UMLとRational Roseによるオブジェクト指向分析・設計

Object-Oriented Analysis and Design with UML and Rational Rose

Romi Satria Wahono

Department of Information and Computer Sciences

Graduate School of Science and Engineering

Saitama University

序論

Introduction

- ✓ モデリングとは
- ✓ 成功のトライアングル
- ✓ 表記法の役割

モデリングとは

- モデリングとは、実世界の概念について構築されたモデルを使用して問題を考える方法である。
- モデリングによって、システムに対する要求事項の理解が深まり、設計が整理され、システムの保守がが容易になる。
- モデルとは、抽象化の産物であり、本質でない詳細を隠することにより複雑な問題や構造の本質的部分を表現し、問題の理解を容易にしたものである。

成功のトライアングル

- 表記法（プロセスを伝達すること）、プロセス（使い方）、ツール（作業の成果物を文書化すること）の三つの面が必要になる。

表記法の役割

- コードそのものからは、自明でない、あるいは推測できない決定を伝達する言語として役割を果たす。
- 言語や実装上のあらゆる重要な決定内容を表現するのに十分なセマンティックスを提供する。
- 人間が論理的に考えたり、ツールによって扱うのに十分な具体的な形式を提供する。

UMLとは

What is the UML

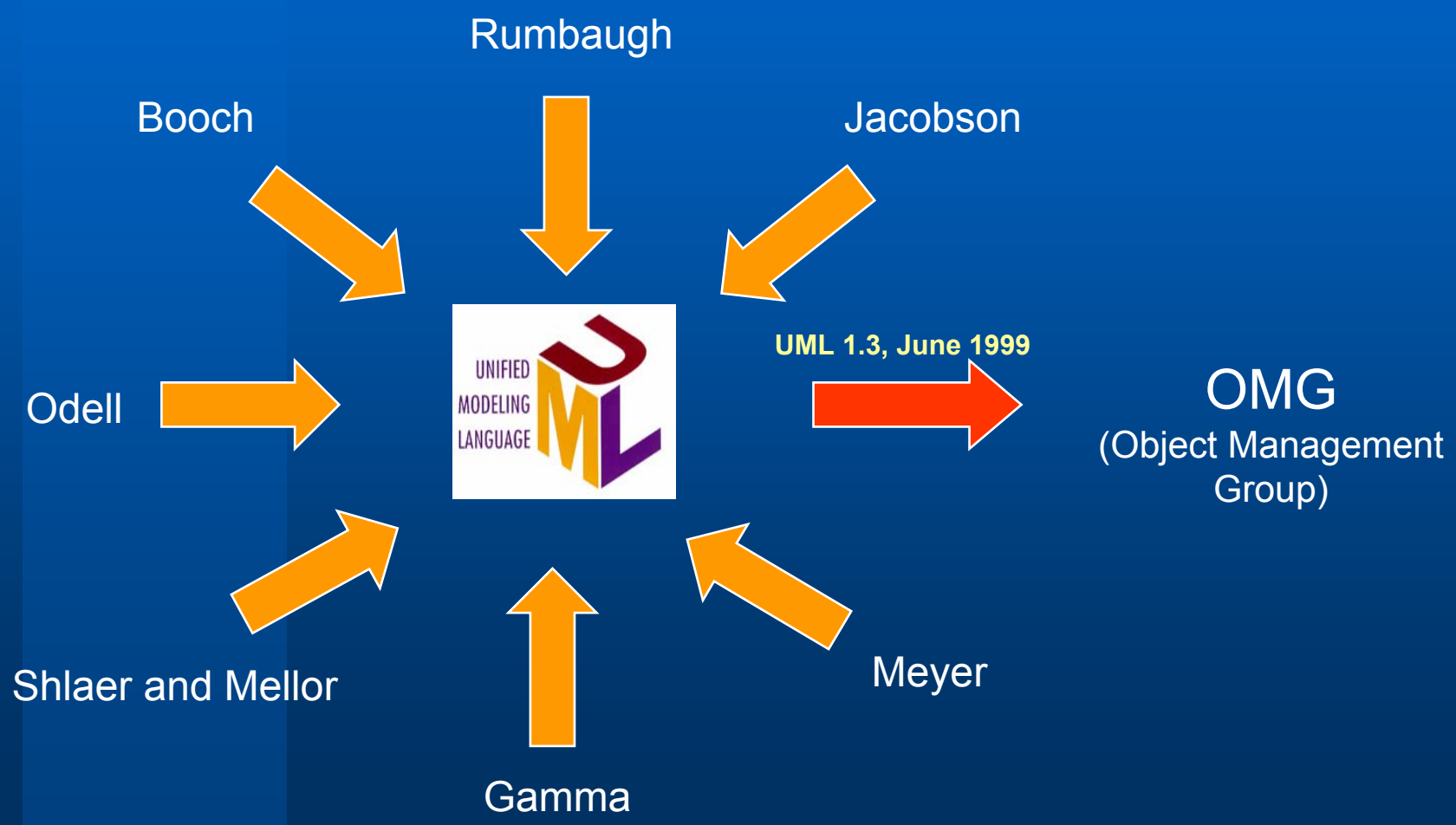
- ✓ UMLの歴史
- ✓ UMLとは
- ✓ UMLを提供しているOODツール

UMLの歴史 -1-

- 1990年に入って、さまざまな種類の方法論がそれぞれの独自の表記法とともにに、OMT法、Booch法、OOSE法など登場した。
- それぞれの方法論は収束し始めましたが、まだ方法論ごとに独自の表記法が使われていた。
- 表記法に関してていえば、方法論戦争は統一モデリング言語(UML)の採択とともに終わりを告げた。

方法論戦争
（Method Wars）

UMLの歴史 -2-


Rumbaugh
Booch
Jacobson
Odell
UNIFIED
MODELING
LANGUAGE
UML 1.3, June 1999
OMG
(Object Management
Group)
Shlaer and Mellor
Meyer
Gamma

UMLとは

- UMLは開発中のオブジェクト指向システムの成果物を定義し、視覚化し、文書化するために用いられる言語である。
- UMLは前のスライドに示すように、Booch法、OMT法、Objectory法の表記法の統一したものであり、その他のさまざまな方法論から最善の概念を採用している方法論である。
- UMLは分析及び設計の成果物であるセマンティックモデル、構文的な表記、図を標準化するための試みである。

UMLを提供しているOODツール

- Rational Rose (www.rational.com)
- Together (www.togethersoft.com)
- Object Domain (www.objectdomain.com)
- Jvision (www.object-insight.com)
- Objecteering (www.objecteering.com)
- MagicDraw (www.nomagic.com/magicdrawuml)
- Visual Object Modeller (www.visualobject.com)
- など

Rational Roseによる オブジェクト指向分析･設計

- ✓ 開発の工程
- ✓ プロジェクトの定義と問題記述
- ✓ ユースケースモデルの作成
- ✓ クラスの識別
- ✓ 関係の識別
- ✓ 振る舞いと属性の追加
- ✓ 継承の識別
- ✓ システムアーキテクチャーの設計

開発の工程

- プロジェクトの定義と問題記述
- ユースケースモデルの作成
- クラスの識別
- 関係の識別
- 振る舞いと属性の追加
- 継承の識別
- システムアーキテクチャーの設計

プロジェクトの定義と問題記述

プロジェクトの定義と問題記述

- プロジェクトを開始する前に、新しいシステムについてアイデア「システムの要求事項や形式」を持たなければならない。
- 解決すべき問題を、チーム内や顧客との間で文書化し、話し合う。仮説が提示されると、概念を立証するためのプロトタイプ技法を使って、それらの仮説を立証したり、却下したりする。
- このフェースのアウトプットは、外部インタフェースの洗い出し、初期のリスク評価、及び一連のシステム要求である。

問題記述の例:
大学コース履修登録　-1-

- 各学期の始めに、学生はその学期に必要なコース科目のリストを記載したコースカタログを請求できる。それには、それぞれのコースについての情報（教授、学部、前提科目など）が含まれており、学生はその情報に基づいてコースを決定できる。
- 新しいシステムを使って、学生は次の学期の4つのコース科目を選択できる。さらに、各学生はコース科目が満員のとき、または中止されたときに備える代替コース科目を2つ指定する。コースの受講人数は10人未満3人以上である。希望する学生が3人に満たないコース科目は中止となる。学生の登録処理が完了すると、登録システムは支払請求システムに情報を送り、学生にその学期の学費を請求できるようにする。

問題記述の例:
大学コース履修登録　-2-

- 教授がオンラインシステムにアクセスして、担当する予定のコースを指定したり、自分のコース科目にどの学生が申し込んでいるかを確認できなければならない。
- 各学期には、学生がそのスケジュールを変更できる期間がある。その期間中は、学生がシステムにアクセスして、コースの追加または削除ができなければならない。

ユースケースモデルの作成

ユースケースモデル

- 開発中のシステムの振る舞い、システムがどのような機能を提供しなければならないかは、ユースケースモデルに記述する。
- ユースケースモデルはシステムが提供する機能（ユースケース）、システム外界（アクター）、及びユースケースとアクターとの関係を明らかにする（ユースケース図）。

アクター

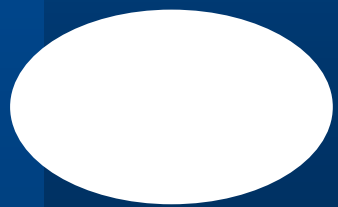
- アクターはシステムの一部ではなくて、システムとやり取りしなければならない人や物を表す。
- アクターは問題記述の中や顧客及び領域の専門家との対話によって発見される。
- アクターのUML表記法:

アクターの識別　（質問型識別）

- 特定の要求に利害関係を持つのは誰か
- システムは組織内のどこで使われるのか
- システムを使うことによって恩恵を受けるのは誰か
- システムにこの情報を供給し、この情報を使い、この情報を削除するのはだれか
- システムをサポートし、保守するのは誰か
- システムは外部のシステム資源を使うか
- 1人の人間がさまざまな役割を担うか
- 複数の人間が同じ役割を担うのか
- システムは従来のシステムとやり取りするのか

ユースケース

- ユースケースはアクターとシステムとの間の対話をモデル化する。システムが提供する機能、システムによってアクターに提供されるサービスを表現する。
- ユースケースとは、システムによって実現される一連の処理であり、かつその処理結果として特定のアクターに対して「明確利用価値」（Measurable Value）が生み出すような処理を表す。(正式な定義）
- ユースケースのUML記法：



ユースケースの識別（質問型識別）

- 各アクターの仕事は何か
- システム内の情報の作成、格納、変更、削除、読み出しを行うアクターが存在するか
- どのユースケースが、この情報の作成、格納、変更、削除、読み出しを行うか
- 突然の外部的な変化をシステムに知らされる必要があるアクターが存在する。
- システム内の特定のできごとについて、通知を受ける必要のあるアクターが存在するか
- どのユースケースがシステムをサポート及び保守するのか
- 機能的な要求をすべて、これらのユースケースによって実現できるか

ユースケースの関係　-1-

- アクターとユースケースとの間には、関連が存在することがある。この関係は、アクターとユースケースとの間のコミュニケーションを表すことであるから、しばしばcommunicates関連と呼ばれている。
- ユースケースの間の関係には、使用（uses）と拡張（extends）の2種類がある。
- 複数のユースケースが同じ機能の部分を共有することがある場合、共通機能を必要とするすべてのユースケースで定義するのではなく、独立した新しいユースケースに分割定義する。
- 使用（uses）関係は、この新しいユースケースとその機能を使うほかのユースケースとの間に使用される。

ユースケースの関係　-2-

- 拡張（extends）関係は次のものを示すために使われる。
  - 選択的な振る舞い
  - アラームを鳴らすような、特定の状況下でのみ実行される振る舞い
  - アクターの選択に基づいて実行される可能性のある各種のフロー
- UMLにはステレオタイプと呼ばれる概念がある。ステレオタイプによって、基本のモデリング要素を拡張して新しい要素を作ることができる。
- ステレオタイプ名は二重の不等号記号（<<>>）で囲んで、関係直線に沿って配置する。

ユースケースの関係　-3-

- ユースケースの関係の表記

Communicates関連

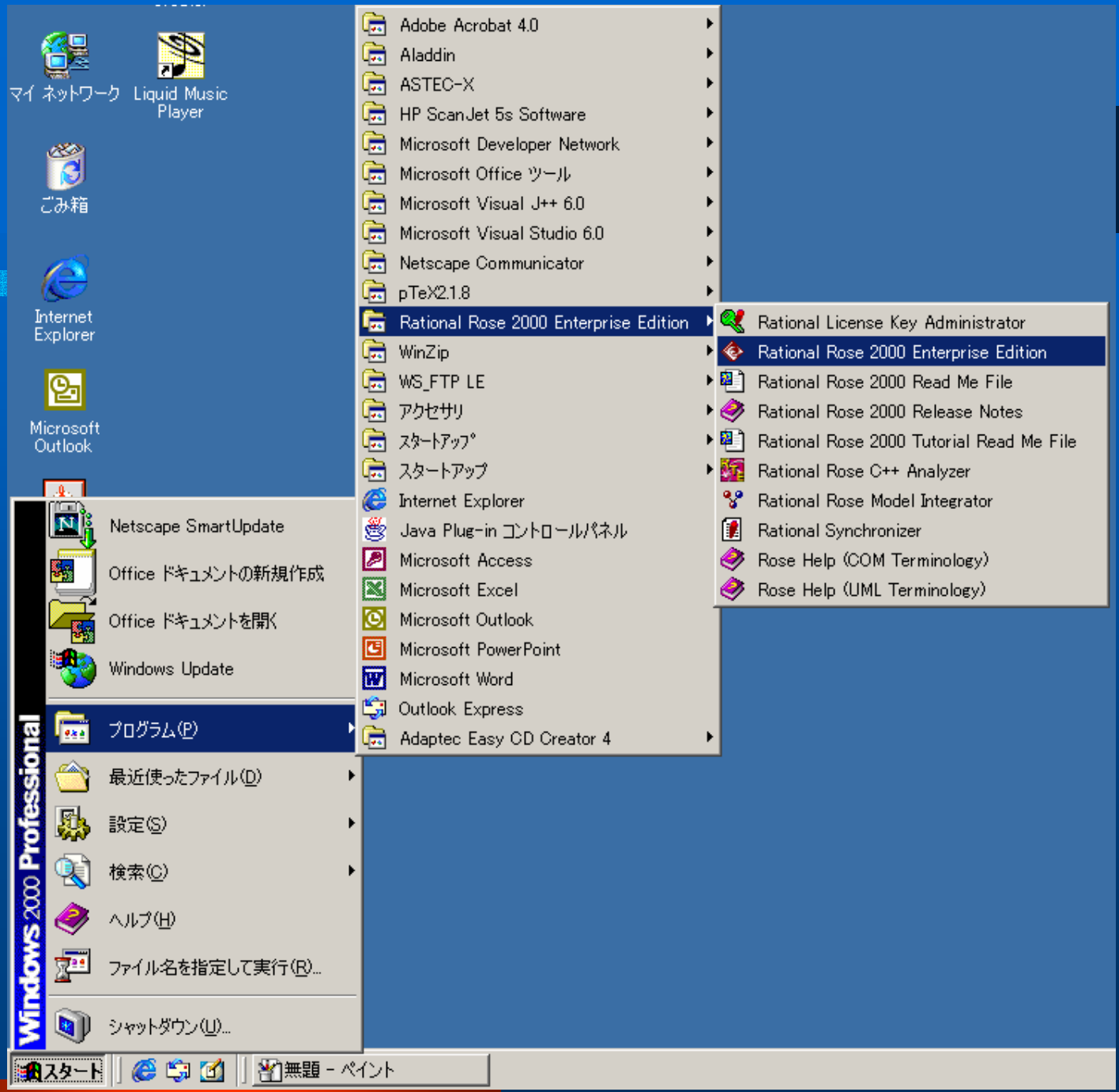
Communicates関連+ステレオタイプ

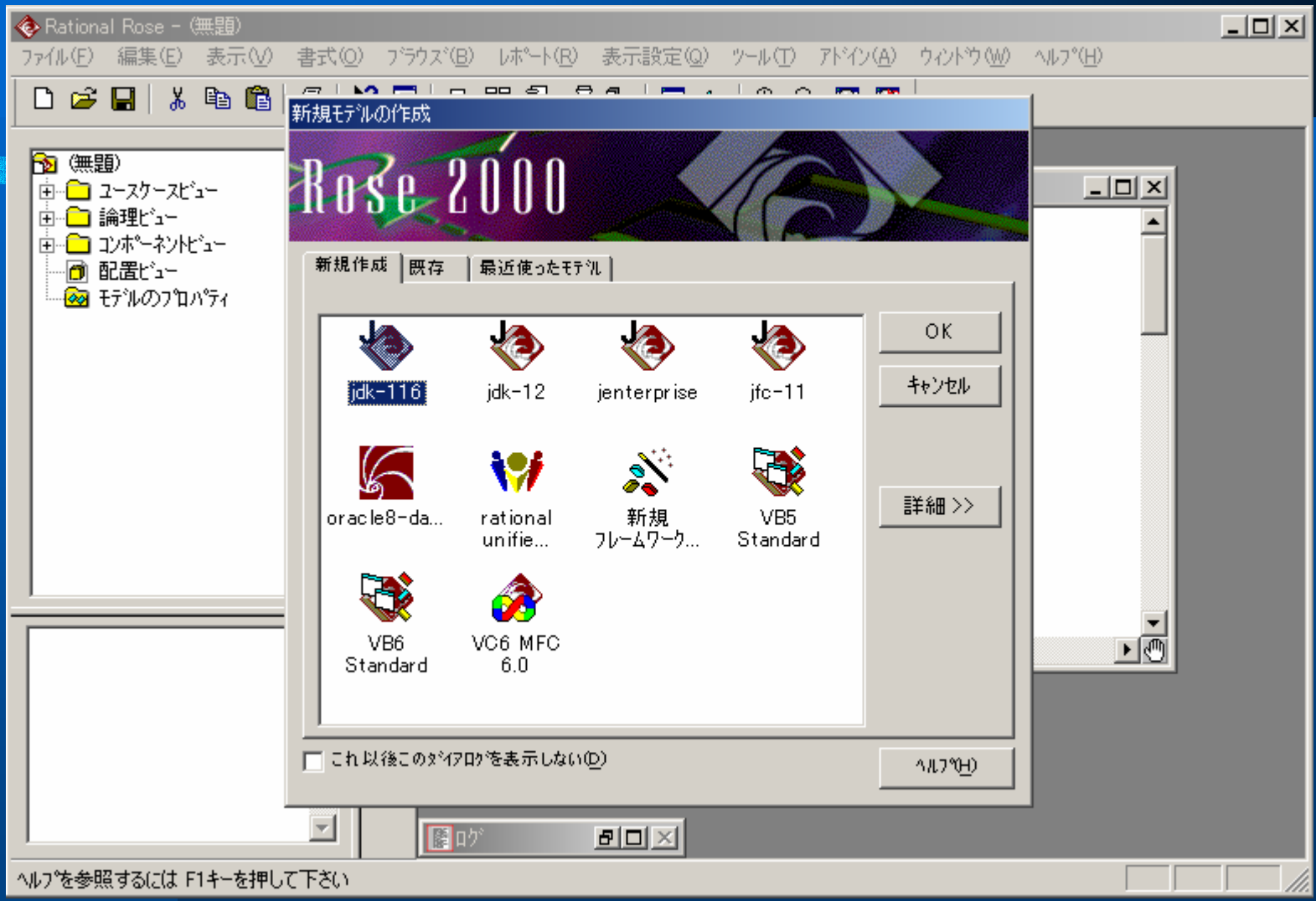
Uses関係+ステレオタイプ

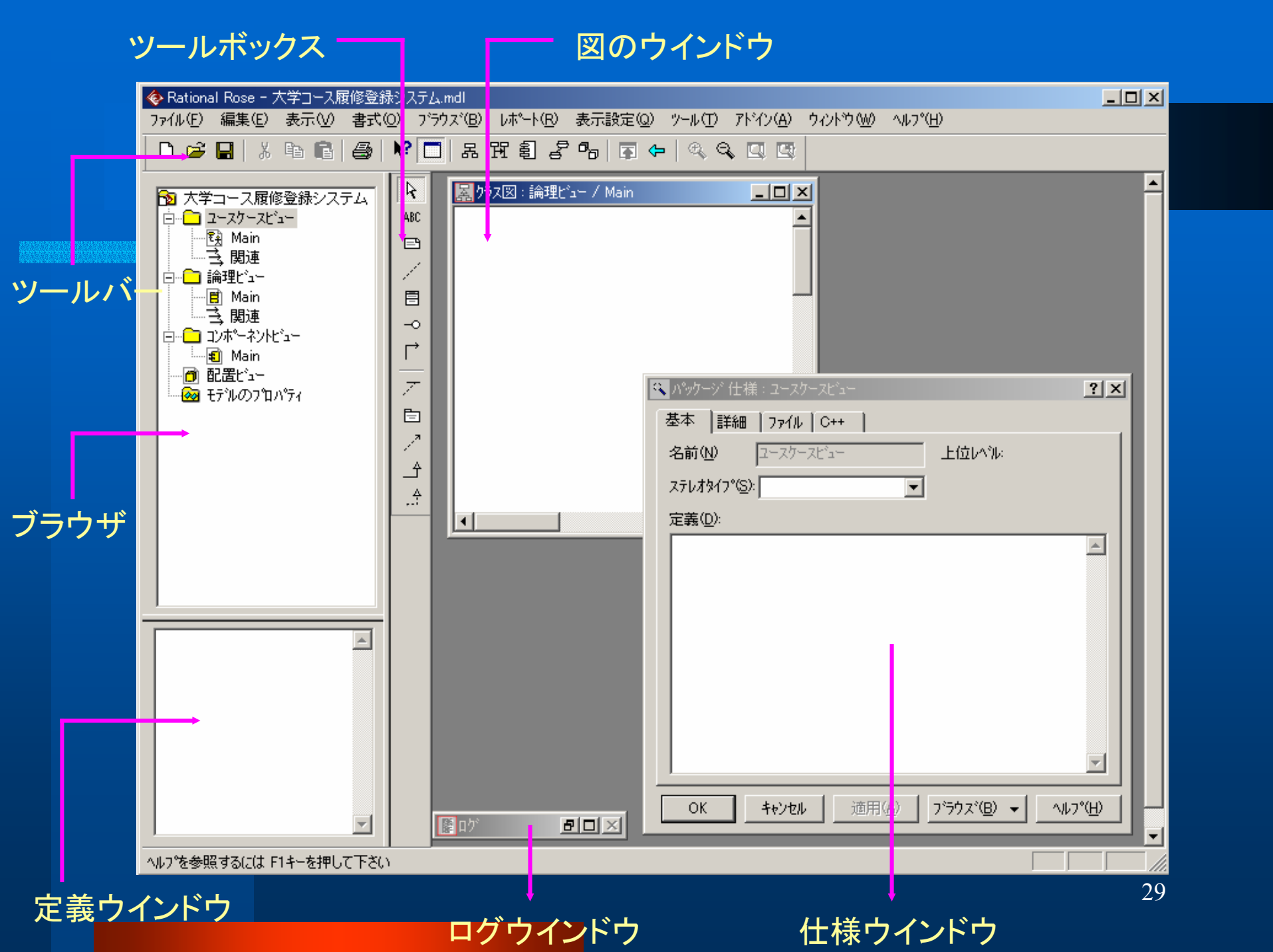
Extends関係+ステレオタイプ

ユースケースモデルの例:
大学コース履修登録: Rose 入門

- まず、Rational Rose 2000というソフトウエアを起動する。（NIMA1とNIMA9）
- 起動ときに[フレームワーク]ダイアログボックスが表示される。このダイアログボックスからは、アイテムを既に定義してあるモデルをロードできるため、すべてのアイテムを一から設計し直すのではなく、それぞれのシステム固有の部分だけに集中することが可能になる。
- フレームワークとは別に、Rational RoseのGUIでは、4種類のウインドウを使用して、モデル内のアイテムの表示、作成、変更、操作、定義などを実行できる。


マイ ネットワーク
Liquid Music Player
ごみ箱
Internet Explorer
Microsoft Outlook
Windows 2000 Professional
Netscape SmartUpdate
Office ドキュメントの新規作成
Office ドキュメントを開く
Windows Update
プログラム(P)
最近使ったファイル(D)
設定(S)
検索(C)
ヘルプ(H)
ファイル名を指定して実行(R)...
シャットダウン(U)...
スタート
無題 - ペイント
Adobe Acrobat 4.0
Aladdin
ASTEC-X
HP ScanJet 5s Software
Microsoft Developer Network
Microsoft Office ツール
Microsoft Visual J++ 6.0
Microsoft Visual Studio 6.0
Netscape Communicator
pTeX2.1.8
Rational Rose 2000 Enterprise Edition
WinZip
WS_FTP LE
アクセサリ
スタートアップ
スタートアップ
Internet Explorer
Java Plug-in コントロールパネル
Microsoft Access
Microsoft Excel
Microsoft Outlook
Microsoft PowerPoint
Microsoft Word
Outlook Express
Adaptec Easy CD Creator 4
Rational License Key Administrator
Rational Rose 2000 Enterprise Edition
Rational Rose 2000 Read Me File
Rational Rose 2000 Release Notes
Rational Rose 2000 Tutorial Read Me File
Rational Rose C++ Analyzer
Rational Rose Model Integrator
Rational Synchronizer
Rose Help (COM Terminology)
Rose Help (UML Terminology)


Rational Rose - (無題)
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
(無題)
ユースケースビュー
論理ビュー
コンポーネントビュー
配置ビュー
モデルのプロパティ
新規モデルの作成
Rose 2000
新規作成
既存
最近使ったモデル
jdk-116
jdk-12
jenterprise
jfc-11
oracle8-da...
rational unifie...
新規 フレームワーク...
VB5 Standard
VB6 Standard
VC6 MFC 6.0
OK
キャンセル
詳細 >>
これ以後このダイアログを表示しない(D)
ヘルプ(H)
ログ
ヘルプを参照するには F1キーを押して下さい


ツールボックス
図のウインドウ
ツールバー
ブラウザ
定義ウインドウ
ログウインドウ
仕様ウインドウ
Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
Main
関連
論理ビュー
Main
関連
コンポーネントビュー
Main
配置ビュー
モデルのプロパティ
クラス図：論理ビュー / Main
パッケージ 仕様：ユースケースビュー
基本
詳細
ファイル
C++
名前(N)
ユースケースビュー
上位レベル:
ステレオタイプ(S):
定義(D):
OK
キャンセル
適用(A)
ブラウズ(B)
ヘルプ(H)
ログ
ヘルプを参照するには F1キーを押して下さい

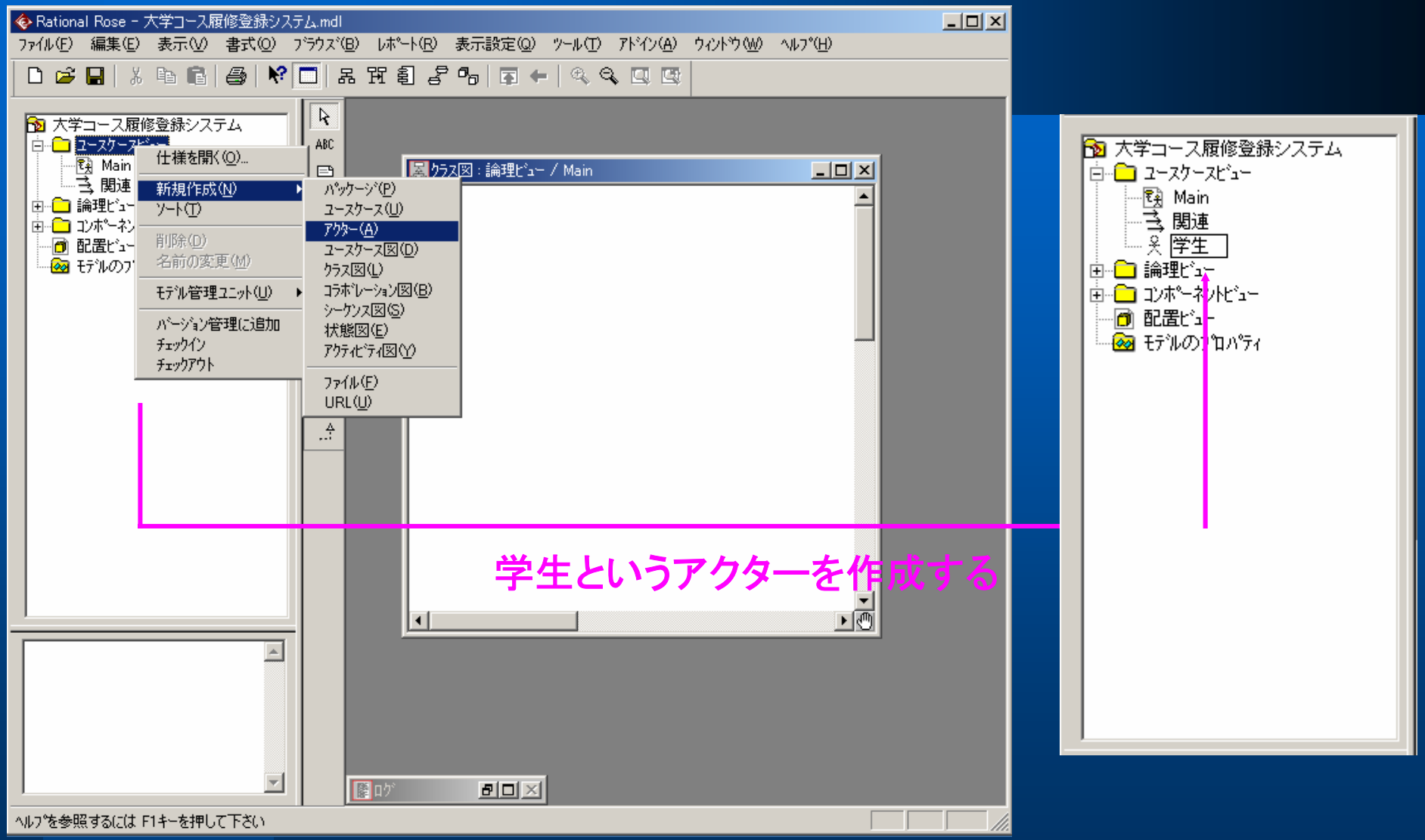
ユースケースモデルの例：
大学コース履修登録：アクターの識別

- 学生は履修コースを登録したい
- 教授は教えるコースを選択したい
- 教務はカリキュラムを作成し、その学期のカタログを生成しなければならない
- 教務はコース、教授、及び学生に関するあらゆる情報を保守しなければならない
- 支払システムは、このシステムから支払請求情報を受け取らなければならない

提出された質問への答えに基づいて、学生（Student）、教授（Professor）、教務（Registrar）、支払請求システム（Billing System）の各アクターが識別された。

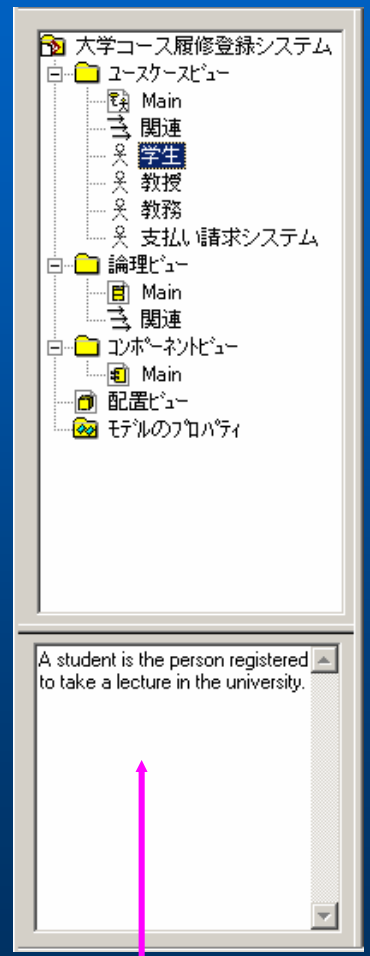
ユースケースモデルの例：
大学コース履修登録：アクターの作成

1. ブラウスの[ユースケースビュー]パッケージを右クリックして、ショートカットメニューを表示する。
2. [新規作成]をポイントし、[アクター]をクリックする。“NewClass”という名前の新しいアクターがブラウズに配置される。
3. NewClassアクターが選択されたままの状態で、適当なアクター名を入力する。


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
Main
関連
論理ビュー
コンポーネントビュー
配置ビュー
モデルのプロパティ
仕様を開く(O)...
新規作成(N)
ソート(T)
削除(D)
名前の変更(M)
モデル管理ユニット(U)
バージョン管理に追加
チェックイン
チェックアウト
パッケージ(P)
ユースケース(U)
アクター(A)
ユースケース図(D)
クラス図(L)
コラボレーション図(B)
シーケンス図(S)
状態図(E)
アクティビティ図(Y)
ファイル(F)
URL(U)
クラス図：論理ビュー / Main
ログ
ヘルプを参照するには F1キーを押して下さい
学生というアクターを作成する
学生

ユースケースモデルの例：
大学コース履修登録:アクター定義の作成

1. 定義ウインドウが表示されていない場合、[表示]メニューの[アイテムの定義]をクリックして、定義ウインドウを開く。
2. ブラウズでアクターをクリックして選択する。
3. 定義ウインドウにカーソルを移動し、定義を入力する。



ユースケースモデルの例:
大学コース履修登録: ユースケースの識別 1

- 学生(Student)アクターは、コースを登録するためのシステムを使う必要がある
- コース選択処理が完了したら、支払請求システム(Billing System)に支払請求情報が送られなければならない
- 教授(Professor)アクターは、ある学期に相当するコースを選択するためにシステムを使う必要があり、コース登録簿をシステムから受け取ることができなければならない
- 教務(Registrar)は、ある学期にコースカタログの生成と、システムが必要とする、カリキュラム、学生及び教授についての全情報の保守に責任を負う

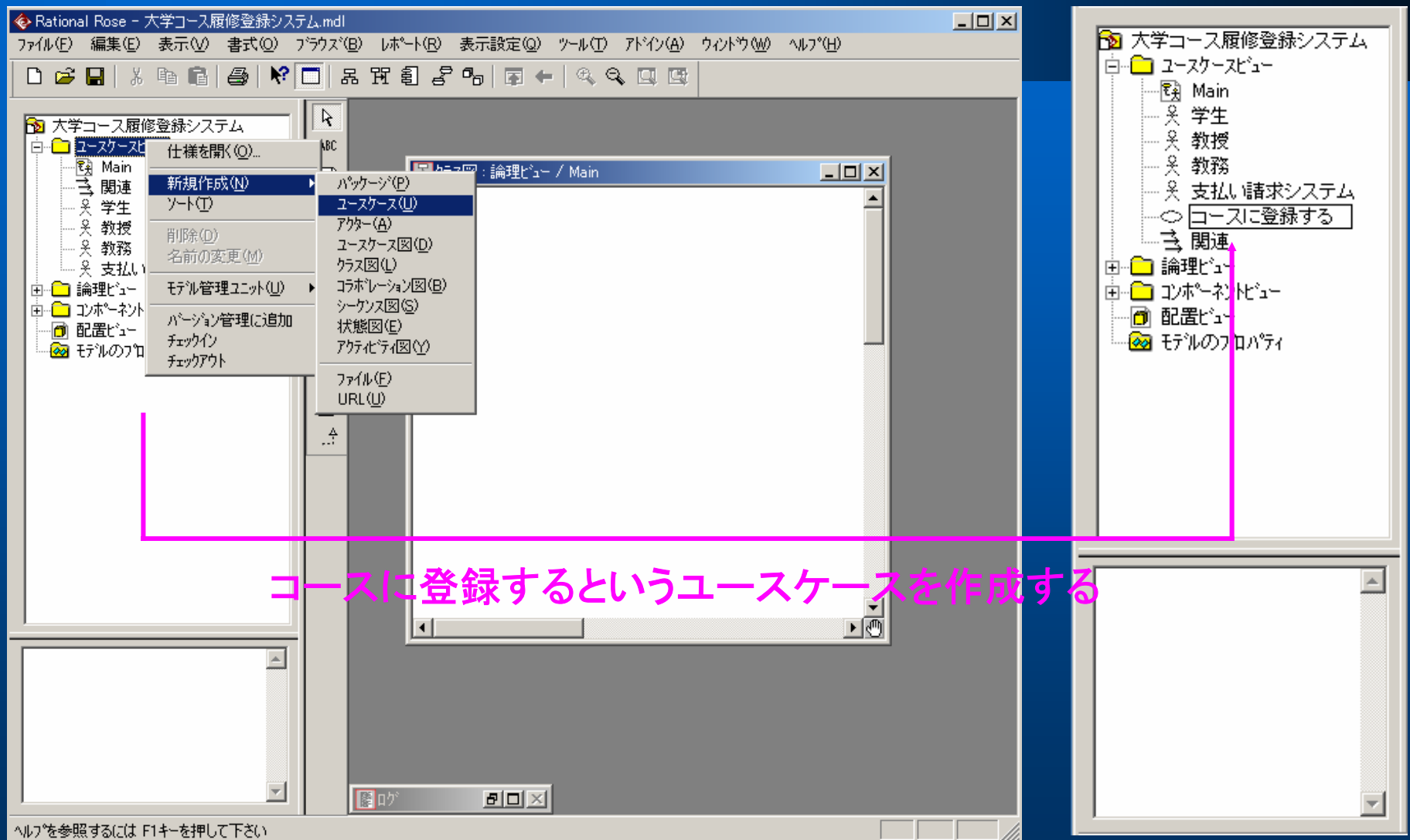
ユースケースモデルの例:
大学コース履修登録: ユースケースの識別 2

前のスライドのニーズに基づいて、次のユースケースが識別された。

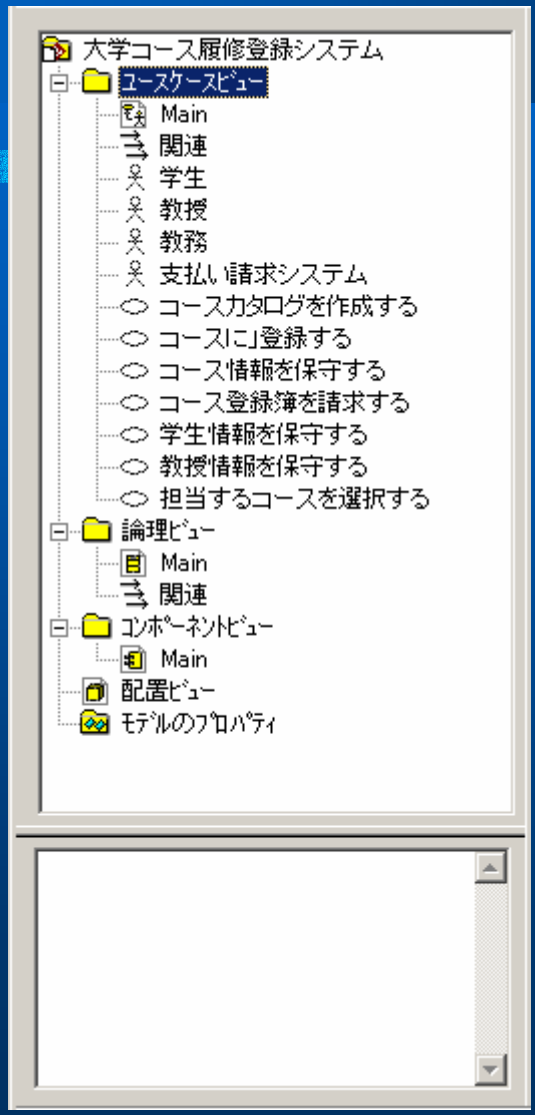
- コースに登録する（Register for Courses）
- 相当するコースを選択する（Select Courses to Teach）
- コース登録簿を請求する（Request Course Roaster）
- コース情報を保守する（Maintain Course Information）
- 教授情報を保守する（Maintain Professor Information）
- 学生情報を保守する（Maintain Student Information）
- コースカタログを作成する（Create Course Catalog）

ユースケースモデルの例:
大学コース履修登録: ユースケースの作成

1. ブラウズで[ユースケースビュー]を右クリックして、ショートカットメニューを表示する。
2. [新規作成]をポイントし、[ユースケース]をクリックする。名前のない、新しいユースケースがブラウズに配置される。
3. ユースケースが選択されたままの状態で、希望するユースケース名を入力する。



コースに登録するというユースケースを作成する

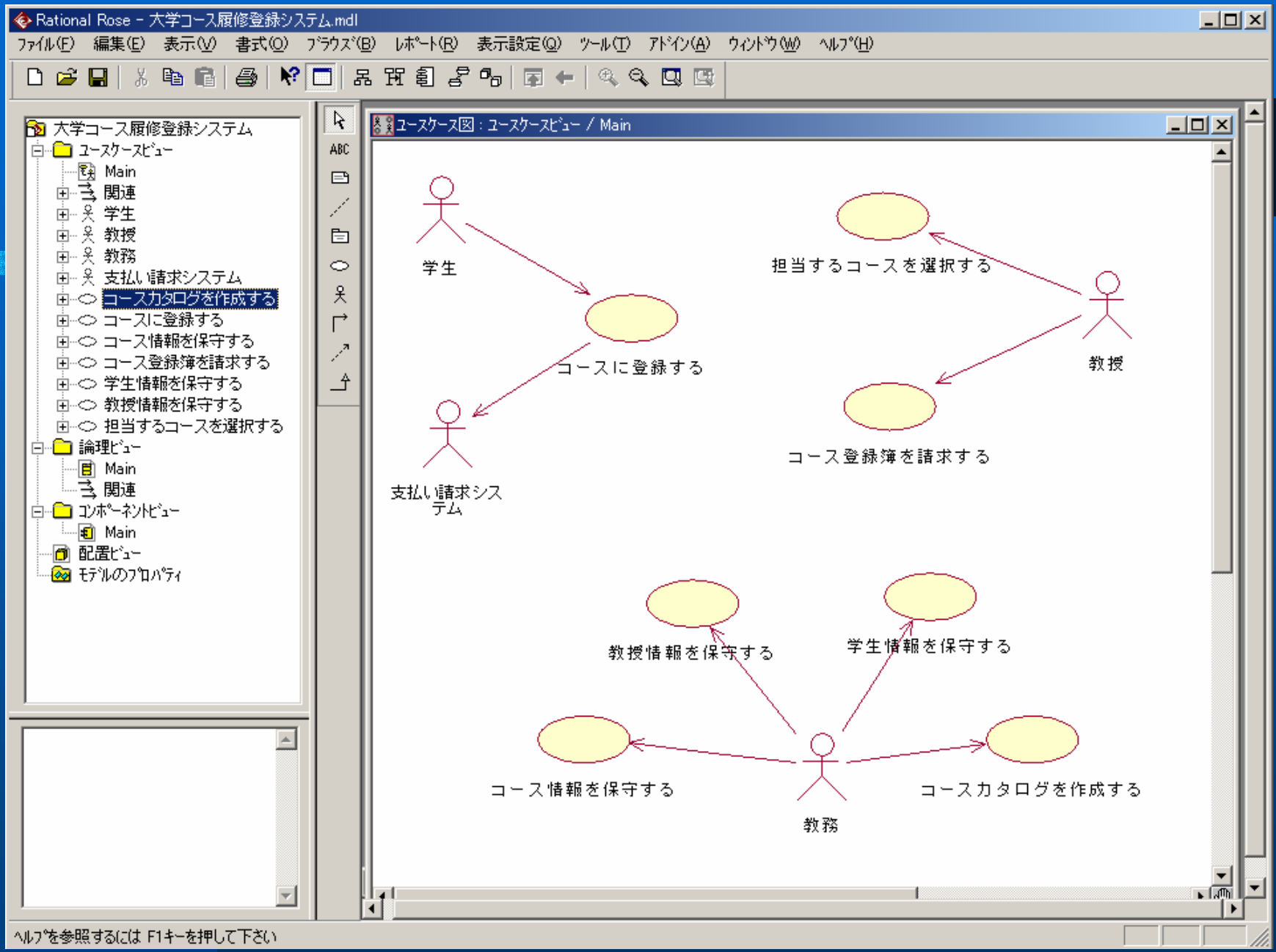

大学コース履修登録システム
ユースケースビュー
Main
関連
学生
教授
教務
支払い請求システム
コースカタログを作成する
コースに登録する
コース情報を保守する
コース登録簿を請求する
学生情報を保守する
教授情報を保守する
担当するコースを選択する
論理ビュー
Main
関連
コンポーネントビュー
Main
配置ビュー
モデルのプロパティ

ユースケースモデルの例：
大学コース履修登録：メインユースケース図の作成

1. ブラウズで[ユースケースビュー]のメイン（Main）図をクリックして、図を開く。
2. ブラウズでアクターを選択して、図上にドラックする。
3. 図に必要なアクターごとに、手順2を繰り返す。
4. ブラウズでユースケースをクリックして選択し、図上にドラックする。
5. 図に必要なユースケースごとに、手順4を繰り返す。

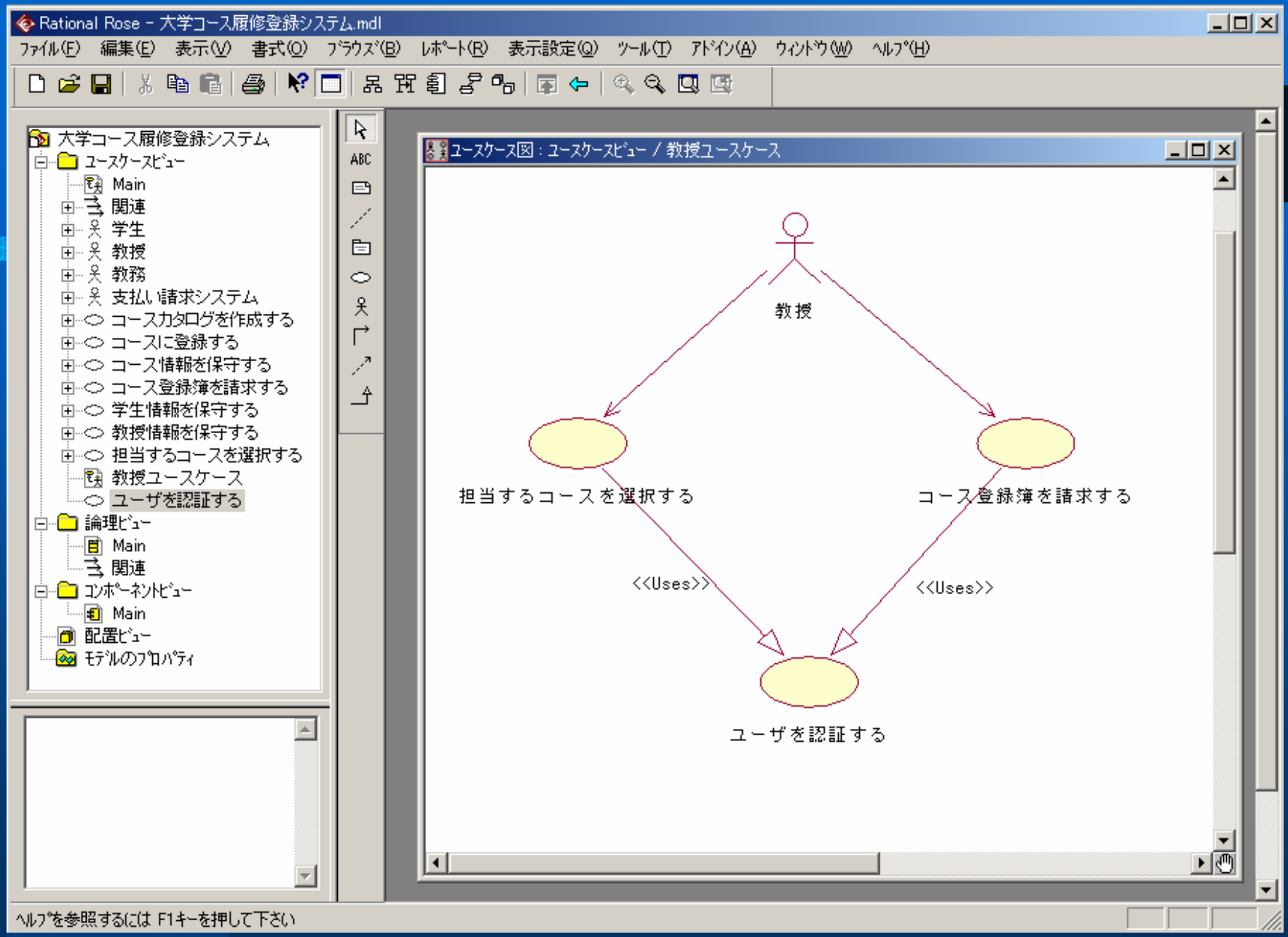
ユースケースモデルの例：
大学コース履修登録:Communicates関連の作成

1. 図のツールバーにある[関連]アイコンまたは[単方向の関連]アイコンをクリックして選択する。
2. コミュニケーションを起動するアクターをクリックし、所望のユースケースに向かって関連直線をドラックする。


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
Main
関連
学生
教授
教務
支払い請求システム
コースカタログを作成する
コースに登録する
コース情報を保守する
コース登録簿を請求する
学生情報を保守する
教授情報を保守する
担当するコースを選択する
論理ビュー
Main
関連
コンポーネントビュー
Main
配置ビュー
モデルのプロパティ
ユースケース図 : ユースケースビュー / Main
学生
担当するコースを選択する
教授
コースに登録する
コース登録簿を請求する
支払い請求システム
教授情報を保守する
学生情報を保守する
コース情報を保守する
教務
コースカタログを作成する
ヘルプを参照するには F1キーを押して下さい

ユースケースモデルの例:
大学コース履修登録: ユースケース図の追加作成

1. ブラウズで[ユースケースビュー]を右クリックして、ショートカットメニューを表示する。
2. [新規作成]をポイントし、[ユースケース図]をクリックする。
3. ユースケース図が選択されている間に、ユースケース図の名前を入力する。
4. 図を開き、必要に応じて、アクター、ユースケース、及びやり取りを図に追加する。


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
Main
関連
学生
教授
教務
支払い請求システム
コースカタログを作成する
コースに登録する
コース情報を保守する
コース登録簿を請求する
学生情報を保守する
教授情報を保守する
担当するコースを選択する
教授ユースケース
ユーザを認証する
論理ビュー
Main
関連
コンポーネントビュー
Main
配置ビュー
モデルのプロパティ
ABC
ユースケース図 : ユースケースビュー / 教授ユースケース
教授
担当するコースを選択する
コース登録簿を請求する
<<Uses>>
<<Uses>>
ユーザを認証する
ヘルプを参照するには F1キーを押して下さい

ユースケースモデルの例:
大学コース履修登録:ステレオタイプの追加

1. 直線関連をダブルクリックして、仕様を表示する。
2. ステレオタイプを含む場合は、[ステレオタイプ]ボックスに“Communicates”と入力する。Communicatesステレオタイプが既に作成されている場合は、[ステレオタイプ]ボックスの矢印のクリックしてドロップダウンメニューを表示し、[Communicates]を選択する。
3. [OK]をクリックして、仕様を閉じる。
4. 関連直線を右クリックして、ショートカットメニューを表示する。
5. [ステレオタイプを表示]をクリックする。

ユースケースモデルの例:
大学コース履修登録:使用(Uses)の作成

1. ツールバーの[汎化]アイコンをクリックして選択する。
2. 使用する側のユースケースをクリックし、使用される側のユースケースに[汎化]アイコンをドラックする。
3. 汎化矢印をダブルクリックして、仕様を表示する。
4. 仕様関係を始めてさくせいする場合、[ステレオタイプ]ボックスに“Uses”と入力する。Usesステレオタイプが既に作成されている場合、[ステレオタイプ]ボックスの矢印をクリックしてドロップダウンメニューを表示する。
5. [OK]をクリックして、仕様を閉じる。
6. 汎化矢印を右クリックして、ショートカットメニューを表示する。
7. [ステレオタイプを表示]をクリックする。

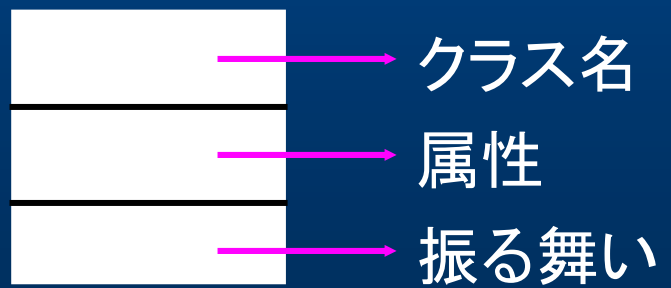
ユースケースモデルの例:
大学コース履修登録:拡張(Extends)の作成

1. ツールバーの[汎化]アイコンをクリックし、選択する。
2. 拡張される機能を含むユースケースをクリックし、基本となるユースケースに[汎化]アイコンをドラックする。
3. 汎化矢印をダブルクリックして、仕様を表示する。
4. 拡張関係を初めて作成する場合、[ステレオタイプ]ボックスに“Extends”と入力する。Extendsステレオタイプが既に作成されている場合、[ステレオタイプ]ボックスの矢印をクリックしてドロップダウンメニューを表示し、[Extends]を選択する。
5. [OK]をクリックして、仕様を閉じる。
6. 汎化矢印を右クリックして、ショートカットメニューを表示する。
7. [ステレオタイプを表示]をクリックする。

クラスの識別

オブジェクトとクラスとは

- オブジェクトとは、実世界または概念上存在するもの、明確な境界と適用目的を持った概念、抽象概念、または物体のことである。
- クラスとは、共通の特性（属性）、共通の振る舞い（操作）、他のオブジェクトとの共通の関係及び共通のセマンティックを持ったオブジェクト群を表したものである。
- クラスのUML表記法：



クラスとステレオタイプ

- ステレオタイプによって、基本のモデリング要素を拡張して新しい要素を作ることができる。
- クラスはユースケース図と同様にステレオタイプを持つことができる。
- 一般なくラスのステレオタイプは、エンティティ（entity）、バウンダリ（boundary）、コントロール（control）がある。
- クラスのステレオタイプは二重の不等号記号（<<>>）で囲んでクラス名の下に表す。

クラスの識別

- クラスを発見あるいは識別するための「虎の巻」は存在しない。Grady Boochも「骨が折れる作業よ！」といったことが知られている。
- Rational Roseは、クラスを発見するため Rational Objectoryプロセスを推薦している。
- Rational Objectoryプロセスでは、バウンダリ、コントロール、及びエンティティの各クラスを探すことによって、開発中にシステムのクラスを発見あるいは識別することを提唱している。

クラスの識別： エンティティクラス

- エンティティクラスは、一般に長い間存続する情報とそれに関連する振る舞いをモデル化する。
- エンティティクラスは、一般何らかの責務を果たすためにシステムに必要となるクラスである。
- 最初のステップは、明らかにされたユースケースのイベントフローに定義された責務（つまり、システムが何をしなければならないか）を検証することである。その責務を記述するために使われる名詞または名詞句を候補にするというやり方はたたき台として有効なことがある。
- しかし、名詞のリストには、問題領域とは無関係の名詞、単なる言語表現に過ぎない名詞、冗長な名詞、クラス構造の記述である名詞などが含まれることがあるため、それらを取り捨て選択しなければならない。

クラスの識別： バウンダリクラス

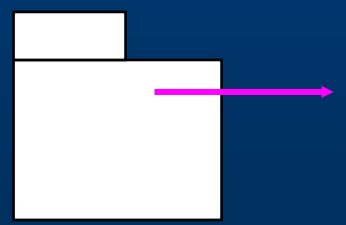
- バウンダリクラスは、システムの外部とシステムの内部との間のやり取りを処理し、ユーザまたは別のシステムへのインタフェイス（アクターへのインタフェイス）を提供する。
- バウンダリクラスは、システムのインタフェイスをモデル化するために使われる。
- バウンダリクラスを発見するためには、物理的なアクターとシナリオの組のそれぞれを検証する。

クラスの識別： コントロールクラス

- コントロールクラスは、1つまたは複数のユースケースに特有な順序付けられた振る舞いをモデル化する。
- コントロールクラスは、ユースケースに定義された振る舞いを実現するために必要なイベントの制御を行う。
- コントロールクラスは、ユースケースを「駆動」したり、実行したりするものと考えられる。

パッケージ

- 大半のシステムは、多数のクラスから構成されているため、使いやすさ、保守性、再利用性を高めるために、クラスを分類整理するメカニズムが必要である。そのメカニズムは、パッケージと言われている。
- モデル論理ビューにおけるパッケージは、互いに関係する下位にのパッケージやクラスを集めたものである。クラスをパッケージにまとめることにより、モデルをより高いレベル視点から見ることができる。
- UMLでは、パッケージはフォルダーとして実現されている。パッケージのUML表記法は:



クラス識別の例：
大学コース履修登録： バウンダリクラスの発見

- ユースケースでは、教授アクターとだけやりとる。
- ユースケースでは、教授が選択を修正、削除、確認、および印刷できることが述べている。
- ユースケースで述べられているような、教授が利用できる全てのオプションを含むクラスが、この必要を満たすために作成される。
- よって、教授用コースオプション（ProfessorCourseOptions）、コース科目追加（AddACourseOption）の2つのバウンダリクラスを識別できる。

クラス識別の例：
大学コース履修登録： エンティティクラスの発見

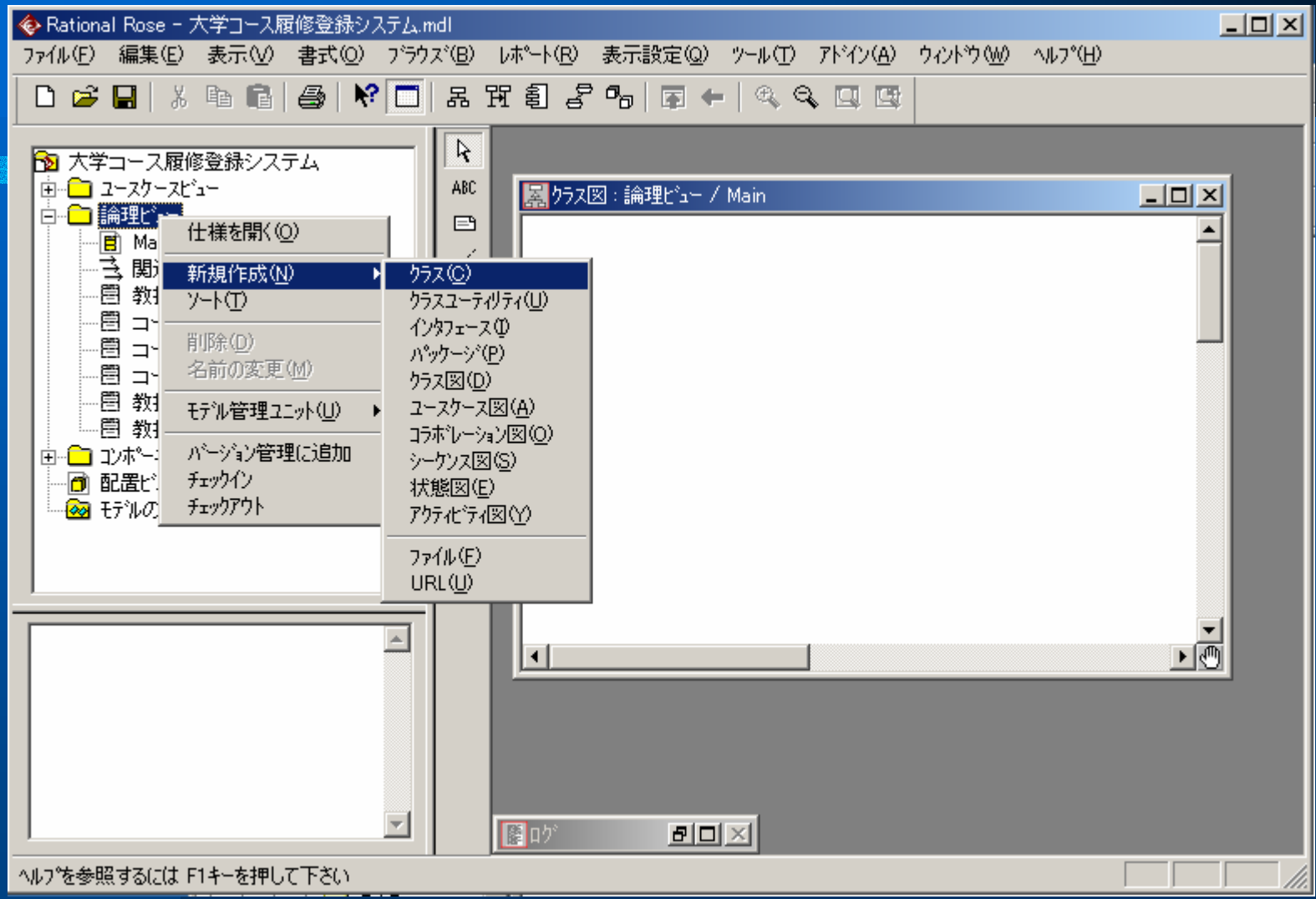
- このシナリオは、コース、コース科目、及び教授の割り当てを扱う。
- コース（Course）、コース科目（CourseOffering）、および教授情報（ProfessorInformation）の3つのエンティティクラスを識別できる。

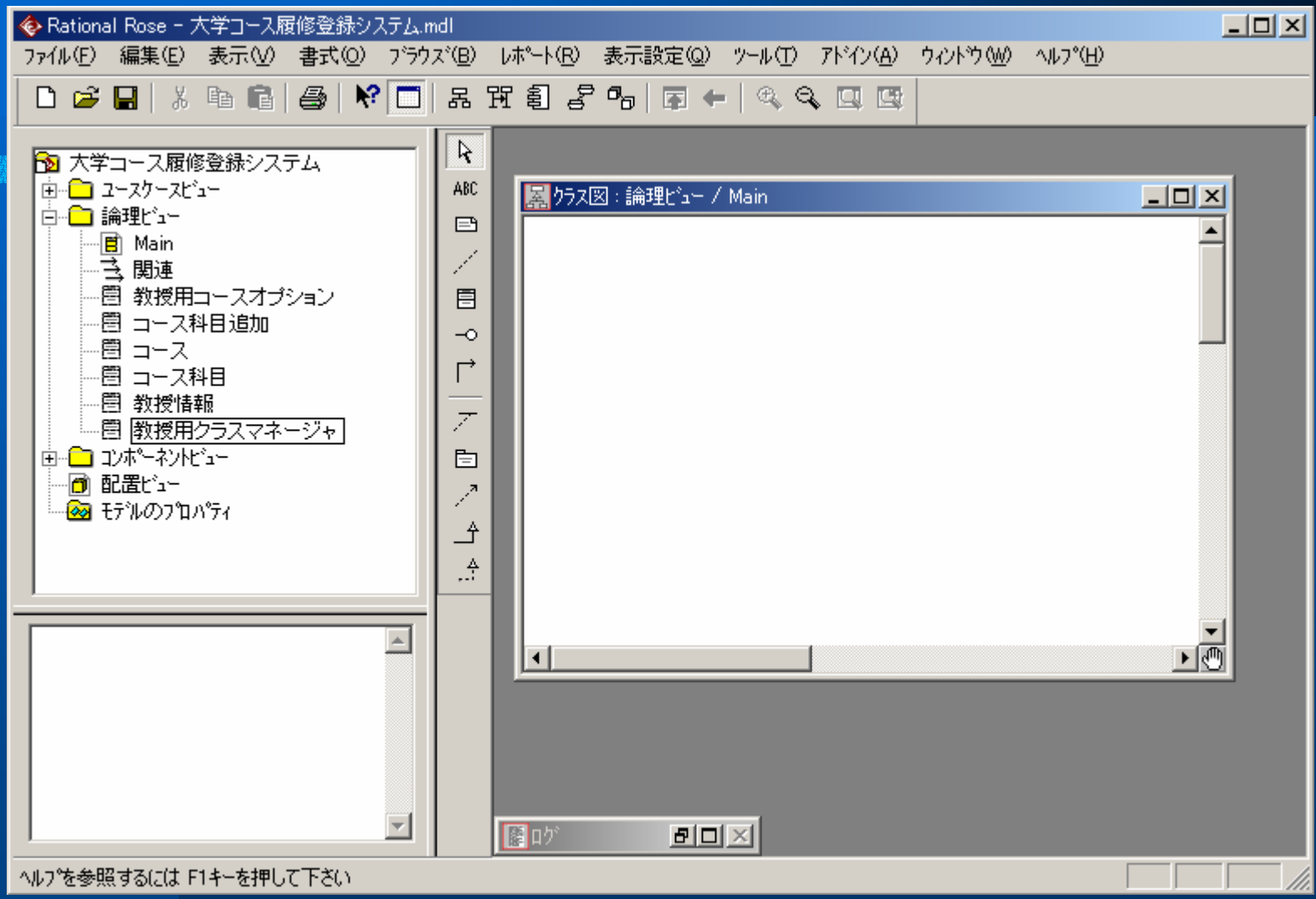
クラス識別の例：
大学コース履修登録： コントロールクラスの発見

- 教授用コースマネージャ（ProfessorCourseManager）1つのコントロールクラスを識別できる。

クラス識別の例:
大学コース履修登録: クラスの作成

1. ブラウズで、[論理ビュー]を右クリックして選択する。
2. [新規作成]をポイントし、[クラス]をクリックする。“NewClass”という名前のクラスがブラウズに配置される。
3. 新しいクラスが選択されている状態で、クラスの名前を入力する。


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
仕様を開く(O)
新規作成(N)
ソート(T)
削除(D)
名前の変更(M)
モデル管理ユニット(U)
バージョン管理に追加
チェックイン
チェックアウト
クラス(C)
クラスユーティリティ(U)
インタフェース(I)
パッケージ(P)
クラス図(D)
ユースケース図(A)
コラボレーション図(O)
シーケンス図(S)
状態図(E)
アクティビティ図(Y)
ファイル(F)
URL(U)
クラス図：論理ビュー / Main
ログ
ヘルプを参照するには F1キーを押して下さい

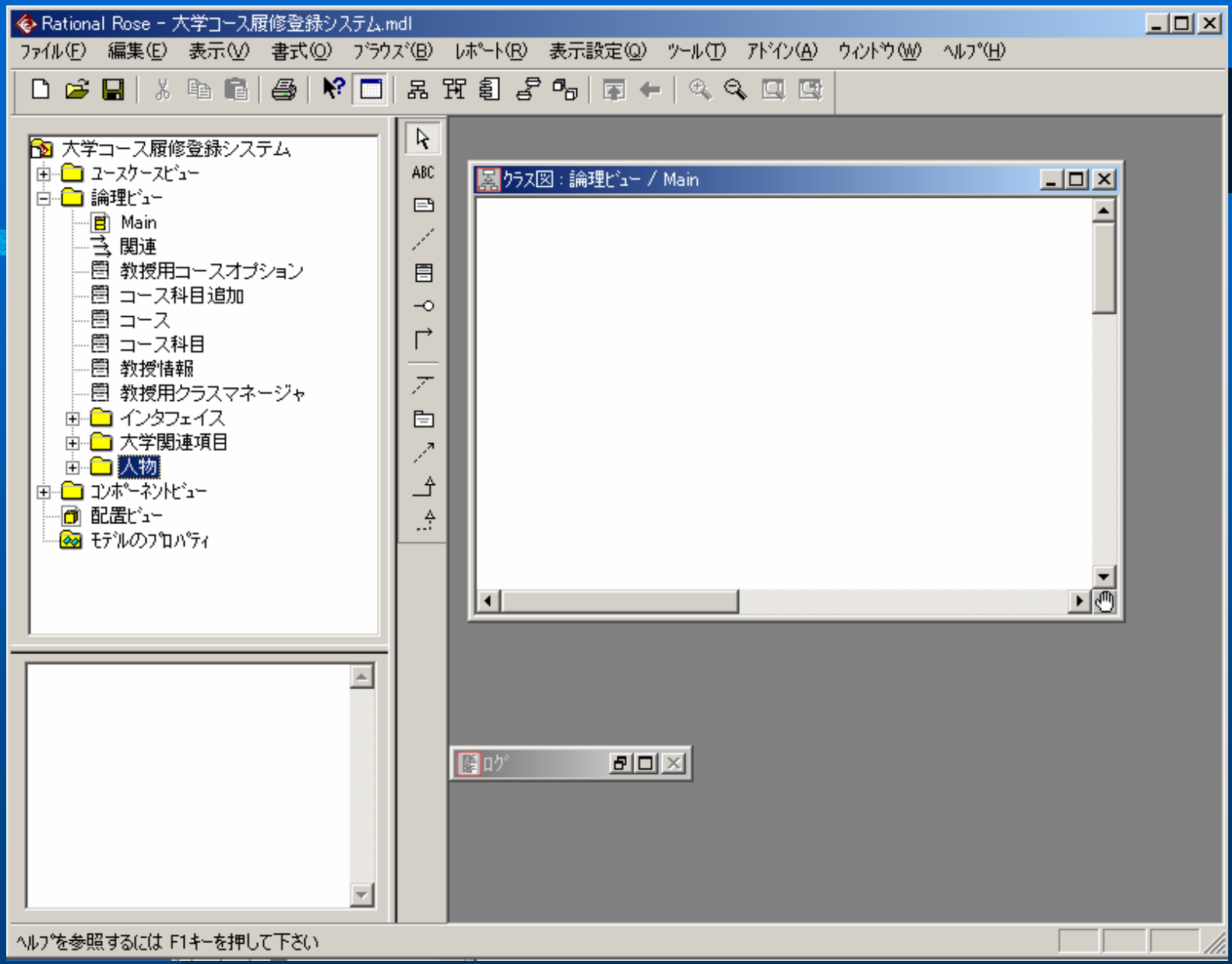

Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
Main
関連
教授用コースオプション
コース科目追加
コース
コース科目
教授情報
教授用クラスマネージャ
コンポーネントビュー
配置ビュー
モデルのプロパティ
ABC
クラス図：論理ビュー / Main
ログ
ヘルプを参照するには F1キーを押して下さい

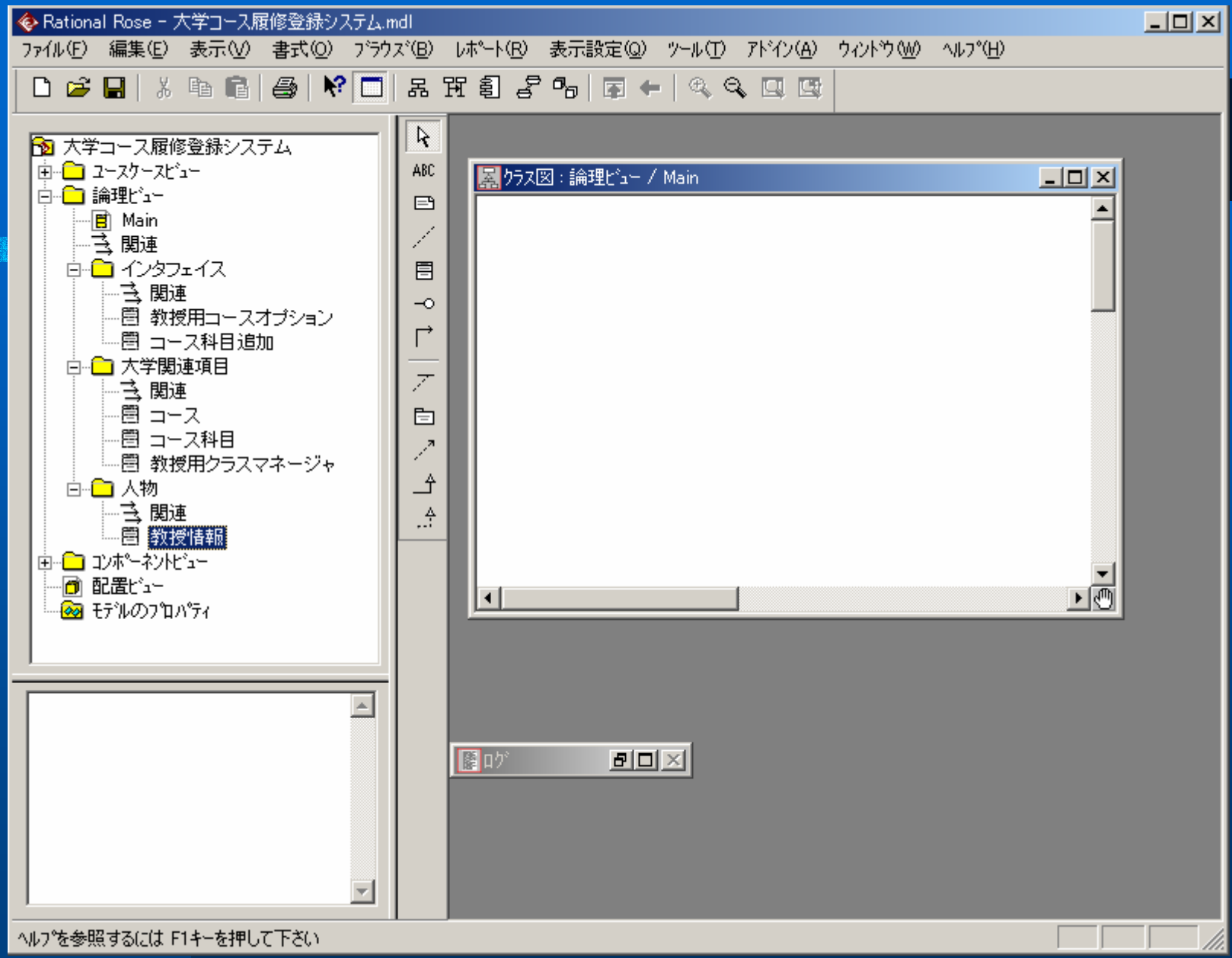
クラス識別の例：
大学コース履修登録： パッケージの識別

- 大学に特有の要素、人物についての情報を含むもの、アクターへのインタフェイスとなるもの、という3つの論理グループに分類される。
- すなわち、インタフェイス（Interfaces）、大学関連項目（UniversityArtifact）、人物（People）という2つのパッケージを識別できる。

クラス識別の例：
大学コース履修登録： パッケージの作成

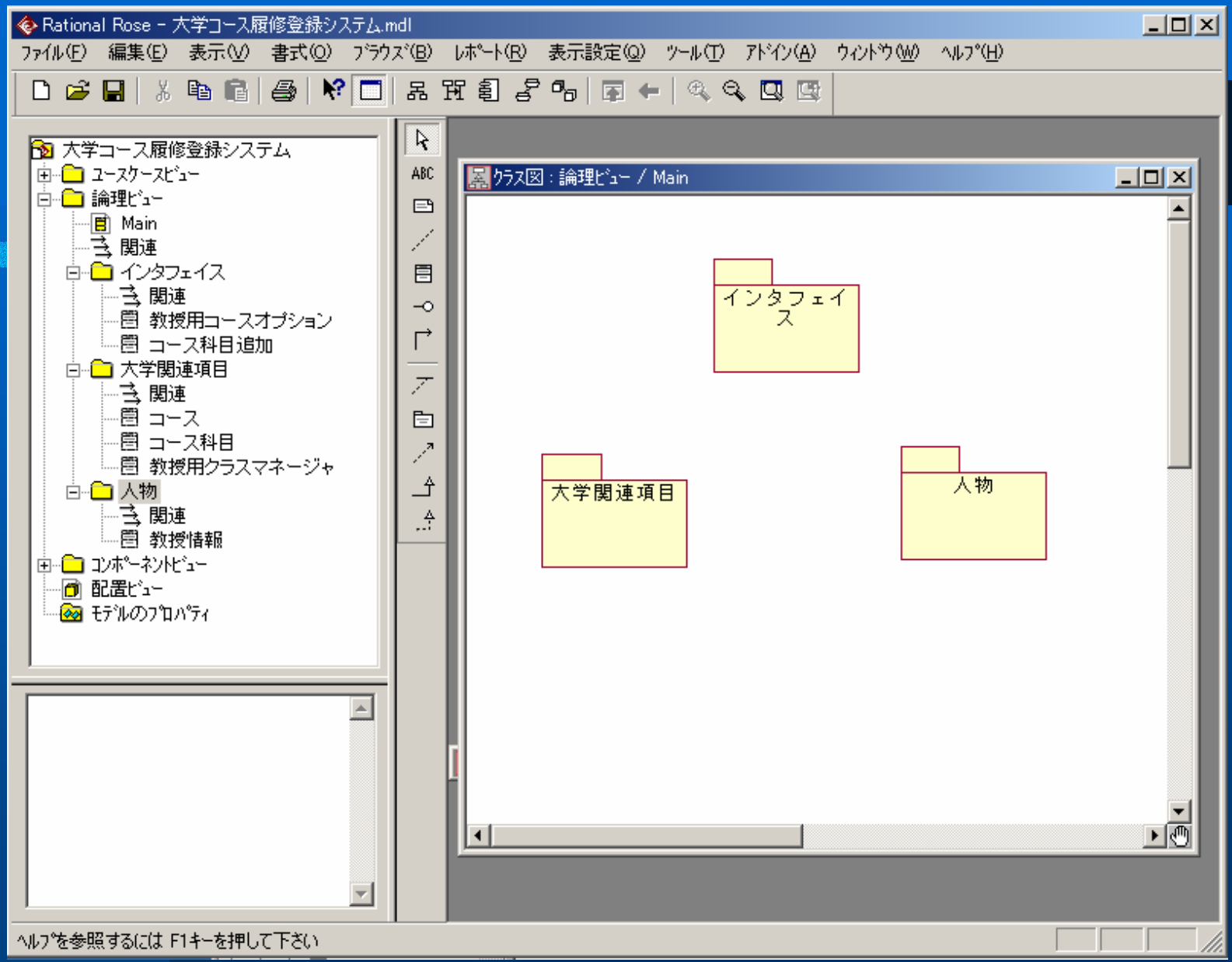
1. ブラウズで[論理ビュー]を右クリックして選択する。
2. [新規作成]をポイントし、[パッケージ]をクリックする。
3. パッケージが選択されたままの状態で、パッケージの名前を入力する。


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
Main
関連
教授用コースオプション
コース科目追加
コース
コース科目
教授情報
教授用クラスマネージャ
インタフェイス
大学関連項目
人物
コンポーネントビュー
配置ビュー
モデルのプロパティ
ABC
クラス図：論理ビュー / Main
ログ
ヘルプを参照するには F1キーを押して下さい


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
Main
関連
インタフェイス
関連
教授用コースオプション
コース科目追加
大学関連項目
関連
コース
コース科目
教授用クラスマネージャ
人物
関連
教授情報
コンポーネントビュー
配置ビュー
モデルのプロパティ
ABC
クラス図：論理ビュー / Main
ログ
ヘルプを参照するには F1キーを押して下さい

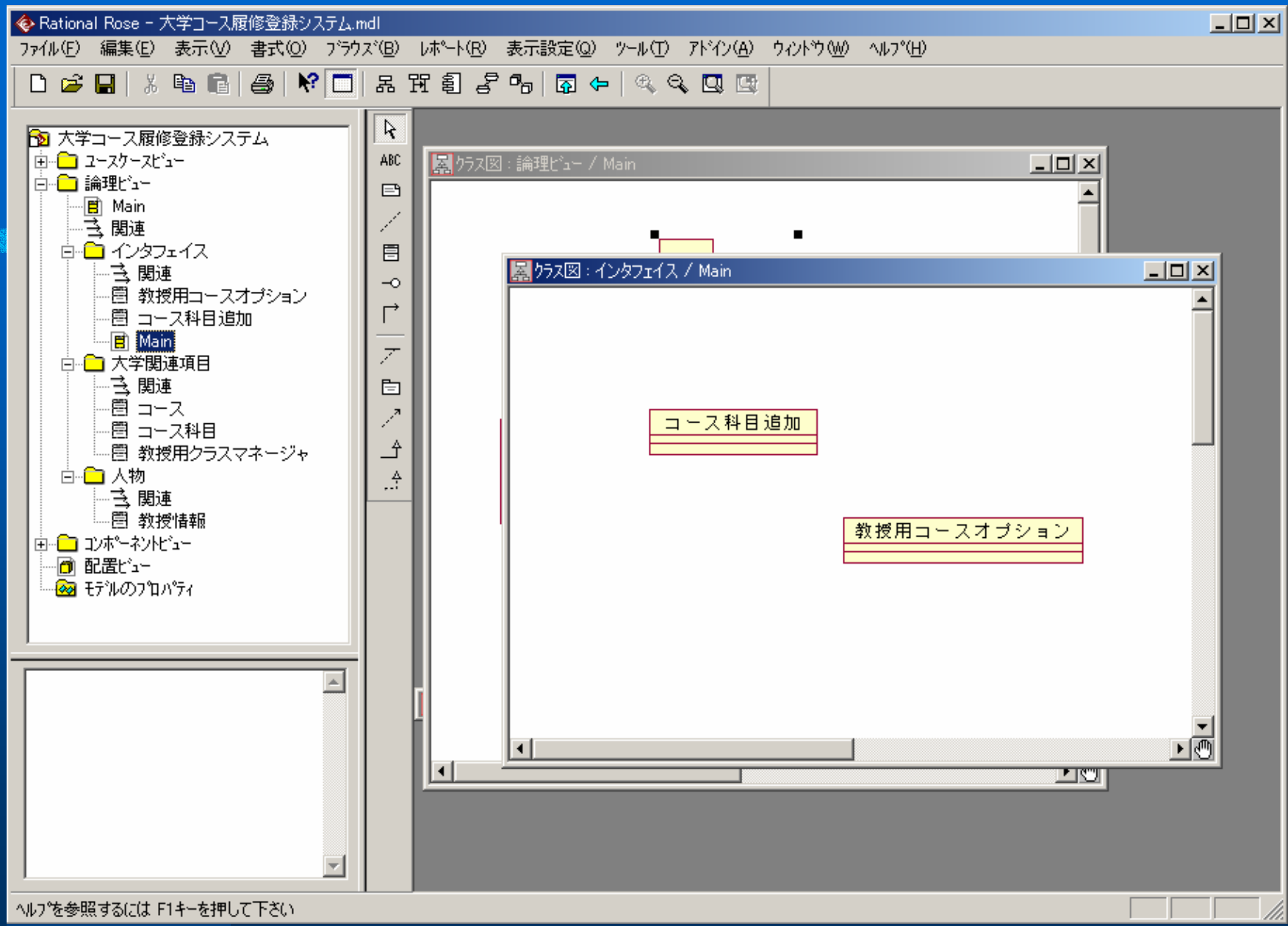
クラス識別の例:
大学コース履修登録:メインクラス図にパッケージの追加

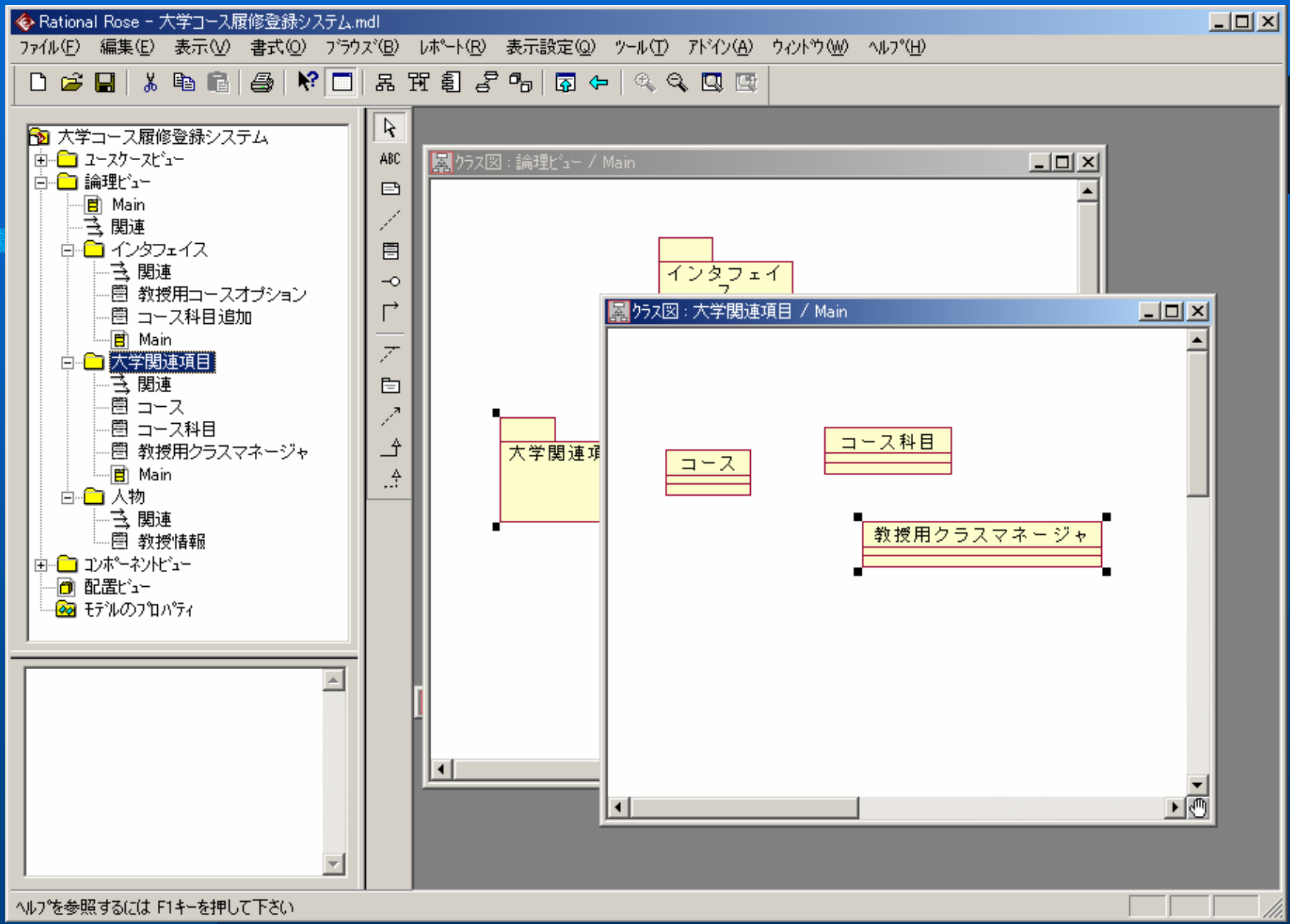
1. ブラウズでメイン図をダブルクリックして、図を開く。
2. ブラウズで、パッケージをクリックして選択する。
3. パッケージを図の上にドラックする。
4. 図に追加するパッケージごとに、上の手順を繰り返す。

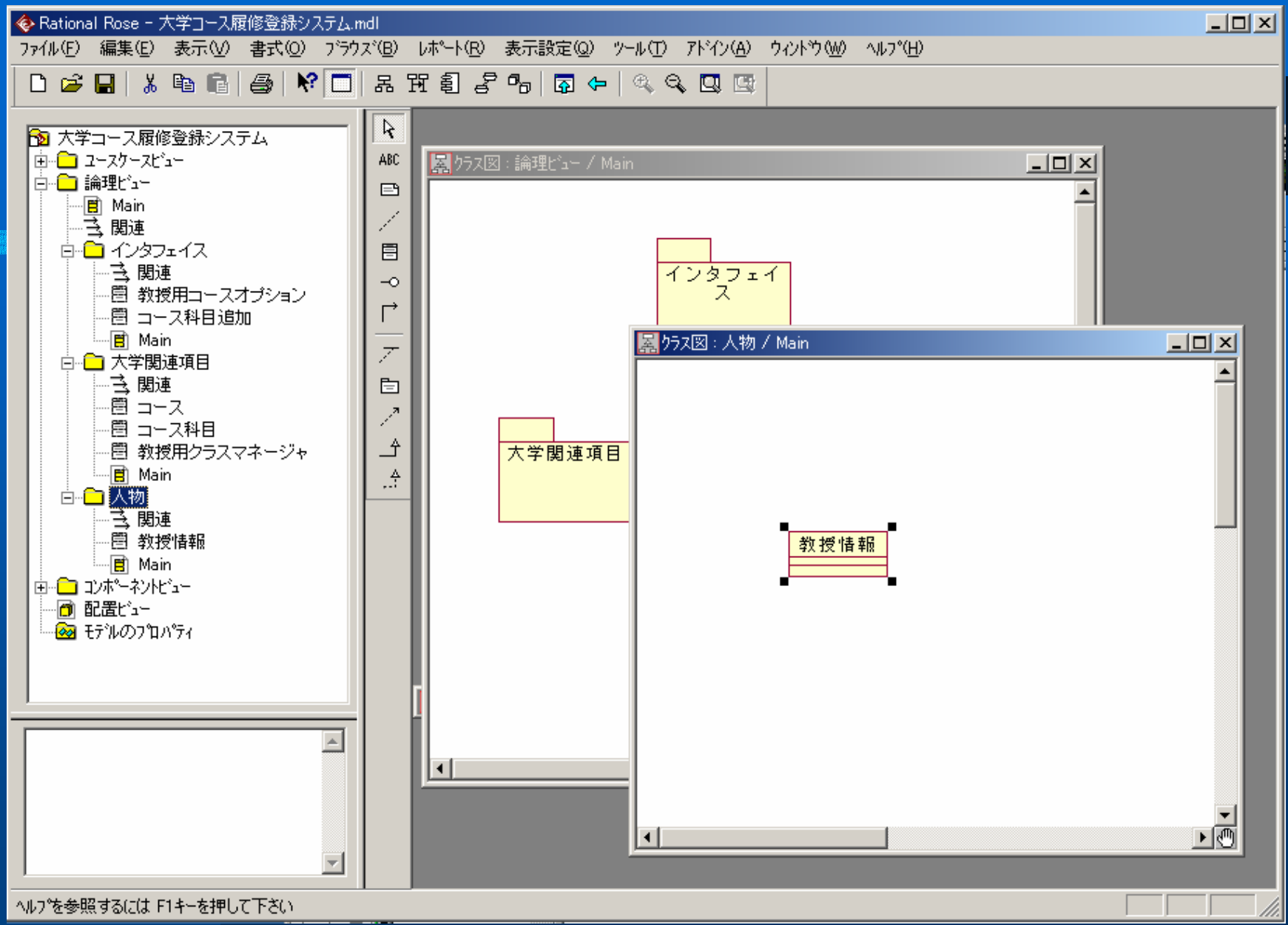

Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
Main
関連
インタフェイス
関連
教授用コースオプション
コース科目追加
大学関連項目
関連
コース
コース科目
教授用クラスマネージャ
人物
関連
教授情報
コンポーネントビュー
配置ビュー
モデルのプロパティ
ABC
クラス図：論理ビュー / Main
インタフェイス
大学関連項目
人物
ヘルプを参照するには F1キーを押して下さい

クラス識別の例：
大学コース履修登録：パッケージのメインクラス図作成

1. クラス図上に、パッケージをダブルクリックする。
2. パッケージが開かれ、そのパッケージに対するメインクラス図が作成され（表示され）る。
3. ブラウズでクラスをクリックして選択し、図の上にドラックする。
4. 図の上に配置する追加のクラスごとに、手順3を繰り返す。


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
Main
関連
インタフェイス
関連
教授用コースオプション
コース科目追加
Main
大学関連項目
関連
コース
コース科目
教授用クラスマネージャ
人物
関連
教授情報
コンポーネントビュー
配置ビュー
モデルのプロパティ
クラス図：論理ビュー / Main
クラス図：インタフェイス / Main
コース科目追加
教授用コースオプション
ヘルプを参照するには F1キーを押して下さい


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
Main
関連
インタフェイス
関連
教授用コースオプション
コース科目追加
Main
大学関連項目
関連
コース
コース科目
教授用クラスマネージャ
Main
人物
関連
教授情報
コンポーネントビュー
配置ビュー
モデルのプロパティ
クラス図 : 論理ビュー / Main
インタフェイス
大学関連項
クラス図 : 大学関連項目 / Main
コース
コース科目
教授用クラスマネージャ
ヘルプを参照するには F1キーを押して下さい


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
Main
関連
インタフェイス
関連
教授用コースオプション
コース科目追加
Main
大学関連項目
関連
コース
コース科目
教授用クラスマネージャ
Main
人物
関連
教授情報
Main
コンポーネントビュー
配置ビュー
モデルのプロパティ
ABC
クラス図：論理ビュー / Main
インタフェイス
大学関連項目
クラス図：人物 / Main
教授情報
ヘルプを参照するには F1キーを押して下さい

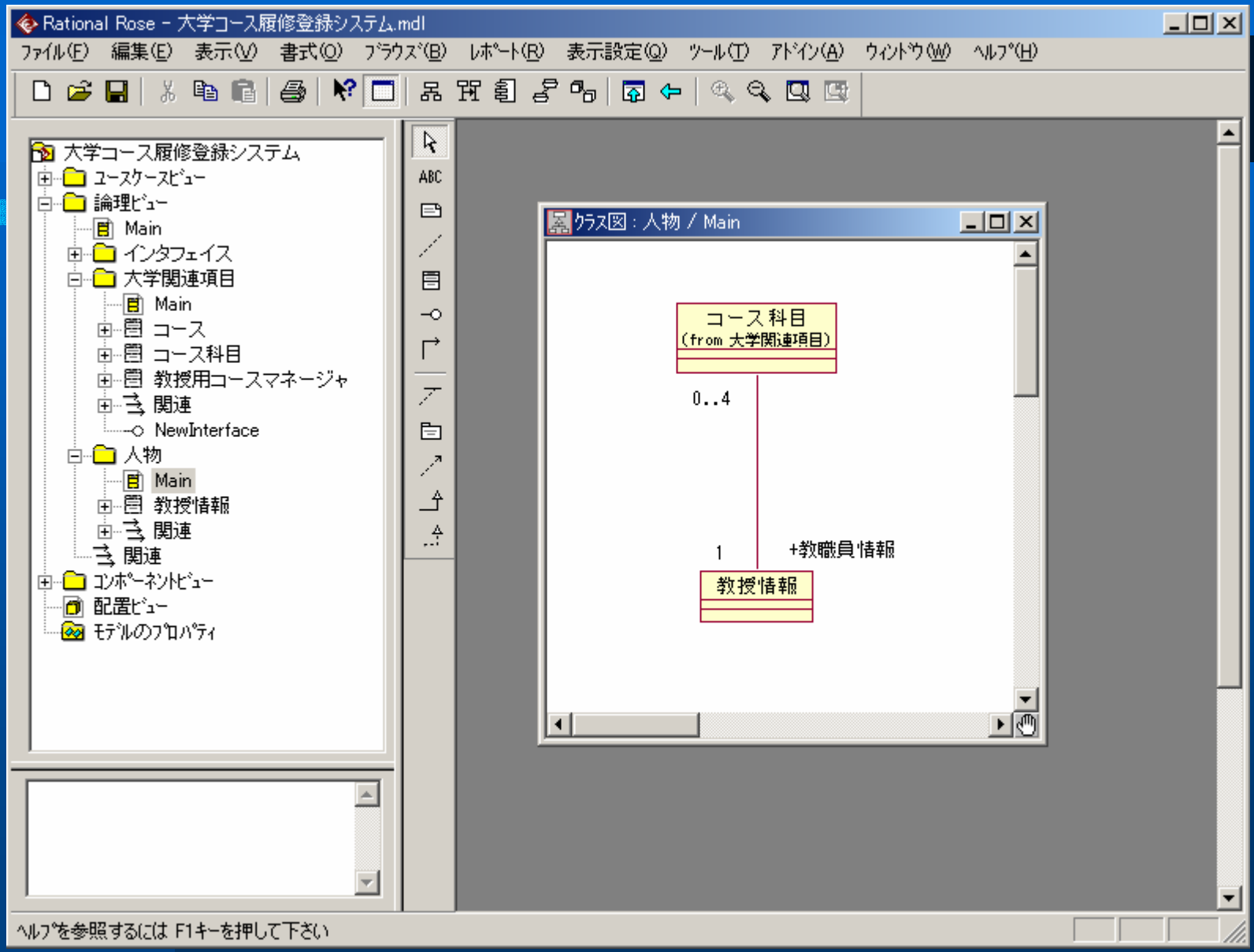
関連の識別

関連

- 関連とは、クラス間の双方向の意味的な結び付きのことである。
- クラス間の関連は、関連付けられた複数のクラス内のオブジェクト間にリンクが存在することを意味する。例えば、コースクラスと関連情報クラスとの間の関連は、コースクラスのオブジェクトが学生情報クラスのオブジェクトに結び付けられていることを意味する。
- 結び付けられるオブジェクトの数は、関連の多重度によって異なる。

関連の作成例:
大学コース履修登録: 関連の作成

1. ツールバーの[単方向の関連]又は[関連]アイコンをクリックする。
2. クラス図で、関連付けるクラスの一方をクリックする。
3. 関連付けるもう一方のクラスに、関連直線をドラッグする。

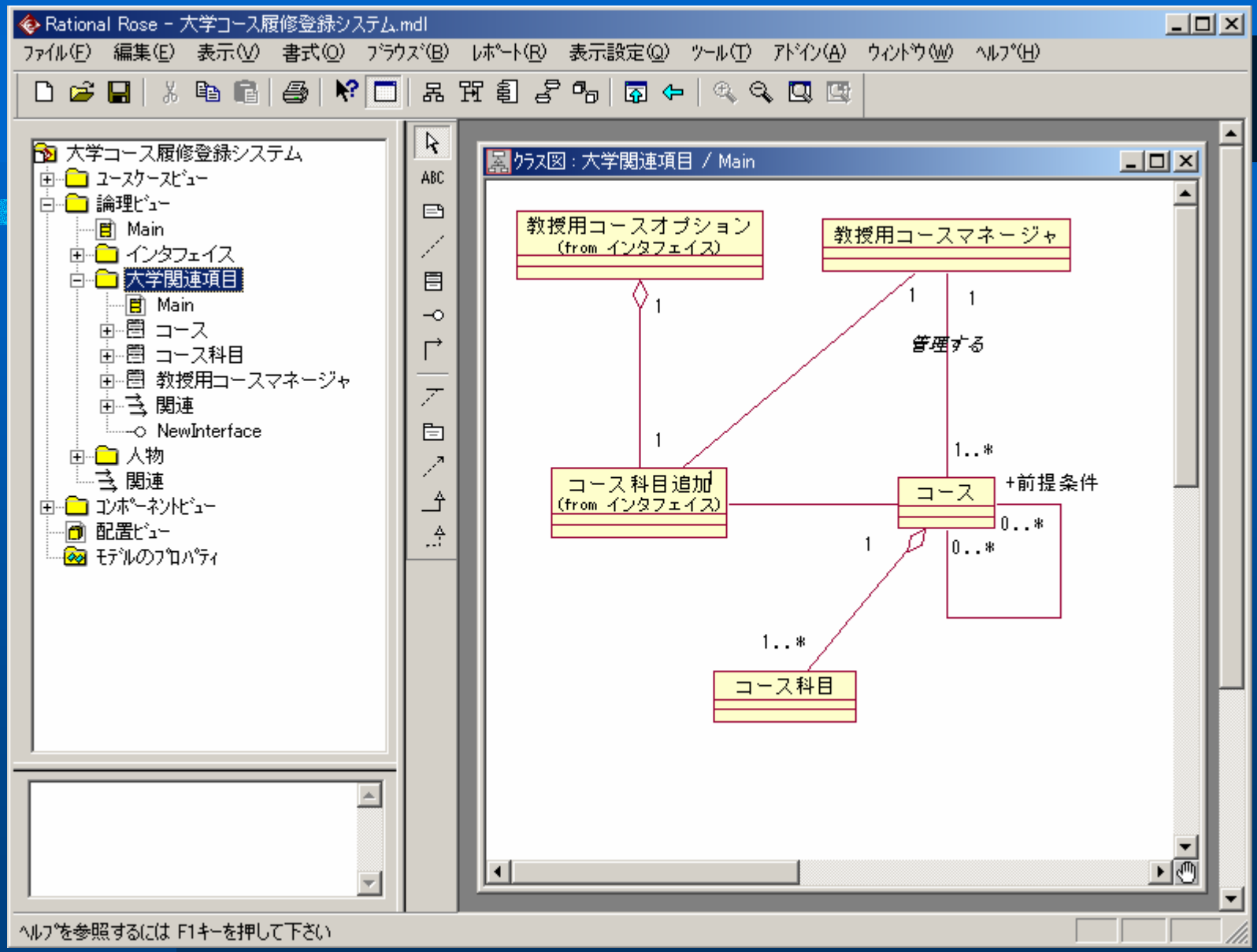

Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
Main
インタフェイス
大学関連項目
Main
コース
コース科目
教授用コースマネージャ
関連
NewInterface
人物
Main
教授情報
関連
関連
コンポーネントビュー
配置ビュー
モデルのプロパティ
ABC
クラス図：人物 / Main
コース科目
(from 大学関連項目)
0..4
1
+教職員情報
教授情報
ヘルプを参照するには F1キーを押して下さい

集約

- 集約は関連の特別な形で、全体と部分の間の関連である。
- 集約は“part-of”関係または包含関係として知られている。
- 例えば、1つのコース（数学1）が学期中の異なる時限に提供されることがある。それぞれの科目はコース科目（数学1の第1部や数学1第2部）として表される。コースとコース科目の間の関係は集約である。つまり、コースがコース科目を「持つ」ものとしてモデル化される。

関連の作成例：
大学コース履修登録： 集約の作成

1. ツールバーの[単方向の関連]又は[集約]アイコンをクリックする。
2. クラス図で「部分」の役割を果たすクラスをクリックし、「全体」の役割を果たすクラスに集約直線をドラッグする。

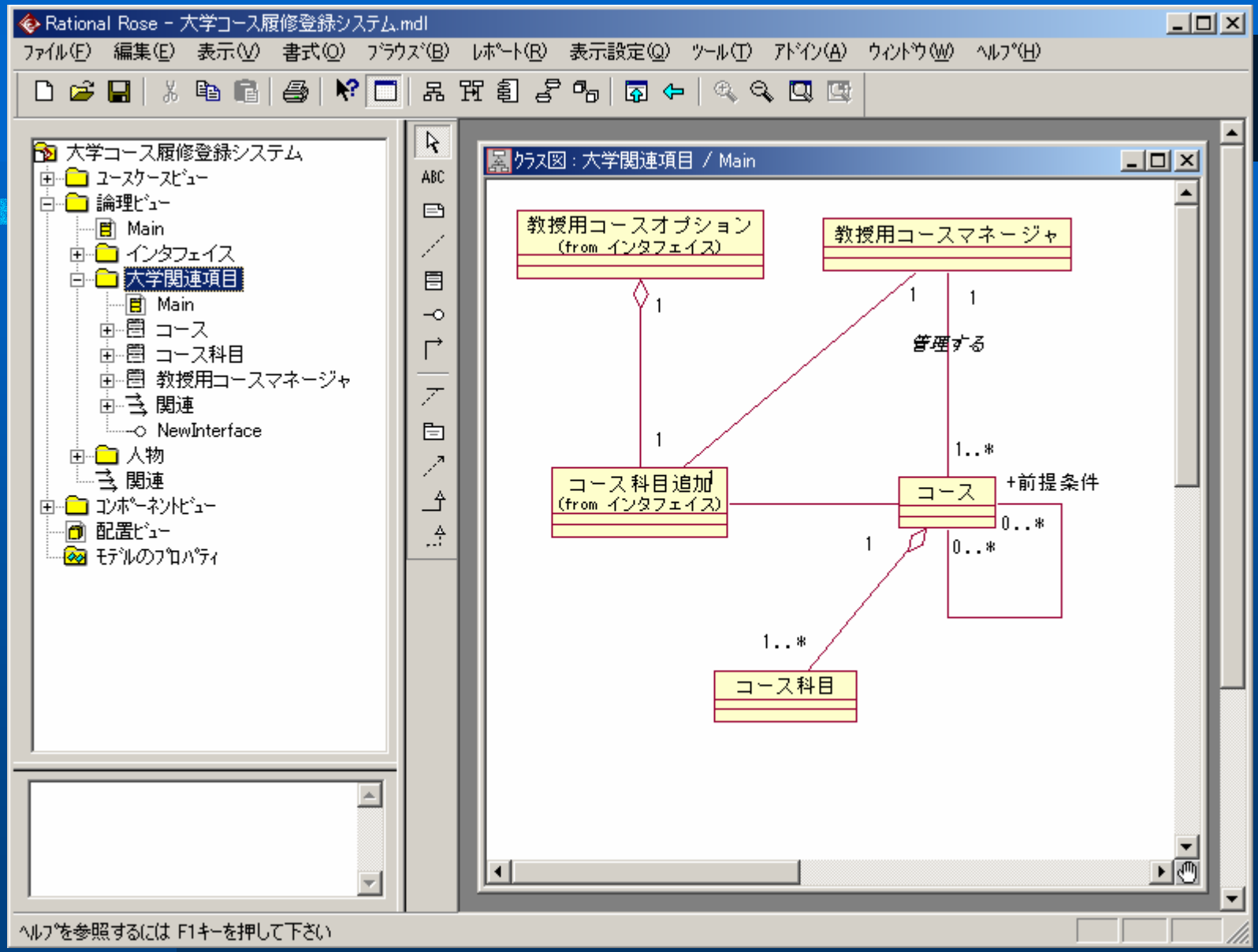

Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
Main
インタフェイス
大学関連項目
Main
コース
コース科目
教授用コースマネージャ
関連
NewInterface
人物
関連
コンポーネントビュー
配置ビュー
モデルのプロパティ
クラス図 : 大学関連項目 / Main
教授用コースオプション
(from インタフェイス)
教授用コースマネージャ
管理する
コース科目追加
(from インタフェイス)
コース
+前提条件
コース科目
1
1..*
0..*
ヘルプを参照するには F1キーを押して下さい

関連の命名

- 関連には名前が付けられることがある。
- 通常その名前は、関係の意味を表す能動態の動詞または動詞句にする。
- 動詞句は一般に読方向を伴うため、左から右へまたは上から下へと関係を正しく読めるように関連を命名するのが理想的である。

関連の作成例:
大学コース履修登録: 関連命名の作成

1. クラス図で、関係直線をクリックして選択する。
2. 関係の名前を入力する。

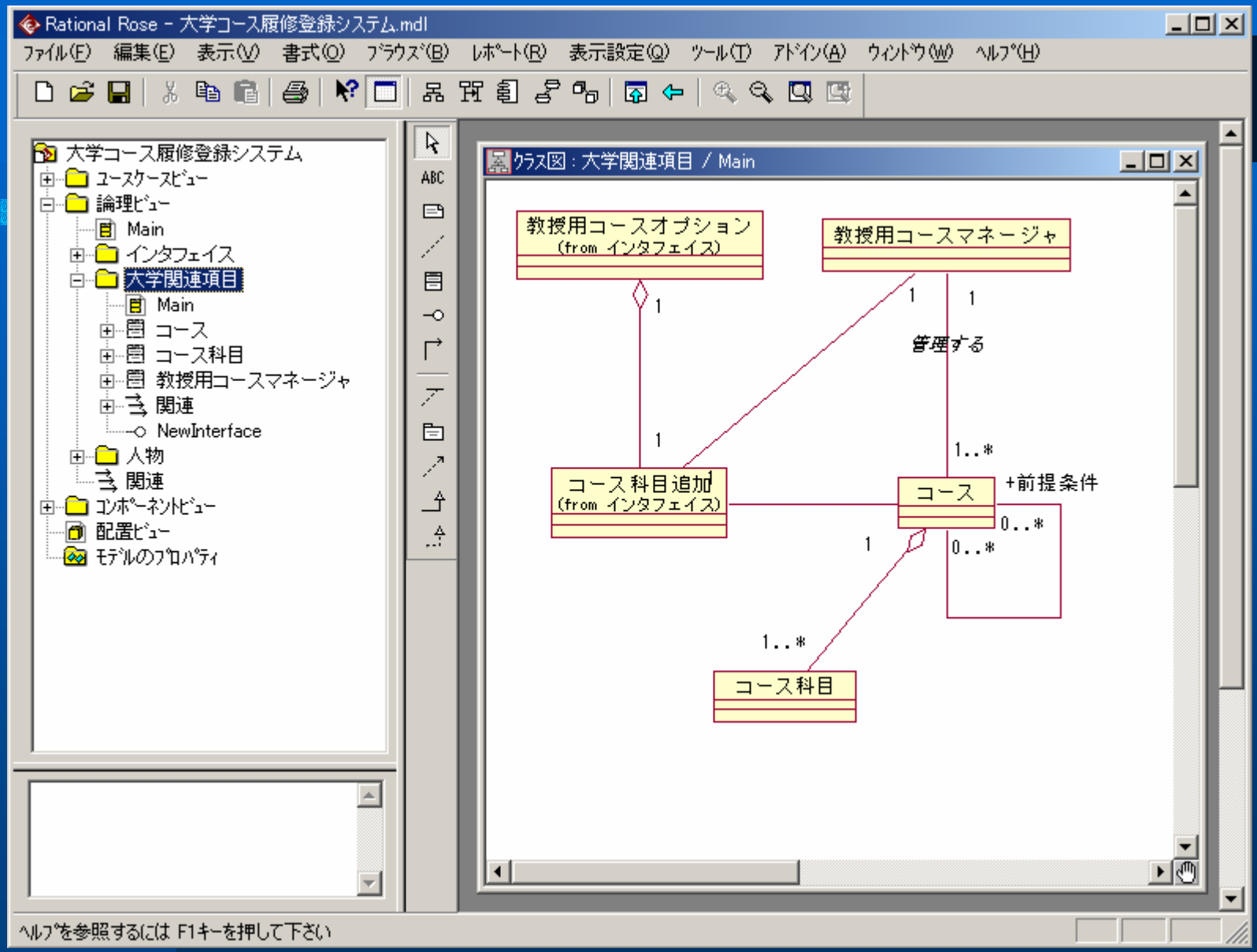

Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
Main
インタフェイス
大学関連項目
Main
コース
コース科目
教授用コースマネージャ
関連
NewInterface
人物
関連
コンポーネントビュー
配置ビュー
モデルのプロパティ
クラス図 : 大学関連項目 / Main
教授用コースオプション
(from インタフェイス)
教授用コースマネージャ
管理する
コース科目追加
(from インタフェイス)
コース
+前提条件
コース科目
1
1..*
0..*
ヘルプを参照するには F1キーを押して下さい

ロール名

- クラスに結び付いた関連の端末をロールと呼ぶ。
- 関連名の代わりにロール名を使うことができる。
- ロール名は、あるクラスが別のクラスと関連する目的や役割を示す名詞である。
- ロール名は、それが修飾する方のクラスの近くに配置する。
- ロール名は関連直線の一方または両方の端に配置できる。
- ロール名と関連名と関連名の両方を付ける必要がない。

関連の作成例:
大学コース履修登録: ロール名の作成

1. 関連直線上の、ロール名が修飾するクラスの近くに右クリックして、ショートカットメニューを表示する。
2. [ロール名]をクリックする。
3. ロール名を入力する。


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
Main
インタフェイス
大学関連項目
Main
コース
コース科目
教授用コースマネージャ
関連
NewInterface
人物
関連
コンポーネントビュー
配置ビュー
モデルのプロパティ
クラス図 : 大学関連項目 / Main
教授用コースオプション
(from インタフェイス)
教授用コースマネージャ
管理する
コース科目追加
(from インタフェイス)
コース
+前提条件
コース科目
1
1..*
0..*
ヘルプを参照するには F1キーを押して下さい

多重度表示

- 多重度はクラスに対して指定され、ある関係に参加するオブジェクトの数を定義する。
- 多重度は、互いにリンクされるオブジェクトの数を定義する。
- 一般の多重度表示を次に示す。
  - 1　　厳密に1
  - 0..*　　0以上
  - 1..*　　1以上
  - 0..1　　0または1
  - 5..8　　指定した範囲(5,6,7, または8)
  - 4..7,9　　組み合わせ (4,5,6,7, または9)

関連の作成例:
大学コース履修登録: 多重度の作成

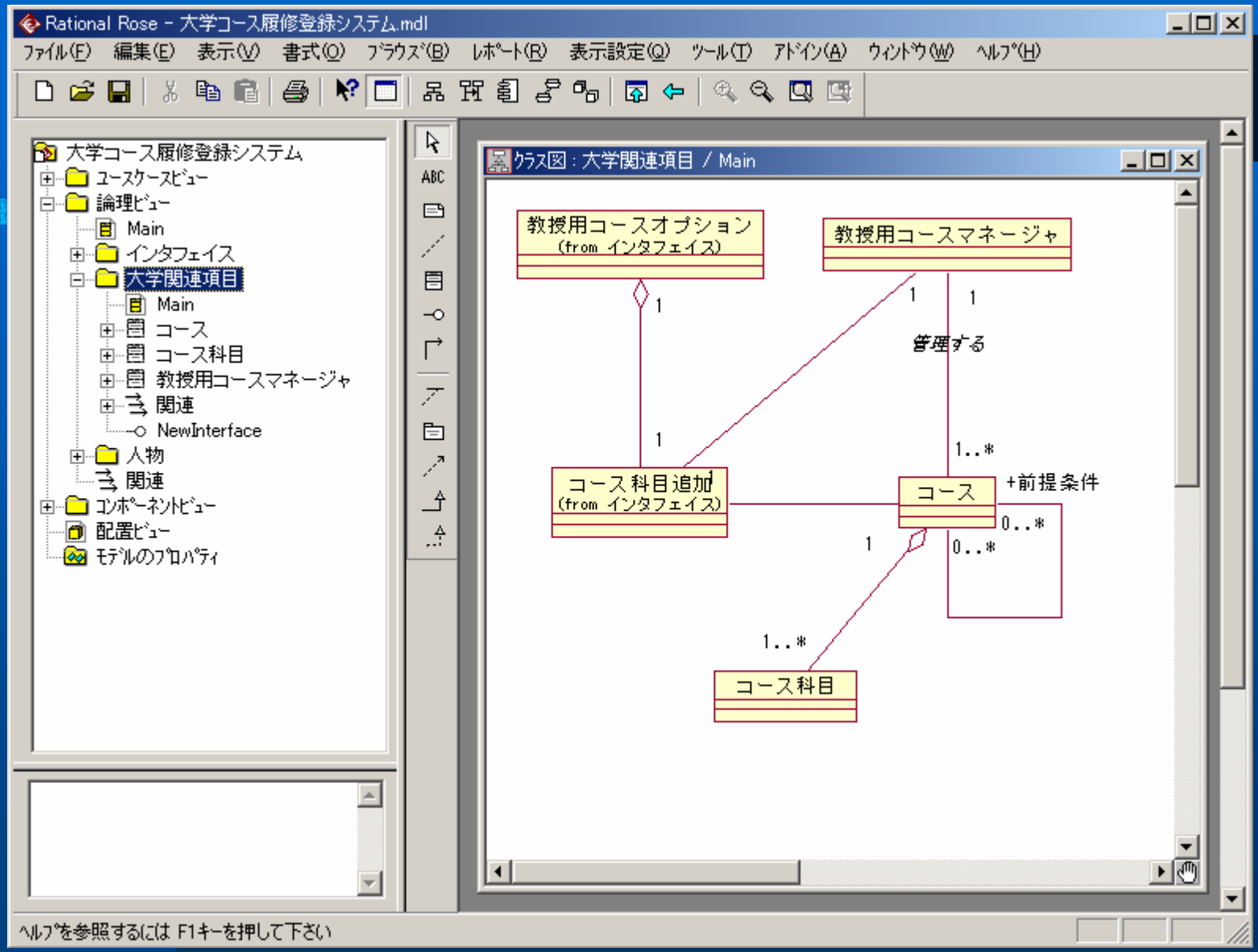
1. 関係直線をダブルクリックして、仕様を表示する。
2. 修飾されえちるロールの[詳細]タブをクリックする([ロールAの詳細]または[ロールBの詳細]タブ)。
3. 多重度を入力する。

再起的な関係

- 同じクラスに属している複数のオブジェクトがメッセージを授受しなければならない。
- これはクラス図では、再起的な関連または集約として示される。
- 再起的な関係については、通常関連名ではなくロール名が使われる。

関連の作成例:
大学コース履修登録: 再起的な関係の作成

1. ツールバーの[単方向関連]又は[関連]アイコンをクリックする。
2. クラスをクリックし、関連(または集約)直線をクラスの外側にドラッグする。
3. マウスボタンを放す。
4. 関連(または集約)直線をクリックし、元のクラスに再びドラッグする。
5. 再起的な関連(または集約)のそれぞれの端に、ロール名と多重度を入力する。

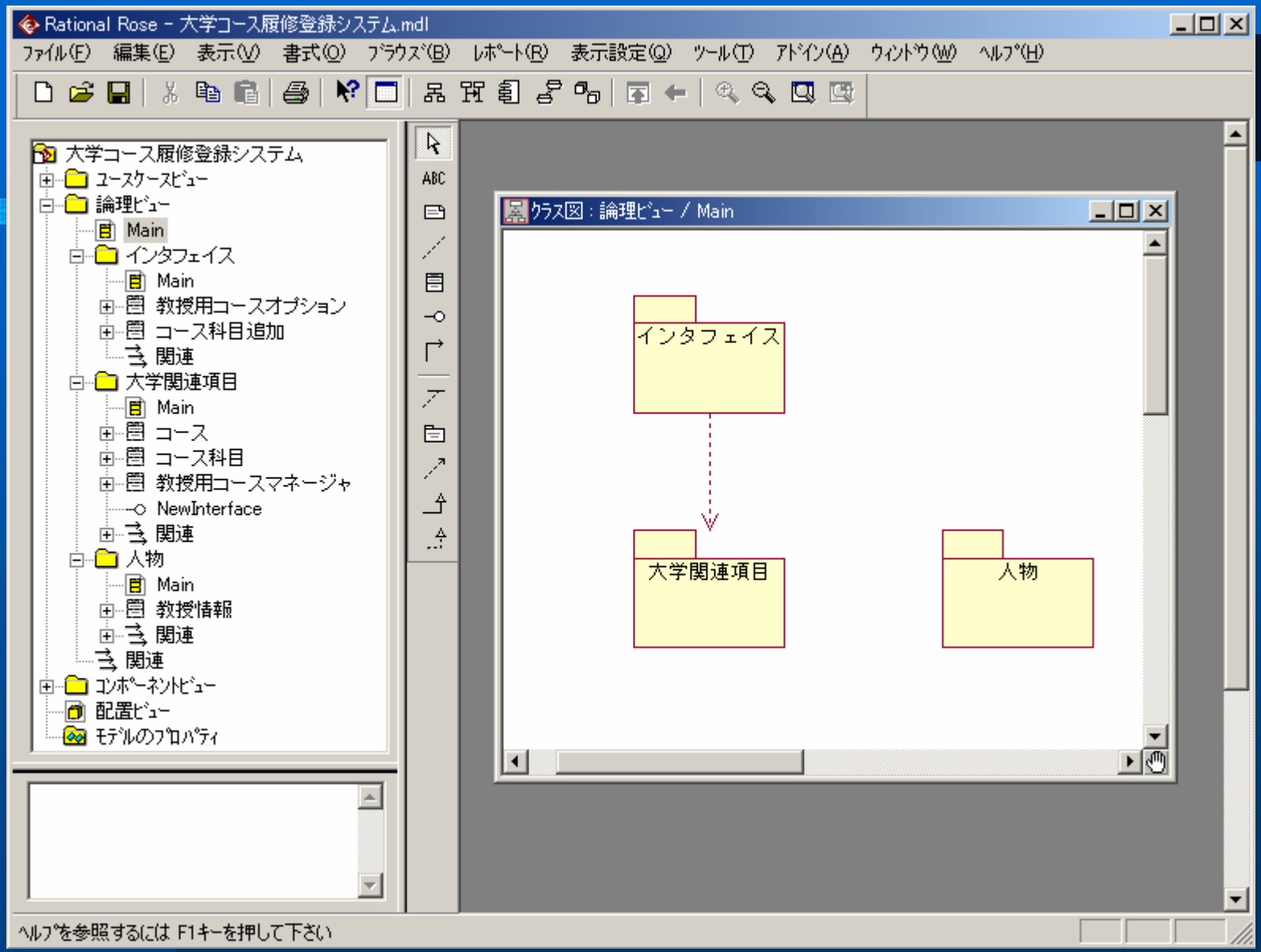

Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
Main
インタフェイス
大学関連項目
Main
コース
コース科目
教授用コースマネージャ
関連
NewInterface
人物
関連
コンポーネントビュー
配置ビュー
モデルのプロパティ
クラス図 : 大学関連項目 / Main
教授用コースオプション
(from インタフェイス)
教授用コースマネージャ
管理する
コース科目追加
(from インタフェイス)
コース
+前提条件
コース科目
1
1..*
0..*
ヘルプを参照するには F1キーを押して下さい

パッケージ関係

- モデルには、パッケージ関係も追加される。
- この種の関係は依存関係であり、依存しているパッケージへの破線の矢印で示される。
- パッケージAがパッケージBに依存している場合、これはパッケージAに含まれる1つ以上のクラスが、パッケージBに含まれる1つ以上の公開クラスの操作を呼ぶことを意味する。このとき、パッケージAをクライアントパッケージ、パッケージBをサプライヤパッケージと呼ぶ。

関連の作成例:
大学コース履修登録: パッケージ関係の作成

1. ツールバーの[依存関係]アイコンをクリックする。
2. 依存する側のパッケージをクリックし、依存される側のパッケージに矢印をドラッグする。


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
Main
インタフェイス
Main
教授用コースオプション
コース科目追加
関連
大学関連項目
Main
コース
コース科目
教授用コースマネージャ
NewInterface
関連
人物
Main
教授情報
関連
関連
コンポーネントビュー
配置ビュー
モデルのプロパティ
ABC
クラス図 : 論理ビュー / Main
インタフェイス
大学関連項目
人物
ヘルプを参照するには F1キーを押して下さい

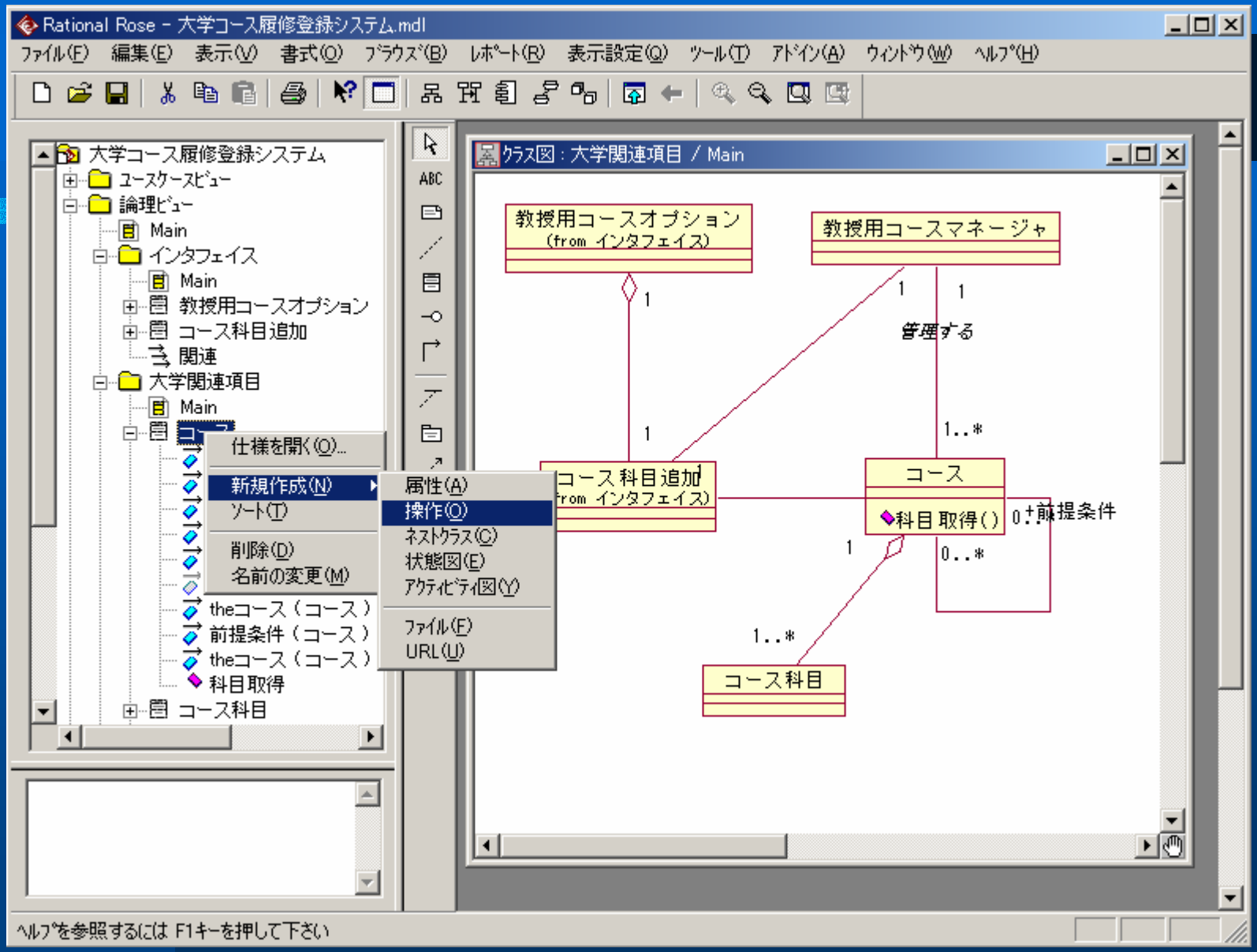
振る舞いと属性の追加

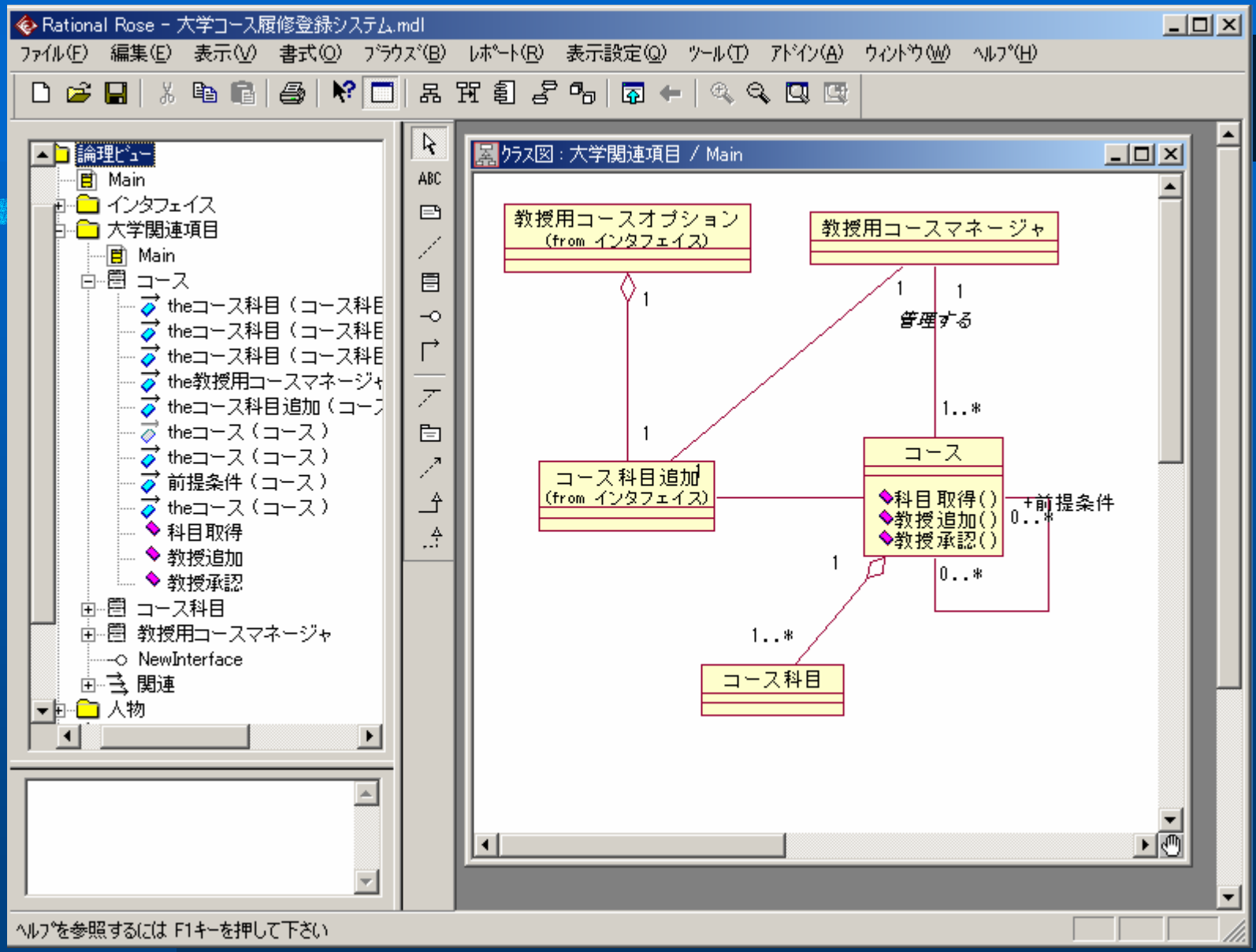
振る舞いと属性

- オブジェクトの構造は、クラスの属性によって規定される。それぞれの属性は、クラスのオブジェクトによって保持されるデータ構造である。クラスに対して定義されたオブジェクトは、そのクラスのすべての属性に対する値を持つ。
- クラスに関していえば、コードの可読性と保守性を向上させるために、スタイルガイドに従って属性と振る舞い（操作）を定義することは大事なことである。

振る舞い(操作)の作成

1. ブラウズでクラスを右クリックし、ショートカットメニューを表示する。
2. [新規作成]をポイントし、[操作]をクリックする。“opname”という名前の操作がブラウズに作成される。
3. 新しい操作が選択されたままの状態で、希望する名前を入力する。


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
Main
インタフェイス
Main
教授用コースオプション
コース科目追加
関連
大学関連項目
Main
仕様を開く(O)...
新規作成(N)
ソート(T)
削除(D)
名前の変更(M)
属性(A)
操作(O)
ネストクラス(C)
状態図(E)
アクティビティ図(Y)
ファイル(F)
URL(U)
theコース（コース）
前提条件（コース）
theコース（コース）
科目取得
コース科目
クラス図：大学関連項目 / Main
教授用コースオプション
(from インタフェイス)
教授用コースマネージャ
管理する
コース科目追加
(from インタフェイス)
コース
科目取得()
前提条件
コース科目
1
1..*
0..*
ヘルプを参照するには F1キーを押して下さい

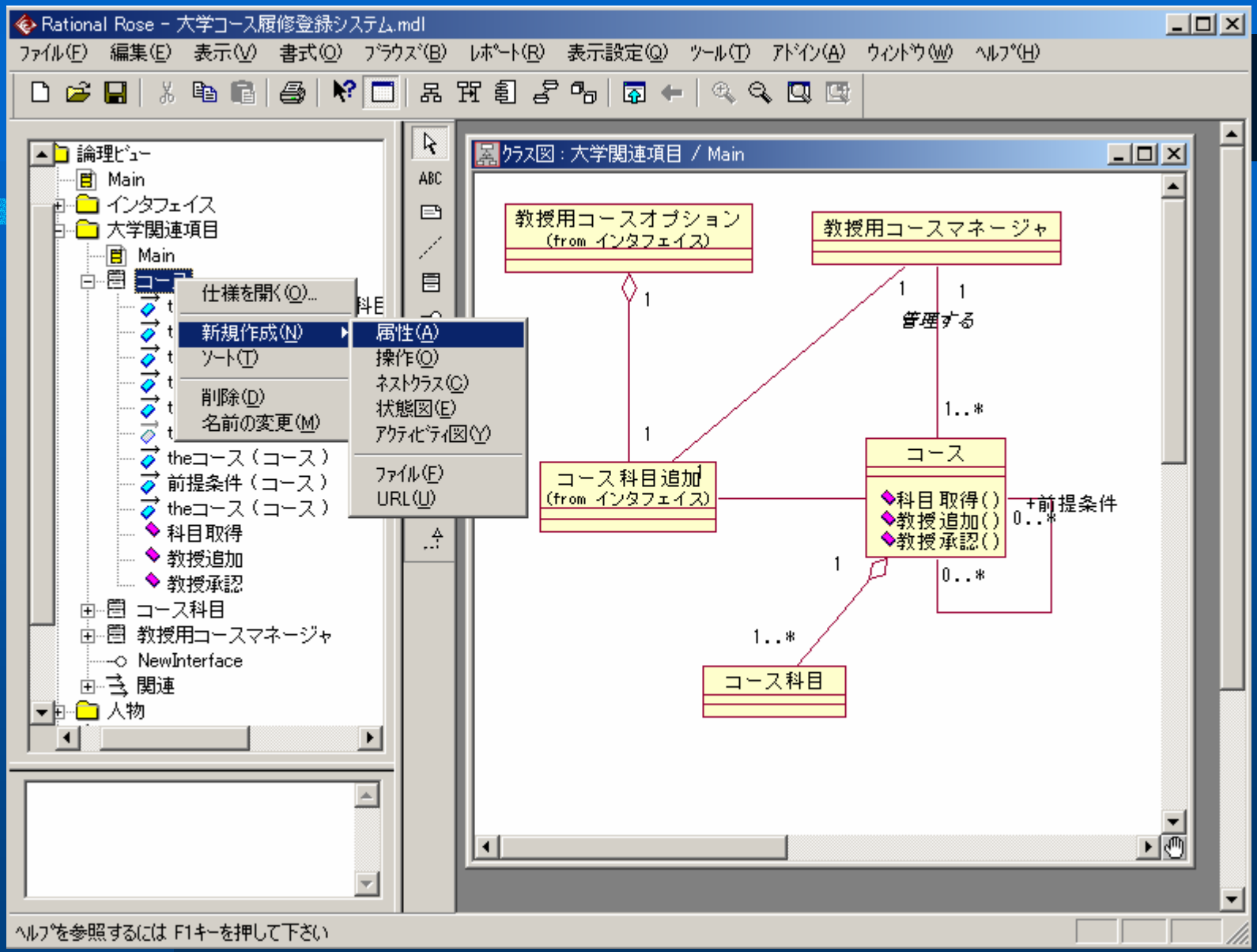

Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
論理ビュー
Main
インタフェイス
大学関連項目
Main
コース
theコース科目（コース科目
theコース科目（コース科目
theコース科目（コース科目
the教授用コースマネージャ
theコース科目追加（コース
theコース（コース）
theコース（コース）
前提条件（コース）
theコース（コース）
科目取得
教授追加
教授承認
コース科目
教授用コースマネージャ
NewInterface
関連
人物
クラス図：大学関連項目 / Main
教授用コースオプション
(from インタフェイス)
教授用コースマネージャ
1
1
管理する
1
1..*
コース科目追加
(from インタフェイス)
コース
科目取得()
教授追加()
教授承認()
+前提条件
0..*
1
0..*
1..*
コース科目
ヘルプを参照するには F1キーを押して下さい

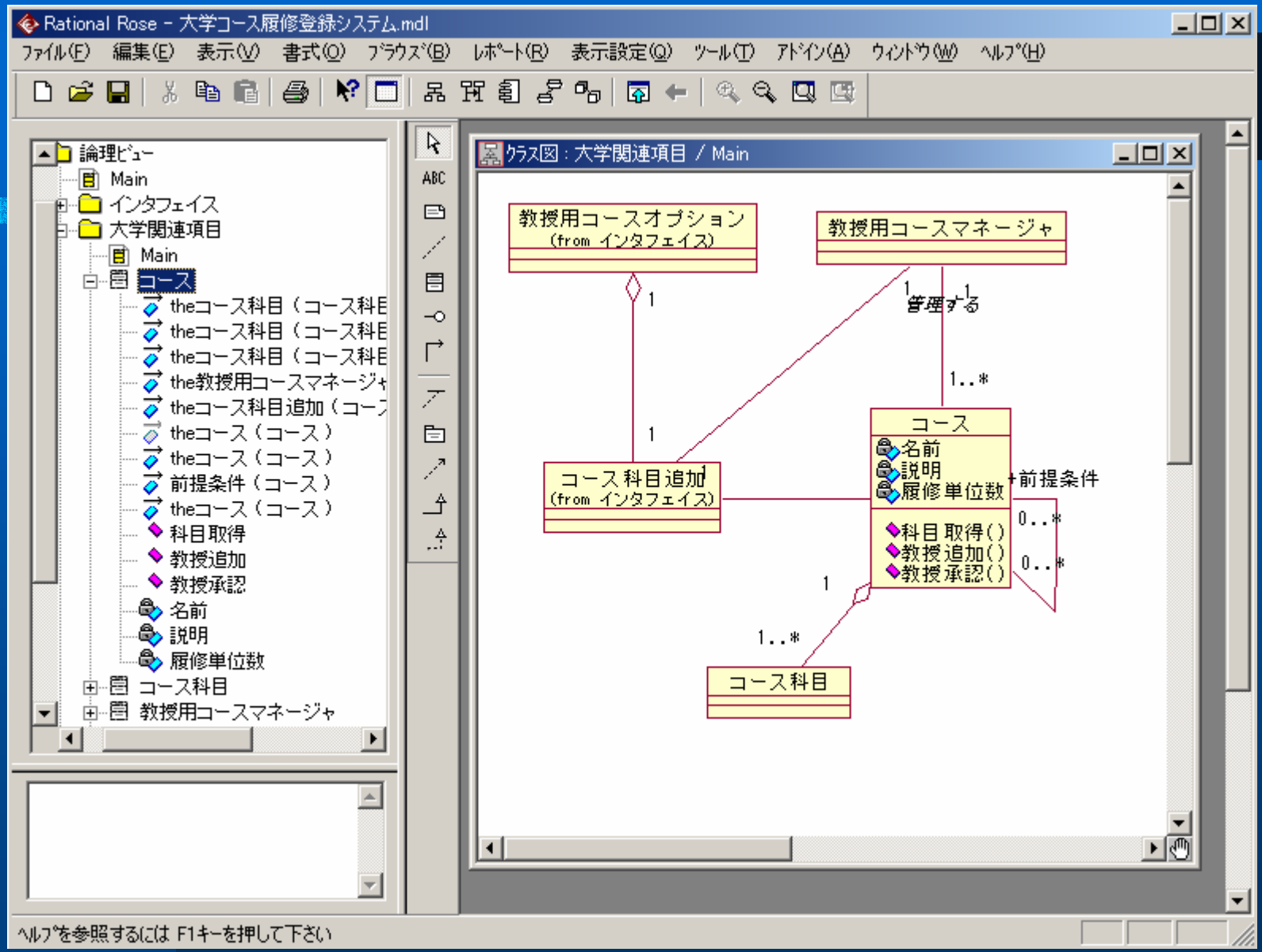
振る舞い(操作)定義の作成

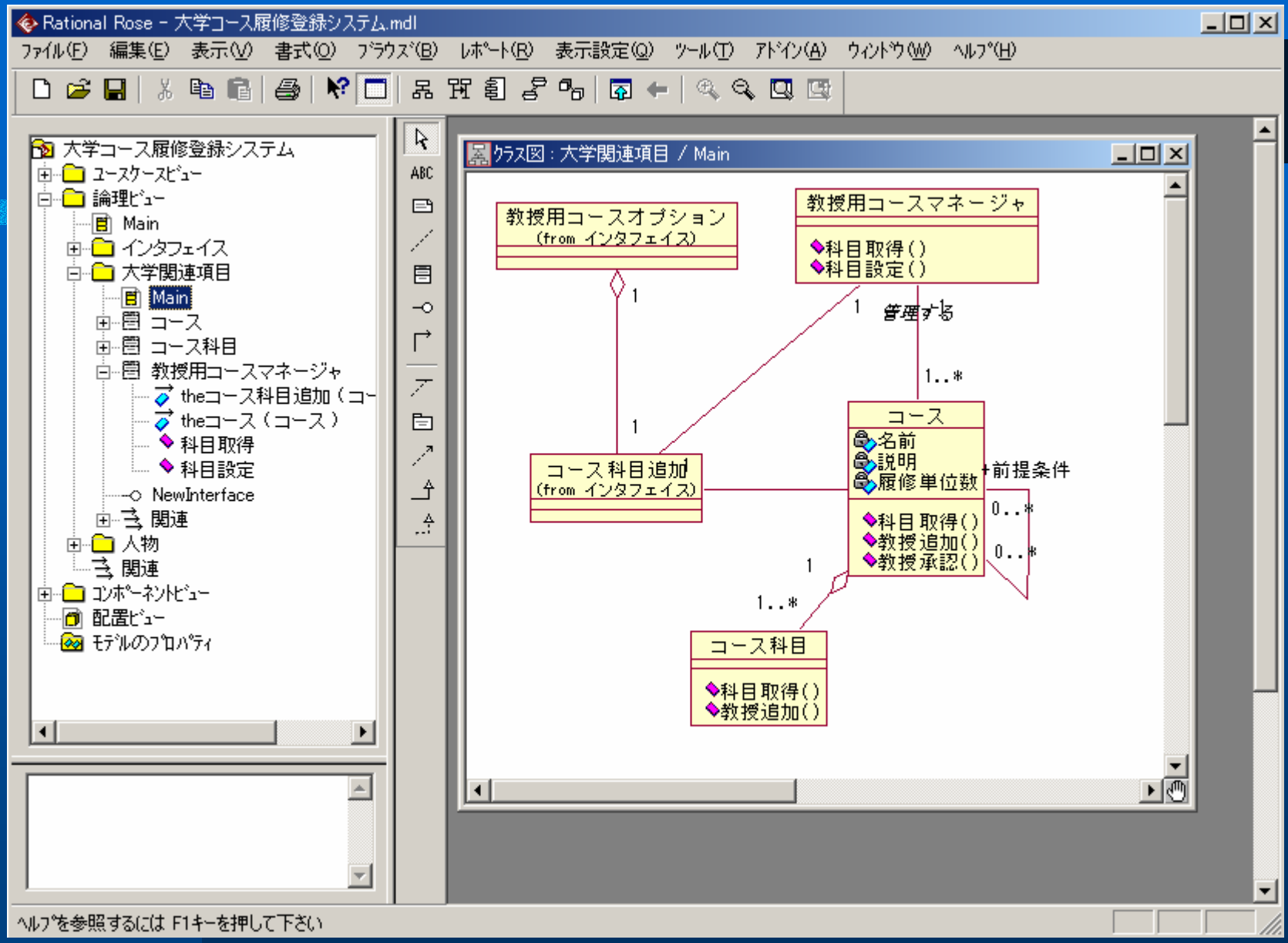
1. ブラウズで、クラスの左の“+”をクリックして、クラスを展開する。
2. 操作をクリックして選択する。
3. 定義ウインドウにカーソルを置き、操作の定義を入力する。

属性の作成

1. ブラウズでクラスを右クリックして、ショートカットメニューを表示する。
2. [新規作成]をポイントし、[属性]をクリックする。“Name”という名前の属性がブラウズに作成される。
3. 新しい属性が選択されたままの状態で、希望の名前を入力する。


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
論理ビュー
Main
インタフェイス
大学関連項目
Main
仕様を開く(O)...
新規作成(N)
ソート(T)
削除(D)
名前の変更(M)
属性(A)
操作(O)
ネストクラス(C)
状態図(E)
アクティビティ図(Y)
ファイル(F)
URL(U)
theコース（コース）
前提条件（コース）
theコース（コース）
科目取得
教授追加
教授承認
コース科目
教授用コースマネージャ
NewInterface
関連
人物
クラス図：大学関連項目 / Main
教授用コースオプション
(from インタフェイス)
教授用コースマネージャ
管理する
コース科目追加
(from インタフェイス)
コース
科目取得()
教授追加()
教授承認()
+前提条件
コース科目
1
1..*
0..*
ヘルプを参照するには F1キーを押して下さい


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
論理ビュー
Main
インタフェイス
大学関連項目
Main
コース
theコース科目（コース科目
theコース科目（コース科目
theコース科目（コース科目
the教授用コースマネージャ
theコース科目追加（コース
theコース（コース）
theコース（コース）
前提条件（コース）
theコース（コース）
科目取得
教授追加
教授承認
名前
説明
履修単位数
コース科目
教授用コースマネージャ
クラス図：大学関連項目 / Main
教授用コースオプション
(from インタフェイス)
教授用コースマネージャ
管理する
コース科目追加
(from インタフェイス)
コース
名前
説明
履修単位数
科目取得()
教授追加()
教授承認()
前提条件
コース科目
1
1..*
0..*
ヘルプを参照するには F1キーを押して下さい


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
Main
インタフェイス
大学関連項目
Main
コース
コース科目
教授用コースマネージャ
theコース科目追加（コー
theコース（コース）
科目取得
科目設定
NewInterface
関連
人物
関連
コンポーネントビュー
配置ビュー
モデルのプロパティ
クラス図：大学関連項目 / Main
教授用コースオプション
(from インタフェイス)
教授用コースマネージャ
科目取得()
科目設定()
1
1
管理する
1..*
1
コース
名前
説明
履修単位数
+前提条件
コース科目追加
(from インタフェイス)
科目取得()
教授追加()
教授承認()
0..*
0..*
1
1..*
コース科目
科目取得()
教授追加()
ヘルプを参照するには F1キーを押して下さい

属性定義の作成

1. ブラウズで、クラスの左の“+”をクリックして、クラスを展開する。
2. 属性をクリックして選択する。
3. 定義ウインドウにカーソルを置いて、属性の定義を入力する。

継承の識別

継承

- 継承とは、1つまたは複数のクラスが1つのクラス構造や振る舞いを共通しているような、クラス間の関係を定義する概念である。継承は、is-a階層またはkind-of階層と呼ばれる場合もある。
- サブクラスが1つまたは複数のスーパークラスを継承する形で抽象化の階層が作成される。
- サブクラスはスーパークラスで定義されている属性、操作及び関連のすべてを継承している。
- 継承を識別するためには、汎化及び特殊化の2通りの方法がある。

汎化

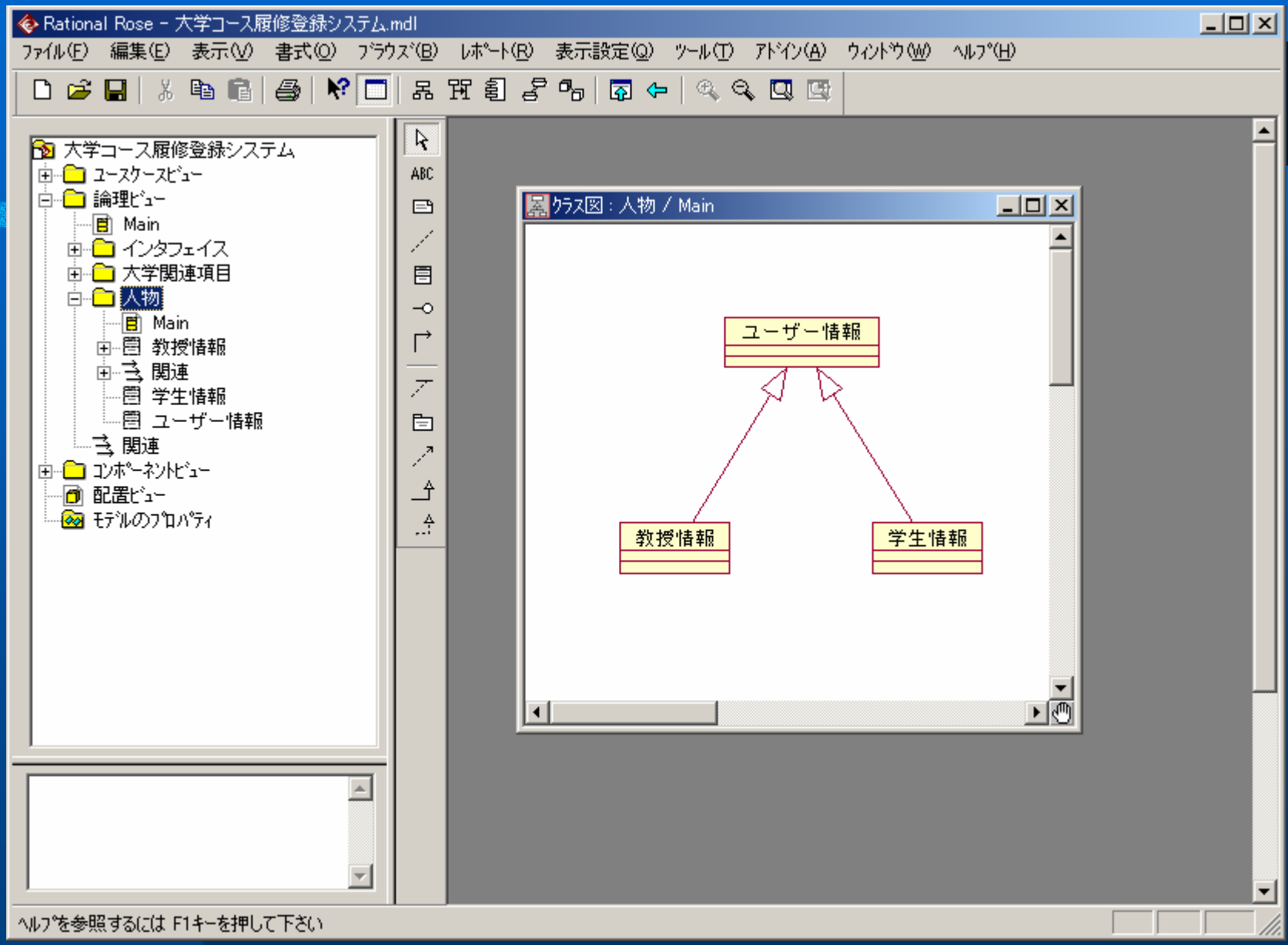
- 汎化とは、複数のクラスに共通な属性や振る舞いをカプセル化したスーパークラスを作ることをいう。
- 例えば、学生情報クラス及び教授情報クラスはどちらも、名前、住所、電話番号を属性として持つので、ユーザ情報クラスの継承と定義できる。

特殊化

- 特殊化とは、サブクラスを作ることであり、スーパークラスの属性や振る舞いを詳細化させたものである。
- 通常、サブクラスには、新たな属性や振る舞いが追加される。
- サブクラスで振る舞いをオーバーライドすることもできる（多相性（polymorphism））。

継承の作成

1. 継承階層を作成するクラス図を開く。
2. ツールバーの[クラス]をクリックし、クラス図をクリックしてクラスを描画する。
3. クラスが選択されたままの状態で、クラス名を入力する。
4. ツールバーの[汎化]をクリックする。
5. サブクラスの1つをクリックし、汎化直線をスーパークラスにドラッグする。
6. 各サブクラスごとに、手順5を繰り返す。

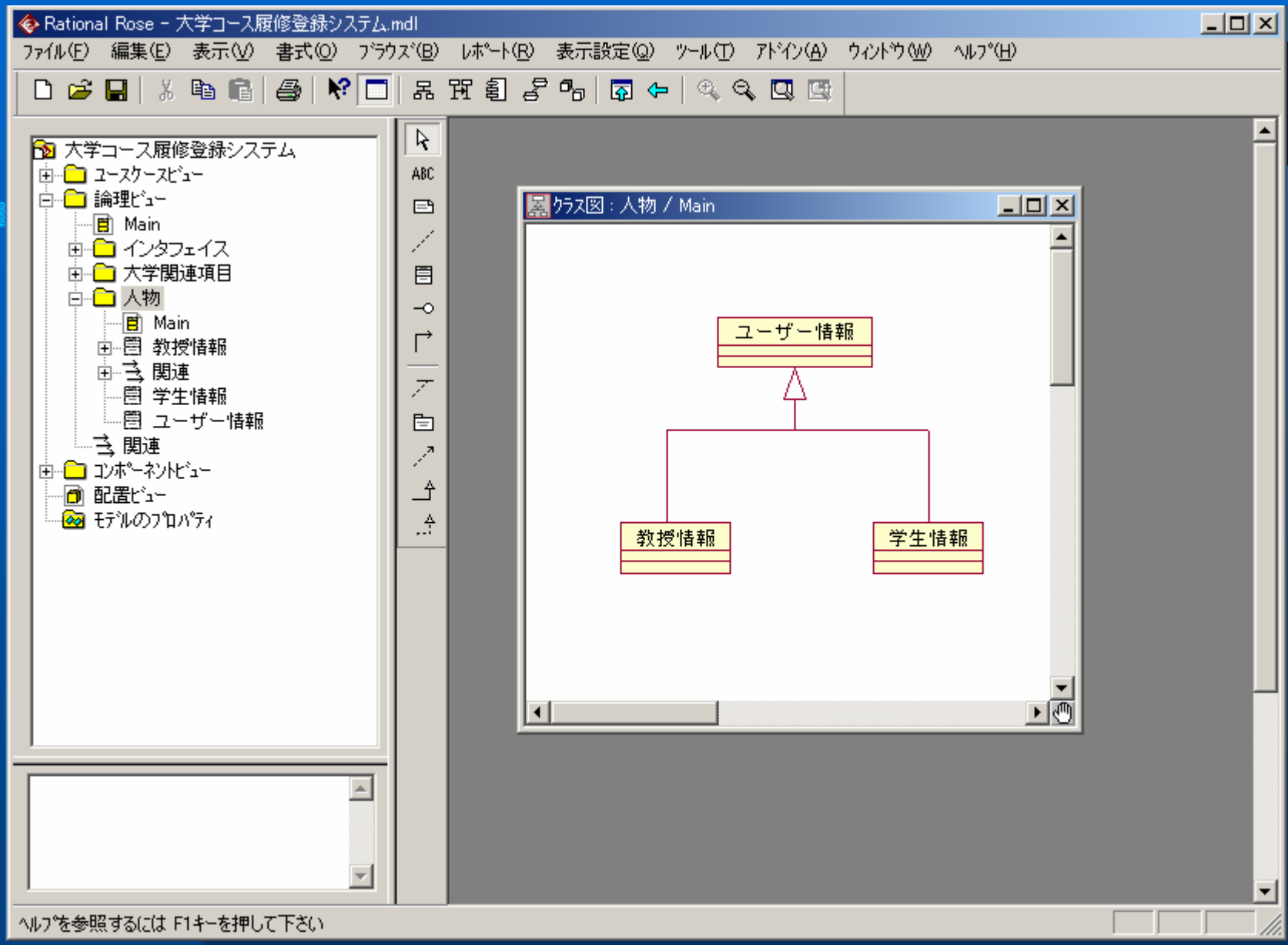

Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
Main
インタフェイス
大学関連項目
人物
Main
教授情報
関連
学生情報
ユーザー情報
関連
コンポーネントビュー
配置ビュー
モデルのプロパティ
ABC
クラス図：人物 / Main
ユーザー情報
教授情報
学生情報
ヘルプを参照するには F1キーを押して下さい

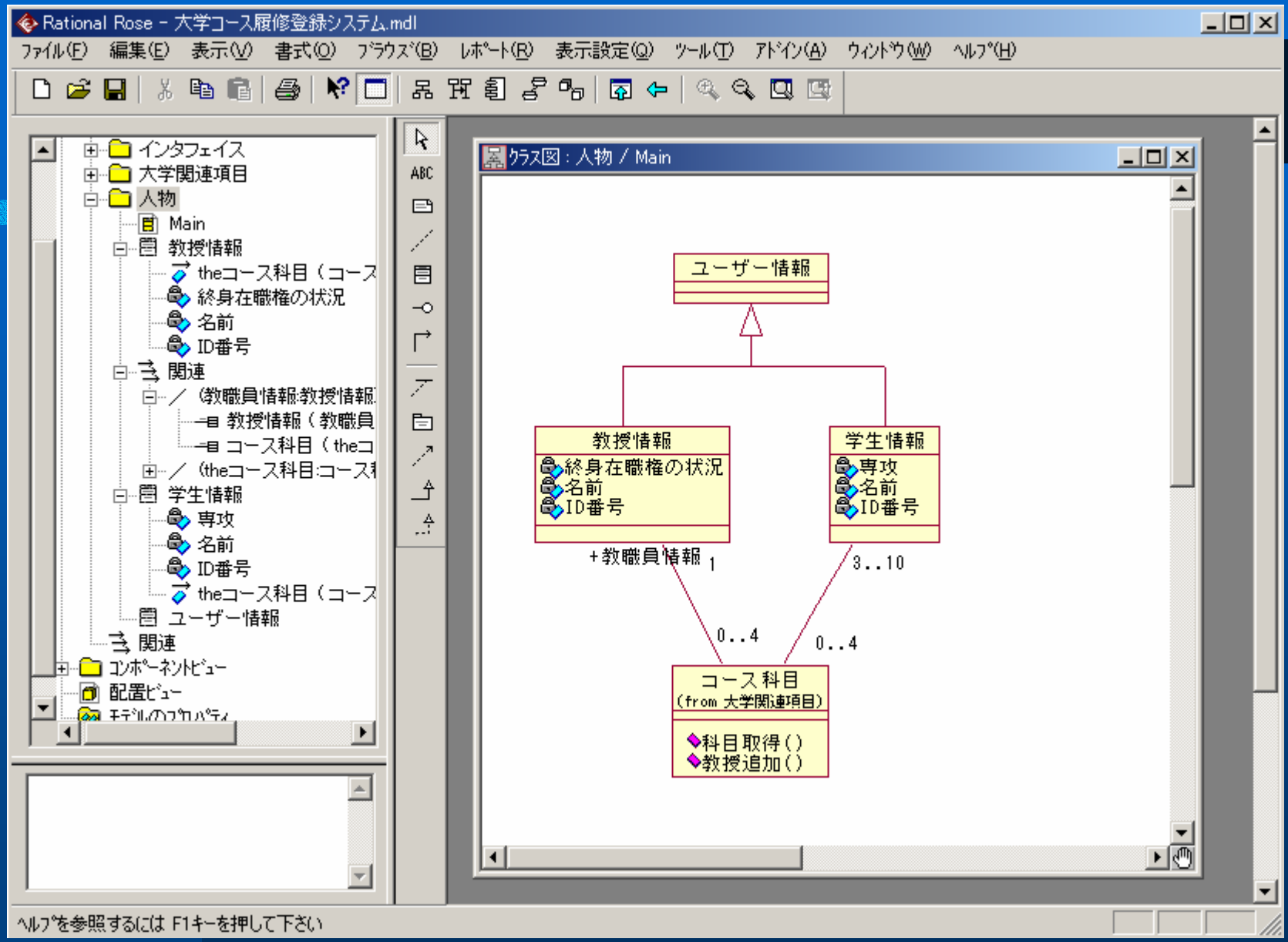
継承ツリー

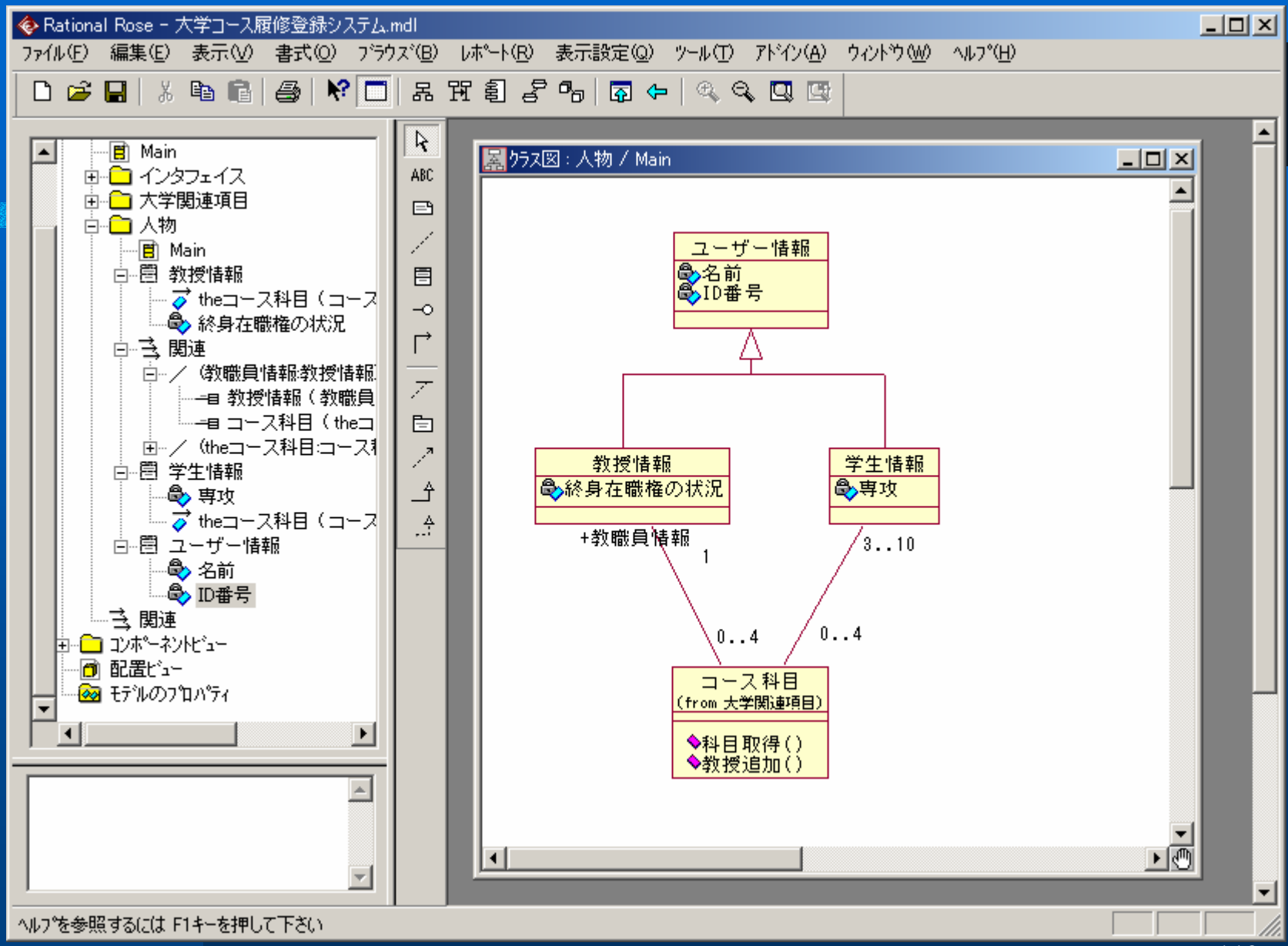
- 継承関係において特殊化を行う根拠（サブクラスを作成理由）を弁別子（discriminator）という。
- 弁別子は、一般的に有限個の値を持ち、各値ごとにサブクラスを作成できる。
- 例えば、コースクラスに対する弁別子の1つとして、コース場所が考えられる。この弁別子に基づいて、コースクラスに構内コースと構外コースの2つサブクラスを作成できる。
- 継承関係では、1つの弁別子から作成されたサブクラスはすべて、1つのツリーとして表現される。

継承ツリーの作成

1. 継承階層を作成するクラス図を開く。
2. ツールバーの[クラス]をクリックし、クラス図をクリックしてクラスを描画する。
3. クラスが選択されたままの状態で、クラス名を入力する。
4. ツールバーの[汎化]をクリックする。
5. サブクラスの1つをクリックして、汎化直線をスーパークラスにドラッグする。


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
Main
インタフェイス
大学関連項目
人物
Main
教授情報
関連
学生情報
ユーザー情報
関連
コンポーネントビュー
配置ビュー
モデルのプロパティ
ABC
クラス図 : 人物 / Main
ユーザー情報
教授情報
学生情報
ヘルプを参照するには F1キーを押して下さい


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
インタフェイス
大学関連項目
人物
Main
教授情報
theコース科目（コース
終身在職権の状況
名前
ID番号
関連
(教職員情報:教授情報
教授情報（教職員
コース科目（the
(theコース科目:コース
学生情報
専攻
名前
ID番号
theコース科目（コース
ユーザー情報
関連
コンポーネントビュー
配置ビュー
クラス図：人物 / Main
ユーザー情報
教授情報
終身在職権の状況
名前
ID番号
学生情報
専攻
名前
ID番号
+教職員情報
1
3..10
0..4
0..4
コース科目
(from 大学関連項目)
科目取得()
教授追加()
ヘルプを参照するには F1キーを押して下さい


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
Main
インタフェイス
大学関連項目
人物
Main
教授情報
theコース科目（コース
終身在職権の状況
関連
(教職員情報:教授情報
教授情報（教職員
コース科目（theコ
(theコース科目:コース
学生情報
専攻
theコース科目（コース
ユーザー情報
名前
ID番号
関連
コンポーネントビュー
配置ビュー
モデルのプロパティ
クラス図：人物 / Main
ユーザー情報
名前
ID番号
教授情報
終身在職権の状況
学生情報
専攻
+教職員情報
1
3..10
0..4
0..4
コース科目
(from 大学関連項目)
科目取得()
教授追加()
ヘルプを参照するには F1キーを押して下さい

単一継承と多重継承

- 単一クラスでは、クラスは一系統の親を持つ。つまり、継承階層の各クラスには1つ以下のスーパークラスが存在する（例えば、自動車はモーター付きの乗り物である）。
- 多重継承には、複数のスーパークラスの系列が含まれる（水陸両用車は、モーと一付きの乗り物であり、水上の乗り物である）。
- 多重継承に関しては、多数の問題が発生し、コードの保守性が低下する可能性がある。
- スーパークラスが増えるほど、どこから何を継承したか、また変更によってどういう影響があるかを判断することが困難になる。

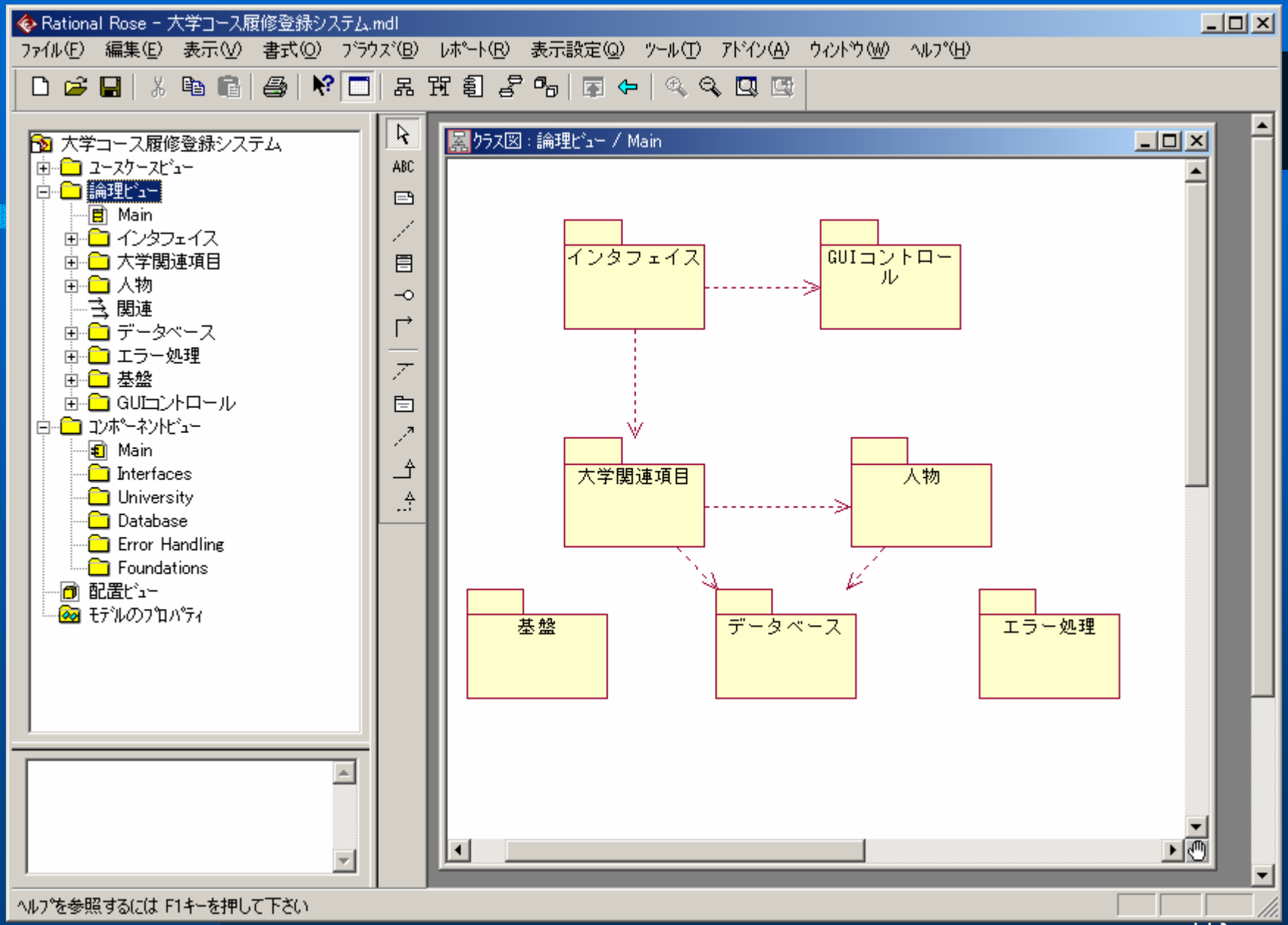
継承と集約

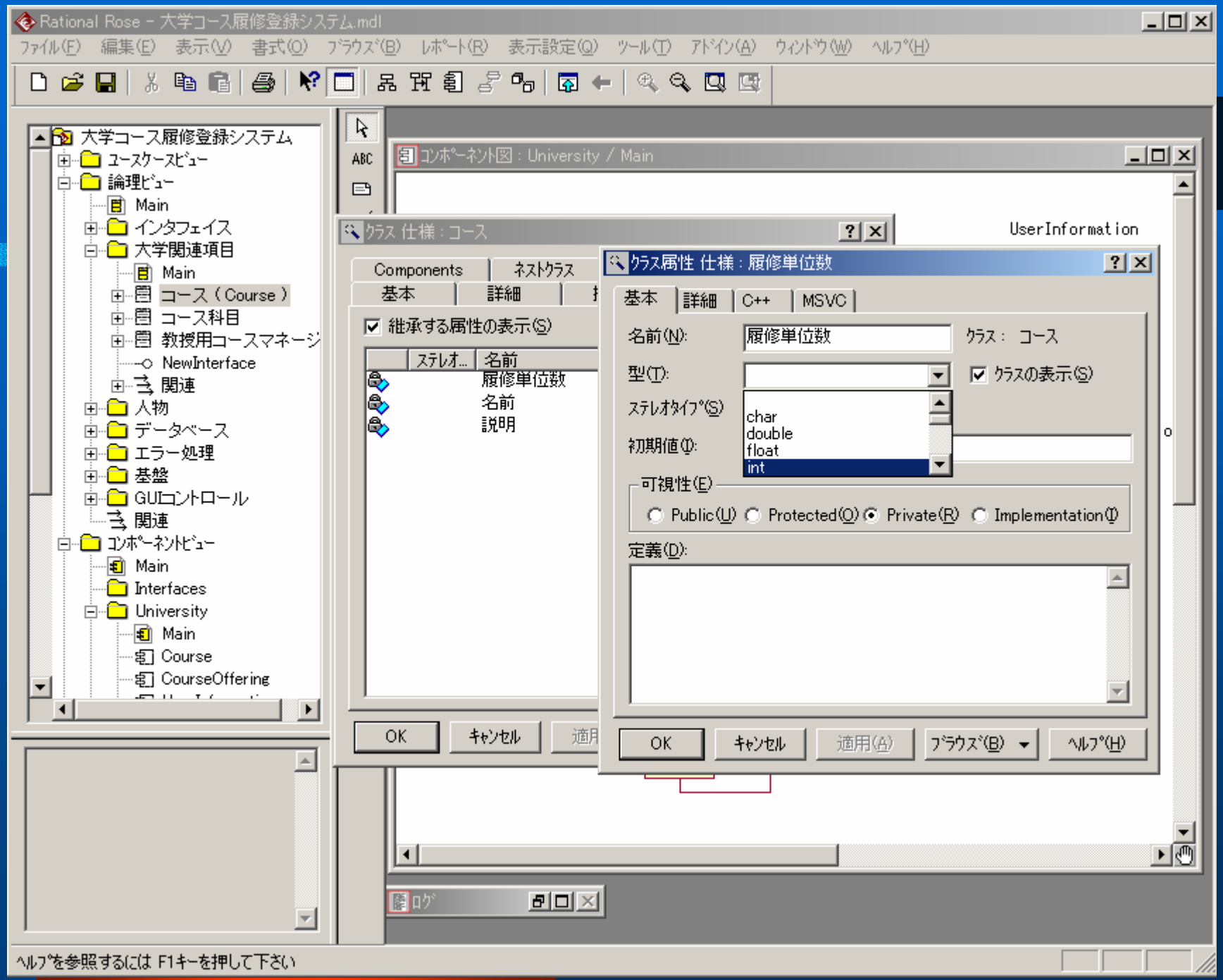
- 「継承は良いものだからたくさん使えばそれだけコードが良くなる」という考え方は、正しくない。継承の誤った使用により、問題が発生する場合がある。
- 継承は、特殊なものから共通性を分離するために使用されるべきである。
- 一方、集約は複数の要素から構成されている関係を示すために使用されるべきである。
- この2種類の関係は、よく併用される。例えば、学生情報クラスは、昼間コースまたは夜間コース（継承）の分類（集約）を持つ。

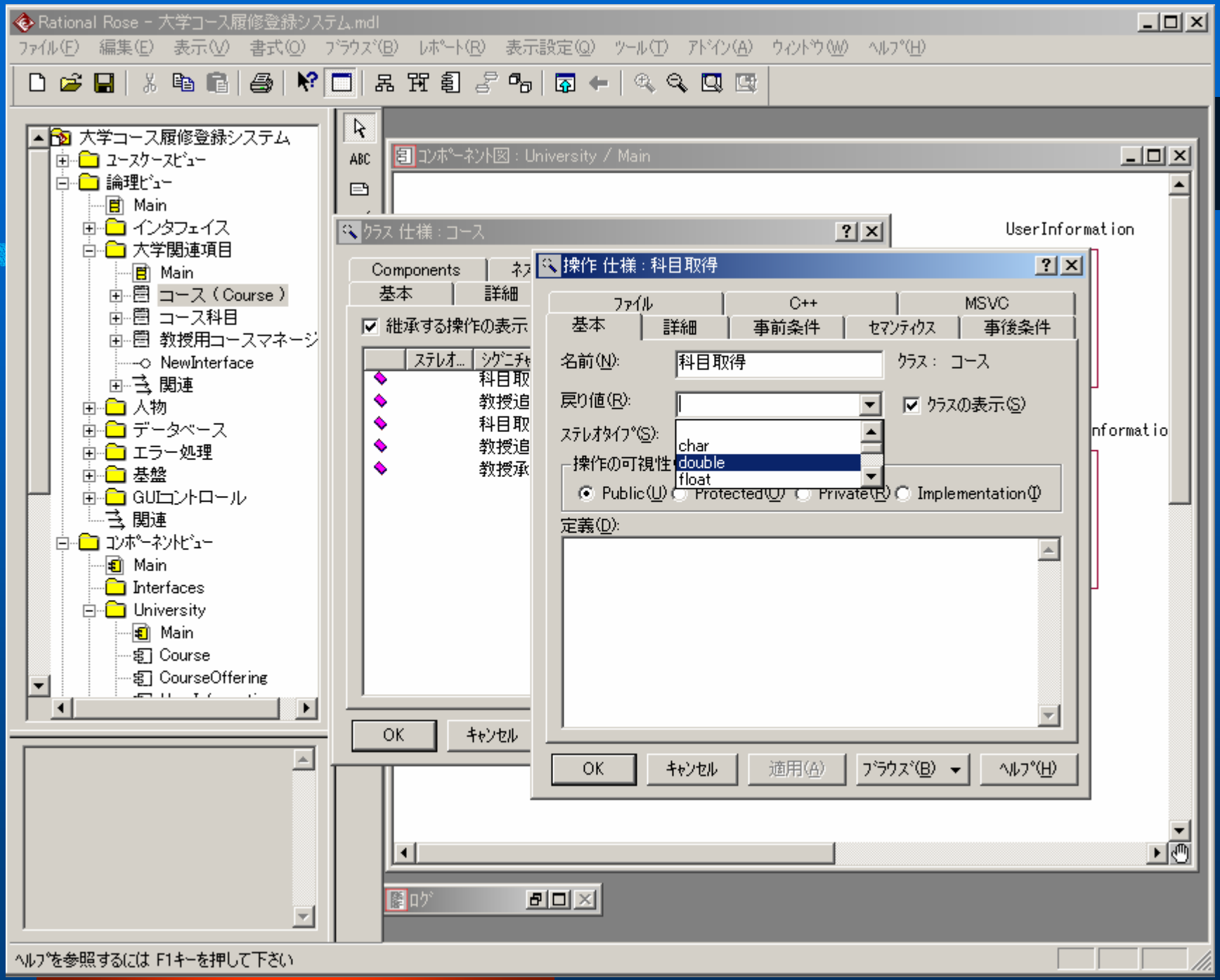
システムアーキテクチャーの設計

論理ビューのチェック

- 論理ビューには、開発中のシステムの主要な抽象概念を表すクラスと関係が含まれる。ここまで述べられてほとんどのUML表記（クラス、関連、集約、汎化、パッケージなど）は、このビューに含まれる。
- コード生成する前に、まず論理ビューに含まれている、クラス、クラスの属性と操作、関連、パッケージなどチェックしなければならない。
- クラスの属性と操作の型と戻り値を設定する。


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
Main
インタフェイス
大学関連項目
人物
関連
データベース
エラー処理
基盤
GUIコントロール
コンポーネントビュー
Main
Interfaces
University
Database
Error Handling
Foundations
配置ビュー
モデルのプロパティ
ABC
クラス図：論理ビュー / Main
インタフェイス
GUIコントロール
大学関連項目
人物
基盤
データベース
エラー処理
ヘルプを参照するには F1キーを押して下さい


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
Main
インタフェイス
大学関連項目
Main
コース (Course)
コース科目
教授用コースマネージ
NewInterface
関連
人物
データベース
エラー処理
基盤
GUIコントロール
関連
コンポーネントビュー
Main
Interfaces
University
Main
Course
CourseOffering
コンポーネント図 : University / Main
UserInformation
クラス 仕様 : コース
Components
ネストクラス
基本
詳細
継承する属性の表示(S)
ステレオ...
名前
履修単位数
名前
説明
OK
キャンセル
適用
クラス属性 仕様 : 履修単位数
基本
詳細
C++
MSVC
名前(N):
履修単位数
クラス : コース
型(T):
クラスの表示(S)
ステレオタイプ(S):
char
double
float
int
初期値(I):
可視性(E)
Public(U) Protected(O) Private(R) Implementation(I)
定義(D):
OK
キャンセル
適用(A)
ブラウズ(B)
ヘルプ(H)
ログ
ヘルプを参照するには F1キーを押して下さい

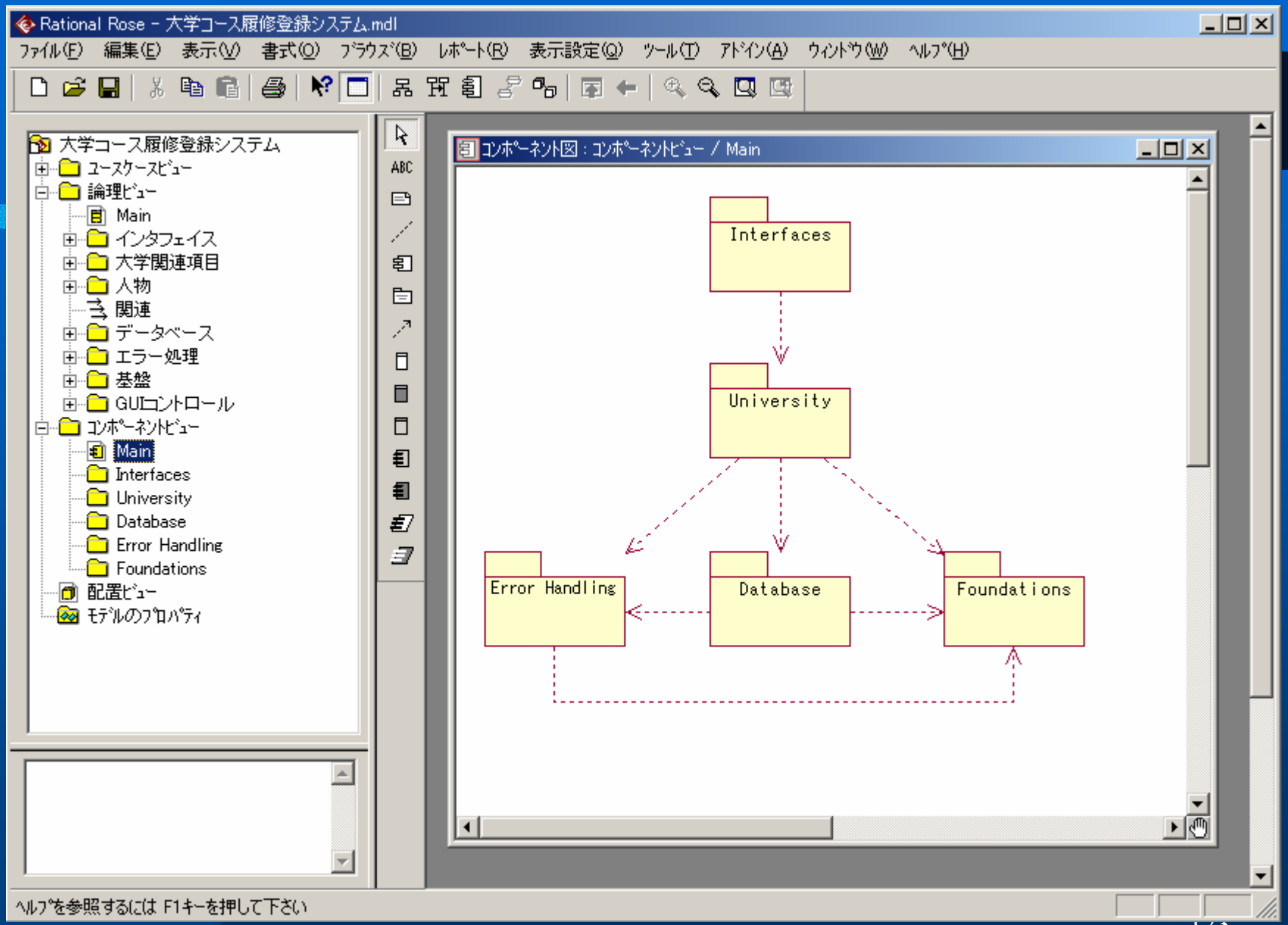

Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
Main
インタフェイス
大学関連項目
Main
コース (Course)
コース科目
教授用コースマネージ
NewInterface
関連
人物
データベース
エラー処理
基盤
GUIコントロール
関連
コンポーネントビュー
Main
Interfaces
University
Main
Course
CourseOffering
コンポーネント図 : University / Main
UserInformation
クラス 仕様 : コース
Components
基本
詳細
継承する操作の表示
ステレオ...
科目取
教授追
科目取
教授追
教授承
OK
キャンセル
操作 仕様 : 科目取得
ファイル
C++
MSVC
基本
詳細
事前条件
セマンティクス
事後条件
名前(N):
科目取得
クラス : コース
戻り値(R):
クラスの表示(S)
ステレオタイプ(S):
char
double
float
操作の可視性
Public(U) Protected(O) Private(R) Implementation(I)
定義(D):
OK
キャンセル
適用(A)
ブラウズ(B)
ヘルプ(H)
ログ
ヘルプを参照するには F1キーを押して下さい

コンポーネントビュー

- コンポーネントビューは、開発環境内の実際のソフトウエアモジュール構成を表す。このビューは、開発、ソフトウエア管理、再利用、及びプログラム言語や開発ツールによる制約を容易にすることを目的に作成する。
- コンポーネントビューの要素は、パッケージとコンポーネント、及びそれらの依存関係である。
- コンポーネントビューのパッケージは、システムの物理的な区分を表し、サブシステムとも呼ばれている。

コンポーネントビューパッケージの作成

1. ブラウズで[コンポーネント]パッケージを右クリックして、ショートカットメニューを表示する。
2. [新規作成を]をポイントし、[パッケージ]をクリックする。ブラウズに“NewPackage”というアイテムが追加される。
3. “NewPackage”が選択されたままの状態で、パッケージの名前を入力する。

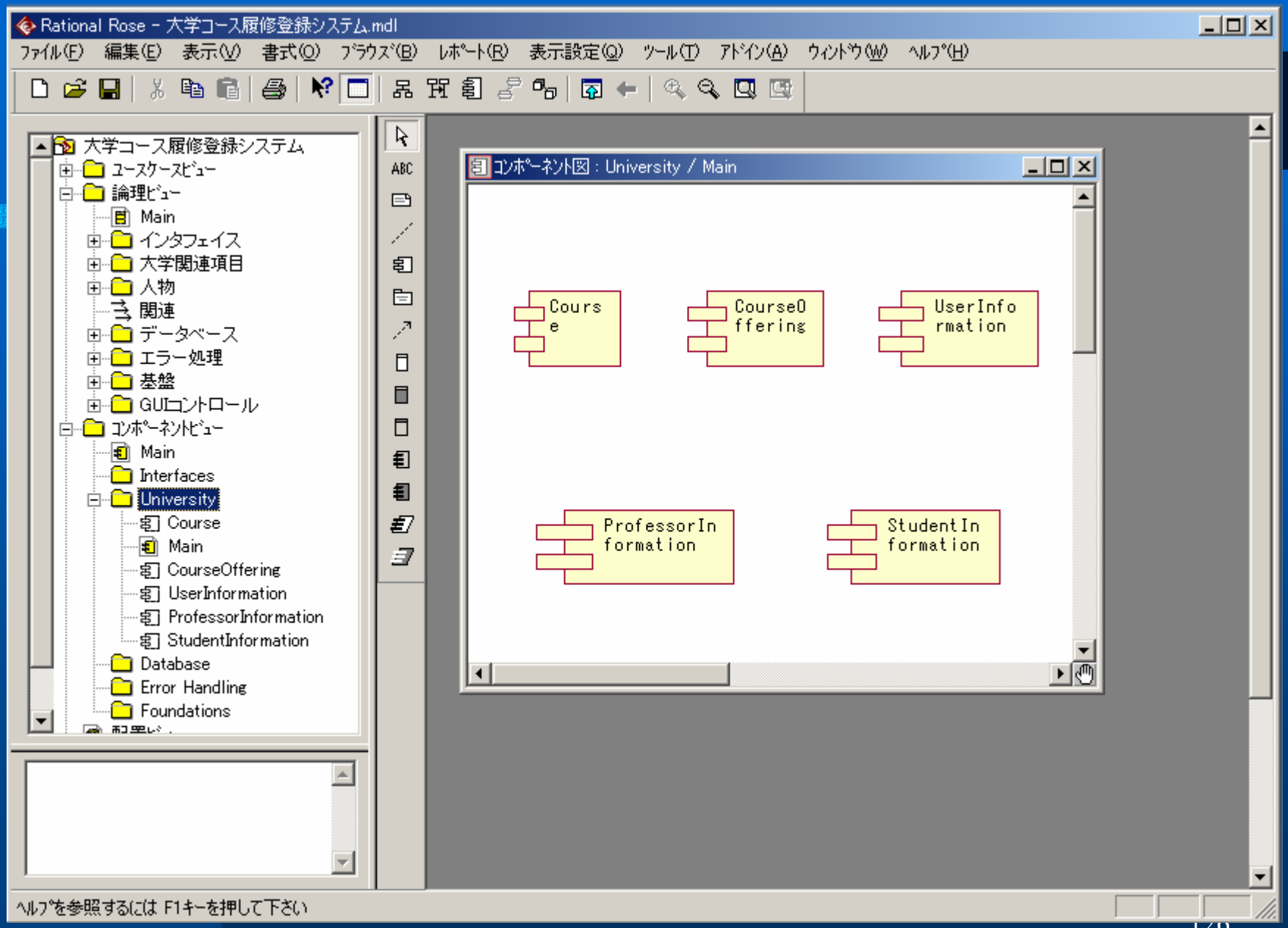

Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
Main
インタフェイス
大学関連項目
人物
関連
データベース
エラー処理
基盤
GUIコントロール
コンポーネントビュー
Main
Interfaces
University
Database
Error Handling
Foundations
配置ビュー
モデルのプロパティ
コンポーネント図 : コンポーネントビュー / Main
Interfaces
University
Error Handling
Database
Foundations
ヘルプを参照するには F1キーを押して下さい

コンポーネント

- コンポーネントは、パッケージ（サブシステム）に含まれるソフトウエアのファイルを表す。
- ファイルの種類は使用言語によって異なる。例えば、C++では.h及び.cppファイル、Javaでは、.javaファイルなどがそれぞれソフトウエアコンポーネントになる。

コンポーネントの作成

1. コンポーネント図を開く。
2. ツールバーの[パッケージ仕様]をクリックする。
3. 図をクリックしてコンポーネントを配置する。同時にブラウズにもコンポーネントが追加される。
4. コンポーネントが選択されたままの状態で、コンポーネントの名前を入力する。

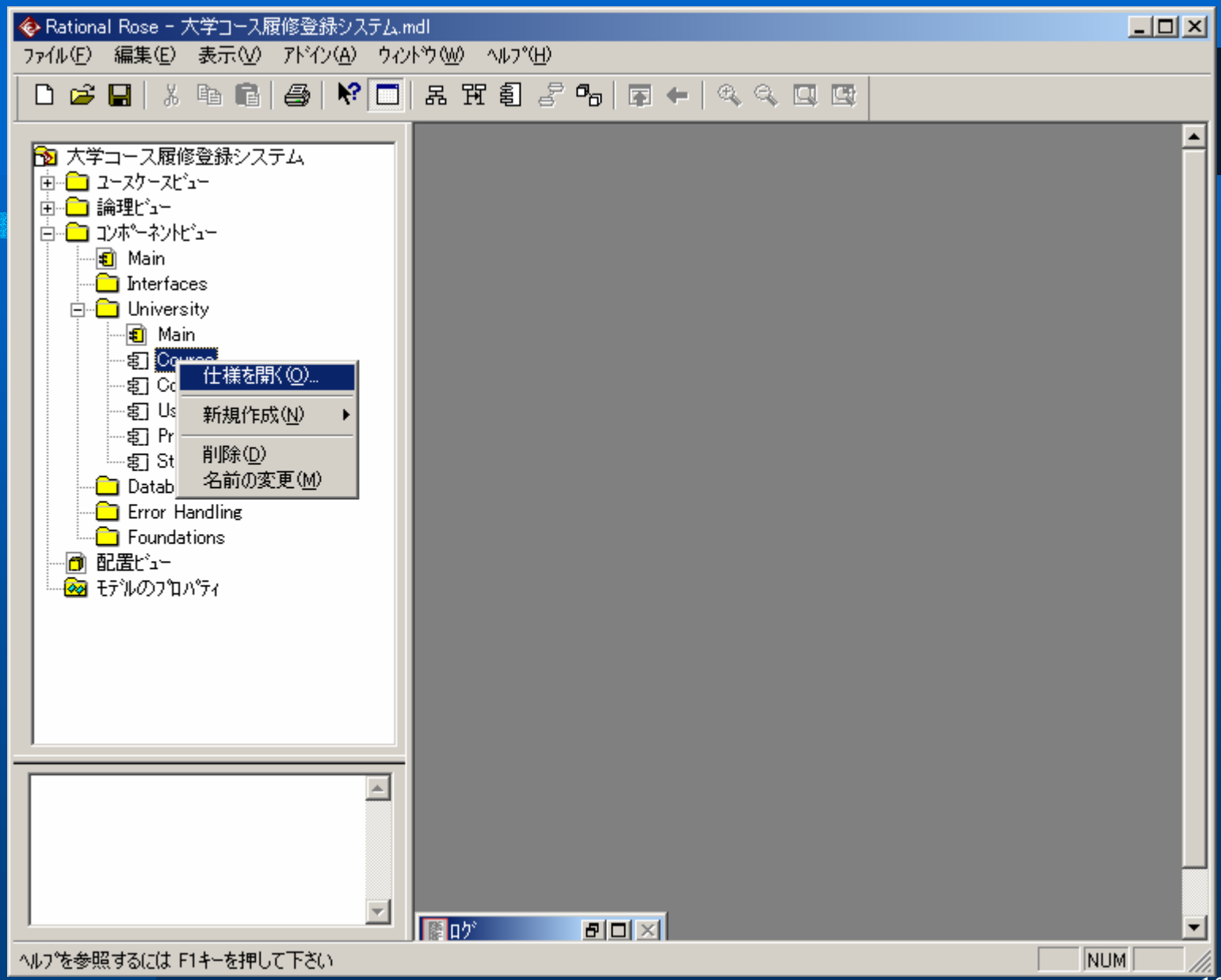

Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
Main
インタフェイス
大学関連項目
人物
関連
データベース
エラー処理
基盤
GUIコントロール
コンポーネントビュー
Main
Interfaces
University
Course
Main
CourseOffering
UserInformation
ProfessorInformation
StudentInformation
Database
Error Handling
Foundations
コンポーネント図 : University / Main
Cours e
CourseO ffering
UserInfo rmation
ProfessorIn formation
StudentIn formation
ヘルプを参照するには F1キーを押して下さい

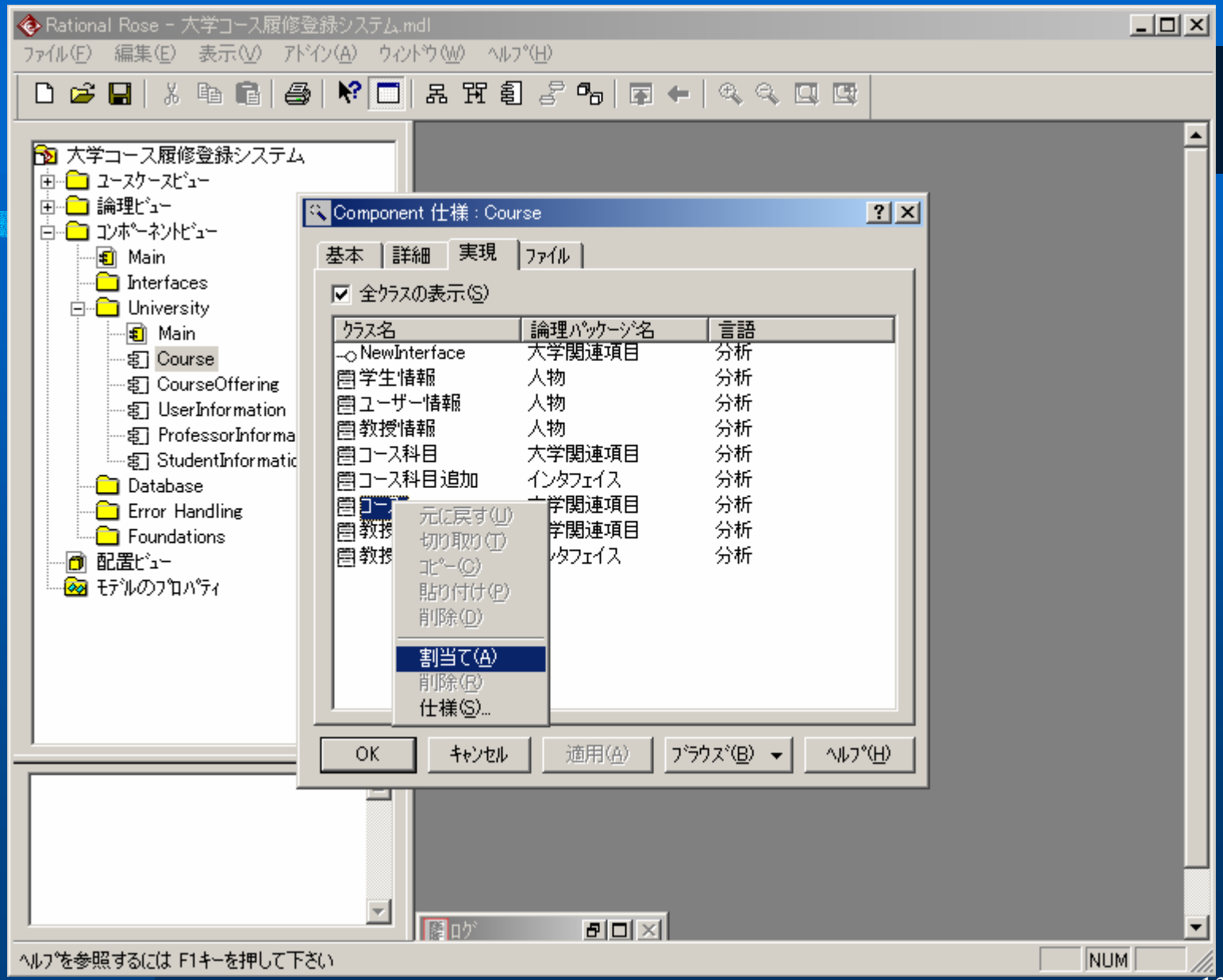
コンポーネントへのクラスの対応付け　1

- 論理ビューのクラスをコンポーネントビューのコンポーネントに対応付けることにより、モデルの論理ビューの情報がコンポーネントビューの情報に関連付けられている。
- 一般的に、論理ビューのクラスはコンポーネントビューのコンポーネントに直接対応（1対1）する。

コンポーネントへのクラスの対応付け　２

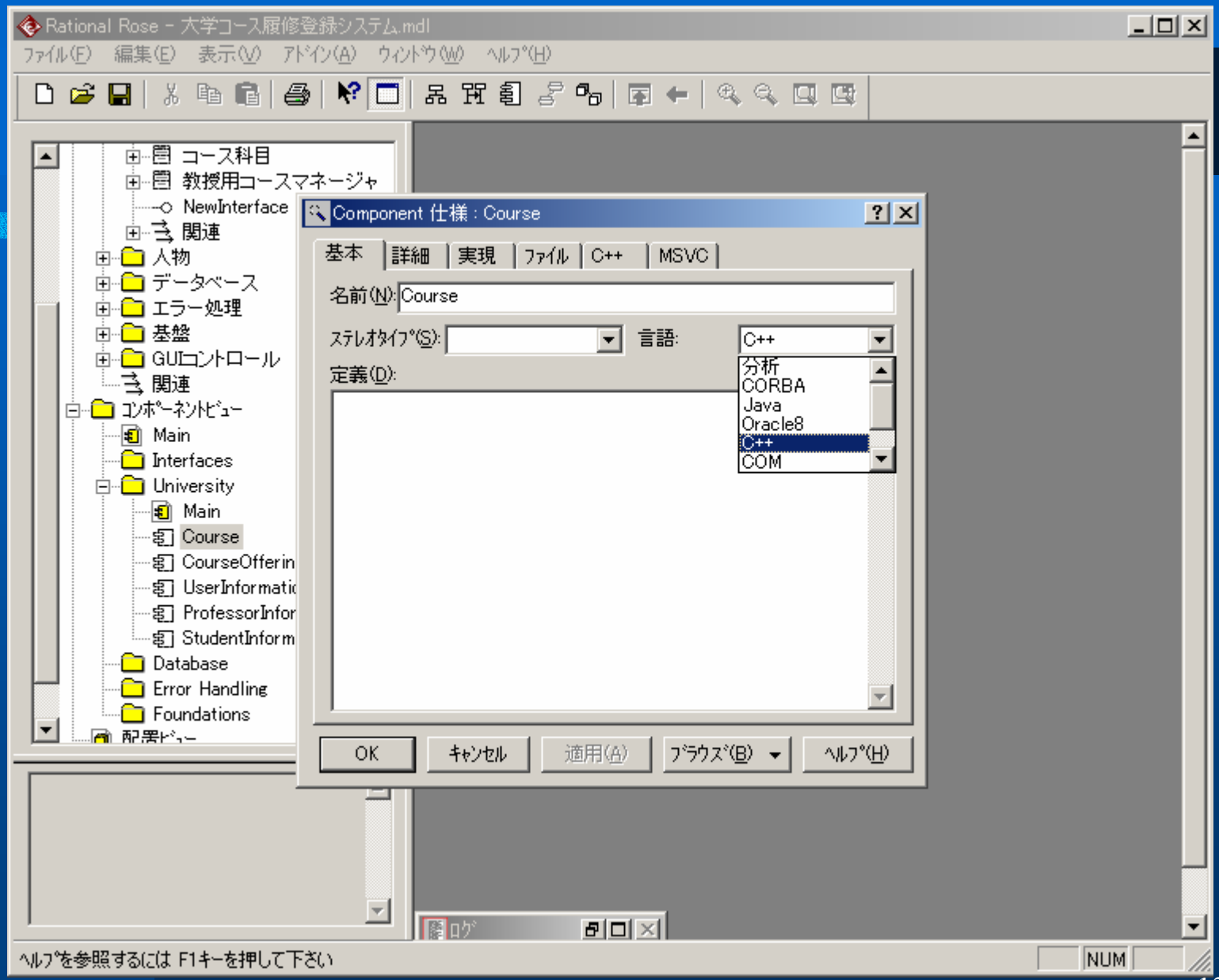
1. ブラウズでコンポーネントを右クリックし、ショートカットメニューを開く。
2. [仕様を開く]をクリックする。
3. [実現]タブをクリックする。
4. クラスを右クリックして、ショートカットメニューを表示する。
5. [割当て]をクリックする。
6. [OK]をクリックして、[コンポーネント仕様]ダイアログボックスを閉じる。


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
コンポーネントビュー
Main
Interfaces
University
Main
仕様を開く(O)...
新規作成(N)
削除(D)
名前の変更(M)
Error Handling
Foundations
配置ビュー
モデルのプロパティ
ログ
ヘルプを参照するには F1キーを押して下さい
NUM


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
コンポーネントビュー
Main
Interfaces
University
Main
Course
CourseOffering
UserInformation
ProfessorInforma
StudentInformatic
Database
Error Handling
Foundations
配置ビュー
モデルのプロパティ
Component 仕様 : Course
基本 詳細 実現 ファイル
全クラスの表示(S)
クラス名 論理パッケージ名 言語
NewInterface 大学関連項目 分析
学生情報 人物 分析
ユーザー情報 人物 分析
教授情報 人物 分析
コース科目 大学関連項目 分析
コース科目追加 インタフェイス 分析
元に戻す(U)
切り取り(T)
コピー(C)
貼り付け(P)
削除(D)
割当て(A)
削除(R)
仕様(S)...
OK キャンセル 適用(A) ブラウズ(B) ヘルプ(H)
ログ
ヘルプを参照するには F1キーを押して下さい
NUM

実装言語の選択

1. ブラウズでコンポーネントを右クリックし、ショートカットメニューを開く。
2. [仕様を開く]をクリックする。
3. [基本]タブをクリックする。
4. 言語を選択する。


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) アドイン(A) ウィンドウ(W) ヘルプ(H)
コース科目
教授用コースマネージャ
NewInterface
関連
人物
データベース
エラー処理
基盤
GUIコントロール
関連
コンポーネントビュー
Main
Interfaces
University
Main
Course
CourseOfferin
UserInformatic
ProfessorInfor
StudentInform
Database
Error Handling
Foundations
Component 仕様 : Course
基本
詳細
実現
ファイル
C++
MSVC
名前(N): Course
ステレオタイプ(S):
言語: C++
定義(D):
分析
CORBA
Java
Oracle8
C++
COM
OK
キャンセル
適用(A)
ブラウズ(B)
ヘルプ(H)
ログ
ヘルプを参照するには F1キーを押して下さい
NUM

Rational Roseによる ラウンドトリップエンジニアリング

- ✓ Forward Engineering（コード生成）
- ✓ Reverse Engineering（ディサイン生成）

Forward Engineering

Forward Engineeringの手順

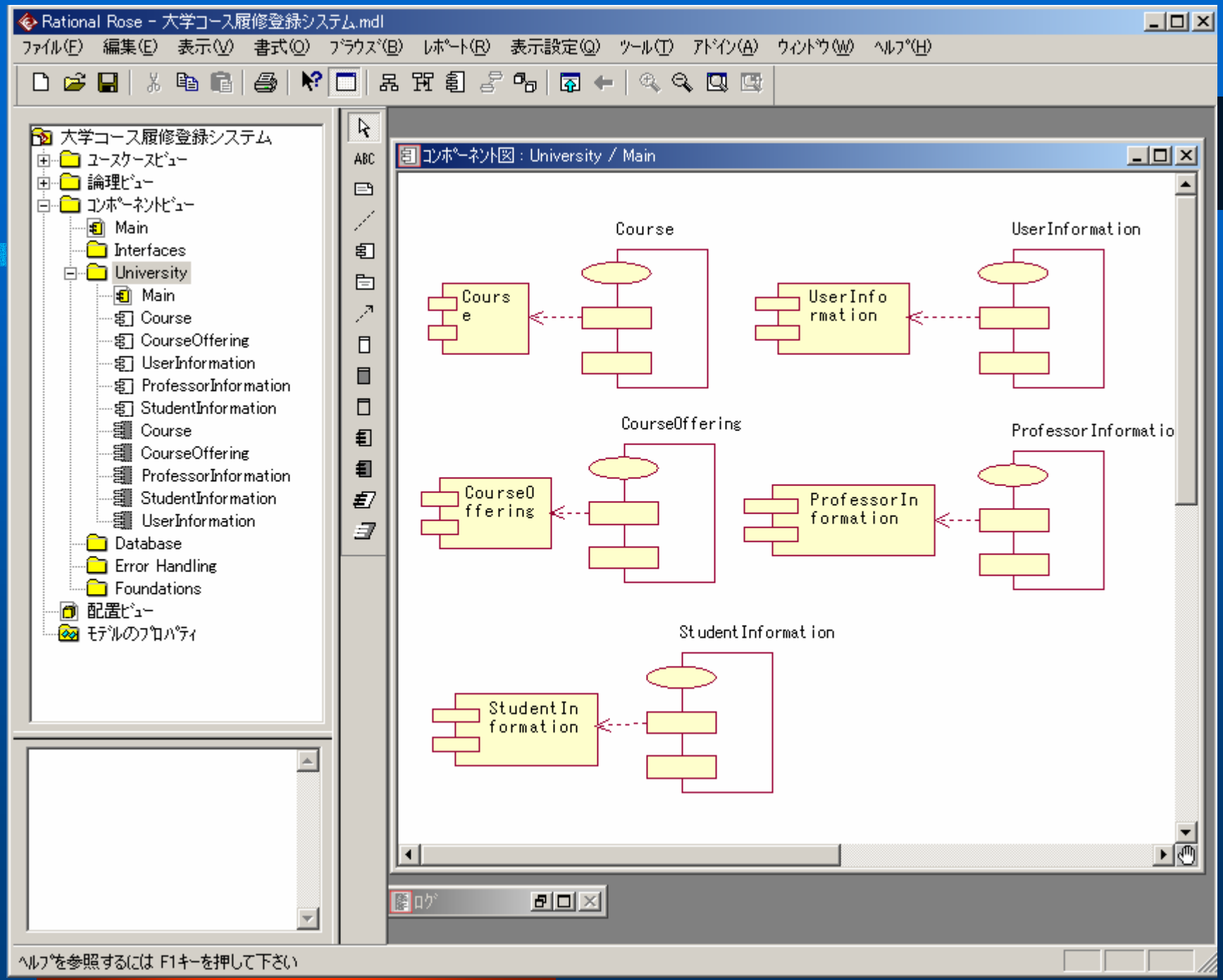
1. コンポーネント図に本体コンポーネントを作成する。
2. コンポーネントを選択し、コードを生成する。
3. コード生成エラーを評価する。

コンポーネント図に本体コンポーネントを作成する 1

- クラスがコンポーネントに対応付けられている場合、Rational Roseでは、コンポーネント型に基づいてコードが生成される。
- ヘッダーファイルを表すコンポーネントしか定義されていない場合は、クラスの定義情報を記述した .h ファイルが1つだけ生成される。ソースファイル本体を表す対応コンポーネントが定義されている場合は、そのクラスの宣言情報を記述した .cpp ファイルが生成される。
- クラスがソースファイル本体を表すコンポーネントに対応付けられている場合は、クラスの .h ファイルと .cpp ファイルが生成される。

コンポーネント図に本体コンポーネントを作成する 2

1. コンポーネント図をダブルクリックして、図を開く。
2. ツールバーの[パッケージ本体]をクリックする。
3. 図をクリックしてパッケージ本体を配置する。
4. 本体が選択されたままの状態で、本体の名前を入力する。一般的には、仕様の名前と同じにする。
5. ツールバーの[依存関係]をクリックする。
6. 図のパッケージ本体をクリックし、依存関係をパッケージ仕様にドラッグする。

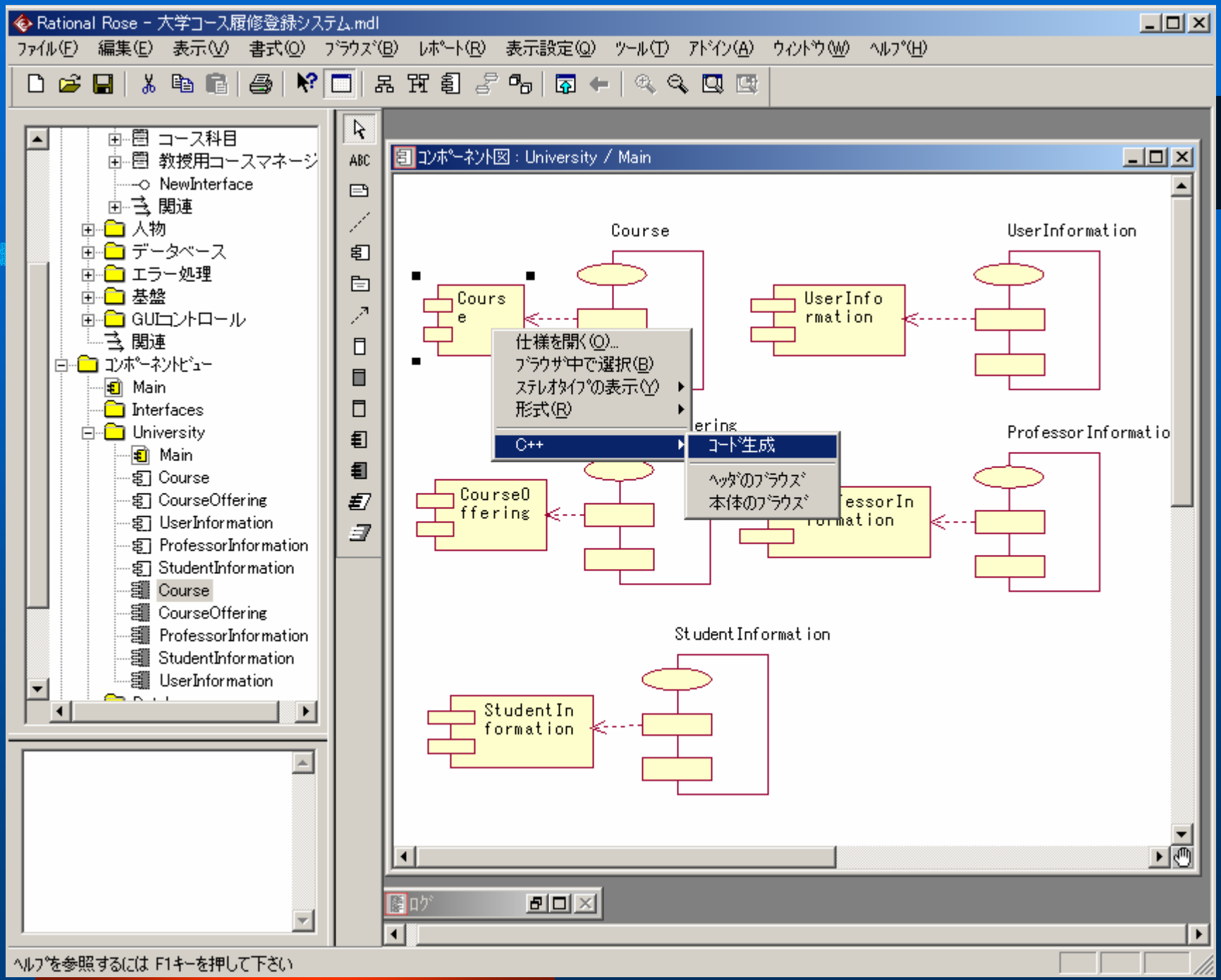

Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
コンポーネントビュー
Main
Interfaces
University
Main
Course
CourseOffering
UserInformation
ProfessorInformation
StudentInformation
Course
CourseOffering
ProfessorInformation
StudentInformation
UserInformation
Database
Error Handling
Foundations
配置ビュー
モデルのプロパティ
コンポーネント図 : University / Main
Course
Cours e
UserInformation
UserInfo rmation
CourseOffering
CourseO ffering
ProfessorInformatio
ProfessorIn formation
StudentInformation
StudentIn formation
ログ
ヘルプを参照するには F1キーを押して下さい

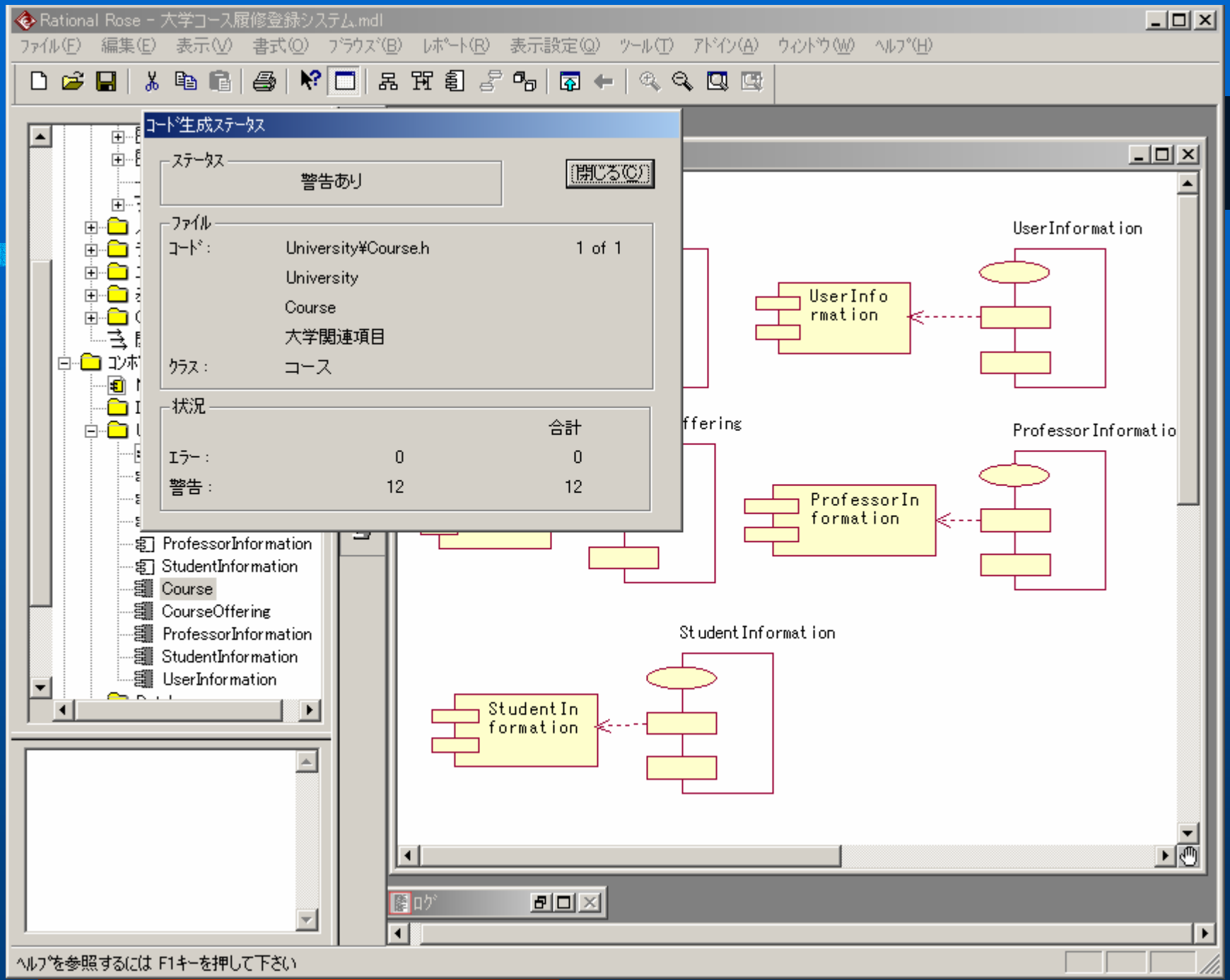
コンポーネントを選択し、コードを生成する　1

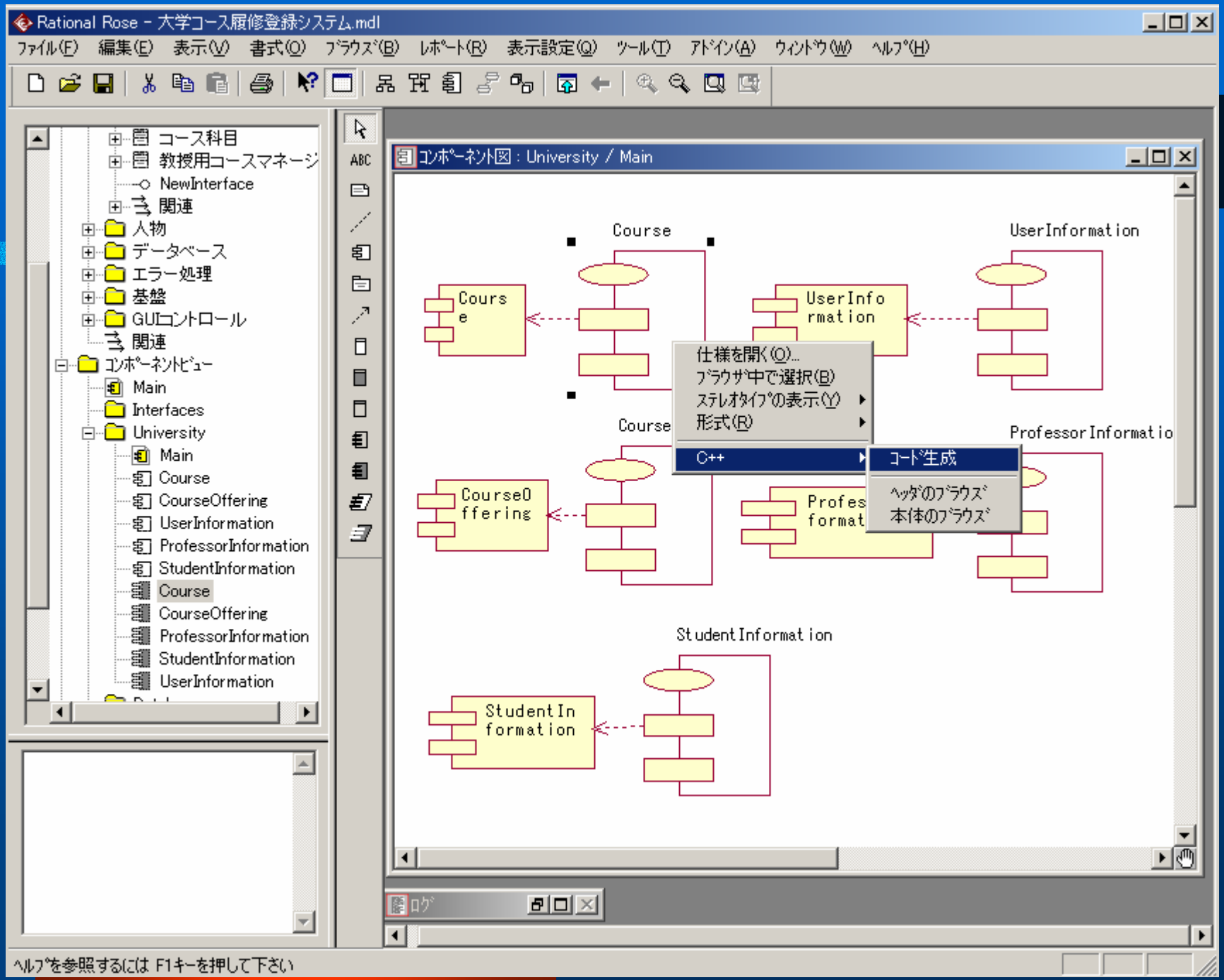
- コードはパッケージ全体、1つのクラス、またはクラスの集合に対して生成可能である。
- クラスがコンポーネントに対応付けられていない場合は、クラスの名前がコードのファイル名に使用される。生成されたコードは、論理ビューのパッケージ名に対するディレクトリに保存される。
- クラスがコンポーネントに対応付けられている場合は、コンポーネントの名前がコードのファイル名に使用される。生成されたコードは、コンポーネントビューのパッケージ名に対応するディレクトリに保存される。

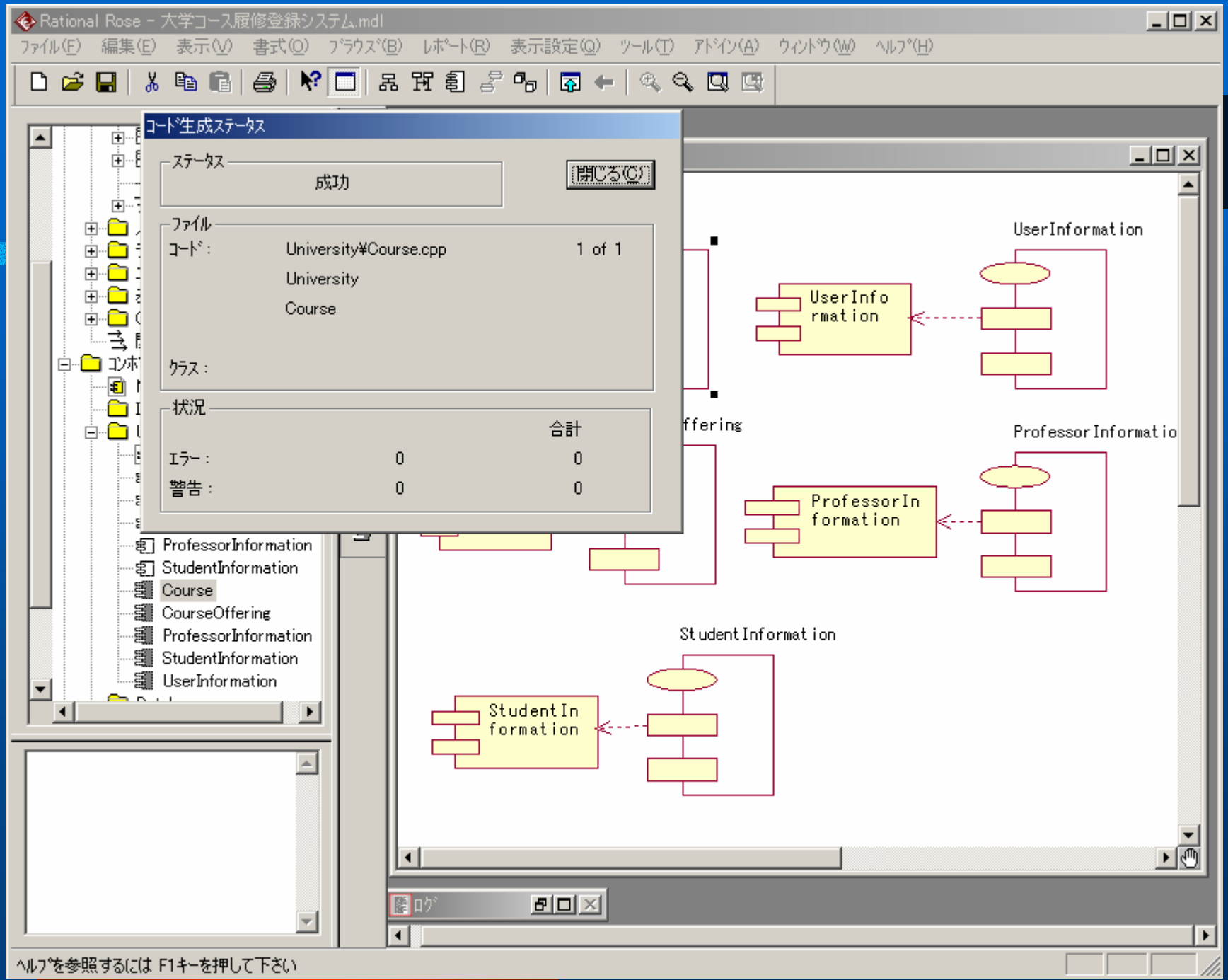
コンポーネントを選択し、コードを生成する 2

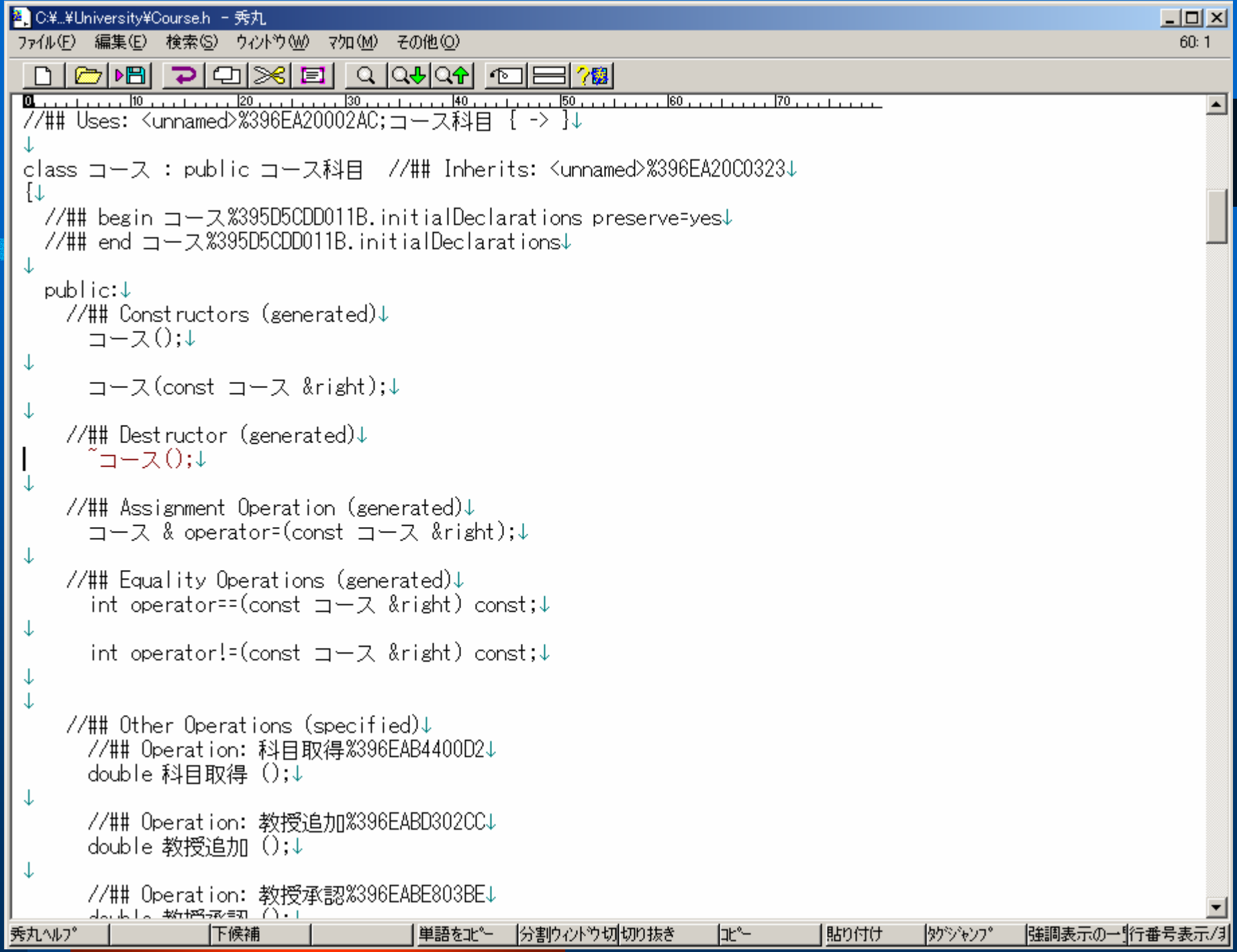
1. パッケージ、クラス、またはクラス集合をクリックする。
2. [ツール]メニューの[C++]をポイントし、[コード生成]をクリックする。
3. [コード生成ステータス]ダイアログボックスに、ステータスが表示される。


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
コース科目
教授用コースマネージ
NewInterface
関連
人物
データベース
エラー処理
基盤
GUIコントロール
関連
コンポーネントビュー
Main
Interfaces
University
Main
Course
CourseOffering
UserInformation
ProfessorInformation
StudentInformation
Course
CourseOffering
ProfessorInformation
StudentInformation
UserInformation
コンポーネント図 : University / Main
Course
UserInformation
Cours e
UserInfo rmation
仕様を開く(O)...
ブラウザ中で選択(B)
ステレオタイプの表示(Y)
形式(R)
C++
コード生成
ヘッダのブラウズ
本体のブラウズ
ProfessorInformatio
CourseO ffering
StudentInformation
StudentIn formation
ログ
ヘルプを参照するには F1キーを押して下さい


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
コード生成ステータス
ステータス
警告あり
閉じる(C)
ファイル
コード:
University¥Course.h
1 of 1
University
Course
大学関連項目
クラス:
コース
状況
合計
エラー:
0
0
警告:
12
12
ProfessorInformation
StudentInformation
Course
CourseOffering
ProfessorInformation
StudentInformation
UserInformation
UserInformation
UserInfo rmation
ffering
ProfessorInformatio
ProfessorIn formation
StudentInformation
StudentIn formation
ログ
ヘルプを参照するには F1キーを押して下さい


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
コース科目
教授用コースマネージ
NewInterface
関連
人物
データベース
エラー処理
基盤
GUIコントロール
関連
コンポーネントビュー
Main
Interfaces
University
Main
Course
CourseOffering
UserInformation
ProfessorInformation
StudentInformation
Course
CourseOffering
ProfessorInformation
StudentInformation
UserInformation
コンポーネント図 : University / Main
Course
UserInformation
Cours e
UserInfo rmation
仕様を開く(O)...
ブラウザ中で選択(B)
ステレオタイプの表示(Y)
形式(R)
C++
コード生成
ヘッダのブラウズ
本体のブラウズ
Course
ProfessorInformatio
CourseO ffering
Profes format
StudentInformation
StudentIn formation
ログ
ヘルプを参照するには F1キーを押して下さい


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
コード生成ステータス
ステータス
成功
閉じる(C)
ファイル
コード:
University¥Course.cpp
1 of 1
University
Course
クラス:
状況
合計
エラー:
0
0
警告:
0
0
ProfessorInformation
StudentInformation
Course
CourseOffering
ProfessorInformation
StudentInformation
UserInformation
UserInformation
UserInfo rmation
ffering
ProfessorInformatio
ProfessorIn formation
StudentInformation
StudentIn formation
ログ
ヘルプを参照するには F1キーを押して下さい


C:¥...¥University¥Course.h - 秀丸
ファイル(F) 編集(E) 検索(S) ウィンドウ(W) マクロ(M) その他(O)
60: 1
//## Uses: <unnamed>%396EA20002AC;コース科目 { -> }
class コース : public コース科目 //## Inherits: <unnamed>%396EA20C0323
{
//## begin コース%395D5CDD011B.initialDeclarations preserve=yes
//## end コース%395D5CDD011B.initialDeclarations
public:
//## Constructors (generated)
コース();
コース(const コース &right);
//## Destructor (generated)
~コース();
//## Assignment Operation (generated)
コース & operator=(const コース &right);
//## Equality Operations (generated)
int operator==(const コース &right) const;
int operator!=(const コース &right) const;
//## Other Operations (specified)
//## Operation: 科目取得%396EAB4400D2
double 科目取得 ();
//## Operation: 教授追加%396EABD302CC
double 教授追加 ();
//## Operation: 教授承認%396EABE803BE
秀丸ヘルプ
下候補
単語をコピー
分割ウィンドウ切
切り抜き
コピー
貼り付け
タグジャンプ
強調表示の一
行番号表示/非

コード生成エラーを評価する

- Rational Roseでは、警告及びエラーはすべて「ログ」ウインドウに記録される。
- クラスの設計の一部が完全でない場合、Rational Roseは警告メッセージをログに書き込み、デフォルトの値を使用する。
- 繰り返し型の開発アプローチで開発を行う場合、1回のイテレーションで1つのクラス全体が実装されない場合もあるので、この機能が特に重要となる。

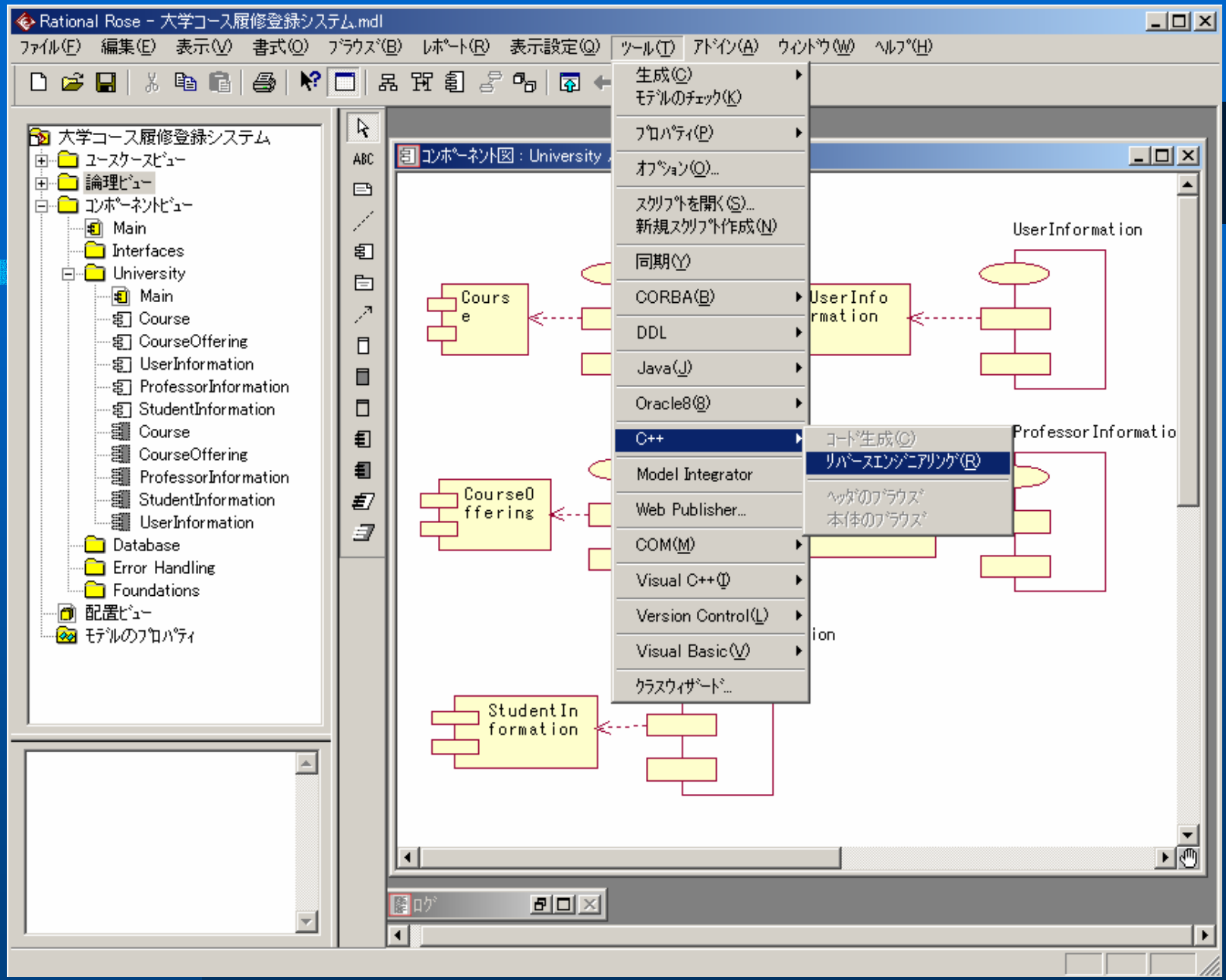
Reverse Engineering

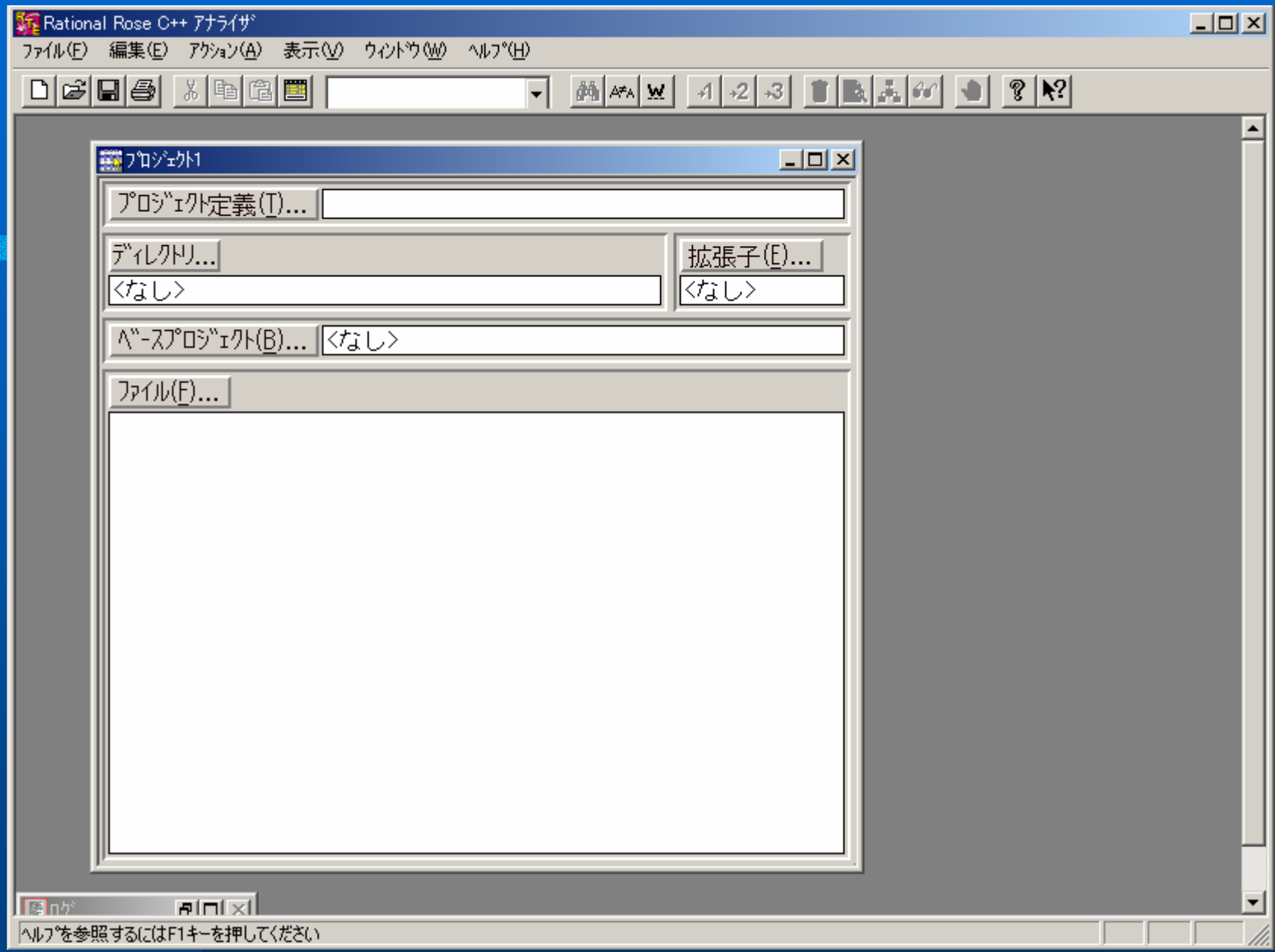
Reverse Engineering の手順

1. プロジェクトを作成する。
2. プロジェクト定義を追加する。
3. 参照ライブラリとベースプロジェクトを追加する。
4. ファイルタイプを設定、ファイルを解析する。
5. エラーを評価する。
6. エクスポートオプションを選択し、Roseにエクスポートする。
7. Roseモデルを更新する。

プロジェクトを作成する

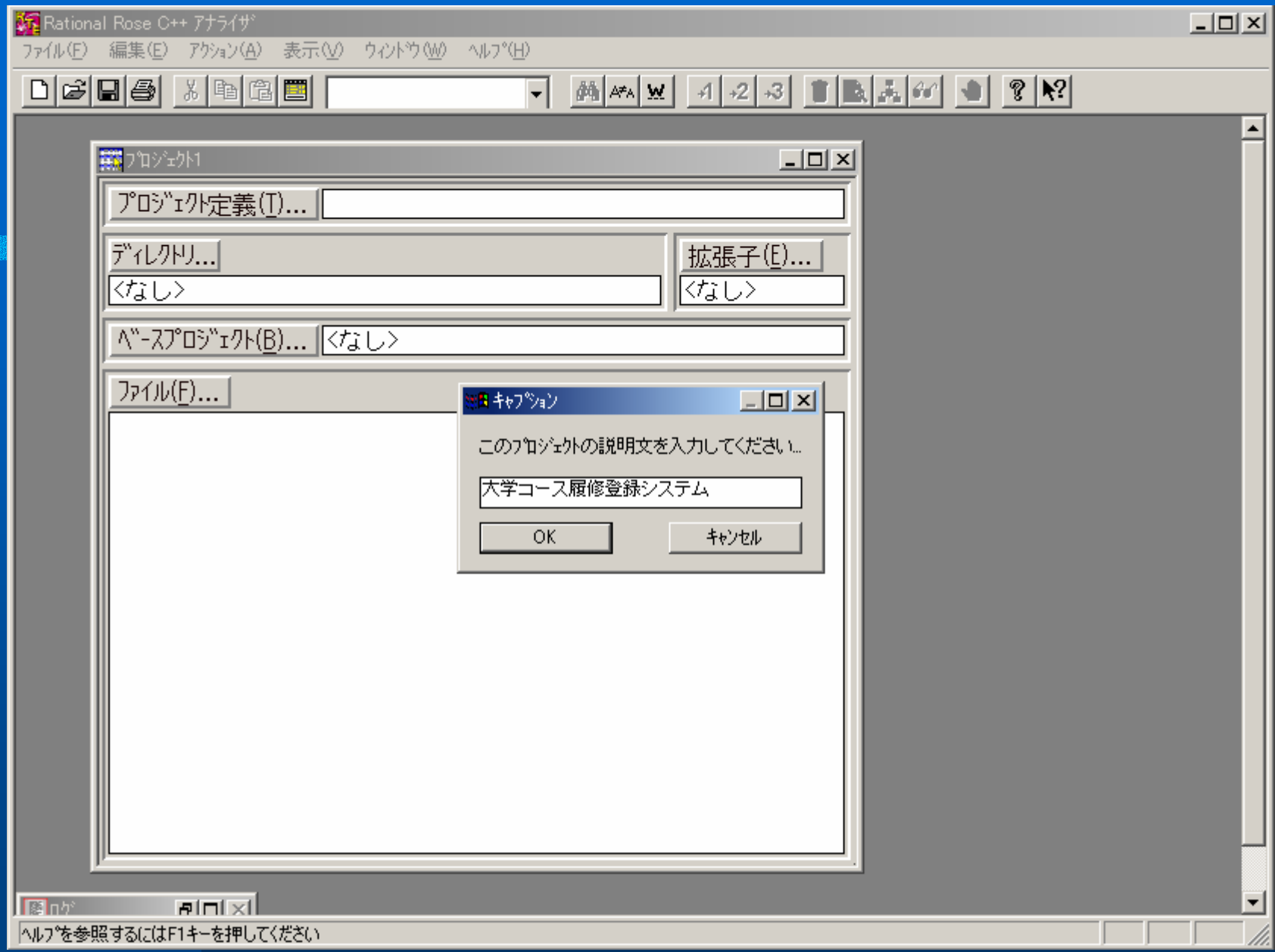
1. Roseで[ツール]メニューの[C++]をポイントし、[リバースエンジニアリング]をクリックする。C++アナライザーが起動する。
2. [ファイル]メニューの[新規作成]をクリックする。


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
大学コース履修登録システム
ユースケースビュー
論理ビュー
コンポーネントビュー
Main
Interfaces
University
Main
Course
CourseOffering
UserInformation
ProfessorInformation
StudentInformation
Course
CourseOffering
ProfessorInformation
StudentInformation
UserInformation
Database
Error Handling
Foundations
配置ビュー
モデルのプロパティ
コンポーネント図 : University
生成(C)
モデルのチェック(K)
プロパティ(P)
オプション(O)...
スクリプトを開く(S)...
新規スクリプト作成(N)
同期(Y)
CORBA(B)
DDL
Java(J)
Oracle8(8)
C++
コード生成(C)
リバースエンジニアリング(R)
ヘッダのブラウズ
本体のブラウズ
Model Integrator
Web Publisher...
COM(M)
Visual C++(I)
Version Control(L)
Visual Basic(V)
クラスウィザード...
UserInformation
Cours e
UserInfo rmation
CourseO ffering
StudentIn formation
ProfessorInformatio
ion
ログ


Rational Rose C++ アナライザ
ファイル(F) 編集(E) アクション(A) 表示(V) ウィンドウ(W) ヘルプ(H)
プロジェクト1
プロジェクト定義(T)...
ディレクトリ...
<なし>
拡張子(E)...
<なし>
ベースプロジェクト(B)...
<なし>
ファイル(F)...
ログ
ヘルプを参照するにはF1キーを押してください

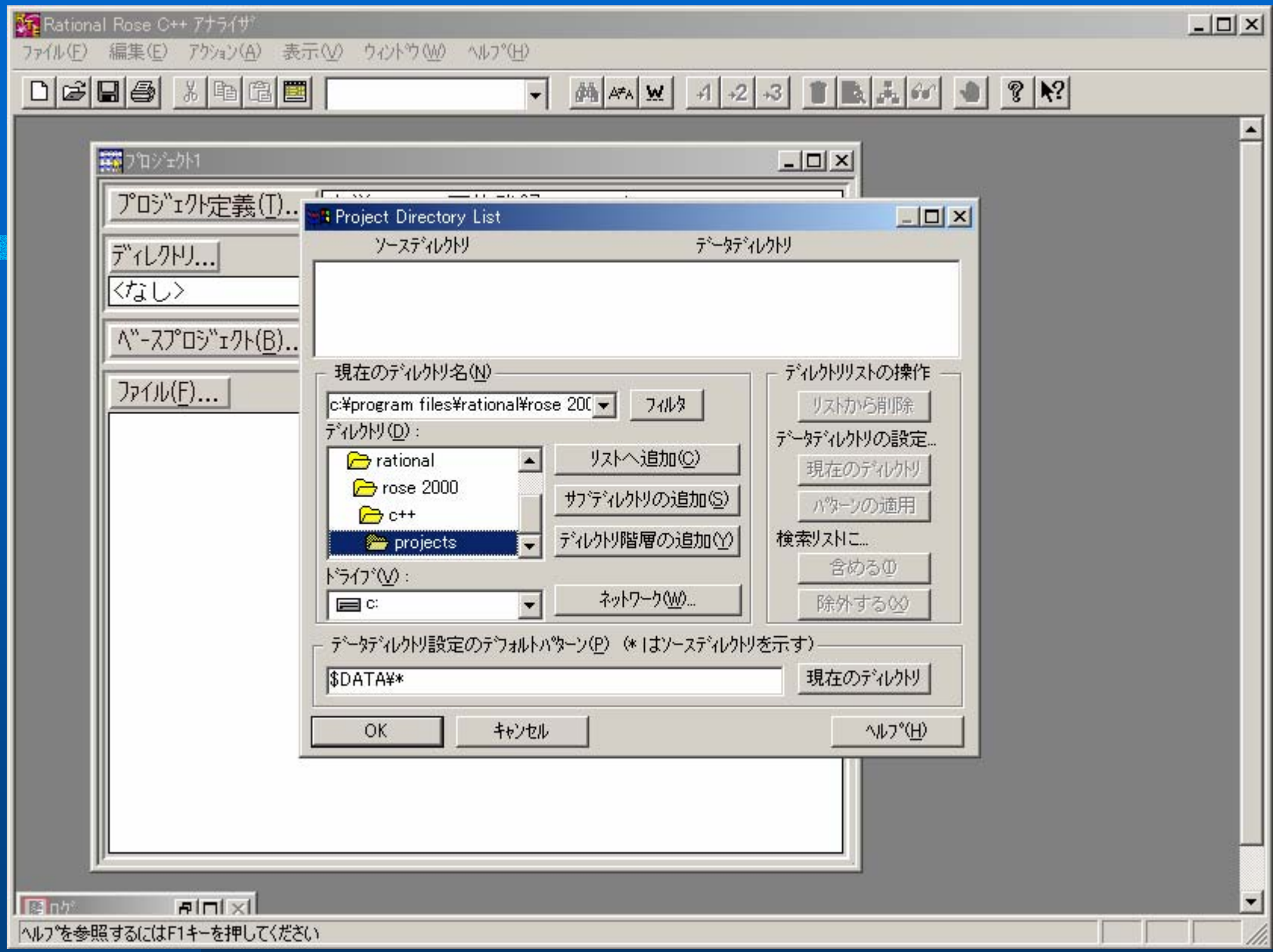
プロジェクト定義を追加する

1. [プロジェクト定義]をクリックして、[プロジェクト定義]ダイアログボックスを表示する。
2. [プロジェクト定義]ダイアログボックスに情報を入力する。
3. [OK]をクリックして、[プロジェクト定義]ダイアログボックスを閉じる。


Rational Rose C++ アナライザ
ファイル(F) 編集(E) アクション(A) 表示(V) ウィンドウ(W) ヘルプ(H)
プロジェクト1
プロジェクト定義(T)...
ディレクトリ...
<なし>
拡張子(E)...
<なし>
ベースプロジェクト(B)...
<なし>
ファイル(F)...
キャプション
このプロジェクトの説明文を入力してください...
大学コース履修登録システム
OK
キャンセル
ログ
ヘルプを参照するにはF1キーを押してください

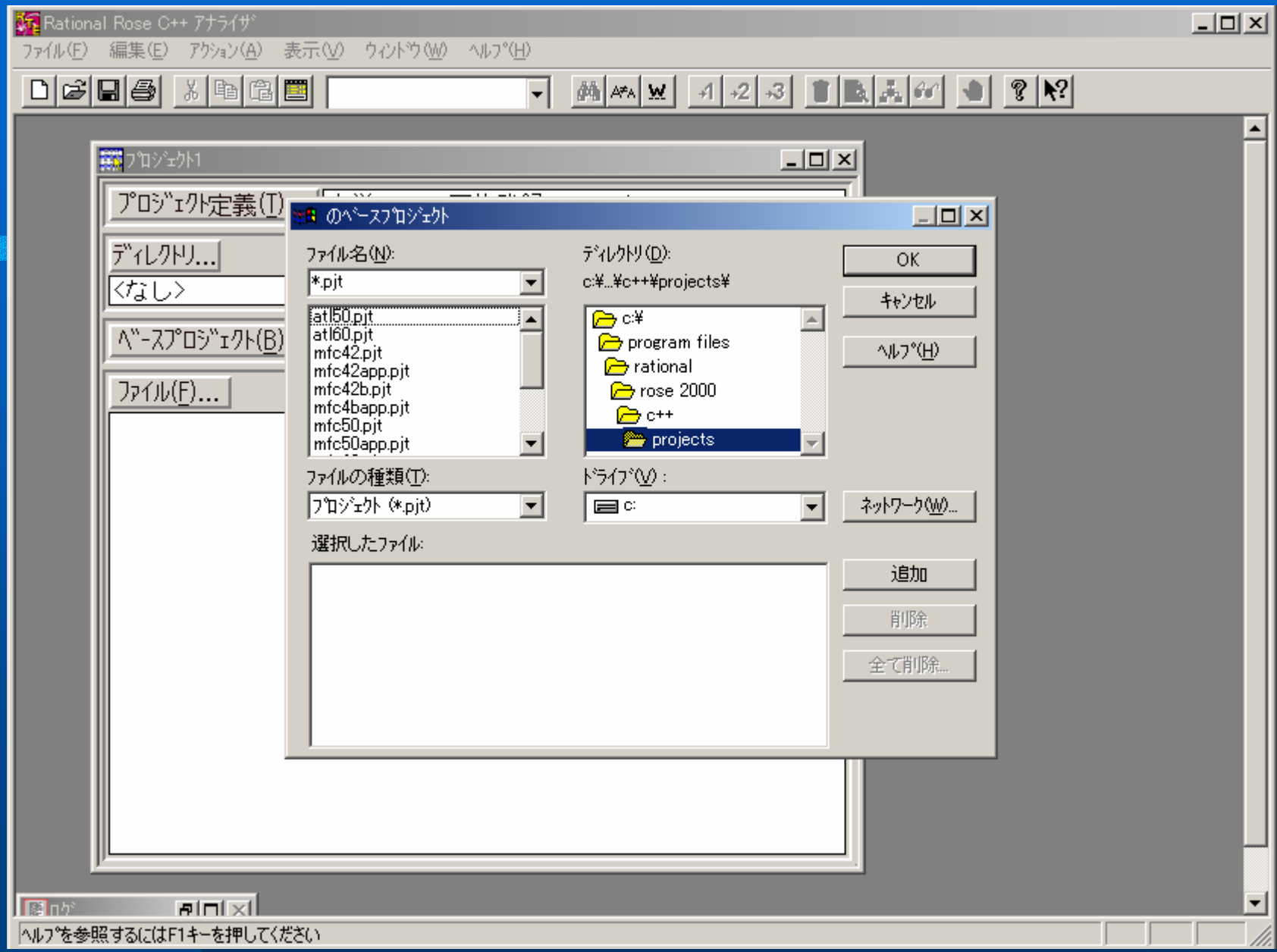
参照ライブラリとベースプロジェクトを追加する： ディレクトリリストの追加

1. [ディレクトリ]をクリックして、[ディレクトリリスト]ダイアログボックスを表示する。
2. [ディレクトリ]ボックスで、ディレクトリをクリックする。ディレクトリは現在のディレクトリに設定されている。
3. 現在のディレクトリをディレクトリリストに追加するには、[リストへ追加]をクリックする。
4. 現在のディレクトリとその直下にあるサブディレクトリだけをディレクトリリストに追加するには、[サブディレクトリの追加]をクリックする。
5. 現在のディレクトリとそれ以下のすべてのサブディレクトリをディレクトリリストに追加するには、[ディレクトリ階層の追加]をクリックする。


Rational Rose C++ アナライザ
ファイル(F) 編集(E) アクション(A) 表示(V) ウィンドウ(W) ヘルプ(H)
プロジェクト1
プロジェクト定義(T)...
ディレクトリ...
<なし>
ベースプロジェクト(B)...
ファイル(F)...
Project Directory List
ソースディレクトリ
データディレクトリ
現在のディレクトリ名(N)
c:¥program files¥rational¥rose 200
フィルタ
ディレクトリ(D):
rational
rose 2000
c++
projects
リストへ追加(C)
サブディレクトリの追加(S)
ディレクトリ階層の追加(Y)
ドライブ(V):
c:
ネットワーク(W)...
ディレクトリリストの操作
リストから削除
データディレクトリの設定...
現在のディレクトリ
パターンの適用
検索リストに...
含める(I)
除外する(X)
データディレクトリ設定のデフォルトパターン(P) (* はソースディレクトリを示す)
$DATA¥*
現在のディレクトリ
OK
キャンセル
ヘルプ(H)
ヘルプを参照するにはF1キーを押してください

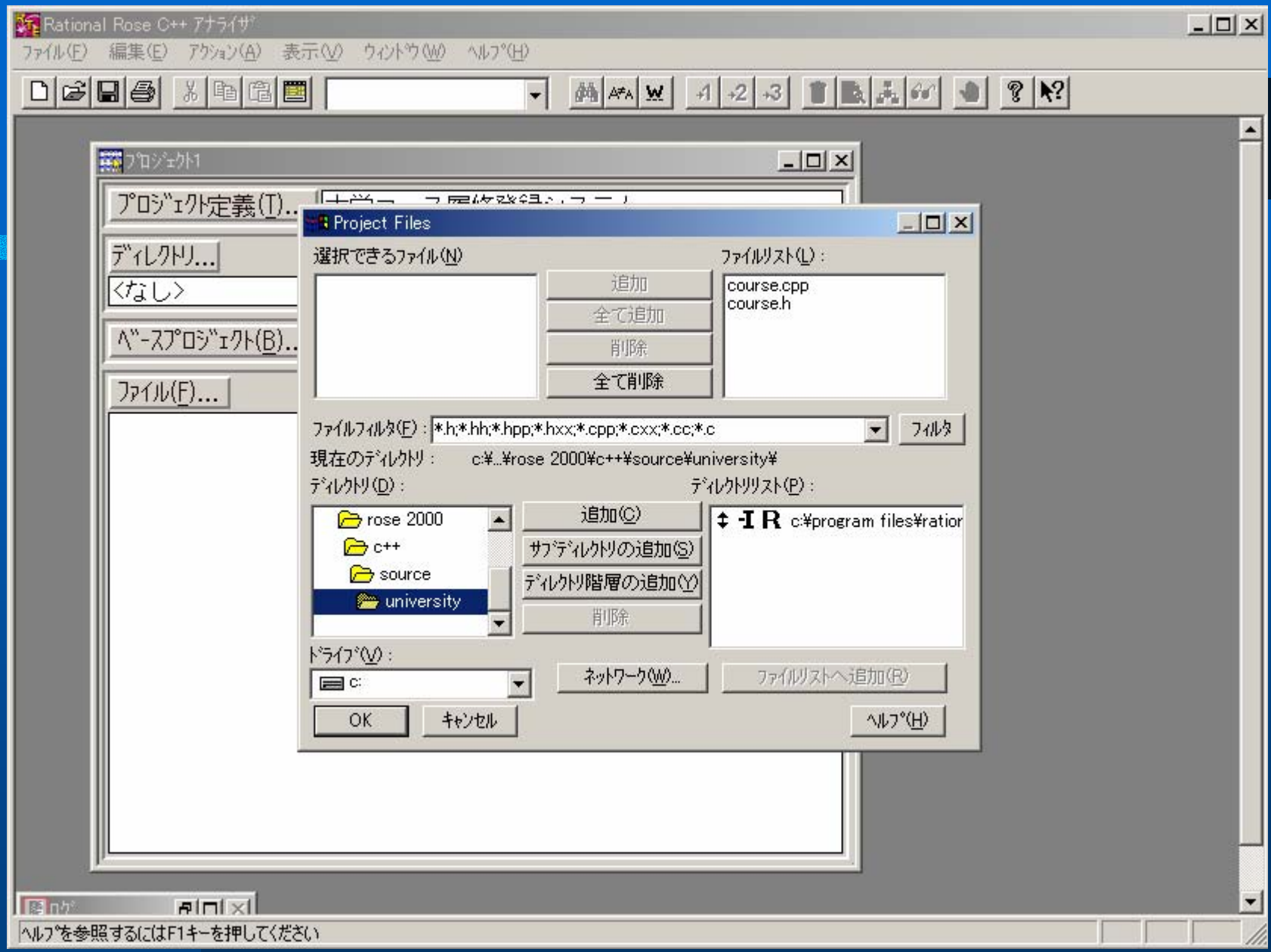
参照ライブラリとベースプロジェクトを追加する： ベースプロジェクトの追加

1. [ベースプロジェクト]をクリックして、[ベースプロジェクト]ダイアログボックスを表示する。
2. [ファイル名]ボックスに、使用するプロジェクト名が表示されるように、ディレクトリを移動する。
3. 追加するプロジェクトをクリックする。
4. [追加]をクリックして、ベースプロジェクトを追加する。


Rational Rose C++ アナライザ
ファイル(F) 編集(E) アクション(A) 表示(V) ウィンドウ(W) ヘルプ(H)
プロジェクト1
プロジェクト定義(T)...
ディレクトリ...
<なし>
ベースプロジェクト(B)
ファイル(F)...
のベースプロジェクト
ファイル名(N):
*.pjt
atl50.pjt
atl60.pjt
mfc42.pjt
mfc42app.pjt
mfc42b.pjt
mfc4bapp.pjt
mfc50.pjt
mfc50app.pjt
ディレクトリ(D):
c:¥...¥c++¥projects¥
c:¥
program files
rational
rose 2000
c++
projects
ファイルの種類(T):
プロジェクト (*.pjt)
ドライブ(V):
c:
選択したファイル:
OK
キャンセル
ヘルプ(H)
ネットワーク(W)...
追加
削除
全て削除...
ログ
ヘルプを参照するにはF1キーを押してください

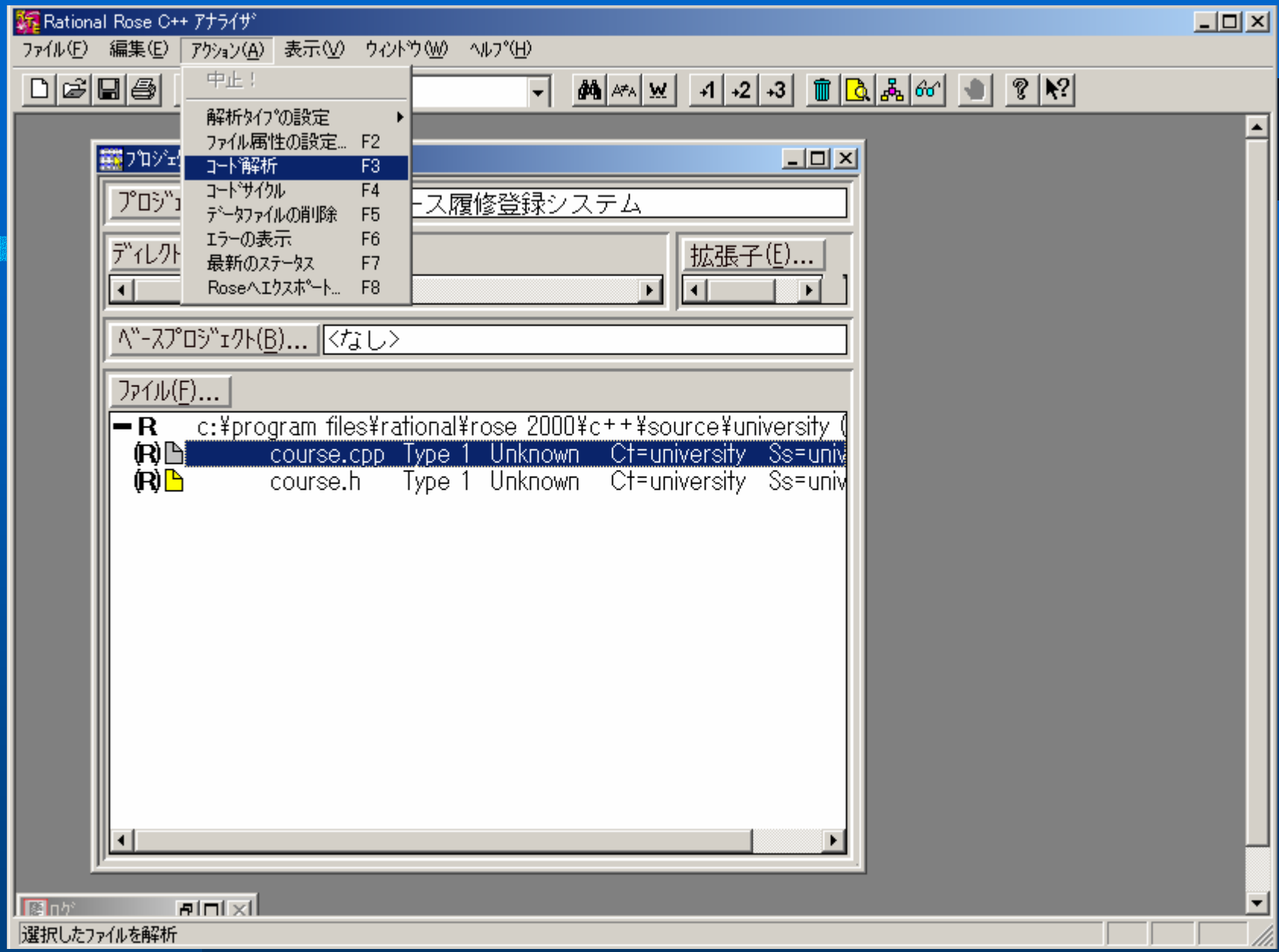
ファイルタイプを設定、ファイルを解析する： 解析タイプの変更

1. [アクション]メニューの[解析タイプの設定]をポイントし、適切なタイプをクリックする。
2. ファイルリストでファイルをクリックする。


Rational Rose C++ アナライザ
ファイル(F) 編集(E) アクション(A) 表示(V) ウィンドウ(W) ヘルプ(H)
プロジェクト1
プロジェクト定義(T)...
ディレクトリ...
<なし>
ベースプロジェクト(B)...
ファイル(F)...
Project Files
選択できるファイル(N)
追加
全て追加
削除
全て削除
ファイルリスト(L):
course.cpp
course.h
ファイルフィルタ(F): *.h;*.hh;*.hpp;*.hxx;*.cpp;*.cxx;*.cc;*.c
フィルタ
現在のディレクトリ: c:¥...¥rose 2000¥c++¥source¥university¥
ディレクトリ(D):
ディレクトリリスト(P):
rose 2000
c++
source
university
追加(C)
サブディレクトリの追加(S)
ディレクトリ階層の追加(Y)
削除
c:¥program files¥ratior
ドライブ(V):
c:
ネットワーク(W)...
ファイルリストへ追加(R)
OK
キャンセル
ヘルプ(H)
ヘルプを参照するにはF1キーを押してください

ファイルタイプを設定、ファイルを解析する： ファイルの解析

1. 解析する各ファイルに対して解析タイプを設定する。
2. ファイルリストでファイルをクリックする。
3. [アクション]メニューの[コード解析]をクリックして、ファイルを解析する。ファイルを解析して、Roseで必要となるアノテーションを挿入する場合は、[アクション]メニューの[コードサイクル]をクリックする。

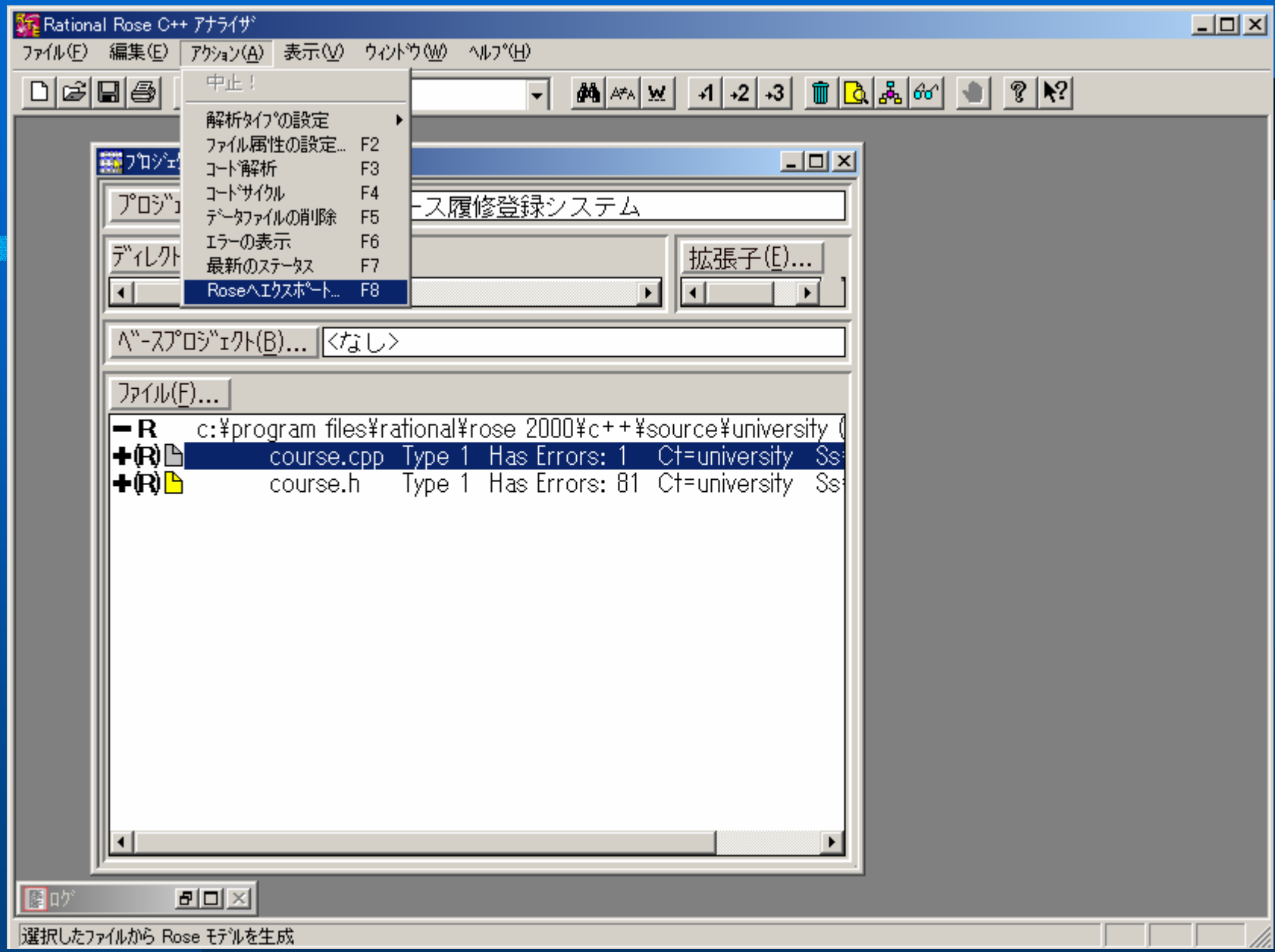

Rational Rose C++ アナライザ
ファイル(F) 編集(E) アクション(A) 表示(V) ウィンドウ(W) ヘルプ(H)
中止！
解析タイプの設定
ファイル属性の設定... F2
コード解析 F3
コードサイクル F4
データファイルの削除 F5
エラーの表示 F6
最新のステータス F7
Roseへエクスポート... F8
ース履修登録システム
拡張子(E)...
ベースプロジェクト(B)...
<なし>
ファイル(F)...
R c:¥program files¥rational¥rose 2000¥c++¥source¥university
course.cpp Type 1 Unknown Ct=university Ss=univ
course.h Type 1 Unknown Ct=university Ss=univ
選択したファイルを解析

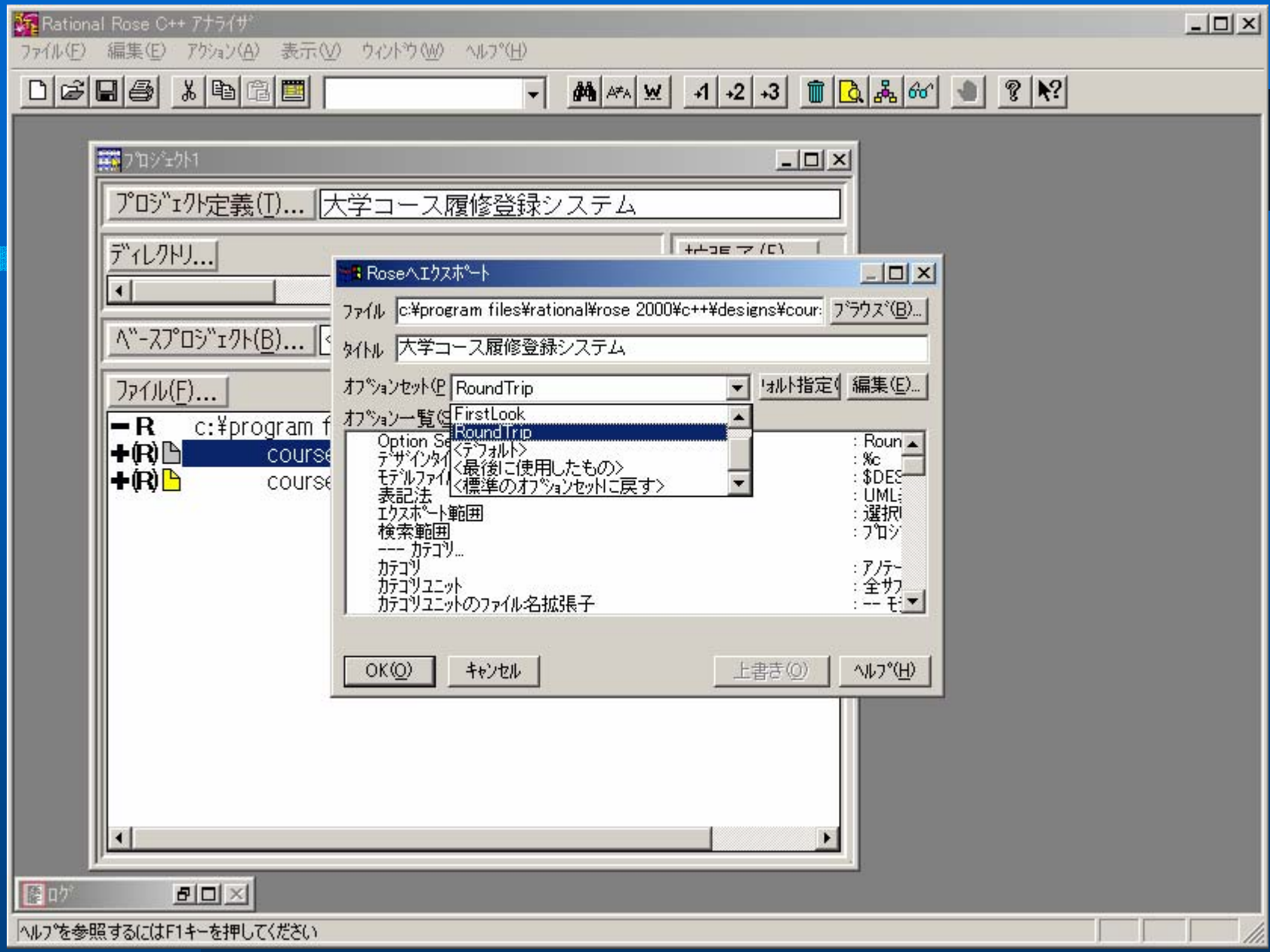
エラーを評価する

- アナライザーは、エラーを全て[ログ]ウインドウに記録する。
- ファイルリストのファイルをダブルクリックして、エラーを確認できる。

エクスポートオプションを選択し、Roseにエクスポートする

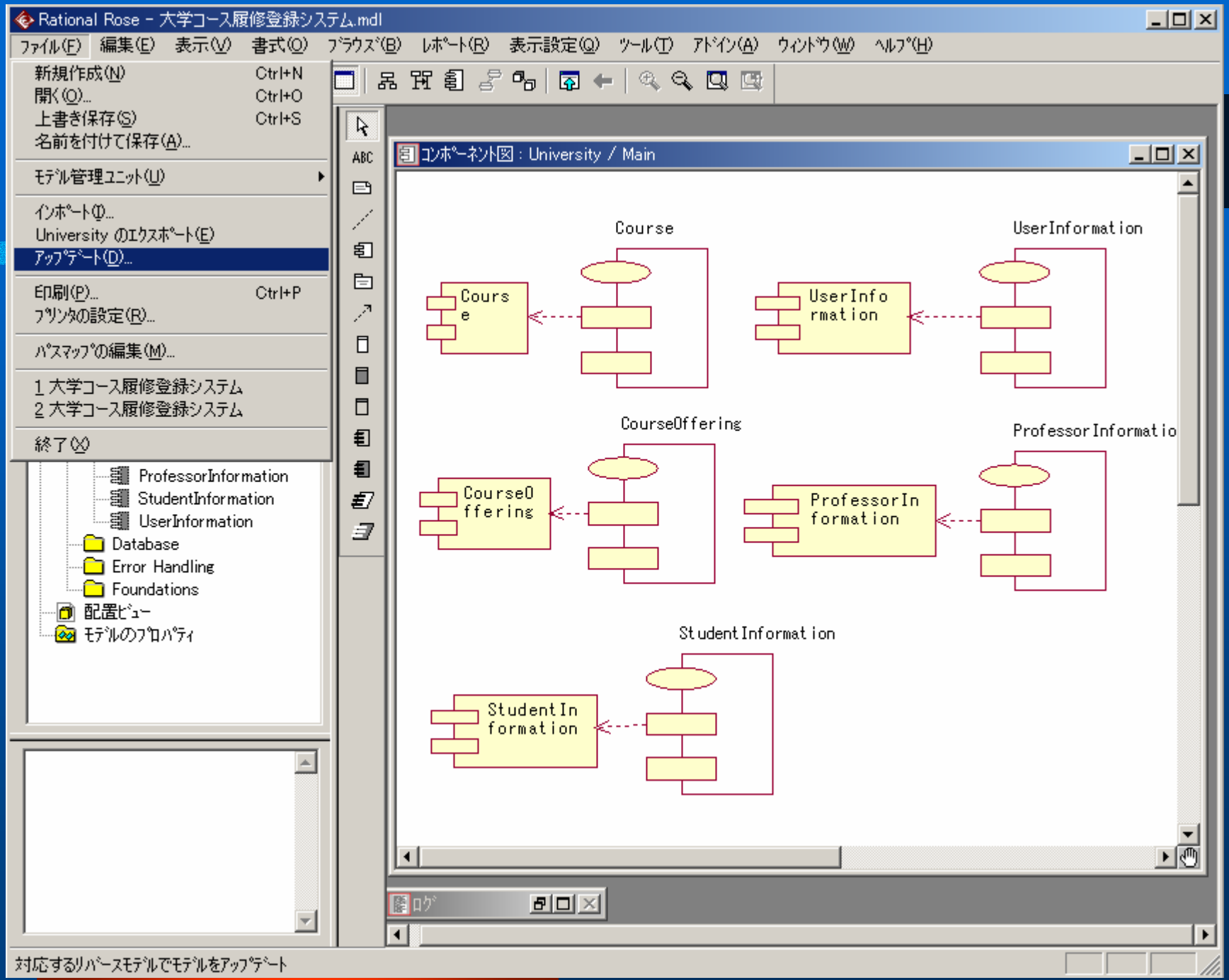
1. エクスポートするファイルをクリックする。
2. [アクション]メニューの[Roseへエクスポート]をクリックする。
3. [アクションセット]ボックスの矢印をクリックして、ドロップダウンリストを表示する。
4. 設定するエクスポートオプションをクリックする。
5. [OK]または[上書き]をクリックして、Roseにエクスポートする。


Rational Rose C++ アナライザ
ファイル(F) 編集(E) アクション(A) 表示(V) ウィンドウ(W) ヘルプ(H)
中止！
解析タイプの設定
ファイル属性の設定... F2
コード解析 F3
コードサイクル F4
データファイルの削除 F5
エラーの表示 F6
最新のステータス F7
Roseへエクスポート... F8
ース履修登録システム
拡張子(E)...
ベースプロジェクト(B)...
<なし>
ファイル(F)...
R c:¥program files¥rational¥rose 2000¥c++¥source¥university
course.cpp Type 1 Has Errors: 1 Ct=university
course.h Type 1 Has Errors: 81 Ct=university
ログ
選択したファイルから Rose モデルを生成


Rational Rose C++ アナライザ
ファイル(F) 編集(E) アクション(A) 表示(V) ウィンドウ(W) ヘルプ(H)
プロジェクト1
プロジェクト定義(T)...
大学コース履修登録システム
ディレクトリ...
ベースプロジェクト(B)...
ファイル(F)...
Roseへエクスポート
ファイル
c:¥program files¥rational¥rose 2000¥c++¥designs¥cour
ブラウズ(B)...
タイトル
大学コース履修登録システム
オプションセット(P
RoundTrip
編集(E)...
FirstLook
RoundTrip
<デフォルト>
<最後に使用したもの>
<標準のオプションセットに戻す>
Option Se
デザインタイ
モデルファイ
表記法
エクスポート範囲
検索範囲
--- カテゴリ...
カテゴリ
カテゴリユニット
カテゴリユニットのファイル名拡張子
OK(O)
キャンセル
上書き(O)
ヘルプ(H)
ログ
ヘルプを参照するにはF1キーを押してください

Roseモデルを更新する

1. 更新するRoseモデルを開く。
2. [ファイル]メニューの[アップデート]をクリックする。
3. ディレクトリを参照して、.red ファイルを検索する。
4. 使用する .redファイルをクリックする。
5. [OK]をクリックして、[モデルのアップデート]ダイアログボックスを閉じる。


Rational Rose - 大学コース履修登録システム.mdl
ファイル(F) 編集(E) 表示(V) 書式(O) ブラウズ(B) レポート(R) 表示設定(Q) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
新規作成(N) Ctrl+N
開く(O)... Ctrl+O
上書き保存(S) Ctrl+S
名前を付けて保存(A)...
モデル管理ユニット(U)
インポート(I)...
University のエクスポート(E)
アップデート(D)...
印刷(P)... Ctrl+P
プリンタの設定(R)...
パスマップの編集(M)...
1 大学コース履修登録システム
2 大学コース履修登録システム
終了(X)
ProfessorInformation
StudentInformation
UserInformation
Database
Error Handling
Foundations
配置ビュー
モデルのプロパティ
コンポーネント図 : University / Main
Course
Cours e
UserInformation
UserInfo rmation
CourseOffering
CourseO ffering
ProfessorInformatio
ProfessorIn formation
StudentInformation
StudentIn formation
ログ
対応するリバースモデルでモデルをアップデート