



معرفی سکوی نرم افزاری HyperNet Platform

ویرایش ۱۰



شرکت مهندسی داده پردازی سما
تلفن: ۶۶۵۲۲۱۷۸ - ۶۶۵۰۹۱۲۵
<http://www.dpsama.com>

اردیبهشت ماه ۱۳۸۴

فهرست مطالب

۳	مقدمه
۴	مزایای استفاده از HyperNet Platform
۵	اجزای HyperNet Platform
۶	چارچوب کاری
۷	معماری سیستم
۸	کتابخانه کلاسها
۸	کلاسهای کاربردی عمومی
۸	کلاسهای پایه ای معماری سیستم
۹	کلاسها و کنترلرهای Windows
۱۱	سیستم‌های عمومی
۱۱	سیستم امنیت
۱۲	سیستم گردش کار
۱۳	سیستم ورود اطلاعات Metadata
۱۴	استانداردها
۱۴	استانداردهای برنامه نویسی
۱۴	استانداردهای رابط کاربر
۱۵	استاندارد طراحی پایگاه داده ها
۱۵	استانداردهای چرخه تولید نرم‌افزار
۱۶	ابزارها
۱۶	پروژه ساز HyperNet
۱۷	کدساز HyperNet

مقدمه

در تهیه نرم‌افزارهای مختلف به مسائل مشترک زیادی برخورد می‌کنیم. این مشترکات بین نرم‌افزارهای یک حیطة خاص بسیار زیاد و در خور توجه است. برای مثال، نرم‌افزارهای مبتنی بر بانک‌های اطلاعاتی چه از نظر معماری و چه از نظر استانداردهای مورد استفاده، مشابهت بسیار زیادی دارند. از این رو تولیدکنندگان نرم‌افزار برای رسیدن به یک استاندارد واحد بین برنامه‌های کاربردی خود، ارائه کیفیت بالاتر و قابلیت اطمینان بیشتر و نیز برای کاهش زمان تولید برنامه‌های کاربردی، این مشترکات را در قالب کتابخانه کلاسها و توابع، چارچوب کاری، ابزارها و استانداردها بطور جداگانه و کلی مورد توجه قرار می‌دهند. ما همین دیدگاه را در سطحی فراتر مورد توجه قرار داده و نسبت به تولید یک سکوی نرم‌افزاری برای تهیه برنامه‌های کاربردی اقدام کرده‌ایم.

مستند حاضر مجموعه ایست حاوی کلیات، سرفصل‌ها و عناوین مربوط به امکانات چارچوب کاری، استانداردها و ابزارهای نرم‌افزاری که در چرخه تولید و اجرای سیستم‌های نرم‌افزاری مبتنی بر بانک‌های اطلاعاتی با اندازه فراتر از متوسط، لازم و ضروری خواهد بود. این مجموعه که تحت عنوان HyperNet Platform عرضه می‌گردد یک راه‌حل جامع برای تولید نرم‌افزارهای مدرن مبتنی بر بانک اطلاعاتی می‌باشد و بر اساس فناوری Net^۱ و الگوهای^۱ شناخته شده تهیه گردیده است. سکوی HyperNet یک راه‌حل مطمئن برای تولید کنندگان نرم‌افزار است که علاوه بر مرتفع نمودن مسائل بسیار پیچیده و زیاد تولید نرم‌افزار، ارتقای دانش فنی در واحدهای تولید نرم‌افزار را نیز به همراه خواهد داشت.

^۱ Pattern

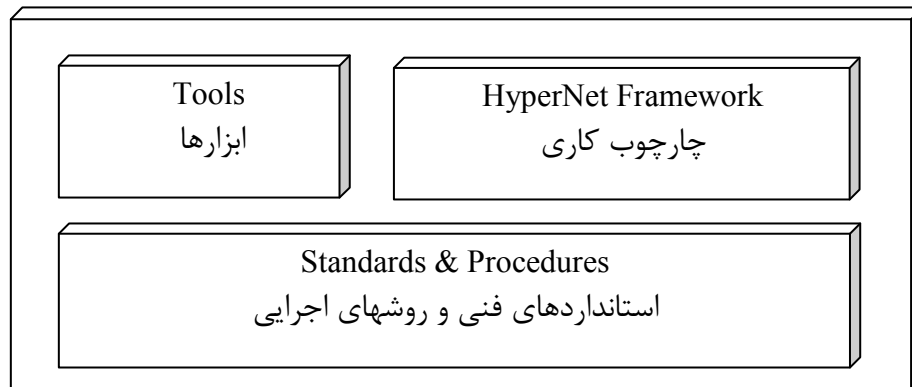
مزایای استفاده از HyperNet Platform

ما در اینجا برخی مزایای استفاده از این سکو در حیطه تولید نرم افزار را بیان می کنیم:

- ✓ داشتن یک سکوی مطمئن، استاندارد و یکسان برای اجرای پروژه های نرم افزاری.
- ✓ تضمین کیفیت بالای نرم افزارهای تهیه شده
- ✓ کاهش ۳۰ تا ۵۰ درصد از زمان توسعه و تولید نرم افزار.
- ✓ کاهش درصد قابل توجهی از هزینه کل اجرای پروژه.
- ✓ تبعیت رابط کاربر از یک استاندارد مشخص و راحتی کاربر در یادگیری سیستم‌های مشابه
- ✓ یکسان سازی معماری سیستم‌های نرم‌افزاری تولید شده بر اساس این مرجع
- ✓ تسهیل در برقراری ارتباط بین سیستم‌های تهیه شده بر این اساس
- ✓ افزایش سرعت در تحویل سیستم های نرم افزاری
- ✓ تولید سریع و اتوماتیک Prototype سیستم از روی مدل تحلیل
- ✓ کاهش هزینه نگهداری و پشتیبانی سیستم های نرم افزاری.
- ✓ کاهش ریسک موانع مشکل آفرین و حل نشده در فاز تولید نرم افزار.
- ✓ ایجاد ارزش افزوده و افزایش توان و بضاعت شرکت در زمینه تولید و ساخت نرم افزار.
- ✓ مدیریت سهل و آسان پروژه در بخش تولید نرم افزار.
- ✓ جلب نظر و رضایت مشتریان و کارفرمایان.
- ✓ جلوگیری از دوباره کاریهای سربار هر پروژه.
- ✓ مطابقت با نیازها و استانداردهای ایزو.

اجزای HyperNet Platform

شکل ۱ ساختار اجزای HyperNet Platform را نشان می‌دهد. چنانچه در شکل دیده می‌شود سکوی HyperNet فقط شامل نرم‌افزار نبوده و استانداردها و روشهای اجرایی نیز جزئی از این سکوی نرم‌افزاری محسوب می‌شود.



شکل ۱ - ساختار اجزای سکوی HyperNet

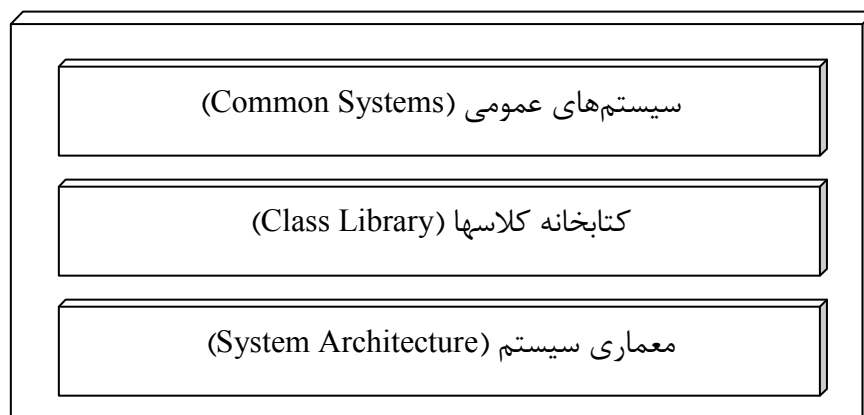
حال به شرح هرکدام از اجزای تشکیل دهنده سکوی HyperNet می‌پردازیم:

چارچوب کاری^۱

چارچوب کاری HyperNet مجموعه‌ایست شامل :

- ✓ معماری سیستم که تعیین کننده ساختار کلی بوده و سایر اجزای تشکیل دهنده چارچوب یعنی کتابخانه کلاسها و سیستم‌های عمومی بر مبنای آن تهیه شده‌اند.
- ✓ کتابخانه کلاسها که شامل کلاسهایی است که در اکثر سیستم‌های کاربردی مورد نیاز می‌باشد و امکانات مختلفی از کار با تاریخ شمسی گرفته تا مدیریت خطاها و پیغامها^۲ و مدل شیء گرای SQL را شامل می‌شود.
- ✓ سیستم‌های عمومی که شامل سیستم‌های مورد نیاز در اکثر برنامه‌های کاربردی مانند سیستم امنیت^۳ و سیستم گردش کار^۴ می‌باشد. این سیستم‌ها با استفاده از امکانات سکوی HyperNet و بر پایه معماری سیستم و کتابخانه کلاسهای چارچوب کاری تهیه شده است و به عنوان نمونه‌ای عملی از کاربرد سکوی HyperNet می‌تواند مد نظر قرار گیرد.

شکل ۲ اجزای تشکیل دهنده چارچوب کاری HyperNet را نشان می‌دهد.



شکل ۲ - اجزای تشکیل دهنده چارچوب کاری HyperNet

حال هر یک از اجزای تشکیل دهنده چارچوب کاری را به تفصیل مورد بررسی قرار می‌دهیم

^۱ Framework

^۲ Exception & Message Handling

^۳ Security

^۴ Workflow

معماری سیستم

انتخاب معماری مناسب برای تهیه یک سیستم نرم‌افزاری از اهمیت ویژه‌ای برخوردار است. این اهمیت پس از تولید نرم‌افزار و در زمان پشتیبانی بیشتر آشکار می‌گردد. زمانی که کاربر بیشتر به نیازهای خود آگاهی پیدا کرده و آنها را مطرح می‌کند. در چنین شرایطی اگر این نیازها در معماری نرم‌افزار لحاظ نشده باشد ممکن است نرم‌افزار موجود جهت برآورده کردن آنها، نیاز به تغییر ساختار اساسی داشته باشد. بنابراین واحدهای تولید برای تهیه نرم‌افزار باید آینده‌نگریهای لازم را داشته باشند و عدم آگاهی کافی کاربران را از نیازهای خود مد نظر داشته باشند.

یکی از مسائلی که با گسترش اینترنت و سیستم‌های تحت وب مورد توجه قرار گرفته است مسئله مقیاس پذیری^۱ برنامه‌ها می‌باشد. افزایش یکباره تعداد استفاده‌کنندگان یک سایت موجب غافلگیر شدن سرویس دهندگان می‌شود که در صورتیکه قبلاً به مسئله مقیاس‌پذیری توجه نشده باشد موجب می‌شود سیستم در روند اجرایی شدن، دچار مشکل اساسی شود. یکی از روشهای معمول برای تهیه یک سیستم مقیاس پذیر استفاده از معماری چند لایه^۲ می‌باشد.

مسائل متعدد دیگری در سطح معماری سیستم پیش می‌آید که هر یک راه حل مناسب خود را می‌طلبد. یکسری از این مسائل و راه‌حل‌های مناسب آنها در سطح کلی مورد توجه قرار گرفته و تحت عنوان الگوهای طراحی^۳ یا الگوهای معماری^۴ مطرح می‌شوند. در تهیه HyperNet Platform از این الگوها بهره گرفته و یک معماری مناسب برای اکثر سیستم‌های اطلاعاتی را پایه کار خود قرار داده‌ایم. در انتخاب معماری مسائلی از قبیل مقیاس‌پذیری و استفاده مجدد^۵ مد نظر قرار گرفته است.

^۱ Scalability

^۲ N-tier

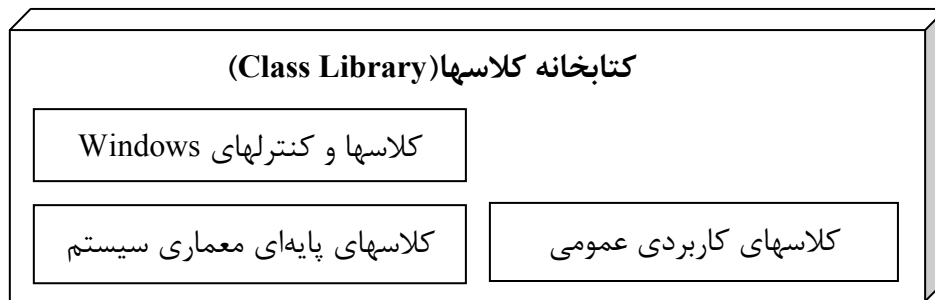
^۳ Design Patterns

^۴ Architecture Patterns

^۵ Reusability

کتابخانه کلاسها

چارچوب HyperNet مشتمل بر کتابخانه‌ای از کلاسهاست که جوابگوی نیازهای عمومی برای تهیه سیستم‌های اطلاعاتی می‌باشد. این کتابخانه کلاسها بر پایه معماری فوق‌الذکر تهیه گردیده است. شکل ۳ اجزای تشکیل دهنده آنرا نشان می‌دهد:



شکل ۳ - اجزای تشکیل دهنده کتابخانه کلاسها

حال به معرفی اجزای کتابخانه کلاسها می‌پردازیم:

کلاسهای کاربردی عمومی

یکسری کلاسها در مجموعه کتابخانه کلاسهای HyperNet وابسته به ساختار خاصی نبوده و بطور عمومی در هر سیستمی قابل استفاده هستند. این کلاسها شامل مواردی مثل کار با تاریخ شمسی، مدت، زمان، اعداد و نظیر آنها می‌باشد. همچنین کلاسهایی برای مدیریت خطاها در سیستم‌های کاربردی در این مجموعه قرار گرفته اند.

یک مجموعه کلاس بسیار مفید نیز تحت عنوان SQL Object Model جزو کلاسهای کاربردی عمومی می‌باشد. این مجموعه کلاس یک مدل شیء‌گرا برای دستورات SQL ارائه می‌کند و برنامه‌نویسان برای نوشتن دستور SQL از این کلاسها استفاده نموده و مدل شیء‌گرای SQL خود را تهیه می‌نمایند. و در نهایت استخراج دستور SQL از این مدل براحتی انجام می‌گیرد. با استفاده از این کلاسها می‌توان دستور SQL را برای Oracle یا SQL Server و یا سایر پایگاه‌های داده تولید کرد. این امر موجب کاهش وابستگی به پایگاه داده می‌شود.

کلاسهای پایه‌ای معماری سیستم

برای حمایت از معماری مشخص شده توسط چارچوب کاری HyperNet کلاسهایی تهیه شده است. با توجه به اینکه معماری سیستم برپایه الگوی چند لایه بنا شده است، برای عمومی سازی در سطح هر لایه، برای هر کدام از لایه‌ها یکسری کلاسهای پایه‌ای تهیه شده است و

امکانات عمومی مانند ذخیره و بازیابی اطلاعات در سطح این کلاسهای پایه ای فراهم گردیده است. همچنین مسائل مربوط به ارتباط و Caching در این کلاسهای پایه ای مورد توجه قرار گرفته اند.

کلاسها و کنترل‌های Windows

برای تهیه رابط کاربر یک سیستم در محیط Windows، معمولاً امکاناتی فراتر از امکانات موجود در کتابخانه کلاسهای Net. مورد نیاز است. برای فراهم آوردن این امکانات در کتابخانه کلاسهای HyperNet تعدادی از کنترل‌های Net. توسعه داده شده و تعدادی کنترل جدید نیز تهیه شده است.

- ✓ کنترل Textbox برای اعمال کنترل بیشتر روی اطلاعات ورودی توسعه داده شده است و می توان حالت‌های تاریخ، فقط عددی، فقط حروف، زمان، مدت زمان و ... را برای آن تعیین نمود.
- ✓ کنترل‌های ShamsiDatePicker و ShamsiMonthCalendar برای انتخاب تاریخ شمسی و کنترل‌های مربوط به تاریخ شمسی در ورود اطلاعات تهیه شده‌اند.
- ✓ کنترل DataGridView نیز توسعه داده شده و با استفاده از آن، امکان استفاده از کنترل‌های موردنظر خود در سلول‌های Grid را خواهیم داشت. امکاناتی برای Paging و فیلتر نمودن اطلاعات نیز در این کنترل فراهم شده است.
- ✓ کنترل‌هایی تحت عنوان Lookup تهیه شده اند که امکان انتخاب از لیست را در هنگام ورود اطلاعات در اختیار کاربر قرار می دهند.
- ✓ کنترل‌های TreeView برای اضافه شدن امکانات راست به چپ توسعه داده شده است.
- ✓ کنترل TabControl برای اضافه شدن امکانات راست به چپ توسعه داده شده است.
- ✓ کنترل StatusBar برای اضافه شدن امکانات راست به چپ توسعه داده شده است.
- ✓ کنترلی برای اضافه نمودن Icon در منوهای سیستم تهیه شده است.

علاوه بر کنترل‌های فوق که بصورت عمومی و در هر سیستمی قابل استفاده می باشند، یکسری امکانات مرتبط با رابط کاربر نیز تهیه شده که از امکانات فراهم شده در معماری سیستم بهره می گیرد. این کلاسها عبارتند از :

- ✓ کلاس پایه ای برای همه فرم‌های سیستم که کنترل دسترسی به فرم را در آن سطح انجام می‌دهد.
- ✓ کلاس پایه‌ای برای فرم اصلی سیستم (MDI) که منوی سیستم در آن قرار گرفته و فراخوانی فرمها از طریق آن صورت می‌گیرد.
- ✓ کلاس پایه ای برای فرمهای ورود اطلاعات که امکاناتی را برای انجام عملیات اصلی یعنی جدید، اصلاح، حذف و جستجو فراهم می نماید. مسائلی مانند چک کردن ورود اطلاعات

اجباری و اعمال محدودیت های امنیتی برای کاربران غیر مجاز نیز در این کلاس پایه انجام می گیرد.

- ✓ کلاس پایه ای برای فرمهای ورود اطلاعات مبتنی بر Grid
- ✓ امکانات و فرم جستجو که در فرمهای ورود اطلاعات در سطح کل فرم یا برای کنترلهای Lookup مورد استفاده قرار می گیرد.
- ✓ فرم نمایش پیغامهای هشدار و خطاها و جزئیات بیشتر جهت راهنمایی کاربر
- ✓ فرم پایه ای کارتابل و خلاصه کارتابل
- ✓ فرم پایه ای پارامترهای گزارش و فرم نمایش گزارش ها

سیستم‌های عمومی

در طراحی سیستم‌های اطلاعاتی، به یکسری مشترکات برخورد می‌کنیم که خود آنها را نیز می‌توان به عنوان یکی از زیرسیستم‌ها در نظر گرفت. هر کدام از این زیر سیستم‌ها خود دارای جداول متعدد در پایگاه داده‌ها و کلاسهای مربوطه در لایه‌های مختلف سیستم و رابط کاربر مستقل می‌باشند. همه این زیرسیستم‌ها بر اساس معماری چندلایه و مبتنی بر سکوی HyperNet تهیه شده‌اند. بنابراین خود این زیر سیستم‌ها نمونه‌ای عملی و کاربردی از یک سیستم مبتنی بر سکوی HyperNet را ارائه می‌نمایند.

این سیستم‌ها شامل سیستم امنیت، سیستم گردش کار و سیستم ورود اطلاعات MetaData می‌باشد. حال هر یک از سیستم‌های عمومی را مورد بررسی قرار می‌دهیم:

سیستم امنیت^۱

سیستم امنیت یکی از زیر سیستم‌های عمومی است که برای کل سیستم مشترک بوده و همه زیر سیستم‌ها به آن نیاز دارند. نقطه شروع یک برنامه کاربردی یا یک سیستم اطلاعاتی فرم ورود به سیستم یا Login است که بخشی از سیستم امنیت می‌باشد. بنابراین وظیفه سیستم امنیت از هنگام شروع به کار با سیستم شروع می‌شود و در حین کار با سیستم از نظر دسترسی به فرمها، عملیات و داده‌های مجاز تاثیر گذاشته و تا زمان خروج از سیستم ادامه خواهد داشت.

وظایف سیستم امنیت را می‌توان به دو بخش اعتبار سنجی^۲ و اختیارسنجی^۳ تقسیم کرد :

- ✓ بخش اعتبار سنجی ورود کاربر به سیستم را کنترل کرده و تضمین می‌کند که فقط کاربر معتبر بتواند وارد سیستم شود.
- ✓ بخش اختیار سنجی سطوح دسترسی کاربران را کنترل کرده و در صورت مجاز نبودن کاربر برای استفاده از یکسری منوها، فرمها، کنترلها یا اطلاعات، آنها را مخفی یا غیر فعال می‌کند. امنیت در سطح داده‌ها و تضمین اینکه هر شخص فقط به داده‌های مجاز خود دسترسی پیدا کند نیز در این حوزه می‌گنجد.

در سیستم امنیت HyperNet امکانات اختیارسنجی در کلاسهای پایه‌ای فرمها قرار داده شده است و جلوگیری از نمایش فرم و یا غیر فعال و غیر قابل رویت کردن کنترلها برای کاربران بدون حق دسترسی در این کلاسهای پایه‌ای انجام می‌گیرد.

^۱ Security

^۲ Authentication

^۳ Authorization

همچنین امکان اعمال محدودیت‌های امنیتی در سطح داده‌ها نیز تعیین شده است. بدین طریق که محدود شدن داده‌ها در پایگاه داده انجام می‌گیرد و هیچ کاربری قادر به رویت داده‌های غیر مجاز نخواهد بود. این محدودیت برای مواردی مورد نیاز است که از پایگاه داده متمرکز استفاده می‌کنیم ولی هر کاربر فقط باید اطلاعات واحد یا شعبه خودش را ببیند. با این روش حتی گزارش‌های سیستم نیز می‌تواند در محدوده مجاز کاربر اجرا شود.

امکاناتی مانند گروه‌بندی کاربران نیز بخشی از سیستم امنیت است که توسط آن تخصیص دسترسی مشترک به چند کاربر فقط با اضافه کردن کاربر به یک یا چند گروه امکان‌پذیر است. قابلیت توسعه و امکان ارتباط سیستم‌های دیگر (سیستم‌های کاربردی و یا سیستم‌های عمومی مثل سیستم گردش کار) با سیستم امنیت نیز بخشی از ویژگیهای مفید این سیستم امنیت محسوب می‌شود.

امنیت از یک بعد دیگر، جلوگیری از نفوذ^۱ را نیز در بر می‌گیرد که در سیستم‌های تحت وب این مسئله اهمیت بیشتری پیدا می‌کند. یکی از راههای مقابله با آن چند لایه بودن سیستم از نظر فیزیکی است. اگر Application Server و Database Server جدا داشته باشیم و بین هر کدام از این سرورها دیواره آتش^۲ قرار دهیم مسلماً نفوذ از چندین سطح محافظت شده به راحتی امکان‌پذیر نخواهد بود. بنابراین چند لایه بودن سیستم در اینجا نقش بسیار مفیدی در امنیت سیستم خواهد داشت.

سیستم گردش کار^۳

امروزه در برنامه‌های کاربردی جداسازی گردش کار اسناد از سایر قسمت‌های سیستم بطور خاص مورد توجه قرار گرفته است. دلیل این امر را در کاهش مؤثر پیچیدگیهای سیستم‌ها می‌توان جستجو کرد. اگرچه این پیچیدگی به هر حال از بین نرفته و در طراحی موتور گردش کار خود را نشان می‌دهد، لیکن پرداختن به مسئله گردش کار به عنوان یک مسئله کلی، موجب بوجود آمدن راه حل‌های کلی و سیستماتیک شده و در نهایت به عنوان یک ابزار می‌تواند در تهیه برنامه‌های کاربردی مورد استفاده مجدد قرار بگیرد.

سیستم گردش کار، مراحل انجام کارها، ترتیب آنها و منابع (اشخاص) تخصیص یافته به آنها را کنترل نموده و توزیع کارها را بصورت الکترونیکی بر عهده می‌گیرد. همه موارد فوق در سیستم گردش کار قابل تعریف و تغییر می‌باشد بنابراین تغییر در روال‌های جاری یک سازمان ساده‌تر شده و به پویایی سازمان کمک می‌کند. همچنین با مشخص شدن روال‌های انجام وظایف مختلف در

^۱ Hack

^۲ Firewall

^۳ Workflow

سازمان، به روشن‌تر شدن نحوه انجام کارها و رفع مشکلات ناشی از مشخص نبودن روال‌ها منجر می‌شود.

در سیستم گردش کار HyperNet که بر اساس سکوی HyperNet تهیه شده است کلیه عملیات قابل انجام بر روی یک سند قابل تعریف است. این عملیات و ترتیب انجام آنها بر اساس یک نمودار فعالیت^۱ قابل نمایش است. اطلاعاتی که توسط نمودار فعالیت قابل بیان است در معرفی گردش کار یک نوع سند خاص مورد استفاده قرار می‌گیرد. برای معرفی گردش کار یک سند، فعالیت‌ها و ترتیب انجام آنها باید به سیستم معرفی شود.

این سیستم حالت‌های انتقال مشروط را نیز پشتیبانی می‌کند. حالت‌های انتقال مشروط زمانی اتفاق می‌افتد که بعد از یک فعالیت، بسته به مقادیر داده‌های فیلدهای اطلاعاتی سند، فعالیت‌های متفاوتی به عنوان فعالیت بعدی تعریف شوند.

با توسعه سیستم‌های مبتنی بر گردش کار دیدگاه‌های طراحی رابط کاربر نیز تغییر نموده و به سمت استفاده از کارتابل گرایش پیدا کرده است. استفاده از کارتابل مدیریت کارهای روزانه اشخاص را به نحو چشمگیری تسهیل می‌نماید و تسریع انجام کارها و نظارت بهتر مدیران به کارهای روزمره افراد تحت مدیریت خود و حذف کاغذ از چرخه عملیات اداری را موجب می‌شود. سیستم گردش کار HyperNet دارای امکان کارتابل و خلاصه کارتابل نیز می‌باشد. لذا امکان طراحی رابط کاربر به صورت مبتنی بر کارتابل موجود می‌باشد.

سیستم ورود اطلاعات Metadata

ساختار جداول سیستم و اسامی و عناوین آنها و فیلدهای جداول، ارتباط بین جداول، فیلدهای کد و نام و توضیحات جداول و ساختار کلاسها و فرمهای سیستم و جزئیات آنها مواردی هستند که در همه جای سیستم کاربرد دارند و بهتر است در یک جا مانند Database یا فایل XML به فرمت مشخصی آنها را داشته باشیم. در این صورت می‌توانیم از این اطلاعات استفاده‌های متعددی بعمل بیاوریم. موارد استفاده این اطلاعات می‌تواند Lookup های سیستم و Code Generator و Form Generator و ... باشد. یکسری از این اطلاعات ممکن است توسط سیستم امنیت نیز استفاده شود مانند لیست فرمهای سیستم و کنترل‌های داخل آنها که برای کنترل دسترسی توسط سیستم امنیت می‌تواند مورد استفاده قرار گیرد.

^۱ Activity Diagram

استانداردها

وجود استانداردهای مدون در تهیه یک سیستم نرم افزاری بسیار ضروری بوده و موجب سازگاری اجزای مختلف سیستم، افزایش کیفیت و تسهیل در نگهداری و پشتیبانی سیستم می‌گردد. کار کردن براساس استاندارد و به صورت روشمند یکی از اصول افزایش کیفیت محصول است که نظام مدیریت کیفیت ایزو بر این اصل استوار شده است. در این نظام به جای بررسی و تست محصول نهایی، روال تهیه محصول و روشمند بودن آن مورد بررسی قرار می‌گیرد.

مسئله مهم دیگری که در حیطه نرم افزار باید مورد توجه قرار گیرد یکپارچگی سیستم تهیه شده با سایر زیر سیستم ها می باشد. در صورتیکه سیستم ها به صورت روشمند و براساس استانداردهای مشترک تهیه شده باشند، برقراری ارتباط بین زیر سیستم های مختلف به آسانی انجام خواهد گرفت. البته این در صورتی است که یکپارچه سازی در تهیه استانداردها، مورد توجه ویژه قرار گرفته باشد.

با توجه به اینکه سکوی HyperNet یک راه حل جامع برای تولید نرم افزارهای مبتنی بر بانک های اطلاعاتی ارائه می کند، استانداردها و روشهای اجرایی در آن جایگاه ویژه ای را به خود اختصاص داده است و از اجزای مهم آن محسوب می گردد. این استانداردها موارد مختلفی را پوشش می دهند که در ادامه به معرفی آنها می پردازیم:

استانداردهای برنامه نویسی

استانداردهای برنامه نویسی، استانداردهایی هستند که لزوماً نمود خارجی ندارند و صرفاً در افزایش کیفیت کد تولید شده تاثیرگذار هستند. نمونه ساده استاندارد برنامه نویسی، استاندارد نامگذاری می باشد که موجب افزایش خوانایی کد برنامه شده و توسعه و پشتیبانی سیستم را آسانتر می سازد. استاندارد مدیریت خطاها و پیغامها نیز جزو یکی از مهمترین استانداردهای برنامه نویسی می باشد که تبعیت از آن موجب همگن شدن و تجمیع پیغامهای سیستم و قابل پیش بینی شدن عملکرد سیستم می گردد. همچنین رعایت این استاندارد چند زبانه شدن سیستم را نیز میسر می سازد. استانداردهای برنامه نویسی دیگر شامل استاندارد مدیریت وضعیت در سیستم های تحت وب و استانداردهای گزارشگیری می باشد.

استانداردهای رابط کاربر

طراحی رابط کاربر مناسب، نقش بسیار مهمی را در موفقیت یک سیستم نرم افزاری ایفا می‌کند. انتخاب رابط کاربر نه تنها در راحتی کاربران در استفاده از سیستم تاثیر دارد بلکه در کارایی سیستم نیز اثرگذار است. در سکوی HyperNet استانداردهای رابط کاربر، برای کارایی بالاتر و

راحتی کاربر طراحی شده است. همچنین برای پیاده سازی رابط کاربر سیستم براساس این استانداردها، امکاناتی نیز در کتابخانه کلاسها در نظر گرفته شده است.

در تهیه این امکانات و استانداردها، سعی شده است روشها و سلیقه های مختلف مدنظر قرار گیرد. برای مثال هم فرمهای تک رکوردی و هم فرمهای مبتنی بر Grid پوشش داده شده است. بدین ترتیب استفاده کنندگان از سکوی HyperNet در طراحی رابط کاربر، حق انتخاب دارند که کارایی بالاتر و راحتی کاربر فصل مشترک این انتخابها می باشد.

استانداردهای رابط کاربر شامل استانداردهای انواع فرم نمایش و ورود اطلاعات، اعم از فرمهای ساده و پدر و فرزندی و چند سطحی می باشد. همچنین نحوه حرکت بین فرمها، جستجو، فیلتر و Lookup و نحوه نمایش منوهای سیستم و مسائلی از این دست در حیطه استانداردهای رابط کاربر می گنجد که در سکوی HyperNet نیز مورد توجه قرار گرفته است.

استاندارد طراحی پایگاه داده ها

در طراحی ساختار جداول و اشیاء پایگاه داده، مسائل مختلفی باید مدنظر قرار گیرد. استانداردهایی که در این زمینه در سکوی HyperNet در نظر گرفته شده عبارتند از : تعیین کلید اصلی جداول و نوع داده های آنها، نحوه برخورد با مسئله همزمانی و نحوه استفاده از تراکنش ها و صفحه بندی اطلاعات در سطح پایگاه داده.

با توجه به اینکه تغییر در پایگاه داده موجب تغییرات نسبتاً زیاد در کد برنامه می شود بهتر است این استانداردها از همان زمان تشکیل پایگاه داده برای سیستم مورد نظر اعمال گردد تا از دوباره کاریها پرهیز گردد.

استانداردهای چرخه تولید نرم افزار

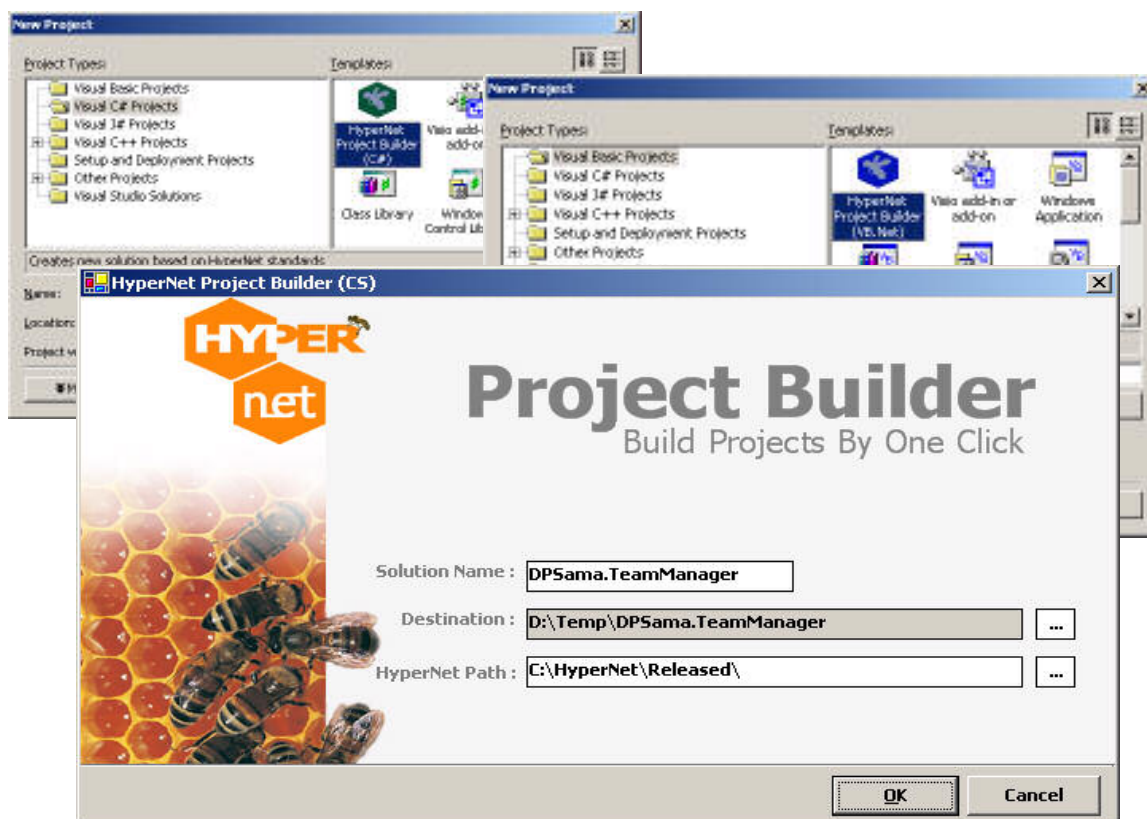
یک تیم نرم افزاری برای اینکه بتواند یک محصول با کیفیت تهیه کند لازم دارد تا روالهای خود را برای تولید آن محصول از ابتدا مشخص و مدون نماید و نیاز به استانداردهایی دارد که بر روی چرخه تولید نرم افزار متمرکز شده باشد. استانداردهای سکوی HyperNet در این مورد شامل مسائل مرتبط با استفاده از ابزارهای کنترل نسخه مانند Visual Source Safe، ارتباط بین زیر سیستم ها و مسئله ارتباط متقابل بین آنها و استاندارد تعیین شماره نسخه برنامه ها و روال عرضه یک نسخه جدید از سیستم تهیه شده، می باشد.

ابزارها

برای تولید سریع نرم‌افزار ابزارهایی نیز تهیه شده است این ابزارها هیچ وابستگی را ایجاد نکرده و کاملا مستقل عمل می‌کنند. البته ما در تهیه این ابزارها حل یا سرعت بخشیدن به کار با چارچوب کاری HyperNet را مورد توجه ویژه قرار داده‌ایم. حال به بررسی این ابزارها می‌پردازیم:

پروژه ساز HyperNet

ساختاری که در سکوی نرم‌افزاری HyperNet برای پیاده‌سازی یک سیستم نرم افزاری پیشنهاد می‌شود یک ساختار چند لایه است که در محیط Visual Studio .Net به صورت یک Solution حاوی چندین پروژه پیاده سازی می‌شود. هر کدام از این پروژه‌ها به تعدادی از پروژه‌های دیگر همین Solution ، تعدادی از اسمبلی‌های کتابخانه کلاسهای HyperNet و تعدادی از اسمبلی‌های کتابخانه کلاسهای .Net. ارجاع دارند. برای تسهیل ایجاد یک Solution جدید با مشخصات و پروژه‌های مورد نظر، پروژه‌ساز HyperNet را تهیه کرده‌ایم. شکل ۴ نمایی از این پروژه‌ساز را نشان می‌دهد.



شکل ۴ - پروژه‌ساز HyperNet

پروژه ساز HyperNet ابزاری است که امکان ایجاد یک Solution به همراه پروژه‌های داخل آن را بر اساس اطلاعات تعریف شده در یک فایل xml فراهم می‌نماید. این ابزار پس از نصب به صورت یک Wizard از داخل محیط Visual Studio .Net قابل دسترسی است و با زدن پروژه جدید، icon مربوط به پروژه ساز را نیز مشاهده خواهید کرد. با اجرای این Wizard همه پروژه‌ها بر اساس ساختار تعریف شده در فایل xml و نامی که برای Solution داده می‌شود ایجاد می‌شوند. اگرچه ساختار تعریف شده برای استفاده از کتابخانه کلاسهای HyperNet و بر اساس معماری پیشنهادی HyperNet تنظیم شده است، شما می‌توانید ساختار مورد نظر خودتان را در فایل xml ورودی تعریف کنید. بطور کلی ابزارهای تهیه شده کاملاً مستقل از معماری و کتابخانه کلاسهای HyperNet می‌باشند.

کدساز HyperNet

یکی از عواملی که باعث افزایش هزینه نگهداری سیستم می‌شود تکرار کد در نقاط مختلف برنامه است. تکرار کد باعث می‌شود که در صورت نیاز به تغییر آن کد، باید همه جاهایی که تکرار شده است تغییر یابد. و بعد از مدتی دیگر آن کدها شبیه هم نخواهند بود و خواهید دید که در هر قسمت با یک روش متفاوت عمل شده است. این مسئله دو راه حل دارد: یکی اینکه کد تکرار شونده را تا حد امکان به داخل یک کلاس عمومی ببرید و کلاسهای خود را از آن Inherit کنید تا همه آن امکانات را به ارث ببرد. و یا اینکه کار تکرار کد را توسط یک کدساز انجام دهید. در سکوی HyperNet هر دو مورد فوق فراهم گردیده است و علاوه بر کلاسهای پایه‌ای غنی برای همه لایه‌های سیستم، شامل یک کدساز انعطاف‌پذیر نیز می‌باشد که در اینجا به معرفی آن می‌پردازیم.

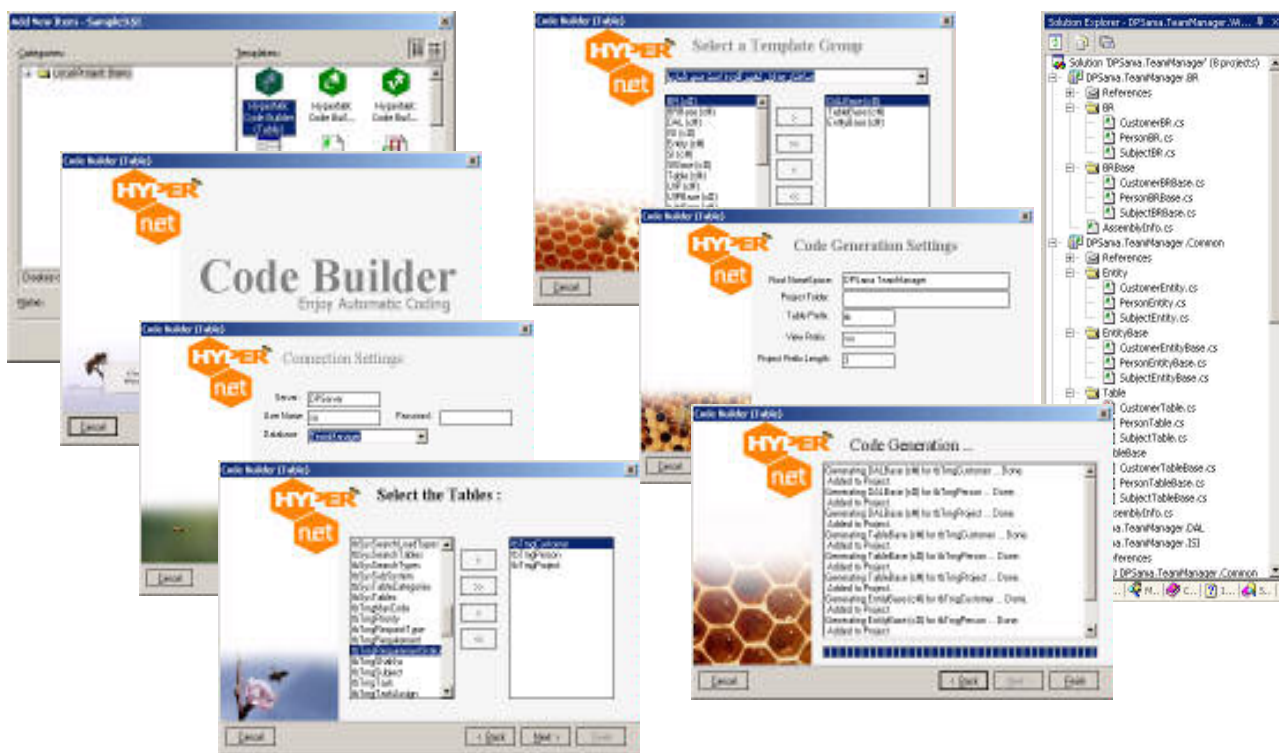
برنامه Hyper Builder یک کدساز مبتنی بر قالب (Template) است که امکان ایجاد هر نوع فایل اعم از کلاسهای لایه‌های مختلف سیستم تا فرمها و Script های SQL را به ما داده و سرعت تولید و تغییر نرم‌افزار را به حداکثر می‌رساند. در تهیه این قالب‌ها می‌توانید از زبانهای VB.Net و یا C# با تمامی امکانات و قابلیت‌های این زبانها و بدون محدودیت استفاده کنید. با داشتن این کدساز شما می‌توانید مدل پیش الگوی^۱ سیستم را از روی جداول سیستم و با یک کلیک تهیه کنید. داده‌های ورودی کدساز می‌تواند جداول پایگاه داده، View ها، Stored Procedure ها و Function های پایگاه داده باشد.

یکی از ویژگیهای مهم Hyper Builder اضافه شدن آن به Visual Studio.Net به صورت Wizard می‌باشد. با توجه به اینکه در سیستم‌هایی که بر اساس معماری چندلایه تهیه می‌شوند، به ازای هر موجودیت سیستم، در هر لایه باید یک یا چند کلاس تهیه شود، تهیه و اضافه کردن این فایلها معمولاً زمان بر بوده و ممکن است در انجام این کار به صورت دستی خطاهایی رخ دهد.

^۱ Prototype

کدساز Hyper Builder در صورت فراخوانی شدن از داخل Visual Studio .Net به صورت Wizard این امکان را فراهم می‌کند که فایل‌های ایجاد شده در پروژه‌های مربوطه و در محل مناسب تعریف شده در یک فایل xml قرار گیرند. بنابراین در صورت تغییر در طراحی یا موجودیت‌های سیستم در پایگاه داده، خیلی سریع می‌توان این تغییرات را در نرم‌افزار تهیه شده اعمال کرد.

شکل ۵ نمایی از فرم‌های کد ساز را نشان می‌دهد. چنانچه در شکل ملاحظه می‌کنید. کدساز از داخل پنجره Add New Item از داخل Visual Studio.Net قابل اجرا بوده و پس از مشخص نمودن جداول و Template های مورد نظر و تنظیمات تولید کد، کدهای مربوطه ایجاد شده و در محل مناسب خود در پروژه‌های Solution جاری اضافه می‌گردد.



شکل ۵ - کدساز HyperNet

قابلیت مهم دیگر کدساز HyperNet پشتیبانی از ابزارهای کنترل نسخه می‌باشد. در صورتیکه Solution شما تحت Source Safe باشد، در صورت استفاده از کد ساز HyperNet برای ایجاد فایل‌های جدید و یا بهنگام سازی فایل‌های موجود، درخواست Check Out فایل‌ها و پروژه‌ها نمایش داده می‌شود و در صورت عدم امکان Check Out این ابزار از بازنویسی روی فایل‌ها خودداری می‌کند. این امکانات عملاً سرعت برنامه‌نویسی را در تیم‌های نرم‌افزاری بطور چشمگیری بالا می‌برد.

با استفاده از این ابزار قادر به ایجاد حجم عظیمی از کد در چند دقیقه خواهید بود. این ابزار به همراه پروژه‌ساز HyperNet و با استفاده از معماری و کتابخانه کلاسهای HyperNet شما را قادر می‌سازد با در دست داشتن جداول سیستم در پایگاه داده، یک سیستم چند لایه قابل اجرا به صورت یکجا و یا توزیع شده را در چند دقیقه فراهم کنید. در این سیستم تولید شده عملیات عمومی مانند درج، اصلاح، حذف، نمایش و جستجوی اطلاعات در فرمهای تهیه شده قابل انجام می‌باشد. بنابراین برای تهیه یک سیستم نرم‌افزاری، شما فقط کافیست بر روی قواعد تجاری خاص سیستم خود تمرکز کنید و موارد تکراری را به سکوی HyperNet بسپارید.