رشنال رز را بصورت عملی بیاموزیم

مقدمه :

در این نوشتار می‌خواهیم 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 جداگانه‌ی مشخص شده در دیسکت ضمیمه می‌باشد، تا مراحل کاری به خوبی تفهیم گردند.

حتما از خود می‌پرسید چرا مدلسازی؟

تا چند سال پیش همه‌ی رشته‌ها برای خود زبان مشترک داشتند، مثلا ریاضیدانان از زبان ریاضی و مهندسین الکترونیک از زبان مداری استفاده می‌کردند ولی مهندسین نرم افزار زبان واحدی نداشتند ، تا اینکه UML پا به عرصه‌ی وجود گذاشت و باعث شد که مهندسین نرم افزار بتوانند مفاهیم مورد نیاز خود را براحتی با همدیگر در میان بگذارند و کارهایشان را مدل کنند.

یک مدل خوب مثل نقشه‌ی ساختمان :

- نیازمندی‌ها را مشخص می‌کند.
- ارتباطات بین قسمت های مختلف پروژه را به ما می‌نمایاند.
- بدون وارد شدن به جزییات می‌توانیم در نحوه‌ی فعل و انفعالات قسمت های مختلف پروژه تمرکز نماییم.
- در یک تیم کاری، بعلت وجود یک زبان گرافیکی مشترک، ارتباط بین افراد تیم بهبود می‌یابد.

در اولین گام از این نوشتار به معرفی عناصر عمده‌ی رشنال رز می‌پردازیم. این قسمت شامل بخش‌های زیر می‌باشد :

- معرفی محیط رشنال رز
- معرفی نمادهای رشنال رز
- معرفی دیاگرام‌های UML که در رشنال رز بکار گرفته می‌شوند.

معرفی محیط رشنال رز

عناصر اصلی رشنال رز عبارتند از:

۱- **Standard toolbar** : که برای تمام دیاگرام‌ها مشترک است و در قسمت بالای پنجره واقع است.

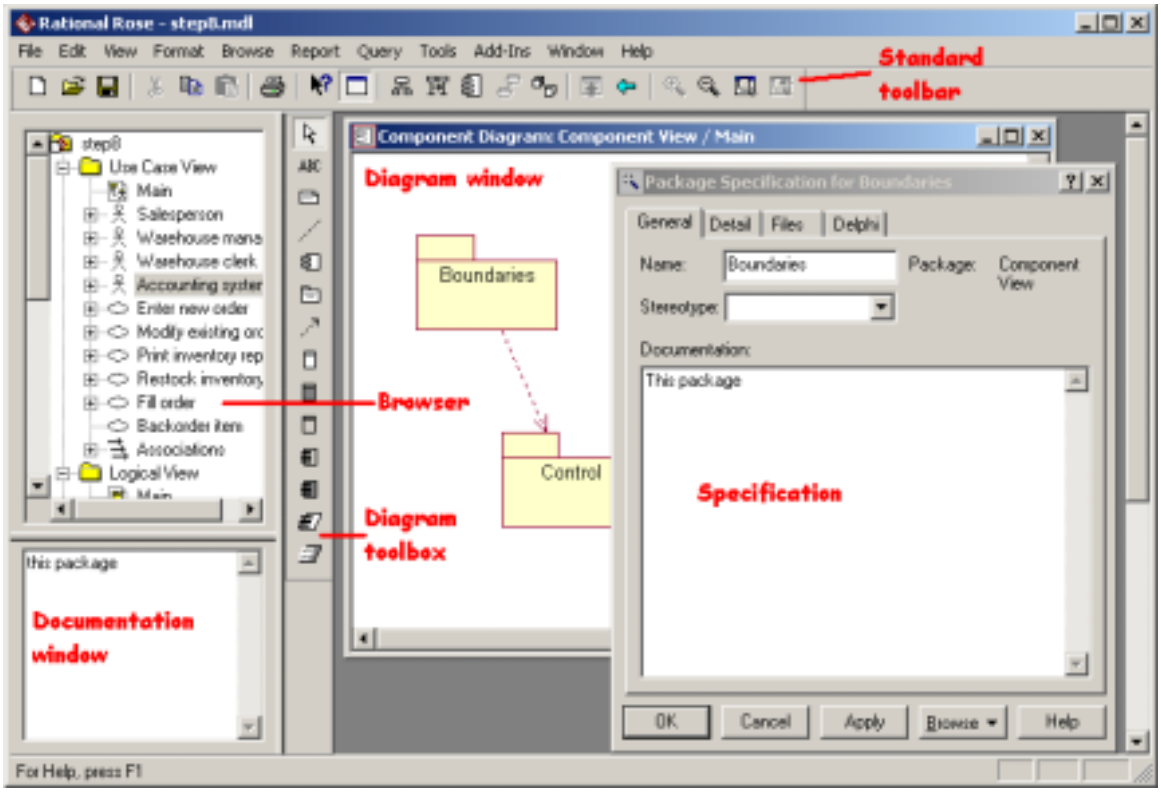
۲- **Diagram toolbar** : که وابسته به پنجره‌ی دیاگرام فعال است و در سمت چپ پنجره‌ی دیاگرام واقع است.

۳- **Browser** : به شما اجازه می‌دهد تا بصورت یک ساختار درختی دیاگرام‌های موجود و عناصر مدلهایتان را مشاهده کنیپ.

۴- **Diagram window** : ساخت و ویرایش دیاگرام‌ها در این قسمت صورت می‌پذیرد.

۵- **Documentation window** : به شما اجازه می‌دهد تا به مدلهایتان مستندات لازم را نیز اضافه نمایید. می‌توانید 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

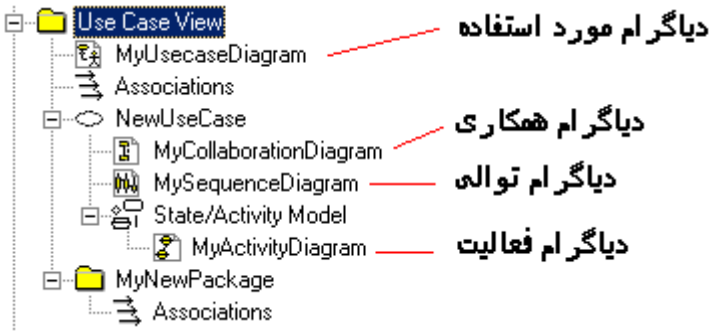
برای یک پروژه‌ی در حال ساخت نماهای مختلفی وجود دارد . رشنال رز نماهای زیر را برای یک پروژه فراهم می آورد که هر کدام یکی از جنبه‌های مختلف مدل را نمایش می‌دهند:

- Usecase view –
- Logical view –
- Component view –
- Deployment view –

نمای مورد استفاده‌ی سیستم (usecase view) :

این نما جهت فهم واستفاده از سیستم پیش‌بینی شده است. در این نما که تشریح رفتار سیستم از دیدگاه کاربر است، فعل وانفعالات متقابل بازیگرها (actors) و موردهای استفاده نمایش داده می‌شود. در این نما چهار دیاگرام زیر وجود دارند :

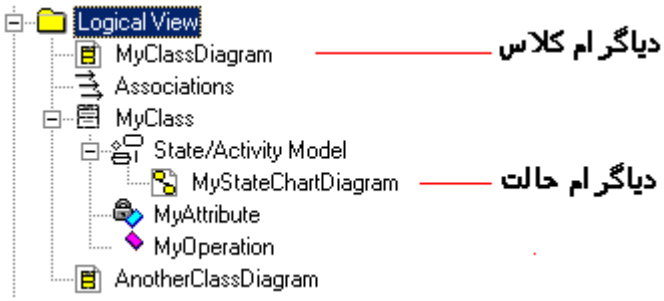
- دیاگرامهای موردهای استفاده (usecase diagrams) –
- دیاگرامهای توالی (sequence diagrams) –
- دیاگرامهای همکاری (collaboration diagrams) –
- دیاگرامهای فعالیت (activity diagrams) –



نمای منطقی سیستم (logical view) :

این نما شامل نیازمندی‌های عملیاتی سیستم می‌باشد که به کلاسها وارتباط بین آنها می‌پردازد. این نما شامل دو دیاگرام زیر می‌باشد:

- دیاگرامهای کلاسها (class diagrams) –
- دیاگرامهای حالت (statechart diagrams) –



نمای اجزای سیستم (component view) :

این نما به سازمان سیستم می‌پردازد و اطلاعاتی در باره‌ی نرم افزار، اجزا قابل اجرا و کتابخانه‌های سیستم دارد.تنها دیاگرام موجود در این قسمت دیاگرام اجزا (component diagram) می‌باشد.

نمای پیاده‌سازی سیستم (deployment view) :

این قسمت شامل نگاشتی از فرایندهای موجود با سخت افزار سیستم می‌باشد. این قسمت بخصوص در جاهایی حایز اهمیت است که شما دارای برنامه‌ها و سرورهایی در قسمت‌های مختلف مکانی هستید و می‌خواهید به بهترین شیوه ساختار توزیعی محیط را نشان دهید.

این نما فقط شامل یک دیاگرام (deployment diagram) می‌باشد.

معرفی دیاگرامهای رشنال رز

یک دیاگرام یک نمایش گرافیکی از عناصر سیستمتان می‌باشد. دیاگرامهای گوناگون به شما اجازه می‌دهند تا پرسپکتیوهای مختلف از سیستم‌تان را ببینید.

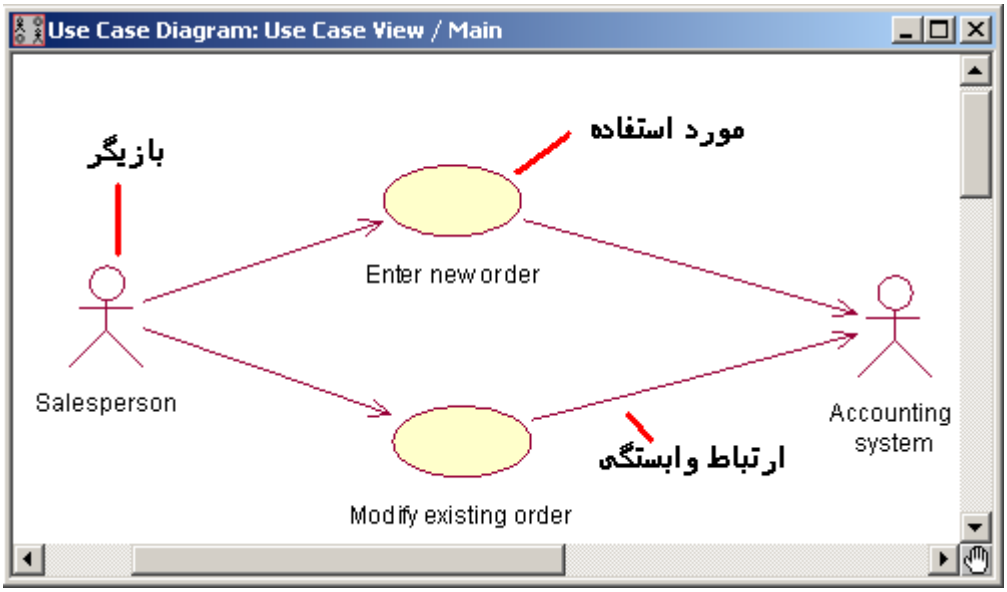
شما در رشنال رز قادر به ایجاد دیاگرامهای زیر می‌باشید:

- 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 پیاده‌سازی (deployment diagram)

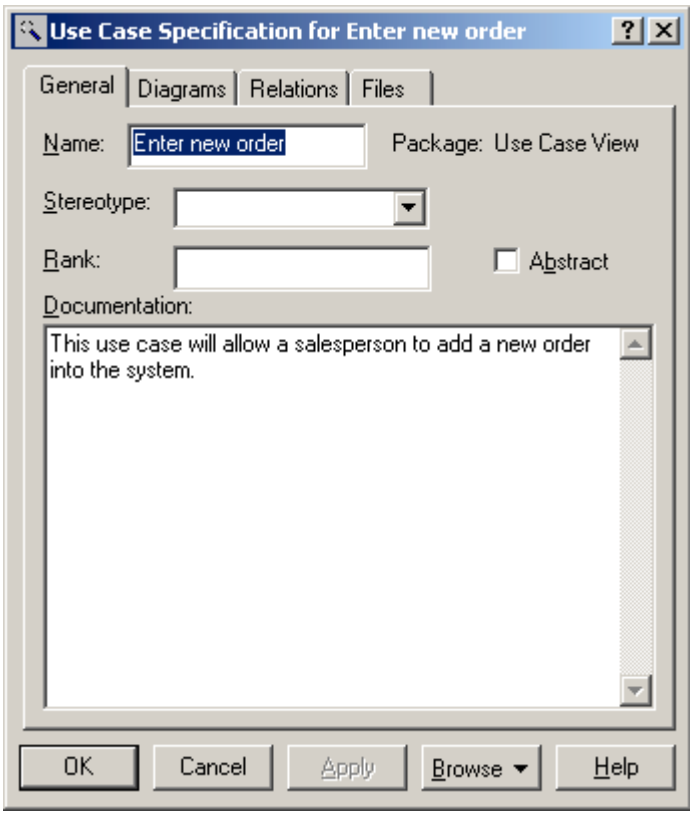
دیاگرام موردهای استفاده (usecase diagram)

یک usecase رفتار سیستم را توصیف می‌کند، که شامل تقابل بین سیستم و بازیگران می‌باشد. بعبارت کلی یک usecase یک الگوی رفتاری توانایی‌های سیستم و یک دنباله تراکنش‌های وابسته به هم می‌باشد، که بوسیله‌ی سیستم و بازیگران انجام می‌پذیرد و دیاگرام موردهای استفاده یک نمای سطح بالای سیستم از دید بازیگران سیستم به ما می‌دهد و نحوه‌ی برخورد آن با دنیای بیرون را مشخص می‌کنند. این دیاگرام در طول تحلیل سیستم برای بدست آوردن نیازمندی‌ها و نشان دادن چگونگی کارکرد سیستم بکار می‌رود.

نکته: در یک مدل واحد ممکن است چندین دیاگرام مورد استفاده داشته باشیم. معمولا در این صورت هر مجموعه‌ای از usecase های مرتبط به هم را در یک بسته قرار می‌دهیم. به این ترتیب کلیه‌ی usecase های سیستم در بسته‌های مرتبط به هم قرار می‌گیرند.



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.



اضافه کردن یک بازیگر هم مثل یک مورد استفاده می‌باشد.برای ساختن یک وابستگی جدید بین دو عنصر نیز روی آیکون association کلیک کردهو 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 ومقصد 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.

اصولا استفاده‌ی مجدد از عناصر موجود در سایر دیاگرامها نیز به همین شکل می‌باشد.

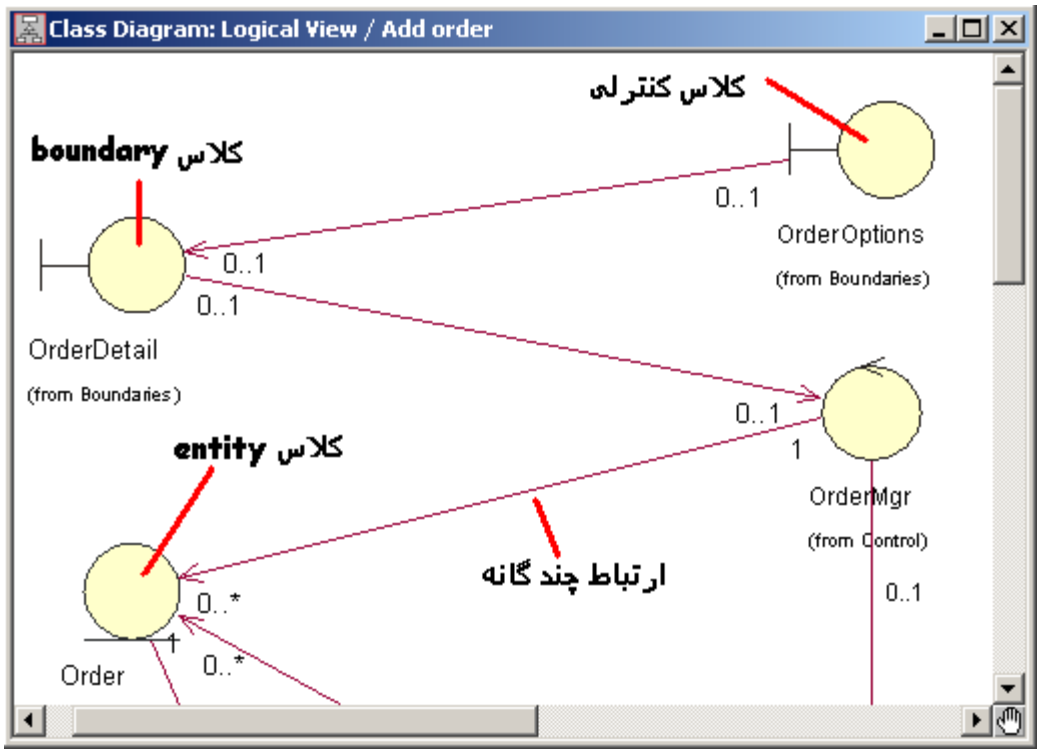
دیاگرام کلاس‌ها (class diagram)

این دیاگرام به شما کمک می‌کند تا نمای ساختاری سیستم‌تان را بصورت بصری (visual) در آورید. این دیاگرام از معمول‌ترین دیاگرامهای UML می‌باشد.این دیاگرامها جزییات هر کلاس و ارتباطات بین آنها را نشان می‌دهد و پایه و اساس دیاگرامهای اجزا و پیاده‌سازی می‌باشد.

در یک مدل واحد ممکن است چندین دیاگرام کلاس داشته باشیم.

در یک دیاگرام کلاس با کلاسهای 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 می‌توانند پنجره ها، سنسورها، ترمینالها یا ... باشند.مثلا نجره‌ی 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 ورفتارهای تعبیه شده در یک usecase را کنترل می‌کنند.

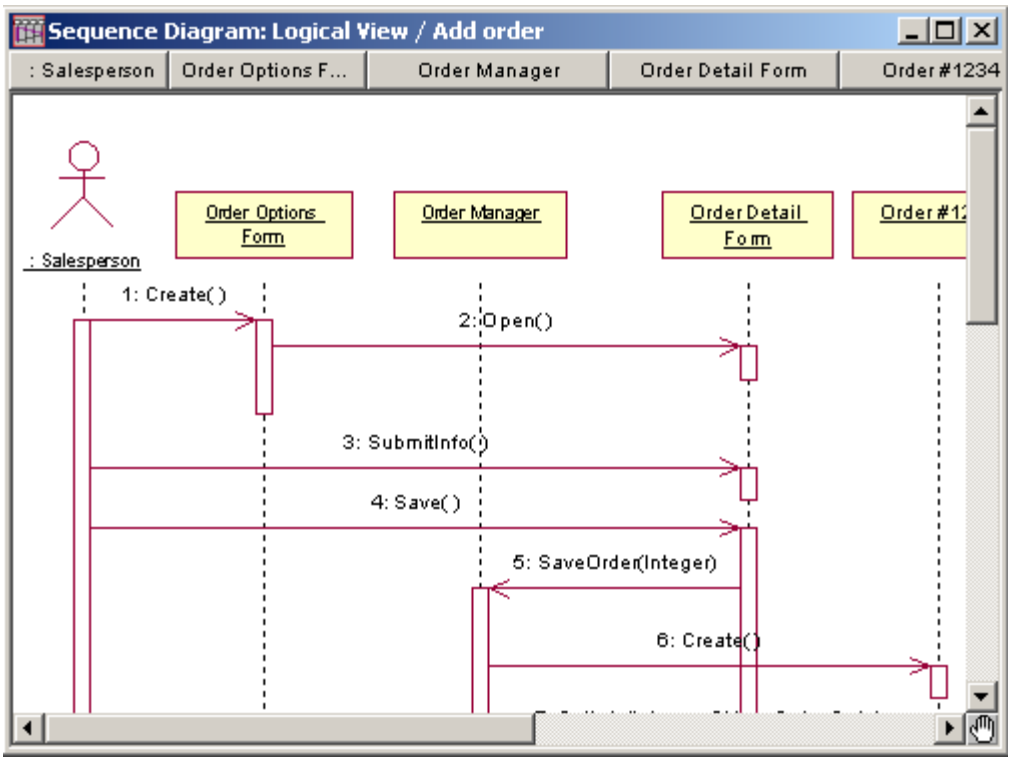
کلاسهای entity : این کلاسها اطلاعاتی را که باید توسط سیستم ذخیره گردند را در خود نگهداری می‌کنند. محل نگهداری ساختمان داده‌های منطقی سیستم، این کلاسها می‌باشند.

دیاگرام توالی (sequence diagram)

دیاگرام توالی بصورت منظم و در یک توالی زمانی پشت سر هم ارتباطات متقابل اشیا را به ما نشان می‌دهد.این دیاگرام برای انجام عمل خاصی در یک usecase مشخص، مراحل انجام کار را مرحله به مرحله به شما نشان می‌دهد، یعنی دنباله‌ای از رویدادها را برای انجام یک عمل مشخص می‌سازد.

در مراحل تحلیل وطراحی برای فهم نحوه‌ی عملکرد سیستم از این دیاگرام استفاده می‌شود.

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



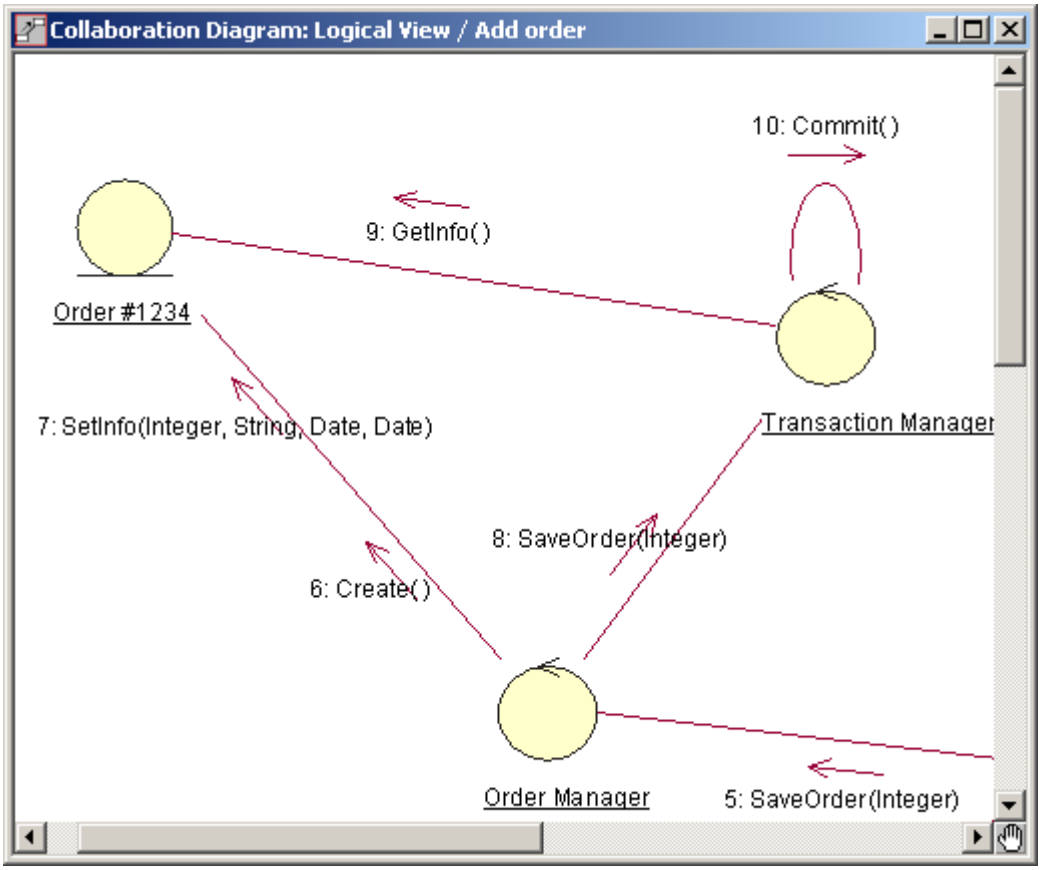
برای اضافه کردن کلاسها ویا بازیگرهای موجود به یک sequence diagram عناصر موجود در قسمت‌های قبلی را در browser کشیده و به داخل این دیاگرام بیندازید.

اگر می‌خواهید هنگام اضافه کردن پیغامها، عددهایی نیز به آنها اضافه گردد (همانند شکل فوق) تا توالی کار را بدرستی نشان دهد در منوی tools روی options کلیک کرده و به برگه‌ی دیاگرام رفته و جعبه‌ی انتخاب sequence numbering را علامت بزنید و سپس ok کنید. برای اضافه کردن پیغامها نیز می‌توانید از جعبه ابزار دیاگرام، آیکون object message استفاده کنید.

دیاگرام همکاری (collaboration diagram)

یک نما از ارتباطلت ساختاری بین اشیا در مدل فعلی را بما می‌دهد. تاکید این دیاگرام بر ارتباط بین اشیا است در حالیکه تاکید دیاگرامهای توالی بر روی دنباله‌ای از رویدادها بود. این نوع دیاگرام شامل اشیا، لینک‌ها، وپیغام‌ها می‌باشد.
برای درک وفهم چگونگی رفتار سیستم واتخاذ 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 می‌دهد. کافی است از منوی browse گزینه‌ی 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 امکان‌پذیر است.



دیاگرام فعالیت (activity diagram)

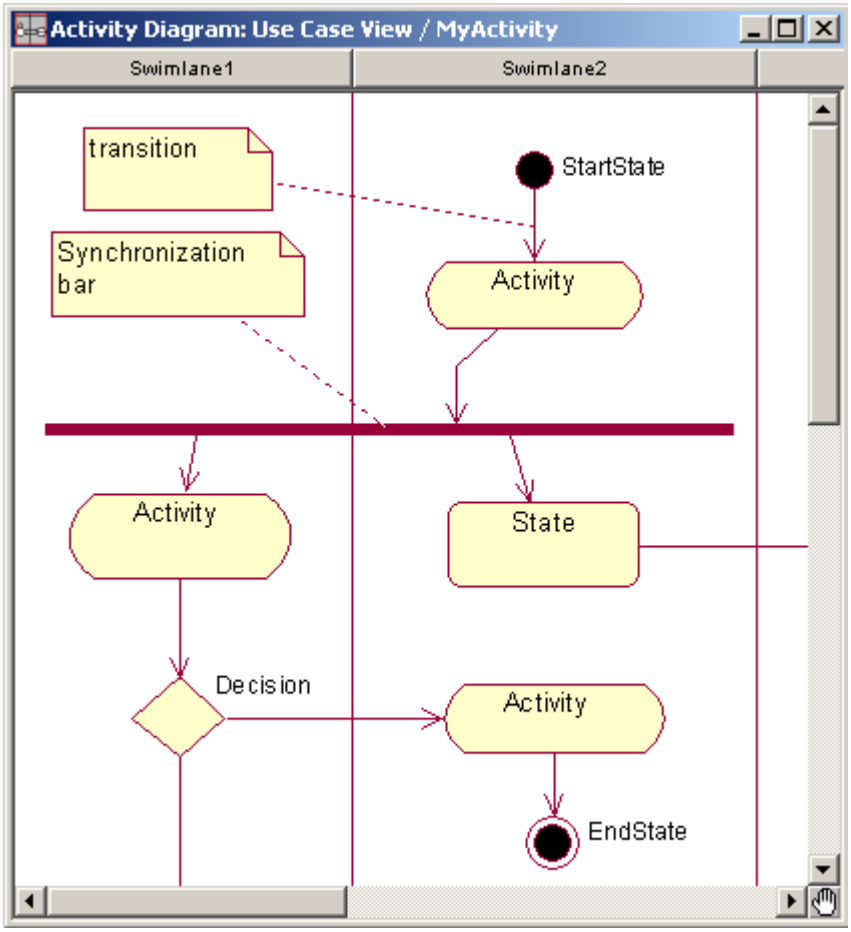
دیاگرام فعالیت، جریان کار و همچنین توالی فعالیت‌ها را در یک فرآیند مشخص می‌کند.این دیاگرام خیلی شبیه فلوچارت است، زیرا شما می‌توانید جریان کار را از یک فعالیت به فعالیت دیگر یا به حالتی دیگر، دنبال نمایید.

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

یک دیاگرام فعالیت می‌تواند عناصر زیر را داشت باشد:

- یک **start state** و یک **end state**
- **activity ها**، که مراحل را در جریان کاری نشان می‌دهند.
- **Transition ها**، که ترتیب کاری را نشان می‌دهند.

- **Decision ها**، که به شما اجازه‌ی تصمیم گیری‌ها را در دیاگرام می‌دهند.
- **Synchronization bar ها** : که اجازه‌ی نمایش کارهای موازی را در دیاگرام به شما می‌دهد.
- **Swimlane ها** : نقش‌های 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 می‌شود، کافی است روی آن دو کلیک نمایید تا دیاگرام مربوط به آن باز شود.

برای ساخت swimlaneها روی آیکون مربوط به آن در جعبه ابزار دیاگرام کلیک کرده و سپس روی نمودار کلیک نمایید. Swimlane جدید ایجاد مشود. برای تنظیم فیلدهای آن کافی است روی آن دو کلیک کنید، فرم مربوط به specification آن باز می‌شود.

سایر قسمت‌ها نیز روی جعبه ابزار دیاگرام مشخص است و اضافه کردن آنها همانند قسمت‌های قبل می‌باشد.

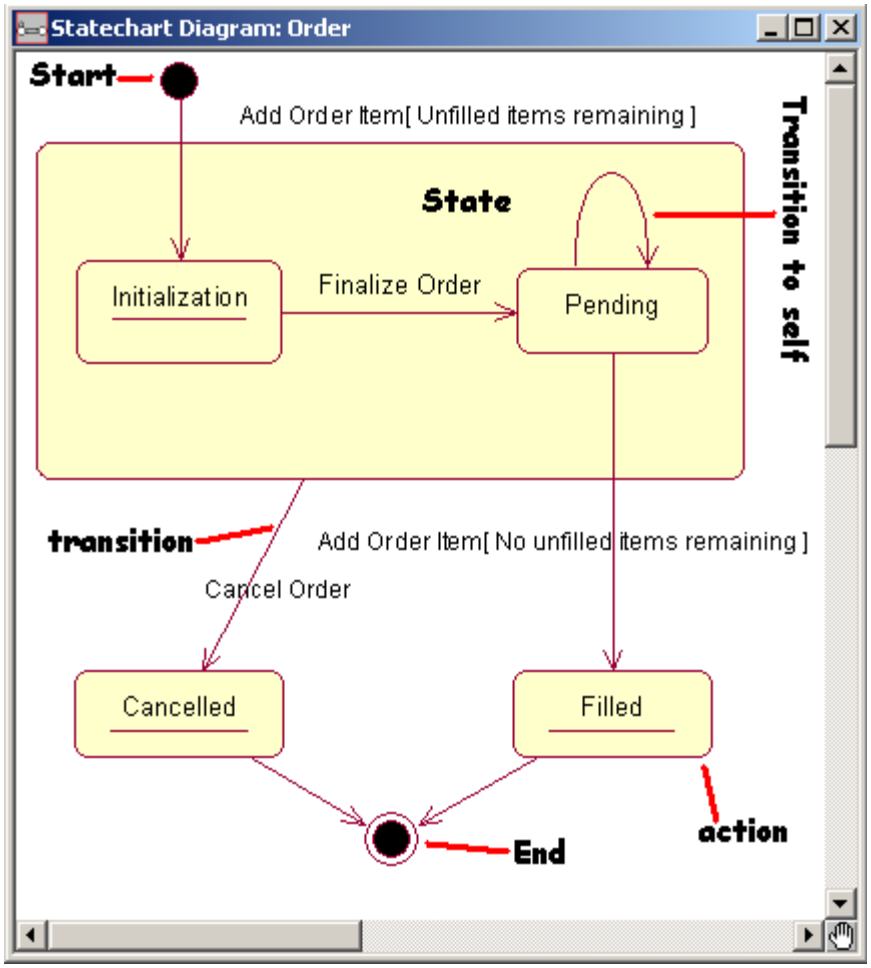
دیاگرام حالت (statechart diagram)

می‌توانید دیاگرام‌های حالت را برای مدل کردن رفتار پویای کلاس‌ها یا اشیا انفرادی استفاده کنید.این دیاگرامها ترتیب حالاتی که یک شی می‌تواند داشته باشد، رویدادهایی را که موجب انتقال از یک حالت یا فعالیت به دیگری می‌شوند و نتایجی را که این انتقال بوجود می‌آورد، را نمایش می‌دهد.

یک دیاگرام حالت، معمولا برای مدل کردن مراحل گسسته‌ی چرخه‌ی حیات یک شی، بکار برده می‌شود ، در حالیکه دیاگرام فعالیت بر توالی فعالیت‌ها یی در یک فرآیند، دلالت دارد.

عناصر اصلی یک دیاگرام حالت عبارتند از:

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



نکته: هر حالت در یک دیاگرام حالت می‌تواند شامل چندین عمل داخلی باشد.هر عمل، وظیفه‌ای است که در یک حالت رخ می‌دهد، که می‌تواند یکی از اعمال زیر باشد:

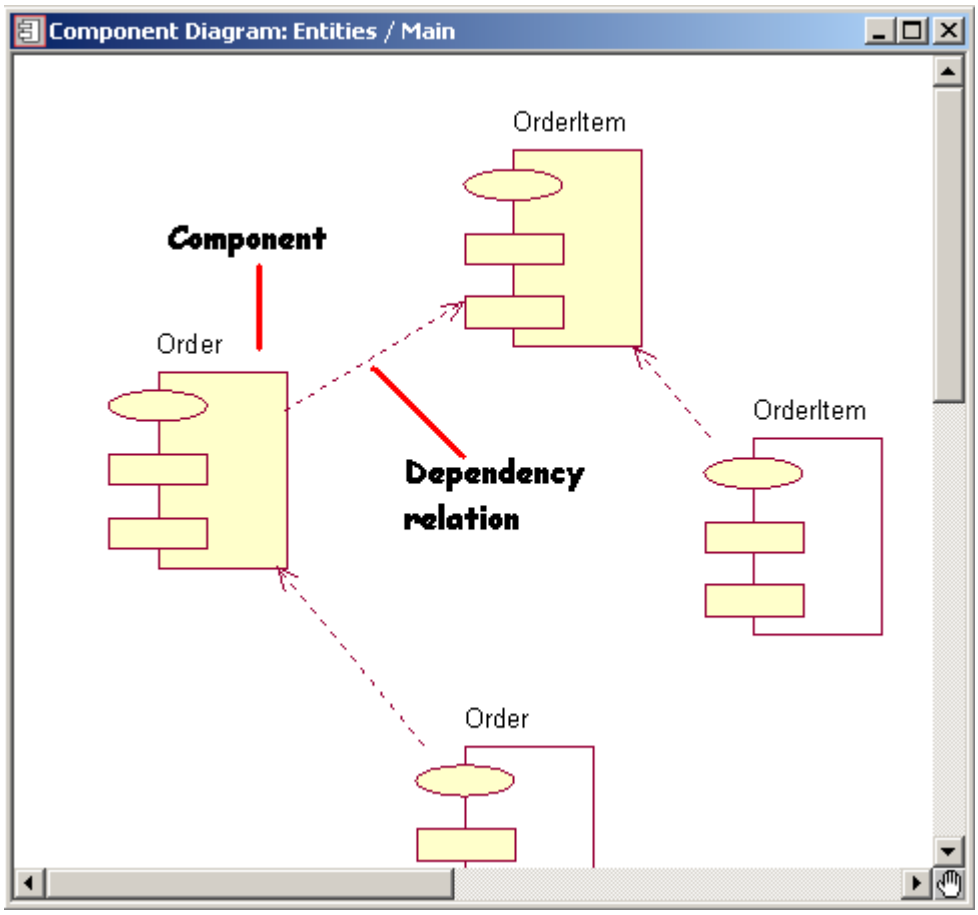
- OnEntry
- OnExit
- Do
- OnEvent

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

دیاگرام اجزا (component diagram)

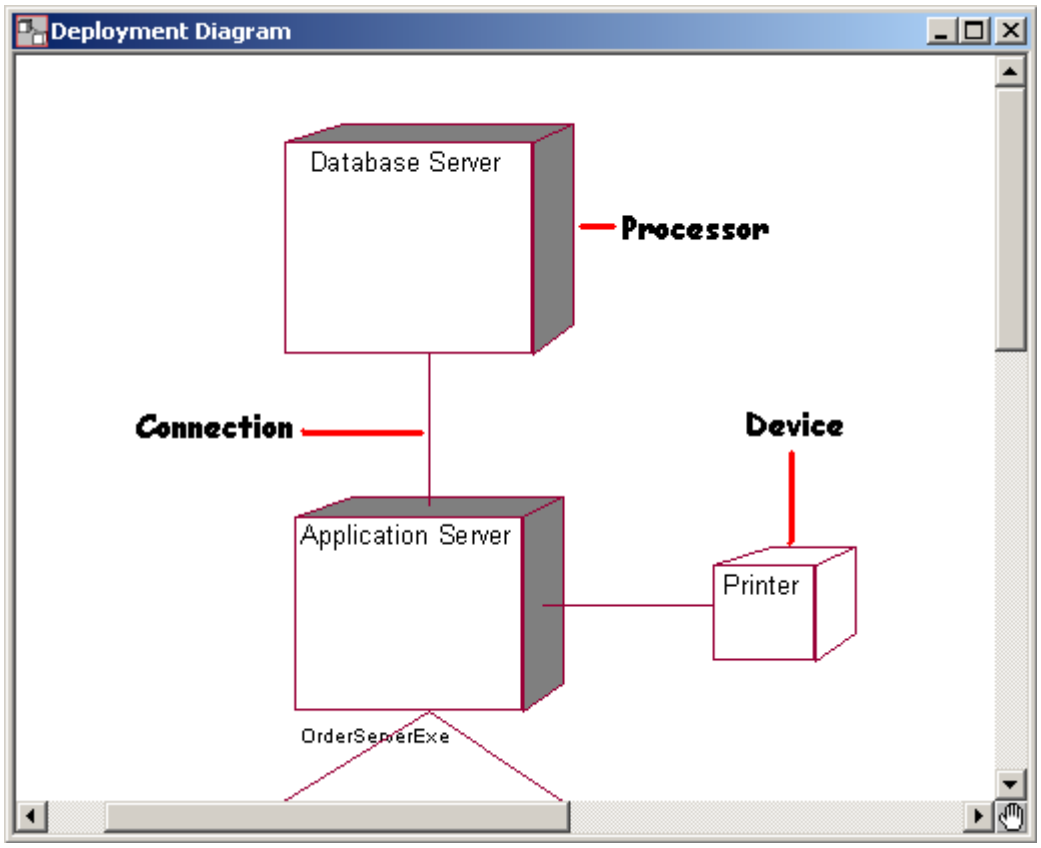
یک نمای فیزیکی از مدل جاری را به ما می‌دهد. این دیاگرام نحوه‌ی سازماندهی اجزای نرم‌افزار و وابستگی بین آنها را به ما می‌دهد، که شامل سورس کد، کد باینری و اجزای قابل اجرا می‌باشد.
شما می‌توانید یک یا چند دیاگرام اجزا را برای مجسم ساختن اجزا وبسته‌ها و یا محتویات هر بسته به کار ببرید.

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



دیاگرام پیاده‌سازی (deployment diagram)

هر مدل شامل فقط یک دیاگرام پیاده‌سازی است، که نگاشتی از فرآیندهای موجود با سخت‌افزار سیستم را نشان می‌دهد.



مراحل کاری انجام یک پروژه در رشنال رز:

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

۲- مدل کردن نیازهای عملیاتی سیستم. که این بخش شامل تعیین بسته‌ها، رسم نمودارهای مورد استفاده ورسم دیاگرامهای فعالیت می‌باشد.

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

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

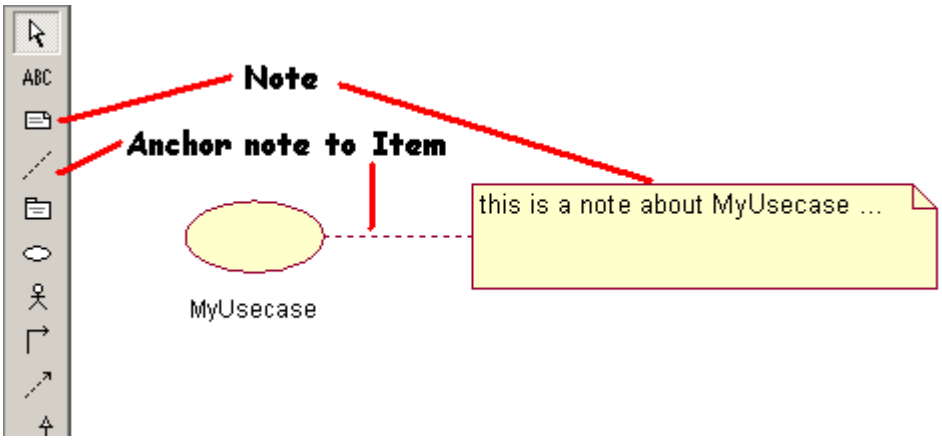
۵- ساخت پایگاه داده‌های سیستم.

۶- گزارشگیری و منتشر کردن پروژه.

حال پروژه‌ی خود را تعریف می‌کنیم و مراحل کاری فوق را تا تولید کد دلفی، در رشنال رز روی آن نشان می‌دهیم:

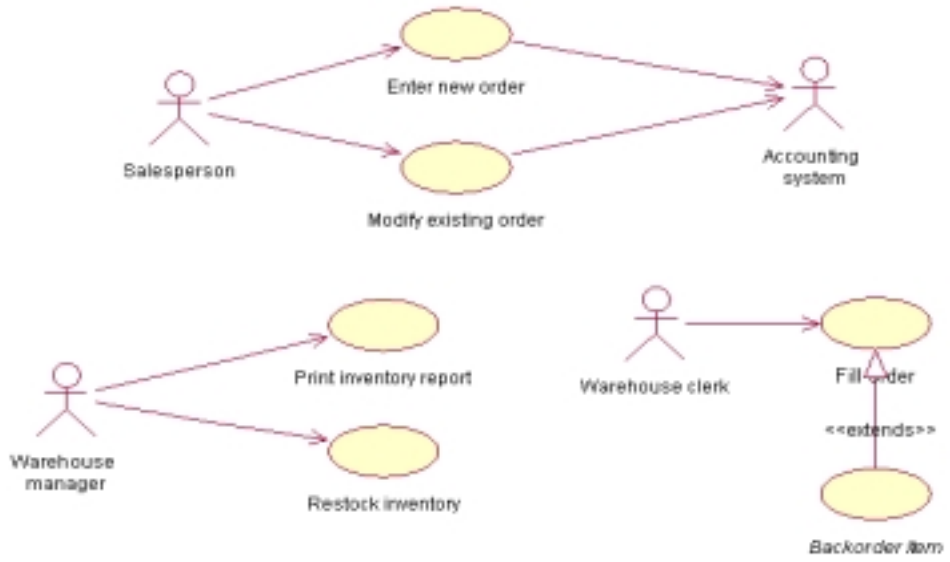
تعریف پروژه: این پروژه یک سیستم سفارش با یک کامپیوتر مرکزی و دو station برای گرفتن سفارشات مشتریان می‌باشد.سفارشات مشتریان در قسمت انبار توسط کارمندان مربوطه انجام می‌پذیرد. سیستم باید قادر باشد در هر لحظه گزارش فهرست کالاهای موجود در انبار را بدهد.

قبل از توضیح مراحل پروژه لازم است به این نکته اشاره شود که برای قابل درک وفهم ساختن اجزای نمودارهای مدل، بخصوص در جاهاییکه تعداد آنها زیاد است شما باید به آنها یادداشت‌هایی را اضافه کنید. برای اینکار در نمودار دیاگرامها در جعبه ابزار دیاگرام، آیکون Note را انتخاب کرده در صفحه‌ی 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 یادداشت‌ها بصورت note خودداری شده است.



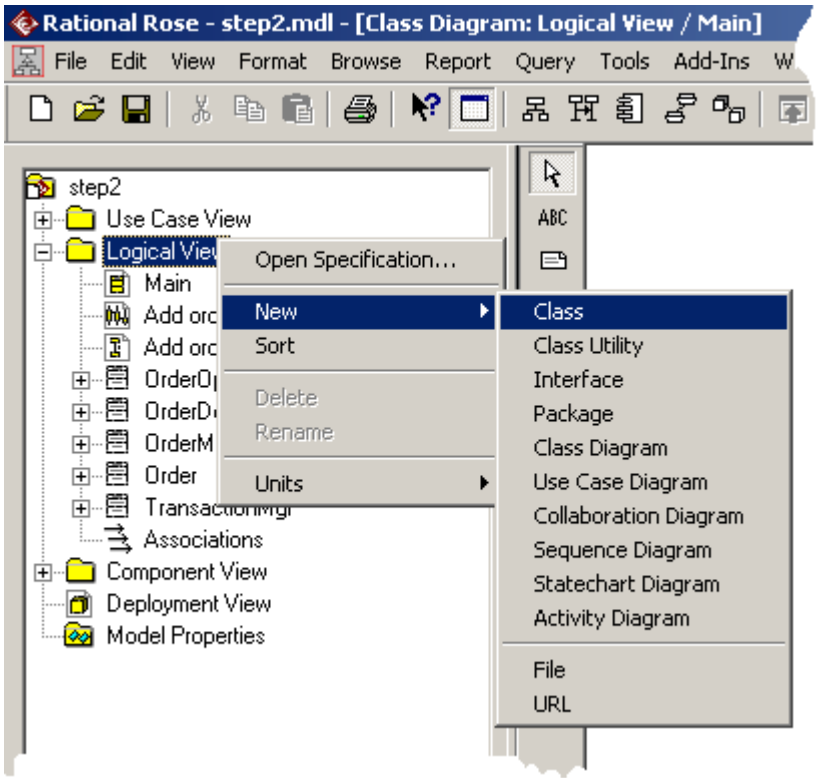
مرحله‌ی اول پروژه (فایل step1.mdl)

رسم نمودارهای usecase ها: همانطوریکه در قسمت قبل توضیح داده شد، در usecase view ما نمودار مورد استفاده‌ی خود را رسم می‌کنیم. برای پروژه‌ی ما این نمودار به شکل زیر در می‌آید:



مرحله‌ی دوم پروژه (فایل step2.mdl)

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

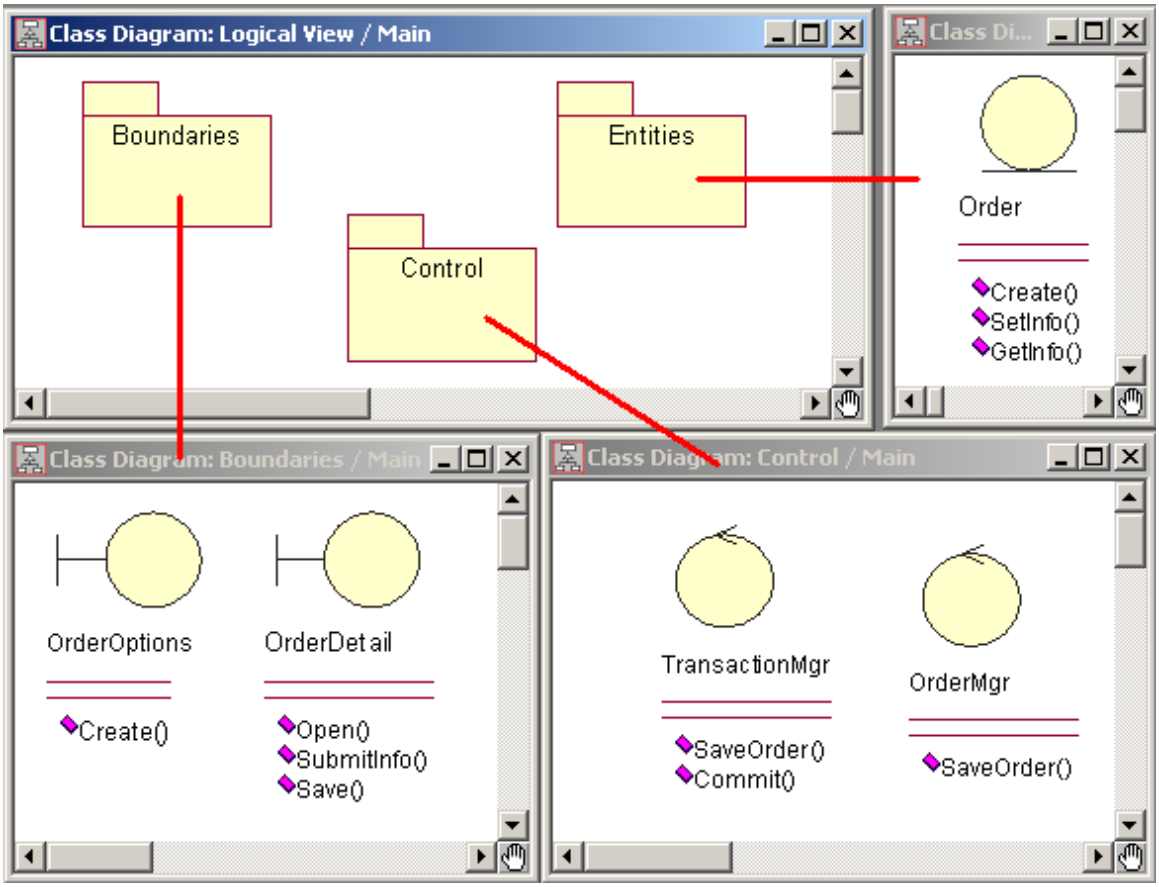


در ادامه‌ی این مرحله به رسم نمودارهای توالی می‌پردازیم. این نمودار با استفاده از اشیا موجود (بازیگرهای معرفی شده در دیاگرام موردهای استفاده و نمونه‌های کلاسهای معرفی‌شده در همین مرحله) ساخته می‌شود. کافی است اشیا را در قسمت browser انتخاب کرده وبه داخل دیاگرام توالی بیندازیم. حال باید پیغامهای ارتباطی، را بین این اشیا برقرار کرد. پس از رسم این پیغام‌ها با استفاده از این نمودار متدهای مربوط به هر کلاس را به آن add می‌کنیم. متد مربوط به هر پیکان در کلاس شی مقصد آن پیکان قرار می‌گیرد.

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.

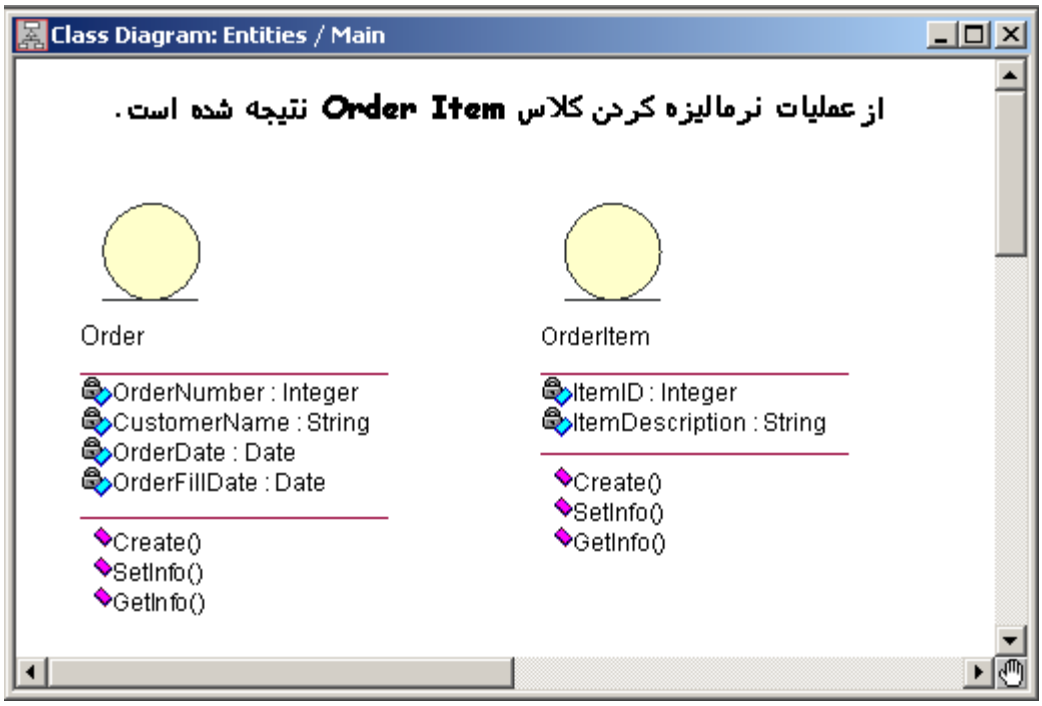
مرحله‌ی سوم پروژه (فایل step3.mdl)

حال در قسمت main مربوط به نمای منطقی سه بسته بعنوان سه عنصر معماری سه لایه قرار می‌دهیم و کلاسها را در بسته‌های مربوطه تقسیم بندی می‌کنیم.این کار با کشیدن هر کلاس و رها کردن آن روی بسته‌ی مورد نظر انجام می‌پذیرد. اکنون strereotype مربوط به هر کلاس را با دوکلیک کردن روی آن کلاس و با توجه به بسته‌ی قرار گرفته در آن تنظیم می‌نماییم.



مرحله‌ی چهارم پروژه (فایل step4.mdl)

حال باید کلاسهای بسته‌ی entity را نرمالیزه نماییم. بعد از نرمالیزه کردن باید در صورت تشکیل کلاسهای جدید، این کلاسهای تولیدی و کلیه‌ی attributeهای مربوط به کلاسهای entity را وارد نماییم. وارد کردن یک attribute جدید به یک کلاس، با کلیک راست کردن روی آن و انتخاب گزینه‌ی new > attribute صورت می‌پذیرد.

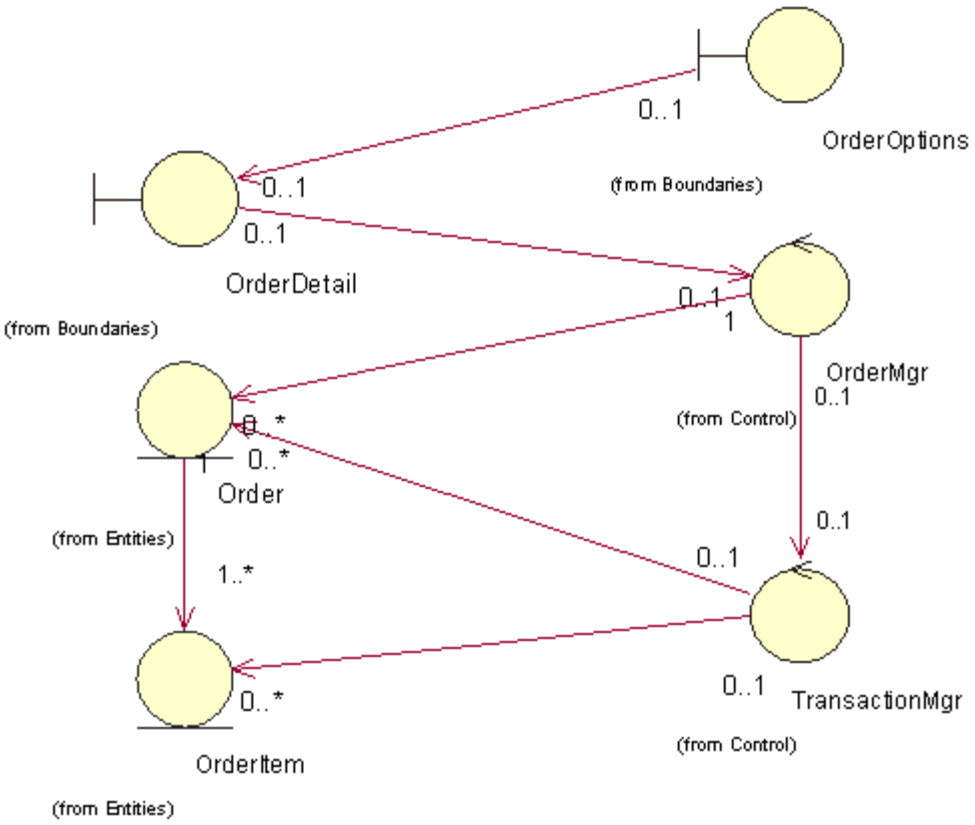


مرحله‌ی پنجم پروژه (فایل step5.mdl)

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

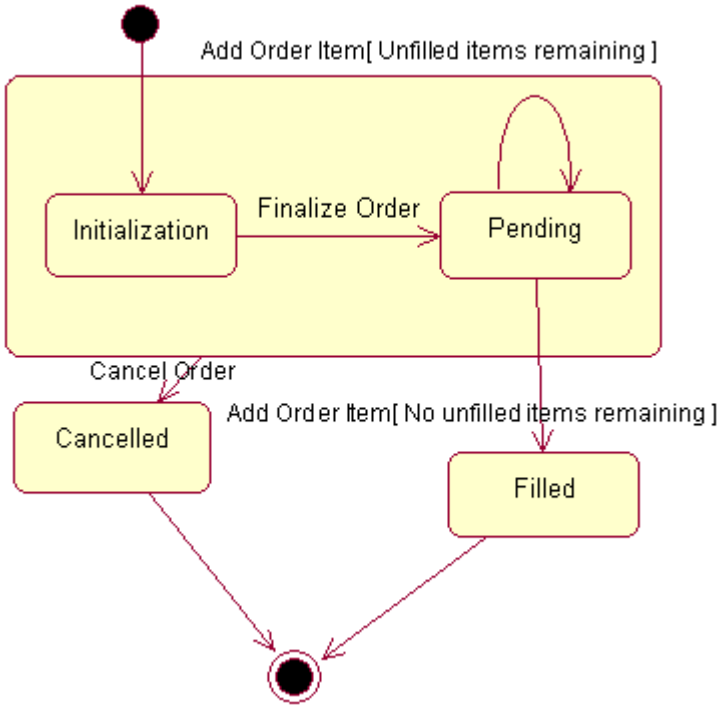
برای چگونگی مشخص کردن رابطه‌های چند گانه بین دو کلاس مشخص، روی فلش ارتباطی بین آنها کلیک کرده و بسته به هدف مورد نظر به یکی از برگه‌های Role A Detail ویا Role B Detail می‌رویم، سپس در فیلد multiplicity نوع رابطه را انتخاب می‌کنیم.

ما در این مرحله بصورت زیر ارتباط منطقی بین کلاسها و رابطه‌ی آنها را مشخص می‌کنیم:



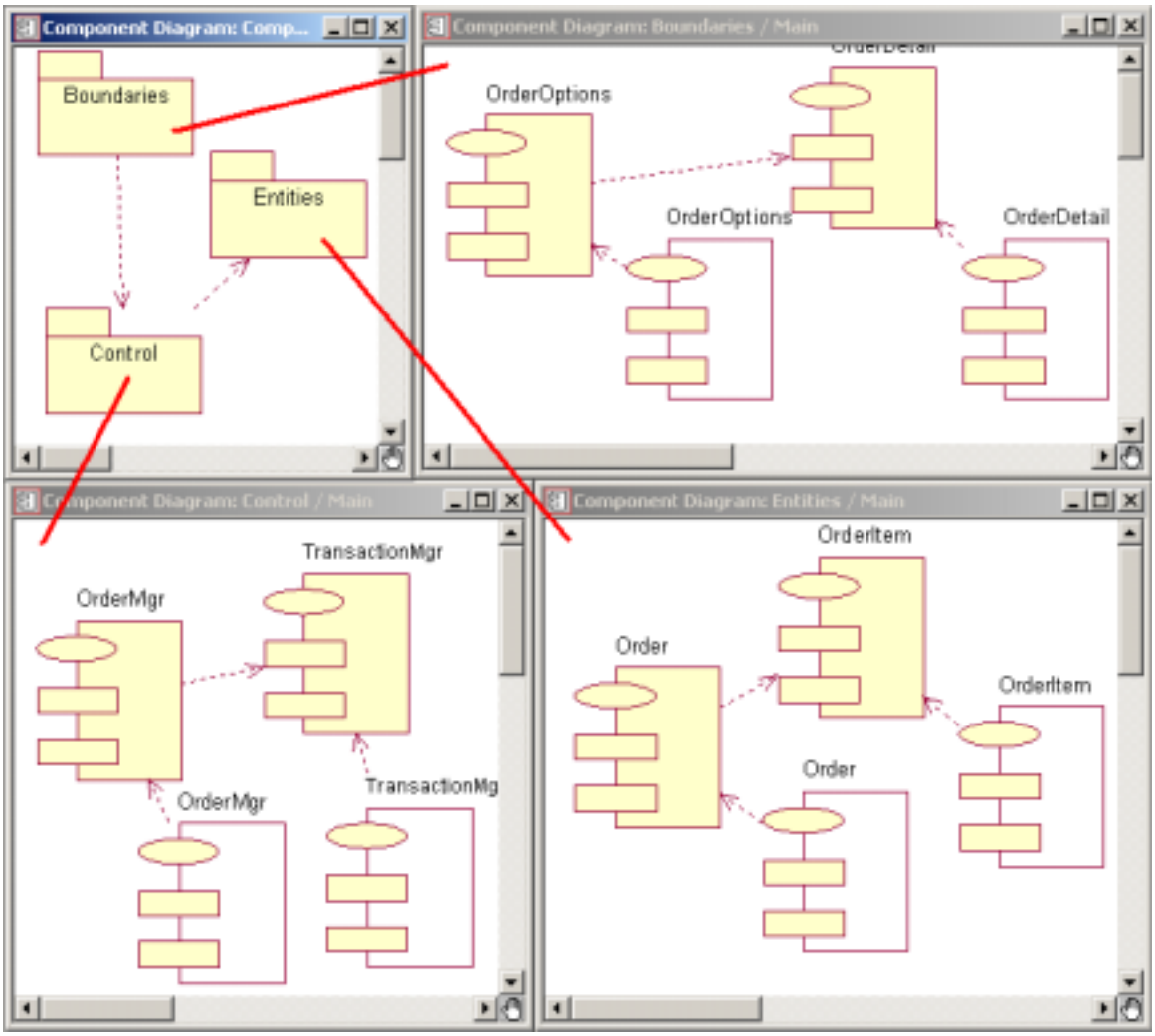
مرحله‌ی ششم پروژه (فایل step6.mdl)

گفتیم که دیاگرام حالت وضعیت‌های مختلف یک شی را در طول عمر آن نمایش می‌دهد، در نتیجه برای اشیایی که لازم باشد (کلاسهای از نوع entity) در صورت لزوم دیاگرام حالت را رسم می‌کنیم.



مرحله‌ی هفتم پروژه (فایل step7.mdl)

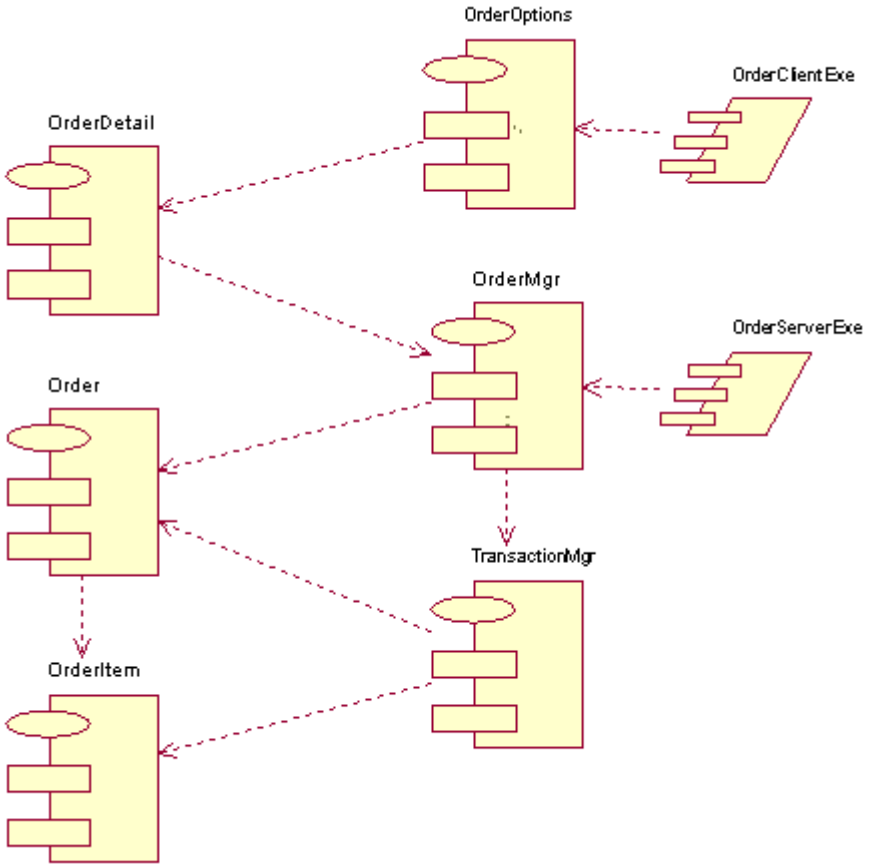
حال نوبت به بخش component view می‌رسد.ابتدا در دیاگرا اصلی این نما سب بسته‌ی اصلی معماری سه‌لایه را قرار داده و ارتباط منطقی بین آنها را برقرار می‌کنیم.



سپس یک نمودار component diagram جدید می‌سازیم (وجود این نمودار به این دلیل است، چون در main ما سه بسته‌ی جدا داریم، در نتیجه نمی‌توانیم ارتباط قسمت‌های مختلف بین دو بسته را بر قرار کنیم)، حال به ازای هر کلاس که در دیاگرام اصلی کلاسهایمان داریم یک جز Package specification قرار می‌دهیم و ارتباط بین آنها را بر قرار می‌کنیم.

چون برنامه‌ی ما یک برنامه‌ی Client\Server می‌باشد، دو جز Task specification در نمودارمان قرار می‌دهیم، که یکی مربوط به برنامه‌ی اجرایی روی Client می‌باشد و دیگری مربوط به Server است.

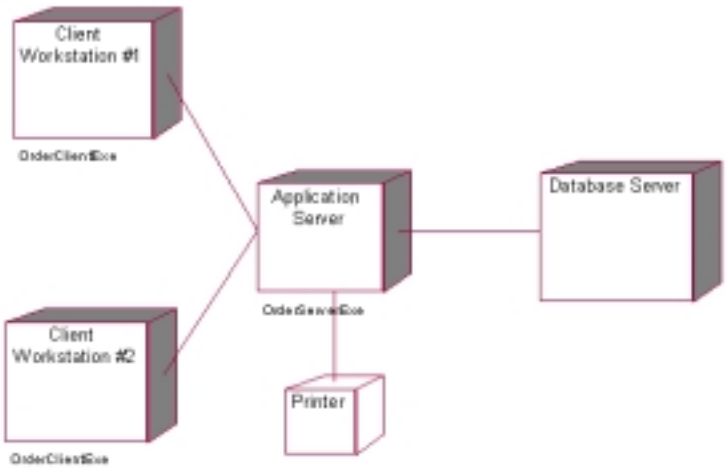
شکل مربوط به این دیاگرام در صفحه‌ی بعد آمده است :



دیاگرام ارتباطی اجزا

مرحله‌ی هشتم پروژه (فایل step8.mdl)

با توجه به تعریف پروژه به سادگی می‌توان نمودار پیاده‌سازی زیر را برای سیستم در نظر گرفت. تنها نکته‌ای که نباید فراموش گردد، نسبت دادن پروسه‌های اجرایی روی کامپیوترهای server و client می‌باشد. برای اینکار در browser روی نود مربوط به پردازشگر مورد نظر راست کلیک کرده و new > process را انتخاب نمایید و پروسه‌ی مورد نظر را به آن نسبت دهید.

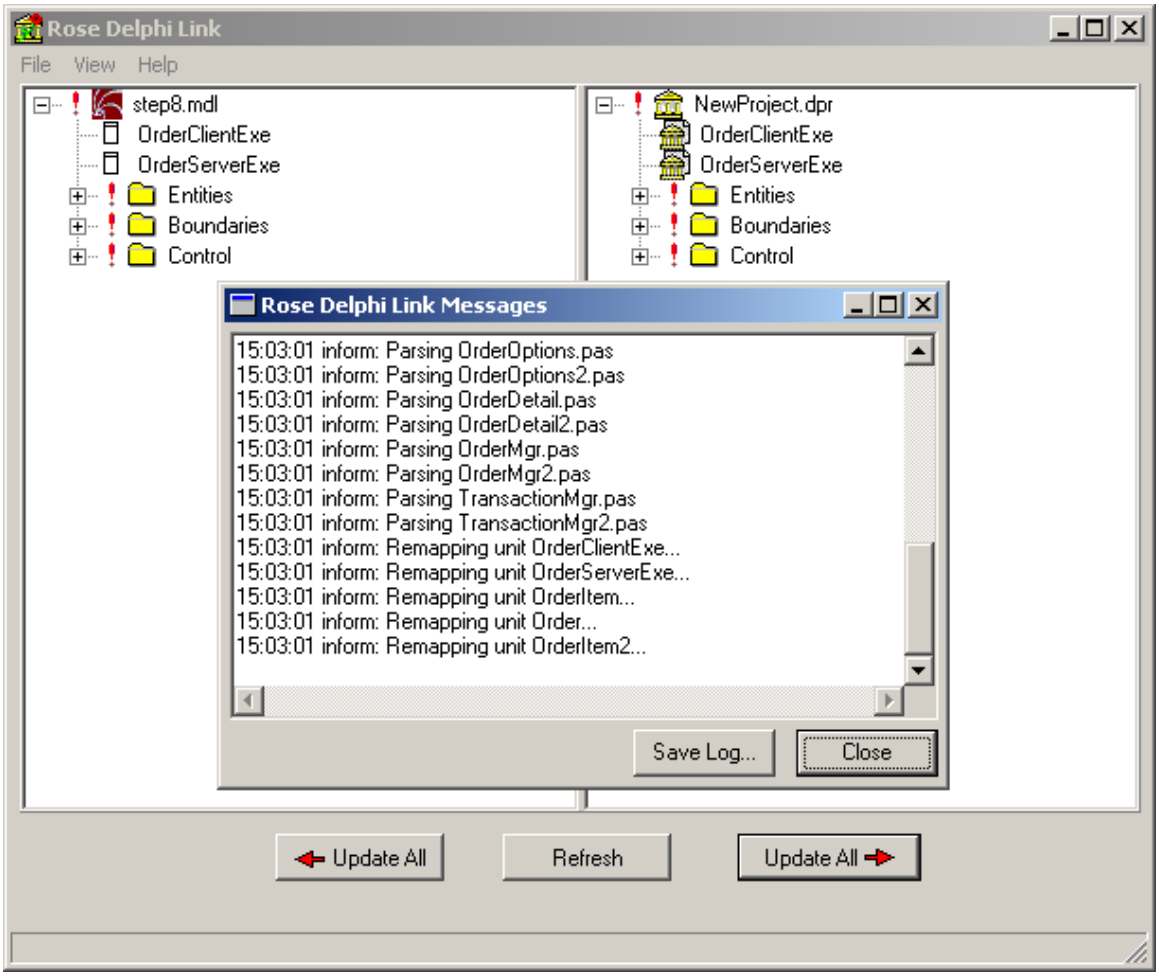


چگونگی تولید کد دلفی از مدل و یا عکس عمل فوق:

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

حال از منوی tools گزینه‌ی Ensemble Tools > Rose Delphi Link ... را انتخاب نمایید. اکنون برای برو ز در آوری کد مورد نظر ویا update کردن مدل با توجه به کد موجود، پروژه‌ی مورد نظر را بازکرده و با توجه به جهت نشان داده شده، عملیات مورد نظر را انجام دهید.

اگر هدف شما تولید کد می‌باشد و پروژه را قبلا نساخته اید، ابتدا می‌بایستی از منوی File گزینه‌ی new project را انتخاب نمایید، ولی بجای اینکار، پیشنهاد می‌کنیم که ابتدا یک پروژه‌ی خالی را توسط دلفی ساخته و بجای انتخاب این گزینه، پروژه‌ی ساخته شده را باز نمایید.

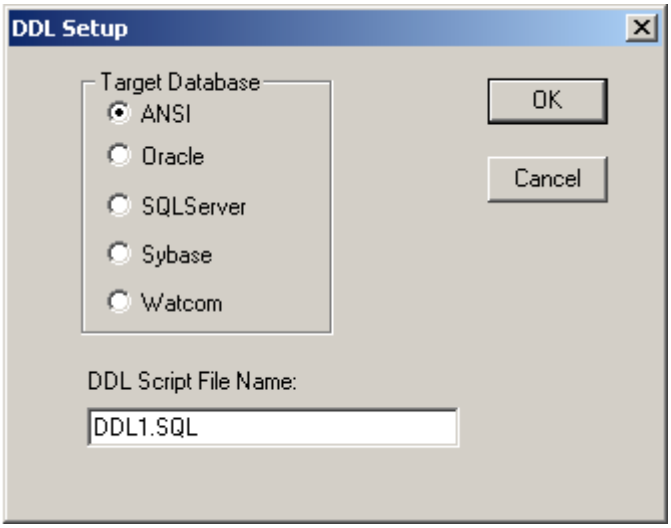


نحوه‌ی تولید پایگاه داده‌ی مربوط به پروژه :

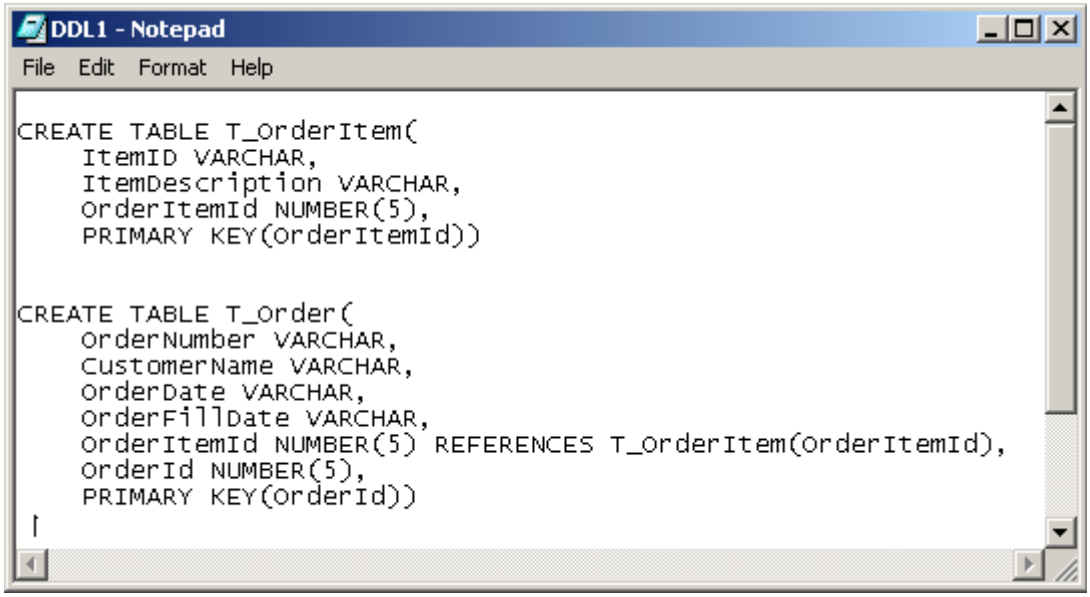
می دانیم که بانک اطلاعات پروژه از کلاسهای entity بدست می‌آیند..برای تولی تولید کد SQL مربوطه، مطابق با هر کدام از بانک‌های موجود نظیر access ، SQL Server، oracle یا ... ، ابتدا باید کلاسهای entity خود را به صورت

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 برگه‌ی Detail بروید و تغییرات لازم را اعمال کنید.

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

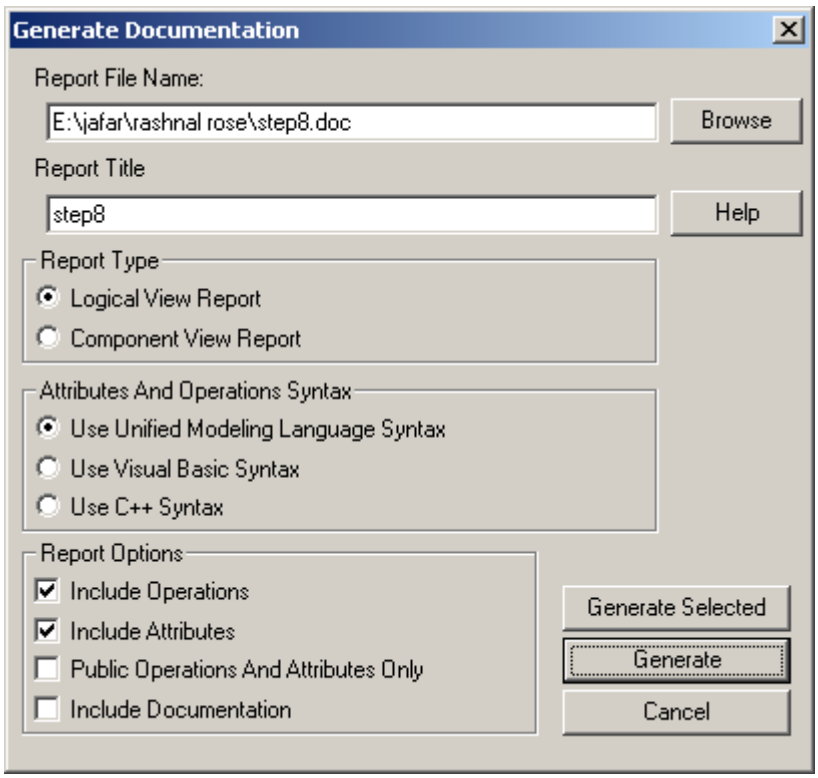


برای مشاهده‌ی کد تولید شده می‌توانید از منوی tools گزینه‌ی DDL > Browse DDL را انتخاب نمایید.



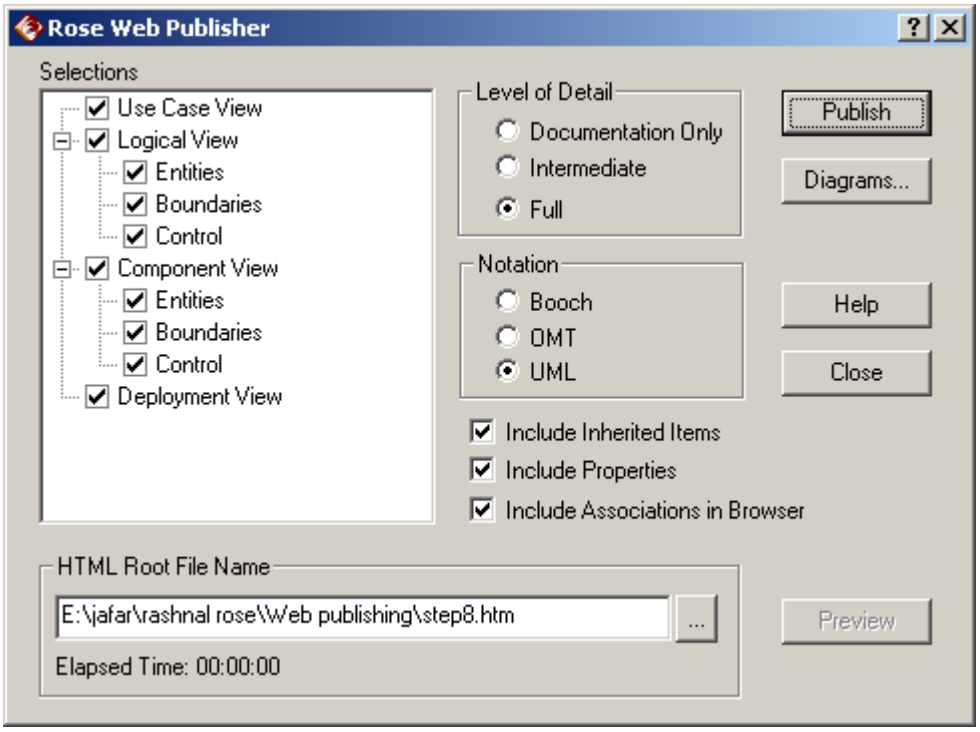
نحوه‌ی گزارش گرفتن از پروژه:

برای تولید مستندات مدلتان (نماهای منطقی و اجزا) و مشاهده‌ی آنچه که رشنال رز برای شما تولید کرده است، از منوی reeport گزینه‌ی ... Documentation report را بکار ببرید. گزارشهای تولیدی برای پروژه‌ی ما در انتهای نوشتار ضمیمه می‌باشد.



نحوه‌ی انتشار پروژه:

در منوی tools گزینه‌ی ...Web publisher عملیات انتشار را برای پروژه‌ی شما انجام می‌دهد. این عمل یک نسخه‌ی html از تمامی مدل شما همراه با مستندات تهیه می‌کند، تا سایر کاربران با دیگر browserهای موجود نیز بتوانند مدل شما را ببینند. فرم مربوطه در صفحه‌ی بعدمی‌باشد و فایل html تولیدی نیز در دیسکت ضمیمه موجود می‌باشد.



فرم تولید فایل html مربوط به مدل