در زمان ارائه چه چیزهایی گفته شد :

***این متن، توضیحاتی است که در زمان ارائه گفته شد.**

در زمان مطالعه فایل ارائه، این متن خوانده شود.

EIPمخفف Enterprise Integration Patterns است که به معنای الگوهای یکپارچه سازی تجاری می باشد. در واقع EIP یک روش دیگر برنامه نویسی است که در این روش با استفاده از الگوهای موجود ، کارکردها و سرویس های برنامه خود را می سازیم.

در برنامه نویسی اکثرا به دنبال این هستیم تا راحت تر برنامه نویسی کنیم. در عین حال به سرعت برنامه هم توجه داریم. این روش برنامه نویسی با الگوهایی که در اختیار ما قرار می دهد باعث می شود تا از المان های قدیمی کمتر استفاده کنیم.

این ارائه کوششی خلاصه ، برای معرفی این روش برنامه نویسی است.

در یک سازمان چند برنامه وجود دارد که با هم یکسری کار را انجام می دهند. معمولا یک برنامه (هرچقدر هم جالب باشد) ، به تنهایی کار خاصی انجام نمی دهد. در واقع وقتی کارایی به بیشترین حد خود می رسد که برنامه ها با هم یکپارچه شوند. برای مثال یک برنامه فروش در یک شرکت اگر با مثلا برنامه انبارداری آن شرکت یکپارچه یا Sync بشود بهتر است.

یکپارچه سازی برنامه ها یکسری چالش دارد. به چند مورد از این چالش ها اشاره خواهیم کرد.

چون روی یک سیستم نیستند و از طریق شبکه (کابل-روتر-سوییچ) به هم مرتبط اند، پس نقص در یک بخش شبکه (کابل-روتر-سوییچ) بر روی عملکرد برنامه ما تاثیرگذار خواهد بود. در شبکه نرخ تبادل اطلاعات پایین است. پس این مورد هم یک چالش بحساب می آید.

مسئله دیگر خود اپلیکیشن ها ست. یعنی هر کدام با یک زبانی نوشته شده اند، پلتفرم هایشان متفاوت است. همچنین اگر در یک برنامه تغییری ایجاد شد در سایر برنامه ها نیز باید تغییر اعمال شود تا یکپارچه شوند. پس برای یکپارچه کردن برنامه ها یکسری قانون و پروتکل باید در نظر بگیریم.

روش های موجود برای یکپارچه سازی:

File Transfer

یک برنامه باید Shared Data را صادر و برنامه دیگر آن را وارد کند.

در این روش باید دو برنامه بر روی نام فایل ها، فرمت اطلاعاتی که رد و بدل می شود و زمان بندی .با هم توافق کنند (چه زمانی بخواند B چه زمانی می نویسد و برنامه A برنامه)

همچنین باید مسئول پاک کردن آن اطلاعات ایجاد شده مشخص بشود

Shared Database

در این روش برنامه ها اطلاعات را بر روی یک Data Base مشترک قرار می دهند. پس باید قوانین و پروتکل هایی را رعایت کنند. در حال حاضر در سیستم هایمان ا این روش استفاده می کند

Remote Procedure Invocation

در این روش یک برنامه، یک تابع (سرویس) را با)API که مخفف Application Programming Interfaceمی باشد و به معنای رابط برنامه سازی کاربردی است ، یک رابط پیاده سازی توسط نرم افزار است که به دیگر برنامه ها اجازه میدهد با آن ارتباط داشته باشند را در اختیار برنامه دیگر قرار می دهد (.در سمت برنامه مصرف کننده یک (اصطلاحا) جا نگهدار قرار دارد که نحوه پیاده سازی اتصال به

این سرویس در آن قرار دارد. برنامه A از API ایی که برنامه B در اختیارش قرار داده استفاده می کند تا بتواند از آن سرویس استفاده کند. این روش بر خلاف سایر روش ها ، سنکرون می باشد. یعنی مصرف کننده منتظر می ماند تا جوابش را بگیرد. که این یک محدودیت برای این روش به حساب می آید

Messaging

در این روش یک برنامه وقتی که پیام تولید کرده را بخواهد در اختیار برنامه های دیگر قرار دهد، اینکار را گذاشته و سایر برنامه ها Message Bus انجام می دهد. یعنی پیام خود را در Message Bus از طریق از آن استفاده می کنند

EIPبا الگوهایی که در اختیار ما قرار می دهد باعث می شود استفاده از المنت های قدیمی مانند If و for کاهش پیدا کند.

با استفاده از الگوها یکسری اجزا یا مولفه یا کامپوننت ساخته می شود و با اتصال این مولفه ها به هم ، قابلیت برنامه ها را می سازیم.

قبل از معرفی الگو ها باید بدانیم که:

پیام ها از طریق Message Channel انتقال پیدا می کنند.

هر پیام اطلاعاتی است که میتواند در سیستم Messaging جابجا شود.

وقتی یک سیستم بخواهد یک پیام را در یک سیستم Messaging انتقال بدهد این کار را به همین سادگی انجام نمیدهد، بلکه پیام را در یک کانال قرار داده و این کانال یک ورودی و خروجی دارد که میتواند این پیام را به یک فیلتر دیگر منتقل کند.

با استفاده از این الگو، یک پیام از طریق یک لوله به یک فیلتر و این چرخه ادامه دارد تا به پیامی میرسد که بتوانیم در سیستم Messaging استفاده شود.

برای مثال در شکل یک پیام از طریق یک Pipe به Filter به نام Decrypt وصل میشود.در این Filterعملیات رمزگذاری انجام شده سپس از طریق Pipe به Filter به نام Authenticate می رود تا عملیات احراز هویت انجام شود و سپس از طریق Pipe به Filter به نام De-Dup می رود که این Filterکاری که انجام می دهد را بایک مثال شرح خواهیم داد. مثلا برای خرید دوبار دکمه ثبت را فشرده اید در صورتی که یک سفارش می خواستید و به اشتباه دوبار ثبت شده ، این Filter این مشکل را حل میکند و عمل ثبت را یک بار در نظر میگیرد. در نهایت پیام نهایی توسط Pipe منتقل می شود.

هر Pipe یک Filter را به یک Filter بعدی اش وصل می کند. ارتباط دهنده یا کانکشن بین یک Pipe و Filter، Portنامیده می شود.در ابتدایی ترین حالت هر Filter یک Port خروجی و یک Port ورودی دارد.

این مولفه یکسری از پیام هایی را که مد نظر ما نیست را حذف میکند و فقط آن اطلاعات و پیام هایی را در کانال خروجی اش می آورد که مدنظر ماست. برای مثال در شکل بالا پیام قرمز رنگ مدنظر ما نبود و در پیام منتقل شده نیز حضور ندارد. در واقع بوسیله Message Filter حذف شده است Message .
Filterیک خروجی دارد در واقعع Single Output می باشد. به این ترتیب که اگر تطبیق شد پیام منتقل می شود و گرنه حذف می شود.

گاهی اوقات که چند برنامه را بهم وصل میکنیم ممکن است فرمت اطلاعات رو عوض کنیم. مثلا یک برنامه که XML میگیرد را به یک برنامه که JSON میگیرد وصل کنیم. دراین موارد باید از Message Translatorاستفاده کنیم.

همانطور که در شکل مشاهده می کنید یک پیامی را گرفته و یک پیام با یک ترتیب دیگر را منتقل کرده است. در واقع فرمت آن را تبدیل کرده است.

این الگو با محتوای پیام کاری ندارد و مقصد پیام را مشخص می کند. این الگو عملکردی شبیه به if دارد.

این مولفه یک پیام را به تعدادی پیام شکسته و ارسال میکند. در برنامه نویسی با روش قدیمی معادلش whileو for می باشند.

این مولفه پیام های تکه تکه شده را گرفته و به هم وصل میکند و در نهایت پیام اصلی را در سیستم منتقل می کند. در برنامه نویسی با روش قدیمی معادلش while و for می باشد

انواع پیام ها در سیستم Messaging بصورت زیر است:

Command Message

Document Message

Event Message

این نوع پیام برای فراخونی یک رویه (Procedure) در یک برنامه دیگر استفاده می شود.

برای اینکه مقداری از اطلاعات (Data Structure) را به قسمت دیگر برنامه بفرستیم، از این نوع پیام ها استفاده میکنیم

این نوع پیام ها ، پیام هایی هستند که اطلاعاتی را به سایر قسمت های برنامه می رسانند. مثلا فلان بخش این اتفاق افتاده است در واقع Event Notification بین برنامه هاست

انواع Message channel

Point to Point

Publish Subscribe

Invalid Message channel

Dead letter channel

همانطوری که در شکل مشاهده می کنید هر پیام بطور نظیر به نظیر منتقل میشود. یعنی با همان ترتیب ارسالی ، منتقل و دریافت می شود.

در شکل هم واضح است. مشابه گروه هایی مانند یاهو یا جیمیل عمل میکند. یعنی یک عضوی از گروه (Subscribe)یک پیام را می فرستد و سایر اعضای گروه (Subscriber) ازآن استفاده می کنند

در سیستم ممکن است یکسری پیام های ناآسنا داشته باشیم ، پیام هایی که نمی توانند پردازش شوند یا فرمت مشخصی ندارند.

این دسته از پیام ها از طریق یک کانال به Invalid Message channel فرستاده می‌شوند تا از طریق مراجعه به این کانال این پیام ها را مشاهده و پیدا کنیم.

وقتی که در هرصورتی انتقال پیام (Delivery) انجام نشود (Fail) چند مرتبه عمل ارسال تکرار شده و در صورت عدم ارسال ، این پیام به Dead letter channel منتقل می شود تا در مراجعات بعدی بفهمیم که این پیام منتقل نشده است.

رو پشتیبانی می‌کنند. EIP ابزارهای (implement) هست که Frame Work در حال حاضر یکسری را ساپورت می کند EIP کانال پیام الگوهای Apache Camel(Apache Camel برای نمونه