معرفی متدولوژی‌های Oracle CDM و RUP و مقایسه این دو متدولوژی

و بررسی معماری CDM Rule Frame

(چهارچوب توسعه لایه business logic در معماری سه لایه)

محمد حسین شرکت - میثم میرزازاده

- مقدمه

یک عامل بسیار مهم و تعیین کننده در موفقیت سیستم‌های جامع نرم افزاری، انتخاب نوع متدولوژی مناسب است. در این تحقیق به دنبال معرفی کلیاتی از دو متدولوژی مطرح در زمینه طراحی سیستم‌های جامع نرم افزاری یعنی RUP، از مجموعه متدولوژی‌های بر اساس تفکر شی گرا و Oracle CDM بر اساس تفکر ساخت یافته بوده ضمن مقایسه این دو متدولوژی به بررسی معماریCDM Rule Frame(چهارچوب توسعه لایه business logic در معماری سه لایه) پرداخته خواهد شد.

۱. معرفی متدلوژی Oracle CDM

متد (CDM-Custom Development Method)، متد چرخه حیات کامل اوراکل برای تحویل سفارشی برنامه های کاربردی است که از لحاظ سیر توسعه و ایجاد آن دنبال روی متدولوژی Case Method ریچارد بارکر محسوب می‌شود.

CDM فرایندها و عملیات های Business را که بوسیله برنامه های کاربردیOff-The-Shelf قابل حل نیستند را مشخص نموده و تمام چرخه حیات توسعه سیستم را پوشش می‌دهد.

CDM Version ۲ شامل دو روش اصلی است: CDM کلاسیک وCDM Fast Track. اگرچه هر دو روش چرخه حیات توسعه سیستم را بطور کامل می پوشانند ولی رویکرد آنها در بعضی از جهات همچون دیدگاه مدیریتی کاملا متفاوت بوده و بنا به موقعیت پروژه انتخاب می‌شوند.

۱-۱- CDM کلاسیک

CDM کلاسیک مجموعه‌ای خوش ساخت است که برای توسعه سیستم‌های اطلاعاتی از ابتدا تا انتها تدوین گردیده است و می‌تواند پروژه‌های حوزه سیستم‌های اطلاعاتی را به طرق مختلف مدیریت نماید.

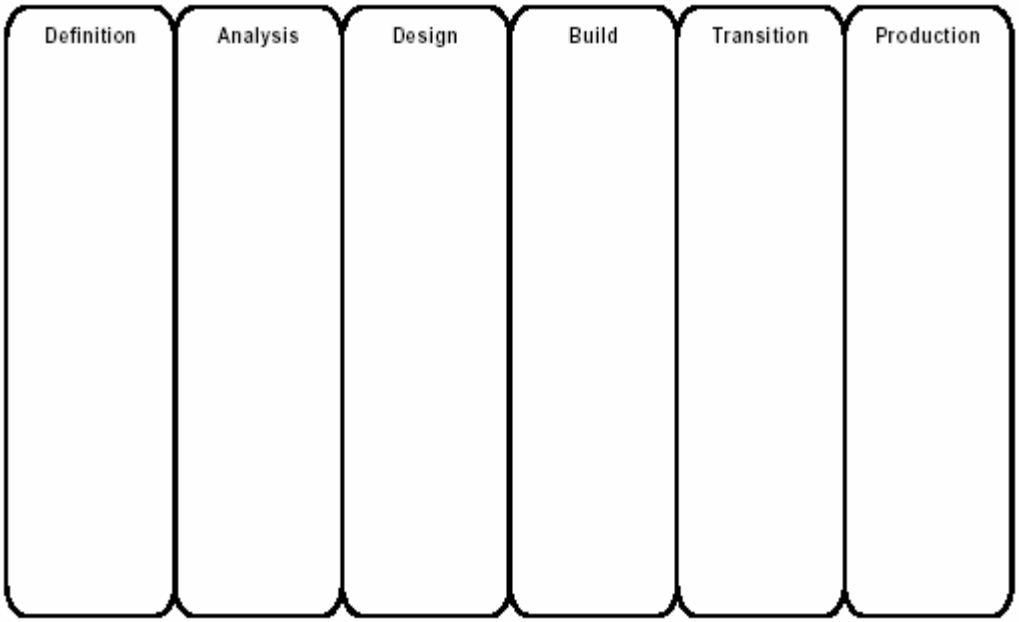
در پروژه‌هایی که توسط CDM تعریف می‌شوند، تمام وقایع داخلی و خارجی کسب و کار اداره شده و یک رخداد در کسب وکار به یک فرآیند سازمانی یا سیستمی پیوند می‌خورد.

www.hajarian.com

CDM برای پروژه‌های متوسط و بزرگ تدوین شده است ولی برای پروژه‌های متوسط نیز قابل استفاده است. این متد با استفاده از Oracle Repository وOracle Database به صورت موثرتری عمل خواهد کرد. در CDM وابستگی طبیعی بین خروجی‌ها کاملا تعریف شده بوده و پیش نیازهای هر وظیفه و وابستگی بین آنها به طور تام و کامل مشخص است.

۱-۱-۱- فازها در CDM کلاسیک

پروژه ها در CDM در قالب فاز ها (مراحل) هدایت می گردند. در این فاز ها کیفیت و نقاط کنترل برای هماهنگی فعالیت هایی که دارای مقاصد مشترکی هستند، ارائه می شود. شکل زیر شش فاز موجود در CDM را نشان می‌دهد.



CDM Classic Phases

شکل ۱: فازهای موجود در متدولوژی CDM

در ادامه به تشریح بیشتر هر یک از این فازها پرداخته می‌شود.

- فاز تعریف(Definition)

مقصود از فاز تعریف، تعیین نیازمندی های سازمان و سیستم های اطلاعاتی، در بالاترین سطح است و برای رسیدن به مجموعه اهداف تعریف شدهٔ سازمان، انجام این فاز لازم می باشد. در این فاز نتایج حاصل از چهارچوب پروژه به روشنی تعریف شده و در انتها برای آغاز فاز تحلیل، تایید مدیر اخذ می‌شود.

- فاز تحلیل (Analysis)

مقصود از فاز تحلیل، تنظیم رسمی نیاز مندی‌های تفصیلی، برای سیستم های کاربردی کامپیوتری می‌باشد. در فاز تحلیل، حوزه های کسب و کاری سازمان که قبلا در فاز تعریف، معین شده است بررسی می‌گردد. اعضای تیم تحلیل با تهیه مجموعه‌ای دقیق از مدل‌ها، به توضیح هرچه دقیق ترفعالیت‌های انجام شده در سازمان و روند گردش اطلاعاتی

آن می‌پردازند و در نهایت اطلاعات استخراج شده به مدل‌هایی تبدیل می‌شوند که نیازمندی‌های تفضیلی سیستم مکانیزه را مشخص می‌نمایند. در این مرحله معماری سیستم تعریف و استراتژی انتقال آن به سیستم پیشنهاد می‌گردد.

- ❖ فاز طراحی (Design)

مقصود از فاز طراحی، گرفتن نیاز مندی ها از فاز تحلیل و تبدیل آن ها به مشخصات تفصیلی سیستم می‌باشد(با توجه به در نظر گرفتن معماری فنی و فن‌آوری‌های در دسترس)

- ❖ فاز ساخت(Build)

مقصود از فاز ساخت، کد کردن و آزمون نمودن کاربردها با استفاده از تکنیک های مناسب است. این تکنیک‌ها بسته به نوع پیمانه‌ها، متفاوت بوده و از ساخت سنتی تا ساخت سریع را با استفاده از روش توسعه گام به گام(Incremental) را شامل می‌شود.

- ❖ فاز انتقال(Transition)

مقصود از فاز انتقال، نصب سیستم کاربردی، آماده سازی پرسنل مشتری برای استفاده و راهبری سیستم و سپس بهره برداری از آن می باشد. تیم پروژه در فاز انتقال اعمالی همچون نصب سیستم، آموزش پرسنل، حمایت از تست های پذیرش و قرار دادن سیستم در وضعیت بهره برداری را بر عهده دارند.

- ❖ فاز بهره برداری(Production)

مقصود از فاز بهره برداری، پشتیبانی از سیستم، اطمینان از اجرای روان آن و طرح ریزی برای توسعه های کارکردی آن ها در آتیه می باشد.

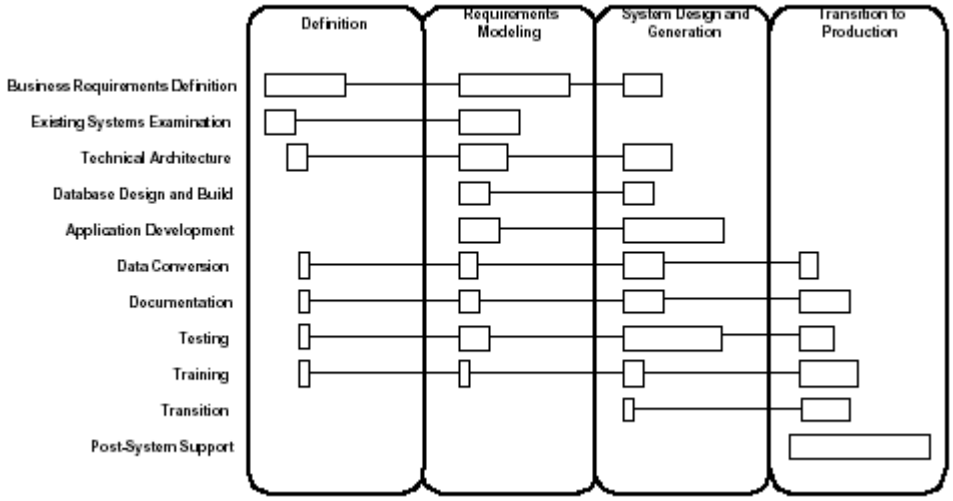
شایان ذکر است که هر یک از فازهای ارائه شده شامل موارد زیر می گردند:

۱. وظائف تفکیک شده
۲. خروجی ها
۳. تخمین زمان برحسب فرآیند
۴. زمان بندی و تهیه گانت چارت
۵. پیشنهاد زمان بندی
۶. جایگزینی نیرو

۱-۱-۲- فرآیند ها در CDM

همانگونه که اشاره گردید CDM یک متدولوژی توسعه سیستم مبتنی بر نگرش فرآیندی است. یک فرآیند از مجموعه وظایف (تکالیف) وابسته و مربوط به هم تشکیل شده که برای حصول به هدف مشخصی تدوین گشته است. هر فرآیند باعث بوجود آمدن خروجی یا خروجی هایی می گردد.

www.hajarian.com



شکل ۲: فازها و فرآیندها در CDM

فرآیندهای CDM کلاسیک به شرح زیرمی‌باشد.

❖ تعریف نیازمندی های کسب و کار

در این فر آیند، نیاز های کسب و کار برای ساخت سیستم‌های کاربردی تعریف می‌شود. در ابتدا تیم تحلیل، مدل فرآیندی کسب و کار را ارائه می‌کنند که شامل تمامی وقایع و پاسخ‌هایی است که متعاقب این وقایع رخ می‌دهد. پس از آن مدل داده‌ای و کارکردی کسب و کار که جزئیات بیشتری در آن ارائه شده همراه با تعریف نیازمندی‌های کسب و کار، به مدل اضافه می‌شود.

❖ بررسی سیستم های موجود

یکی از نیازمندی های مهم در پروژه‌های توسعه سیستم های اطلاعاتی، جایگزین نمودن سیستم‌های مکانیزه با معماری فنی قدیمی، با سیستم هایی با معماری فنی جدید است. این وظیفه در این فرآیند انجام می‌پذیرد. واضح به نظر می‌رسد که درصورت نبود سیستم موجود، بسیاری از تکالیف موجود در این فرآیند حذف می‌شود.

❖ معماری فنی

در فرآیند معماری فنی، عناصر پیکره ی فنی توسعه سیستم مشخص می‌گردد. درابتدا فرض بر آن است که یک برنامه کلان استراتژیک برای معماری سیستم وجود دارد که عناصر معماری سیستم جدید می‌بایست در چهارچوب آن قرار گیرد.

❖ طراحی و ساخت پایگاه داده

این فرایند با ایجاد طرح پایگاه داده منطقی از روی مدل داده‌ای سیستم شروع می‌شود و با تکمیل تعریف داده‌های پایگاه خاتمه می‌یابد.

❖ طراحی و ساخت پیمانه

قلب پروژه‌ای که در آن CDM استفاده می‌شود، طراحی و ساخت پیمانه(Module) است. طراحان با استفاده از مدل فرآیندی سیستم، مدل داده‌ای سیستم و مدل کارکردی سیستم به همراه معماری فنی، اقدام به طراحی یک پیمانه نموده و سپس جزئیات فنی و کارکردی هر پیمانه مشخص می‌گردد.

❖ تبدیل داده ها

هدف از تبدیل داده ها، انتقال دادن، تبدیل کردن و آزمون نمودن تمامی داده‌های موجود است که برای آزمون یا کار عملیاتی سیستم مورد نیاز است.

❖ مستند سازی

فرآیند مستند سازی بر روی تهیه مستندات متنی با کیفیت بالا تمرکز دارد. تمام مستندات آموزشی و فنی کاربران، در پروژه تهیه می‌شود. دو مستند برای کاربران تهیه می‌شودکه یکی مرجع کاربر و دیگری مرجع راهنمای کاربر می‌باشد.

❖ تست

فرآیند تست یک راه کار برای تست نمودن کیفیت کار تمامی عناصر یک سیستم کاربردی است. این فرآیند شامل آزمون کارکردی پیمانه ها، تست یکپارچگی پیمانه ها در سطح کسب و کار و تست پذیرش و یکپارچگی سیستم‌ها می-باشد. همه تست‌ها در سطح کسب و کار توسط مدل فرآیندی هدایت می‌شود تا چهارچوبی برای یک تکیه‌گاه قابل پی-گیری برای نیازمندی‌های کسب و کار ایجاد می‌شود.

❖ آموزش

هدف فرآیند آموزش، تمهید کاربران و راهبرانی است که به اندازه کافی آموزش دیده باشند تا بتوانند وظایف اجرایی سیستم جدید را انجام دهند.

❖ انتقال

فرآیند انتقال به وسیله تعریف زود هنگام نیازمندی‌های معین، برای تغییر و تحول به سیستم جدید شروع می‌شود و شامل تکالیفی است که برای اجرای عناصر اولیه همچون توسعه سیستم جدید، برنامه نصب، تهیه محیط بهره برداری، انجام تغییر و تحول و کنار زدن هرگونه سیستم قدیمی مورد نیاز می‌باشد.

❖ پشتیبانی بعدی سیستم

این فرآیند چهار هدف برای انجام پشتیبانی‌های بعدی دارد:

۱. مراقبت و پاسخگویی به مشکلات سیستم از طریق دفتر کار راهنمایی؛
۲. بروز رسانی کاربرد برای درست نمودن خطاها و رفع مشکلاتی که جنبه افزایش کارایی دارند؛
۳. ارزیابی سیستم در هنگام بهره برداری؛

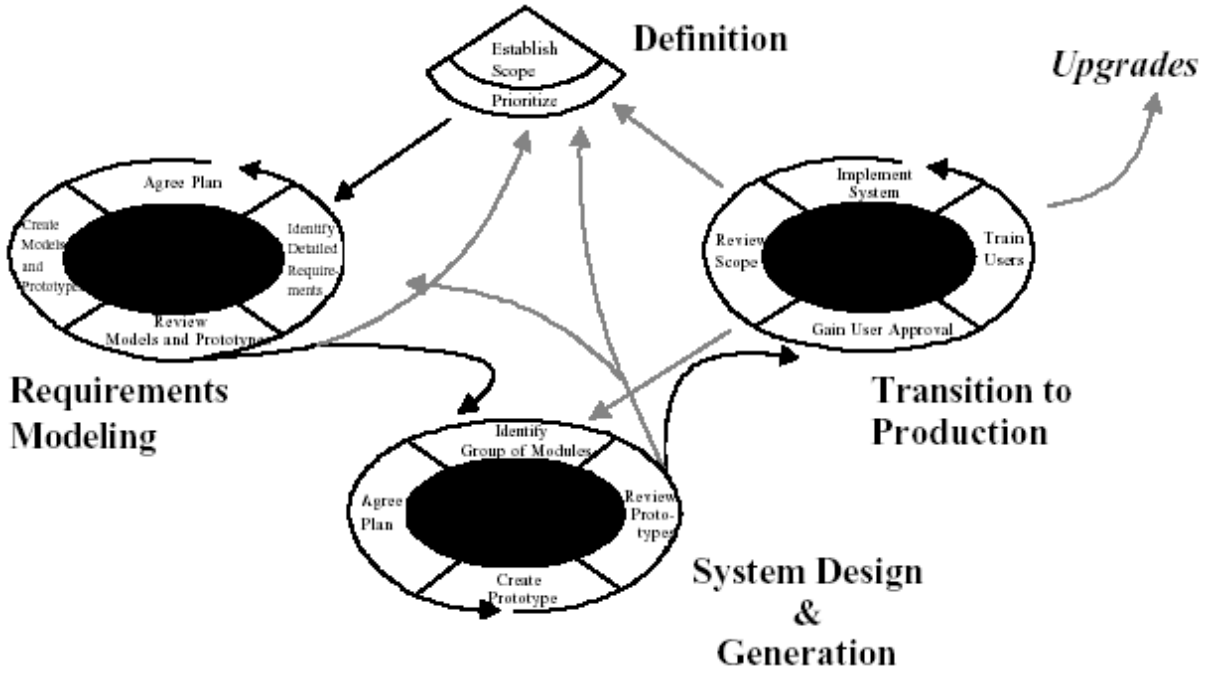
www.hajarian.com

۴. طرح ریزی برای گسترش کاربرد سیستم؛

۱-۲- CDM Fast Track

CDM Fast Track یک متد توسعه برنامه‌های کاربردی است که از خانواده تکنیک‌های Rapid Application Development (RAD) محسوب می‌شود. این متد از استاندارد روش توسعه پویای سیستمDSDM (Dynamic System Development Method) تبعیت می کند. این نگرش برای متدولوژی‌های تحت اوراکل طراحی شده و کاربران را در پروسه تولید و توسعه مشارکت می‌دهد.

با نگرش Fast Track فاز های متدولوژی CDM به شکل زیر تغییر کرده و عمل کرد جدید زیر را دارا می‌باشد. پروژه ها درCDM Fast Track در قالب فاز ها (مراحل) زیر هدایت می‌گردند. در این فاز ها کیفیت و نقاط کنترل برای هماهنگی فعالیت‌هایی که دارای مقاصد مشترکی هستند، ارائه می شود. در طول فازهای پروژه، تکالیفی از فرآیندهای مختلف توسعه سیستم اجرا می‌گردد.



شکل ۳: چرخه حیات در CDM

۱-۲-۱- متدولوژی CDM با نگرش Fast Track

❖ فازبرنامه ریزی

هدف این فاز دستیابی به یک تعریف روشن از حدود و ثغور پروژه، اولویت بندی اجزا پروژه و برنامه اجرایی پروژه است. معمولا حدود و ثغور با کمک ترکیبی از جلسات کارگاهی، نمونه سازی و یا گرفتن بازخورد از کاربران انجام می-

www.hajarian.com

شود و شامل تعیین اهداف، تعیین فعالیت‌های عمده و تعیین واحدهای سازمانی درگیر صورت می‌گیرد. اولویت بندی اجزا پروژه براساس روش زمان ثابت انجام می‌شود و از اقلام اجباری تا اقلام غیرضروری تفکیک می‌گردد.

برنامه اجرایی پروژه شامل برنامه نیازهای پروژه به منابعی مانند نیروی انسانی، مالی، سخت افزار و نرم افزار و غیره، اقلام قابل تحویل پروژه، جزئیات فعالیت‌های پروژه، طرح کیفیت پروژه، برآوردهای مالی و زمانی پروژه، برنامه زمانی و فرسنگ شمارهای پروژه می‌شود.

❖ فاز مدل سازی نیازها

درمرحله مدل سازی نیازها درCDM بیشتر به این توجه می شود که چه کاری انجام می‌شود. باید اشاره نمود که بررسی چگونه انجام شدن آن در مرحله طراحی روشن می شود. در نگرش Fast Trackچنین مرزی وجود ندارد و جلسات کارگاهی و نمونه سازی‌ها ممکن است هر دو موضوع را شامل شود. هدف این مرحله دستیابی به یک مدل اطلاعاتی، یک مدل عملیاتی، مدل معماری سیستم و یک نمونه ساخته شده است.

اهم وظائف مرحله مدل‌سازی عبارت است از:

۱- تعیین نیازها از طریق تشکیل جلسات کارگاهی

۲- قبل از نمونه سازی معمولا استانداردهای صفحات نمایش، گزارش‌ها و امثال آن نمونه سازی شده و نظر کاربران اخذ می‌گردد که بدان FEEL PROTYPING & LOOK "یعنی دیدن و پسندیدن " نمونه سازی گفته می شود.

۳- نمونه سازی سیستم

این وظیفه به منظورتعیین هدف نمونه سازی، طراحی پایگاه داده سیستم نمونه، طراحی عملیات سیستم نمونه، طراحی ساختار یا معماری سیستم نمونه، ساخت سیستم نمونه و بررسی، مرور، اصلاح و نهایی سازی نمونه ساخته شده در جلسات کارگاهی انجام می‌شود.

❖ فاز طراحی و ساخت

در این مرحله نمونه ساخته شده در مرحله مدل سازی نیاز، اصلاح و تکمیل گردیده و سیستم نهایی ساخته می شود.

در این مرحله نیز هر سیستم پس از ساخت، درجلسات پیوسته کارگاهی یا به صورت‌های دیگر بررسی شده و نظر کاربران و سایر افراد ذینفع اخذ می‌شود و بتدریج سیستم ساخته شده نهایی می‌گردد. این نوع ساخت را ساخت تدریجی می‌گویند.

خروجی‌های نهایی این فاز عبارتنداز:

۱- متن برنامه ها؛

۲- پایگاه داده‌های ساخته شده ؛

۳- سیستم ساخته شده، تست شده و آماده اجرا؛

۴- مدارک کاربری و راهبری سیستم ؛

۵- مدارک تست سیستم؛

❖ فاز انتقال و تولید

این مرحله CDM در روش CDM Fast Track تفاوت چندانی با مرحله مشابه در روش ندارد و اهم فعالیت‌های آن عبارتند از:

۱- نصب و آموزش سیستم؛

۲- تست پذیرش سیستم توسط کاربران و انجام تحویل موقت؛

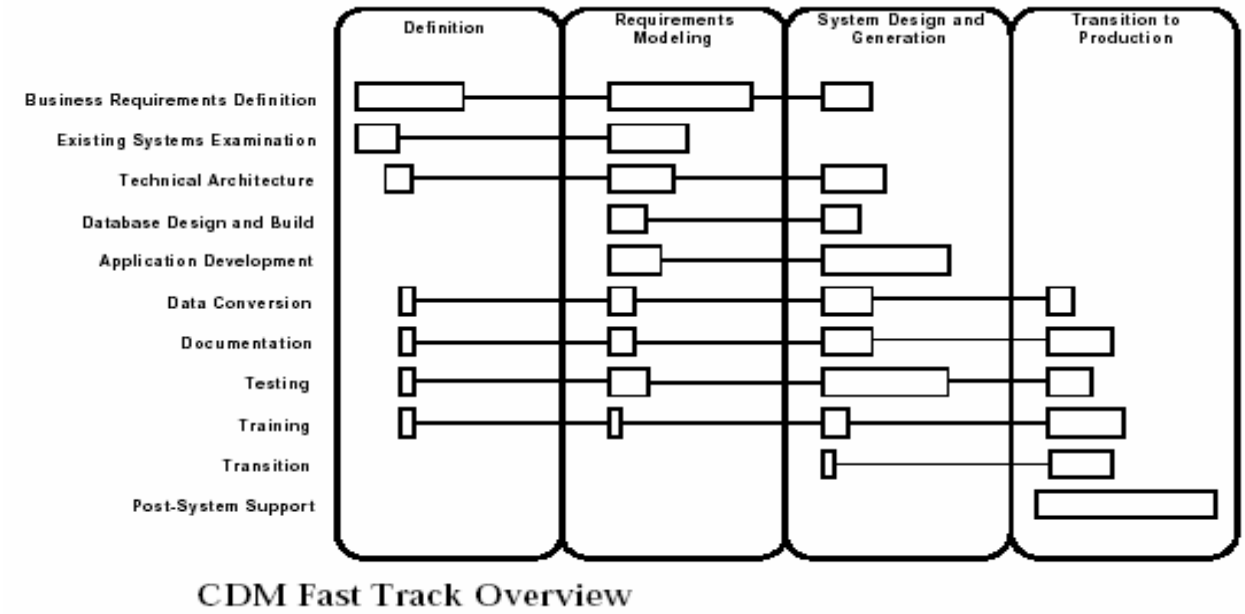
۳- اجرای موازی و اجرای آزمایشی سیستم؛

۴- انتقال از سیستم قبلی به جدید؛

۵- تحویل قطعی؛

۱-۲-۲- فرآیندهای متدولوژی CDM با نگرش Fast Track

یک فرآیند از مجموعه وظایف وابسته و مربوط به هم تشکیل شده است که برای حصول به هدف مشخصی تدوین گشته است. هر فرآیند باعث بوجود آمدن خروجی یا خروجی هایی می‌گردد.



شکل ۴: فرآیندهای متدولوژی CDM با نگرش Fast Track

www.hajarian.com

❖ فرآیندتعریف نیاز های کسب وکار:

فرآیند تعریف نیاز های کاری، یک سری از مدل سازی نیازها و یک لیست از اولویت بندی نیازها می باشد که به عنوان خروجی تولید می‌شود.

❖ بررسی سیستم های موجود

وظایف و خروجی ها مشابه CDMکلاسیک تعریف می‌شوند با این تفاوت که نقش‌های جدید و تغییر یافته در این فرآیند قرار می گیرد.

❖ معماری فنی

وظایف و خروجی‌ها مشابه CDM کلاسیک تعریف می‌شوند با این تفاوت که نقش‌های جدید و تغییر یافته در این فرآیند قرار می گیرد.

❖ طراحی و ساخت پایگاه داده

این فرایند با ایجاد طرح پایگاه داده منطقی، از روی مدل داده‌ای سیستم شروع می‌شود و با تکمیل تعریف داده‌های پایگاه داده برای بهره برداری خاتمه می‌یابد. وظائف و خروجی‌ها شبیه CDM کلاسیک می باشد ولی بعضی از موارد در پیشرفت Fast Track کمک می کنند. این فرآیند با فرآیند توسعه کاربردی ارتباط نزدیکی دارد.

❖ فرآیند توسعه کاربردی

این فرآیند از مجموعه وظایف اصلی تکراری که به توسعه سیستم کمک می کند، تشکیل شده است. این فرآیند با تعریف استاندارد طراحی و ساخت شروع می شود.

این فرآیند مخصوص Fast Track بوده و جایگزین فرآیند طراحی و ساخت پیمانه در CDM کلاسیک شده و ارتباط محکمی با فرآیند تعریف نیاز های کسب وکار و طراحی و ساخت پایگاه داده دارد .

❖ فرآیند تبدیل داده

فرآیند انتقال داده در ین روش آسانتر از CDM کلاسیک است. به این ترتیب که با استفاده از آرشیو داده و جلوگیری از تاخیر، کیفیت بهتری را بوجود می‌آورد.

❖ فرآیند مستند سازی

فرآیند مستند سازی درCDM Fast Track خلاصه تر از فرآیند مستند سازی درCDM کلاسیک بوده و مستندات با همکاری ما بین توسعه دهنده و کاربر تولید می‌شود.

❖ فرآیند آزمون

فرآیند تست یک راه کار یکپارچه برای تست نمودن کیفیت کار تمامی عناصر یک سیستم کاربردی است. این فرآیند شامل آزمون‌های زیر است:

۱- آزمون کارکردی پیمانه‌ها؛

۲- تست یکپاچگی پیمانه ها در سطح کسب و کار؛

۳- تست پذیرش و یکپارچگی سیستم ها؛

❖ فرآیند آموزش

هدف فرآیند آموزش، ایجاد زمینه آموزش کاربران سیستم است که به اندازه کافی آموزش دیده باشند تا بتوانند وظایف اجرایی سیستم جدید را انجام دهند .

❖ فرآیند انتقال

وظایف و خروجی ها مشابه CDM کلاسیک تعریف می‌شوند.

❖ فرآیند پشتیبانی بعدی سیستم

وظایف و خروجی ها مشابه CDM تعریف می شوند با این تفاوت که فرآیند آزمون ارتقاء سیستم در فرآیند آزمون دو باره تعیین و مشخص می گردد.

۱-۲-۳- تکنیک های CDM Fast Track

مجموعه‌ای ازتکنیک‌ها و فنونی است که برای ایجاد شتاب بیشتر در فرایند توسعه سیستم به طورمشترک با کاربران و سایر طرف‌های ذینفع صورت می‌گیرد. به عبارت دیگر در این روش تاکید عمده بر کار مشترک در مراحل مختلف توسعه سیستم است. این تکنیک‌ها به صورت خلاصه عبارتند از:

❖ شکستن پروژه های بزرگ به پروژه های کوچکتر و قابل اجرا(Partion)

در فاز برنامه‌یزی پروژه با بررسی بزرگی و ابعاد پروژه امکان شکستن آن به پروژه های کوچکتر مورد بررسی قرار می-گیرد.

❖ اولویت بندی انجام کار براساس زمان ثابت (MoSCoW List)

در اولویت بندی انجام کار بر اساس زمان ثابت چهار نوع عملیات ممکن است وجود داشته باشد:

۱- عملیات یا جنبه های اجباری: عبارت از اقلامی که حتما باید در پروژه موردنظر گنجانده شوند.

۲- عملیات یا جنبه های الزامی: عبارت از اقلامی که بودن آنها ضروری است ومعمولا در برنامه پروژه نیز گنجانده می شوند .

۳- عملیات یا جنبه‌های شایسته: عبارت از اقلامی که بودن آنها در پروژه مفیداست ولی جنبه الزامی ندارد.

۴- عملیات یا جنبه های غیرضروری: عبارت از اقلامی است که بهتر است در پروژه موردنظر وجود نداشته باشند .

❖ نمونه سازی (Prototyping)

www.hajarian.com

نمونه سازی روشی است که در آن یک نمونه سریع و تقریبی از یک سیستم یا بخشی از آن ساخته می شود. چنین نمونه هایی معمولا برای نمایش به کاربران و سایرافراد ذینفع تهیه می شوند.

❖ آنالیز و طراحی مشترک ازطریق برگزاری کارگاه (Workshop)

برگزاری کارگاه یاجلسات مشترک بین عوامل ذینفع درتوسعه سیستم یکی ازابزارهای الزامی در Fast Track است به طوری که درتمام مراحل توسعه سیستم، کارگاه‌ها و جلساتی به منظورهای مختلف تشکیل می شود.

۱-۲-۴- ارزیابی متدولوژی CDM با نگرش Fast Track

❖ مزایا

۱- سرعت بخشیدن به روند توسعه سیستم ها

۲- در تمام مراحل، مشارکت عملی کاربران نهایی را در تصمیم گیری و حتی طراحی سیستم تضمین کند. به این ترتیب فرآورده نهایی، هنگامی که برای استفاده عملی تحویل می شود، قائدتا باید بیشترین تناسب را با نیازها و سلیقه های کاربران داشته باشد.

۳- محدود بودن گروه‌های تولید کننده سیستم در روش Fast Track (بین ۳ تا ۴ نفر) ارتباطات لازم را برای هماهنگی و کنترل پروژه ها به میزان قابل توجهی کاهش می‌دهد و به همین دلیل راهبری و مدیریت پروژه های Fast Track را بسیار ساده تر می کند.

❖ معایب

۱- اصولا این نگرش برای تهیه سیستم های کوچک و متوسط طراحی شده است.

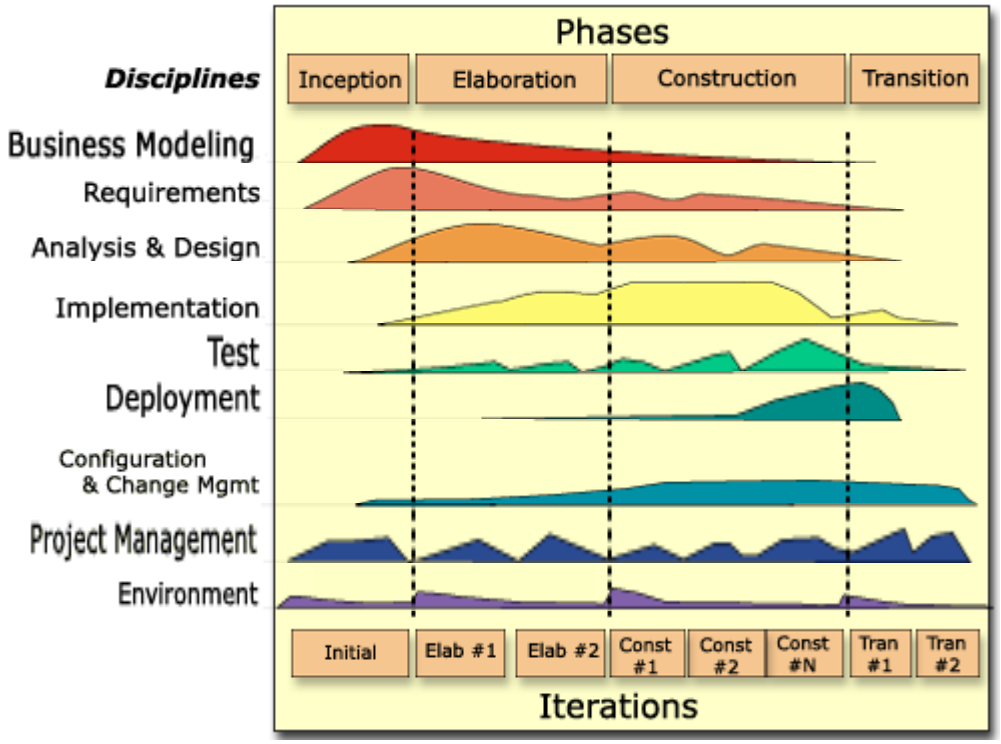
۲- استفاده از ابزار اصلی Fast Track یعنی جلسات طراحی مشترک در بسیاری از موارد دشوار یا حتی غیر ممکن است .

۳-تفهیم کارایی و ضرورت این روش به مدیران سازمان و سایر کاربران آسان نیست.

در کشور هایی مانند کشور ما نباید انتظار داشت که همه مدیران شرکت کننده در طراحی مشترک با مفاهیم و روش-های تهیه سیستم و به ویژه با ابزار های مدل سازی آشنا باشند.

۲- معرفی متدلوژی RUP

محصول Rational Unified Process یا RUP یک فرایند مهندسی نرم افزار است. RUP یک روش منظم را برای تعیین کارها و مسئولیت ها با یک سازماندهی پیشرفته فراهم می کند که هدف آن اطمینان از تولید یک نرم افزار با کیفیت بالا است که در آن نیازهای کاربران نهایی با بودجه و زمانبندی از قبل پیش بینی شده هماهنگ باشد.



شکل ۵ – نمای کلی فازها و نوع عملیات در Rational Unified Process

ساختار کلی RUP دارای دو بعد است:

- محور افقی که نمایان‌گر زمان بوده و جنبه های چرخه حیات فرایند را نشان می دهد. به عبارت دیگر نشان دهنده جنبه پویای فرایند است.
- محور عمودی، ترتیب منطقی فعالیت‌ها را نشان داده و جنبه ایستای فرایند را به تصویر می کشد. چگونگی آن در واژه هایی از قبیل قطعات، ترتیب ها، فعالیت ها، جریان های کار، محصولات و نقش ها شرح داده شده است.

نمودار ۵ نشان می دهد که در طول زمان تاکید روی مسائل متفاوتی است مثلاً در ابتدای تکرار زمان بیشتری روی نیازمندی‌ها صرف شده و در انتهای تکرار زمان بیشتری صرف پیاده سازی می شود.

۲-۱- ویژگی‌های RUP

این فرایند با استفاده از تکنیک‌هایی شبیه تکنیک‌های طراحی نرم افزار تهیه شده است. RUP بسیاری از بهترین روش‌های توسعه نرم افزارهای مدرن را در سازگارترین و مناسب ترین فرم برای محدوده وسیعی از پروژه‌ها وسازمان‌ها استفاده می‌کند. بعضی از این موارد عبارت است از:

- توسعه نرم افزار مبتنی بر تکرار (Iterative)
- مدیریت نیازمندی‌ها
- استفاده از معماری Component-Base

- مدل سازی بصری نرم افزار(Visual Modeling)
- کنترل مستمر کیفیت
- کنترل تغییرات برای نرم افزار

۲-۱-۱- توسعه نرم افزار مبتنی بر تکرار

بیشتر تیم‌های نرم افزاری هنوز از مدل آبشاری(Waterfall) برای توسعه پروژه ها استفاده می‌کنند و تکامل در یک ترتیب دقیقی از فازهایRequirement Analysis Design implementation/Integration وTest انجام می شود.

در روش آبشاری، وقت اعضای کلیدی تیم در دوره های وسیعی تلف می شود و مرحله تست تا پایان چرخه حیات پروژه به تعویق می افتد. در مقایسه RUP یک روش مبتنی بر تکرار را نشان می دهد که به دلایل زیر بهتر است:

- RUP اجازه می دهد که نیازمندی‌های تغییر یافته را به حساب آوریم. حقیقت این است که نیازمندیها معمولاً تغییر می کنند. تغییر نیازمندی‌ها و Requirement Creep ، نیازمندی‌های اضافی که غیر لازم هستند و یا از جانب مشتری مشتق نشده اند، همیشه منبع اصلی آشفتگی پروژه، تحویل با تاخیر، ازبین رفتن زمانبندی ونارضایتی مشتری و خنثی کردن کارتوسعه دهندگان است.
- تکامل، یک حرکت بزرگ(Big Bang) در پایان نیست، درعوض عناصر بصورت گام به گام و تقریباً پیوسته کامل می شوند.
- ریسک ها معمولاً در طول تکامل کشف می شوند و مورد توجه قرار می گیرند. با دسترسی مبتنی بر تکرار می‌توان ریسک ها را به آسانی کم کرد.
- تکرار استفاده مجدد را تسهیل می کند. شناسایی بخش‌هایی که جداگانه طراحی یا پیاده سازی شده‌اند برای تشخیص درطول طرح آسانتر است. بازبینی طراحی در تکرارهای اولیه به طراح و معمار اجازه می دهد تا فرصت‌هایی را برای استفاده مجدد درنظر بگیرند وسپس کدهای مشترک را برای این فرصت ها در تکرارهای بعدی توسعه دهند وتکامل بخشند.
- هنگامیکه بتوان خطاها رادرطول تکرارها تصحیح نمود، نتیجه معماری قویتری خواهد بود. هنگامیکه محصول از فاز Inception به فاز Elaboration می رود ، نقص ها زودتر از وقتی که یک مرحله تست عظیم در پایان داریم کشف می شوند. .
- فرایند توسعه می‌تواند در طول راه اصلاح و تفکیک شود. ارزیابی در پایان هر تکرار فقط در نظرگرفتن حالات پروژه یا زمانبندی نیست بلکه آنالیزآنچه که باید در سازماندهی و فرایند تغییر کند تا در تکرار بعدی بهتر انجام شود .

www.hajarian.com

مدیران پروژه اغلب در مقابل این شیوه مبتنی بر تکرار مقاومت می کنند و آن را به عنوان یک روش بی پایان و غیر قابل کنترل می بینند. در RUP شیوه مبتنی بر تکرار بسیار کنترل شده است چراکه تعداد، مدت واهداف تکرارها طراحی شده اند وکارها و مسئولیت های افراد بخوبی تعریف شده است. بعضی از دوباره کاری‌ها از یک تکرار تا تکرار بعدی رخ می دهند اما این موضوع نیز به دقت کنترل شده است.

۲-۱-۲- مدیریت نیازمندی‌ها

مدیریت نیازمندی‌ها یک شیوه سیستماتیک برای ،سازماندهی، ارتباط ومدیریت تغییرات نیازمندی‌های سیستم متمرکز نرم افزار و یا برنامه کاربردی است.

فواید مدیریت موثر نیازمندی‌ها متعدد است که در ادامه به مواردی از آن اشاره می‌گردد:

- ❖ کنترل بهتر پروژه های پیچیده

این شامل درک بهتر از رفتار سیستم موردنظر بعلاوه جلوگیری از Requirements Creep است.

- ❖ بهبود کیفیت نرم افزار و رضایت مشتری

معیار اساسی کیفیت این است که سیستم آنچه را برایش درنظرگرفته شده تا چه حد انجام می دهد. با RUP این تشخیص ساده تر است چون همه ذینفع ها درک مشترکی از آنچه که باید ساخته وتست شود دارند .

- ❖ کاهش هزینه ها وتاخیرهای پروژه

تثبیت خطاها در نیازمندی‌ها پر هزینه است. بوسیله مدیریت موثر نیازها می‌توان خطاها را در راه توسعه زودتر کاهش داده و بدین وسیله هزینه‌های پروژه را کم و از بروز تاخیرات جلوگیری نمود.

- ❖ بهبود ارتباطات تیمی

مدیریت نیازمندی‌ها درگیری کاربران را در فرایند زودتر تسهیل می‌کند و کمک می‌نماید تا کاربران از هماهنگی برنامه کاربردی با نیازهایشان مطمئن شوند. مدیریت خوب نیازمندی‌ها یک درک مشترک از نیازمندی‌های پروژه وارتباطات بین ذینفع‌های مختلف از جمله کاربران، مشتریان، مدیریت، طراحان وآزمایش کننده ایجاد می کند.

۲-۱-۳- استفاده از معماری Component-Base

در RUP، چرخه توسعه اولیه، شکلی از یک پیش نمونه معماری قابل اجراگرفته است که از طریق تکرارهای بعدی تدریجاً تکامل یافته و به سیستم نهایی تبدیل می شود.

یکComponent می‌تواند به عنوان یک تکه مهم نرم افزاری تعریف شود: یک ماژول، یک بسته و یا یک زیرسیستم که عملکرد روشنی را انجام می دهد و حد و مرز مشخصی دارد.

RUP بوسیله راه‌های مختلف از توسعه Component-Base حمایت می‌کند:

www.hajarian.com

- شیوه تکرار به توسعه دهنده اجازه می دهد تا مکرراً قطعات را شناسایی کند و تصمیم بگیرد که کدامیک را خریداری کند.
- مفاهیمی مانند بسته‌ها و زیرسیستم‌ها ولایه‌ها در طول آنالیز و طراحی برای سازماندهی قطعات و تعیین Interface ها استفاده می‌شوند.
- تست در ابتدا در اطراف قطعات به صورت تک تک سازماندهی شده و سپس تدریجاً به مجموعه‌های بزرگتر برای قطعاتی کامل تر گسترش پیدا می کند.

۲-۱-۴- مدلسازی بصری نرم افزار (Visual Modeling)

مدل‌ها، مختصرسازها و ساده کننده‌های واقعی هستند که برای فهمیدن یک مساله و راه حلش به ما کمک کرده و فهم سیستم‌های پیچیده و بزرگتر را ساده‌تر می‌نمایند. یک بخش بزرگ از RUP درباره رشد و گسترش ونگهداری مدل‌های سیستم‌های در حال توسعه است.

Unified Modeling language-UML یک زبان گویا برای مجسم کردن، مشخص کردن و ساخت و مستند سازی سیستم‌های فشرده نرم فزاری است.UML ابزار و وسیله استاندارد برای نوشتن طرح و عملکرد سیستم‌ها است. اگرچه UML واژه‌هایی را برای مدل‌های مختلف فراهم می‌کند ولی چگونگی توسعه نرم افزار را نمی‌گوید.

۲-۱-۵- کنترل مستمر کیفیت

کیفیت، وظیفه و مسئولیت هر فرد در سازماندهی توسعه سیستم است. در توسعه نرم افزار اهمیت کیفیت در دو قلمرو مورد توجه قرار گرفته است: کیفیت محصول وکیفیت فرایند.

کیفیت محصول: کیفیت محصول اصلی تولید شده(نرم افزار و یا سیستم) و همه عناصر تشکیل دهنده آن را شامل می‌شود. (برای مثال قطعات ، زیرسیستم ها ، معماری و...)

کیفیت فرایند: درجه‌ای که فرایند قابل قبول(شامل سنجش و معیاری برای کیفیت)پیاده سازی می‌شود و در طول ساخت محصول رعایت می‌شود را شامل می‌گردد. به علاوه، کیفیت فرایند وابسته به کیفیت محصولات (مانند طرح‌های مکرر یا تکراری، طرح های تست وتحقق وفهم Use Case و مدل طراحی و...) تولیدی در حمایت ازتولید اصلی است.

۲-۱-۶- مدیریت تغییرات برای نرم افزار (Change Management)

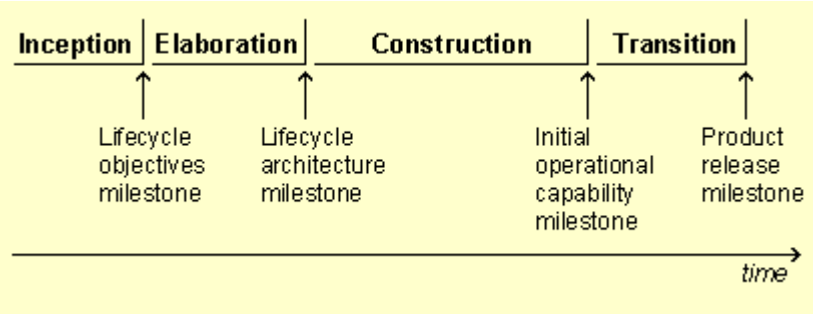
از جمله مسائل مهمی که پروژه‌های نرم افزاری را با ناکامی مواجه می‌سازد، تغییرات نیازهای کاربران است. RUP مکانیزم و ابزارهایی برای کنترل سیستماتیک و اعمال این تغییرات فراهم می سازد.

۲-۲- چرخه حیات RUP

از دیدگاه مدیریتی، چرخه حیات نرم افزار Rational Unified Process (RUP) به چهار فاز متوالی تقسیم می-شود که هریک بوسیله یک مرحله مهم به پایان می رسند.

www.hajarian.com

هرفاز در واقع یک مدت و فاصله معین بین دو مرحله عمده است. در پایان هر فاز یک ارزیابی برای تعیین اهدافی که در این فاز بدست آمده انجام می‌شود. یک ارزیابی رضایت بخش اجازه حرکت پروژه به مرحله بعد را می دهد.



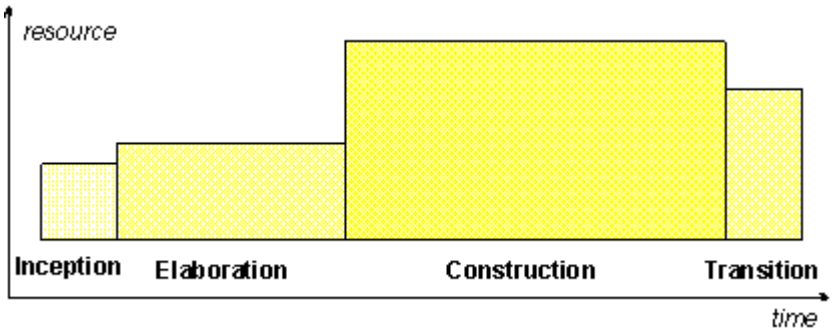
شکل ۶ – فازها ونقاط مهم یک پروژه

۲-۲-۱- طرح ریزی فازها

همه فازها در میزان تاثیر و زمان بندی مساوی نیستند. اگرچه این تفاوت‌ها به پروژه بستگی دارد، یک چرخه توسعه ابتدایی برای یک پروژه با سایز متوسط باید بر طبق توزیع زمانبندی و تاثیر زیر پیش رود.

	Inception	*Elaboration*	*Construction*	*Transition*
Effort	~۵٪	۲۰٪	۶۵٪	۱۰٪
Schedule	۱۰٪	۳۰٪	۵۰٪	۱۰٪

شکل ۷ – توزیع آماری حجم و زمان اجرا بین فازها در RUP



شکل ۸ – دیاگرام توزیع آماری حجم و زمان اجرا بین فازها در RUP

۳- مقایسه RUP و OracleCDM

هر دو متدولوژی RUP و CDM از CMM-Level۳ پشتیبانی می‌نمایند. در زیر چند نمونه از تفاوت‌های این دو متدولوژی به صورت کلی اشاره گردیده است:

- در CDM مشابه روش‌های قدیمی، تحلیل و طراحی بعنوان فاز مطرح شده است که تجربه نشان می دهد این نوع فازبندی کار را با مشکل مواجه خواهد ساخت.
- در CDM تاکید روی فرایندهای مربوط به بانک‌های اطلاعاتی است: طراحی DB ، طراحی ماژول و نوشتن شرح ماژول ها، تبدیل داده و مانند آن.
- در CDM تاکید روی نوشتار است و Visual Modeling جایگاهی بجز بیان ساختار بانک اطلاعاتی ندارد.
- یکی از مهم‌ترین خصوصیات UML در Encapsulation داده و فرایند هاست. در متدولوژی‌های ساخت یافته، فرایندها جداگانه در نظر گرفته شده و به‌جای مفاهیمی مانند Generalization، مفاهیم کلی‌تری مانند Grouping مطرح می‌باشد.

۴- بررسی معماری CDM Rule Frame

در این دوره که رشد و ساخت برنامه های اینترنتی روز به روز گسترش یافته است فهم این مطلب که این برنامه‌ها و سیستم‌ها به صورت معماری سه لایه سازماندهی می‌گردد خالی از لطف نیست. این سه لایه عبارتند از لایه ارائه (Presentation layer)، لایه منطق داد و ستد (bussiness logic layer) و لایه داده ها (Data layer)

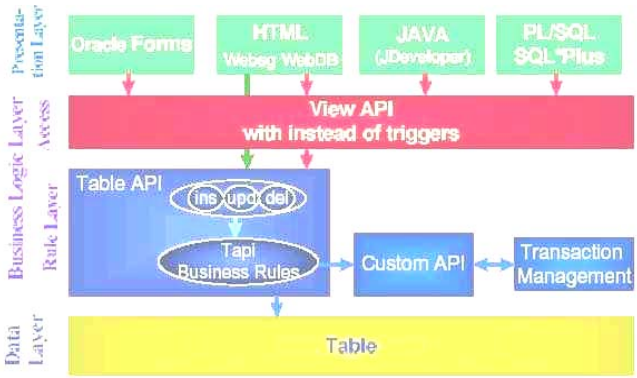
این سازماندهی در حقیقت تامین کنندهٔ دوام و پایایی سیستم‌ها گردیده، مدیریت و طراحی و توسعه سیستم‌ها را بهبود بخشیده و می رود تا جایگزین معماری مشتری- کارگزار (Client – Server) سنتی شود.

CDLRuleframe نیز به صورت یک مخزن و انبار قدرتمند در آمده و چهارچوب توسعه لایه business logic مستقل را فراهم می کند. این حمایت توسط ابزارهایی از قبیل:

- web DB
- disigner webserver generation
- designer forms generator Oracle

در جهت توسعه سیستم‌ها در دسترس قرار گرفته اند.

شکل زیر معماری CDM ruleframe را به صورت کلی نشان می دهد.



شکل ۱- نمای شماتیک معماری CDM ruleframe

www.hajarian.com

۴-۱- Business Rule

قبل از ورود به بحث بهتر است در مورد معنا و واژه Business Rule بیشتر تامل کنیم و با چند مثال این عبارت را بیشتر بشناسیم. در یک تعریف خوب و کامل می توان Business Rule را یکی از موارد زیر در نظر گرفت.

۴-۱-۱- Data Constraint

این گروه از Business Rule ها در واقع نگهدارندهٔ درستی، صحت و اعتبار داده هاست و شامل دو بخش زیر می شوند:

- یک محدوده از این گروه مربوط به نگهداری وضعیت داده هاست (Static Data Constraint)

مثال: یک سفارش کلی عبارت است از مجموع خطوط سفارشات

- محدودهٔ دیگری از این گروه عبارتند از قوانینی که تغییرات داده ها را روشن می کنند (Dynamic Data Constraint)

مثال: شما نمی توانید سفارش خودتان را بعد از این که این سفارش ارسال شد لغو کنید.

۴-۱-۲- Authorization

این بخش از Business Rule ها چگونگی استفاده از داده ها توسط کاربران معینی را مشخص کرده و در واقع وظیفه استفاده کننده را به صراحت مطرح می‌کند.

مثال: یک مشتری می تواند تنها سفارشات خود را لغو (کنسل) کند.

۴-۱-۳- Change event

این قواعد در برگیرندهٔ عملیاتهای اتوماتیکی هستند که پس از وقوع تغییرات در داده ها بایستی اتفاق بیافتد و به دو گروه زیر تقسیم می شوند.

- آن دسته که وظیفه دستکاری داده ها را مشخص کرده و انجام می دهند یا به عبارتی با استفاده از زبان دستکاری داده ها (DML) رویدادها و تغییرات را انجام می دهند.

مثال: هنگامیکه سفارشی تغییر می کند. این تغییر باید در جدول journal ثبت شود.

- آن دسته که راه اندازی رویدادها را خارج از پایگاه داده نشان می دهند.

مثال: هنگامیکه یک سفارش حمل شد (ارسال شد) یک email اطلاعاتی به منظور اطلاع رسانی به مشتری زده می شود.

۴-۲- Business logic

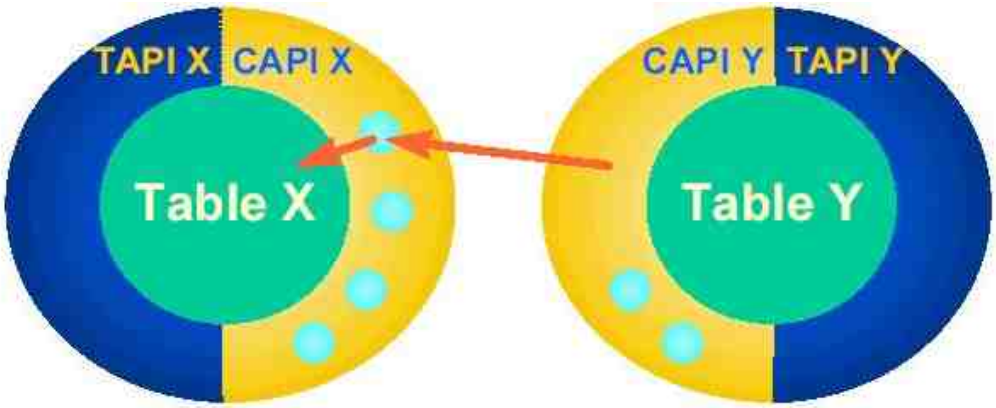
در ابتدا می توان این طور عنوان کرد که لایه Business logic تشکیل شده از دسترسی های عمومی لایه ای به همراه لایه قواعد (Rule layer).

یک Rule layer با استفاده از مجموعه ای از بسته های PL/SQL توسعه یافته می‌شود که درون پایگاه داده مستقر شده اند. ابتدا یک پکیج مدیریتی مسئول، سطوح قوانین را تعریف، آنها را توسعه و در انتها مورد تائید و امضاء قرار می دهد. (تمامی قواعد به صورت سنتی به شکل API تنها در این سطح تراکنش قرار می‌گیرند.)

دوم اینکه هر جدول، مجموعه‌ای از دو بسته به صورت زیر خواهد بود.

- Table API Package: که توسط Oracle designer تولید می شود.
- Custom API package: که به همراه CDM Rulefrane شرایط و امتیازات oracle designer محسوب می شوند.

شکل زیر نشان دهندهٔ یک business rule توسعه یافته است.



شکل۲- شمایی از یک business rule توسعه یافته

این دو بسته با یکدیگر عملیات انجام داده و اصول عملیاتی business Rule را نشان داده و تائید می کنند.

این عملیات و توابع بصورت زیر سازماندهی می شوند.

ـ تعیین منشأ و مقادیر پایه‌ای

ـ معتبر سازی و اعتبار سنجی داده ها

ـ تصدیق عملیات‌ها

ـ کنترل کردن اختیارات

ـ راه اندازی عملیاتهای اضافی و افزوده

هر TAPI,CAPI می‌توانند ساختارهای درونی جدول خودشان را ملاحظه کرده و قابلیت دسترسی به داده‌های دیگر جداول از طریق TAPJ / CAP۳ خواهند داشت. اجازه دسترسی بصورت عمومی می تواند از ملاحظه و دیدن راه اندازها صورت گیرد (با استفاده از راه اندازهای TAPE ۲ پایگاه داده)

این دسترسی ها می‌تواند به صورت مسئولیت دریافت درخواست‌های DML از لایه ارائه و جهت دهی و هدایت آنها به سمت لایه قواعد (Rule layer) پیاده سازی شود.

۵- اجزای کلیدی در تعاریف Business Rule:

در تنظیم و تشریح کامل یک Business Rule بایستی تمامی اطلاعات ذیل بصورت صریح و شفاف تعیین گردد.

۵-۱- شرایط راه اندازی

زمانیکه تخلفی از این قانون به وقوع می پیوندد، زمانیکه ما احتیاج به تغییر رویدادها و وقایع راه اندازی داریم، زمانیکه یک ردیف وارد می شود، به روز می شود، حذف می شود، در هنگام بروزرسانی چک کردن قوانینی زمانیکه ستونها به روز (Update) می شود و یا قرار دادن شرایط خاص بر روی مقادیر یک ستون داده ای همگی از شرایط ویژه قواعد و قوانین Business هستند که بایستی به صورت شفاف تعریف شوند.

۵-۲- کاربرد و استعمال داده ها

جهت کد کردن یک قانون CAPI نیازمند داده هایی از ستونهای جداول خواهیم بود که بایستی این داده ها از ستونهای جدید و قدیمی تامین شود.

۵-۳- پیامدهای خطا

چه پیامهای بایستی در هنگام نقض قانون به نمایش درآید (این پیامها در ابتدا بایستی در دایرکتوری از پیش تعریف شده پیامها تعریف شوند)

۵-۴- نوع قانون

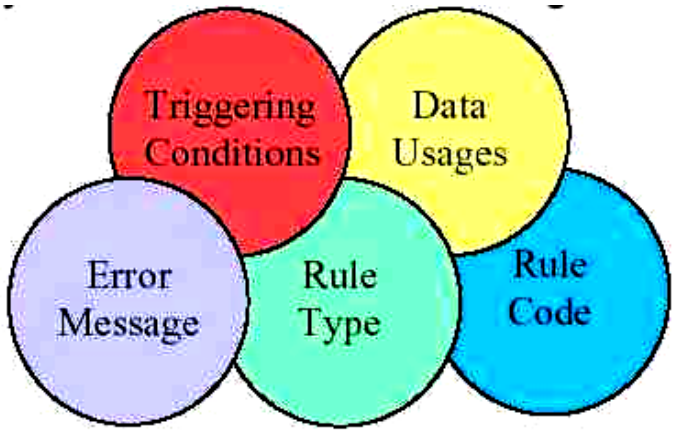
اینکه یک قانون با DML رویدادهای تغییرات را اعمال کند (Data constraint) یا اینکه این قانون این تغییرات را بدون استفاده از DML انجام می دهد بایستی مشخص شود. پس از آنکه تمامی قواعد CAPI در تراکنش‌ها اعمال گردید، این اطلاعات در جهت روشن کردن این موضوع که این تنظیمات صحیح، در زمان درست و به موقع انجام می شوند حائز اهمیت خواهد بود. رویدادهای مبتنی بر DML در ابتدا سازماندهی می‌شوند و سپس این قواعد موجب راه اندازی و بکارگیری دیگر قواعد و قوانین خواهند شد. در واقع بایستی تفاوتی بین ثابت و یا متغیر بودن Data Constraint قائل شد.

در واقع Static Data Constraint باعث رجوع به داده ها و چک کردن تمام یا قسمتی از داده و اعتبار سنجی آنها خواهند شد. این پردازش‌ها برروی جداول موجود شکل می گیرند. در صورتیکه درDynamic Data Constraint این مراجعه امکان پذیر نبوده و تنها می تواند عملیاتهای جاری را سازماندهی کرد.

۵-۵- کد قانون

عبارت است از یک یا بیشتر از یک دستور و فرمان که پس از اجرای آنها نتیجه وقوع قانون و انجام عملیاتهای مورد نظر آن خواهد بود. برگردان مقدار False بیانگر تناقض در برقراری این قانون خواهد شد.

موارد کلیدی در تعاریف و طراحی یک business Rule در شکل زیر نشان داده شده است.



۶- معرفی CDM Ruleframe

Business Rule ها به صورت واسط توسعه دهنده و نگهدارنده چهارچوب کاری یک سازمان در نظر گرفته می‌شود. می‌توان گفت پیاده سازی و اعمال Business Rule در واقع یکی از عملیاتهای خسته کننده و طاقت فرسا در توسعه نیازمندیهاست. دلیل این موضوع را می‌توان در تعاریف پر ابهام و نامشخص Business Rule ها جستجو کرد که به کرات به وقوع پیوسته و اکثراً به صورت کاملاً صحیح و شفاف ثبت نمی شوند. این عامل موجب عدم توانایی در تخمین و برآورد زمانی و ارزیابی هزینه و منابع خواهد شد که یکی از موارد مهم و مورد نیاز در فاز طراحی می باشد. این مورد همچنین به این معناست که هر طراح می تواند یک تفسیر و ترجمه مخصوص به خود از Business Rule داشته باشد. نداشتن تعریف دقیق و واضح از Business Rule را می‌توان نشات گرفته از فقدان چهارچوب کاری (Frame work) در مورد آن دانست که فقدان این رمز و راز برای رهبری و مدیریت می تواند عواقب ناخوشایندی را به همراه داشته باشد، از جمله نگهداری را با سختی همراه کند، عملیاتهای مورد نظر به گونه متفاوتی تفسیر شوند و در نهایت یکپارچگی سرویس دهندگان ما را دچار مخاطره کند.

اینترنت در طول عمر خود موارد مهمی را با خود به ارمغان آورده است. همچنین بالا رفتن سرعت توسعه سیستم ها و انعطاف پذیر کردن آنها در مقابل تغییرات در نیازمندیهای یک Business که همین مورد به خودی خود یکی از موارد مهم و کلیدی در پذیرش پروژه ها محسوب شده و به عنوان یک فاکتور کیفی برای پروژه ها محسوب می شود.

اجرا، قالب بندی و توسعه Business Rule ها به صورت چهارچوب (Frame worK) مشخص و واضح می‌تواند راه حلی مناسب برای حل این مشکل باشد که در صورت عدم پرداختن به آنها بدست آوردن فاکتورهای کیفیت برای پروژه

های مورد نظر غیر ممکن به نظر می رسد. سرانجام می توان گفت که از دست دادن فرصت در زمینه طراحی و لحاظ کردن این چهارچوب می تواند ریسک بزرگی محسوب شده و موجب کاهش انگیزه و تحرک در میان کارکنان گردد.

CDM Ruleframe را می‌توان عرضه کننده و توسعه دهنده چهارچوب Business Rule دانست که تامین کننده تمامی اهداف و مقاصد اشاره شده خواهند بود.

CDMRuleframe یک چهارچوب قدرتمند برای ساختار به کارگیری و پیاده سازی یک BusinessRule می توان معرفی کرد. این ساختار توسعه برنامه ها و Application های اینترنتی سه لایه ای (three-tier) را مورد حمایت قرار می دهد. از آن رو که oracle server تامین کنندهٔ لایه داده ها بوده و کیت ابزار توسعه (Development Tools) لایه ارائه را سازماندهی و آماده می کنند، می توان اینطور عنوان کرد CDM Ruleframe هم وظیفه توسعه لایه business logic را بر عهده دارد. با این ترتیب تلفیق این مفاهیم با یکدیگر و قرار دادن این مورد در معماری نرم افزارها می‌تواند موجب ترقی و پیشرفت در ارائه محصول شود.

مثلاً وقتی که ما سیستم خود را به صورت اینترنتی ارائه می‌کنیم، نرم افزارهای نهایی متنوعی را نیاز خواهیم داشت که بایستی با پذیرش هر یک به ارائه آنها بپردازیم. مثلاً یک نرم افزار نهایی (Font end) پیچیده مانند Web Forms برای استفاده کنندگان پر اهمیت درون سازمانی مورد نیاز خواهد بود و همچنین یک نرم افزار نهایی ساده و آسان برای دیگر استفاده کنندگان بایستی در نظر گرفته شود. (html based می تواند در این سیستم لحاظ شود).

یک Business logic بایستی قابلیت استفاده مجدد در این نرم افزار های نهایی متفاوت را داشته باشد و در هر کدام قابل پیاده سازی باشد. به علاوه اینکه ما نیازمند یک معماری منعطف خواهیم بود که نه تنها Business logic را توسعه داده بلکه نگهداری این قواعد و قوانین را ساده نماید. اما این نکته را نباید فراموش کرد که نیازمندیها و قوانین Business به مرور زمان دچار تغییر و تحول خواهد شد. CDM Ruleframe نه تنها از استفاده مجدد کردن Business logic در نرم افزارهای نهایی متفاوت حمایت می کند بلکه همچنین توسعه و نگهداری این منطق را سهل و آسان می نماید.

۶-۱- پی آمد توسعه Business logic

تجزیه و تحلیل، طراحی، توسعه و نگهداری Business Rule ها یکی از عملیات پیچیده و مشکل در توسعه سیستمها محسوب می شود که دقت در این عمل موجب حصول نتایج خوبی خواهد شد. لحاظ کردن این موضوع دربرگیرنده پیامدهای زیر است.

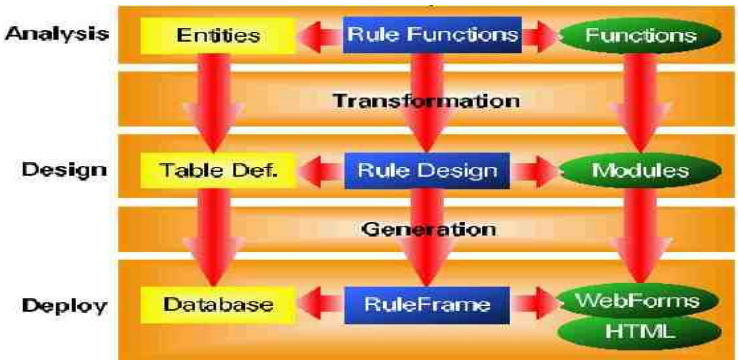
- Business Rule در موجودیت های مستقل مخفی سازی شده یا توابع Business تشریح داده می‌شوند که این موضوع به کرات به صورت عبارات ابهام آمیز انجام می شود.

www.hajarian.com

- برآورد تخمین فاز design مشکل است زیرا به صورت واضح مشخص نیست که نیاز به توسعه چه چیزی داریم. با در نظر گرفتن این موضوع این تخمین و برآورد به صورت شفاف مشخص خواهد شد.
- قواعد مختلط و پیچیده درگیر موجودیت های مستقل متفاوت زیادی شده و به دنبال زیاد شدن آنها تنها بخشهایی از آن توسعه می یابند. با در نظر گرفتن این منطق و پیاده سازی آنها به صورت سیستماتیک این توسعه با مشکلات کمتری مقدور خواهد شد.
- اگر یک قانون تنها در یک سیستم نهایی در نظر گرفته شده و توسعه یابد ممکن است قدرت سرویس دهنده در صورت اضافه شدن این قانون به ماشین نهایی دیگر کاهش یابد. با بکارگیری این منطق این فقدان تا حدی بهبود یابد.
- تمامیت سیستم قابل بازبینی و تحقیق نبوده زیرا در اینها مکانیزم تائیده شده و صریحی برای برگشت به عقب در اسناد تجزیه و تحلیل وجود ندارد اکنون ما می‌توانیم تصور کنیم که نگهداری قواعد Business خود به چه مفهوم و معنایی است. این خواست با احتمال زیاد موجب افزایش هزینه ها در توسعه و تغییرات خواهد شد و مطمئناً زمان زیادی برای ارائه این محصول در بازار تلف خواهد شد. بر این اساس است که یک چهارچوب قدرتمند برای تجزیه و تحلیل، طراحی و توسعه Business logic به صورت شفاف مورد نیاز است.

۶-۲- چهارچوب CDM Ruleframe

CDM Ruleframe در حقیقت یک چهارچوب مناسب جهت تجزیه و تحلیل، طراحی و توسعه Business Rule آماده می کند. این چهارچوب در حقیقت توضیح دهندهٔ این موضوع خواهد بود که Business Rul ها چگونه در Oracle Designer ثبت و نگهداری می شوند. روابط موجود بین لایه داده ها و لایه ارائه بسیار واضح و شفاف معین شده که این امر موجب راحتی و یکپارچگی در تجزیه و تحلیل خواهد شد.



همانگونه که در شکل قبل مشاهده می شود بعد از ثبت Business Rule از فاز تجزیه و تحلیل، CDM Ruleframe این قواعد را برای تعاریف موجود در فاز طراحی تغییر شکل داده و آنها را آمادهٔ استفاده در این فاز می کند. دقیقاً شبیه

اینکه intity, oracle designer ها را به صورت جداول تغییر شکل داده و توابع business به module تبدیل می شوند.

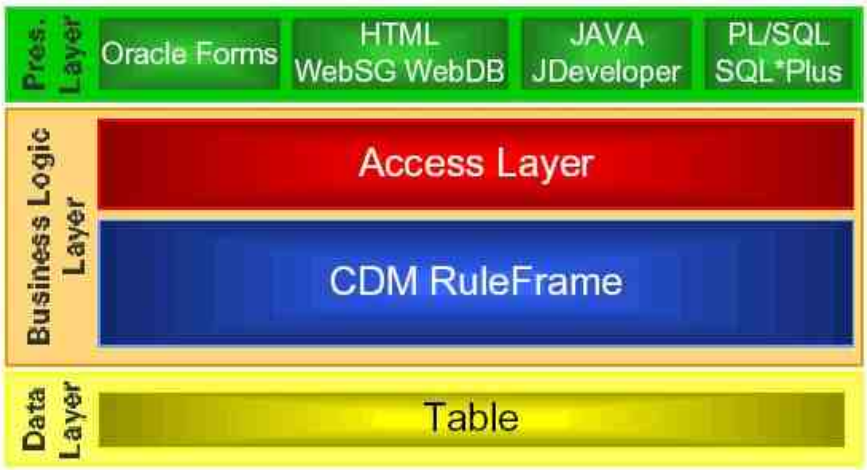
سپس CDM Ruleframe به صورت اتوماتیک تولید کد معتبر را برای این Rule مورد نظر انجام خواهد داد. برای دیگر قواعد موجود نیز سیستم به صورت ساده کد مورد نظر rule ها را تولید کرده که این امر موجب کاهش قابل ملاحظه در کد برنامه ها خواهد شد. هنگامیکه تمامی قواعد به صورت کد درآمد ما می توانیم این کدها را در بسته (package) های مختلف به صورت ساختار یافته ای در لایه Business logic آرایش دهیم.

۶-۳- آرایش معماری CDM Ruleframe

معماری CDM Ruleframe بر روی یک تکنولوژی ثابت شده‌ای مبتنی است. این مورد در سرویس دهندهٔ پایگاه داده oracle ۸i با استفاده از PL/SQL گسترش یافت. این امر مزیت انکار نشدنی با خود به همراه دارد. مواردی همچون امنیت، مقیاس پذیری و قابلیت استفاده مجدد.

دستیابی عمومی به این لایه مبتنی بر تکنولوژی پایه ای java ممکن خواهد بود (مواردی مثل قبیل jsp,Servelts) که یک شیوهٔ قدرتمند برای دستیابی و عملیات بر روی CDM Ruleframe است.

Ruleframe در ماشین‌های نهایی را می توان با کمک کیت ابزار توسعه Oracle (همچنین , PL/SQL jDeveloper,web DB, oracle forms) تولید نمود. در ادامه نمای شماتیکی از منطق معماری چند لایه‌ای که جایگاه CDM Ruleframe را نشان خواهد می‌دهد ارائه گردیده است.



۶-۴- مزایای استفاده از CDM Ruleframe

تعدادی از مهمترین مزایای بکارگیری CDM Ruleframe در ادامه اشاره خواهد شد.

۱. سرعت و سهولت توسعه business Rulesها

www.hajarian.com

۲. استفاده از برنامه های سودمند در جهت تبدیل اطلاعات تحلیلی به عناصر محیط طراحی
۳. استفاده از برنامه های سودمند جهت تولید business Rule کامل به همراه کدهای معتبر برخی خصوصیات این قواعد.
۴. استفاده از برنامه های سودمند جهت تولید و آرایش بسته ها، business Logic برای هر جدول.
۵. توانایی کنترل، هدایت و توسعه به صورت اتوماتیک.

این موارد در ادامه بیشتر توضیح داده خواهند شد.

۱. سرعت و سهولت در نگهداری business Rule ها

CDM Ruleframe سرعت و سهولت نگهداری قواعد و قوانین business را بوسیله موارد ذیل تامین می‌کند:

- مکان یابی قواعد و توانایی جداسازی انواع عوامل محیطی در Oracle disigner .
- استفاده از اصول OO نظیر مخفی سازی اطلاعات و encapsulation .

Ruleframe در واقع توسط لایه business Rules لایه داده ها را encapsulate می کند و پیچیدگی ساختاری داده ها را از دید توابع و روالهایی که از این داده ها استفاده می کنند مخفی می‌سازد.

ـ استفاده از برنامه های سودمند CDM Ruleframe یکسان جهت ایجاد هر چه بهتر نگهداری و حفاظت business Rules .

۲. توانایی ردیابی business Rule ها

به کمک ثبت قواعد و قوانین business و جداسازی اجزاء آن در خلال تجزیه و تحلیل و طراحی می توان از این اجزاء شبیه دیگر اجزا Oracle Designer استفاده کرد. این ارتباطات به ما خواهند گفت چه قواعد و قوانینی در اسناد تجزیه و تحلیل مورد تائید واقع شده و توسعه یافته است. در هنگام تست و پذیرش پروژه این اطلاعات از ارزش بالایی برخوردار خواهد بود و در جهت تامین تمامیت و کمال پروژه ما را یاری خواهد کرد.

۳. اجرا و انجام تراکنشها به صورت یکدست و یکپارچه

ایجاد یکپارچگی تراکنشها یکی از کلیدهای پذیرش CDM Ruleframe است.

ـ تصویب امکان قرارگیری قواعد و قوانین در سمت سرویس دهنده (که قبلاً فقط این قواعد می توانست در سمت client توسعه یابد. شبیه ارتباطات master/detail الزامی و اجباری)

ـ تامین انعطاف پذیری زیاد در سمت استفاده کننده های نهایی (end user) با اعطای آزادی و استقلال بیشتر از زمانیکه قواعد business مجبور به اجرا و قرارگیری در سمت انتهای یک تراکنش شد.

ـ گزارش برگشتی تمامی مشکلات به صورت یکباره و یکپارچه به مصرف کنندهٔ نهایی (end user) که موجب کاهش RTT شبکه شده و باز خور و عکس العمل استفاده کننده را بهینه می سازد.

www.hajarian.com

۴. استقلال و آزادی نرم افزارهای نهایی در استفاده مجدد کردن از business Logic

استقلال ماشین های نهایی در استفاده از business Logic به این معنی است که نحوه استفاده از این منطق در ماشین‌های نهایی در کل سیستم اهمیتی ندارد. اما این نکته را نباید فراموش کرد که خود business Logic و داده های آن نبایستی دچار تغییرات شوند. زیرا اصل منطق بایستی توسط دیگر ماشین ها استفاده شود.

استفاده چندگانه ماشین‌های نهایی از این منطق کاملاً عمومی است. به عنوان مثال یک سند html ساده پایه برنامه های web based می باشد و به عنوان مثال form ها در oracle به عنوان یک UI قدرتمند شناخته می شوند. حال اینکه این دو همزمان می توانند از business Logic، بسته به شرایط خود استفاده کنند. CDM Ruleframe نیز به طور کامل از استفاده مجدد کردن business Logic حمایت می کند. این حمایت در واقع از آنجا نشات می گیرد که قواعد در یک لایه میانی مجزا توسعه یافته اند.

۵. آزادی و عدم وابستگی به لایه داده ها

عدم وابستگی به لایه داده ها به این معنی است که ما قادر خواهیم بود منابع دادهٔ خود را در لایه داده ها و زیر لایه های آن و همچنین قواعد حاکم بر این داده ها بدون اعمال تغییرات در ماشین‌های نهایی انجام دهیم. به عنوان مثال هنگامیکه ما سیستم بخشهای مختلف را با یکدیگر جمع کرده و در یک سیستم واحد در نظر می‌گیریم، باید قدرت ادغام منابع داده‌ای با یکدیگر را داشته باشیم. با این کار و استفاده از این وجه مشترک یکسان هم توسعه را در ماشینهای نهایی بوجود آورده و هم توسعهٔ سیستمها و نرم افزارهای نهایی را با زمان آموزش کم تثبیت می کنیم.

بدلیل وجود این مزیتها و نکات مثبتی که در طراحی CDM Rule frame مشاهده شد oracle نیز سعی در بکارگیری و لحاظ کردن این مورد نموده البته Component های جداگانه ای برای CDM Rule frame وجود ندارد اما توابع، رویه ها و متدهای مختلف موجود در Oracle در جهت توسعه سیستمها CDM Rule frame در نظر گرفته شده است و ارائه گردیده. البته web assistant های موجود درoracle designer به صورت کامل از CDM Rule frame حمایت می کند. امکانات و موارد بسیار خوب و عالی در زمینه تعاریف business Rule گنجانده شده است که امروز مورد استفاده تحلیگران، طراحان و توسعه دهندگان سیستمها قرار گرفته است.

۶-۴- چگونگی ذخیره سازی تعاریف طراحی business Rule در oracle designer

oracle designer فاقد ساختار اشیاء جهت ثبت business Rule در هنگام طراحی و پیاده سازی یک پروژه است. اما برخی از business Rule را می توان توسط ترکیب و وابسته کردن مدل منطقی داده ها به صورت ساختار یافته ثبت کرد مانند:

- خواص تعریف یک ستون:

نوع داده ـ ماکزیمم طول ـ شرایط ـ حساس به بزرگ بودن کارکترها ـ مقادیر پیش فرض

❖ کلیدهای اصلی، یکتا و خارجی:

با استفاده از قالب و اعمال فشار توسط این کلیدها می توان این قواعد را معرفی کرد.

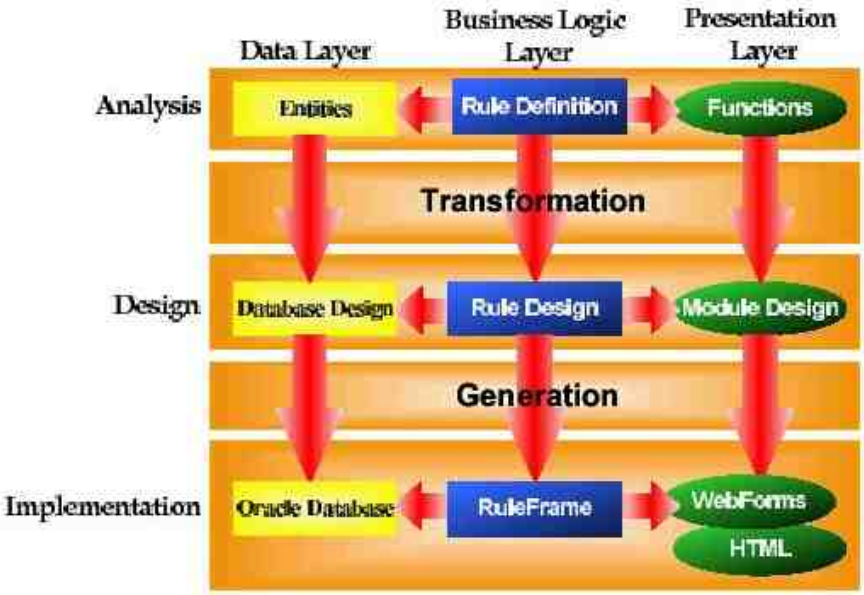
برای تمامی business Rule های دیگر با استفاده از CDM Ruleframe می توانیم این تعاریف و قواعد را ثبت کنیم. این کار به کمک اشیاء راه انداز (Tigger Object) و تعاریف PL/SQL مربوطه به صورت خواهد گرفت.

جهت نامگذاری و تعیین هویت قواعد و قوانین می توان از قانون نامگذاری به صورت زیر پیروی کرد.

BR-<Table Alias> # # # - <Rule Class>

برای مثال BR-ENT ۰۰۱ - ATT می تواند یک نام برای business Ruleباشد.

ضمن اینکه Oracle designer یک سری برنامه های سودمند را در جهت تعریف و طراحی business Rule فراهم کرده است که در شکل زیر لایه های پردازش در توسعه لایه business Logic به نمایش در آمده است.

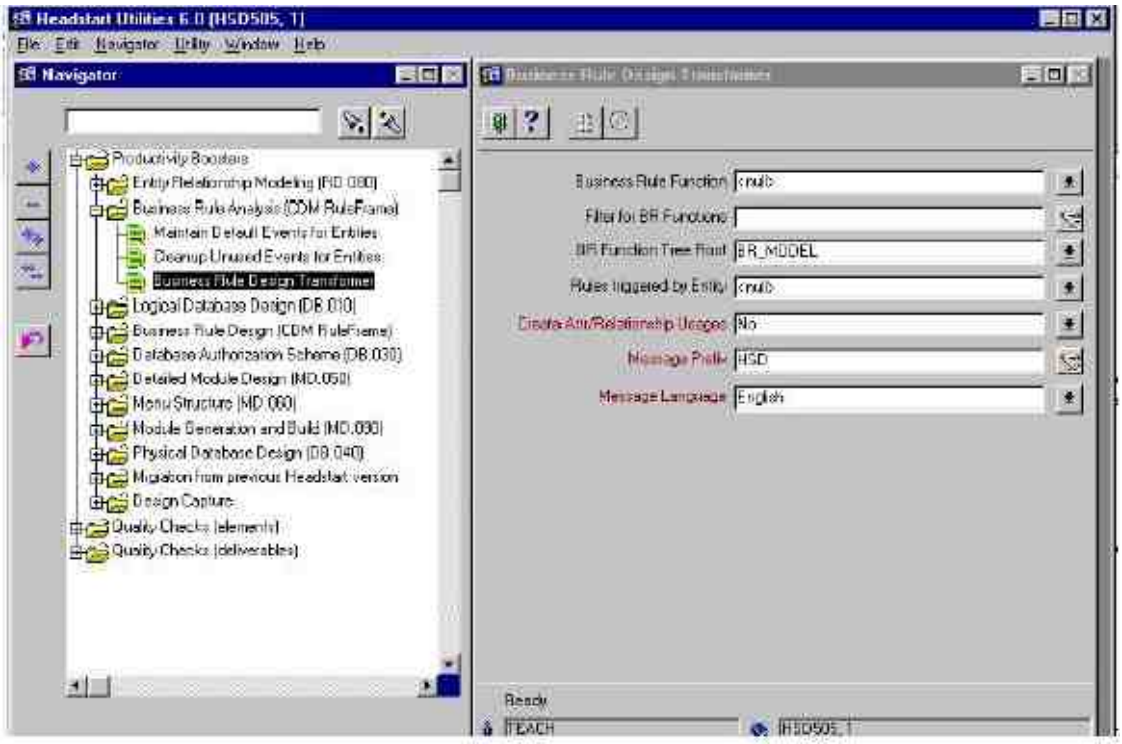


۶-۵- مبدل business Rule

بهترین راه برای ساخت و طراحی business Rule استفاده از محیط Transformer است. این محیط جهت تعریف قواعد و قوانین موجود در فاز تحلیل پروژه در اختیار استفاده کنندگان قرار گرفته و این موضوع موجب افزایش انگیزه در جهت فرموله کردن و مدل کردن business Rule ها در این محیط گردیده است. ما با استفاده از این Transformer می توانیم قواعد و قوانین خود را سطح بندی کرده و مورد استفاده قرار دهیم. عملیات در این محیط بگونه‌ای است که توقف و استفاده از قواعد و قوانین به صورت سطح بندی شده ممکن خواهد بود. هنگامیکه مبدل اجرا شود گزینه های ساده و توابع ریشه ای جهت ساخت توابع مربوط به business Rule توسط یک دیاگرام سلسله مراتبی در اختیار قرار خواهد گرفت. این گزینه ها ما را در ساخت قواعد و قوانین، توابع مربوط به آن و داده های مورد استفاده این توابع یاری خواهند کرد.

www.hajarian.com

به علاوه در این محیط می‌توان با بکارگیری آرگومان ها و ماژولهای راه انداز PL/SQL تمامیbusiness function خود را معرفی و ثبت کنیم. این محیط به صورت اتوماتیک پیامهای error مربوطه را در دایرکتوری پیامها به صورت متنی تولید می کند. rule type نیز در این محیط قابل تعریف بوده و در نهایت پس از ساخت و تعریف rule می‌توان با باز کردن ویرایشگر بلاک PL/SQL برای هر business Rule مربوط به آن را کد کرده و راه انداز های مربوط به این قانون را پیاده سازی نمائیم . شکل زیر نمایی از محیط مبدل مورد نظر می‌باشد.

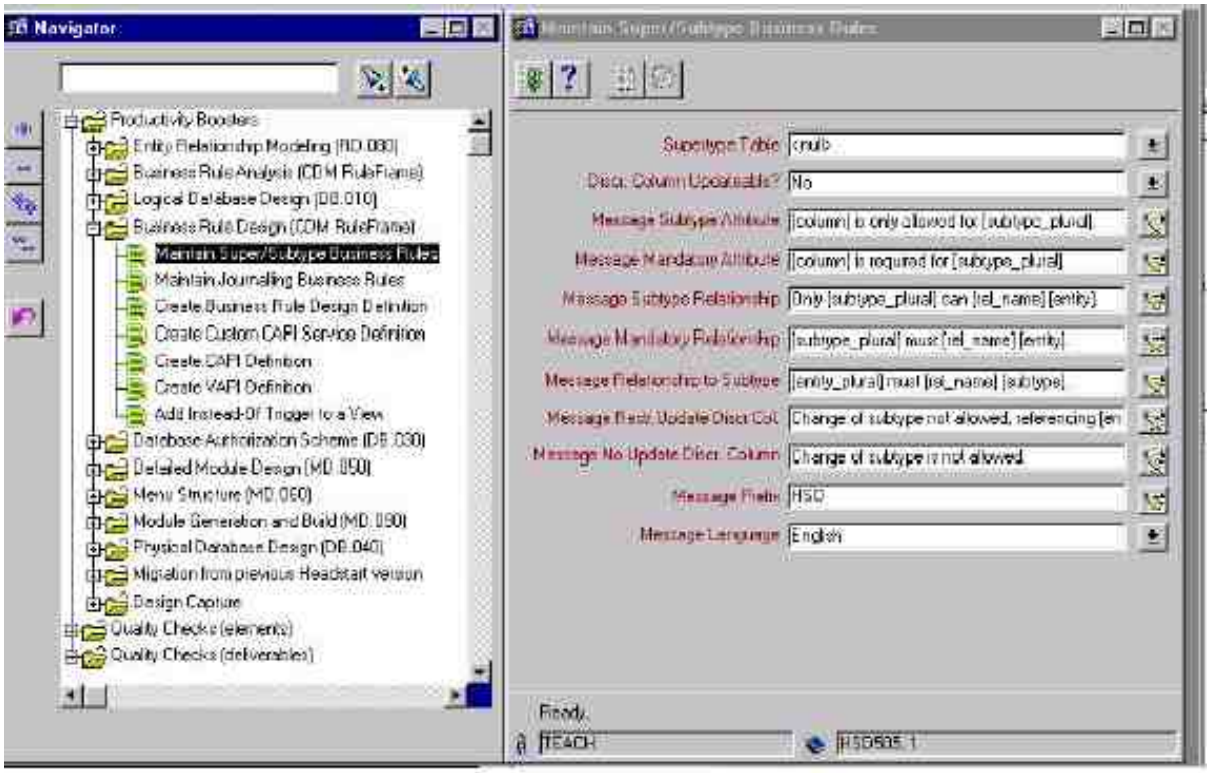


۶-۶- تولید کننده استاندارد business Rule

در اینجا یک کلاس مخصوص از business Rule معرفی خواهند شد که صد درصد این قواعد با استفاده از CDM Ruleframe utility قابل تولید هستند. آنها را business Rule های استاندارد می نامیم. بطور معمول اینجا دو مجموعه از business Rule ها تولید خواهند شد. اولین مجموعه مجموعه‌ای است که با یک جدول و با هویت supertype/subtype توسعه می یابند و دومین گروه به صورت ثبت روابط موجود و به صورت روزمره توسعه می یابند. این utility نه تنها قادر به تعریف مشخصات و طراحی business Rule هاست بلکه کدهای واقعی را نیز به نحو عالی تولید خواهد کرد.

شکل زیر نشان دهنده یک business Rule , supertype/subtype است.

www.hajarian.com



خلق rule ها به صورت دستی و خارج از محیط transformer نیز ممکن است. ما با استفاده از کد PL/SQL به صورت بسیار سریع و عالی قادر به خلق business Rule با تمامی اجزاء کلیدی آن خواهیم بود.

۶-۷- کد کردن business Rule

شکل زیر نمونه ای از یک تابع معتبر business Rule است که توسط CDM Ruleframe transformer تولید شده است

Generated

Hand crafted

Generated

```
/*****************************************************************
Purpose      Employee Salary must be a multiple of 100.
Remarks

Revision History
When         Who
 Revision    What
-----------------------------------------------------------------
22-02-2000   IDEVCOE
 1.0         Using utility HSU_CRER (revision 6.0.0.9)
*****************************************************************/
   l_rule_ok boolean := true;
begin

   l_rule_ok := mod(p_salary,100) = 0;

   return l_rule_ok;
exception
   when others
   then
      gms$errors.unhandled_exception(package_name||' hr_emp001_att (f)');
end hr_emp001_att;
```


www.hajarian.com

با استفاده از CDM Ruleframe پیاده سازی کدهای واقعی برای قوانین و قواعد business به سادگی و با سرعت زیاد مقدور خواهد بود. تنها زمانیکه می‌خواهیم قوانین پیچیده را کد کنیم ارزش این بخش واضح تر خواهد بود. این سادگی توسط سرویسهای استاندارد CAPI مقدور شده است. این سرویس ها شامل مجموعه ای از توابع و رویدادها خواهد بود که به صورت اتوماتیک در هر CAPI برای آن business Rule به صورت استاندارد تولید خواهد شد. در ادامه چند نمونه از این سرویس‌های استاندارد را بررسی می‌نمائیم.

- num-value/char/get-date

این توابع مقدار موجود در ستون معین و سطر معین داده شده از یک CAPI را بازیابی می کنند.

- Exists – row

این توابع با انجام بازیافت شرایط و ضوابط به صورت ورودی، در صورتیکه یک ردیف موجود با این ضوابط و شرایط سازگار باشد، مقدار true را برمی‌گرداند.

- display-lable

این توابع با دریافت یک کلید اصلی به صورت پارامتر در ورودی، یک برچسب مناسب نمایش را برمی‌گرداند.

- aggreagate - value

این توابع انجام دهندهٔ محاسباتی از قبیل Connt, max, min , . . . روی مجموعه‌ای از ردیفهایی است که در شرایط و ضوابط درخواست پرس وجو مشخص شده اند.

در انتهای این بخش ذکر این نکته ضروری به نظر می رسد که می توان در صورت نیاز سرویس های مورد نظر خود را به صورت یک سرویس استاندارد CAPIپیاده سازی کنیم.

در ادامه با ارائه کد چندین قانون و قاعده مختلف خواهیم پرداخت. هر rule مثل یک تابع boolean پیاده سازی شده است که در صورتیکه این تابع مقدار true را برگرداند نشان دهنده پذیرفته شدن rule خواهد بود و در صورتیکه false برگرداند بیان کنندهٔ نقض قانون و قاعده می باشد. در هر کدام از این قوانین از یک متغیر منطقی محلی بنام -rule-۱ ok استفاده می شود که در واقع نگهدارنده مقدار برگشتی خواهد بود. کالبد و استخوان بندی این توابع شامل آرگومانهای مورد نیازی است که در ورودی ذکر شده و در حقیقت قسمتی از تعاریف business Rule محسوب می شوند.

- Attribute Rule

حقوق یک کارمند بایستی مضرب صحیح از ۱۰۰ باشد.

- typle Rule

هر کارمند با ماموریت فروشندگی بایستی دارای مقدار کارمزد و حق العمل باشد.

- inter – entity Rule

زمان آغاز واگذاری پروژه بایستی بزرگتر یا مساوی با زمان شروع پروژه باشد.

- entity rule

در هر شعبه بیش از یک منشی وجود ندارد.

- update Rule

شما می توانید تاریخ یک سفارش را تغییر داده و به روزسازید.

- transition Rule

تغییرات و انتقال در حالت civil (وضعیت مدنی) یک کارمند به صورت زیر مجاز محسوب می شود.

حالت تنهایی (مجرد) به حالت ازدواج (تاهل)

حالت ازدواج به حالت طلاق

حالت ازدواج به حالت بیوه

حالت طلاق به حالت ازدواج

حالت بیوه به حالت ازدواج

- Relationship Rule

هر قلم سفارش بایستی متعلق به یک و تنها یک سفارش باشد و کوچکترین واحد در هر سفارش یک قلم سفارش خواهد بود.

ابتدا در business Rule اول وجود قلم داده شده ، چک شده و در rule بعدی امکان حذف قلم داده از یک سفارش با وجود شرایط خواسته شده صادر خواهد شد.

- inter- entity Rule

در مورد انتصاب چندین وظیفه و عمل در پروژه به یک کارمند خاص بایستی زمان انتصاب این عملیاتها روی هم افتادگی یا تداخل (overlap) نداشته باشد. این rule با استفاده از یک تابع جداگانه و فراخوانی آن صورت خواهد گرفت.

- change even with DML

اگر تاریخ پایان واگذاری پروژه بزرگتر یا مساوی با زمان پایان پروژه بود بایستی زمان پایان پروژه را مساوی با تاریخ پایان واگذاری پروژه جدید تنظیم کرد.

- change even without DML

زمانیکه تاریخ پایان واگذاری پروژه تغییر کرد یک email جهت اطلاع و تذکر به مدیریت کارمندان زده شود.

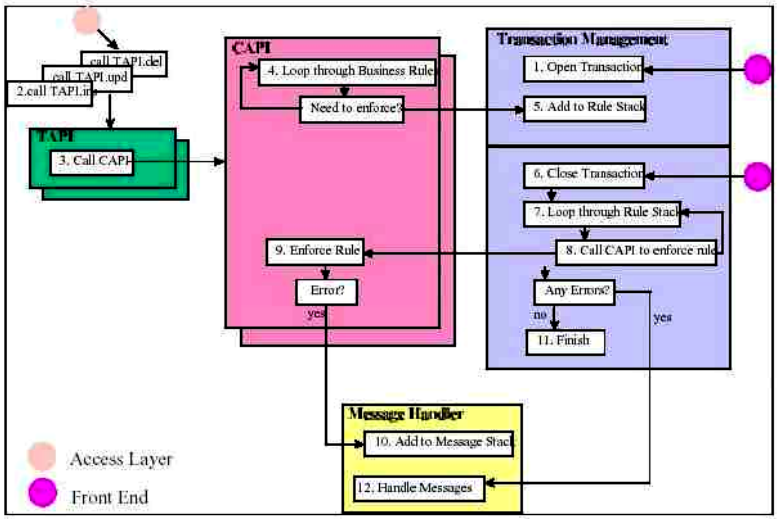
حال پس از آنکه business rule ها ساخته شد و تعریف و طراحی آنها به پایان رسیده و کدهای مورد نظر اختصاص یافت، بایستی به تولید لایه business logic در پایگاه داده بپردازیم .

ابتدا Headstart utility را اجرا کرده و برای تمامی جداول گزینه create CAPI Definition را انتخاب می کنیم .

این گزینه تمامی business Rule ها را به همراه custom service آنها ارائه کرده و بسته های CAPIرا برای هر جدول می سازد.این بسته ها در نهایت پس از ساخت و تعریف در مخزن Oracle Designer جا داده شده و نگهداری می شوند.

۶-۸- چگونگی اجرا و عملیاتهای رایج business Rule در زیر ساخت Oracle Designer

سرانجام پس از معرفی و تشریح موارد ذکر شده اینک در ادامه به توضیح پردازشهای پایه‌ای و بنیادی و جریان این اطلاعات را در لایه business logic مورد بررسی قرار می دهیم. شکل زیر جهت جریان اطلاعات را در این لایه به نمایش در می آورد.



حال به توضیح مختصر در مورد هر یک از عملیاتهای متداول در این لایه می پردازیم.

۱. یک ماشین نهایی در این مرحله یک تراکنش را با فراخوانی یک TAPI در این لایه باز می کند.
۲. این TAPI در صدد اجرای یک business Rule برآمده و آن را بصورت DML سازماندهی می کند.
۳. این TAPI یک CAPI را فراخوانی می کند.
۴. CAPI با یک چرخشی و بررسی برای هر business Rule موارد مورد نیاز برروی آن را مشخص می کند.
۵. در صورتیکه اجرای مرحله ها مواردی را جهت اجرای این قانون مورد نیاز دانست آن rule را روی پشته rule های موجود در بسته مدیریت تراکنشها قرار خواهد داد (مرحله ۲تا ۵ برای هر ردیف موجود در transaction تکرار می کند)
۶. ماشین نهایی تراکنش را می بندد.
۷. طرز کار تراکنش به صورت چرخش و تحقیق در پشته rule ها در میان آن ها انجام خواهد شد.
۸. برای هر rule موجود در پشته یک CAPI اختصاص داده و فراخوانی می کند.
۹. جهت توسعه کد هر قانون در CAPI بایستی اطلاعات مورد نظر جهت اجرای آن rule را بدست آورد. برای حصول به این نتایج بایستی از سرویس های CAPI استاندارد سود برد.

www.hajarian.com

۱۰. اگر مشکلی (error) پیدا شد، پیام مشکل سریعاً روی پشته پیامها قرار خواهد شد .

۱۱. اگر هیچ مشکلی در تمام تراکنش پیش نیامده باشد. کنترل به محل فراخوانی Application برگشته که با توجه به تراکنش آن را اجرا و به کارمی برد.

۱۲. اگر یک یا چندین error پیش آمد. مدیریت تراکنش این مشکلات را در Application به صورت error بازسازی می کند. ماشینهای نهایی جهت نمایش پیامهای error بایستی پرس و جوهای خود را به message handler ارسال و پاسخها را دریافت کند.

منابع و مراجع

۱- Oracle Corporation, Custom Development Method Classic (CDM Classic) Method Handbook, Oracle Corporation, Release ۲,۶,۰, Feb ۲۰۰۰.

۲- Oracle Corporation, Project Managemet Method Handbook, Oracle Corporation, Release ۲,۵,۰, Feb ۲۰۰۰.

۳- کاربرد UML(تحلیل و طراحی شی گرا)/ نویسنده: ریچارد نالس ترجمه: حسن حسینیان، ابوالفضل لاکدشتی (انتشارات علوم رایانه بابل)

۴- Rational Unified Process ۲۰۰۰