**Найден Ненков** **Валентин Вичев**

# Р Ъ К О В О Д С Т В О

## за упражнения

## по „Софтуерни технологии и производства”

**Шуменски университет “Епископ Константин Преславски”**
**Факултет по математика и информатика**

**Шумен**
**2007**

# СЪДЪРЖАНИЕ

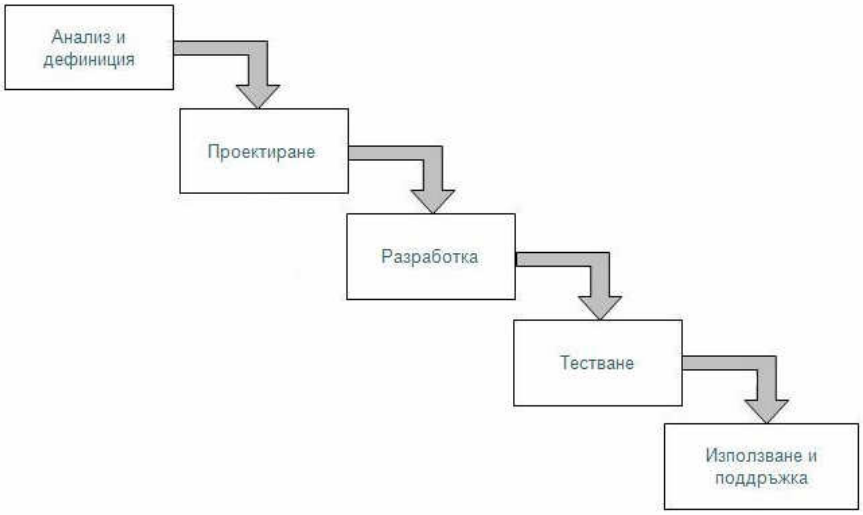
# 1. Моделиране на софтуерния процес.

## 1.1. Основни понятия

Какво е софтуерният процес на разработка?

Софтуерен развоен процес (software development process) е цялостният процес на поставяне на задачата, планиране, реализиране и оценка на едно софтуерно и хардуерно приложение, включително и използваните помощни средства, методи и необходимия персонал.

На фиг 1.1 са показани основните фази на софтуерния процес.




**Фиг 1.1.** Фази на софтуерния процес

Има различни модели на софтуерния процес, които се разглеждат в посочената литература в края. В по-нантъшното изложение главно внимание се отделя на каскадния модел и обектноориентирания подход за проектиране.

Унифицираният език за моделиране (**UML** –Unified Modeling Language) представлява фамилия от графични нотации, обединени от общ мета-модел, които помагат при описването и проектирането на софтуерни системи, по-специално обектно-ориентирани. UML е стандарт, контролиран от

отворения консорциум от компании Object Management Group (OMG), който е разработил CORBA (Common Object Request Broker Architecture) стандартите в тази пбласт. Пълното описание може да се намери на **http://www.uml.org/**.

Unified Modeling Language (UML) е графичен език за даване на ясна зрителна представа, определение, създание, и документиране на артефактите на софтуерната система. UML дава възможност за изразяване както на абстрактни така и на конкретни неща, както например, класове написани на специфичният език за програмиране, схеми на базата от данни , и компоненти на софтуера за многократно ползване. При софтуерните проекти UML се използва за:

- визуализиране
- изясняване (на детайли в проекта)
- конструиране (дава възможност за генериране на програмен код - основата на проекта)
- документация

## 1.2. Концептуален модел на UML

Концептуалния модел на езика изисква изучаване на три главни елемента: **UML** изграждащи блокове, правилата за връзка между тези блокове, и някои общи механизми които се прилагат в **UML**. Разбирането на тези идеи позволява разбирането на **UML** моделите и възможност за създаване на някой основни неща.

Изграждащи блокове на **UML** - те са три вида:

1. Инструменти
2. Връзки (отношения)
3. Диаграми

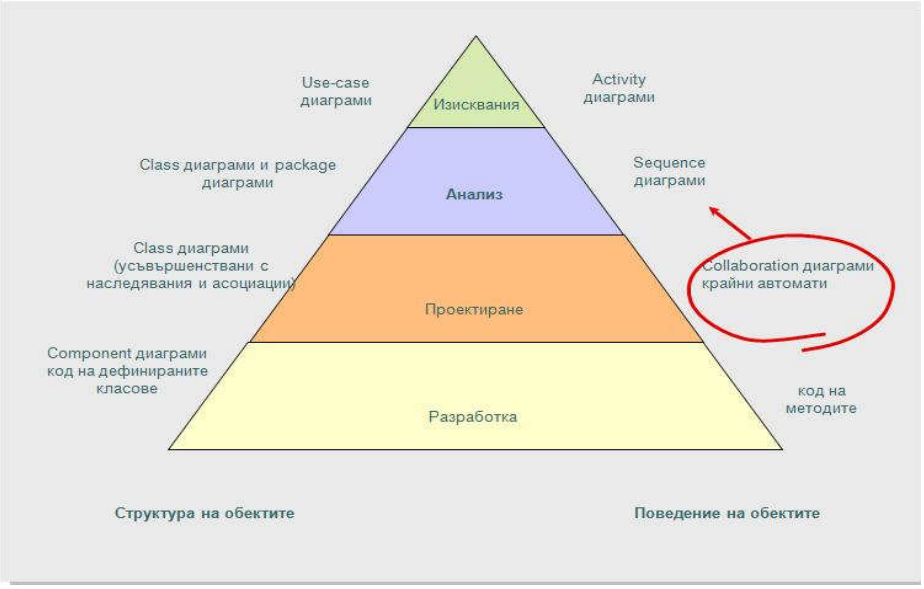
Инструментите са първостепенни в модела. Връзките ги свързват помежду им. Диаграмите групират инструментите.

Инструментите биват четири вида:

1. Структурни инструменти
2. Поведенчески инструменти

3. Групиращи инструменти
4. Обясняващи инструменти

Това са базовите изграждащи блокове, служещи за правилно моделиране в **UML**. На фиг.1.2. е показно приложението на UML диаграмите във фазите на софтуерен процес.




**Фиг.1.2.** Използване на UML диаграмите при софтуерния процес

## 1.3. CASE среда RATIONAL ROSE

Rational Rose е продукт, разработен от IBM с цел да автоматизира обектно-ориентираната технология за анализ и проектиране на софтуерни системи. Той осигурява поддръжка на два съществени елемента на съвременното софтуерно инженерство: компонентно-базирано развитие и контролирано итеративно развитие. За работа е необходимо съобразяване със следните минимални хардуерни и софтуерни изисквания:

Наличието на всякаква Pentium-базирана PC-съвместима система със:

- SVGA-съвместим дисплей;
- Всякакво посочващо устройство (мишка) с минимум два бутона;
- Microsoft Windows NT 4.0 (SP6a) или Windows 2000 (SP2 or 3);
- 64 MB RAM минимум, като се препоръчва 128 MB;
- 255 MB дисково пространство;
- Microsoft Internet Explorer 5.01 със SP2 или по-късен;

- Netscape Navigator версия 4.72 до 4.77 и 7.0;
- Препоръчва се твърдо използването на RUP (Rational Unified Process) с най-актуалната версия на web броузъра, като RUP изисква Microsoft Internet Explorer 4.0 със SP1 или следващ.

Интерфейсът на Rational Rose (RR) притежава пет основни елемента:

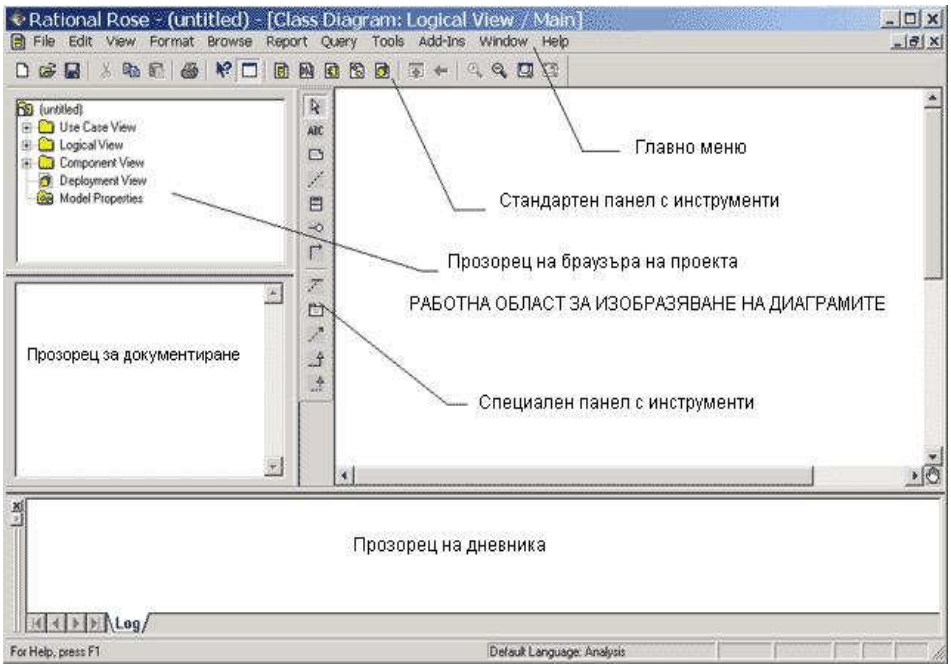
- Браузър (brouser) използва се за бърза навигация по модела;

- прозорец на документа (documentation window) – текстово описание на елементите на модела;

- панел с инструменти (tollbars) – използва се за бърз достъп до най-разпространените документи;

- прозорец на диаграмата – работно поле, в което се изобразяват диаграните на модела;

- log (описание, дневник- log) – използва се при проверка на грешки и създаване на отчети за резултатите при изпълнението на различни команди. На фиг. 1.3 са показани различните части на интерфейса на RR.




**Фиг. 1.3**. Интерфейс на Rational Rose

### 1.3.1. Браузър

Браузърът представлява йерархична структура, която позволява да се извършва навигация по модела. Всичко, което се добавя към тази структура, а именно – действащи лица, варианти на използване, класове, компоненти – се показва в прозореца на браузъра.

С помощта на браузъра може да се:

- добавят към модела елементи (действащи лица, варианти на използване, компоненти, диаграми и т.н.);
- разглеждат съществуващите елементи на модела;
- разглеждат съществуващите връзки между елементите на модела;
- преместват елементите на модела;
- преименуват тези елементи;
- добавят елементи на модела към диаграмата;
- свързват елемент с файл или адрес в Интернет;
- групират елементи в папки;
- работи с детайлизирани спецификации на елемента;
- отваря диаграмата.

Браузърът поддържа четири ***представяния (view***):

- представяне на вариантите на използване;
- представяне на компонентите;
- представяне на разполагането;
- логическо представяне.

Браузърът е организиран като дървовидна структура. Всеки елемент на модела може да съдържа други елементи, които се намират по-ниско в йерархията. Знакът “-“ около елемента показва, че неговият клон е отворен до край. Знакът “+” означава, че клонът е свит.

### 1.3.2. Прозорец за документиране

С негова помощ е възможно документирането на елементи на модела в RR. Например, може да се направи кратко описание на всяко действащо лице. При документиране на клас, всичко написано в прозореца на документацията

ще се появи по-късно като коментар в генерирания код, което пък изключва необходимостта от включването на тези коментари допълнително ръчно. Документацията също така се извежда и в отчетите, които се създават в средата RR.

### 1.3.3. Панели с инструменти

Панелите с инструменти в RR осигуряват бърз достъп до най-разпространените команди. В тази среда съществуват два вида панели с инструменти: стандартен панел и панел на диаграмата.

***Стандартният панел*** се вижда винаги. Кликването върху бутоните съответства на команди, които могат да се използват за работа с всяка диаграма.

***Панелът на диаграмата*** е уникален за всеки тип диаграми в UML.

Всички панели с инструменти могат да се променят и настройват от потребителя. За това е необходимо да се избере менюто ***Tools -> Options*** и след това да се избере ***Toolbars***.

За да *се покаже или скрие стандартният панел* с инструментите (или пък панелът с инструментите на диаграмата) се прави следното:

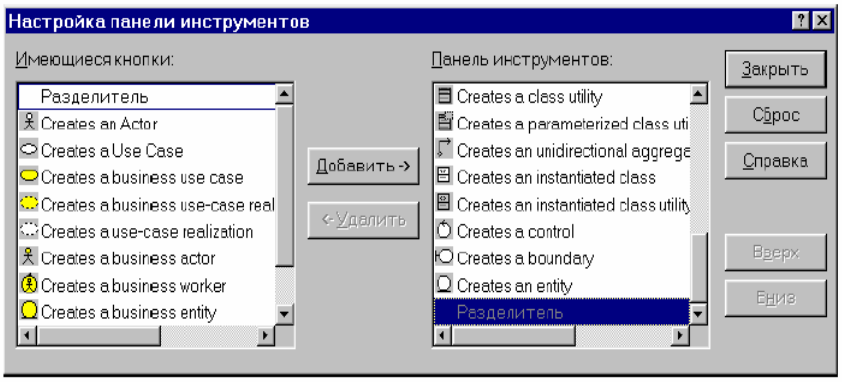
1. Избирате ***Tools -> Options***.
2. Избира се ***Toolbars***.
3. За да се направи видим или невидим стандартния панел с инструментите отбележете (или премахнете отбелязаното) контролния превключвател ***Show Standard Toolbar*** (или ***Show Diagram Toolbar***).

За да се *увеличи размера на бутоните* от панела с инструменти:

4. Кликнете с десен бутон на мишката върху дадения панел.
5. Изберете в появилото се меню ***Use Large Buttons***.

За да се *настрои* панелът с инструменти:

1. Кликнете с десен бутон на мишката върху дадения панел.
2. Изберете Customize (настройване).
3. За да се добави или премахне бутон, изберете съответния бутон и след това кликнете с мишката върху ***Add*** (добавяне) или ***Remove*** (премахване), както е показано на фиг. 1.4.




**Фиг. 1.4.** Настройки на стандартния модел от инструменти

Съществуват два начина за премахване на елементи на модела – от една диаграма или от целия модел.

За да се *премахне елемент* на модела от диаграмата:

1. Маркирайте елемента на диаграмата.
2. Натиснете бутона ***Delete***.
3. Обърнете внимание, че въпреки че този елемент е премахнат от диаграмата, то той остава в браузъра и на другите диаграми на системата.

За да се *премахне елемент* от модела:

1. Маркирайте елемента от диаграмата.
2. Изберете от менюто ***Edit -> Delete from Model*** или натиснете едновременно бутоните ***CTRL+D***.

### 1.3.4. Прозорец на диаграмата

В прозореца на диаграмата се вижда, как изглежда една или няколко диаграми в UML модела. При внасяне на изменения в елементите на диаграмата RR автоматично обновява браузъра. Аналогично при внасяне на изменения в даден елемент с помощта на браузъра, RR автоматично обновява съответните диаграми. Това помага да се поддържа модела в непротиворечиво състояние.

### 1.3.5. Описание (дневник)

Дневникът съхранява определена информация, която се изпраща в него при работа върху даден модел. Например, там се съхраняват съобщения за

грешки, възникнали при генериране на кода. Няма начин дневникът да бъде премахнат напълно, но неговият прозорец може да се минимизира.

### 1.3.6. Четири изгледа на модела в RR

В модела RR се поддържат четири изгледа (***views***) на: варианти на използване; логиката; компонентите; разполагането.

- **Изглед на вариантите за използване**

Този изглед съдържа модела на бизнес – процесите и модела на вариантите на използване.

- **Логически изглед**

Логическият изглед се концентрира върху това, как системата ще реализира поведението, описано във вариантите на използване. То дава подробна картина на съставните части на системата и описва взаимодействието между тези части. Логическото представяне включва основно класове и диаграми на класовете. С тяхна помощ се конструира детайлния проект на създаваната система. Основните му компоненти са:

- ***класове***;
- ***диаграми на класовете*** - като правило, за описание на една система се използват няколко диаграми на класовете, всяка от които отразява някакво подмножество на всички класове на системата;
- ***диаграми на взаимодействието***, използва се за представяне на обектите, участващи в едно от събитията на варианти на използване;
- ***диаграми на състоянията***;
- папки, представляващи ***групи от взаимосвързани класове***.

- **Изглед на компонентите**

Тук се разположени:

- компоненти, представляващи физически модули на кода;
- диаграми на компонентите;
- папки, представляващи групи от взаимосвързани компоненти.

**Изглед на разполагането (развръшането)**

Последният изглед в RR съответства на физическото разполагане на системата, което може да се различава от нейната логическа структура. Тук се включват:

- ***процеси***, представляващи потоци (***threads***), изпълнявани в отделена за тях област от паметта;
- ***процесори***, включващи произволни компютри, които обработват данни. Кой да е процес може да се изпълнява от един или няколко процесора;
- ***устройства*** или ***апаратура***, които не обработват данни като такива устройства могат да са терминали за въвеждане – извеждане и принтери.
- ***диаграма на разполагането***.

### 1.3.7. Параметри за настройка на изобразяването

В RR има възможност да се настройват диаграмите така, че:

- да се покажат всички атрибути и операции;
- да се скриват операции;
- да се скриват атрибути;
- да се покажат само някои атрибути или операции;
- да се покажат операциите заедно с техните пълни сигнатури или само техните имена;
- да се показват или да не се показват видимо атрибутите и операциите;
- да се показват или да не се показват стереотипите на атрибутите и операциите.

Значенията на всеки параметър по премълчаване може да се задават с помощта на прозорец, който се отваря при избор от менюто на ***Tools -> Options***. За всеки даден клас на диаграмата може да се: покажат всички атрибути; скрият всички атрибути; покажат само избраните от вас атрибути; потисне извеждането на атрибутите

Потискането на извеждане на атрибутите ще доведе не само до изчезване на атрибутите от диаграмата, но и до отстраняване на линията, показваща месторазположението на атрибутите в класа.

Съществуват два начина за изменение на параметрите за представяне на параметрите в диаграмата. Необходимите значения може да се определят за всеки клас индивидуално. Също така може да се променят значенията на необходимите параметри по премълчаване до началото на създаване диаграмата на класовете. Внесените по този начин изменения ще повлияят само на новосъздаваната диаграма.

За да се покажат ***всички атрибути на класа*** трябва:

1. Да се маркира в диаграмата дадения клас.
2. Да се кликне върху него с десен бутон на мишката и да се отвори контекстно-зависимото меню.
3. Изберете в него Optins -> Show All Attributes.

За да се покажат ***само избрани атрибути*** трябва:

1. Да се маркира в диаграмата дадения клас.
2. Да се кликне върху него с десен бутон на мишката и да се отвори контекстно-зависимото меню.
3. Изберете в него ***Options -> Selact Compartment Items***.
4. Посочете дадените атрибути в прозореца ***Edit Compartment***.

За да се ***потисне извеждането на всички атрибути на даден клас*** от диаграмата трябва:

1. Да се маркира в диаграмата дадения клас.
2. Да се кликне върху него с десен бутон на мишката и да се отвори контекстно-зависимото меню.
3. Изберете в него ***Options -> Suppres Attributes***.

За да се ***промени приетия по премълчаване вид на атрибута***:

1. В менюто на модела изберете ***Tools -> Options***.
2. Преминете към поле ***Diagram***.

3. За да въведете значения на параметрите на атрибутите по премълчаване, използвайте контролните превключватели ***Suppres Attributes и Show All Attributes***.

Промяната на тези значения по премълчаване ще повлияе само на новите диаграми. Видът на вече съществуващите диаграми на класовете няма да се промени. Както и при случая с атрибутите, има ***няколко варианта на представянето на операциите*** на диаграмите.

- Да се покажат всички операции.
- Да се покажат само някои операции.
- Да се скрият всички операции.
- Да се потисне извеждането на операциите.

Освен това е възможно:

- Да се покаже само името на операцията. Това означава, че на диаграмата ще бъде представено само името на операцията, но не и аргументите или типа на възстановяваното значение;
- Да се покаже пълната сигнатура (характеристика) на операцията. На диаграмата ще бъде представено не само името на операцията, но всичките и параметри, типа на тези параметри и типа на възстановяваното значение на операцията.

За да бъдат показани ***всички операции на класа*** трябва:

1. Да се маркира на диаграмата желания от вас клас.
2. Кликнете върху него с десен бутон на мишката, за да се отвори контекстно-зависимото меню.
3. Изберете в това меню ***Options -> Show All Operations***.

За да се покажат ***само избраните операции от класа***:

1. Маркирайте от диаграмата желания от вас клас.
2. Кликнете върху него с десен бутон на мишката, за да се отвори контекстно-зависимото меню.
3. Изберете в това меню ***Options -> Select Compartment Items***.
4. Посочете нужните Ви операции в прозореца Edit Compartment.

За да се ***потисне извеждането на всички операции на класа*** от диаграмата:

1. Маркирайте от диаграмата желания от вас клас.
2. Кликнете върху него с десен бутон на мишката, за да се отвори контекстно-зависимото меню.
3. Изберете в това меню ***Options -> Suppress Operations***.

За да се ***покаже от диаграмата на класовете*** сигнатурата на операцията:

1. Маркирайте от диаграмата желания от вас клас.
2. Кликнете върху него с десен бутон на мишката, за да се отвори контекстно-зависимото меню.
3. Изберете в това меню ***Options -> Show Operation Signature***.

За да се ***промени приетия по премълчаване вид на операцията***:

1. В менюто на модела изберете ***Tools -> Options***.
2. Преминете към разширението ***Diagram***.
3. За установяване на значения на параметрите в дадената операция по премълчаване използвайте контролните превключватели ***Suppress Operations, Show All Operations и Show Operation Signatures***.

За да се покаже ***изгледа на атрибута или операцията*** от класа:

1. Маркирайте от диаграмата желания от вас клас.
2. Кликнете върху него с десен бутон на мишката, за да се отвори контекстно-зависимото меню.
3. Изберете в това меню ***Options -> Show Visibility***.

За ***превключване на видимите нотации*** на Rose и UML:

1. В менюто на модела изберете ***Tools -> Options***.
2. Преминете към разширението ***Notation***.
3. За превключване между нотациите използвайте превключвателя ***Visibility as Icons***.

Ако този превключвател е маркиран, ще бъде използвана нотация на RR. Ако пък не е маркиран, ще бъде използвана нотация на UML.

Изменението на този параметър ще се отрази само на новите диаграми. Съществуващите вече диаграми на класовете няма вече да се променят.

# 2. Реализиране на учебен проект.

## 2.1 Моделиране на бизнес-процеси.

***Постановка на задачата.***

Пред ръководителя на информационния отдел на университета поставена задача да се разработи автоматизирана система за регистрация на студентите от платеното обучение. Системата трябва да позволява студентите сами да се регистрират за дадена специалност и да могат самостоятелно да следят за своята успех от ПК, включени в локалната университетска мрежа. Лекторите по всяка дисциплина трябва да имат достъп до системата, да могат да посочват дисциплините, по които ще четат лекции и да внасят съответно оценки на студентите по тези дисциплини.

Сега в университета съществува база данни (БД), която съдържа пълна информация за преподаваните дисциплини. Регистрирането на дисциплините става по следния начин: в началото на всеки семестър студентите могат да поискат учебния план с дисциплините (задължителни, избираеми и факултативни), които ще се изучават през този семестър.

Информацията за всяка дисциплина трябва да включва името на лектора, катедрата и изискванията по предварителната подготовка (какви дисциплини трябва да са преминати успешно предварително).

Всеки студент може да избира по 4 дисциплини за всеки семестър. Допълнително всеки студент може да посочи и 2 алтернативни дисциплини в случай, че се окаже че някоя от избраните дисциплини е вече попълнена или пък няма да се чете. За всяка дисциплина могат да се запишат не повече от 10 и не по-малко от 3-ма студента, (ако са по-малко от 3, то дисциплината няма да се чете). През всеки семестър има

период от време, когато студентите могат да променят своите планове (да добавят или да се откажат от своята дисциплина). След като е завършен процесът на регистриране на студентите, тогава административен служител изпраща информацията във финансов отдел, където всеки студент може да внесе необходимата сума за дадения семестър. Ако дисциплината се окаже попълнена вече в процеса на регистриране, то студентът трябва да бъде уведомен за това до окончателно формиране на неговия личен учебен план. В края на семестъра всеки студент може да получи информация за своята успех.

**2.2 Създаване на модела на варианти на използване.**

Действащи лица (bisiness actors):

- Студент – избира дисциплини и проверява своята успех;
- Лектор – предлага дисциплини и оценява студентите;
- Финансов отдел – получава информация за плащане;
- Списък на дисциплините – БД, съдържаща информация за отделните дисциплини.

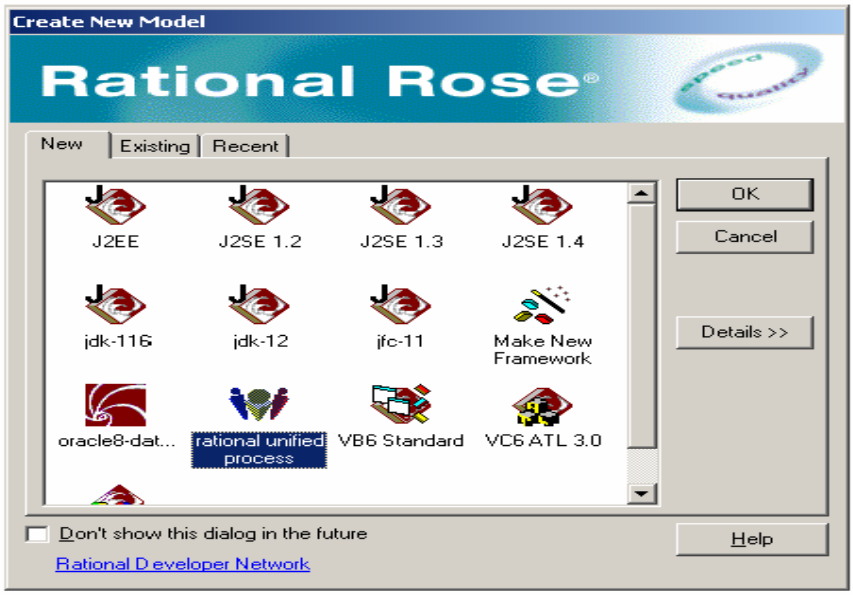
***Упражнение 1. Създаване на действащите лица в Rational Rose.***

При стартирането на Rational Rose в прозореца Create New Model изберете варианта Rational Unified Process (фиг. 2.1). Като резултат на екрана ще се появи фиг. 1.1.

За да се регистрира действащо лице в браузъра:

1. Кликнете с десен бутон на мишката върху папка Business Use Case Model от папка Use Case View на браузъра.
2. Изберете от отвореното меню New -> Actor.
3. В браузъра ще се появи ново действащо лице с име NewClass. Вляво от това име вие ще видите пиктограмата на действащото лице UML.
4. Маркирайте новото действащо лице и въведете името му.
5. Кликнете с десен бутон на мишката върху действащото лице.

6. В отвореното меню изберете Open Specification.
7. В полето на стереотипа изберете Business Actor и натиснете бутона OK.
8. След създаването на действащите лица запазете модела под името courseregs (analysis) с помощта на менюто File -> Save.




фиг. 2.1. Прозорец за избор на шаблонни модели

***<u>Варианти на използване :</u>***

Изхождайки от потребностите на действащите лица са възможни следните варианти на използване (Business Use Case):

- регистриране на дадени дисциплини;
- запознаване с успеваемостта;
- избиране на дисциплини, които ще бъдат преподавани;
- оценяване.

***<u>Упражнение 2. Създаване на варианти на използване в Rational Rose.</u>***

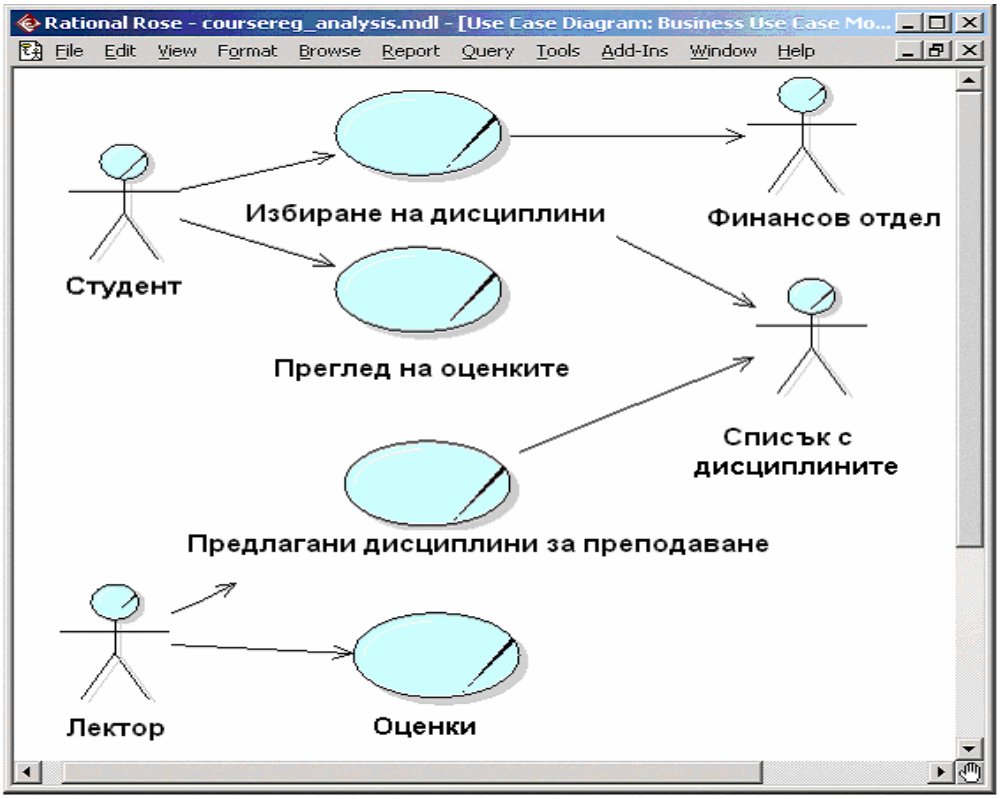
За да се разположи даден вариант на използване в браузъра:

1. Кликнете с десен бутон на мишката върху папка Business Use Case Model от главната папка Use Case View на браузъра.

2. От появилото се меню изберете New -> Use Case.
3. Новият вариант на използване ще се появи в браузъра под името NewUseCase. В ляво от него ще видите пиктограмата на варианта на използване в UML.
4. Маркирайте този нов вариант на използване и въведете името му.
5. Кликнете с десен бутон на мишката върху варианта на използване.
6. В отвореното меню изберете Open Specification.
7. В полето на стереотипа изберете Business Use Case и натиснете бутона OK.

Диаграма на вариантите на използване:

Създайте диаграма на вариантите за използване за бизнес-модела "Система за регистрация". Необходимите за това действия са подробно изброени по долу. Готовата диаграма трябва да изглежда, както е показано на фиг. 2.2.




фиг. 2.2. Диаграма на вариантите на използване

***Упражнение 3. Построяване на диаграмата на вариантите за използване***

За създаване на нова диаграма на вариантите за използване:

1. Кликнете с десен бутон на мишката върху папка Business Use Case Model от главната папка Use Case View на браузъра.
2. От падащото меню изберете New -> Use Case Diagram.
3. Маркирайте новата диаграма и въведете името и (Business Use Case Diagram).
4. Кликнете два пъти върху името на тази диаграма с ляв бутон на мишката, за да я отворите.
5. За да поставите действащо лице към вариант на използване в диаграмата, провлачете го с мишката от браузъра в диаграмата на вариантите за използване.
6. С помощта на Unidirectional Assocation (Еднопосочна асоциация) от панела с инструментите нарисувайте асоциации между действащите лица и вариантите на използване.

***Упражнение 4. Допълване на описания към вариантите на използване.***

1. Маркирайте в браузъра вариант на използване "Избиране на дисциплини".
2. В прозореца на документацията въведете описание към този вариант на използване:

"Този Business Use Case позволява на студента да избере конкретни дисциплини за текущия семестър. Студентът може да промени своя избор, при положение, че промяната е направена своевременно в началото на семестър."

3. Създайте с помощта на MS Word текстов файл с описание на варианта на използване "Избиране на дисциплини".

Спецификация Business Use Case "Избиране на дисциплини":

*Име:*

Избиране на дисциплини.

*Кратко описание:*

Даденият Business Use Case позволява на студента да избира от предложените дисциплини за текущия семестър. Студентът може да промени своя избор, ако изборът е направен своевременно в началото на семестъра.

*Основен сценарий:*

1. Студентът посещава административния служител и моли да направи избор от предложените дисциплини или да промени направения от него вече избор.
2. В зависимост от конкретния случай се изпълнява един от подчинените сценарии – създаване на нов план или изменение на съществуващ план.

*Подчинен сценарий "Създаване на нов план":*

1. Административният служител търси в каталога списъка на предложените в този семестър дисциплини и го предоставя на студента.
2. Студентът избира от този списък 4 основни дисциплини и 2 алтернативни.
3. Административният служител записва плана на студента.
4. Преминава се към подчинения сценарий "Регистриране на плана".

*Подчинен сценарий "Промяна на плана":*

1. Административният служител намира избрания от студента план.
2. Извежда от списъка на дисциплините тези, които все още могат да бъдат избрани и ги предоставя на студента.
3. Студентът може да промени своя план като отменя или добавя от предлаганите дисциплини.
4. След промяната административният служител обновява плана на студента.
5. Изпълнява се подчинения сценарий "Регистриране на плана".

*Подчинен сценарий "Регистриране на плана":*

1. За всяка избрана от студента дисциплина служителят потвърждава дали студентът е изпълнил предварително поставените изисквания (сдаване на определени дисциплини). Също така той регистрира всяка избрана дисциплина и следи за отсъствие на конфликти.
2. Служителят въвежда името на студента за всяка избрана дисциплина. Дисциплината се фиксира в плана.

*Алтернативни сценарии:*

*Неизпълнение на предварителните изисквания, дисциплината вече е попълнена или съществува конфликт в плана:*

Ако по време на изпълнение на подчинен сценарий "Регистриране на плана" служителят установи, че студентът не е изпълнил необходимите предварителни изисквания, или пък избраната от него дисциплина е попълнена (има вече 10 студента, които са я избрали), или са налице конфликти в плана (две или повече дисциплини са със

съвпадащо разписание), то тогава той предлага на студента да промени избраните от него дисциплини, или пък да отложи избрания план, към който може да се върне по-късно.

*Системата от дисциплини е недостъпна:*

Ако по време на търсенето в списъка от дисциплини се окаже, че този списък не е достъпен, то следва да се прекрати регистрацията и да се изчака възстановяването на достъпа.

*Изборът на дисциплини е завършил:*

Ако в самото начало на регистрацията се окаже, че за този семестър тя вече е приключила, то процесът се прекратява.

***Упражнение 5. Прикачване на файл към варианта на използване.***

1. Кликнете с десен бутон на мишката върху варианта на използване.
2. В отвореното меню изберете Open Specification.
3. Минете на команда Files.
4. Кликнете с десен бутон на мишката върху белия екран и от отворилото се меню изберете Insert File.
5. Маркирайте създадения по-рано файл и натиснете Open, за да прикачите файла към варианта на използване.

**Създаване на модел на бизнес-анализа.**

Изпълнители:

1. Административен служител – създава учебния план и списъка с дисциплини, записва студентите за дадени дисциплини, обновява всички данни за дисциплините, лекторите, успеха и студентите.

Същности:

2. Студент.
3. Лектор.
4. График на студента (списък с дисциплините).
5. Дисциплина (по плана за обучение).
6. Предлагана дисциплина (дисциплина от разписа).

***<u>Упражнение 6. Създаване на класове, участващи в бизнес-процеса "Избиране на дисциплини". Операции описващи реализацията на бизнес-процеса.</u>***

1. Кликнете с десен бутон на мишката върху папка Business Object Model от Logical View на браузъра (фиг. 1.1).
2. В отвореното меню изберете New -> Class. Новият клас ще се появи в браузъра под име NewClass.
3. Маркирайте го и въведете името "Регистратор" (административен служител).
4. Кликнете с десен бутон на мишката върху този клас.
5. В отвореното меню изберете Open Specification.
6. В полето стереотип изберете Business Worker и натиснете OK.
7. Създайте по аналогичен начин класове за същност от стереотипа Business Entity.
8. Кликнете с десен бутон на мишката върху папка Object Model от Logical View в браузъра.
9. В отвореното меню изберете New -> Package.
10. Именувайте новата папка като Business Use-Case Realizations.
11. В папка Business Use-Case Realizations създайте "Избиране на дисциплини" (това е вариант на използване със стереотип "business use-case realizations", който се задава в спецификацията на варианта на използване).
12. Кликнете с десен бутон на мишката в/у "Избиране на дисциплини".
13. В отвореното меню изберете New -> Class Diagram.
14. Именувайте новата диаграма на класовете VOPC.
15. Отворете тази диаграма и пренесете класовете на отворената диаграма в съответствие с фиг. 2.3.

Диаграмата на класовете от модела на бизнес-анализа, описваща Business Use Case "Избиране на дисциплини" е показана на фиг. 2.3 (за дадените класове е използвано изображение на стереотипа във вид на отметка – label.


фиг.2.3. Диаграмата на класовете от модела на бизнес-анализа

Настройката на изображението на стереотипа може да бъде направено по следния начин:

1. За целия модел – в меню Tools -> Options -> Diagram -> Stereotype Diaplay.
2. За отделен елемент от модела – в неговото контекстно меню Options -> Stereotype Display.
3. За няколко групирани елемента от модела – в меню Format -> Stereotype Display.

**Спецификация на изискванията към програмното осигуряване.**

<u>Уточнено условие на задачата за системата</u>

Пред ръководителя на информационния отдел на университета се поставя задача да бъде разработена нова система клиент-сървър за регистриране на студентите. Тази нова система трябва да позволява студентите да избират дисциплини, да могат да проверяват своята успеха чрез PC, включени в локалната мрежа на университета.

Лекторите трябва да притежават on-line достъп, за да посочват дисциплините, които ще четат и да внасят оценки на студентите. За създаваната система е необходима база от данни, която може да е била вече създадена или да се създаде наново. Базата от данни се поддържа от релационна СУБД.

В началото на всеки семестър могат да поискат достъп до списъка от дисциплини, които ще се четат през този семестър. Информацията за всяка дисциплина обхваща: лектор, катедра, изисквания за предварителното ниво на подготовка (преминати успешно дисциплини), хорариум на дисциплината, начин на оценяване.

Новата система трябва да позволява студентите да избират по 4 дисциплини на семестър. Допълнително всеки студент може да посочи и 2 алтернативни дисциплини, ако дадена дисциплина вече е попълнена или отменена. Всяка дисциплина се попълва от не повече от 10 студента и не по-малко от 3-ма (ако студентите са по-малко от 3-ма, то дисциплината се отменя). За всеки семестър има период от време, когато студентите могат да променят своя план. През това време те трябва да имат достъп до системата, за да добавят или отхвърлят дисциплини. След като е извършено регистриране на дисциплините информацията се представя на финансов отдел, където студентите следва да внесат необходимата такса. Ако дадена дисциплина се попълни в процеса на регистриране, то студентът трябва да бъде уведомен до окончателното оформяне на неговия индивидуален план.

В края на семестъра студентите трябва да имат достъп до системата, за да проверят своите оценки. Доколкото тази информация е конфендициална, то системата трябва да осигури защитата от несанкциониран достъп.

Лекторите имат on-line достъп до системата, за да внасят дисциплините, които ще четат, да преглеждат списъка на студентите, избрали дадена дисциплина, както и да напишат оценките на студентите по нея.

**2.5 Съставяне на речник на проекта.**

В речника се описва терминологията на предметната област. Той може да бъде използван като неформален речник.

| **Дисциплина** | **Учебна дисциплина, която се предлага от университета** |
|---|---|
| Предлагана дисциплина (Course Offering) | Планиране на дисциплината през дадения семестър (една дисциплина може да се изучава в няколко семестъра), при което се посочват се точно дните, часа и седмиците. |
| Каталог на дисциплините | Учебен план, който е пълен каталог на всички дисциплини по дадена специалност. |
| Система за разплащане | Система за обработка на информацията за разплащане. |
| Оценка | Оценката, получена от студента за дадена дисциплина. |
| Лектор | Преподавател от университета. |
| Таблица на успеваемост (Report Card) | Оценките получени от студента по всички дисциплини през даден семестър. |
| Списък (Roster) | Списък на всички студенти, записали дадена дисциплина. |
| Студент | Името на студента, обучаващ се в университета. |
| Учебен график | Дисциплините, избрани от студента за текущия семестър. |

**Описание на допълнителните спецификации.**

Въвеждането на допълнителни спецификации цели да определи изискванията към системата за регистриране на дисциплини, които моделът на вариантите на използване не обхваща. Заедно те образуват пълния набор от изисквания към системата.

Допълнителните спецификации определят нефункционални изисквания към системата, такива като: надеждност, удобство на използване, производителност, съпровождаемост, а също така редица функционални изисквания, явяващи се общи за няколко варианта на използване.

**Функционални възможности**

Системата трябва да осигурява многопотребителски режим на работа.

Ако дадена дисциплина се окаже вече запълнена през времето, когато студентът съставя своя учебен график, включващ тази дисциплина, то системата трябва да го уведоми за това.

**Удобство на използване**

Потребителският интерфейс трябва да бъде съвместим с WINDOWS XP.

**Надеждност**

Системата трябва да работи 24ч. на ден, 7 дена в седмицата. Време за евентуално прекъсване – не повече от 10%.

**Производителност**

Системата трябва да поддържа БД за ФМИ. Едновременно тя трябва да осигури достъп на не-повече от 100 потребителя в локалната мрежа.

**Безопасност**

Системата не трябва да разрешава на студентите да променят кой да е учебен график, а само своя личен. Тя трябва да разрешава на лекторите (лично) да внасят изменения на дисциплините. Само лекторът има право и възможност да нанася оценки. Само административният служител може да коригира информацията за студентите.

**Проектни ограничения**

Системата трябва да бъде интегрирана със съществуващия списък на дисциплините на основата на релационна СУБД.

**Създаване на начална версия на модела на вариантите на използване.**

**Действащи лица:**

* ***Регистратор*** (админ. служител) – внася учебния план, каталог на дисциплините, записва студентите за отделните дисциплини, актуализира всички данни за дисциплините, лекторите, успеха и студентите.
* ***Финансова система*** – получава информация от дадената система и отчита плащанията.
* ***Каталог на дисциплините*** – База данни, съдържа информация за дисциплините.

***Упражнение 7. Създаване на действащите лица в средата Rational Rose.***

За да разположите действащите лица в браузъра:

1. Кликнете с десен бутон на мишката върху папка Use Case Model от Use Case View на браузъра (фиг. 1.1).
2. В отвореното меню изберете New -> Actor.
3. В браузъра ще се появи ново действащо лице под названието NewClass. Отляво на името му ще видите пиктограмата на действащите лица в UML.
4. Маркирайте новото действащо лице и въведете неговото име.

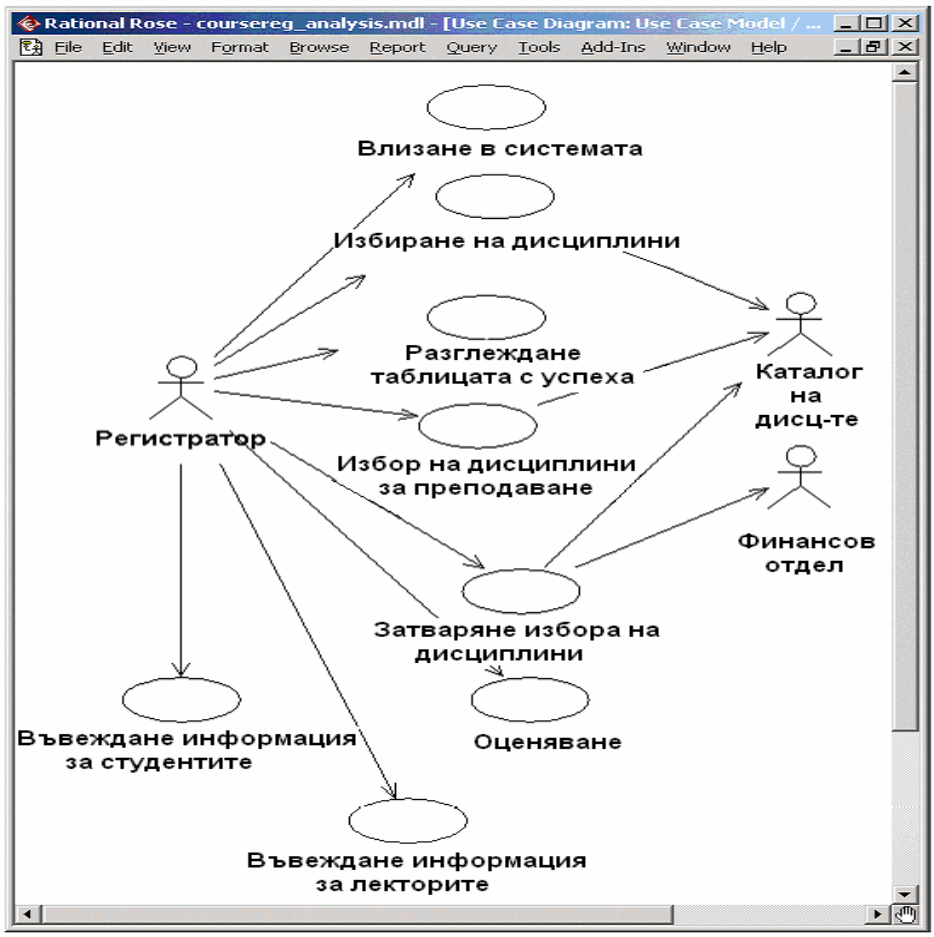
**Варианти на използване:**

Изхождайки от потребностите на действащите лица, се оформят следните варианти на използване:

- Влизане в системата;

- Избиране на дисциплини от студента;
- Попълване на таблица с успеха;
- Определяне на дисциплините за преподаване;
- Оценяване;
- Въвеждане на информация за лекторите;
- Въвеждане на информация за студентите;
- Прекратяване избора на дисциплини.

Началната версия на диаграмата на вариантите на използване е показана на фиг. 2.4.




фиг. 2.4. Началната версия на диаграмата на вариантите на използване

***Упражнение 8. Създаване на варианти на използване в Rational Rose.***

За да разположите един вариант на използване в браузъра:

1. Кликнете с десен бутон на мишката върху папка Use Case Model от Use Case View на браузъра (фиг. 1.1.).
2. В отвореното меню изберете New ->Use Case.
3. Новият вариант на използване ще се появи в браузър под име NewUseCase. Отляво на него може да се види пиктограмата на варианта на използване в UML.
4. Маркирайте този нов вариант на използване и въведете името му.

**Диаграма на вариантите на използване:**

Създайте диаграма на вариантите на използване за системата на избиране на дисциплини. Стъпките за нейното създаване са изброени по-долу. Готовата диаграма е показана на фиг. 2.4.

В средата Rose диаграмите на вариантите на използване се създават чрез представяне на вариантите на използване. Главната диаграма (Main) се предлага по премълчаване. За моделиране на системата след това може да са разработени толкова диаграми, колкото са необходими.

За да получите достъп до главната диаграма на вариантите на използване:

1. Кликнете върху символа “+” от вариантите на използване в браузъра. Това ще предизвика отваряне на даденото представяне.
2. Кликнете два пъти върху главната диаграма Main, за да я отворите. Заглавният ред ще се промени, включвайки фразата [Use Case Diagram: Use Case View / Main].

За да създадете нова диаграма на вариантите на използване:

1. Кликнете с десен бутон на мишката върху папката за представяне на вариантите на използване в браузъра.
2. От появилото се меню изберете New -> Use Case Diagram.
3. Маркирайте новата диаграма и я именувайте.
4. Кликнете два пъти върху името на тази диаграма в браузъра, за да я отворите.

***Упражнение 9. Построяване на диаграмата на вариантите на използване***

1. Отворете диаграмата на вариантите на използване Main.

2. За да разположите действащо лице или вариант на използването върху диаграмата провлачете ги с мишката от браузъра върху диаграмата на вариантите на използване.
3. С помощта на Unidirectional Assocation (еднопосочна асоциация) от панела с инструментите нарисувайте асоциацията между действащите лица и вариантите на използване.

**Модификация на модела на вариантите на използване**

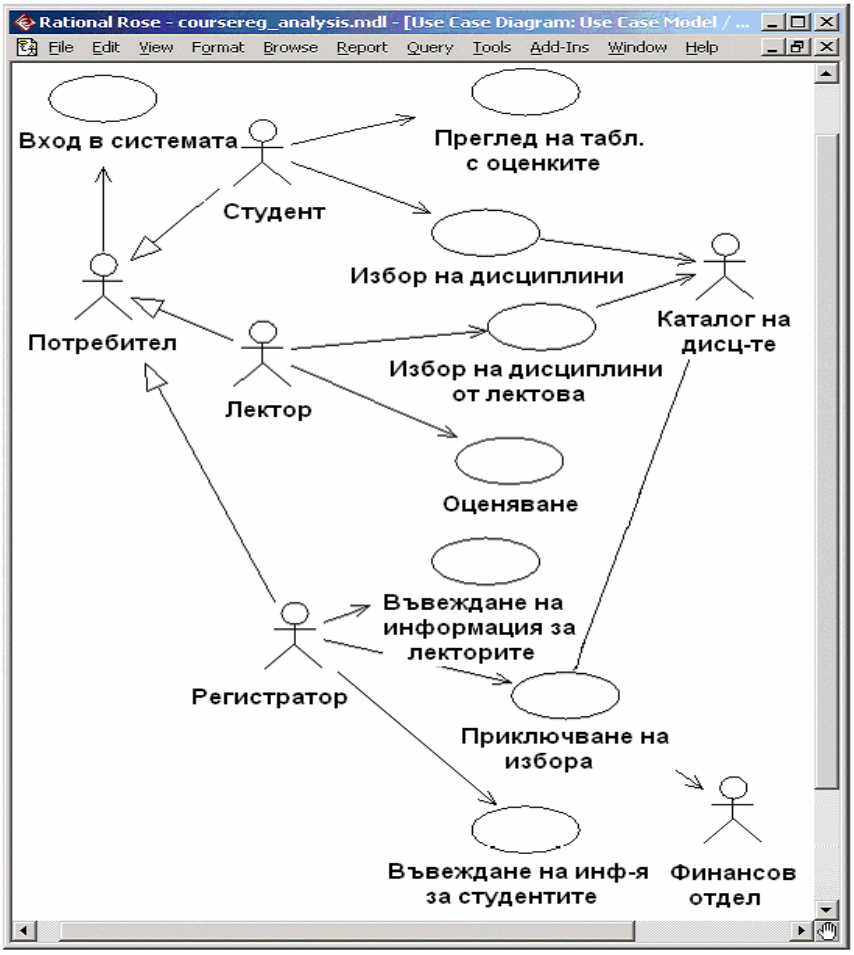
Съгласно условието на задачата, като потребители на системата се въвеждат студентите и лекторите. При това в описанието на действащите лица и вариантите на използване се внасят промени. Модифицираната версия на диаграмата на вариантите на използване е показана на фиг. 2.5. Доколкото влизането в системата е еднакво за регистратора, студента и лектора, тяхното поведение може да се обобщи и да се въведе ново действащо лице "Потребител" (супертип) с общ вариант на използване "Вход в системата". Подтипове на него са Регистратор, Студент и Лектор.

<u>Действащи лица:</u>

- Студент – избира дисциплини и преглежда таблицата с оценките.
- Лектор – предлага дисциплини за преподаване и оценява студентите.
- Регистратор – въвежда учебния план и каталог на дисциплините, обновява всички данни за дисциплините, лекторите и студентите.
- Финансов отдел – получава информация за разплащане на дисциплините.
- Каталог на дисциплините – БД, съдържа информация за дисциплините.

<u>Варианти на използване:</u>

- Вход в системата;
- Избиране на дисциплини;
- Преглеждане на таблицата с оценките;
- Предлагане на дисциплини за преподаване;
- Оценяване;
- Въвеждане на информация за лекторите;
- Въвеждане на информация за студентите;
- Прекратяване на избора на дисциплини.




фиг. 2.5. Модифицираната версия на диаграмата на вариантите на използване

***<u>Упражнение 10. Допълване на описанията на вариантите на използване.</u>***

1. Маркирайте в браузъра варианта на използване “Избиране на дисциплини”.
2. В процеса на документацията въведете следното описание към този вариант на използване: “ Този вариант на използване дава на студента възможност да избере дисциплини за текущия семестър”.
3. Създайте с помощта на MS WORD три текстови файла с описание на вариантите на използване “Вход в системата” ,

Избиране на дисциплини” и “Прекратяване избора на дисциплини”.

Спецификации на вариантите на използване

Вариант на използване “Вход в системата”:

*Кратко описание:*

Този вариант описва как се влиза в системата от потребителя за регистриране на дисциплини.

*Основен поток на събитията:*

Началото на този вариант е всеки опит на потребителя да влезе в системата и да регистрира дисциплини.

1. Системата прави запитване за име и парола на потребителя.
2. Потребителят въвежда име и парола.
3. Системата потвърждава името и паролата и дава достъп до системата.

*Алтернативни потоци:*

Неправилни име или парола:

Ако при проверка на въведените от потребителя име и парола системата установи, че името и/или паролата са неправилно въведени, то се извежда съобщение за грешка. Потребителят може да направи нов опит или да откаже от влизане в системата, при което изпълнението на този вариант на използване се прекратява.

*Предусловия:*

Няма.

*Постусловия:*

Ако този вариант на използване е изпълнен успешно, то потребителят влиза в системата. В противен случай състоянието на системата не се променя.

**Вариант на използване “Избиране на дисциплини”:**

*Кратко описание:*

Този вариант на използване позволява студентът да избере от предлаганите за този семестър дисциплини. Той може да промени своя избор (да го обнови или да се откаже от някои дисциплини), при положение, че това стане в определеното за това време в началото на семестъра. Системата предоставя каталог (списък) на дисциплините, които се предлагат за дадения семестър.

*Основен поток на събитията:*

Този вариант на използване започва да се изпълнява, когато студентът иска да избере конкретни дисциплини или пък да промени вече направен избор.

1. Системата прави запитване какво ще се извърши – избор на дисциплини, обновяване или отказване.
2. Когато студентът посочи определено действие, то се изпълнява един от подчинените потоци – избор, обновяване, отказване или възприемане.

*Създаване на график:*

1. Системата прави търсене в каталога на дисциплините и извежда списък на достъпните към момента.
2. Студентът избира от този списък 4 основни и 2 алтернативни дисциплини.
3. След направения избор системата създава график на студента с избраните от него дисциплини.
4. Изпълнява се подчинения поток "Възприемане на графика".

*Обновяване на графика:*

1. Системата извежда вече избрания и възприет график на студента.
2. Системата прави търсене на достъпните дисциплини и извежда текущия списък.
3. Студентът може да обнови своя график като премахне и добави от предложените дисциплини.
4. След направения избор системата обновява графика.
5. Изпълнява се подчинения поток "Възприемане на графика".

*Отстраняване (отказване от) на график:*

1. Системата извежда вече избрания и възприет график на студента.
2. Системата иска потвърждение от студента, че този график се премахва.
3. Студентът потвърждава премахването.
4. Системата премахва графика. Ако този график включва предложени дисциплини, то студентът се изтрива от списъка на тези дисциплини.

*Възприемане на графика:*

За всяка избрана, но все още на "зафиксирана" (утвърдена) дисциплина в графика, системата проверява дали студентът е изпълнил предварителните изисквания (предварително сдадени дисциплини) и отсъствието на конфликти в графика. След това системата добавя студента към списъка с избраните дисциплини. Дисциплините се фиксират в графика и той се съхранява в системата.

*Алтернативни потоци:*

*Съхраняване на графика:*

Всеки момент студентът може да посочи вместо възприемане на графика, неговото съхраняване. В този случай стъпката “Възприемане на графика” се заменя с:

1. “Незафиксираните” конкретни дисциплини се отбелязват в графика като ‘избрани”.
2. Графикът се запазва в системата.

*Не са изпълнени предварителните изисквания, дисциплината вече е запълнена или има конфликти в графика:*

Ако при изпълнение на подчинения поток “Възприемане на графика” системата установи, че студентът не е изпълнил предварително поставените условия, или избраната от него дисциплина е вече попълнена (има над 10 студента, които са я избрали), или са налице конфликти в графика, то се издава съобщение за грешка. Студентът може или да избере друга предлагана дисциплина и да продължи изпълнението на варианта на използване, или да съхрани графика, или да отмени тази операция, след което основният поток започва отначало.

*Графикът не е намерен:*

Ако при изпълнение на подчинените потоци “Обновяване на графика” или “Отстраняване на графика” системата не може да открие графика на студента, то се извежда съобщение за грешка. Ако след това студентът потвърди това съобщение, основният поток започва отначало.

*Каталогът на дисциплините е недостъпен:*

Ако се окаже, че не е възможно установяването на връзка с каталога на дисциплините, то се извежда съобщение за грешка. След като студентът потвърди това съобщение, вариантът на използване завършва.

*Изборът на дисциплини е завършен:*

Ако в самото начало на изпълнение на варианта на използване се окаже, че изборът на дисциплини за този семестър вече е приключил, то се извежда съобщение за това и вариантът на използване завършва.

*Отстраняването се отменя:*

Ако по време на изпълнението на подчинения поток “Отстраняване (отказване от) на графика” студентът реши да не го отстранява, то самото отстраняване се отменя и основният поток започва отначало.

*Предусловия:*

Преди началото на изпълнение на този вариант на използване студентът трябва да влезе в системата.

*Постусловия:*

Ако вариантът на използване е завършен успешно, то графикът на студента ще бъде създаден, обновен или отстранен (премахнат). В противен случай състоянието на системата не се променя.

**Вариант на използване “Прекратяване на избора”:**

*Кратко описание:*

Този вариант на използване позволява на регистратора да прекратява процеса на избиране на дисциплини. Предложените дисциплини, за които не са записани достатъчно студенти (по-малко от трима), се отменят. Във финансов отдел се подава информация за всеки студент по избраните дисциплини, за да може той да направи плащането.

*Основен поток на събитията:*

Този вариант на използване започва да се изпълнява, когато регистраторът извърши прекратяване на избора на дисциплини.

1. Системата проверява състоянието на системата за избор на дисциплини. Ако избирането все още се изпълнява, то се издава съобщение и вариантът на използване се завършва.
2. За всяка предлагана дисциплина системата проверява дали е заявена от някой лектор и дали е избрана от не по-малко от трима студенти. Ако тези условия се изпълняват, то системата фиксира предлаганата дисциплина във всеки график, който включва тези дисциплини.
3. За всеки студентски график се проверява наличието на максимален брой основни дисциплини; ако този брой не е достатъчен, системата се опитва да го допълни от алтернативните дисциплини, посочени от студента. Ако няма такива дисциплини, допълнение на графика не се прави.
4. Системата затваря всички предложени дисциплини. Ако дадена дисциплина е избрана от по-малко от трима студента (като се отчете и допълването по т.3), то системата изключва тази дисциплина от всеки график, където тя е заложена.
5. Системата изчислява плащането за обучението на всеки студент и насочва информацията към финансовия отдел. Той от своя страна издава документ за необходимото плащане и копие от окончателния график.

*Алтернативни потоци:*

*Дисциплината не се чете от никого:*

Ако при изпълнението на основния поток се констатира, че дадена дисциплина не е посочена от никой лектор, то тя се отменя. Системата я изхвърля от всички графици, в които е включена.

*Системата за разплащане е недостъпна:*

Ако връзката с финансовия отдел не е възможна, то след определено време системата прави отново опит за установяване на такава. Опитите за свързване продължава, докато се осъществи.

*Предусловия:*

Преди началото на изпълнение на този вариант на използване регистраторът трябва да влезе в системата.

*Постусловия:*

Ако вариантът на използване завърши успешно, то изборът се затваря. В противен случай състоянието на системата не се променя.

***<u>Упражнение 11. Прикачване на файл към варианта на използване.</u>***

1. Кликнете с десен бутон на мишката върху варианта на използване.
2. В отвореното меню изберете Open Specification.
3. Преминете към добавяне на файл.
4. Кликнете с десен бутон на мишката на бялото поле и в отвореното меню изберете Insert Fail.
5. Маркирайте предварително създадения файл и натиснете Open, за да прикачите файла към варианта на използване.

## 2.9 Анализ на системата.

### 2.9.1 Архитектурен анализ.

Архитектурният анализ се прави от архитекта (разработчика) на системата и включва:

- Утвърждаване на общите стандарти (съглашения) на моделиране и документиране на системата;
- Предварително определяне на механизмите на анализа;
- Формиране на съвкупност от основните абстракции на предметната област;
- Формиране на предварително (начално) представяне на архитектурните нива.

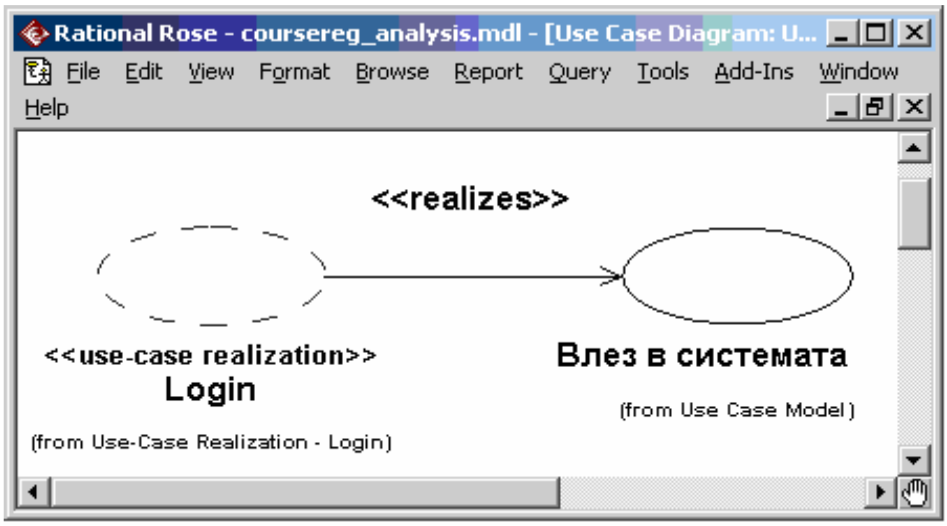
Съглашенията на моделирането определят:

- използваните диаграми и елементи на модела;
- правила за тяхното използване;
- съглашение за наименуване на елементите на модела;
- организация на модела.

*Пример за съвкупност на съглашения на моделирането:*

- Имената на вариантите на използване трябва да бъдат кратки глаголни фрази;
- Имената на класовете трябва да бъдат съществителни, съответстващи по възможност на понятията от предметната област;
- Имената на класовете започват с главна буква;

- Имената на атрибутите и операциите трябва да започват с малка буква;
- Съставните имена са слети, без подчертаване, всяка отделна дума започва с главна буква;
- Всички класове и диаграми, описващи предварителния системен проект, се разлагат в папка Analysis Model;
- Диаграмите на класовете, реализиращи вариант на използване, и диаграмите на взаимодействие, отразяващи взаимодействието на обектите в процеса на реализиране на сценариите на варианта на използване, се разполагат съвместно с името на дадения вариант на използване и със стереотип “use-case relization”. Всички тези обединения се разполагат в папка Use Case Relizations. Връзката между варианта на използване и неговата реализация на специална диаграма на трасиране (фиг. 2.6).




фиг. 2.6. Диаграма на трасиране

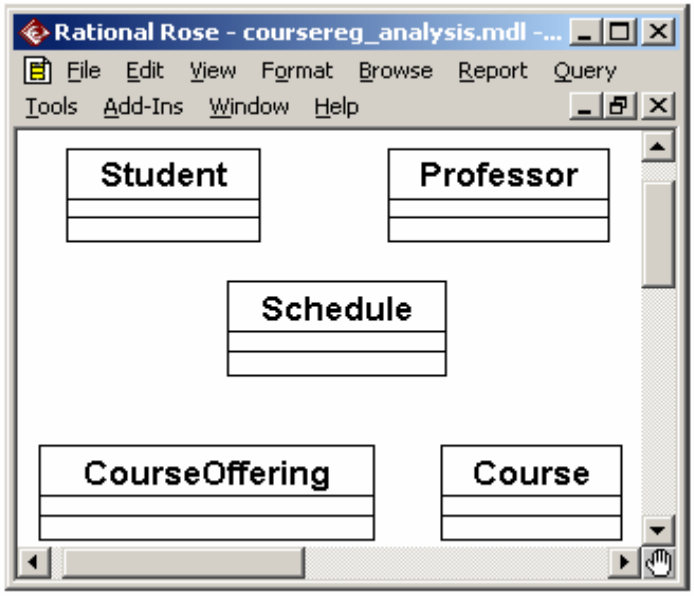
Идентификацията на основните абстракции се състои в предварителното определяне на съвкупността на класовете на системата (класовете на анализа) на основата на описанието на предметната област и на спецификацията на изискванията към системата. Начините за идентификация на основните абстракции са аналогични на начините за идентификация на същностите в модела ERM.

За системата на избор са идентифицирани пет класа на анализа:

- Student (студент);
- Proffesor (лектор);

- Schedule (учебен график);
- Course (дисциплина);
- CourseOffering (предлагана дисциплина).

Класовете на анализа са показани на фиг. 2.7.




фиг. 2.7. Класовете на анализа

***<u>Упражнение 12. Създаване структурата на модела и на класовете на анализа в съответствие с изискванията на архитектурния анализ.</u>***

Създаване на пакетите и на диаграмата Traceabilities:

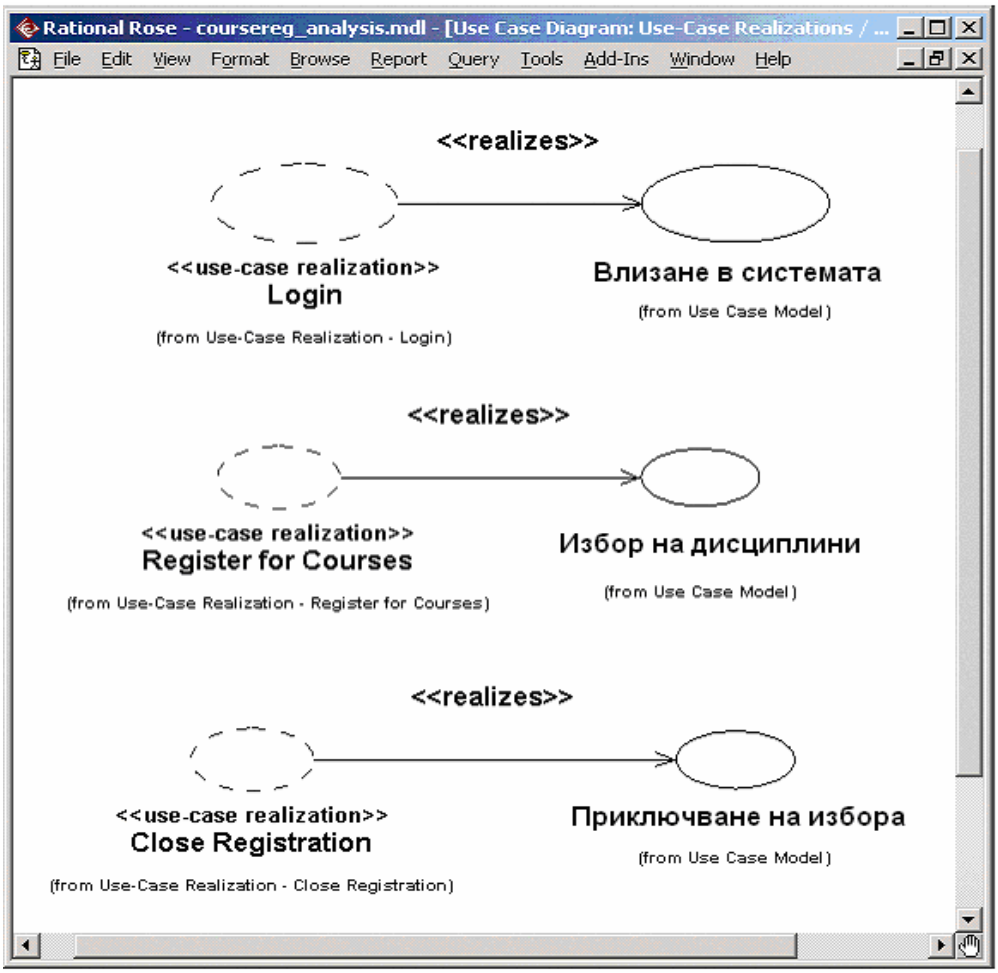
1. Кликнете с десен бутон на мишката върху папка Design Model от браузъра.
2. В отвореното меню изберете New -> Package.
3. Създайте папка Use-Case Relizations, след това в нея папки Use-Case Relization – Close Registration, Use-Case Relization – Login и Use-Case Relization – Register for Courses.
4. Във всяка една от папките от типа Use-Case Relization създайте съответните кооперации (обединения) Close Registration, Login и Register for Courses (всяка кооперация (обединение) представлява вариант на използване със стереотип “use-case relization”, който се задава в спецификацията на варианта на използване).

5. Създайте в папка Use-Case Relizations нова диаграма на вариантите на използване с име Traceabilities и я постройте в съответствие с фиг. 2.8.

Създаване на класовете на анализа и на съответната диаграма Key Abstractions:

1. Кликнете с десен бутон на мишката върху папка Analysis Model (фиг. 1.1.).
2. В отвореното меню изберете New -> Class. Новият клас ще се появи под името NewClass в браузъра.
3. Маркирайте го и въведете името Students.
4. По аналогичен начин създайте класовете Proffesor, Schedule, Course и CourseOffering.
5. Кликнете с десен бутон на мишката върху папка Analysis Model (фиг. 1.1.).
6. В отвореното меню изберете New -> Class Diagram.
7. Въведете име на новата диаграма на класовете Key Abstractions.
8. За да разположите така създадените класове върху диаграмата на класовете, отворете и я провлачете с мишката класовете в отворената диаграма. Диаграмата на класовете трябва да бъде като тази на фиг. 2.7.

Архитектурните нива се представят в модела (папка Design Model) във вид на папки със стереотип “layer”. Броят и структурата на нивата зависят от сложността на предметната област и от средата на реализация. В рамките на архитектурния анализ се определя началната структура на модела (броя на папките и техните зависимости) и се разглеждат само горните нива (Application и Business Services).




фиг. 2.8. Архитектурните нива

### 2.9.2. Анализ на вариантите на използване

<u>Идентификация на класовете, които участват в реализацията на потоците от събития на вариантите на използване</u>

В потока от събития на варианта на използване се формират класовете от три типа:

1.Гранични класове (Boundary) – служат като посредници при взаимодействие на външни обекти със системата. Като правило за всяка двойка “действащо лице – вариант на използване” се определя един граничен клас. Видовете гранични класове са:

- потребителски интерфейс (обмен на информация с потребителя, без детайлите на самия интерфейс – бутони, списъци, прозорци);
- системен интерфейс;

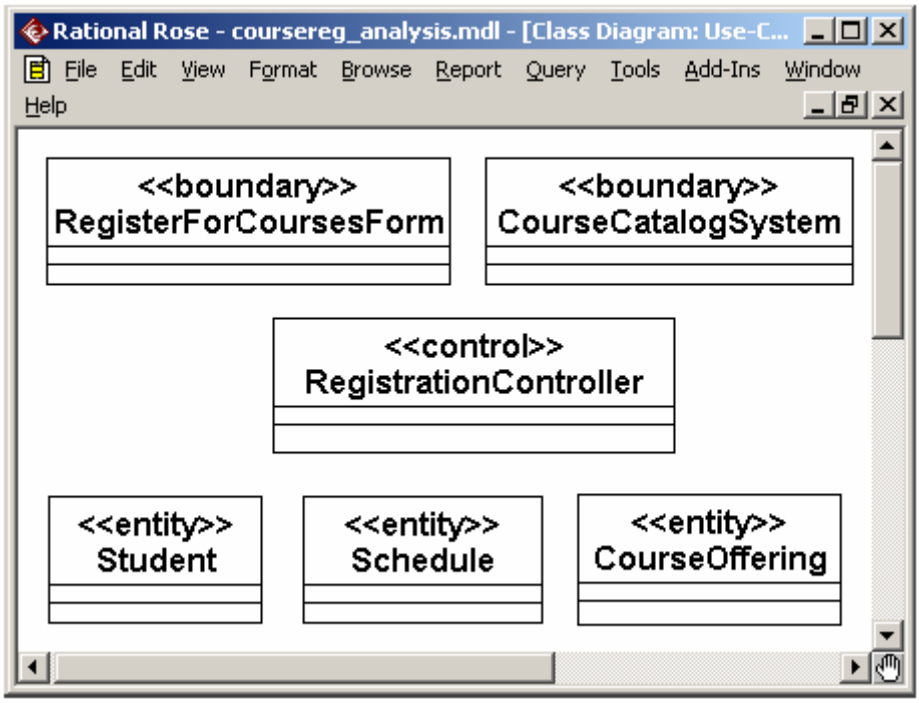
- апаратен интерфейс – използваните протоколи, без тяхната детайлна реализация.

2.Класове на същността (Entity) – това са ключовите абстракции (понятия ) на разработваната система. Източници за определяне на класовете на същностите са:

- ключови абстракции, създадени в процеса на архитектурния анализ;
- речник;
- описанието на потоците от събития на вариантите на използване.

3.Управляващи ключове (Control) – осигуряват координация на поведението на обектите в системата. При някои варианти на използване те могат да отсъстват. В този случай се извършват прости манипулации върху съхраняваните данни. Като правило, за всеки вариант на използване се определя един управляващ клас. Примери за управляващи класове: менажер на транзакциите, координатор на ресурсите, коректор на грешки.

Пример с броя на класовете, участващи в реализацията на варианта на използване “Избор на дисциплини” е показан на фиг. 2.9.




фиг.2.9. Варианта на използване “Избор на дисциплини”

***Упражнение 13. Създаване на класове, участващи в реализацията на варианта на използване Refister of Courses и на диаграмата на класовете "View Of Participating Classes" (VOPC).***

1. Кликнете с десен бутон на мишката върху папка Analysis Model (фиг. 1.1).
2. В отвореното меню изберете New -> Class. В браузъра ще се появи новия клас под име NewClass.
3. Маркирайте го и въведете име RegisterForCoursesForm.
4. Кликнете с десен бутон на мишката върху класа RegisterForCoursesForm.
5. В отвореното меню изберете Open Specification.
6. В полето на стереотипа изберете Boundary и натиснете OK.
7. По аналогичен начин създайте и класовете CourseCatalogSystem със стереотип Boundary и RegistrationController със стереотип Control.
8. За класовете Schedule, CourseOffering и Student определете стереотип Entity.
9. Кликнете с десен бутон на мишката върху Register for Courses от папка Use-Case Relization – Register for Courses.
10. В отвореното меню изберете New -> Class Diagram.
11. Новата диаграма на класовете назовете като VOPC (classes only).
12. Отворете диаграмата и провлачете класовете в нея, както е показано на фиг. 2.9.

***Разпределяне на поведението между класовете за даден вариант на използване.***

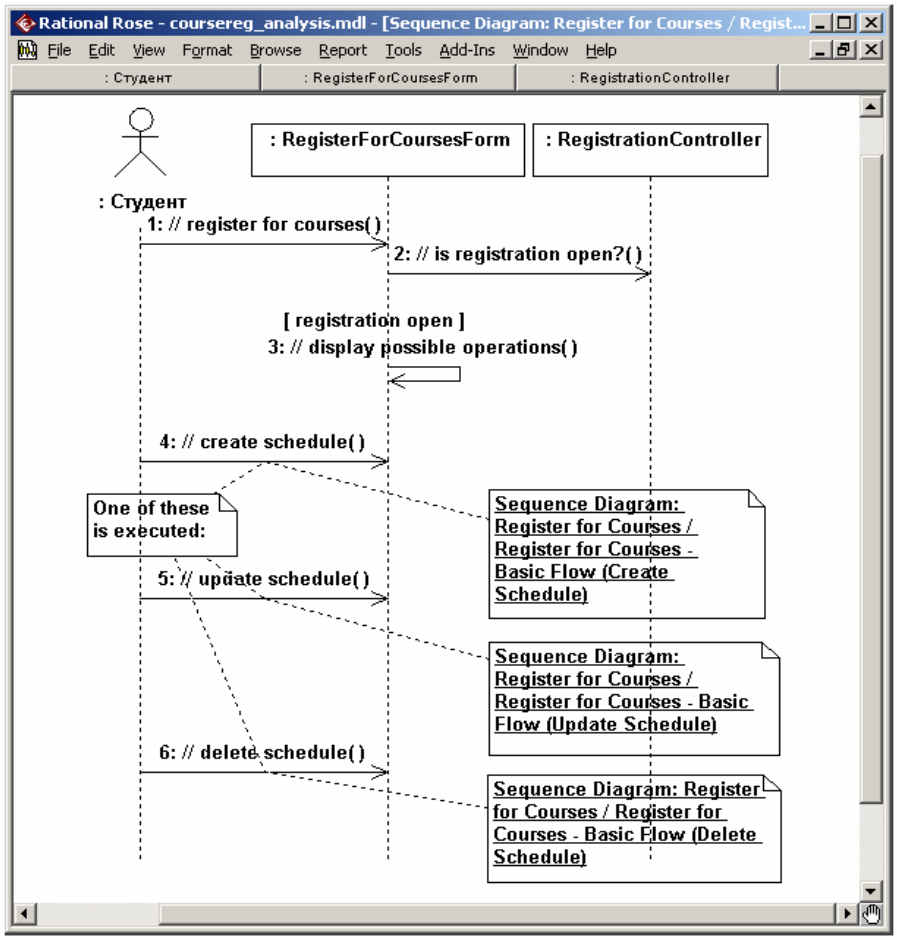
Това разпределение се осъществява с помощта на диаграми на взаимодействието (диаграми на последователността и обединени диаграми). На първо място се построява диаграма (една или повече), описваща основния поток на събитията и неговите подчинени потоци. За всеки алтернативен поток на събитията се построява отделна диаграма. Примери:

- обработка на грешки;
- контрол на времето за изпълнение;
- обработка на неправилно въведени данни.

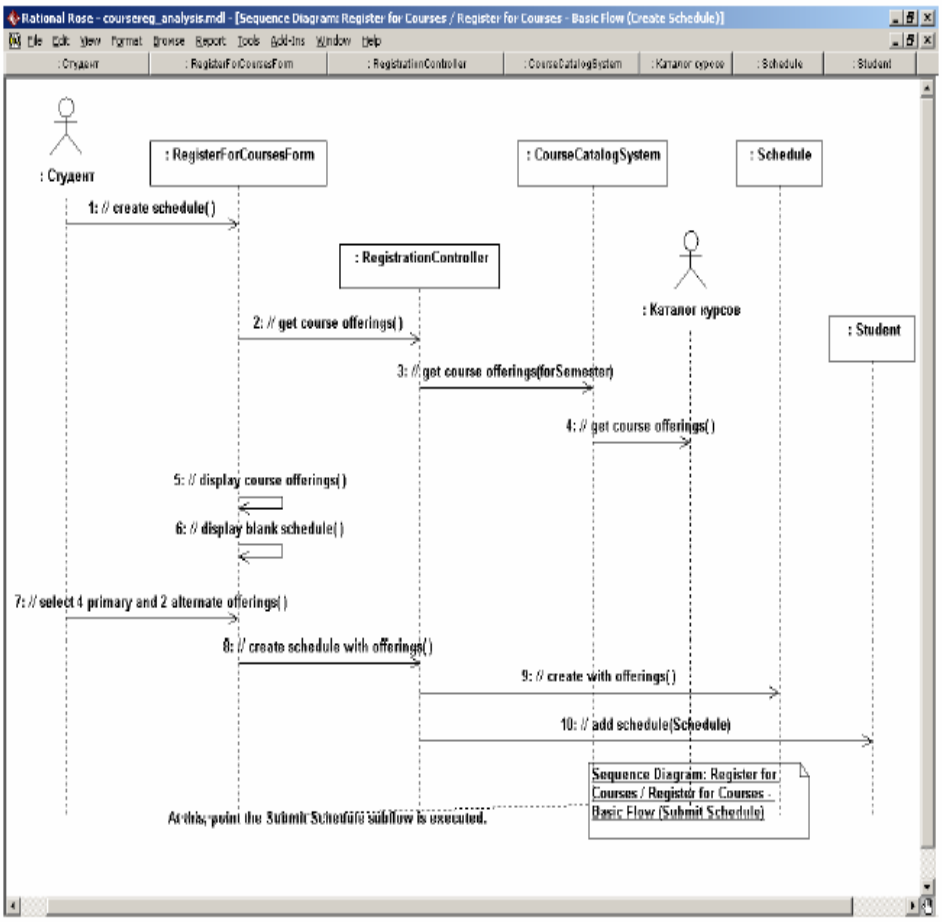
Описването на тривиални потоци на събития е нецелесъобразно ( напр. ако в потока участва само един обект).

***<u>Упражнение 14. Създаване на диаграми на взаимодействието.</u>***

Нека създадем диаграми на последователността и обединим диаграми за основния поток на събитията на варианта на използване Register for Courses. Готовите диаграми на последователността трябва да са от вида, показан на фиг. 2.10 – 1.14.




фиг. 2.10 Диаграма на последователността Register for Courses – Basic Flow




фиг. 2.11 Диаграма на последователността Register for Courses – Basic Flow (Create Schedule)

Настройка

1. От менюто на модела изберете Tools -> Options.
2. Преминете към диаграмата.
3. Контролните превключватели Sequence Numbering, Collaboration Numbering трябва да са маркирани, а Focus of Control – не.
4. Натиснете OK, за да излезете от прозореца на параметрите.

Създаване на диаграмата на последователностите

1. Кликнете с десен бутон на мишката върху Register for Courses от папка Use-Case Relization – Register for Courses.
2. В отвореното меню изберете New -> Sequence Diagram.
3. Поставете име на новата диаграма Register for Courses – Basic Flow.

4. Кликнете два пъти върху нея, за да я отворите.

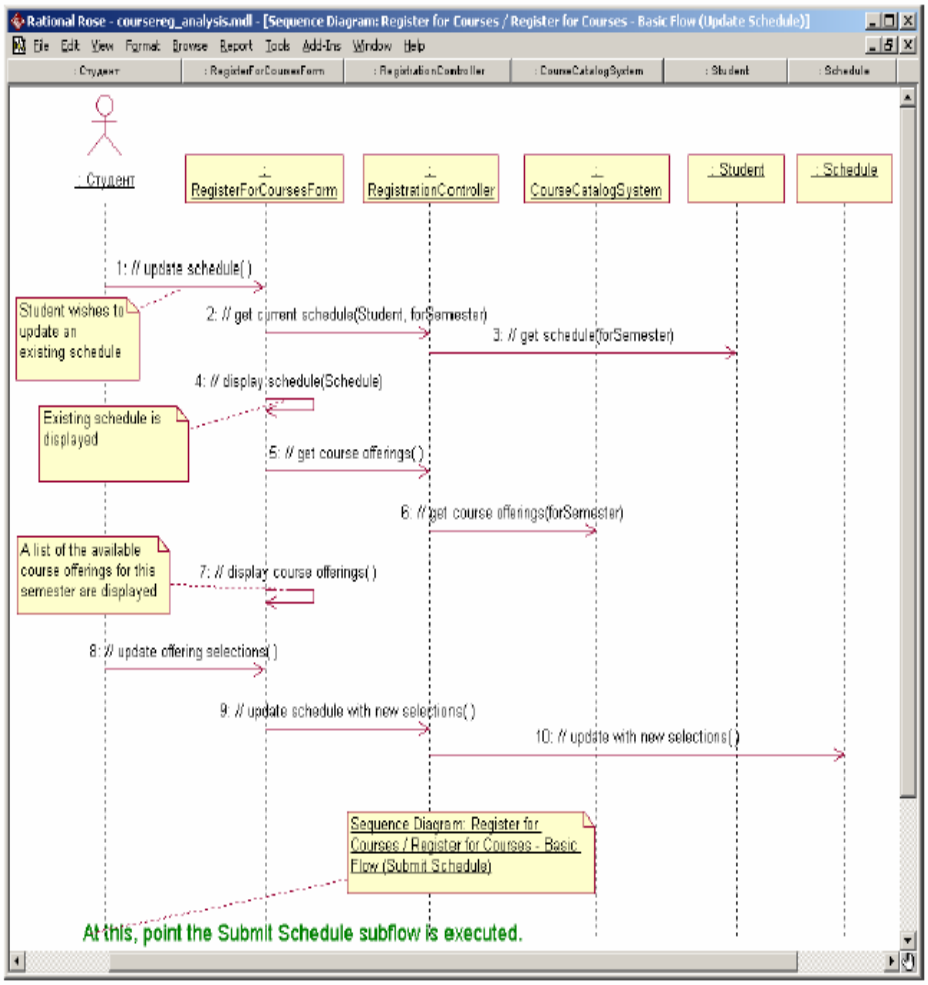
Добавяне в диаграмата на действащото лице, обекти и съобщения.

1. Прехвърлете действащото лице “Студент” от браузъра в/у диаграмата.
2. Прехвърлете класовете RegisterForCourses и RegistrationController от браузъра върху диаграмата.
3. От панела с инструменти натиснете Open Message.
4. С помощта на мишката прекарайте линия със стрелка от вертикалната линия на действащото лице “Студент” до линията на обекта RegisterForCoursesForm.
5. Въведете името на съобщението: //register for courses (фиг. 2.10).
6. Повторете действията от 3÷5, за да разположите върху диаграмата и останалите съобщения (фиг. 2.10) (за рефлексивното съобщение 3 се използва Message to Self).

Съставяне на съобщенията с операциите.

1. Кликнете с десен бутон на мишката върху текста на съобщение 1, //register for courses (фиг. 2.10).
2. В отвореното меню изберете “new operation”. Ще се появи прозорец със спецификации на операцията.
3. В полето за име поставете името на съобщението - //register for courses.
4. Натиснете OK, за да затворите прозореца със спецификациите на операцията и да се върнете в диаграмата.
5. Повторете действията от 1÷4, докато не съпоставите с операциите всички останали съобщения.

Изпълнете аналогични действия за създаване на диаграми на последователности, показани на фиг. 2.11÷2.14. обърнете внимание, че на диаграмата от фиг. 2.14 се появява обект на нов клас PrimaryScheduleOfferingInfo), асоциация на класа описващ връзката между класовете Schedule и OfferingInfo), който трябва предварително да се създаде.

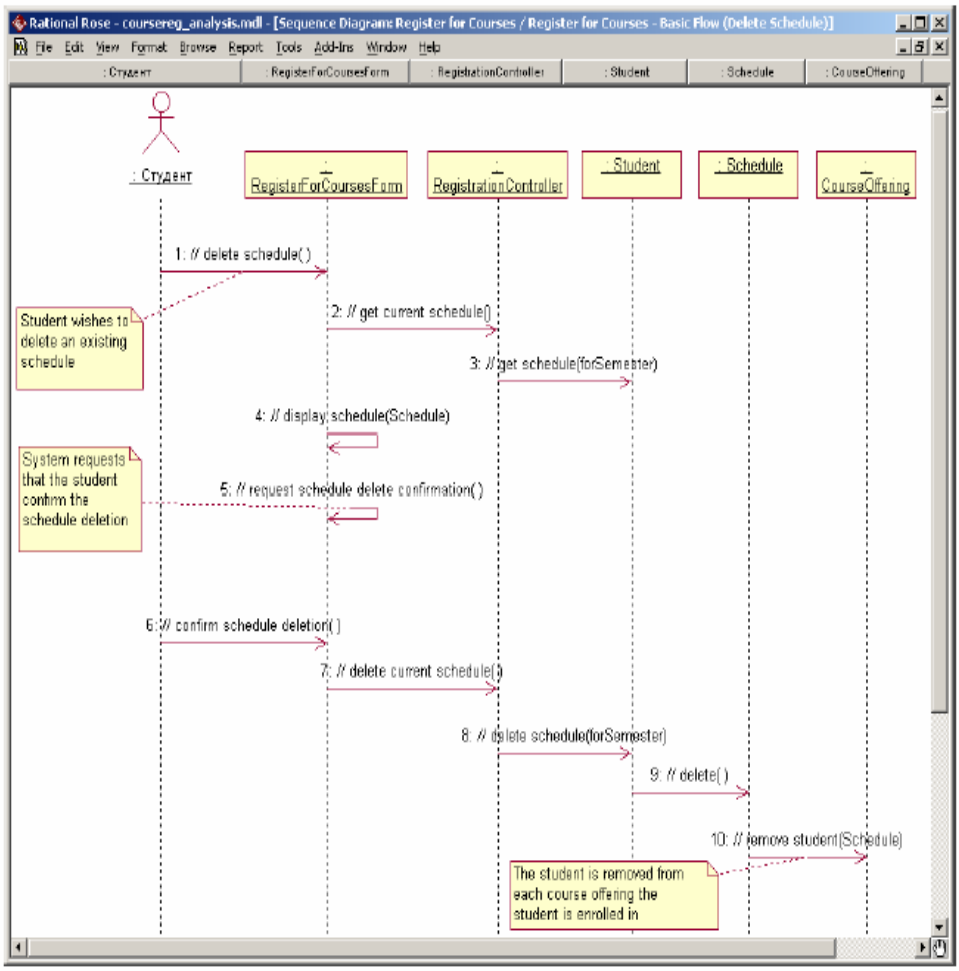



фиг. 2.12. Interaction диаграма

***Създаване на обяснения.***

За да разположите обяснения в/у диаграмата:

1. От панела с инструментите натиснете Note.
2. Кликнете с мишката в диаграмата там, където искате да поставите обяснение.
3. Отделете ново обяснение и въведете вътре текст.
4. За да прикрепите обяснението към даден елемент от диаграмата, от панела с инструменти натиснете Anchor Notes To Item (прикрепи обяснението към елемента).
5. С левия бутон на мишката свържете обяснението е елемента от диаграмата. Между тях ще се формира пунктирна линия.



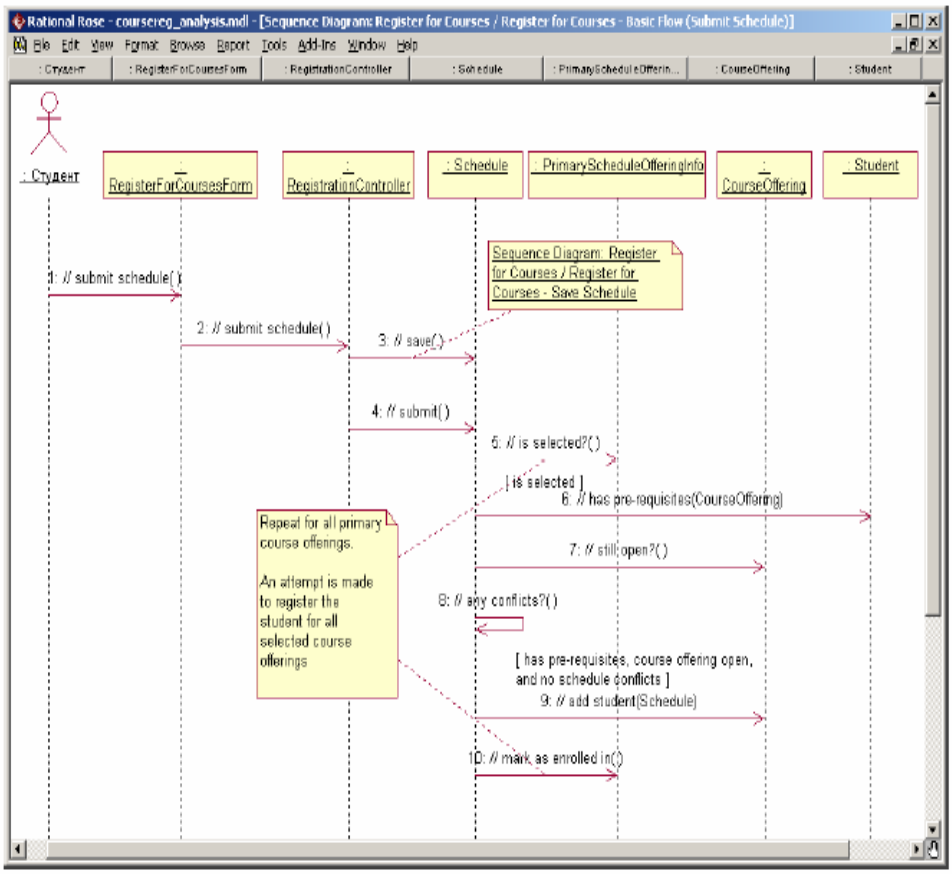

фиг. 2.13. Interaction диаграма - допълнена

6. За да създадете обяснение – препратка (цитат) на друга диаграма (както е направена на диаграмата от фиг.2.7), създайте празно обяснение (без текст) и вмъкнете в него от браузъра нужната диаграма.

Освен обяснения, в диаграмата може да се разположи и текстова област. С нея може да се добавят към диаграмата заглавия.

За да разположите в диаграмата текстова област:

1. От панела за управление натиснете Text Box.
2. Кликнете с мишката вътре в диаграмата, за да разположите там текста.
3. Отделете тази област и въведете текста.



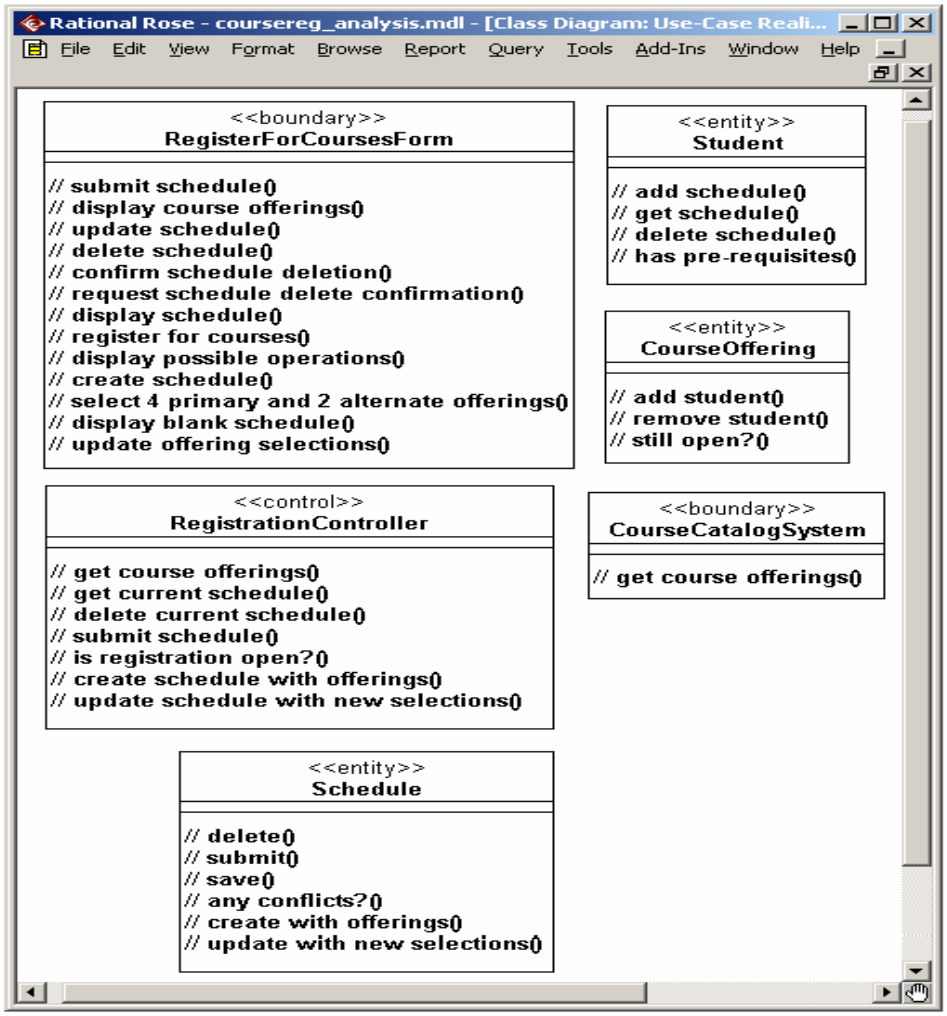

фиг. 2.14. Модифицирана . Interaction диаграма

Създаване на обединена диаграма.

За създаването на обединена диаграма е достатъчно да се отвори диаграмата на последователност и да се натисне F5.

<u>Определяне ангажиментите (responsibilities), атрибутите и асоциациите на класовете.</u>

Ангажиментът е действие, което обектът трябва да изпълни по заявка на други обекти. Ангажиментът се преобразува в една или повече операции на класа на етапа проектиране. Ангажиментите се определят въз основа на съобщенията на диаграмата на взаимодействие и се документират в класовете във вид на операции от “анализа”, които се появяват там автоматично в процеса на построяване на диаграмите на взаимодействие.




фиг.2.15. Клас диаграма

Така например, диаграмата на класовете VOPC (classes only) (фиг. 2.9). След построяване на диаграмите на взаимодействие от упражнение 8 ще приеме вида от фиг. 2.15.

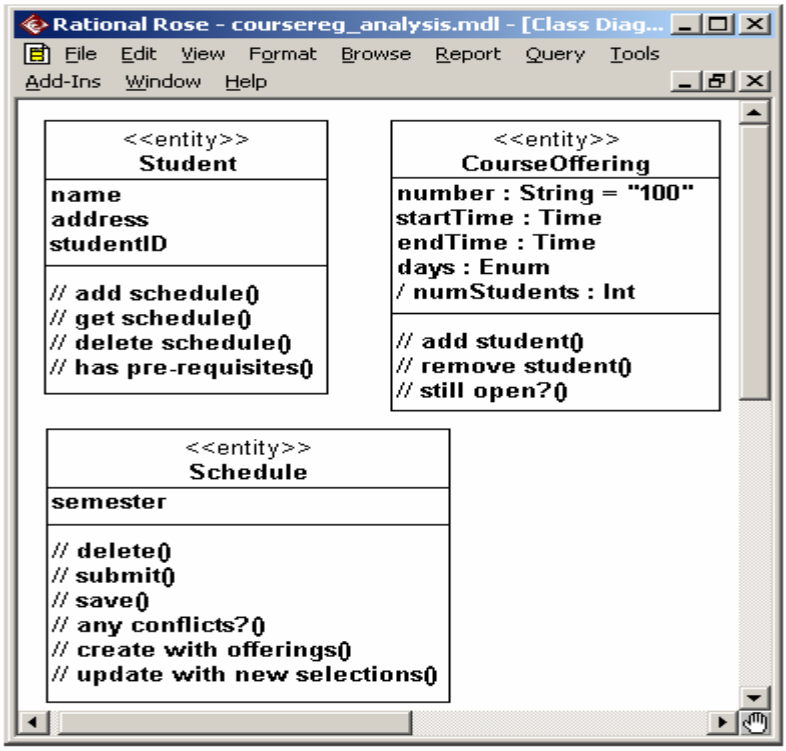
Атрибутите на класовете при анализа се определят като се изхожда от знанията за предметната област, за изискванията към системата и за речника.

***<u>Упражнение 15. Добавяне на атрибути към класовете.</u>***

Настройка.

1. В менюто на модела изберете Tools -> Options.
2. Преминете към Diagram.
3. Убедете се, че превключвателя Show All Attributes е маркиран.

4. Убедете се, че превключвателите Suppress Attributes и Suppress Operations не са маркирани.




фиг. 2.16 Клас диаграма

Добавяне на атрибути.

1. Кликнете с десен бутон на мишката върху клас Student.
2. В отвореното меню изберете New Attribute.
3. Въведете нов атрибут adress.
4. Натиснете клавиша Enter.
5. Повторете стъпките от 1÷4 като добавите атрибутите name и studentID.
6. Добавете атрибутите към класа CourseOffering и Schedule, както е показано на фиг. 2.16.

Връзките (асоциациите) между класовете се определят на два етапа:

1. Началният брой връзки се определя въз основа на анализа на обединените диаграми. Ако два обекта си взаимодействат (обменят съобщения), на обединената диаграма трябва да има връзка между тях (пътя на взаимодействието), която се преобразува в двупосочна асоциация между съответните

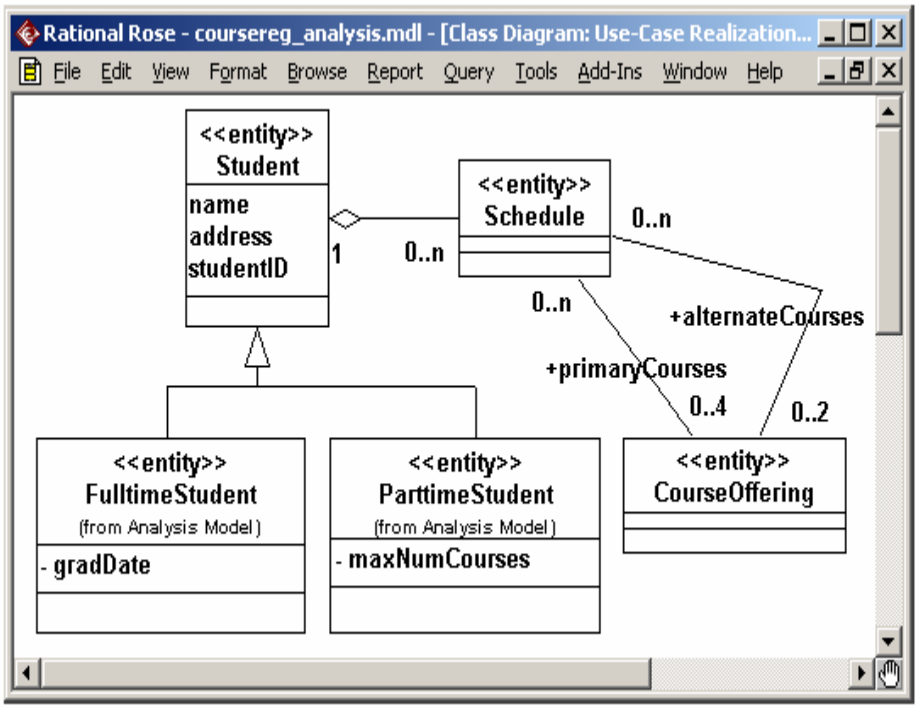
класове. Ако съобщенията между два обекта се предават само в едно направление, то тогава съответната асоциация е еднопосочна.

2. Анализират се и се уточняват асоциациите между класовете. Задават се мощности на асоциациите, а също така могат да се използват множествени асоциации, агрегации, обобщения и асоциации–класове.

***<u>Упражнение 16. Добавяне на връзки.</u>***

Ще добавим връзки към класовете, взимащи участие във варианта на използване Register for Courses. За изобразяване на връзките между класовете ще построим три нови диаграми на класовете в обединението Register for Courses от папка Use-Case Realization – Register for Courses (фиг. 2.17÷2.19).

На фиг. 2.17 са показани само класовете-същности. Агрегацията между класовете Student и Schedule отразява факта, че всеки график е собственост на конкретен студент, принадлежи само на него.



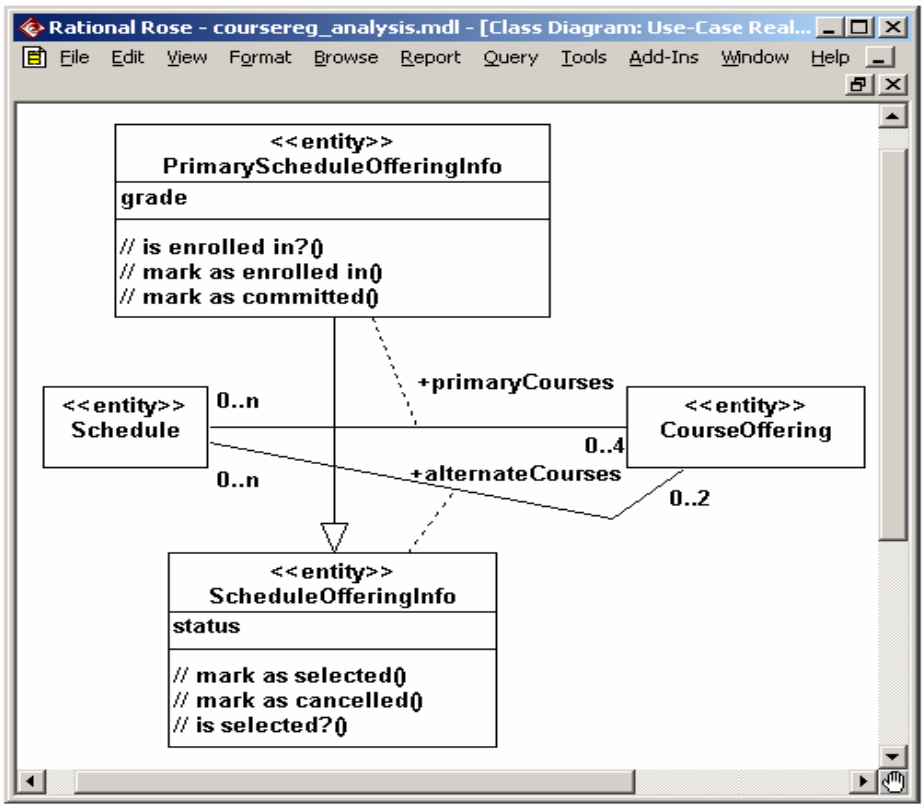

фиг. 2.17. Клас диаграма с връзките

Предполага се също, че в системата ще се пази не само графика за текущия семестър, но и графиците на студента за останалите семестри. Между класовете Schedule и CourseOffering са въведени две асоциации,

доколкото графика на студента може да включва основен (не повече от 4 дисциплини) и алтернативен (не повече от 2 допълнителни дисциплини). Към клас Student са добавени два подкласа – FulltimeStudent (студент редовно обучение) и ParttimeStudent (студент задочно обучение).

На фиг. 2.18 са показани асоциациите – класове, представляващи свойствата на връзките между класовете Schedule и CourseOffering. Асоциацията, свързваща графика с алтернативна дисциплина има само един атрибут – това е статуса на дисциплината в конкретния график (status), който може да приема значения <включен в графика>, <отменен>, <въведен в списъка на дисциплините> и <зафиксиран в графика>. Ако дисциплината в процеса на прекратяване на избора премине от алтернативна в основна, то към съответната асоциация се добавя атрибут <оценка> (grade). По такъв начин асоциацията – клас PrimaryScheduleOfferingInfo наследява свойствата на асоциацията – клас ScheduleOfferingInfo (атрибутите и операциите, съдържащи се в този клас, се отнасят както към основната, така и към алтернативната дисциплина) и добавя свои собствени (оценка и окончателно включване на дисциплина в графика може да има само за основните дисциплини). Това е показано с помощта на връзките за обобщаване.

На фиг. 2.19 е показана главната диаграма на класовете от варианта на използване “Избор на дисциплини” (без атрибути и операции). Асоциациите между граничните и управляващи класове, а също така между управляващите класове и класовете – същности, са въведени въз основа на анализа на обединените диаграми и в отличие от устойчивите структурни (семантически) връзки между същностите, отразяват връзки, които възникват динамично между съответните обекти в потока на управление (в процеса на работа на приложението). Доколкото за асоциациите това не е свойствено, по-нататък (в процеса на проектиране) те могат да бъдат преобразувани в зависимости.




фиг. 2.18. Главната диаграма на класовете от варианта на използване "Избор на дисциплини"

**Създаване на асоциации.**

Асоциациите се създават непосредствено на диаграмата на класовете. Панелът с инструменти на диаграмата на класовете има бутони за създаване на едно – двупосочни асоциации. За да се създаде **асоциация**:

1. От панела с инструменти натиснете Association.
2. С мишката прокарайте линия на асоциация от единия до другия клас.

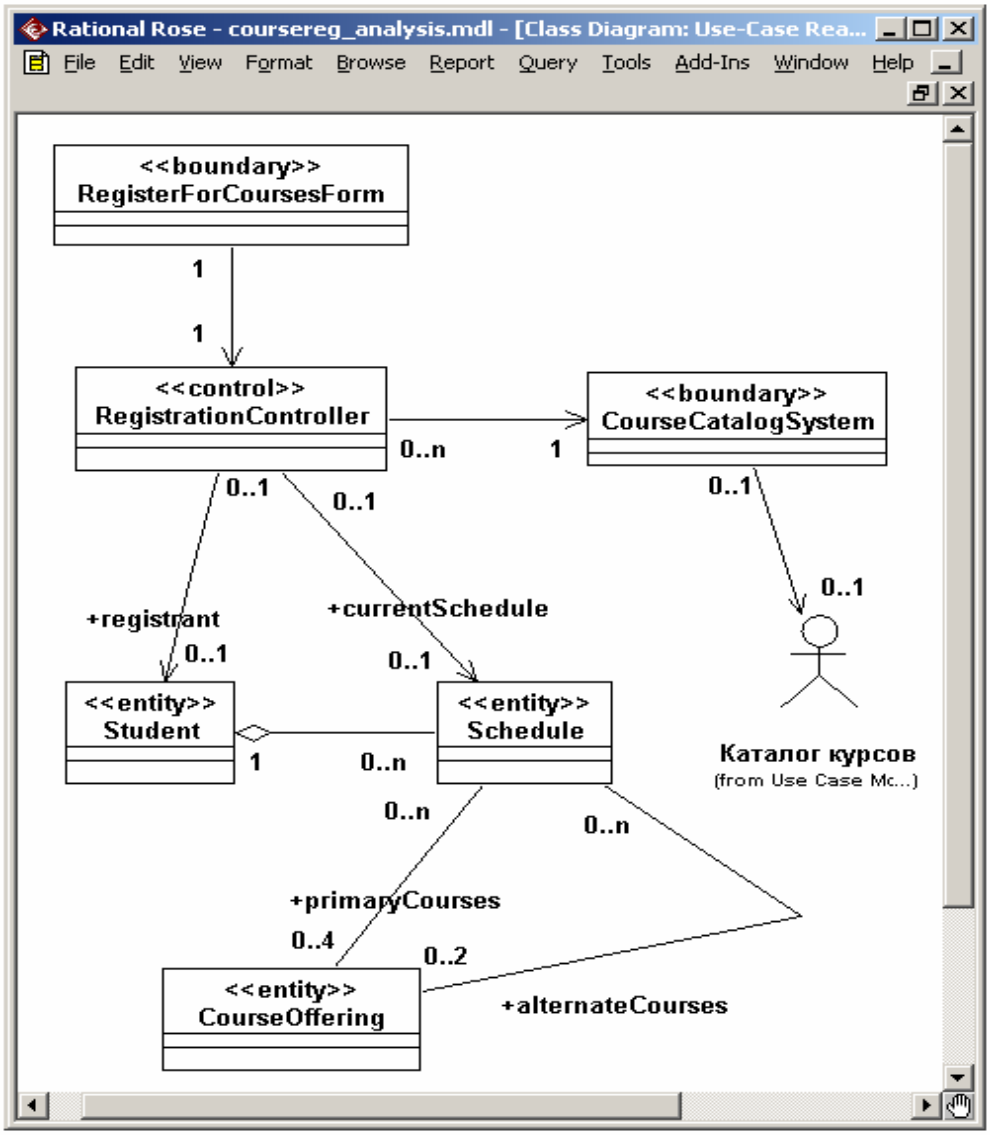
За да създадете възможност за **навигация** (насочване) на асоциацията:

1. Кликнете с десен бутон на мишката в този край на връзката, на който искате да поставите стрелка.
2. В отвореното меню изберете Navigable.

За да създадете рефлексивна асоциация:

1. От панела с инструментите натиснете Association.
2. Прекарайте линия на асоциация от класа до някое място извън класа.
3. Освободете бутона на мишката.

4. Прекарайте линия на асоциация обратно към класа.




фиг. 2.19. Диаграма на класовете с асоциациите

**Създаване на агрегации.**

1. От панела с инструментите натиснете Association.
2. Прекарайте линия на агрегация от класа - част към целия.

За да създадете върху диаграмата на класовете рефлексивна агрегация:

1. От панела с инструментите натиснете Aggregation..
2 Прекарайте линия на агрегация от класа до някое място извън класа.

3 Освободете бутона на мишката.
4 Прекарайте линия на агрегация обратно към класа.

**Създаване на обобщения.**

При създаване на обобщения може да се наложи да бъдат пренесени някои атрибути или операции от един клас в друг. Ако например, се наложи те да се пренесат от един подклас в суперклас, то в браузъра за това е достатъчно просто да пренесете атрибутите или операциите от един клас в друг. Не забравяйте да премахнете копието на атрибута от втория подклас, ако има такъв.

За да разположите обобщение върху диаграмата на класовете:

1. Натиснете Generalization от панела с инструментите.
2. Прокарайте линия на обобщение от подкласа до суперкласа.

**Спецификация на връзките.**

Спецификацията на връзките се отнася до имената на асоциациите, имената на ролите, множествеността и до класовете на асоциациите.

За да създадете множественост на връзка:

1. Кликнете с десен бутон на мишката в единия край на връзката.
2. В отвореното меню изберете Multiplicity.
3. Посочете необходимата множественост.
4. Повторете това и за другия край на връзката.

За да създадете име на връзката:

1. Маркирайте дадената връзка.
2. Въведете име.

За да създадете име на ролята:

1. Кликнете с десен бутон на мишката в дадения край на асоциацията.
2. В отвореното меню изберете role Name.
3. Въведете име на ролята.

За да създадете елемент на връзката (клас на асоциации).

1. Отворете прозореца на спецификациите за дадената връзка.
2. Преминете към Detail.
3. Задайте елемент на връзката в полето Liub Element.

Задание за самостоятелна работа.

Направете анализ на варианта на използване Close Registration и постройте съответните диаграми на взаимодействието и на класовете.

**2.10 Проектиране на системата.**

Целта на обектно-ориентираното проектиране е да се адаптира предварителния системен проект (съвкупността от класовете на “анализа”), представляващ значима основа на архитектурата на

системата, към средата за реализиране с отчитане на всички нефункционални изисквания.

Обектно-ориентираното проектиране включва два вида дейности:

- проектиране архитектурата на системата;
- проектиране елементите на системата.

**2.10.1 Проектиране архитектурата на системата.**

Проектирането архитектурата на системата се извършва от архитекта на системата и включва:

- идентификация на архитектурните решения и механизми, необходими за проектиране на системата;
- анализ на взаимодействията между класовете на анализа, определяне на подсистемите и интерфейсите;
- създаване на архитектурните нива;
- проектиране структурата на потоците на управление;
- проектиране конфигурацията на системата.

Първото, което архитектурата трябва да направи при определяне на подсистемите, е преобразуване на класовете на анализа в проектни класове (design classcs). За всеки клас от анализа се приема едно от две възможни решения:

- класът от анализа се преобразува в проектен клас, ако той е прост или представя единствена логическа абстракция;
- сложен клас от анализа може да бъде разбит на няколко класа, преобразуван в папка или в подсистема.

Обединяването на класовете в подсистеми става като се изхожда от следните съображения:

- функционална връзка: обединяват се класове, участващи в различни варианти на използване и взаимодействащи само един с друг;
- обезателност: съвкупност от класове, реализираща функционалност, която може да бъде премахната от системата или заменена с алтернативна;
- свързаност: обединяване в подсистеми на силно свързани класове;
- разпределение: обединяване на класове, разположени в конкретен възел на мрежата.

Примери за възможни подсистеми:

- съвкупност от класове осигуряващи сложен комплекс от функции (напр. осигуряване на безопасност и защита на данните);
- гранични класове, реализиращи сложен потребителски интерфейс или интерфейс с външни системи;

- различни продукти: комуникационно ПО, достъп до бази данни, общи библиотеки, различни приложни пакети.

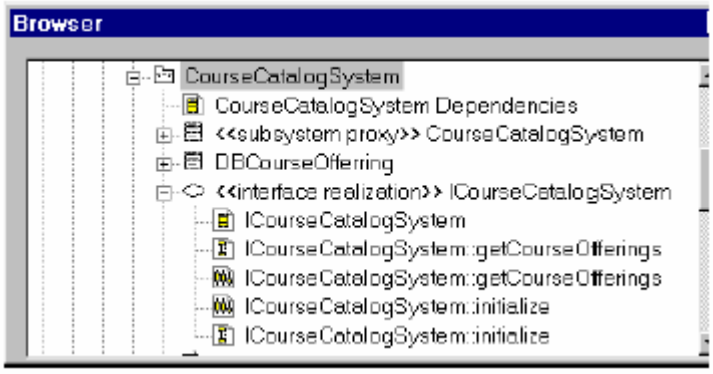
При създаването на подсистеми в модела се извършват следните преобразования:

- обединяваните класове се разполагат в специална папка с името на подсистемата и със стереотип <subsystem>;
- спецификациите на операциите в класовете, образуващи подсистема, се извеждат в интерфейса на подсистемата – клас със стереотип <Interface>;
- описанието на интерфейса трябва да включва:
  * име на интерфейса, отразяващо неговата роля в системата;
  * текстово описание на интерфейса с размер на кратък абзац, отразяващ неговото предназначение;
  * описание на операциите на интерфейса (име, отразяващо резултата от операцията, възстановяваното значение, параметри с техните типове);
  * характерът на използване на операциите на интерфейса и реда за тяхното изпълнение се документира с помощта на диаграми на взаимодействието, описващи взаимодействието на класовете от подсистемата при реализиране операциите на интерфейса, които заедно с диаграмата на класовете от подсистемата се обединяват като кооперация с името на интерфейса и със стереотип <interface realization>;
- в подсистемата се създава клас-посредник със стереотип <subsystem proxy>, управляващ реализацията на операциите на интерфейса.

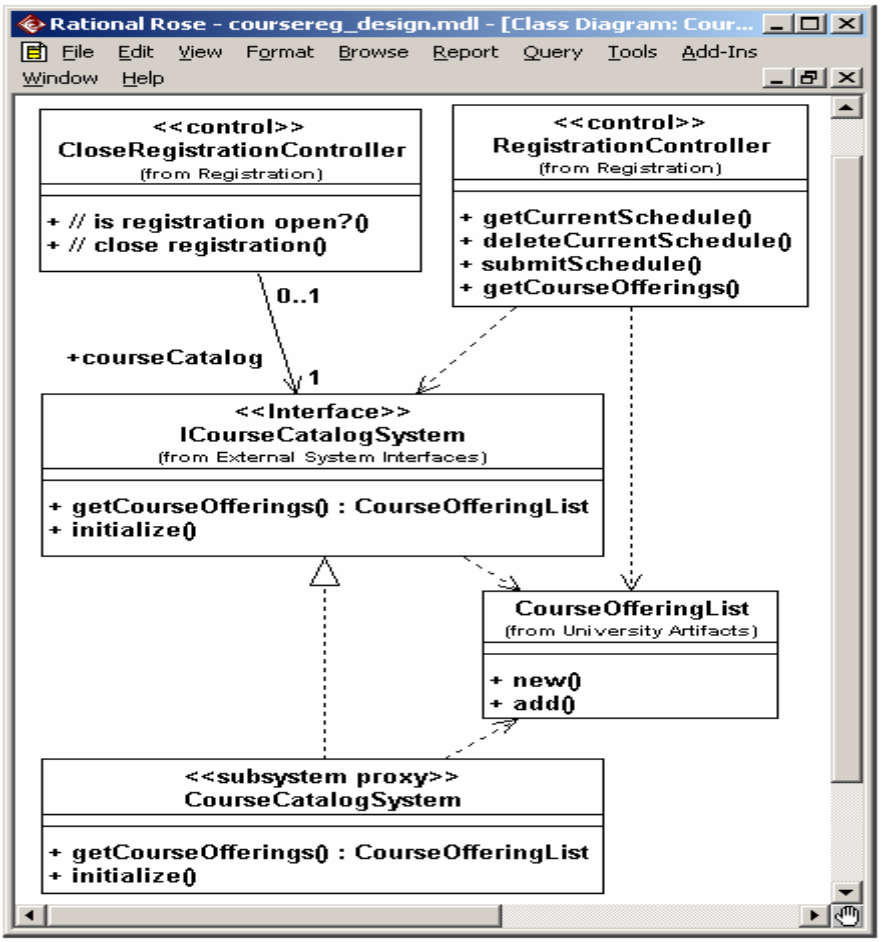
Всички интерфейси на подсистемите трябва да бъдат напълно определени в процеса на проектиране на архитектурата, доколкото те ще служат като точки на синхронизация при паралелната разработка на системата.

В качеството на пример (за системата на избор) ще посочим подсистемата CourseCatalogSystem, която е създадена вместо граничния клас CourseCatalogSystem. Структурата и диаграмите на папката (подсистемата) CourseCatalogSystem (диаграмите на класовете и на взаимодействието, описващи дадената подсистема и нейния интерфейс), са показани на фиг.2.20÷2.23. Класовете DBCourseOffering и CourseOfferingList отговарят на взаимодействието с БД на каталога (списъка) с дисциплините в рамките на JDBC. Обектът CourseCatalogSystem Client от фиг.2.23 може да принадлежи или на класа CloseRegistrationContriller, или на класа RegistrationContriller, в

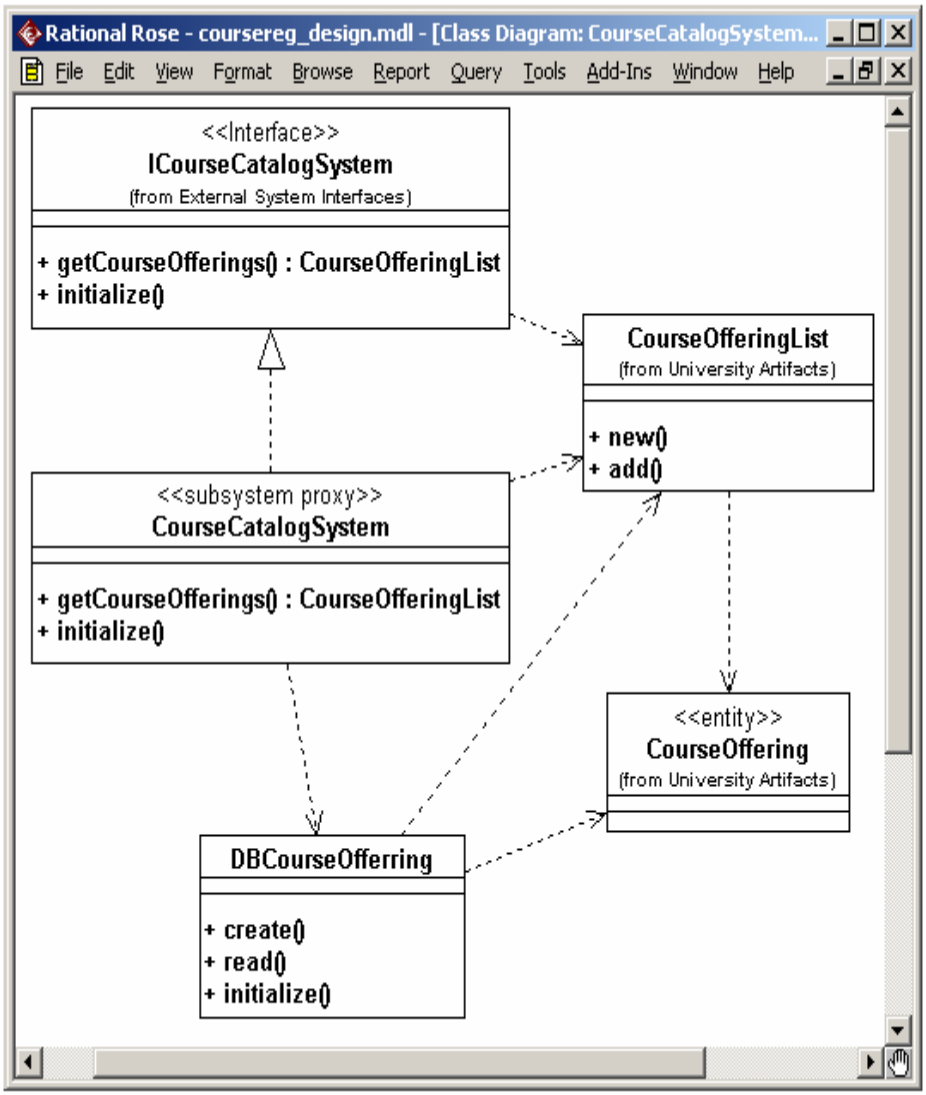
зависимост от това в кой от вариантите на използване се прави обръщение към каталога с дисциплините.




фиг.2.20 Структура на пакета CourseCatalogSystem.




фиг.2.21. Контекст на подсистемата CourseCatalogSystem от гледна точка на архитекта.




фиг.2.22. Диаграма на класовете от подсистема CourseCatalogSystem от гледна точка на проектанта.

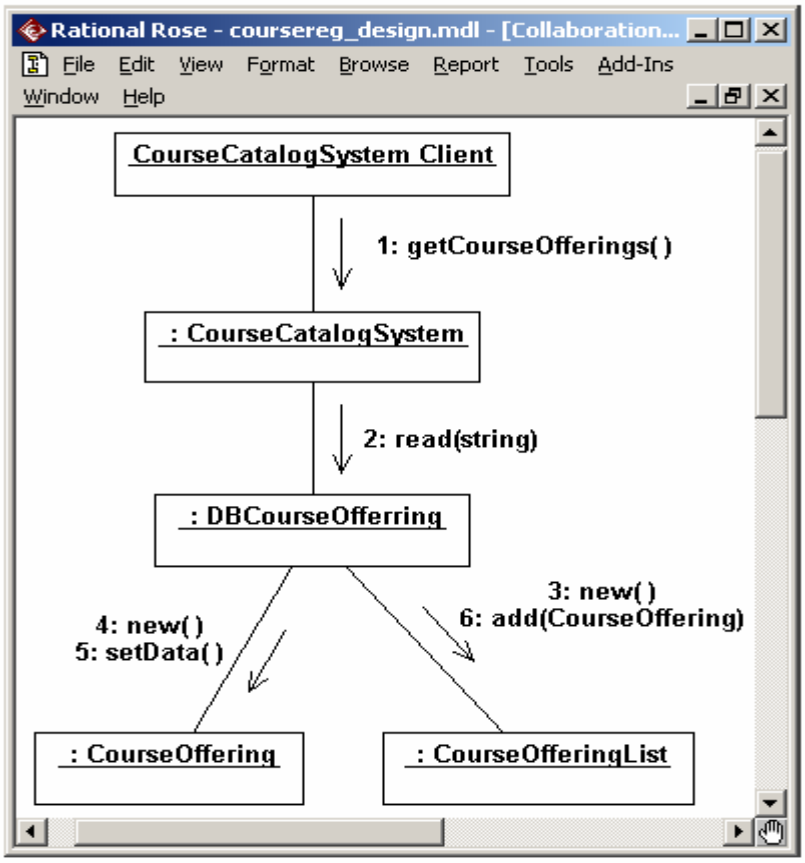
**Форматиране на архитектурните нива**

В процеса на анализа беше взето решение за отделяне на архитектурните нива. В процеса на проектиране всички елементи на системата трябва да бъдат разпределени по нива. От гледна точка на модела това означава разпределяне на проектните класове по папки, съответстващи на архитектурни нива (папки със стереотип <layer>). При сложна система архитектурните нива могат да се разбият на поднива, броят и структурата на които зависят от сложността на

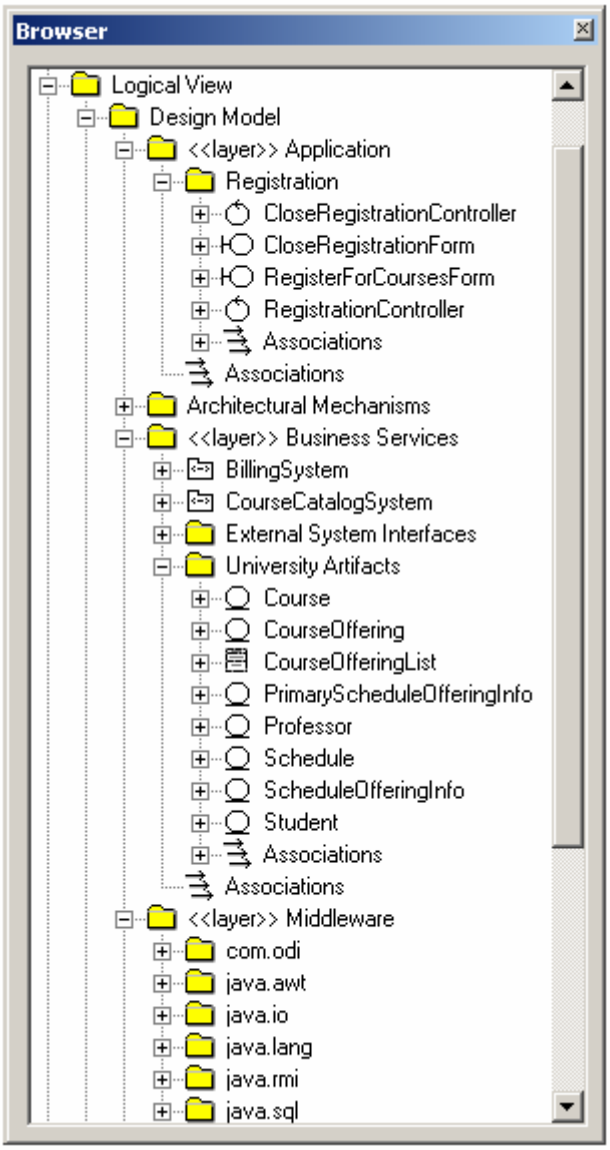
предметната област и от средата за реализация. Поднивата се формират като се изходи от следните критерии:

- групиране на елементите с максимална свързаност;
- разпределяне в съответствие със структурата на организация (обикновено това касае горните нива на архитектурата);
- разпределяне в съответствие с различните области на компетенция на разработчиците (предметна област, мрежи, комуникации, бази данни, безопасност и др.);
- групиране на отделните компоненти на разпределената система;
- разпределяне в съответствие с различната степен на безопасност и секретност.

Пример за отделяне на архитектурните нива и за обединяване на елементите на модела в папки за системата на избор е показан на фиг.2.24.




фиг.2.23. Обединена диаграма, описваща реализацията на операцията на интерфейса getCourseOfferings.




фиг.2.24 Браузъра на проекта

Това представяне отразява следните решения, възприети от архитекта:

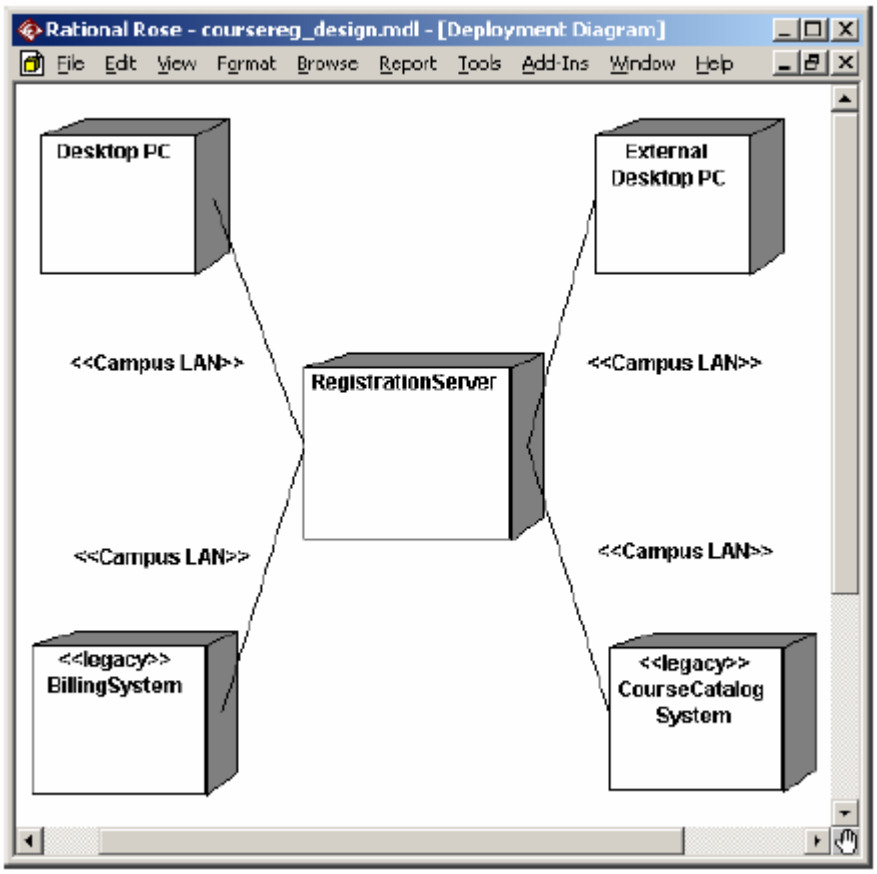
- определени са три архитектурни нива 9създадени са три папки със стереотип <layer>);
- в папка Application е създадена папка Registration, където са включени граничните и управляващи класове;
- граничните класове BillingSystem и CourseCatalog System са преобразувани в подсистеми;

- в папка Susiness Services, освен подсистемите, са включени още две папки: папка External System Interfaces включва интерфейсите на подсистемите (класовете със стереотип <Interface>, а папка University Artifacts включва всички класове на същността.

**Моделиране на разпределената конфигурация на системата.**

- възел (node) – изчислителен ресурс (процесор или друго устройство (дискова памет, контролери на различните устройства и др.)). За възела може да се зададат изпълняваните на него процеси;
- връзка (connection) – канал за взаимодействие между възлите (мрежа).

Пример: мрежова конфигурация на системата за избор (без процесите) (фиг. 2.25).

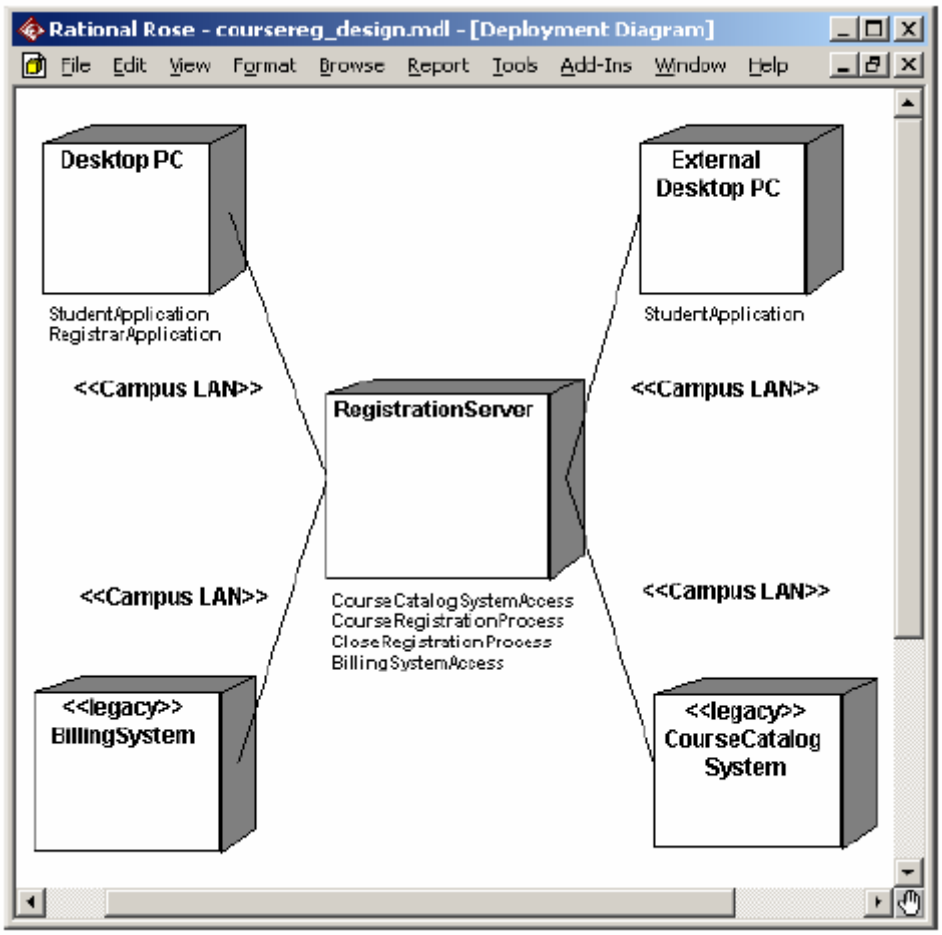



фиг. 2.25. Мрежова конфигурация на системата за избор.

Разпределението за процесите по възлите на мрежата се прави като се отчетат следните фактори:

- използваните начини на разпределяне (различни конфигурации);
- време за обратен отговор;
- минимизация на мрежовия трафик;
- мощност на възела;
- надеждност на оборудването и комуникациите.

Пример за разпределяне на процесите по възлите на мрежата е показан на фиг.2.26.




фиг.2.26. Мрежова конфигурация на системата за избор с разпределени на процесите по възли.

***Упражнение 17. Създаване на диаграма на разполагането от системата за избор.***

За да се отвори диаграмата на разполагане, трябва да се кликне два пъти върху Deployment View (представяне на разполагането) в браузъра.

За да се разположи процесор в/у диаграмата:

1. От панела с инструменти натиснете Processor.
2. Кликнете в диаграмата на разполагането там, където искате да го разположите.
3. Въведете името на процесора.

В спецификациите на процесора може да се въведе информация за неговия стереотип, за неговите характеристики и планиране. Стереотипите се използват за класификация на процесорите (напр. компютри под управлението на UNIX или PC).

Характеристиките на процесора – това е неговото физическо описание. В частност то може да включва скорост на процесора, обем на паметта и др.

Полето за планиране (sheduling) на процесора съдържа описание на това, как се осъществява планирането на неговите процеси:

- Preemptive (с приоритет). Високоприоритетните процеси имат предимство пред тези с по-малък приоритет.
- Non preemptive (без приоритет). За процесите не е предвиден приоритет. Текущият процес се изпълнява до завършването му, след което започва изпълнението на следващия.
- Cyclic (цикличен). Управлението се предава между процесите кръгово. За всеки процес се отделя определено време за изпълнението му, след което управлението преминава към следващия процес.
- Executive (изпълнителен). Съществува определен изчислителен алгоритъм, който управлява планирането на процесите.
- Manual (ръчно). Потребителят сам планира процесите.

За да определите стереотип на процесора:

1. Отворете процесора със спецификацията на процесора.
2. Преминете на General.
3. Въведете стереотипа в поле Stereotype.

За да въведете характеристики и планиране на процесора:

1. Отворете процесора със спецификацията на процесора.
2. Преминете на Detail.
3. Въведете характеристики в полето на характеристиките.
4. Посочете един от видовете планиране.

За да покажете планирането на диаграмата:

1. Кликнете с десен бутон на мишката върху процесора.

2. В отвореното меню изберете Show Scheduling.

За да добавите връзка в диаграмата:

1. От панела с инструменти натиснете Connection.
2. Кликнете във възела на диаграмата.
3. Прекарайте линия за връзка от единия до другия възел.

За да изберете стереотип на връзката:

1. Отворете прозореца на спецификацията на връзката.
2. Преминете на General.
3. Въведете стереотипа в поле Stereotype.

За да добавите процес:

1. Кликнете с десен бутон на мишката върху процесора в браузъра.
2. В отвореното меню изберете New -> Process.
3. Въведете името на новия процес.

За да покажете процесите на диаграмата:

1. Кликнете с десен бутон на мишката върху процесора.
2. В отвореното меню изберете Show Process.

### 2.10.3 Проектиране на класовете.

Всеки граничен клас се преобразува в набор от класове, в зависимост от своето предназначение. Това може да бъде набор от елементи на потребителският интерфейс, зависещ от възможностите на средата за разработване, или пък набор от класове, реализиращ системния или апаратния интерфейс.

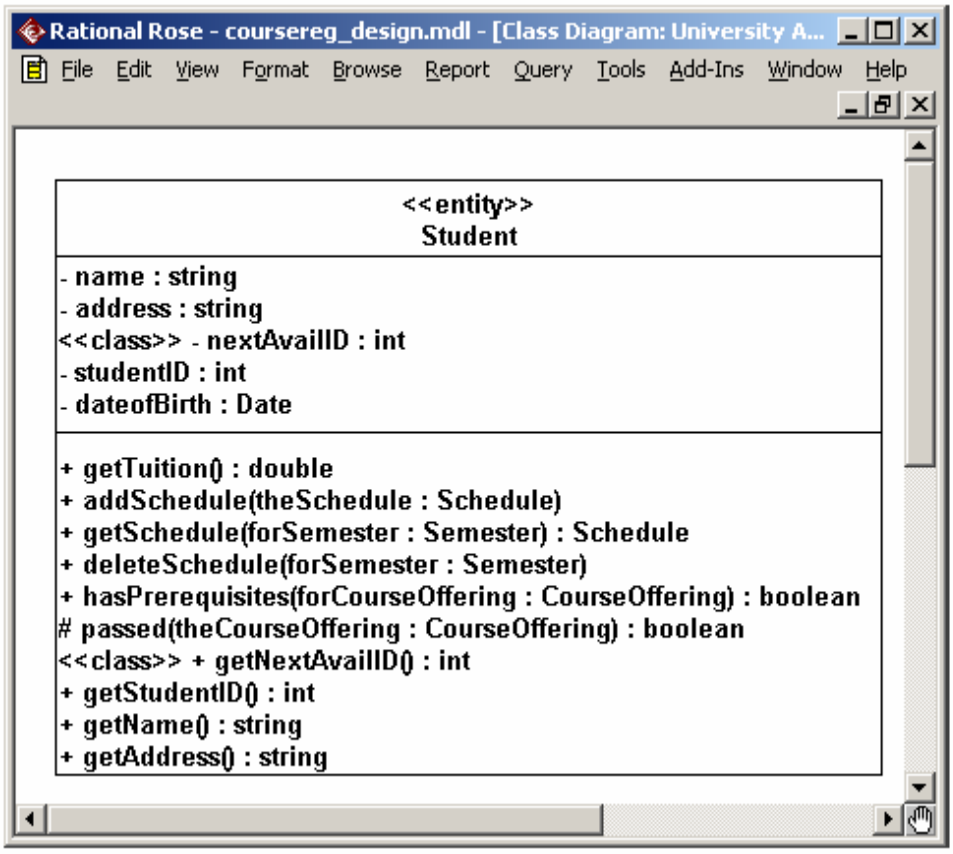
Класовете на същността с отчитане на съображенията за съобразителност и на защитата на данните могат да се разбият на редица класове. Предпоставка за разбиването е наличието в класа на атрибути с различна честота на използване или видимост. Такива атрибути, като правило, се отделят в отделни класове.

Що се касае до управляващите класове, то класовете, реализиращи елементарно предаване на информация от6 граничните класове към същностите, могат да бъдат отстранени. Запазват се класовете, изпълняващи съществена работа до управлението на потоците от събития (управление на транзакциите, разпределена обработка и т.н.).

Ангажиментите на класовете, определени в процеса на анализа, се преобразуват в операции. На всяка операция се съпоставя име, което характеризира нейния резултат. Определя се пълната сигнатура на операцията: operationName (paremeter:class, ...):returnType. Прави се кратко описание на операцията, като се включва смисъла на всички нейни параметри. Определя се видимостта на операцията: public, private, protected. Определя се областта на действие на операцията: екземпляр или класификатор.

Определят се (уточняват се) атрибутите на класовете:

- Освен името, се задават и типа и значението по премълчаване: attributeName:Type=Default;
- Отчитат се съглашенията по именуване на атрибутите, възприети в проекта, а също и езика за реализация;
- Задава се видимостта на атрибутите: public, private, protected.
- При необходимост се определя и производни атрибути.




фиг.2.27. Клас Sudent с начално определени операции и атрибути.

***<u>Упражнение 18. Определяне на атрибутите и операциите на класа Sudent.</u>***

За да бъдат зададени типа на данните, значението по премълчаване и видимостта на атрибутите:

1. Кликнете с десен бутон на мишката върху атрибута на браузъра.
2. В отвореното меню изберете Open Specification.

3. В отворения списък на типовете укажете типа на данните или въведете собствен тип на данните.
4. В полето Initial Field (начално значение) въведете значение на атрибута по премълчаване.
5. В полето Export Control изберете видимост на атрибута: Public, Protected, Private или Implementation. По премълчаване видимостта на всички атрибути съответства на Private.

За да промените нотацията за означаване на видимостта:

1. В менюто на модела изберете Tools -> Options.
2. Преминете към Notation.
3. Маркирайте превключвателя Visibility as Icons, за да използвате нотацията от Rose, или изберете разклонението, за да ползвате нотацията от UML.

Забележка: Изменението на значението на този параметър ще доведе до смяна на нотацията само за новите диаграми и няма да засегне вече съществуващите диаграми.

За да зададете тип на възстановяваното значение, стереотипа и видимостта на операцията:

1. Кликнете с десен бутон на мишката върху операцията от браузъра.
2. Отворете прозореца със спецификацията на класа за тази операция.
3. Укажете (посочете) типа на възстановяваното значение в отворения списък или въведете свой тип.
4. Укажете (посочете) стереотипа в отворения списък или въведете свой тип.
5. В полето Export Control посочете значението за видимост на операцията: Public, Protected, Private или Implementation. По премълчаване видимостта на всички операции е public.

За да добавите аргумент към операцията:

1. Отворете прозореца със спецификация на операцията.
2. Преминете към Detail.
3. Кликнете с десен бутон на мишката в областта на аргумента, в отвореното меню изберете Insert.
4. Въведете име на аргумента.
5. Кликнете върху Data type и въведете там типа на данните на аргумента.
6. Ако е необходимо, кликнете върху default и въведете значение на аргумента по премълчаване.

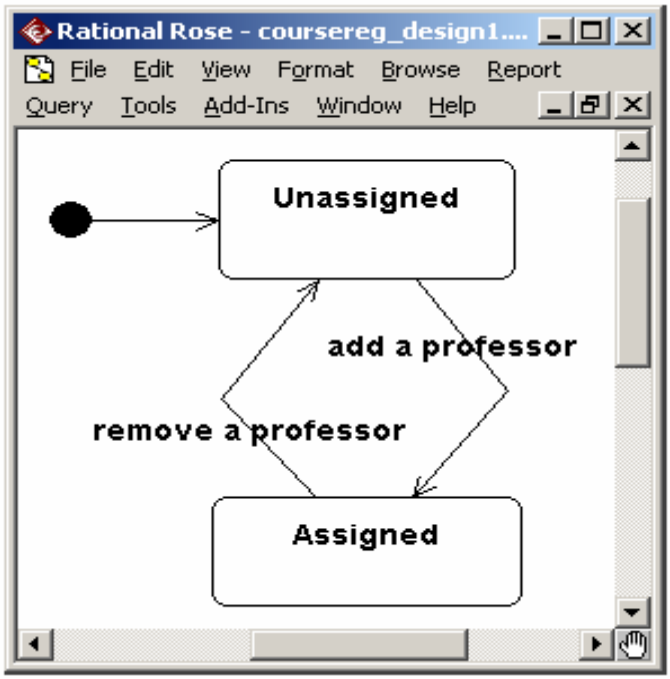
**Моделиране състоянията на класовете.**

Ако един обект реагира винаги еднозначно на дадено събитие, то казваме той е независим от състоянието по отношение на това събитие. В отличие от него, зависимите от състоянието обекти реагират различно на едно и също събитие в зависимост от своето състояние. Обикновено в икономическите информационни системи има съвсем малко обекти, които са зависими от състоянието, докато при системите за управление на технологични процеси (системите в реално време) често съдържат множество от такива обекти.

Ако в системата са налице зависими от състоянието обекти със сложна динамика на поведение, то за тях може да се построи модел, описващ състоянията на обекта и преходите между тях. Такъв модел се представя във вид на диаграми на състоянията.

В качеството на пример, свързан със системата за избор, ще разгледаме поведението на обект от класа CourseOffering. Диаграмата на състоянията се построява на няколко етапа:

1. *Идентификация (определяне) на състоянията.* Признаци за определяне на състоянията са изменението на значенията на атрибутите на обекта и създаването или премахването на връзки с другите обекти. Така напр. обектът CourseOffering може да бъде в състояние Open (изборът на дисциплина е отворен) до тогава, докато броят на избралите тази дисциплина студенти не стане повече от 10. щом броят стане равен на 10, то обектът преминава в състояние на Closed (изборът на дисциплината е прекратен). Освен това обектът CourseOffering може да бъде и в състояние Unassigned (означава, че тази дисциплина не е предложена от никой лектор и няма връзка с Professor) или я състояние Assigned (когато има такава връзка).
2. *Идентификация на събитията.* Като правило, събитията са свързани с изпълнението на определени операции. Така напр. в класа CourseOffering в резултат на разпределяне на ангажиментите при анализ на варианта на използване “Избор на дисциплини за преподаване” се определят две операции – addProfessor и removeProfessor, свързани с избора на дадена дисциплина от лектор (създава се нова връзка) или отказ от дадена дисциплина (премахване на връзката). На тези операции се поставят в съответствие две събития - addProfessor и removeProfessor.
3. *Идентификацията на преходите между състоянията.* Преходите са предизвикани от събития. По такъв начин, състоянията Unassigned и Assigned се свързват с два прехода (фиг.2.28).



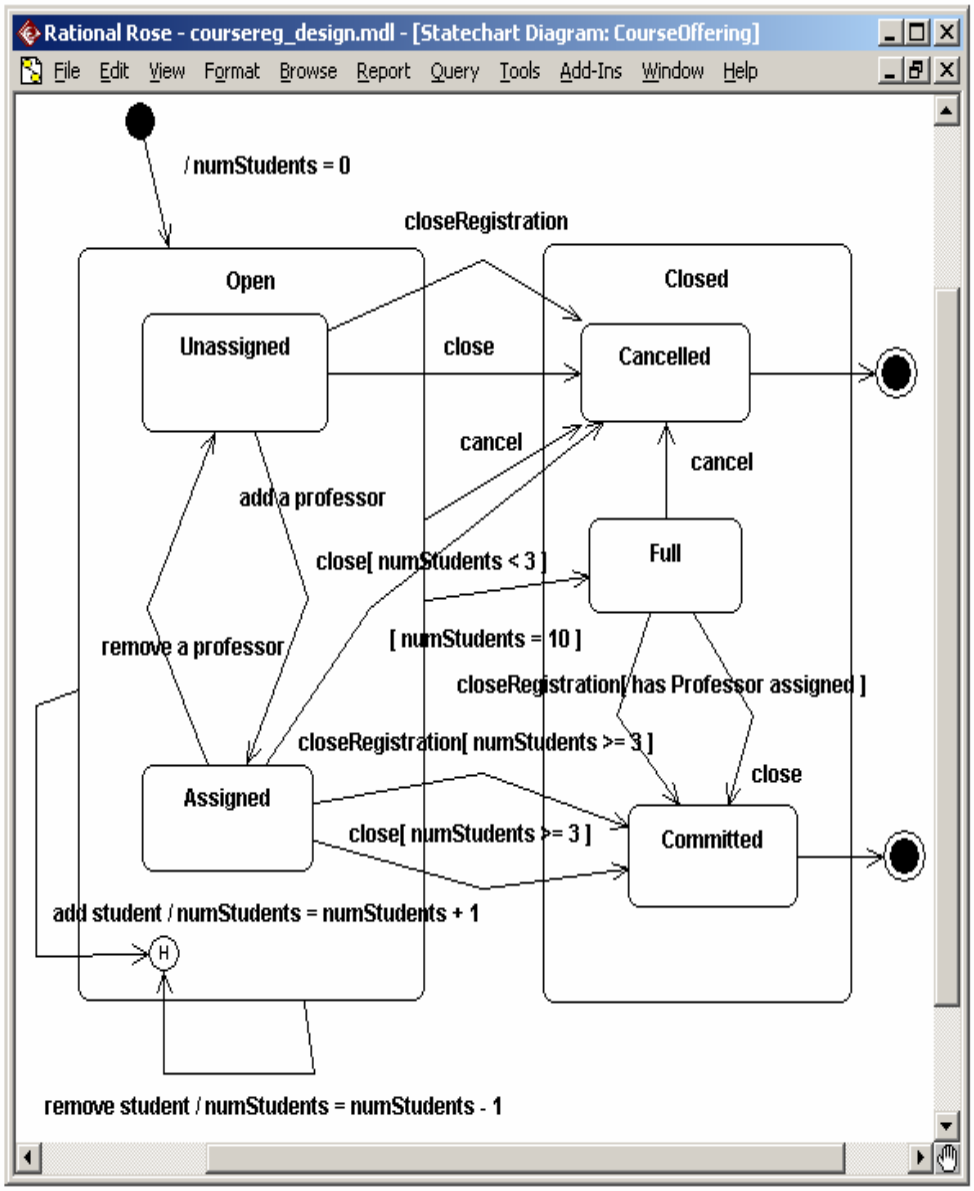

фиг.2.28.Преходи между състоянията.

По нататъшната детайлизация на поведението на обекта CourseOffering ще доведе до построяване на диаграма на състоянията – фиг.2.29. на тази диаграма са използвани такива възможности на моделирането на състоянията, като

- композитни състояния (composite state) и
- историческо състояние (history state).

В дадения случай композитни състояния са Open и Closed, а вложени състояния са – Unassigned, Assigned и Cancelled (дисциплината е отменена), Full (дисциплината е запълнена) и Committed (дисциплината е включена в разписанието). Композитните състояния позволяват да се опрости диаграмата като се намали броя на преходите, доколкото вложените състояния наследяват всички свойства и преходи на композитното състояние

Историческото състояние (на диаграмата с "H") – е псевдо-състояние, което възстановява предишното активно състояние в композитно състояние. То позволява на композитното състояние Open да запомни кое от вложените състояния (Unassigned или Assigned) е било текущо в момента на използване от Open. Това е необходимо, за да може кой да е от преходите в Open (add student или remove student) да се връща именно в това вложено състояние, а не в началното състояние.




фиг.2.29. Диаграма на състояния с композитни състояния.

***<u>Упражнение 19. Създаване на диаграма на състояния за класа CourseOffering.</u>***

За да създадете диаграма на състоянията:

1. Кликнете с десен бутон на мишката върху избрания клас от браузъра.
2. В отвореното меню изберете New -> Statechart Diagram.

За да добавите състояние:

1. От панела с инструментите натиснете State.
2. Кликнете с мишката на това място от диаграмата на състоянията, където искате да го разположите.

Всички елементи на състоянието може да добавите като използвате Detail от прозореца на спецификацията на състоянието.

За да добавите дейност:

1. Отворете прозореца на спецификацията за даденото състояние.
2. Преминете към Detail.
3. Кликнете с десен бутон на мишката в прозорец Actions.
4. В отвореното меню изберете Insert.
5. Кликнете два пъти на новото действие.
6. Въведете действие в полето Actions.
7. В прозореца When посочете Do, за да стане новото действие дейност.

За да добавите входно действие, в прозореца When посочете On Entry.

За да добавите изходно действие, в прозореца When посочете On Exit.

За да о5тправите събитие:

1. Отворете прозореца на спецификацията на даденото състояние.
2. Преминете към Detail.
3. Кликнете с десен бутон на мишката в прозорец Actions.
4. В отвореното меню изберете Insert.
5. Кликнете два пъти на новото действие.
6. Като тип на действие посочете Send Event.
7. В съответните полета въведете събитие (event), аргументи (arguments) и целеви обект (Target).

За да добавите преход:

1. Натиснете бутон Transition от панела с инструментите.
2. Кликнете с мишката върху състоянието, от което ще се прави преход.
3. Прекарайте линия на прехода до това състояние, където той завършва

За да добавите рефлексивен преход:

1. Натиснете бутон Transition to Self от панела с инструментите.
2. Кликнете върху това състояние, където се изпълнява рефлексивен преход.

За да добавите събитие, аргументите му, ограничително условие и действие:

1. Кликнете два пъти върху прехода, за да отворите прозореца с неговите спецификации.
2. Преминете към General.
3. Въведете събитие в полето Event.
4. Въведете аргументи в полето Arguments.
5. Въведете ограничително условие в полето Condition.
6. Въведете действие в полето Action.

За да отправите събитие:

1. Кликнете два пъти върху прехода, за да отворите прозореца с неговите спецификации.
2. Преминете към Detail.
3. Въведете събитие в полето Send Event.
4. Въведете аргументи в полето Send Arguments.
5. Задайте цел в полето Send Target.

За да посочите началното или крайното състояние:

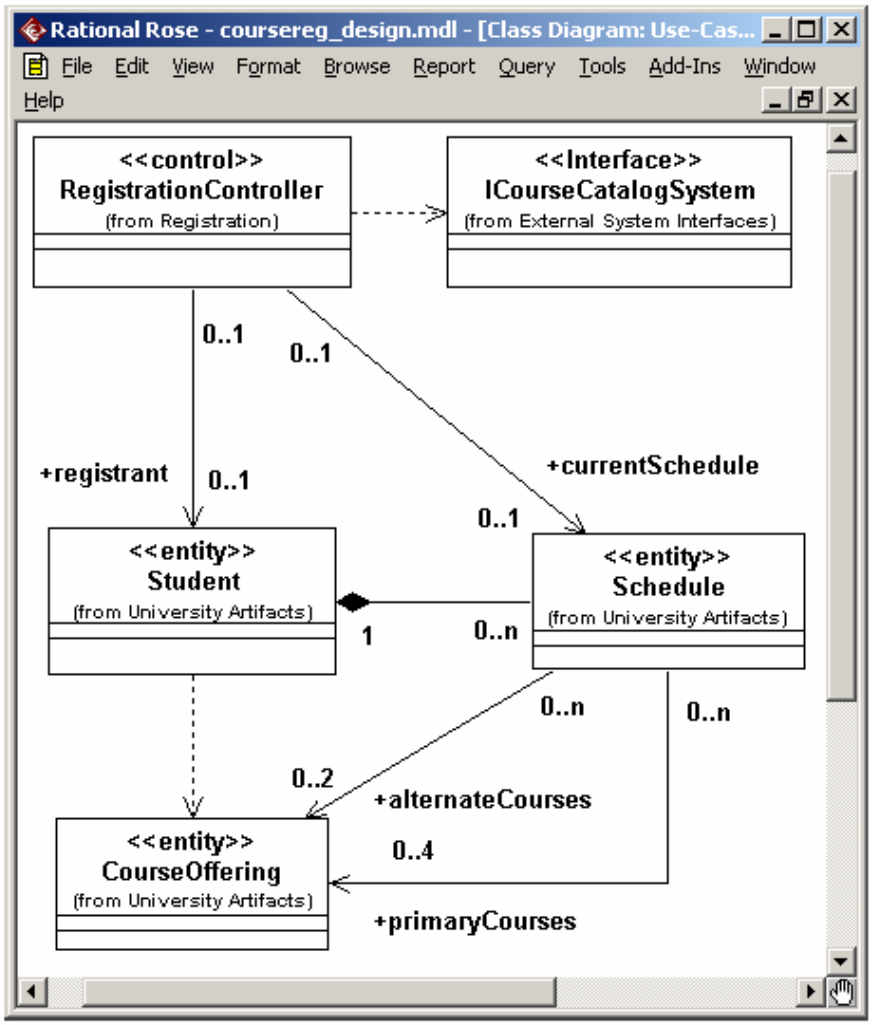
1. Върху панела с инструменти натиснете бутон Start State или End State.
2. Кликнете с мишката върху диаграмата на състоянията, където искате да разположите състоянието.

***Уточняване на връзките между класовете.***

В процеса на проектиране връзките между класовете (асоциации, агрегации и обобщения) подлежат на уточняване.

- Асоциациите между граничните и управляващите класове отразяват връзките, динамически възникващи между съществуващите обекти в потока на управление. За такива връзки е достатъчно да се осигури видимост на класовете, и затова те се преобразуват в зависимости.
- Ако за някоя асоциация не е необходима двупосочна връзка, то се въвеждат направления за навигация.
- Агрегации, които притежават свойствата на композиция се преобразуват във връзки на композиция.

Пример за преобразуване на връзките в съответствие с направените препоръки за класовете на варианта на използване “Избор на дисциплини” е показан на фиг.2.30. Асоциацията между управляващите и граничните класове е преобразувана в зависимост. Агрегацията между класовете Student и Schedule притежават свойствата на композиция. Направлението за навигация на асоциациите между класовете Schedule и CourseOffering са въведени от следните съображения: няма необходимост от получаване на списък на графика, в който няма нито една дисциплина, и броят на графиците е сравнително малък по отношение на броя на конкретните дисциплини.



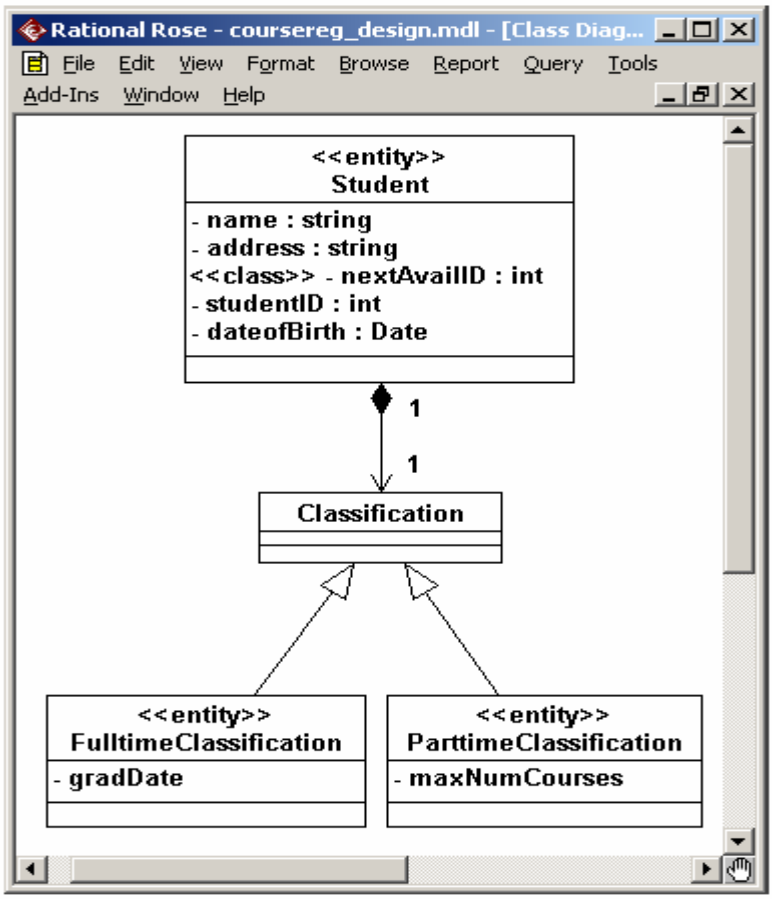

фиг.2.30. Пример за преобразуване на асоциации и агрегации.

Връзките на обобщението могат да се преобразуват в ситуации с така наречената метаморфоза на подтиповете. Например, в случая със системата за избор студентът може да преминава от редовна във задочна форма на обучение, т.е. обектът Student може да променя своя подтип.при такава промяна следва да се модифицира и описанието на обекта в системата. За да се избегне тази модификация, а същевременно и да се повиши устойчивостта на системата, йерархията на наследяване се реализира с помощта на класификации, както е показано на фиг.2.31.

За да преобразувате агрегацията в композиция:

1. Кликнете с десен бутон на мишката в този край на агрегацията, който се допира до класа-част (на фиг.2.20 – Schedule).
2. В отвореното меню изберете Containment.
3. Посочете метод на включване By Value.

Забележка: Значението By Value предполага, че цялото и частта се създават и разрушават едновременно, което съответства на композиция. Агрегацията (By Reference) предполага, че цялото и частта се създават и разрушават в различно време.




фиг.2.31. Йерархията на наследяване

**2.10.4. Проектиране на базите данни.**

Проектирането на релационни бази данни става с използването на средството Data Modeler. Неговата работа е основана на известния механизъм на представяне на обектния модел в релационен. Като резултат се получава построяването на диаграмата “същност-връзка” и последващо генериране на описанието на БД чрез SQL.

***<u>Упражнение 20. Проектиране на релационна БД.</u>***

Проектирането на БД преминава през следните стъпки:

Създаване на нов компонент на БД:

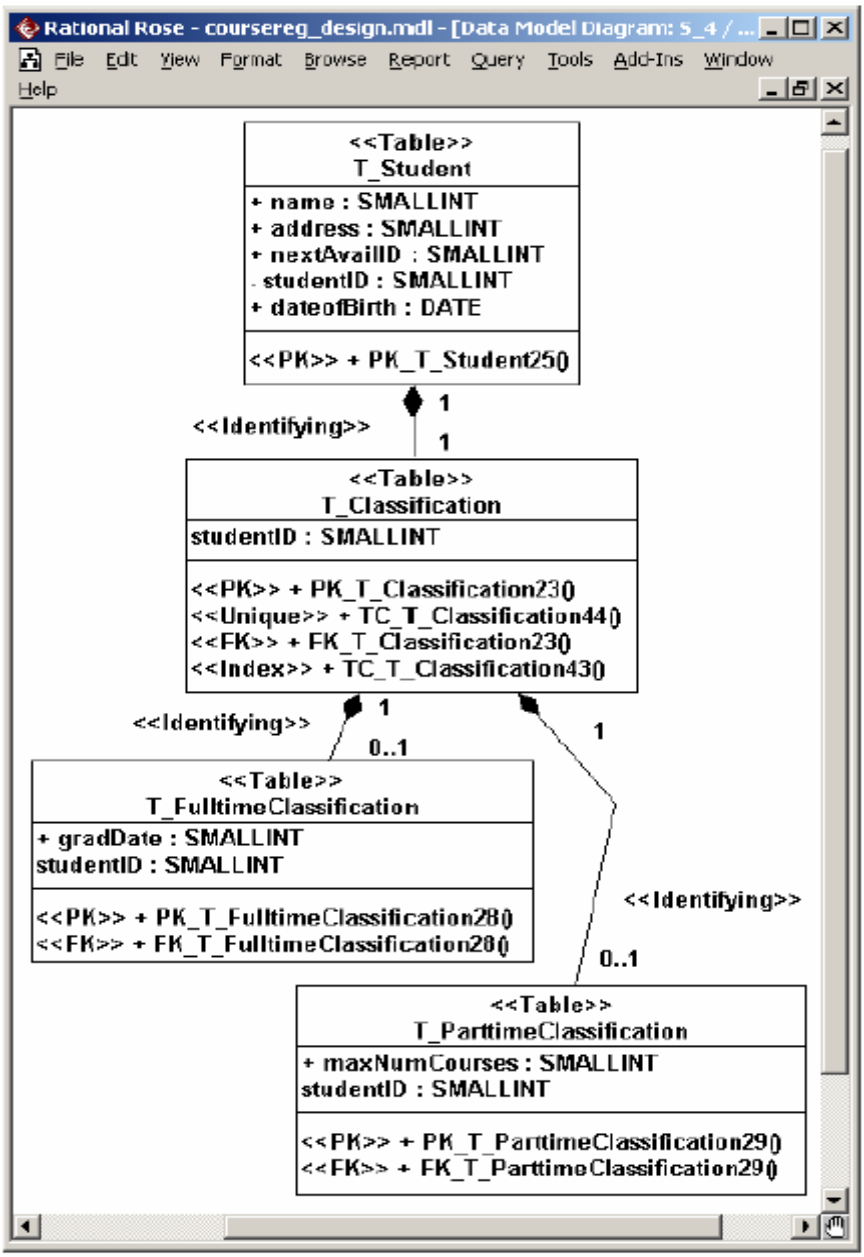
1. Кликнете с десен бутон на мишката върху представянето на компонентите.
2. В отвореното меню изберете Data Modeler -> New ->Database.
3. Отворете прозореца на спецификацията на създадения компонент DB_0 и в списъка Target изберете Oracle 8.x.

Определяне на устойчивите (persisten) класове:

1. Отворете прозореца на спецификацията на класа Student от папка University Artifacts.
2. Преминете към Detail.
3. Установете превключвателя Persistence в Persistent.
4. Изпълнете същите действия и за класовете Classification, Fulltimatelassification и Parttimelassification.
5. Отворете клас Student от браузъра, като натиснете “+”.
6. Кликнете с десен бутон на мишката върху атрибута studentID.
7. В отвореното меню изберете Data Modeler -> Part of Object Identity (посочване на атрибута в качеството на част от първичния ключ.

Създаване на схемата на БД:

1. Кликнете с десен бутон на мишката върху папка University Artifacts.
2. В отвореното меню изберете Data Modeler -> Transform to Data Model.
3. В появилия се прозорец в списъка Target Database посочете DB_0 и натиснете OK. Като резултат в логическото представяне ще се появи нова папка Schemas.
4. Отворете папка Schemas и кликнете с десен бутон на мишката върху папка << Schema>> S_0.
5. В отвореното меню изберете Data Modeler -> New -> Data Model Diagram.
6. Отворете папка, след това отворете създадената диаграма “същност-връзка” New Diagram и пренесете в нея всички класове-таблици, намиращи се в папка << Schema>> S_0. Получената диаграма е показана на фиг.2.32.




фиг.2.32. Диаграма “същност-връзка”

Генерирайте описанието на БД на SQL.

1. Кликнете с десен бутон на мишката в/у папка << Schema>> S_0.
2. В отвореното меню изберете Data Modeler -> Forward Engineer.

3. В отворения прозорец на Forward Engineering Wizard натиснете Next.
4. Отменете всички флагове на генерирането DDL и натиснете Next.
5. Посочете името и разположението на текстовия файл с резултатите на генерирането и натиснете Next.
6. След края на генерирането отворете създадения текстови файл и прегледайте резултатите.

## 2.11. Реализация на системата.

### 2.11.1. Създаване на компонентите.

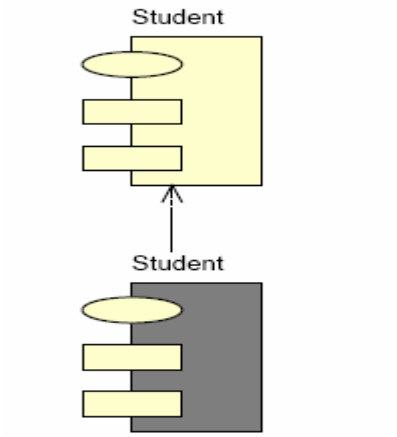
В RationalRose диаграмите на компонентите се създават като представяния на компонентите на системата. Отделните компоненти могат да се създават непосредствено в/у диаграмата или да се пренасят там от браузъра.

***Упражнение 21. Създаване на компонентите.***

В качеството на език за програмиране ще изберем C++ и за класа Student ще създадем съответните на този език компоненти.

Създаване диаграмата на компонентите:

1. Кликнете два пъти с мишката в главната диаграма на компонентите в представянето на компонентите.
2. От панела с инструменти натиснете Package Specification.
3. Разположете спецификацията на папката върху диаграмата.
4. Въведете име на спецификацията на папка Student и посочете в прозореца на спецификацията езика C++.
5. От панела с инструменти натиснете Package Body.




фиг.2.33. Диаграма на компонентите

6. Разположете пакета в/у диаграмата.
7. Въведете име на папка Student и посочете в прозореца на спецификацията езика C++.
8. На панела с инструменти натиснете Dependency.
9. Прекарайте линия на зависимост от тялото на папката към спецификацията на папката.

Съпоставяне на класовете с компонентите.

1. В логическото представяне на браузъра намерете класа Student.
2. Преместете този клас в спецификацията на папка на компонента Student в представянето на компонентите на браузъра. В резултат на това класът Student ще бъде съпоставен със спецификацията и тялото на папка на компонента Student.

**2.11.2.Генериране на кода.**

Процесът на генериране на кода се състои от шест основни стъпки:

1. Проверка коректността на модела.
2. Определяне свойствата на генериране на кода.
3. Избор на класа, компонента или папката.
4. Генериране на кода.

За да проверите модела:

1. Изберете в меню Tools -> Check Model.
2. Проанализирайте всички открити грешки в прозореца.

Към най-разпространените грешки могат да се причислят:

- съобщения на диаграмата на последователността или на обединената диаграма;
- несъответствие с операцията на обектите на тези диаграми или несъответствие с класа.

С помощта на Check Model от менюто могат да се покажат голяма част от неточностите и грешките на модела. Опцията Access Violations от менюто позволява да се открият нарушения на правилата за достъп, възникващи тогава, когато съществува връзка между два класа от различни папки, но по същество няма връзка между самите папки.

За да откриете нарушение на правилата за достъп:

1. Изберете в меню Report -> Show Access Violations.
2. Анализирайте всички нарушения на правилата за достъп в прозореца.

Може да се определят няколко параметъра на генерирането на кода за класове, атрибути, компоненти и други елементи на модела. Тези свойства определят начина на генерирането на програмите. За всеки език в Rose са предвидени редица определени свойства на генерирането

на кода. Преди генерирането на кода се препоръчва да се анализират тези свойства и да се внесат необходимите промени.

За анализа на свойствата на генериране на кода изберете Tools -> Options, а след това съответният език. В прозореца със списъка може да изберете клас, атрибут, операция или други елементи на модела. За всеки език в този списък са посочени негови собствени елементи на модела. При избиране на различните значения на екрана се появяват различни набори от свойства.

Промените, внасяни в набора от свойства в прозореца изберете Tools -> Options, се отразяват на всички елементи на модела, за които се използва този набор.

Понякога трябва да се променят свойствата на генериране на кода за един клас, за един атрибут, за една операция и т.н. за да направите това, отворете прозореца на спецификацията на елемента от модела. Изберете съответния език (C++, Java, ...) и направете промените там. Всички промени, внесени в прозореца на спецификацията на елемента от модела, оказват влияние само на този елемент.

При генерирането на кода наведнъж могат да се създадат клас, компонент или цяла папка. Кодът се генерира с помощта на диаграмата или браузъра. При генериране на кода от папката може да се избере или папката с логическото представяне в/у диаграмата на класовете или папката с представянето на компонентите в/у диаграмата на компонентите. При избиране на папката с логическо представяне се генерират всички класове на тази папка. При избиране на папката с представяне на компонентите се генерират всички компоненти на тази папка.

След избора на класа или компонента в/у диаграмата изберете в менюто съответния вариант за генериране на кода. Съобщенията за грешки, възникващи в процеса на генериране на кода, ще се появят в прозореца на журнала.

По време на генерирането на кода Rose избира информация от логическото и компонентно представяне на модела и генерира голяма част от “скелетния” (skeletal) код:

- Класове. Генерират се всички класове на модела.
- Атрибути. Кодът включва атрибутите на всеки клас, в това число: видимост, тип на данните и значение по премълчаване.
- Сигнатура на операциите. Кодът съдържа определения на операциите със всички параметри, типове данни на параметрите и с типа на възстановяваното значение на операцията.
- Връзки. Някои от връзките на модела предизвикват създаване на атрибути при генериране на кода.

- Компоненти. Всеки компонент се реализира във вид на съответен файл с изходен код.

***<u>Упражнение 22. Генериране на кода на C++.</u>***

1. Отворете диаграмата с компонентите на системата.
2. Изберете всички обекти от диаграмата на компонентите.
3. Изберете Tools -> C++ -> Code Generation.
4. Изпълнете генерирането на кода.
5. Прегледайте резултатите от генерирането (меню Tools -> C++ -> Browse Header и Tools -> C++ -> Browse Body).

Найден Ненков Валентин Вичев

ЕКСПЕРТНИ СИСТЕМИ

Рецензент:

доц. д-р Петър Стойков

Научен редактор:

доц. д-р инж. Иван Кръстев Цонев