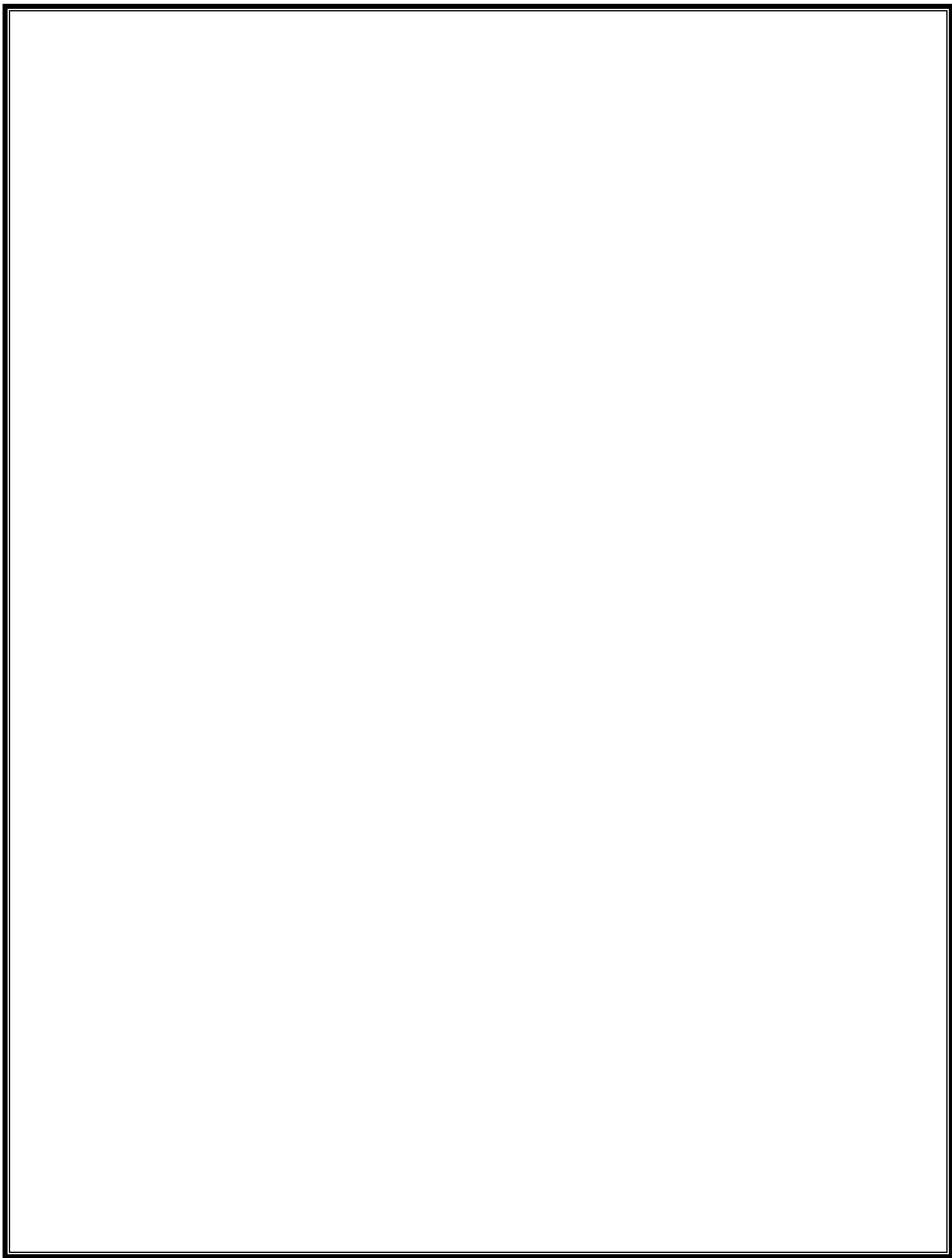




آزمایشگاه مهندسی نرم افزار

رشنال رز

مقدمه :

در این نوشتار می خواهیم همراه با یک مثال ساده، چگونگی استفاده از رشنال رز را برای مدل کردن یک سیستم تشریح کنیم. در هر مرحله همراه با آموزش مفاهیم مثال خود را نیز گسترش خواهیم داد. برای راحتی کار شما پروژه در هشت مرحله انجام شده است که هر قسمت در فایل جداگانه ی مشخص شده در دیسکت ضمیمه می باشد، تا مراحل کاری به خوبی تفهیم گردند.

حتما از خود می پرسید چرا مدلسازی؟

تا چند سال پیش همه ی رشته ها برای خود زبان مشترک داشتند، مثلا ریاضیدانان از زبان ریاضی و مهندسين الكترونيك از زبان مداری استفاده می کردند ولی مهندسين نرم افزار زبان واحدی نداشتند ، تا اینکه UML پا به عرصه ی وجود گذاشت و باعث شد که مهندسين نرم افزار بتوانند مفاهيم مورد نیاز خود را براحتی با همدیگر در میان بگذارند و کارهایشان را مدل کنند.

یک مدل خوب مثل نقشه ی ساختمان :

- نیازمندی ها را مشخص می کند.
- ارتباطات بین قسمت های مختلف پروژه را به ما می نمایاند.
- بدون وارد شدن به جزییات می توانیم در نحوه ی فعل و انفعالات قسمت های مختلف پروژه تمرکز نماییم.
- در یک تیم کاری، بعلت وجود یک زبان گرافیکی مشترک، ارتباط بین افراد تیم بهبود می یابد.

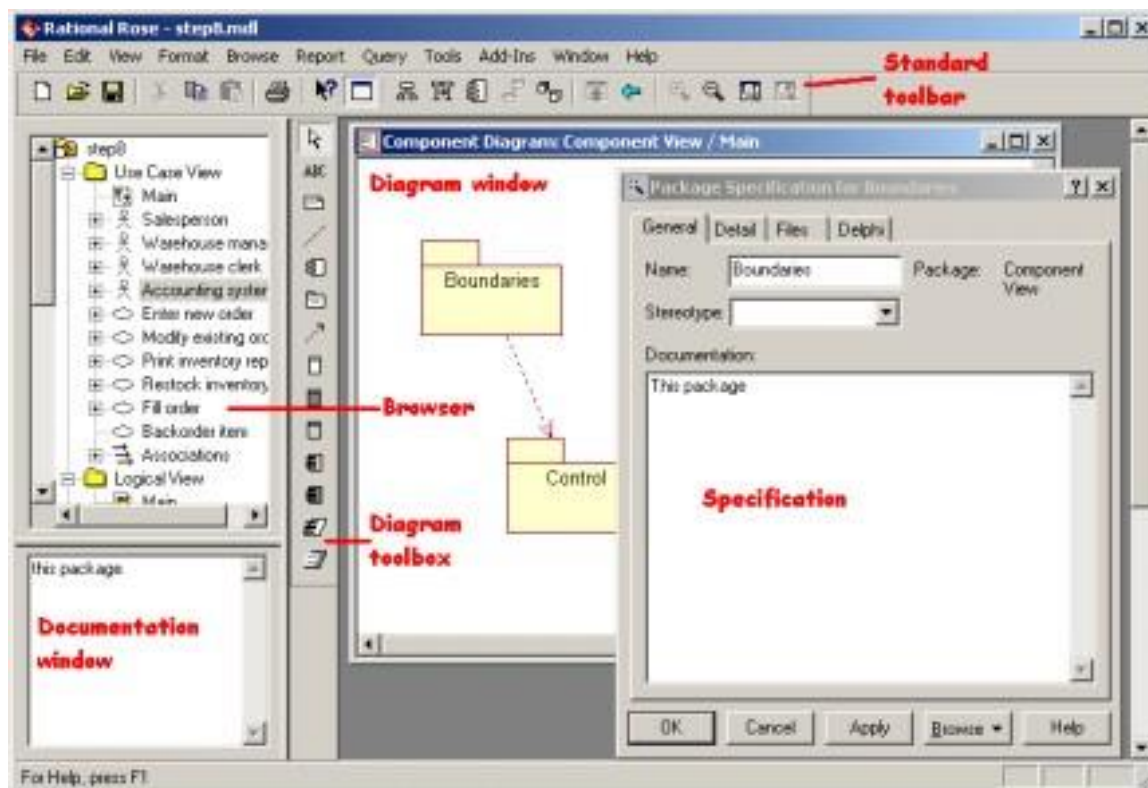
در اولین گام از این نوشتار به معرفی عناصر عمده ی رشنال رز می پردازیم. این قسمت شامل بخش های زیر می باشد :

- معرفی محیط رشنال رز
- معرفی نمادهای رشنال رز
- معرفی دیگرام های UML که در رشنال رز بکار گرفته می شوند.

معرفی محیط رشنال رز

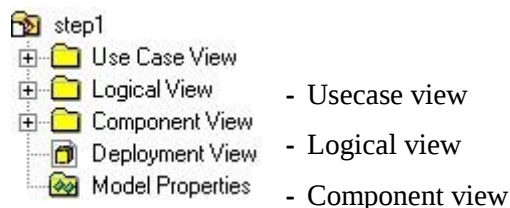
عناصر اصلی رشنال رز عبارتند از:

۱. **Standard toolbar** : که برای تمام دیاگرام ها مشترک است و در قسمت بالای پنجره واقع است.
۲. **Diagram toolbar** : که وابسته به پنجره ی دیاگرام فعال است و در سمت چپ پنجره ی دیاگرام واقع است.
۳. **Browser** : به شما اجازه می دهد تا بصورت یک ساختار درختی دیاگرام های موجود و عناصر مدلهایتان را مشاهده کنید.
۴. **Diagram window** : ساخت و ویرایش دیاگرام ها در این قسمت صورت می پذیرد.
۵. **Documentation window** : به شما اجازه می دهد تا به مدلهایتان مستندات لازم را نیز اضافه نمایید .
می توانید مستنداتتان را در این قسمت یا در قسمت specification ویرایش نمایید.
۶. **Specification** : محیط ویرایشی برای اضافه کردن مستندات به مدل.



معرفی نماهای رشنال رز

برای یک پروژه ی در حال ساخت نماهای مختلفی وجود دارد . رشنال رز نماهای زیر را برای یک پروژه فراهم می آورد که هر کدام یکی از جنبه های مختلف مدل را نمایش میدهند:



- Deployment view

نمای مورد استفاده ی سیستم (usecase view) :

این نما جهت فهم و استفاده از سیستم پیش بینی شده است . در این نما که تشریح رفتار سیستم از دیدگاه کاربر است، فعل و انفعالات متقابل بازیگرها (actors) و موردهای استفاده نمایش داده می شود. در این نما چهار دیگرام زیر وجود دارند :

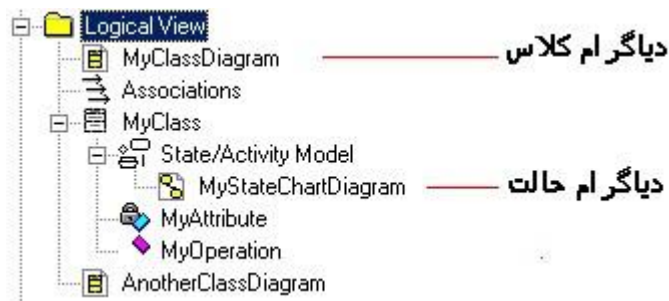
- دیگرامهای توالی (sequence diagrams)
- دیگرامهای همکاری (collaboration diagrams)
- دیگرامهای فعالیت (activity diagrams)



نمای منطقی سیستم (logical view) :

این نما شامل نیازمندی های عملیاتی سیستم می باشد که به کلاسها و ارتباط بین آنها می پردازد. این نما شامل دو دیگرام زیر می باشد:

- دیگرامهای کلاسها (class diagrams)
- دیگرامهای حالت (statechart diagrams)



نمای اجزای سیستم (component view) :

این نما به سازمان سیستم می پردازد و اطلاعاتی در باره نرم افزار، اجزا قابل اجرا و کتابخانه های سیستم دارد. تنها دیاگرام موجود در این قسمت دیاگرام اجزا (component diagram) میباشد.



نمای پیاده سازی سیستم (deployment view) :

این قسمت شامل نگرانی از فرایندهای موجود با سخت افزار سیستم می باشد. این قسمت بخصوص در جاهایی حایز اهمیت است که شما دارای برنامه ها و سرورهایی در قسمت های مختلف مکانی هستید و می خواهید به بهترین شیوه ساختار توزیعی محیط را نشان دهید.

این نما فقط شامل یک دیاگرام (deployment diagram) می باشد.



معرفی دیاگرامهای رشنال رز

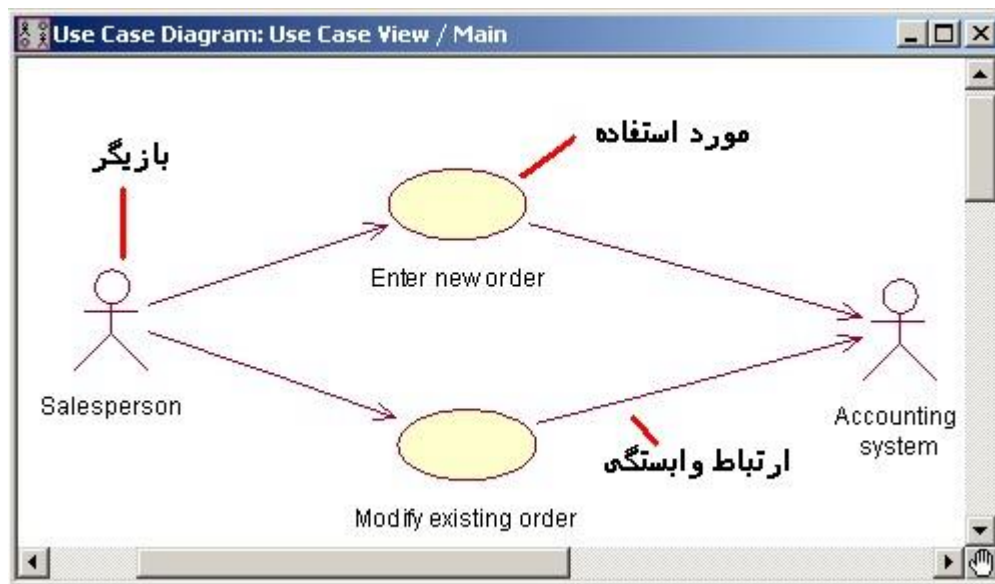
یک دیاگرام یک نمایش گرافیکی از عناصر سیستم‌تان می باشد. دیاگرامهای گوناگون به شما اجازه می دهند تا پرسپکتیوهای مختلف از سیستم‌تان را ببینید. شما در رشنال رز قادر به ایجاد دیاگرامهای زیر می باشید:

- دیاگرام مورد استفاده (usecase diagram)
- دیاگرام کلاس (class diagram)
- دیاگرام توالی (sequence diagram)
- دیاگرام همکاری (collaboration diagram)
- دیاگرام فعالیت (activity diagram)
- دیاگرام حالت (statechart diagram)
- دیاگرام اجزا (component diagram)
- دیاگرام پیاده سازی (deployment diagram)

دیاگرام موردهای استفاده (usecase diagram)

یک usecase رفتار سیستم را توصیف می کند، که شامل تقابل بین سیستم و بازیگران می باشد. بعبارت کلی یک usecase یک الگوی رفتاری توانایی های سیستم و یک دنباله تراکشن های وابسته به هم می باشد، که بوسیله ی سیستم و بازیگران انجام می پذیرد و دیاگرام موردهای استفاده یک نمای سطح بالای سیستم از دید بازیگران سیستم به ما می دهد و نحوه ی برخورد آن با دنیای بیرون را مشخص می کنند. این دیاگرام در طول تحلیل سیستم برای بدست آوردن نیازمندی ها و نشان دادن چگونگی کارکرد سیستم بکار می رود.

نکته: در یک مدل واحد ممکن است چندین دیاگرام مورد استفاده داشته باشیم، معمولاً در این صورت هر مجموعه ای از usecase های مرتبط به هم را در یک بسته قرار می دهیم. به این ترتیب کلیه ی usecase های سیستم در بسته های مرتبط به هم قرار می گیرند.



برای رسم یک دیاگرام مورد استفاده ابتدا در نمای usecase view روی آیکون main دو کلیک کنید تا دیاگرام مربوط به آن باز شود، کافی است که از جعبه ابزار سمت چپ، آیکون new use case را انتخاب کرده و بعد از قرار دادن روی دیاگرام با دو کلیک کردن روی آن در فرم specification نام، توضیحات و stereotype مربوط به آن را وارد نمایید.

Use Case Specification for Enter new order

General | Diagrams | Relations | Files

Name: Enter new order Package: Use Case View

Stereotype: []

Rank: [] ☐ Abstract

Documentation:

This use case will allow a salesperson to add a new order into the system.

OK Cancel Apply Browse Help

اضافه کردن یک بازیگر هم مثل یک مورد استفاده می باشد. برای ساختن یک وابستگی جدید بین دو عنصر نیز روی آیکون association کلیک کرده و این وابستگی را با کلیک کردن روی عنصر اول و کشیدن تا عنصر دوم بسازید. سپس با دو کلیک کردن روی آن مشخصات آن را وارد نمایید.

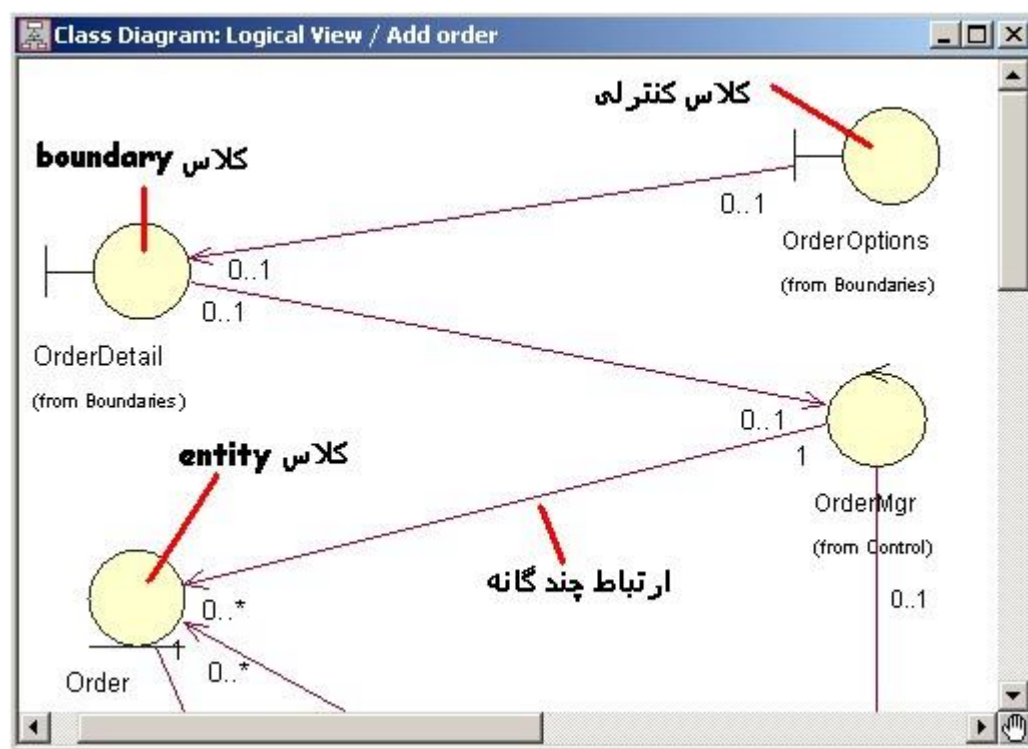
چگونگی اضافه کردن یک usecase موجود به یک دیاگرام دیگر: در browser دو دیاگرام مبدا و مقصد را باز (expand) کرده و سپس روی usecase مورد نظر کلیک کرده و آن را به دیاگرام مقصد بکشید. usecase مورد نظر در دیاگرام مقصد نیز ظاهر خواهد شد. اصولاً استفاده ی مجدد از عناصر موجود در سایر دیاگرامها نیز به همین شکل میباشد.

دیاگرام کلاس ها (class diagram)

این دیاگرام به شما کمک می کند تا نمای ساختاری سیستم تان را بصورت بصری (visual) در آورید. این دیاگرام از معمول ترین دیاگرامهای UML می باشد. این دیاگرامها جزئیات هر کلاس و ارتباطات بین آنها را نشان می دهد و پایه و اساس دیاگرامهای اجزا و پیاده سازی می باشد. در یک مدل واحد ممکن است چندین دیاگرام کلاس داشته باشیم.

در یک دیاگرام کلاس با کلاسهایی با سه نوع stereotype متفاوت زیر سروکار داریم:

- boundary
- control entity



کلاسهای boundary : اجزای لازم برای برقراری ارتباط سیستم با یک بازیگر را در خود دارند . این کلاسها می توانند پنجره ها، سنسورها، ترمینالها یا ... باشند. مثلا نجره ی گرفتن اسم رمز جهت ورود به برنامه، یک کلاس boundary میباشد. این نوع کلاسها معمولا جهت استفاده، با یک کلاس از نوع کنترلی در ارتباط هستند.

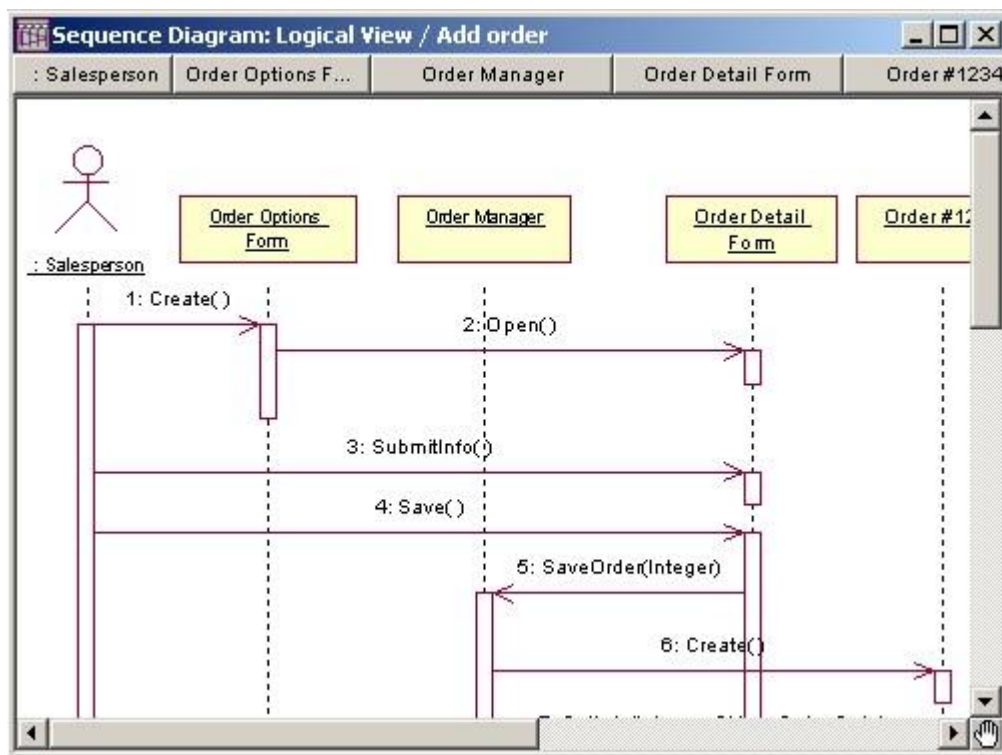
کلاسهای کنترلی: این کلاسها معمولا اشیا دیگر و رفتارهای تعبیه شده در یک usecase را کنترل می کنند.

کلاسهای entity : این کلاسها اطلاعاتی را که باید توسط سیستم ذخیره گردند را در خود نگهداری می کنند . محل نگهداری ساختمان داده های منطقی سیستم، این کلاسها می باشند.

دیاگرام توالی (sequence diagram)

دیاگرام توالی بصورت منظم و در یک توالی زمانی پشت سر هم ارتباطات متقابل اشیا را به ما نشان می دهد . این دیاگرام برای انجام عمل خاصی در یک usecase مشخص، مراحل انجام کار را مرحله به مرحله به شما نشان می دهد، یعنی دنباله ای از رویدادها را برای انجام یک عمل مشخص می سازد. در مراحل تحلیل و طراحی برای فهم نحوه ی عملکرد سیستم از این دیاگرام استفاده می شود.

جهت ساخت یک دیاگرام توالی در قسمت browser رشنال رز در قسمت usecase view روی نود مورد استفاده ی مورد نظر راست کلیک کرده و new و سپس sequence diagram را انتخاب نمایید . روی نود جدید ایجاد شده دو کلیک کنید تا پنجره ی دیاگرام مربوط به آن باز شود.



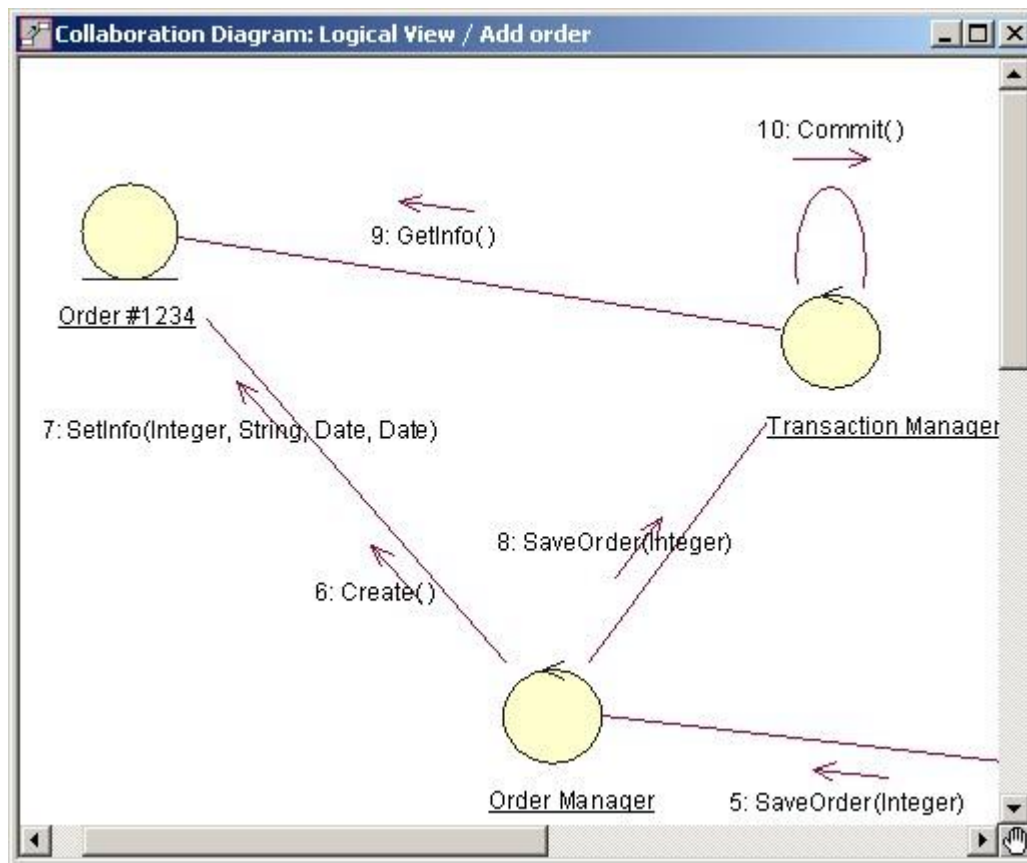
برای اضافه کردن کلاسها و یا با زیگرهای موجود به یک sequence diagram عناصر موجود در قسمت های قبلی را در browser کشیده و به داخل این دیاگرام بیندازید.

اگر می خواهید هنگام اضافه کردن پیغامها، عدهایی نیز به آنها اضافه گردد (همانند شکل فوق) تا توالی کار را بدرستی نشان دهد در منوی tools روی options کلیک کرده و به برگه ی دیاگرام رفته و جعبه ی انتخاب sequence numbering را علامت بزنید و سپس ok کنید. برای اضافه کردن پیغامها نیز می توانید از جعبه ابزار دیاگرام، آیکن object message استفاده کنید.

دیاگرام همکاری (collaboration diagram)

یک نما از ارتباطات ساختاری بین اشیا در مدل فعلی را بما می دهد. تاکید این دیاگرام بر ارتباط بین اشیا است در حالیکه تاکید دیاگرامهای توالی بر روی دنباله ای از رویدادها بود. این نوع دیاگرام شامل اشیا، لینک ها، وپیغام ها می باشد. برای درک و فهم چگونگی رفتار سیستم و اتخاذ هر گونه تصمیمی در این مورد از این دیاگرام استفاده کنید.

ساختن دیاگرامهای همکاری از دیاگرامهای توالی: رشنال رز این کار را بصورت خودکار برای شما انجام می دهد. کافی است از منوی browse گزینه ی create collaboration diagram را انتخاب نمایید، یا کلید F4 را بزنید. عکس این عمل هم، یعنی ساخت دیاگرام توالی از دیاگرام همکاری، با همین ترتیب فوق امکان پذیر است.



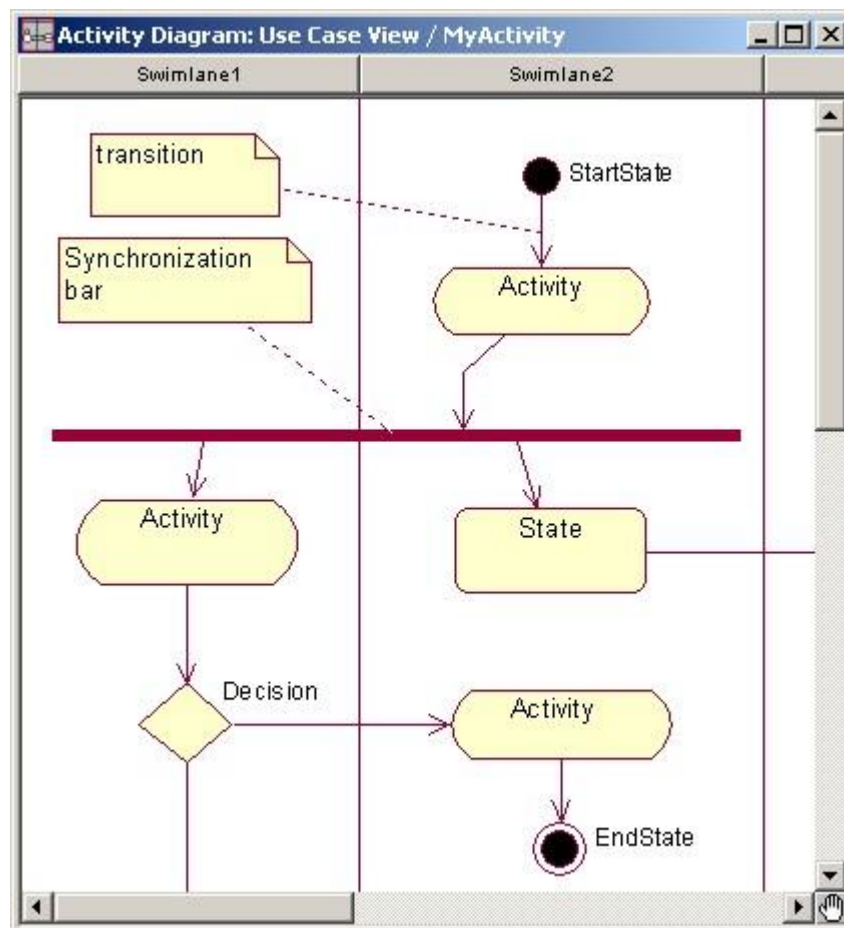
دیاگرام فعالیت (activity diagram)

دیاگرام فعالیت، جریان کار و همچنین توالی فعالیت ها را در یک فرآیند مشخص می کند. این دیاگرام خیلی شبیه فلوچارت است، زیرا شما می توانید جریان کار را از یک فعالیت به فعالیت دیگر یا به حالتی دیگر، دنبال نمایید.

دیاگرامهای فعالیت همچنین در جاهایی که می خواهید رفتارهای موازی را توصیف کنید، یا چگونگی نشان دادن عکس العمل در مقابل یک وضعیت چندگانه را مشخص کنید، مفید هستند.

یک دیاگرام فعالیت می تواند عناصر زیر را داشت باشد:

- یک **start state** و یک **end state**
- **activity** ها، که مراحل را در جریان کاری نشان می دهند.
- **Transition** ها، که ترتیب کاری را نشان می دهند.
- **Decision** ها، که به شما اجازه ی تصمیم گیری ها را در دیاگرام می دهند.
- **Synchronization bar** ها : که اجازه ی نمایش کارهای موازی را در دیاگرام به شما می دهد.
- **Swimlane** ها : نقش های مسوول در یک فعالیت معین را مشخص میکند.



ساخت یک دیاگرام فعالیت

در browser ، روی نود usecase view راست کلیک کرده و new > activity diagram را انتخاب نمایید . یک دیاگرام فعالیت جدید ظاهر می شود، کافی است روی آن دو کلیک نمایید تا دیاگرام مربوط به آن باز شود.

برای ساخت swimlane ها روی آیکون مربوط به آن در جعبه ابزار دیاگرام کلیک کرده و سپس روی نمودار کلیک نمایید. Swimlane جدید ایجاد میشود. برای تنظیم فیلد های آن کافی است روی آن دو کلیک کنید، فرم مربوط به specification آن باز می شود.

سایر قسمت ها نیز روی جعبه ابزار دیاگرام مشخص است و اضافه کردن آنها همانند قسمت های قبل می باشد.

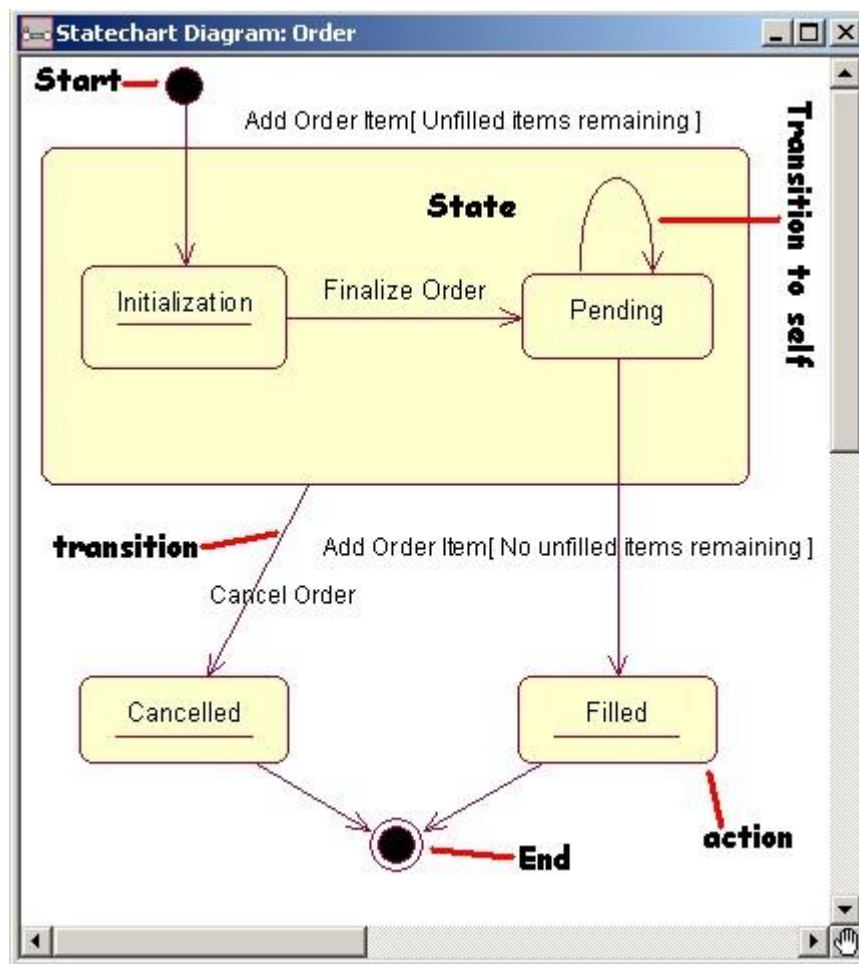
دیاگرام حالت (statechart diagram)

می توانید دیاگرام های حالت را برای مدل کردن رفتار پویای کلاس ها یا اشیا انفرادی استفاده کنید .این دیاگرامها ترتیب حالاتی که یک شی می تواند داشته باشد، رویدادهایی را که موجب انتقال از یک حالت یا فعالیت به دیگری می شوند و نتایجی را که این انتقال بوجود می آورد، را نمایش می دهد.

یک دیاگرام حالت، معمولاً برای مدل کردن مراحل گسسته ی چرخه ی حیات یک شی، بکار برده می شود ، در حالیکه دیاگرام فعالیت بر توالی فعالیت ها یی در یک فرآیند، دلالت دارد.

عناصر اصلی یک دیاگرام حالت عبارتند از:

- state ها (وضعیت های یک شی در طول حیات آن)
- حالت های شروع (start) و پایان (end)
- انتقال حالت ها (transitions)
- عملهای Entry ، Do و exit.



نکته: هر حالت در یک دیاگرام حالت می تواند شامل چندین عمل داخلی باشد. هر عمل، وظیفه ای است که در یک حالت رخ میدهد، که می تواند یکی از اعمال زیر باشد:

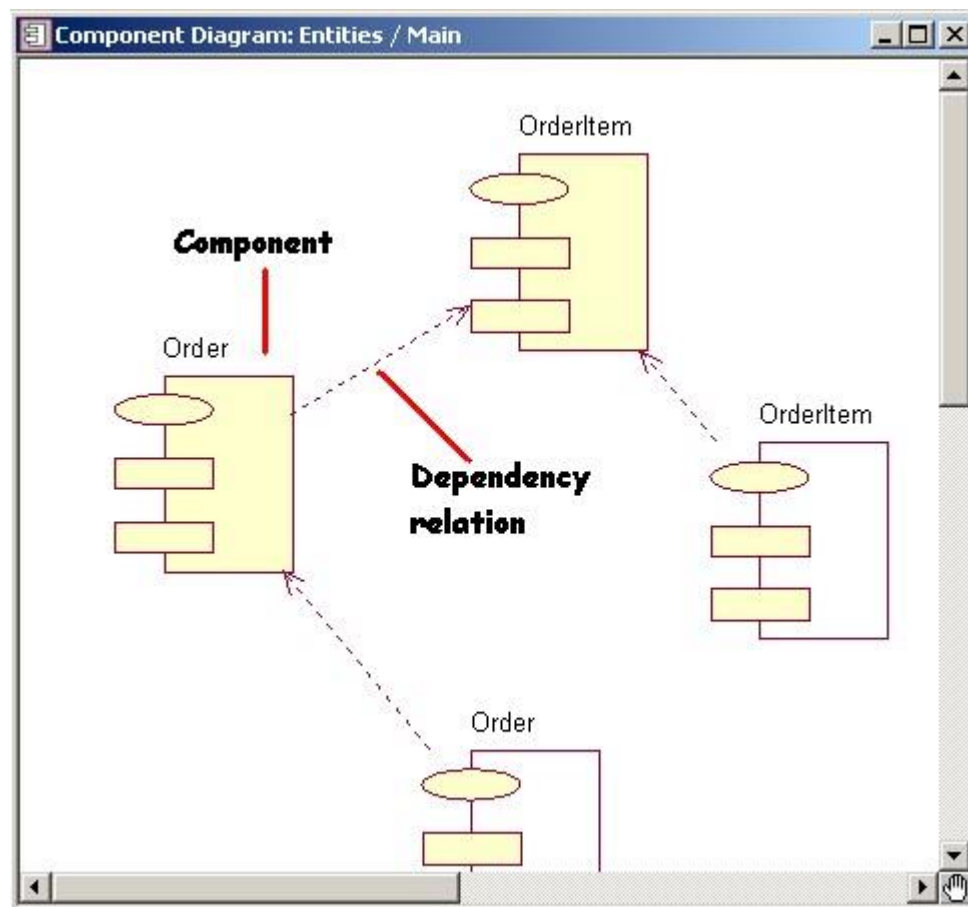
- OnEntry
- OnExit Do
- OnEvent

برای ساخت یک دیاگرام حالت برای یک کلاس کافی است روی کلاس مورد نظر در browser در نمای منطقی کلیک راست کرده و new statechart diagram را انتخاب کنید.

دیاگرام اجزا (component diagram)

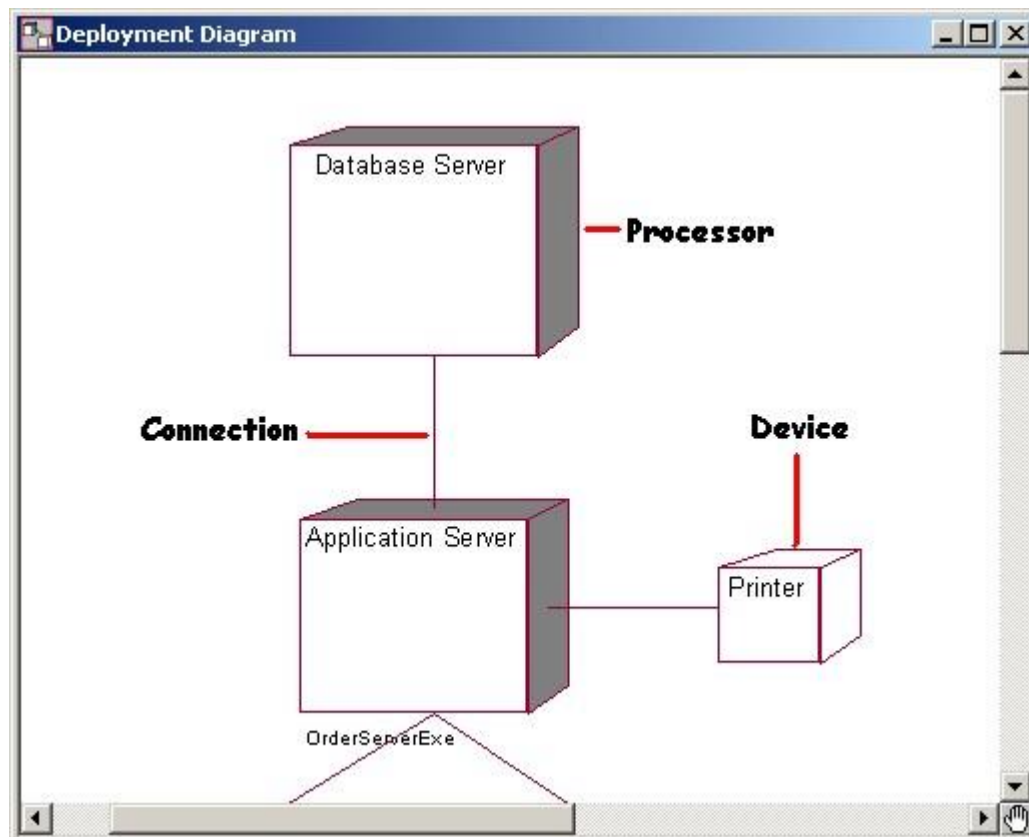
یک نمای فیزیکی از مدل جاری را به ما می دهد. این دیاگرام نحوه ی سازماندهی اجزای نرم افزار و وابستگی بین آنها را به ما میدهد، که شامل سورس کد، کد باینری و اجزای قابل اجرا میباشد. شما می توانید یک یا چند دیاگرام اجزا را برای مجسم ساختن اجزا و بسته ها و یا محتویات هر بسته به کار ببرید.

نکته: هر مدل می تواند شامل چندین component با زبانهای متفاوت باشد، ولی هر کلاس می تواند فقط به component هایی با یک زبان یکسان نسبت داده شود.



دیاگرام پیاده سازی (deployment diagram)

هر مدل شامل فقط یک دیاگرام پیاده سازی است، که نگاشتی از فرآیندهای موجود با سخت افزار سیستم را نشان می دهد.



مراحل کاری انجام یک پروژه در رشنال رز:

۱- شناخت سیستم موجود و آشنایی با پروژه و در صورت نیاز در سیستم های حجیم، مدل کردن سیستم موجود بصورت اجمالی شامل نمودارهای موردهای استفاده، نمودارهای فعالیت جاری و نمودار کلاس ها.

۲- مدل کردن نیازهای عملیاتی سیستم . که این بخش شامل تعیین بسته ها، رسم نمودارهای مورد استفاده و رسم دیاگرامهای فعالیت می باشد.

۳- ساخت مدل تحلیلی سیستم، شامل دیاگرامهای کلاس، توالی، همکاری و حالت.

۴- ساخت مدل پیاده سازی برنامه، شامل دیاگرامهای اجزا، پیاده سازی و بروز در آوری کد و مدل.

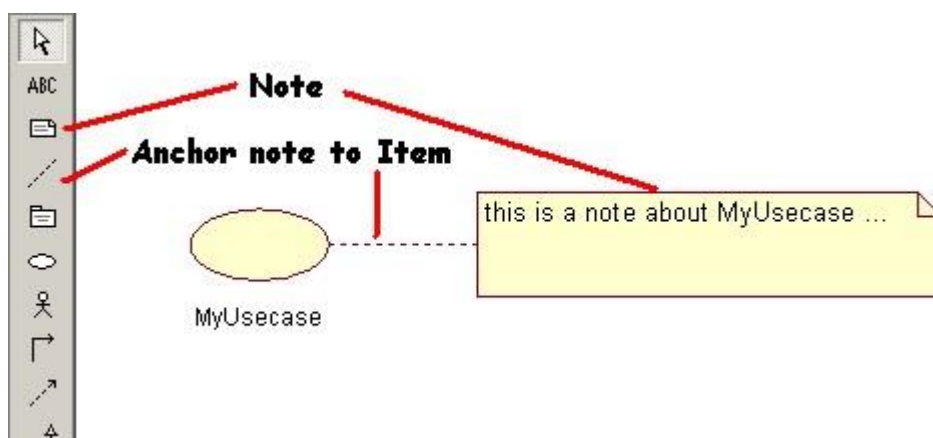
۵- ساخت پایگاه داده های سیستم.

۶- گزارشگیری و منتشر کردن پروژه.

حال پروژه ی خود را تعریف می کنیم و مراحل کاری فوق را تا تولید کد دلفی، در رشنال رز روی آن نشان می دهیم:

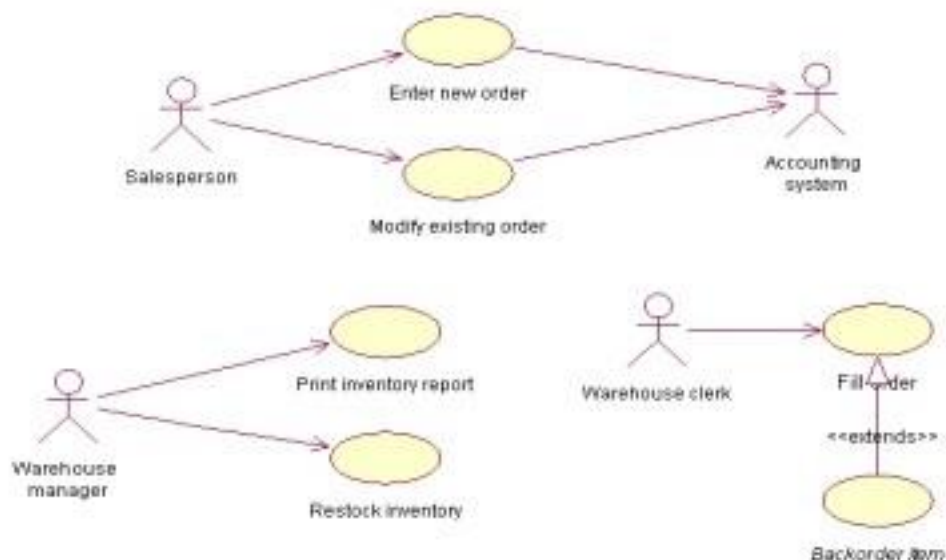
تعریف پروژه: این پروژه یک سیستم سفارش با یک کامپیوتر مرکزی و دو station برای گرفتن سفارشات مشتریان می باشد. سفارشات مشتریان در قسمت انبار توسط کارمندان مربوطه انجام می پذیرد . سیستم باید قادر باشد در هر لحظه گزارش فهرست کالاهای موجود در انبار را بدهد.

قبل از توضیح مراحل پروژه لازم است به این نکته اشاره شود که برای قابل درک و فهم ساختن اجزای نمودارهای مدل، بخصوص در جاهاییکه تعداد آنها زیاد است شما باید به آنها یادداشت هایی را اضافه کنید . برای اینکار در نمودار دیاگرامها در جعبه ابزار دیاگرام، آیکن Note را انتخاب کرده در صفحه ی دیاگرام قرار دهید. سپس با یک خط چین ارتباط این یادداشت را به تکهی مورد نظر برقرار کنید . ولی در این پروژه بعلت کوچک بودن و برای جلوگیری از شلوغ شدن متن و پرداختن به نکات اصلی از آوردن یادداشت ها بصورت note خودداری شده است.



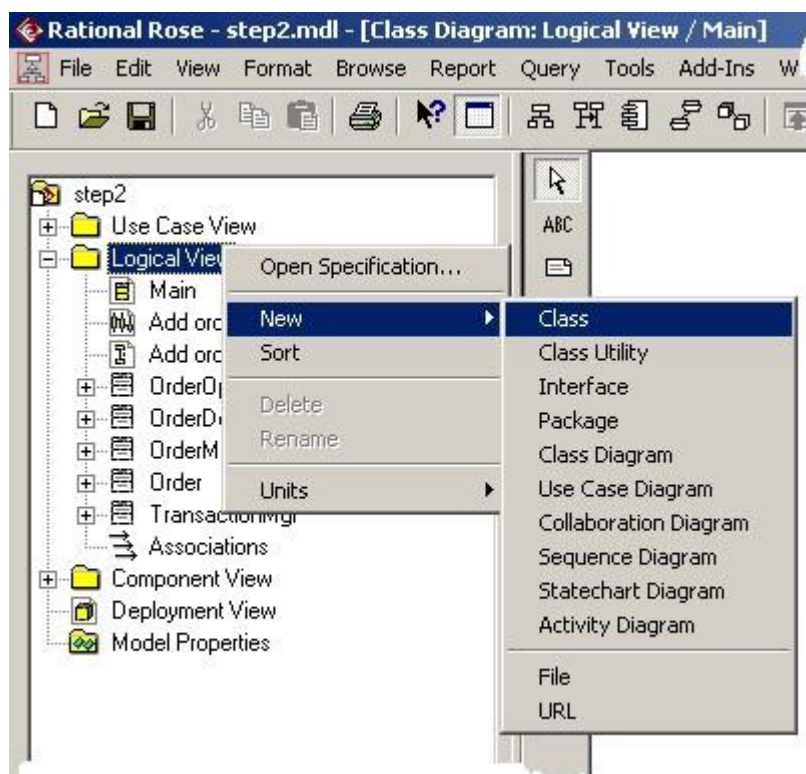
اول پروژه (فایل step1.mdl)

رسم نمودارهای **usecase** ها: همانطوریکه در قسمت قبل توضیح داده شد، در **usecase view** ما نمودار مورد استفاده ی خود را رسم می کنیم. برای پروژه ی ما این نمودار به شکل زیر در می آید:



مرحله ی دوم پروژه (فایل step2.mdl)

ابتدا کلاسهای مورد نظر را با در نظر گرفتن اشیا سیستم، مشخص می کنیم. سپس این کلاسها را به قسمت **logical view** اضافه می کنیم. اضافه کردن یک کلاس به این صورت می باشد که ابتدا روی نود **logical view** راست کلیک می کنیم، سپس از منوی ظاهر شونده، **new > class** را انتخاب می نماییم. سپس مشخصات کلاس مورد نظر را وارد می نماییم.

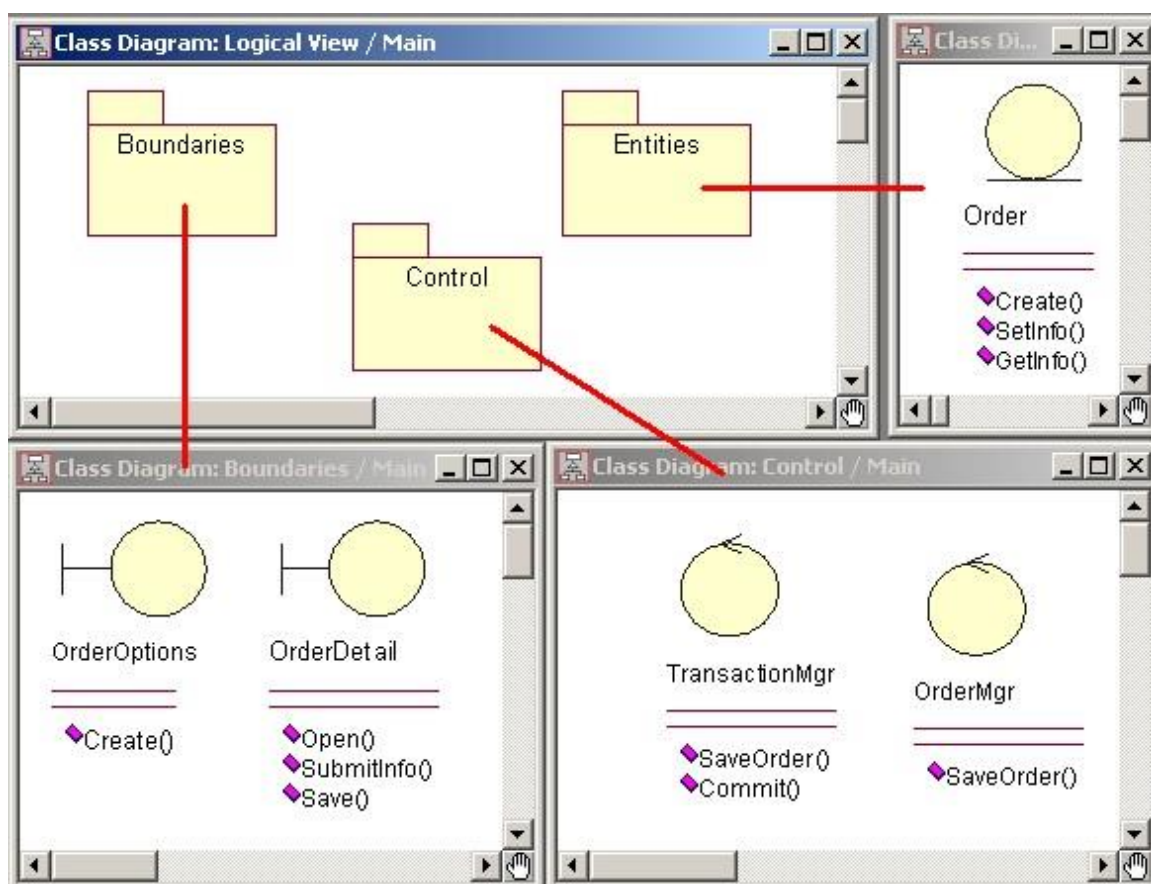


در ادامه ی این مرحله به رسم نمودارهای توالی می پردازیم . این نمودار با استفاده از اشیا موجود (بازیگرهای معرفی شده در دیاگرام موردهای استفاده و نمونه های کلاسهای معرفی شده در همین مرحله) ساخته می شود . کافی است اشیا را در قسمت browser انتخاب کرده و به داخل دیاگرام توالی بیندازیم . حال باید پیغامهای ارتباطی، را بین این اشیا برقرار کرد. پس از رسم این پیغام ها با استفاده از این نمودار متدهای مربوط به هر کلاس را به آن add می کنیم. متد مربوط به هر پیکان در کلاس شی مقصد آن پیکان قرار می گیرد.

برای add کردن یک متد به یک کلاس کافی است، در browser روی آن راست کلیک کرده و new > operation را انتخاب نماییم، سپس در کادر فعال شده مشخصات متد مورد نظر را وارد کنیم.

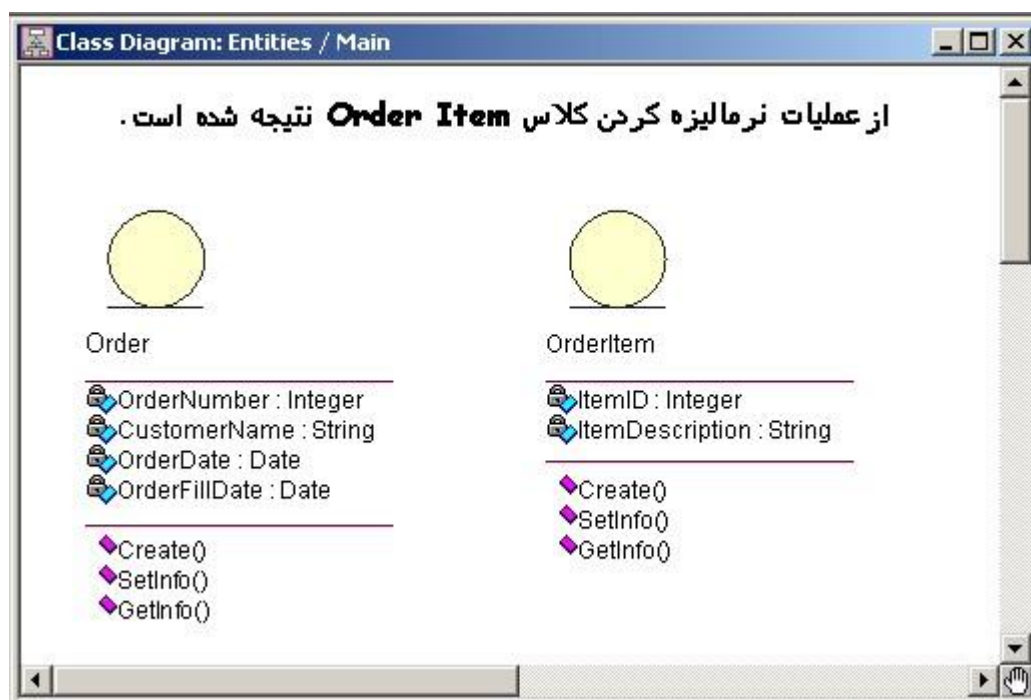
مرحله ی سوم پروژه (فایل step3.mdl)

حال در قسمت main مربوط به نمای منطقی سه بسته بعنوان سه عنصر معماری سه لایه قرار می دهیم و کلاسها را در بسته های مربوطه تقسیم بندی می کنیم . این کار با کشیدن هر کلاس و رها کردن آن روی بسته ی مورد نظر انجام می پذیرد . اکنون stereotype مربوط به هر کلاس را با دوکلیک کردن روی آن کلاس و با توجه به بسته ی قرار گرفته در آن تنظیم می نماییم.



چهارم پروژه (فایل step4.mdl)

حال باید کلاسهای بستهی entity را نرمالیزه نماییم. بعد از نرمالیزه کردن باید در صورت تشکیل کلاسهای جدید، اینکلاسهای تولیدی و کلیه ی attribute های مربوط به کلاسهای entity را وارد نماییم. وارد کردن یک attribute جدید به یک کلاس، با کلیک راست کردن روی آن و انتخاب گزینه ی new > attribute صورت می پذیرد.



مرحله ی پنجم پروژه (فایل step5.mdl)

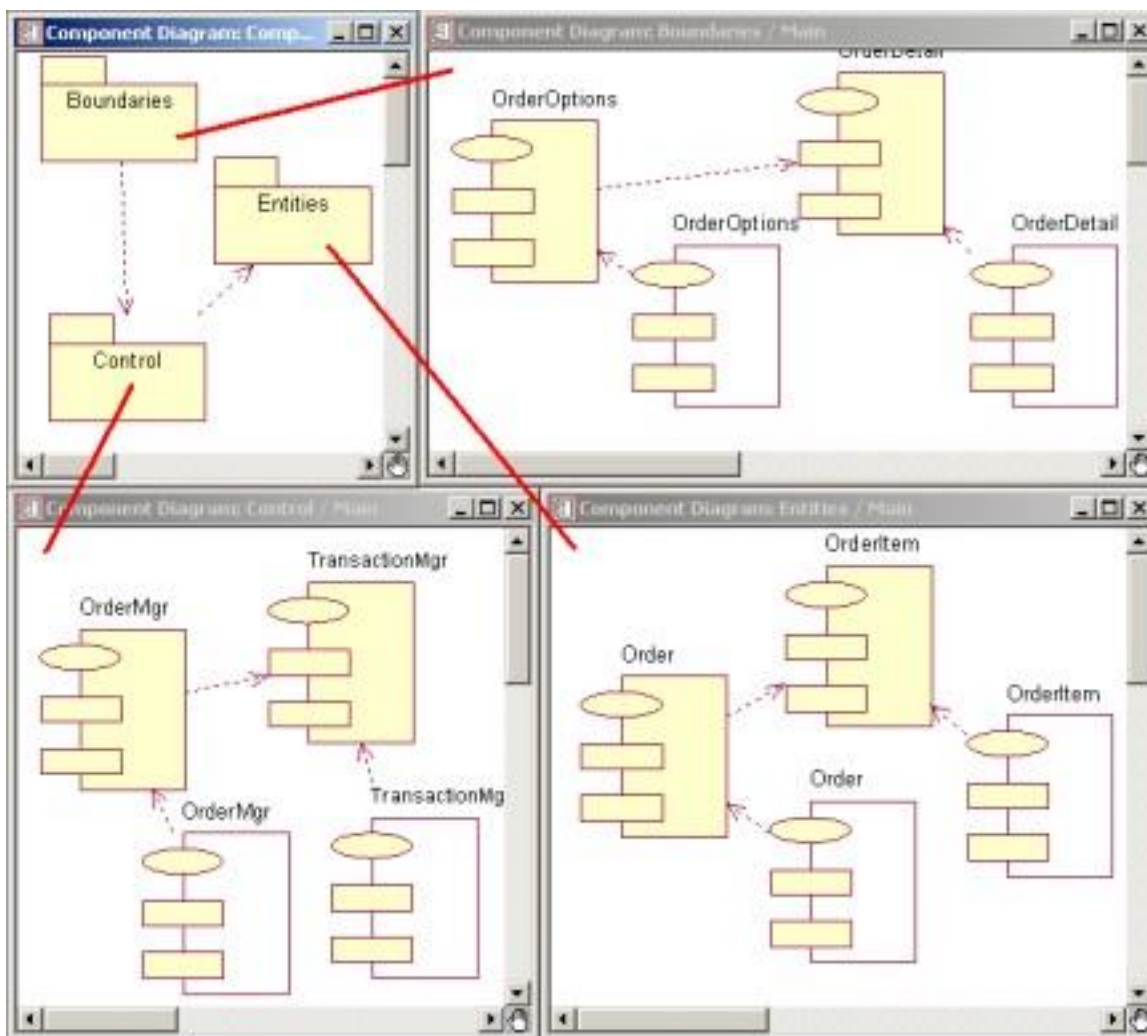
در کنار دیاگرام کلاس اصلی که بصورت سه بسته طراحی شد، ما باید یک ارتباط بین کلیه ی کلاسهای موجود را نشان دهیم، چون ممکن است ارتباطی بین کلاس موجود در یک بسته با کلاس موجود در بسته ی دیگر وجود داشته باشد، لذا ناچاریم دیاگرامهای کلاس دیگری نیز داشته باشیم که کلیه ی کلاسهای مربوط به هم را یکجا جمع آوری نموده و ارتباط بین آنها را مشخص نماید.

برای چگونگی مشخص کردن رابطه های چند گانه بین دو کلاس مشخص، روی فلش ارتباطی بین آنها کلیک کرده و بسته به هدف مورد نظر به یکی از برگه های Role A Detail و یا Role B Detail می رویم، سپس در فیلد multiplicity نوع رابطه را انتخاب می کنیم.

ما در این مرحله بصورت زیر ارتباط منطقی بین کلاسها و رابطه ی آنها را مشخص می کنیم:

هفتم پروژه (فایل step7.mdl)

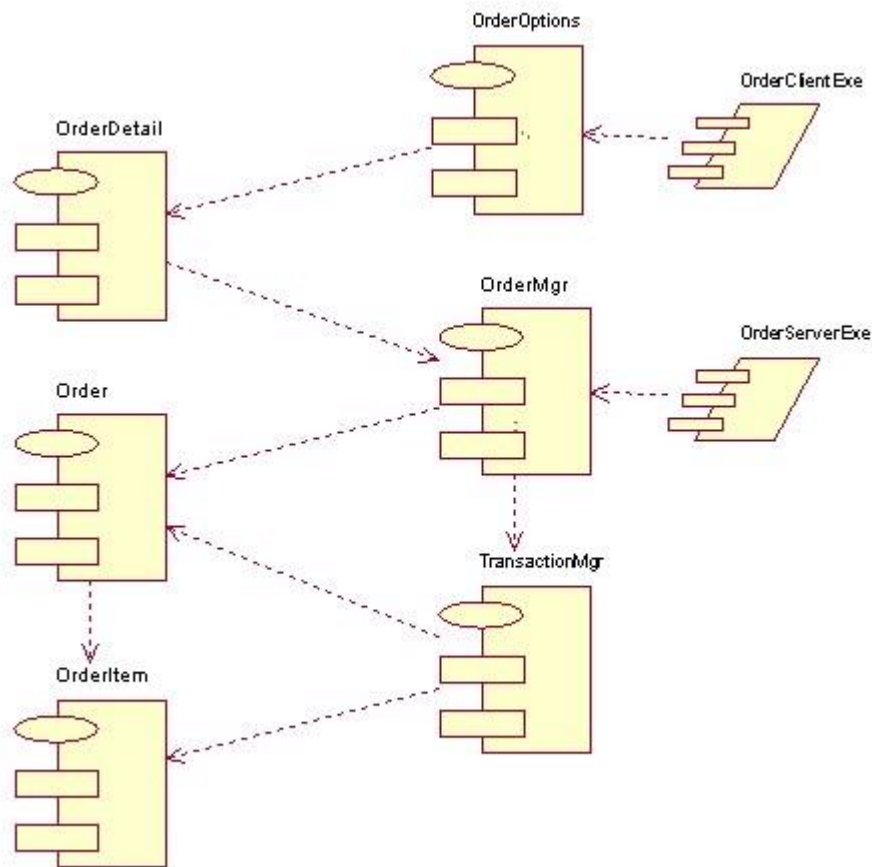
حال نوبت به بخش component view می رسد. ابتدا در دیاگرا اصلی این نما سب بسته ی اصلی معماری سه لایه را قرارداده و ارتباط منطقی بین آنها را برقرار میکنیم.



سپس یک نمودار component diagram جدید می سازیم (وجود این نمودار به این دلیل است، چون در main ما سه بسته ی جدا داریم، در نتیجه نمی توانیم ارتباط قسمت های مختلف بین دو بسته را برقرار کنیم)، حال به ازای هر کلاس که در دیاگرام اصلی کلاسهایمان داریم یک جز Package specification قرار می دهیم و ارتباط بین آنها را برقرار می کنیم.

چون برنامه ی ما یک برنامه ی Client/Server می باشد، دو جز Task specification در نمودارمان قرار می دهیم، که یکی مربوط به برنامه ی اجرایی روی Client می باشد و دیگری مربوط به Server است.

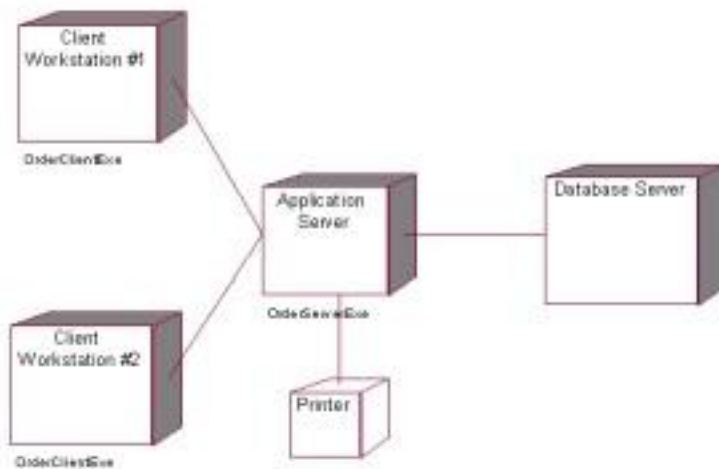
شکل مربوط به این دیاگرام در صفحه ی بعد آمده است :



دیگرام ارتباطی اجزا

مرحله ی هشتم پروژه (فایل step8.mdl)

با توجه به تعریف پروژه به سادگی می توان نمودار پیاده سازی زیر را برای سیستم در نظر گرفت . تنها نکته ای که نباید فراموش گردد، نسبت دادن پروسه های اجرایی روی کامپیوترهای client و server می باشد . برای اینک ار در browser روی نود مربوط به پردازشگر مورد نظر راست کلیک کرده و `new > process` را انتخاب نمایید و پروسه ی مورد نظر را به آن نسبت دهید.

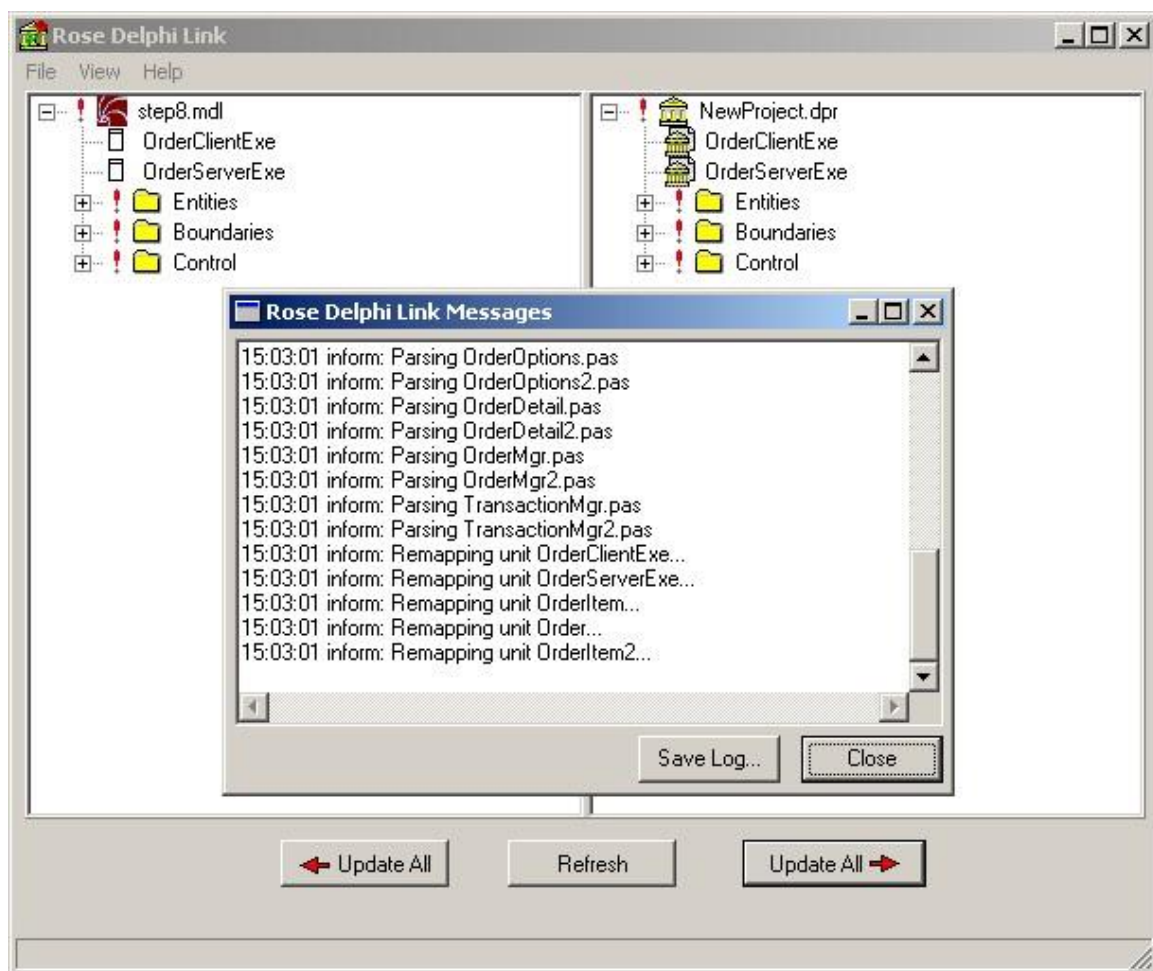


چگونگی تولید کد دلفی از مدل و یا عکس عمل فوق:

ابتدا روی component های مورد نظر دو کلیک نمایید. سپس در specification مربوط به آن در قسمت language زبان دلفی را انتخاب نمایید.

حال از منوی tools گزینه ی Ensemble Tools > Rose Delphi Link ... را انتخاب نمایید. اکنون برای بروز کردن مدل با توجه به کد موجود، پروژه ی مورد نظر را باز کرده و با توجه به جهت نشان داده شده، عملیات مورد نظر را انجام دهید.

اگر هدف شما تولید کد می باشد و پروژه را قبلا نساخته اید، ابتدا می بایستی از منوی File گزینه ی new project را انتخاب نمایید، ولی بجای اینکار، پیشنهاد می کنیم که ابتدا یک پروژه ی خالی را توسط دلفی ساخته و بجای انتخاب این گزینه، پروژه ی ساخته شده را باز نمایید.



نحوه ی تولید پایگاه داده ی مربوط به پروژه :

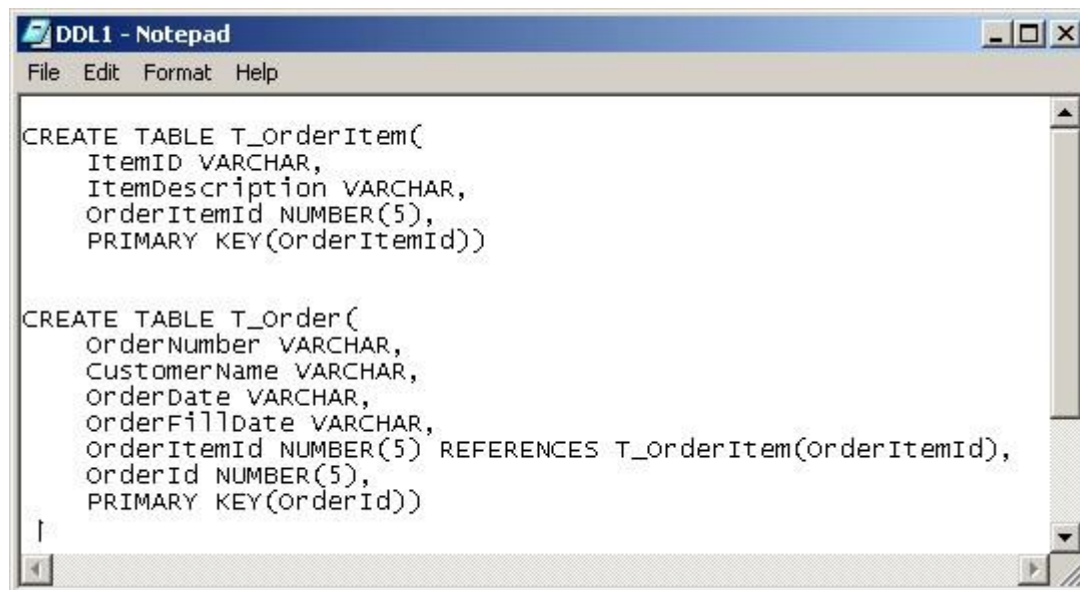
می دانیم که بانک اطلاعات پروژه از کلاسهای entity بدست می آیند.. برای تولی تولید کد SQL مربوطه، مطابق با هر کدام از بانک های موجود نظیر access ، SQL Server ، oracle یا ... ، ابتدا باید کلاسهای entity خود را به صورت

persistant در آوریم . این کلاسها، همانند دیگر کلاسها بصورت پیش فرض transient هستند . برای این تغییر کافی است روی هر کلاس دوکلیک کرده و به در فرم specification به برگه ی Detail بروید و تغییرات لازم را اعمال کنید.

بعد از این تبدیل کلاسهایی را که می خواهید برای آنها کد SQL تولید گردد، را انتخاب نمایید. سپس در منوی tools گزینه ی DDL > Generate code را انتخاب نمایید. کد مورد نظر در فایل انتخابی شما ذخیره می گردد.

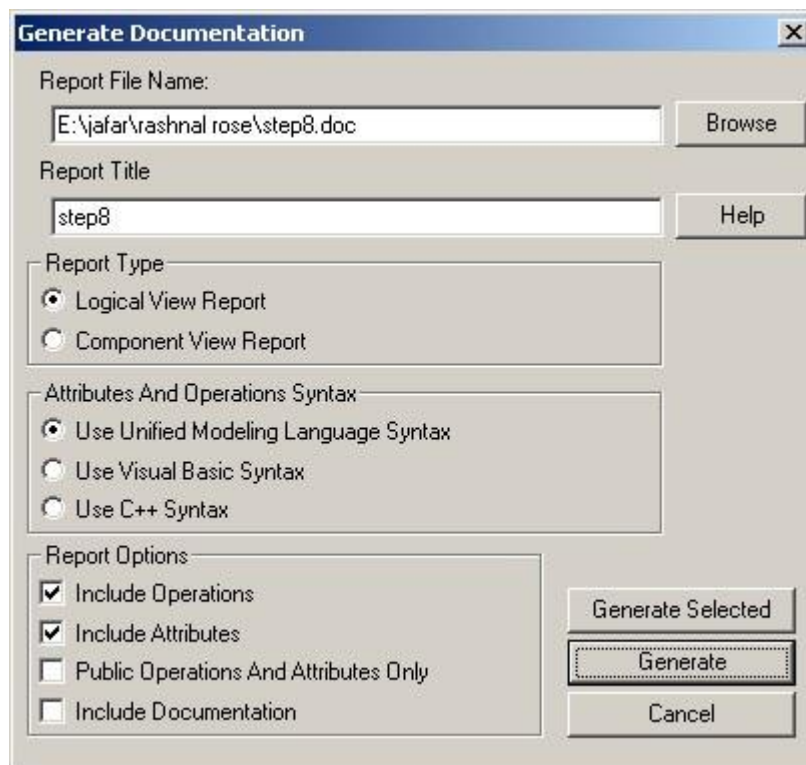


برای مشاهده ی کد تولید شده می توانید از منوی tools گزینه ی DDL > Browse DDL را انتخاب نمایید.



```
CREATE TABLE T_OrderItem(  
    ItemID VARCHAR,  
    ItemDescription VARCHAR,  
    OrderItemId NUMBER(5),  
    PRIMARY KEY(OrderItemId))  
  
CREATE TABLE T_Order(  
    OrderNumber VARCHAR,  
    CustomerName VARCHAR,  
    OrderDate VARCHAR,  
    OrderFillDate VARCHAR,  
    OrderItemId NUMBER(5) REFERENCES T_OrderItem(OrderItemId),  
    OrderId NUMBER(5),  
    PRIMARY KEY(OrderId))
```

نحوه ی گزارش گرفتن از پروژه:
برای تولید مستندات مدل‌تان (نماهای منطقی و اجزا) و مشاهده ی آنچه که رشنال رز برای شما تولید کرده است، از منوی
reeport گزینه ی Documentation report ... را بکار ببرید. گزارشهای تولیدی برای پروژه ی ما در انتهای نوشتار
ضمیمه می باشد.



Generate Documentation

Report File Name:
E:\jafar\rashnal rose\step8.doc Browse

Report Title
step8 Help

Report Type
☒ Logical View Report
☐ Component View Report

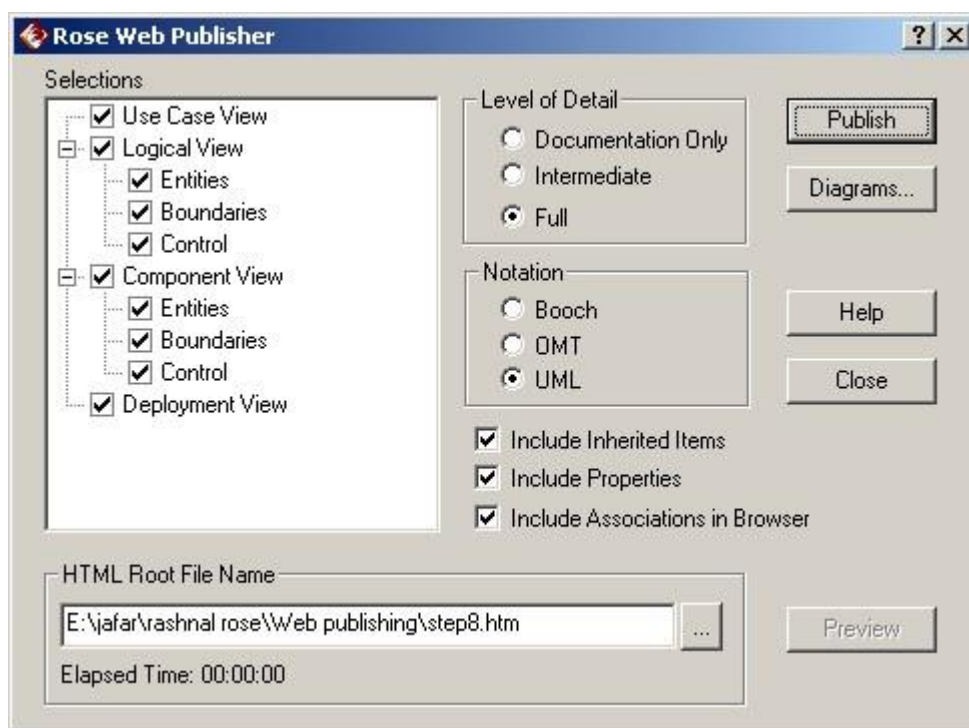
Attributes And Operations Syntax
☒ Use Unified Modeling Language Syntax
☐ Use Visual Basic Syntax
☐ Use C++ Syntax

Report Options
☒ Include Operations
☒ Include Attributes
☐ Public Operations And Attributes Only
☐ Include Documentation

Generate Selected
Generate
Cancel

نحوه ي انتشار پروژه:

در منوی tools گزینه ی Web publisher... عملیات انتشار را برای پروژه ی شما انجام می دهد . این عمل یک نسخه ی html از تمامی مدل شما همراه با مستندات تهیه می کند، تا سایر کاربران با دیگر browser های موجود نیز بتوانند مدل شما را ببینند. فرم مربوطه در صفحه ی بعدی باشد و فایل html تولیدی نیز در دیسکت ضمیمه موجود می باشد.



فرم تولید فایل html مربوط به مدل