName
addCourse( )
print( )

RegistrationUser
name
phone
address
IDNumber

CourseOffering
(from UniversityArtifacts)
ddaysOffered
timeOfDay
location
addStudent( )
addProfessor( )
isFull( )

Course
(from UniversityArtifacts)

Developers and DBA's - One Team



Customer Requirements

The Team - Users, Development and DBA

During Analysis

Developers Extend Models

Generate Application Code

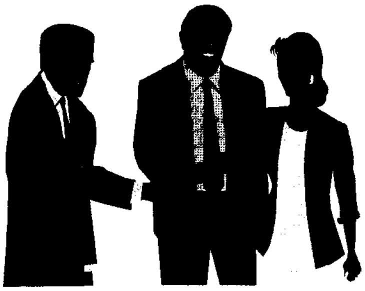
DDL Generation from Rose

Input to Data Tools

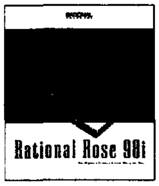
Insure Objects and Entities Map at Implementation

Benefits of Visualization

- "Bridge the Gap" between IT and the Business
- A common "Language" for IT and the Business
- Automated Documentation of your Application
- Foundation for Component Development



Rose - "The Tool for Visualization"



- Model the Business and the Design
- Generate Code: Component and Class Skeletons
- Reverse Engineer an Application
- Visualize and Document the Design of an Application
- Visualize Components Libraries and API's
- Visualize Frameworks for Line of Business - Solution Development

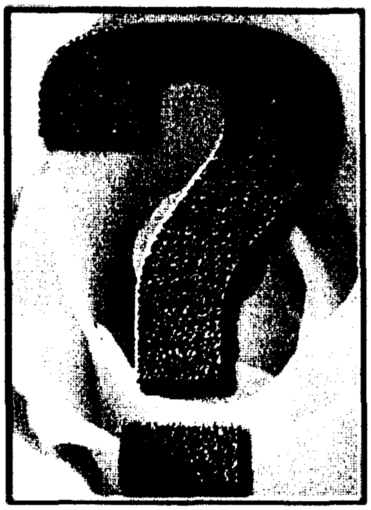
RATIONAL
SOFTWARE
한국레쇼날소프트웨어(주)

Rose Benefits

- Productivity Gains for the Software Developer, Architect, Designer and Analyst
  - Document your Design
  - Model the Business
  - Generate Skeleton of your Components/Classes
- "Change is Manageable"
  - Rose Documents the Design
  - IT Development can Extend/Change the Design
- Use Rose to Establish Line of Business (Vertical) Frameworks - The Drive to Reuse

UNIFIED
MODELING
LANGUAGE

RATIONAL
SOFTWARE
한국레쇼날소프트웨어(주)

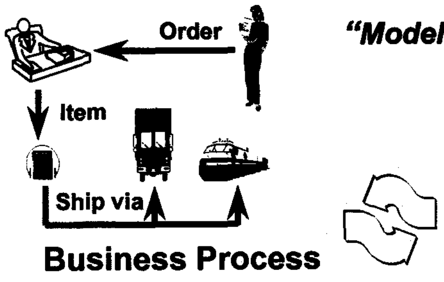


RATIONAL SOFTWARE
한국래쇼날소프트웨어(주)

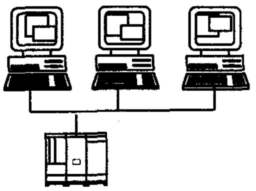
Visual Modeling and the UML
A Technology Overview

RATIONAL SOFTWARE
한국래쇼날소프트웨어(주)

What is Visual Modeling?



"Modeling captures essential parts of the system."
Dr. James Rumbaugh



Computer System

Visual Modeling is modeling using standard graphical notations

Visual Modeling ...

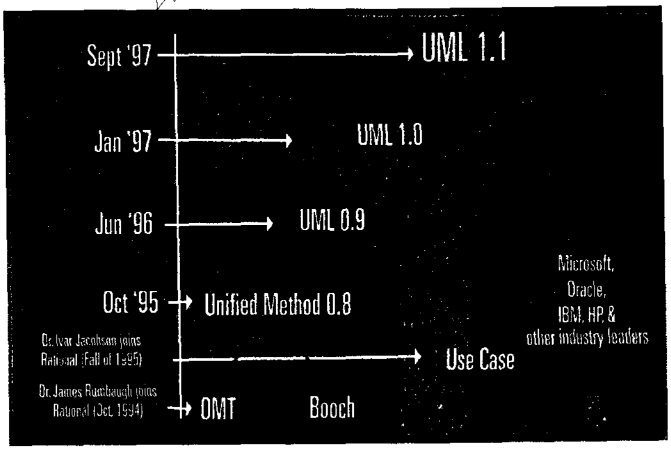
- capture business process
- is a communication tool between business domain and computer domain
- manages complexity
- define software architecture
- promote reuse
- and more

What is the UML?

- UML stands for Unified Modeling Language
- The UML combines the best of the best from:
  - Data Modeling concepts (Entity-Relationship Diagrams)
  - Business Modeling (work flow)
  - Object Modeling
  - Component Modeling
- The UML is the standard language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system
- It can be used with all processes, throughout the development life cycle, and across different implementation technologies

History of the UML

Nov '97 UML approved by the OMG


Sept '97
UML 1.1
Jan '97
UML 1.0
Jun '96
UML 0.9
Oct '95
Unified Method 0.8
Dr. Ivar Jacobson joins Rational (Fall of 1995)
Use Case
Dr. James Rumbaugh joins Rational (Oct. 1994)
OMT
Booch
Microsoft, Oracle, IBM, HP, & other industry leaders

UML Supports Application Development

Objects

Business Objects

Relationships

large scale system

ORDBMS
Oracle

Classes

UNIFIED
MODELING
LANGUAGE

application partitioning

Components
Microsoft

Scenarios

CORBA
OMG

Use Cases

ActiveX/COM
Microsoft

Java Beans

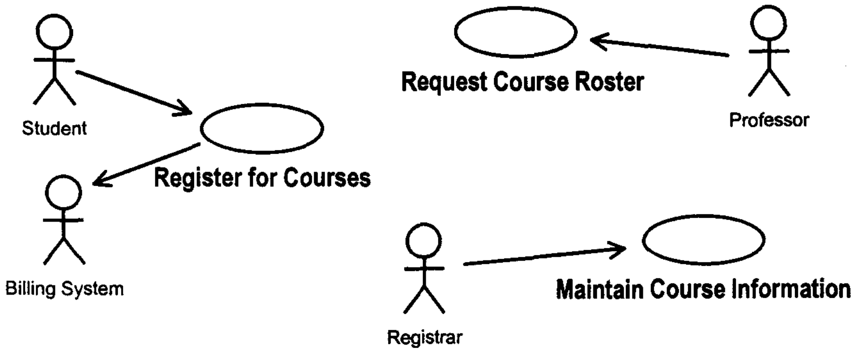
Business Process

UML Concepts

- The UML may be used to visually model:
  - The interaction of your application with the outside world
  - The behavior of your application
  - The structure of your system
  - The architecture of your enterprise
  - The components in your system

Use Case Diagram

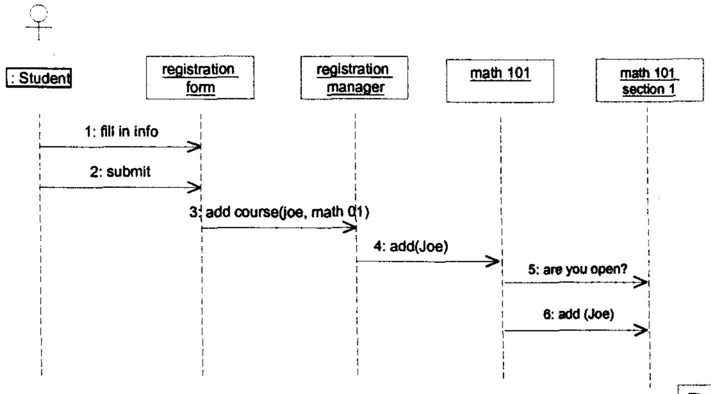
- Use case diagrams are created to visualize the interaction of your system with the outside world



RATIONAL SOFTWARE 한국래쇼날소프트웨어(주)

Sequence Diagram

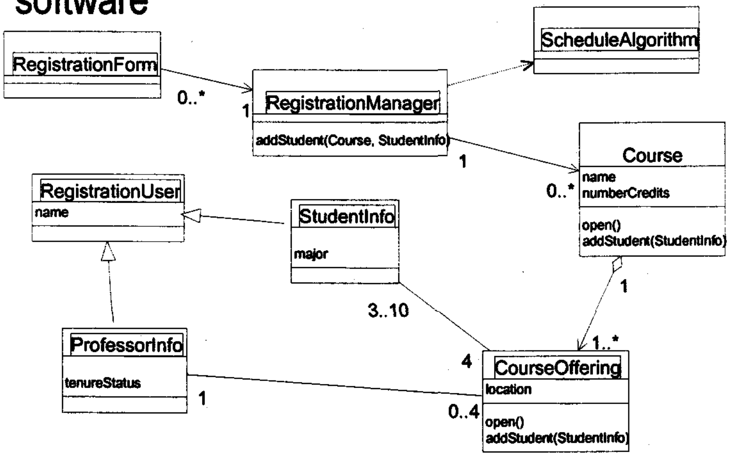
- A sequence diagram shows step-by-step what has to happen to accomplish a piece of functionality provided by the system



RATIONAL SOFTWARE 한국래쇼날소프트웨어(주)

Class Diagram

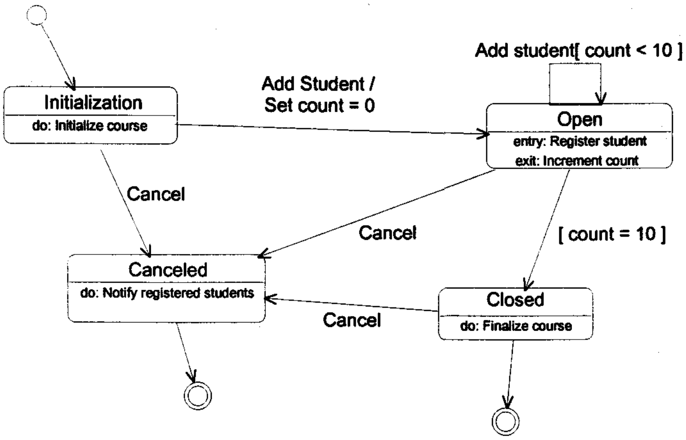
- A class diagram shows the structure of your software



RATIONAL SOFTWARE 한국래쇼날소프트웨어(주)

The State of an Object

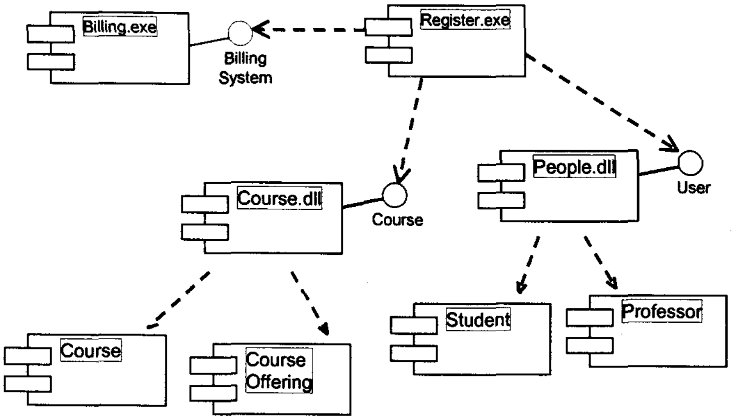
- A state transition diagram shows the lifecycle of a single class



RATIONAL SOFTWARE 한국래쇼날소프트웨어(주)

The Physical World

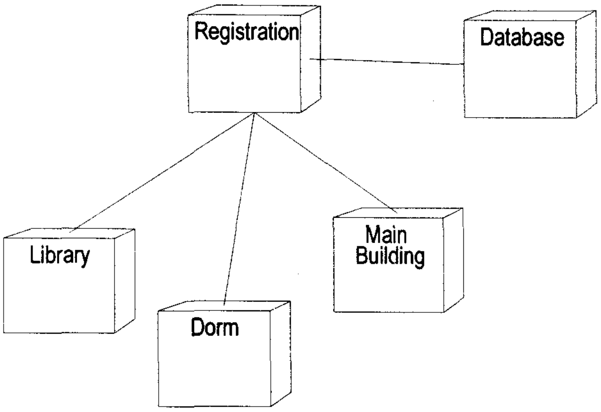
- Component diagrams illustrate the organization and dependencies among software components



RATIONAL SOFTWARE 한국래쇼날소프트웨어(주)

Deploying the System

- The deployment diagram visualizes the distribution of components across the enterprise



RATIONAL SOFTWARE 한국래쇼날소프트웨어(주)

Visual Modeling and UML Summary

- A Use Case Diagram describes the high level view of an application. Used to level set the organization (user, developer, DBA)
- A Sequence Diagram graphically captures the flow of the business operation
- A Class Diagram describes the classes in an application and their relationships
- A State Transition Diagram expresses the detailed behavior of a particular piece of the application
- A Component Diagram shows the relationships between .exe and .dll (physical components) and documents application interdependencies
- The Deployment Diagram visualizes the actual computer systems that the applications run on

UNIFIED MODELING LANGUAGE

RATIONAL SOFTWARE 한국래쇼날소프트웨어(주)

RATIONAL SOFTWARE 한국래쇼날소프트웨어(주)

RATIONAL SOFTWARE

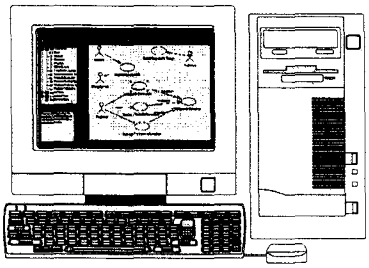
Rose 98*i* Product Overview

RATIONAL SOFTWARE 한국래쇼날소프트웨어(주)

Rose 98*i* - Enterprise Edition

UML Support

Visual Studio Integration



Team Development

Extensible

Rational Suite Integration

Broad Industry Support

RATIONAL SOFTWARE 한국래쇼날소프트웨어(주)

Values and Features of Rational Rose 98*i*

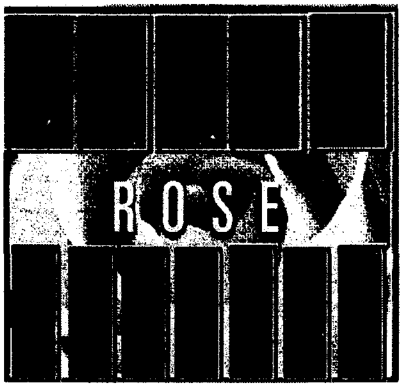
Shorter development cycles	**Controlled iterative development**
Increased developer productivity	**Model-driven development**
Improved software quality	**Use-case and business focused development**
Significant software reuse	**Focus on software architecture and components**
Improved team communication	**Common, standard language-the UML**
Integration with legacy systems	**Reverse-engineering capabilities**

RATIONAL
SOFTWARE
한국래쇼날소프트웨어(주)

How Rose 98*i* is Packaged

- Rose Modeler
  - No language add-ins
- Rose Professional
  - One language add-ins
  - Some non-language add-ins
- Rose Enterprise
  - All language add-ins
  - All non-language add-ins

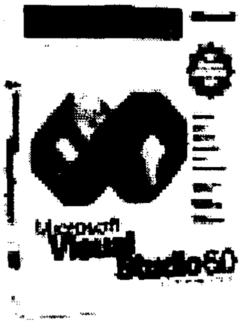
Language Add-Ins



Non-Language Add-Ins

RATIONAL
SOFTWARE
한국래쇼날소프트웨어(주)

Leading Integration with Visual Studio™



- Model Assistant for help with common modeling tasks
- Model Update Tool for reverse engineering
- Code Update Tool for code generation
- Visual Component manager 2.0 integration
  - Rational Rose is the only tool that can publish models directly to VCM
  - Drag and drop models, classes, packages from VCM to Rational Rose
- MS Repository integration

RATIONAL SOFTWARE 한국래쇼날소프트웨어(주)

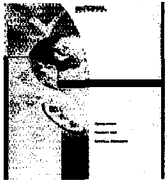
Team Development

- Integration with Rational ClearCase
  - Perform configuration management operations directly from browser or diagram
  - Perform frequent configuration management operations from context menus
  - Get visual feedback on status of units
- Web Publisher add-in (Rose 98*i* Enterprise only)
  - Format and distribute software design information on the world wide web
  - View with any browser such as MS Explorer or Netscape Navigator

RATIONAL SOFTWARE 한국래쇼날소프트웨어(주)

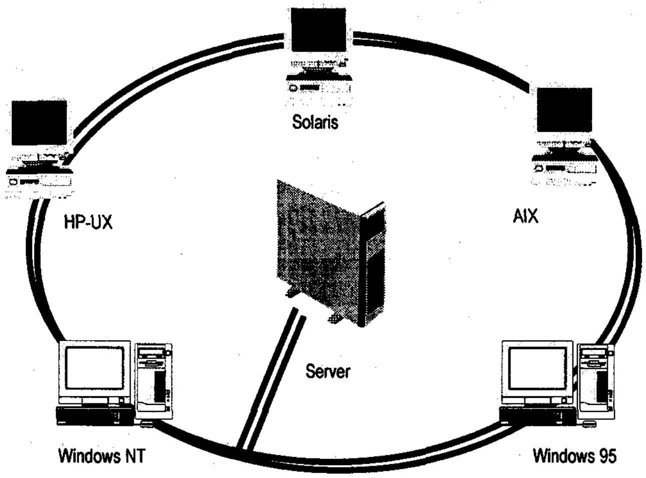
Broad Industry Support

- Add-ins created by Rational RoseLink Partners(over 60+)
  - Languages
    - PowerBuilder, SmallTalk, Delphi, GDMO, etc
  - DB
    - All kind of OODB such as ObjectStore, Versant, O2, Objectivity, etc
  - Middleware
    - Forte, Dynasty, Uniface, BEA M3, OMG CORBA, etc
  - Metrics
  - Reports



RATIONAL SOFTWARE 한국래쇼날소프트웨어(주)

Windows / UNIX Interoperability



RATIONAL SOFTWARE 한국래쇼날소프트웨어(주)

Rose 98*i* Summary - What Rose Does Best!

- Industry Leadership and Standards
  - Business Modeling - ***Ivar Jacobson***
  - Object/Visual Modeling - ***Grady Booch***
  - Data Modeling/Migration to Components - ***Jim Rumbaugh***
- Code Generation is more complete
- Rose can reverse engineer “difficult” code
- Rational/Partner Technology Expertise
- Breadth of Language, IDL and Database support
- Integration with Rational’s leading documentation, requirements management, testing and configuration management tools

RATIONAL SOFTWARE
한국래쇼날소프트웨어(주)

RATIONAL SOFTWARE
한국래쇼날소프트웨어(주)