دانشکده فنی و مهندسی

کارشناسی ارشد مهندسی کامپیوتر نرم‌افزار
گروه مهندسی کامپیوتر و فناوری اطلاعات

موضوع:

برسی روش های موجود در خودسازگاری و اجرای سرویس ها در سیستم های نرم افزاری

نگارش:

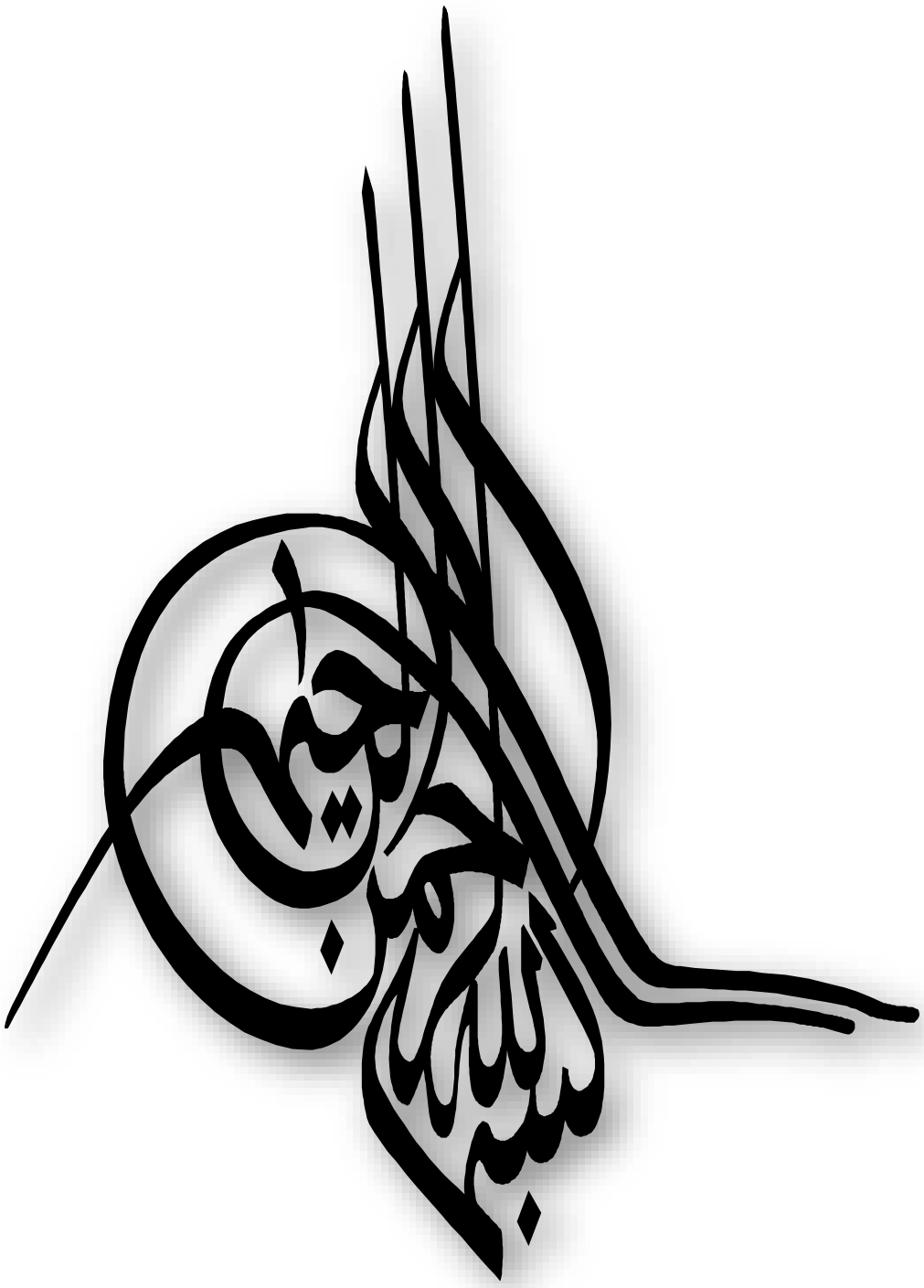
علیرضا رزمجو

۹۸۰۲۱۸۷۶۱

استاد راهنما:

دکتر رضوی

بهمن ۱۳۹۹


بسم الله الرحمن الرحيم

چکیده

تا کنون فعالیت های تحقیقاتی و کاربردی زیادی جهت ارائه فناوری ها، محصولات و استانداردهای محاسبات مبتنی بر سرویس های کامپیوتری صورت گرفته شده است اما کارهای گزارش شده بسیار محدودی در زمینه تحلیل و طراحی عملکرد و همچنین معماری سرویس ها موجود است که همین کارهای محدود نیز در برگیرنده مشکلات عمده و اساسی هستند. روش های ارائه شده در خصوص تحلیل سرویس ها به صورت خود محور نیست و شخصی که آن هارا تحلیل میکند باید آن را شخصا به انجام برساند و ادامه آن توسط افراد دیگر امکان پذیر نیست، همچنین این روش ها از تکمیل مدل ها پشتیبانی نکرده ، و ارائه راه حلهای بهینه که وابسته هستند به میزان مهارت شخص تحلیل گر در استفاده از این شیوه ها. در چرخه حیات مدلسازی سرویس های نرم افزاری سه رکن اصلی وجود دارد که عبارتند از: ۱) شناسایی سرویس ها، ۲) مشخصه سازی سرویس ها و ۳) طرح و پیاده سازی سازی سرویس ها . هدف از این گزارش بررسی روش های موجود در زمینه فاز سوم یعنی طرح و پیاده سازی سازی سرویس هاست . تا پس از بررسی روش های موجود بتوان به راه حلی خود کار در زمینه طرح و پیاده سازی سازی مدل سرویس رسید. مهم ترین ورودی های این فاز، مدل سرویس و مستندات مربوط به معماری هستند و خروجی آن مدل طراحی است و به دلیل این که مدل طراحی تنها یک گام تا پیاده سازی سرویس ها فاصله دارد این فاز از اهمیت ویژه ای برخوردار است . همچنین این فاز تا کنون به طور خودکار انجام نشده است . برای خودکار سازی این فاز پس از بررسی کارهای مرتبط ، مدل مطلوب خود را برای پیاده سازی این سرویس های نرم افزاری معرفی می کنیم.

در قسمت توصیف و شناسایی مولفه ها ، از روی سرویس هایی که در قسمت مشخصه سازی این سرویس ها پیدا کرده ایم، مولفه هایی با بر قراری اتصال سست را پیدا خواهیم کرد .

کلمات کلیدی : معماری سرویس گرا، تحلیل و طراحی سرویس گرا، مدلسازی سرویس گرا، طرح و پیاده سازی – سازی مدل سرویس

فهرست مطالب

فصل اول : مقدمه

۱,۱ تعریف مسئله

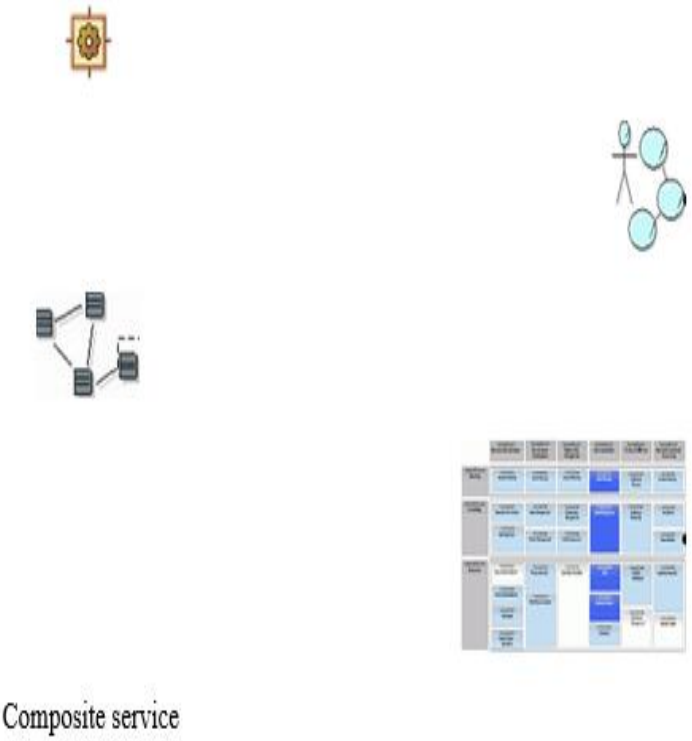
امروزه یکی از راه حلهای نوین به منظور ساخت سیستم های کاربردی سازمانی ، راه حل مبتنی بر سرویس تلقی می شود. به علاوه معماری مبتنی بر سرویس به عنوان یکی از شیوه های معماری یشرو در راه حل های سازمانی مطرح است . تاکنون فعالیتهای تحقیقاتی و کاربردی زیادی جهت ارائه فناوری ها، محصولات و استانداردهای محاسباتی مبتنی بر سرویس انجام شده است ، ولی کارهای گزارش شده بسیار محدودی در زمینه های تحلیل و طراحی سرویس ها (مدلسازی سرویس گرا) موجود است که همین کارهای محدود نیز دربردارنده مشکلاتی هستند. شیوه های ارائه شده جهت پیاده سازی سرویس ها به صورت تجویزی بوده و خودکارسازی آنها میسر نیست و شخصی که آن ها را تحلیل می کند باید آن را شخصاً به انجام برساند. این روش ها در محدوده های با اندازه های گسترده تر همچون سازمان ، دچار مشکلات عملیاتی نظیر عدم بهینه بودن فرآورده های تولیدی و یا عدم بهره مندی از سرعت مطلوب با توجه به معیارهای مربوطه می گردد. به جهت گسترش سیستم های سازمانی ، سطح تجرید اجزاء معماری باید به اندازه سطح دامنه کاری سازمان ارتقاء یافته باشد [۱]. روش های سنتی شی ء گرایی و مبتنی بر اساس سیستم هایی ، کافی نبوده و منجر به ایجاد روش های جدید مبتنی بر سرویس که از سطح تجرید بهتری برخوردار است ، شده اند [۲،۳]. معماری مبتنی بر اساس سرویس به عنوان یکی از سبک های معماری [۴] در راه حل های سازمانی نیز مطرح است .

این سبک معماری-که از سبکهای معماری فناوری اطلاعات است - نه تنها راجع به محصولات و استانداردهای مربوطه است ، بلکه به سایر جنبه های توسعه سیستم ، نظیر تحلیل و طراحی راه حل نیز مرتبط است . مراحل اولیه چرخه حیات راه حل مبتنی بر سرویس که تحلیل و طراحی سرویس ها و یا مدلسازی سرویس ها [۳] نامیده می شوند شامل شناسایی ۲، مشخصه سازی ۳ و محقق سازی ۴ سرویس ها در فرآیند مدلسازی سرویس ها می باشند که با بالابردن سطح تجرید سازمان به راه حلی با کیفیت بالاتر جهت پیاده سازی SOA سازمان می پردازند. هدف اصلی این گزارش ، بررسی درباره اینکه به راه حلی برسیم که برای اتوماتیک ساختن سومین فاز چرخه حیات مدلسازی سرویس گرا، یعنی فاز محقق سازی سرویس است . مهم ترین ورودی های این فاز مدل سرویس و مستندات مربوط به معماری هستند و خروجی آن مدل طراحی است و به دلیل این که مدل طراحی تنها یک گام تا پیاده سازی سرویس ها فاصله دارد این فاز از اهمیت ویژه ای برخوردار است . همچنین ، انجام اتوماتیک این قسمت از اهداف این پژوهش است و می توان به کمک این روش، مستقلا به هدف تبدیل خودکار مدل کاری سازمان به سطح های پایین تر از طراحی هم دست یافت . روش های لازم جهت تبدیل خودکار مدل کاری سازمان به مدل سرویس های سازمانی (شامل دو فاز شناسایی و مشخصه سازی سرویس ها) را باهم

برسی خواهیم کرد . ارائه روشی جهت خودکار ساختن مدیریت تکامل مدل های سرویس و پارامتر هایی به منظور کنترل شاخص ها ی کیفی در فازهای مختلف معماری سرویس گرا می باشد..

نمودار مفهومی این معماری را در شکل ۱,۱ ملاحظه می کنید:

.



شکل ۱-۱

این چارچوب شامل تکنیک ها، روش ها و ابزارهای مربوطه جهت خودکارسازی فعالیت های مدلسازی ها که منجر به ایجاد اتوماتیک مدل از مدل فرآیندها می گردد، است . این اقدام به جهت رسیدن به راه حلی با کیفیت انجام می شود . در شکل ۲,۱ مشاهده میکنیم که فعالیت هایی شامل شناسایی ، توصیف و همچنین محقق سازی سرویس ها را ، شخص معمار نرم افزار به دنبال خودکار سازی این تسک ها به جهت راهنمایی کردن نرم افزار ، به اتمام می رساند شکل زیر بیان گر سه فاز است که تست و اعتبار سنجی داده ها را انجام می دهد .

شکل ۲.۱ چرخه حیات مدلسازی سرویس گرا بدون استفاده از چارچوب [۵]

شکل ۳.۱ چرخه حیات مدلسازی سرویس گرا با استفاده از چارچوب [۵]

۲,۱ اهداف تحقیق

همان گونه که توضیح داده شد ، هدف از این تحقیق ، بررسی در زمینه ارائه راه حلی جهت اتوماتیک ساختن مرحله سوم معماری های سرویس گرا، که همان فاز محقق سازی است ، تا بتواند در یک قالب ASOMFجایگاه مولفه Automated Service Realization Methodرا مختص خود کند. پس در این تحقیق ابتدا به دسته بندی و مرور موارد موجود در این حوزه می پردازیم و بعد از آن با مقایسه این روش ها به یک پیش نمایش کلی از راه حل خود می - رسیم و مرحله های آینده جهت به سرانجام رساندن تحقیق را مشخص می کنیم . در این تحقیق با مورد بحث قرار دادن فرایند های موجود، امکان ارائه یک روش خودکار را بررسی خواهیم کرد، و با معرفی معماری مدل رانه و آشنایی مفهوم های اساسی آن ، پیش نمایش روش خود را در قالب آن چارچوب معرفی می کنیم..

۳,۱ ساختار تحقیق

سرویس ها، مدل سازی سرویس ها ، مولفه ها و ... آشنا خواهیم شد. در فصل ۳ نیز روش های موجود را به چهار دسته ی چارچوب های مبتنی بر SOMA، روش های یاده سازی معماری سرویس گرا ، روش های شناسایی مولفه ها و دیگر روش های محقق سازی ، تقسیم می کنیم ، در این فصل با جزییات این روش ها آشنا و در نهایت به مقایسه آن ها خواهیم پرداخت .

فصل دوم : ادبیات تحقیق

در این فصل با ادبیات تحقیق و مروری بر قسمت ها و منطق معماری های سرویس گرا یا ، راه حل های مبتنی بر آن ، مدلسازی و معرفی فازهای آن ، مولفه ها و ویژگی های آن ها و در نهایت قواعد و مفاهیم مورد نیاز در فصل های بعدی آشنا خواهیم شد.

۱,۲ معماری سرویس گرا

عبارت SOAبا مبالغه و کلیات احاطه شده است. ما به تعریفی احتیاج داریم که بتوانیم با آن کار کنیم.
قبل از اقدام جهت پاسخ دادن صحیح به انتظارات می بایستی SOAرا با اندکی دقت تعریف کنید. حجم زیادی از مبالغات که SOAرا در سالهای اخیر احاطه کرده است، تا حدودی مسئله را سخت ساخته. اما این کار قابل انجام است. تعریف های زیادی از SOAوجود دارد. در اینجا تعریف مورد نظر به قرار زیر است:

SOAنوعی معماری است که سرویس را برای ساده سازی فعالیت های یکپارچه سازی و استفاده از اجزا با قابلیت استفاده مجدد به روش اتصال سست به شکل بلوک ساختمانی به کار می گیرد. در حالی که ممکن است این مطلب اغراق آمیز کردن یک امر واضح باشد، اما یک درس کوچک در آن وجود دارد. ایجاد مجموعه ای از سرویس ها به طور خودکار یک معماری را به سادگی ارائه نمی دهند. معماری، درحالی که راهنمای ایجاد سرویس می باشد، یک موضوع مجزا از اجرای خود سرویس می باشد.
در اصطلاح SOA، سرویس گرای را به عنوان صفتی از معماری به کار می برد. یعنی SOAیک روند معماری، که قابل اجرا می باشد را توصیف می کند. این یک حالت توسعه نمی باشد. نمی توان به سادگی یک EJBرا طوری تفسیر کرد که نشان دهد یک سرویس تحت شبکه است و حالا یک سرویس تحت شبکه داریم. باید یک آغاز مناسب برای SOAوجود داشته باشد. اگر با دقت به هدف نگاه کنیم، امکان دارد شروعی بسیار پرزحمت و گران قیمت داشته باشیم.

ساخت SOAنیازمند تجزیه تحلیل و یکپارچگی قابل ملاحظه ای است و نمی توان به سادگی خرید یک کالا به آن دست یافت

این سرویس بخشی از معماری سرویس گرا می باشد و با تاکید بر تعریف گفته شده پیش می رویم. اگر معماری وجود نداشته باشد یا سرویس بخش اصلی معماری نباشد SOAبی معناست. (مانند یک سیستم شئ گرا که شئ بخش اصلی آن است.)

بدون وجود سرویس ها چیزی برای ساخت، نظارت و چیزی برای اجرا وجود ندارد .از سوی دیگر، کسی که زحمت بسیار بکشد می تواند یک مجموعه ای از نمودارهای معماری تولید کند که با دنیای واقعی ارتباط دارد. اگر شما با دقت بر کار روی مجموعه ای از اجرای سرویس ها تمرکز کنید و معماری را فراموش کنید، نه تنها SOAندارید بلکه چیزی بدتر از آن دارید. و وقت وهزینه قابل توجهی از شما را برای حل مشکل نامربوط می گیرد. این روند نادرست را دنبال نکردن الگو (anti-pattern)می گویند که باعث ایجاد مجموعه ای از وب سرویس های بی هدف می شوند. همچنین برنامه نویسان ناشی که بدون راهنمایی معماری کار می کنند ناخواسته کسب و کار را از دنبال کردن الگو منحرف می سازند، که حاصل مجموعه ای است از سرویس های بدون حاکمیت که قادر به ساخت SOAنمی باشند..

SOAیک طرز تفکر درباره سیستم های یکپارچه بیان می کند SOA .پیشنهاد کسب و کار جدید یا گسترش یافته یا فرصت هایی با اتصال سیستم های چندگانه با هم را ارائه می دهد. در ساده ترین شکل آن، این بدان معناست که شما می توانید کار B2Bرا سریعتر انجام دهید.

CORBA، معماری اتصال، EDI، point-to-point EAIبیش از چند دهه تلاش کردند که همه نشان دهند توانایی یکپارچه کردن سیستم های گوناگون را دارند. اما این گونه تلاش ها می تواند پرهزینه و شکننده باشد. آنها می توانند برنامه نویسان و بخشهای ITرا از حل مشکلات واقعی کار منحرف سازند. به طور کلی SOAبا استفاده از XMLبه عنوان فرم استاندارد برای پیام ما را به هدف یکپارچه سازی نزدیک می کند.

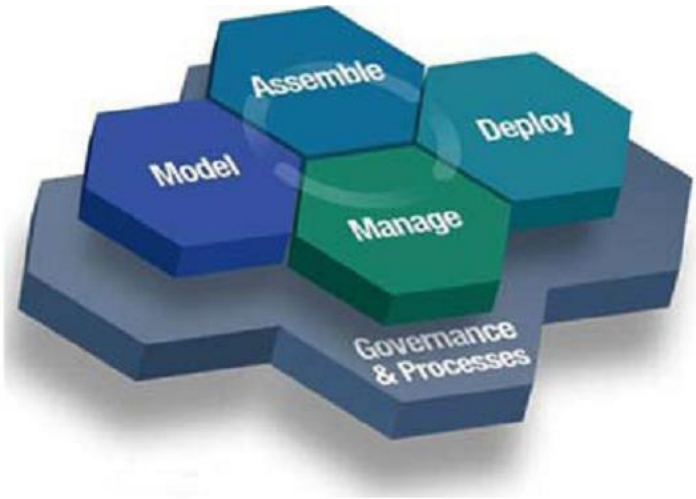
۱,۱,۲ معماری سرویس گرا از دیدگاه کلی

معماری سرویس‌گرا یکی از رهیافت‌های طراحی نرم‌افزار با تمرکز بر طراحی سامانه‌های توزیع‌شده است. در این معماری کارکردهای نرم‌افزاری در قالب سرویس توسط مؤلفه‌های برنامه‌های کاربردی به دیگر مؤلفه‌ها در بستر ارتباطات تحت شبکه ارائه می‌شود .

امروزه از این معماری در دستگاه‌های دولتی و شرکت‌های خصوصی برای توسعه برنامه‌های کاربردی و سیستم‌ها، یکپارچگی سیستم‌های اطلاعاتی سازمانی یا تعاملات اطلاعاتی بین سازمانی و استاندارد وب سرویس (Web Service) و پروتکل‌های آن استفاده می‌شود.

۲,۱,۲ چرخه حیات معماری سرویس گرا

IBM برای معماری سرویس گرا، چرخه حیاتی بر مبنای شکل ۱,۲ ارائه داده است :



این چرخه دارای چهار مرحله مدل ، گردآوری ، نصب و مدیریت می باشد و به حاکمیت فرآیند

ها به عنوان تعاریف کلی نگریسته شده است :

- یک چارچوب استراتژیک از فناوری که به تمام سیستم های داخل و خارج اجازه ارائه یا دریافت سرویس های خوش تعریف را می دهد .

- روشی برای طراحی و پیاده سازی نرم افزارهای گسترده سازمانی به وسیله ارتباط بین سرویس هائی که دارای خواص اتصال سست، دانه درشتی و قابل استفاده مجدد هستند .
- سبکی از معماری که از اتصال سست سرویس ها جهت انعطاف پذیری و تعامل پذیری حرفه و بصورت مستقل از فناوری پشتیبانی می کند و از ترکیب مجموعه ای از سرویس های مبتنی بر حرفه تشکیل شده که این سرویس ها انعطاف پذیری و پیکربندی پویا را برای فرآیندها محقق می کنند .
- چارچوبی وسیع و استاندارد که سرویس ها در آن ساخته، استقرار و مدیریت می شوند و هدفش افزایش چابکی زیر ساخت های فناوری اطلاعات در جهت واکنش سریع به تغییرات در نیازهای کسب و کار می باشد .
- سبکی از معماری که هدف آن دستیابی به اتصال سست در ارتباطات بین مولفه های نرم افزاری است. سرویس واحدی از کار است که توسط ارائه دهنده سرویس انجام می شود تا نتیجه مطلوب برای درخواست کننده سرویس را مهیا نماید. هر دوی ارائه دهنده و درخواست کننده سرویس، نقش هائی هستند که بوسیله عوامل نرم افزاری به جای عوامل انسانی انجام می شوند .
- رهیافتی جهت سازماندهی و بهینه سازی قابلیت های توزیع شده که تحت کنترل حوزه قلمرو چندین مالک می باشد و ارائه دهنده روشی یک شکل برای سفارش، شناسائی، تعامل و استفاده از قابلیت هاست .
- در محیطی که سرویس گرائی بر بستر معماری سازمانی بنا شده، به دنبال مجموعه گسترده ای از تجارب و قوانین جهت طراحی
- و تکامل واحدهای سازمانی هستیم که منابع حرفه را به شکل سرویس درآورد. به این مجموعه از قوانین و تجارب معماری سرویس گرا گوئیم .
- سبکی ازمعماری برای ساخت نرم افزارهایی که از سرویس های منتشر شده در یک شبکه مانند وب استفاده می کنند. اتصال سست بین مولفه های نرم افزاری باعث قابلیت استفاده مجدد از آنها می شود و نرم افزارها بر مبنای سرویس ساخته می شوند، سرویس در اینجا به معنای پیاده سازی یک کارکرد حرفه خوش تعریف است که می تواند در فرآیندهایا نرم افزاهای مختلف مورد استفاده و فراخوانی قرار بگیرد .
- چارچوبی برای یکپارچه سازی فرآیندهای حرفه و پشتیبانی آنها توسط فناوری اطلاعات با کمک مولفه های استاندارد و امن تحت عنوان سرویس که قابلیت استفاده مجدد و الحاق به یکدیگر جهت پوشش تغییرات حرفه را دارا می باشند .

۲,۲ مدلسازی

مدلسازی سرویس ها اولین قدم در چرخه حیات معماری سرویس گرا است .اصلی ترین هدف از مدلسازی سرویس ها بالا بردن سطح تجرید و کاهش پیچیدگی در کار با سرویس ها می باشد .برای این منظور ابتدا در دامنه مورد بررسی ، سرویس های مورد نیاز را شناسایی می کنیم .این شناسایی بر اساس اهداف و فرآیندهای حرفه مورد بررسی انجام می شود.

سپس ویژگی های هریک از این سرویس ها را به صورت جزئی تر مشخص کرده و نحوه تعامل آنها را با هم تعیین می کنیم . برای این کار از نمودارهای مختلفی مانند نمودار ارتباط سرویس ها، نمودار همکاری بین سرویس ها و ... استفاده می شود، در آخر به عنوان آخرین مرحله از مدلسازی سرویس ها، آنها را بر اساس ویژگی ها و خصوصیاتشان و همچنین با توجه به ویژگی های حرفه مورد بررسی از نظر چگونگی تولید مورد بررسی قرار می دهیم.

۱,۲,۲ تعریف سرویس

سرویس عملی ست که به وسیله یک سرویس دهنده انجام می شود و از نظر سرویس - گیرنده ارزشمند است . البته مفهوم سرویس در معماری سرویس گرا بی شباهت به مفهوم شی و مولفه نیست . با وجود اینکه سطح تجرید سرویس از شی و مولفه بالاتر است شباهت آن را می توان در این که هم سرویس هم شی و هم مولفه به نوعی منطق سازمان را پیاده می کنند و مسائلی مانند اتصال و چسبندگی در آن ها باید رعایت شود دانست .مولفه ها ارتباط منطقی بیشتری به نسبت سرویس ها با هم دارند.

۲,۲,۲ مراحل مدلسازی سرویس

همانگونه که اشاره شد مدلسازی سرویس ها عبارت است از سه فاز شناسایی ، توصیف و محقق سازی سرویس ها. در این بخش به مرور مختصری روی وظائف این فازها می پردازیم

۱) شناسایی سرویس

این فاز دارای رویکردهای بالا به پایین ، پایین به بالا و میانی ی م باشد. در رویکرد بالا به پایین از وظائف و فرآیندهای حرفه ، سرویس ها را شناسایی می کنیم ، عملیات اصلی در این فاز تحلیل دارایی های موجود و مدلسازی سرویس -هدف می باشد.

۲) توصیف سرویس

این فاز فاز توصیف سرویس هاست ، چه خدماتی ارائه می دهد، چه درخواست هایی دارد و چگونه نمایش داده می شود. همچنین این فاز به پیام ها، روابط ، تعاملات و وابستگی های بین سرویس های گوناگون می پردازد. ۸ مدل اصلی مرتبط با این فاز مدل سرویس است . در این فاز به فاکتور مهم اتصال سست در سرویس ها می پردازیم زیرا این فاکتور به کمک دیدن نمای بیرونی سرویس ها از دنیای خارج مهیا می شود.

۳) محقق سازی سرویس

در این فاز به این مرحله پرداخته می شود که برای یک سرویس خاص چه راه حلی اندیشیده می شود، مدل اصلی مرتبط با این فاز مدل طراحی ست .

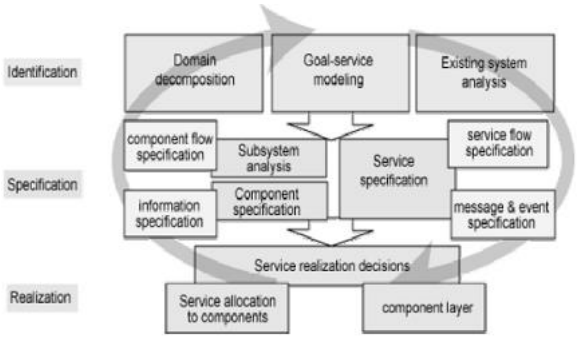
مرزبندی های مختلفی در منابع مختلف برای این فاز ارائه شده است ولی کارهایی که در این گزارش برای این فاز تعریف می کنیم عبارتند از دو بخش شناسایی و توصیف مولفه ها و تصمیمات مربوط به پیاده سازی سرویس ها.

۳,۲,۲ اهمیت محقق سازی سرویس

مهم ترین ورودی های فاز محقق سازی که در بخش پیش بدان اشاره شد،مدل سرویس و مستندات مربوط به معماری هستند و خروجی آن مدل طراحی است و به دلیل این که مدل طراحی تنها یک گام تا پیاده سازی سرویس ها فاصله دارد این فاز از اهمیت ویژه ای برخوردار است . در این فاز است که ویژگی های حیاتی مانند دسترس پذیری ، کارایی و امنیت از سرویس ها باید با محقق سازی صحیح وارد مولفه ها و در نهایت مرحله پیاده سازی شود. علاوه بر این ، این فاز به صورت مستقیم توسط معمار و طراح سیستم انجام میشود و از اهمیت ویژه - ای از نظر سطوح بالای مدیران ارشد سازمان برخوردار است .

درباره Soma :

چارچوبی است برای درک تغییر در محیط های روزمره استفاده از نقشه های شیء معنایی و دارای فازهای سه گانه شناسایی ، توصیف و تحقق سرویس های سازمانی ست ، این چارچوب مبنای کارهای تحقیقاتی زیادی قرار گرفته است . در زیر سه فاز مدلسازی در SOMA به همراه وظائف مربوط به آن ها نشان داده شده است ، این چارچوب به توصیف کلی این سه مرحله ، پیش نیازها و ورودی ها و خروجی هایشان می پردازد. این سه فاز برای او ین بار در [۷] مطرح شده است . گام های آن نیز مطابق شکل ۲,۲ است :



شکل ۲,۲ فرآیندهای شناسایی ، مشخصه سازی و محقق سازی در چارچوب SOMA [۷]

۴,۲ معماری مدل رانه

روشی برای توسعه سیستم است که در آن ، مدلها از جایگاه خاصی برخوردارند .این روش یک روش با محوریت مدل یا مدل محور است . زیرا مدلها وظیفه هدایت و راهبری جریان فهم ، طراحی ، ساخت ، استقرار، بهره برداری، نگهداشت ، و اصلاح را بر عهده دارند.

معماری مدل رانه ، مدلهای مهم حرفه را از بخش های فنی جدا می کند و این امر باعث مطرح شدن ریزه کاری های مربوط به فناوری می شود. با این جداسازی، برنامه خواهد توانست بدون مشکل و به نسبت به شکل ساده ای، بر روی یک سکوی فن آوری دیگر استقرار یابد، بدون آنکه نیاز باشد تغییری عمومی و یا مهم در مدلهای مربوط به بخش های حرفه رخ دهد و به این شکل یکی از مشکلات شرح داده شده برای روشهای بهبود یافته را حل کرده و قابلیت حمل در معماری مدل رانه ایجاد می شود.

۵,۲ مولفه

مولفه نرم افزاری یک واحد قابل اجرا برای ترکیب بوده که واسط های آن بصورت قراردادی معرفی شده و وابستگی های رفتاری آن کاملا تعیین شده است .مولفه نرم افزاری مستقل بوده و قابل ترکیب با محصولات دیگر می باشد. این تعریف به نکات جالب توجهی اشاره دارد؛ اما نیاز به نکاتی دارد که آن را کامل کند، مولفه ها قابل نمونه سازی هستند و می توانند به طور مستقل در ساخت بسته های مولفه شرکت کنند. مدل مولفه مشخصه هایی در رابطه با مولفه ها، ترکیب آن ها و شرایط محیط اجرای آن ها ارائه می کند.

۶,۲ نتیجه گیری

در این فصل پس از معرفی مفهوم معماری سرویس گرا، اهداف فازها و چرخه حیات آن با گام های موجود در چرخه حیات معماری سرویس گرا تا حدی آشنا شدیم .همچنین گام اولین این چرخه ، یعنی مدلسازی سرویس گرا را شرح داده و نسبت به فعالیت های موجود در این گام شناخت کلی پیدا نمودیم .در ادامه فعالیت محقق سازی سرویس ها مورد بررسی قرار گرفت و دلایل اهمیت این گام تعیین گردید.

در ادامه تلاش داریم روشی را به منظور انجام این فعالیت معرفی کنیم .برای این کار در فصل های آینده به معرفی مفاهیم موجود در حیطه تشخیص سرویس ها پرداخته و سپس به معرفی روش های موجود در این زمینه می پردازیم و با مقایسه روش ها و بررسی نقاط قوت و ضعف هریک از آن ها، به معرفی روشی جدید جهت رفع این نواقص می پردازیم .

فصل سوم : بررسی روش های موجود

مقدمه

در این فصل به بررسی روش های مختلف ارائه شده در زمینه مدل سرویس می پردازیم ، در پژوهش های مختلف زمینه های گوناگونی برای مرحله پیاده سازی در نظر گرفته شده است ، برخی از پژوهش های صورت گرفته طی چند سال اخیر حاکی از آن است که به طور کلی پیاده سازی معماری سرویس گرا در سطح سازمان پرداخته می شود و در این زمینه روش های گوناگونی نیز معرفی شده است ، در این فصل ما به بررسی این پژوهش ها و به ویژه تمرکز بر حیطه "محقق سازی سرویس ها" در این روش ها خواهیم پرداخت .

روش های مطالعه شده در این فصل را به سه دسته زیر تقسیم بندی کرده ایم ، دسته اول روش هایی که یک چارچوب ارائه برای پیاده سازی معماری سرویس گرا در سازمان ارائه می دهند و بر مبنای متدولوژی SOMA شرکت IBM چرخه حیات مدلسازی سرویس را به سه فاز شناسایی ، مشخصه سازی و محقق سازی تقسیم میکنند. دسته دوم روش هایی که هدفشان پیاده سازی معماری سرویس گرا در سطح سازمان است ولی لزوما به سه فاز مدل سازی سرویس ها پایبند نیستند ولی با بررسی کلی این روش ها می توان ایده های آن ها در زمینه چگونگی پیاده سازی سرویس هایشان را بررسی کرد، دسته سوم روش های شناسایی مولفه ها در سطح سازمان هستند، این روش ها از این جهت اهمیت دارند که یکی از مهمترین گام هایی که ما در فاز محقق سازی بدان توجه می کنیم فاز شناسایی مولفه های سازمان هاست .

۱,۳ قالب های بر اساس معماری SOMA

محیط مدلسازی SOMA-ME چارچوبی جهت پیاده سازی و بکارگیری رهنمودها و روش های موجود در SOMA است . این محیط مدلسازی بر روی یکی از ابزارهای قدرتمند توسعه راه حل های نرم افزار به نام Rational Software Architect(RSA) توسعه داده و موجب گسترش آن نیز گردیده است . این محیط همچنین از پروفایل مدلسازی سرویس UML استفاده می نماید. با وجود اینکه ادعا شده است این چارچوب جهت خودکارسازی فعالیتهای مدلسازی سرویس مورد استفاده قرار می گیرد و به دلیل اینکه پایه و بنیان آن بر روشSOMA ، که یک روش ذاتا دستی است ، قرار دارد صرفا تعدادی دستورالعمل اجرایی به عنوان راهنما در این محیط تعبیه شده است. نویسندگان مقاله [۱۱] معتقدند SOMA-ME پیوند بین فرآورده های راه حل (فرآیند، سرویس و مؤلفه سرویس) و ساختارهای داده ای موجود در صنعت را فراهم می کنند.

مؤلفه های کلیدی SOMA-ME عبارتند از:

الف) رهنمودهائی که مبتنی بر آگاهی از زمینه هستند

ب) الگوهای طراحی قابل استفاده مجدد

ج) الگوریتم با قابلیت سفارشی سازی برای انتخاب مجموعه ای از سرویس هائی که باید نمایانده شوند که تحت عنوان SLT[۱۱] شناخته می شود. SLT یا تست لیتموس سرویس به این سؤال که " از میان سرویس های کاندیدا کدامیک برای پیاده سازی مناسب است ؟ "پاسخ می دهد. باید توجه داشت تمامی سرویس هایی که در مرحله تجزیه دامنه مشخص شدند برای پیاده سازی مناسب نیستند، برای پیدا کردن زیر مجموعه ای از این سرویس ها در روش تست لیتموس ، براساس موارد مختلف از جمله تطابق سرویس با نیازهای حرفه ، تکراری نبودن سرویس و ... تصمیم گیری می شود .منظور از تکراری نبودن سرویس این است که وظایفی که سرویس در دست بررسی انجام می دهد توسط سرویس های دیگر پوشش داده نشوند.

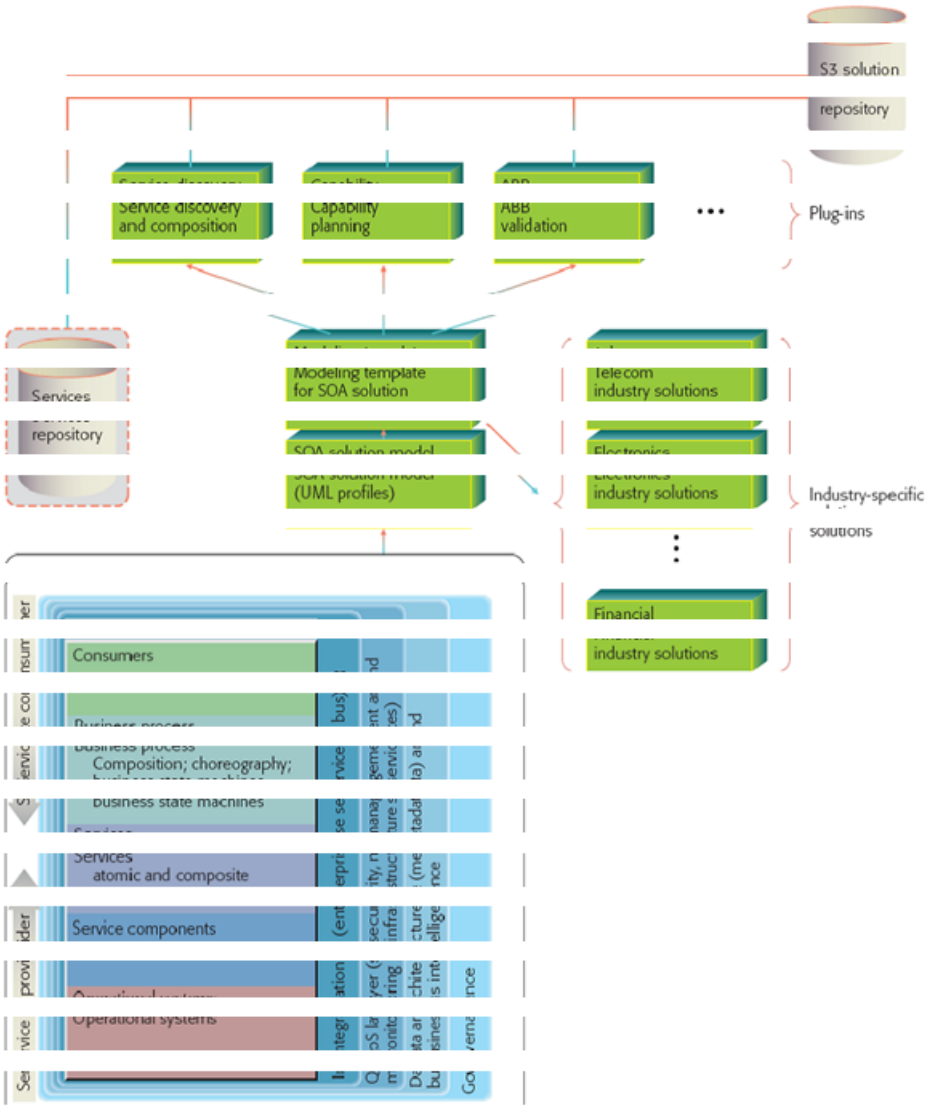
د) حمایت از تولید خودکار مستندات حرفه ای و فرآورده های کد

هـ) واسط کاربری گرافیکی با موتور قابل سفارشی شدن برای یازمندیهای جدید کسب و کار

و) ماژول (واحد) اعتبارسنجی برای بررسی صحت و کمال ماژولهای بدست آمده از طریق قواعد اعتبارسنجی مبتنی بر تجربیات موفق .

اگرچه دراین روش ادعا شده است که دریافت نیازمندیها، شناسایی سرویس ها و تولید مستندات فنی ، به صورت خودکار درآمده اند، این ابزار بدلیل استفاده از روش توصیی ف SOMA و الگو های طراحی ی م تواند در رده چارچوبهای نیمه خودکار قرار بگیرد. همچنین این چارچوب راه حلی جهت پشتیبانی از تغییرات در سطح مدلهای کاری یشنهاد نداده است .

در شکل ۱,۳ مولفه های کلیدی SOMA-ME نشان داده شده اند:



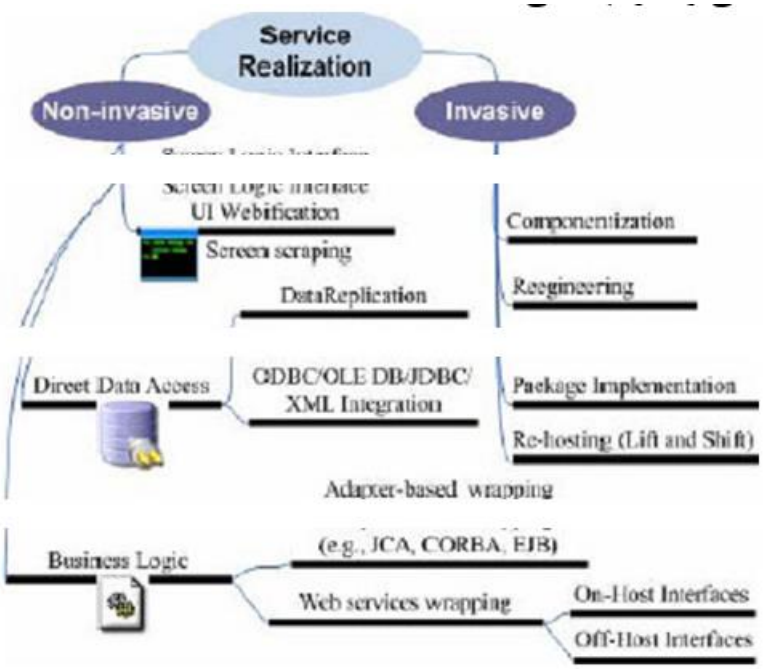
در طراحی سرویس گرای [۱۲] سرویس های با دانه بندی مناسب ، اتصال سست و مستقل ، واسط ها را به اشتراک می گذارند و به جزییات پیاده سازی کاری ندارند، سازگاری سرویس نیز به کمک تعریف سیاست هایی تعیین می شود.

فازهای ارائه شده در [۱۲] روش عبارتند از:

- Information elicitation
- Service identification
- Service definition
- Service realization
- Roadmap & Planning

در فاز عیی ن سازی آن بیشتر به روش پایین به بالا روی آورده و اهداف روش را گذار از معماری قدیمی به معماری جدید به کمک استفاده مجدد،ساخت ، برون سپاری و ... دانسته است . بدین ترتیب کارهای اصلی در این روش ایجاد تطابق و به روز رسانی و تغییرات متناسب در سیستم های موجود برای منطبق ساختن آن ها با سرویس های توصیف شده و سرویس های جدیدی ست که تعریف می شوند، برای این انطباق و به ویژه با سرویس های جدید لازم است گاهی مولفه های جدیدی تعریف شوند و بخش هایی از کد تغییر یابند و یا کد های جدیدی به مولفه ها اضافه شوند.

در این روش در فاز عیی ن سازی دو روش تهاجمی ۱۳ و غیر تهاجمی ۱۴ آمده است ، در روش غیر تهاجمی با ایجاد لفافه ۱۵ ها و میان افزارهایی از مولفه های موجود مطابق با سرویس ها و نیازهای جدید استفاده می کنیم ، در روش تهاجمی نیز به کمک مهندسی مجدد۱۶، ساختاردهی مجدد۱۷، توسعه مجدد۱۸ و بازسازی ۱۹، مولفه های موجود را با سرویس ها منطبق می کنیم . با توجه به تعریف این دو روش می بنی یم که به ترکیی ب از آن ها برای فاز non invasive محقق سازی در این روش نیازمندیم . در شکل ۳,۳ جزییات فاز محقق سازی این روش را نشان می دهد که در آن دو روش تهاجمی و غیر تهاجمی آمده است .



شکل ۳.۳ فاز محقق سازی سرویس [۱۲]

ورود SOA به عرصه مهندسی نرم افزار و پیاده سازی راه حل های مبتنی بر سرویس منجر به تطابق اهداف حرفه و فناوری اطلاعات سازمان ، افزایش اتصال سست و افزایش قابلیت استفاده مجدد در سازمان شد. سهم اصلی [۱۳] بررسی مبانی SOA برای مشخص سازی اطلاعات در آن از طریق نمایش اطلاعات به عنوان سرویس ها(IaaS۲۰) و بحث روی چالش های محقق سازی و مدل سازی سرویس های اطلاعاتی در چارچوب SOMA می باشد. بدین ترتیب در سه فاز شناسایی ، مشخصه سازی و محقق - سازی ، یازمند تغییرات متناسب هستیم تا بتوان شناسایی و مدل سازی "اطلاعات به عنوان سرویس ها"

با وجود این که به کارگیری مفاهیم SOA بسیار در سازمان ها فراگیر شده است ، ولی تمرکز فعالیت های SOA بر روی فرآیندهای حرفه می باشد که باعث عدم رسیدن به یک پیاده سازی موثر در سازمان می شود ولی با به کارگیری IaaS می توان به افزایش قابلیت استفاده مجدد، اتصال سست و افزایش کیفیت SOA سازمان رسید. برای تحقق چنین امری (استفاده از IaaS در (SOAنیازمند تغییر برخی راهکارها و توسعه متدولوژی های موجود در SOA هستیم که بتوان به تحقق و عینی - سازی IaaS ها پرداخت .

در این روش استفاده کنندگان از سرویس های اطلاعاتی و داده ای به جزییات داده ای و پیاده - سازی این سرویس ها نمی پردازند، در حقیقت مهم نیست که سرویس اطلاعاتی ، داده مورد نیاز ما را به چه طریی ق تامین می کنند و این امکان تغییر در لایه های پایین پیاده سازی این سرویس ها را می دهد.

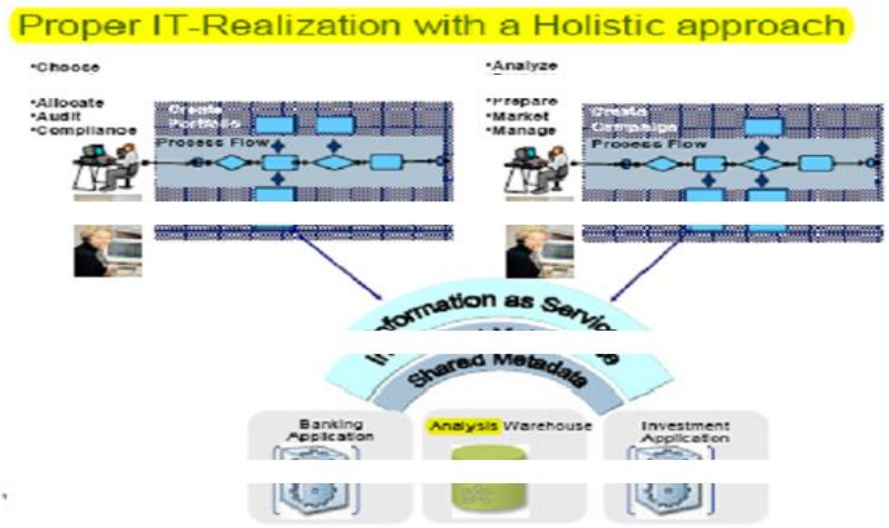
جدایی منطق حرفه از منطق دسترسی به داده ها، منجر به افزایش قابلیت استفاده مجدد از سرویس های داده ای سازمان می شود.

۳,۱,۳ برخی مزایای استفاده از IaaS

- اتصال سست بین مدل های داده ای و داده های ذخیره شده (باعث تسهیل در تغییر و تجمیع داده ها می شود)
- استفاده مجدد از منطق دستیابی به داده
- پشتیبانی از امکان نظارت بر داده ها

- فراهم آوردن امکان توسعه سرویس های داده ای به کمک DBA (به طوریکه به کمک ابزار سطح بالای آن امکان مشخص سازی اینکه چه عناصر داده ای به کمک کدام سرویس پیاده سازی می شوند، میسر باشد

نمایش محقق سازی سرویس ها به کمک IaaS برای یک مثال خاص در شکل ۴,۳ مشخص شده است



شکل ۴,۳

پیاده سازی بالا منجر به چالاکی در فرایند و داده شده است . استفاده از IaaS در روش SOMA منجر به تغییرات و توسعه هایی در مدل سازی و محقق سازی SOA سازمان می شود. طبق [۱۳] به این سه فاز علاوه بر فعالیت های گفته شده در SOMA ، فعالیت های جدیدی اضافه می شود.

۲,۳ روش های یاده سازی معماری سرویس گرا

۱,۲,۳ انتقال مدل رانه کسب و کار

به جای محوریت فعالیت ها بر روی موجودیتها تمرکز کرده است یعنی این پارادایم از فناوری ساخت نرم افزار به صورت مدل رانه برای تولید مؤلفه های سرویس کسب وکار از مدلهای فرآیند کسب و کار سطح بالا استفاده می کند. در این پژوه [۱۸] از تکنیک تحلیل چرخه حیات موجودیت کسب و کارBELA۲۱ برای تحقق راه حل معماری سرویس گرا مبتنی بر MDBT استفاده شده محوریت موجودیتها فرآورده های خود را تولید و توسعه می دهد نویسندگان مقاله این فرآیند را بر روی پروژه های با سطح بسیار بزرگ در صنایع مختلف انجام داده اند و بهبودهای بسیار خوبی را در چند پارامتر مدیریت پروژه رصد کرده اند که از آن جمله می توان به ۴۰ درصد کاهش در هزینه و ۲۰ درصد کاهش در زمان چرخه موقعیکه با رهیافتهای مرسوم SOA مقایسه می شود، اشاره کرد . چون پارادایم BELA بر روی موجودیتها تمرکز دارد بنابراین رفتار هر موجودیت اطلاعاتی با چرخه حیاتش با استفاده از ماشین حالت محدود (FSM) نشان داده می شود و کدی که رفتار مؤلفه را نشان می دهد از مدل حالت ماشین تولید می شود از این رو این رهیافت بطور مؤثری چابکی کسب و کار را افزایش می دهد. این پژوهش بر روی موجودیتها تأکید بسیاری دارد در حالیکه این امر مزایا و معایی ب دارد که به نظر نمی توان از معایب آن به راحتی چشم پوشید همچنین در این پژوهش بر روی طراحی واسط کاربر تأکید بسیاری شده است و یکی از مسائل نیز حل مشکل طراحی واسط کاربر پیچیده است . از سوی دیگر در طراحی مدل رانه تأکید بر تحلیل موجودیتهای کسب و کار بوده است یعنی ستنز راه حل مدل رانه با توجه به تحلیل موجودیتهای کسب و کار صورت گرفته

است و به جای آنکه فرآیند کسب و کار مدنظر قرار گیرد موجودیتها مهم شمرده شده اند. در این پژوهش تحلیل آماری مناسبی صورت گرفته است اما از متریک هیچ سخنی به میان نیامده است .

۳.۲.۲ روش های پیاده سازی در خصوص M4SOD

SOA در سازمان است که در آن یک مدل مفهومی به منظور شناسایی و اتخاذ تصمیمات محقق سازی معرفی می شود. در این روش روال ها و تکنیک هایی براساس موارد کاربری و تحلیل دامنه حرفه ارائه می شود. این روش یک روش نیمه خودکار است و دارای دو فاز شناسایی و محقق سازی مدل سرویس می باشد، در فاز محقق سازی سرویس ، این روش دارای سه استراتژی توسعه سرویس ، نگاشت سرویس توسط سیستم های موجود و تغییر و انطباق سرویس های مشابه می باشد. در این فاز ارائه روشی برای ارزیابی هر سرویس و اتخاذ یک استراتژی مناسب به کمک تحلیل هزینه فایده انجام می شود.

محقق سازی در این روش به کمک ارزیابی سرویس ها انجام می شود. ارزیابی در واقع توافق میان ذینفعان به کمک امتیاز دهی در خصوص معیارهای مختلفی ست . به این منظور از ذینفعان درخواست می شود تا نظر خود را در خصوص معیارهای مختلف در قالب ۵ مرتبه (LL, L, M, H, HH) اعلام کنند. ارزیابی سرویس ها در این روش بیان کند :

گام اول : ابتدا باید به نحوه پیاده سازی سرویس ها با توجه به بازگشت سرمایه آن ها پرداخت . با تعریف معیارهای ارزیابی می توان به بررسی نحوه پیاده سازی سرویس ها از نظر این معیارها پرداخت . برخی از این معیار ها عبارتند از : اهمیت ، مانایی ، ریسک و فرکانس استفاده .

گام دوم امتیاز دهی به سرویس ها توسط ذینفعان ، نمونه آن را در جدول ۱,۳ مشاهده می کنید:

	Importance (20)	Persistency (40)	Risks (20)	FOU (20)
Stk 1	HH	H	M	H
Stk 2	H	H	LL	HH
Stk 3	H	M	L	M
Stk 4	LL	H	M	H
Stk 5	M	M	L	H

همانگونه که در جدول ۱،۳ می بینید به هر یک از معیارها وزنی از نظر درجه اهمیت آن برای یستم اختصاص داده می شود و ذینفعان نیز نظر خود را در قالب دامنه (LL, M,H, HH,L) اعلام می کنند. برای نسبت دهی وزن ها نیز در این روش از نظر ذینفعان استفاده می شود.

گام سوم : ارزیابی میزان تطبیق به کمک ضریب کندال ، که ضریبی برای اندازه گیری ترتیب میان قضاوت های مختلف در ارزیابی یک پدیده است .R به صورت زیر محاسبه می شود:

$$R = (P-Q).(P+Q)$$

به طوریکه P تعداد زوج های منطبق با هم و Q تعداد زوج های نامنطبق باشد. در صورتی که این ضریب محاسبه شده از مقدار درجه تطبیق مورد نظر کم تر باشد، گام دوم و سوم باید تکرار شود.

گام چهارم : جستجوی سرویس ها ست در این گام مقدماتی به منظور تصمیم گیری برای انتخاب یک استراتژی مناسب اندیشیده می شود. کارهایی مانند تخمین هزینه و زمان توسعه سرویس ، لیستی از مولفه های قابل نگاشت به سرویس ها به همراه توصیف آن ها در این گام انجام می شود. منظور از مولفه های قابل نگاشت در این مرحله مولفه هایی هستند که می توان آن به کمک آن ها پیاده سازی کرد. تهیه این لیست می تواند در مرحله بعد به ما در مورد انتخاب بین دو تصمیم توسعه اولیه و یا استفاده از سیستم های موروثی قدیمی در مورد سرویس بینجامد.

روش های شناسایی مولفه ها

شناسایی مولفه ها در یک سازمان به کمک شناسایی فرآیند و موارد کاربری تسهیل می شود، نکاتی که در شناسایی مولفه ها باید مورد توجه قرار گیرد چسبندگی ، اتصال ، استقلال ، قابلیت نگه داشت ، ریزدانگی و خوش واسط بودن آن هاست .

در [۱۶] به روشی بر مبنای گراف ها برای شناسایی مولفه های با ریز دانگی مناسب ، اتصال سست و چسبندگی بالا پرداخته می شود.

مراحل روش به شرح زیر است :

تعریف معماری : در معماری یستم ، از یک دید سطح بالا به آن نگاه می کنیم . معماری یک مکانیزم تجمیع برای گردهم آوردن مولفه های نرم افزاریست که به لایه بندی و مشخص سازی زیر سیستم ها می پردازد و عدم مطابقت هایی را که ممکن است میان مولفه ها پیش بیاید محدود کرده است . علاوه بر این قابلیت استفاده مجدد در یک طراحی معماری قوی افزایش می یابد. بر اساس علل فوق هنگام شناسایی مولفه ها باید ابتدا به معماری یستم توجه نمود. درمعماری لایه ای زیر سیستم ها و ارتباطاتشان نمایان می شود. با تقسیم سیستم به زیرسیستم ها، کار شناسایی مولفه ها از شناسایی در سطح کل سیستم به شناسایی در سطح مولفه ها (که دارای کلاس های کوچکتر هستند) کاهش می - یابد. همچنین اگر معماری خوش تعریف باشد وابستگی زیر سیستم ها کم و در نتیجه وابستگی مولفه ها کاهش می یابد. چسبندگی مولفه ها به کمک یافتن کلاس های کلیدی : با یافتن کلاس های کلیدی در

این مرحله می توان مولفه ها را تشخیص داد و سپس کلاس های کمکی را به آن ها اضافه کرد.

شناسایی کلاس های کلیدی به کمک موارد کاربری باید ابتدا به شناسایی توابع موجود در موارد کاربری بپردازیم . توابع را می توان درون عملیات موارد کاربری و یا به کمک ارتباط بین عامل و سیستم در موارد کاربری تشخیص داد. توابع شناسایی شده را می توان به سه دسته تقسیم کرد: توابع کلیدی ، کمکی و انتخابی . سپس کلاس ها را بر مبنای نوع توابع ، دامنه دانش و مسئولیتشان به ۵ دسته تقسیم می کنیم .

کلاس های نوع A : کلاس های کلیدی که دارای منطق قوی حرفه هستند.

کلاس های نوع B : کلاس های کمکی که در کنار کلاس های کلیدی اجرا می شوند.

کلاس های نوع C : دارای منطق سبک حرفه هستند.

کلاس های نوع D : کلاس های مرزی .

کلاس های نوع E : کلاس های دارای کم ترین اهمیت .

اتصال بین مولفه ها: اگر مولفه ها باهم پیام ردوبدل کنند با هم اتصال دارند. اتصال مولفه ها به تعداد پیام هایی که با هم ردوبدل می کنند نیز گفته می شود. برای این منظور می توان تعداد متدهایی که فراخوانی می شوند را در نظرگرفت .

نوع دیگری از اتصال که اتصال ایستا نام دارد به وابستگی ، ترکیب و وراثت و تجمیع کلاس ها برمی گردد. که در این روش به وابستگی دو سویه وزن ۴، یکسویه وزن ۳، تجمیع وزن ۵، ترکیب وزن ۶، رابطه پدر فرزندی وزن ۲ نسبت می دهیم.

واسط مولفه : واسط مولفه نمای یرونی یک مولفه است که مشخص کننده عملیاتی ست که مولفه قادر به انجام آن است . تصمیم این که دو موردکاربری در یک مولفه قرار میگیرند یا جدای از هم به کمک برنامه نویس تعیین می شود.

الگوریتم خوشه بندی مولفه ها:

در این روش کلاس ها و روابطشان گرافی با وزن های مختلف را تشکیل می دهند ، سپس بر اساس وزن کلاس ها و به کمک رابطه هم ارزی ، گراف خوشه بندی شده و در نهایت هر خوشه به یک مولفه تبدیل می شود. برای تعیین وابستگی یا عدم وابستگی کلاس ها و یا میزان آن ها باید بتوان یک d تعیین نمود، که این d اگر از مقدار ثابتی کمتر نباشد دو کلاس در یک خوشه و در نهایت در یک مولفه قرار می گیرند. به همین دلیل ریزدانگی مولفه ها برمی گردد به مقدار ثابتی که خودمان انتخاب می کنیم . علاوه بر این به رابطه کلاستر می توان محدودیت های دیگری مانند محدودیت دامنه را نیز افزود.

برای تعیین مقدار d میتوان به ویژگی های مولفه های خوش تعریف توجه نمود، برخی از این ویژگی ها از این قرارند:

- فرآیند های حرفه درون مولفه ها نهان باشند
- مولفه های مانا در سطوح پایین تری نسبت به مولفه های نامانا قرار گیرند
- وابستگی مولفه ها به حداقل برسد

پس از شناسایی مولفه ها از بین آن ها خوش تعریف ترین ها راشناسایی کرد، برخی معیارهای انتخاب به شرح زیرند:

- مولفه ها باید فرآیندهای حرفه را در خود جای دهند.
- مولفه ها باید مانا باشند تا در سطح پایین تری در سیستم جای گیرند.

- ارتباط بین مولفه ها باید کمینه شود.
- تعداد کلاس های درون مولفه ها باید به حدی باشد که بتوان آن ها را مدیریت کرد.

در [۱۷] به شناسایی در قالبی نزدیک تر به بحث SOA نگریسته است . و سوالات زیر را مطرح نموده است :

- چه سرویس هایی باید به عنوان سرویس های عمومی نمایانده شوند تا توسط مشتری و شرکای حرفه در دسترس قرار گیرند؟
- مولفه های سازمانی با ریزدانگی بزرگ ، چگونه شناسایی و توصیف می شوند؟
- مولفه های شناسایی شده چگونه طراحی می شوند؟

در این مقاله اشاره شده است که تکنیک های

- کارت های CRC و ...
- مورد کاربری

۲۳

- تحلیل و طراحی مبتنی بر مسئولیت

روشی که در [۱۷] به آن اشاره میشود با نظر گرفتن اهداف حرفه ، پر کردن گپ اهداف به کمک فناوری اطلاعات است . این روش دارای دیدی انتزاعی تر از روش های قد می ست .

به طور کلی در پروژه های بر مبنای مولفه ها یکی از اهداف ، پر کردن گپ بین اهداف حرفه و نیازمندی های فناوری اطلاعات می باشد. به این صورت که در قالب یک روش مبتنی بر هدف از فرآیندهای سازمانی شروع کرده و با یک تجمیع بر اساس سرویس ها و مولفه های معماری نرم افزار سازمان ، به سمت پر کردن گپ موجود و کم تر کردن عدم تطابق معنایی اهداف و فرآیندها و سرویس ها و مولفه های سازمان بپردازیم .

در [۱۷] برای پرکردن این کمداشت و فضای خالی به دو فعالیت تحلیل حرفه مدل رانه پرداخته است : اولین فعالیت تحلیل حرفه تعریف مرزهای مولفه ها بر حسب فرآیندهای حرفه است که به تجزیه دامنه حرفه معروف است . این گام به کمک تعریف فرآیندهای حرفه و موارد کاربری سطح بالا قابل انجام است . دومین گام تحلیی ل ، ردیابی سرویس های فناوری اطلاعات به اهداف حرفه سازمان است . البته باید توجه داشت که در این روش ما اهداف حرفه را به زیر اهدافی تقسیم می کنیم و کار نسبت دهی سرویس ها به مولفه ها هنگامی تسهیل می - شود که بدانیم هر سرویس چه اهدافی از حرفه ما را می پوشاند. برای انجام فعالیت دوم در این روش از یک مدل هدف استفاده می شود و برای رسیدن اهداف تعریف شده ، اهداف را به زیر اهدافی تقسیم میکنیم و برای رسیدن به آن ها نیز سرویس ها یا مجموعه ای از سرویس ها که آن اهداف را محقق می کنند، تعیین می کنیم . و در نهایت سرویس ها هم به مولفه هایی که آن ها را محقق می سازند، نگاشت می شوند.

پس از آشنایی کلی با مولفه های خوش تعریف و نحوه شناسایی ها درون سازمان ، می توان ایده های بالا را در چارچوب SOA پیاده کرد. در [۱۸] به بررسی متریک هایی برای شناسایی کدهای قابل استفاده مجدد پرداخته است . در این گزارش به ذکر سه جدول کلیدی برای شناسایی این کدها بسنده می کنیم :

در جدول ۲,۳ به ویژگی های که یک کلاس باید داشته باشد تا احتمال استفاده مجدد آن بالاتر رود پرداخته شده است ، در ستون دوم جدول نحوه اندازه گیری ویژگی ذکر شده آمده است :

کمیت سنج	ویژگی های کلاس
اتصال بین اشیا کلاس (کلاس های خارجی)	اتصال (کلاس های خارجی)
اتصال بین اشیا کلاس (کلاس های کتابخانه ای)	اتصال (کلاس های کتابخانه ای)
تعداد متد ها	تعداد متدها
تعداد صفات	تعداد صفات
رتبه دهی از ۱ تا ۵	اسم های بامعنی
تعداد کلمات موجود در متن مستند	مستندسازی
تعداد خطوط کد غیر از خطوط توضیحی	تعداد خطوط
تعداد خطوط توضیحی به ازای هر خط	خطوط توضیحی
عمق درخت ارث بری	عموق ارث بری
تعداد فرزندان	تعداد فرزندان
کمبود چسبندگی در متدها	چسبندگی
تعداد متغیرهای کلاس به ازای هر خط کد	متغیر های کلاس

جدول ۲.۳ معیارهای کلاس

در [۱۹] به روشی ارائه شده که با کمک هوش سازمانی به شناسایی مولفه های سازمان و اتوماسیون فرآیندهای حرفه می پردازد. در روش های قدیمی تضمیی ن برای مناسب بودن کامل ، جامع و مانع بودن مولفه های شناسایی شده (کمبود مولفه ها، افزونه نبودن و ریزدانگی ...) وجود ندارد و این ها پارامترها یست که این روش به کمک هوش سازمانی ادعای لحاظ کردن و حل مشکلات روش های قدیمی را دارد. در [۲۰] با ارائه یک مدل دانش در چارچوب سازمانی به شناسایی ، فراخوانی و ترکیب سرویس های سازمان مطابق با نیازهای کاربران می پردازد. در این مقاله به ترکیب هستان شناسی با مفاهیم Web service ها می پردازد.

دیگر روش ها

این تحقیق به استفاده از روش های رسمی در ساخت نرم افزار ها می پردازد، در این مقاله به یک استراتژی برای محقق سازی سرویس بر مبنای فرمال مدل ها اشاره شده است .(برای نگهداشت اطلاعات معنایی قرارداد سرویس را معرفی ی م کنیم) به خاطر عمومی بودن ، این روش در سیستم های م یاس وسیع کاربرد دارد.

پرداخته شده است ، بدین صورت که آن را به دو لایه تقسیم کرده است ، SOAدر مقاله [۲۱] به لایه بندی لایه بالایی به آرایشات سرویس و ارتباط سرویس ها با هم و دیگر برنامه ها و به اصطلاح به توصیف سرویس ها می پردازد و در لایه پایین تر به محقق سازی سرویس ها و چگونگی ارتباط سرویس ها با مولفه ها می پردازد. ممکن است از ترکیب برخی سرویس ها به مولفه هایی برسیم

و از طرفی برای محقق سازی یک سرویس خاص ، چند مولفه را با هم ترکیب کنیم . شکل زیر لایه

بندی چارچوب را برای رسیدن به محقق سازی نشان می دهد :

wsdlدر این مقاله همچنین به بحث امکان ترکیب مولفه ها و سرویس ها پرداخته و توصیف سرویس به کمک را به عنوان یکی از راهکارهای محقق سازی مطرح ساخته است .UMLو سپس تبدیل آن به

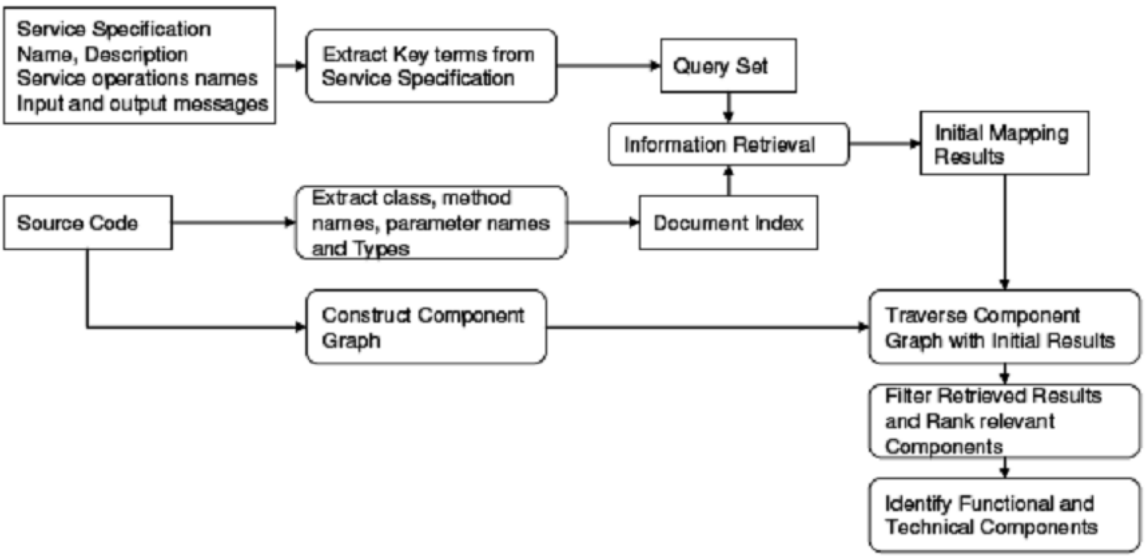
بخش مدیریت مولفه ها نیز به مبحث پیاده سازی مولفه ها ، امکانات ترکیب آن ها و ارتباط آن ها با توصیف سرویس ها می پردازد. و در ادامه به بررسی روش های تعریف و قضایای مربوط به توصیف و محقق سازی

2.4.3 Locating components realizing services in سرویس ها به گونه ای فرمال پرداخته است .

exisitind systems

روش کار این مقاله مربوط به تحلیل پایین به بالا می باشد و روش با این پیش فرض جلو می رود که ما دارای یک سری مولفه هایی هستیم و حال می خواهیم از آن ها استفاده کرده به سرویس ها نگاشتشان کنیم و یا با ترکیبشان به مولفه های جدیدی برسیم . به همین منظور در این مقاله یک روش نیمه خودکار و ایستا ارائه شده که سه گام دارد: بازیابی لینک های ین سرویس و مولفه کد منبع ، فیلتر کردن لینک هایی که مولفه های ساختاری ایستا دارند و در نهایت دسته بندی شان به مولفه های تابعی و تکنیکی .

شکل ۷,۳ نحوه کلی کار را نشان میدهد:



شکل ۷,۳

نحوه استقرار مولفه هایی که سرویس ها را محقق میکنند

و در نهایت به کمک گراف مولفه ها و به دست آوردن برخی متریک ها به ارزیابی نتایج کار در این مقاله می - پردازد.

3.4.3 Realization of personal mobility services

در این مقاله معماری TINA به عنوان یک معماری تضمینی برای محقق سازی سرویس های تغییرپذیر ارائه شده است به طوریکه در دامنه های مختلف قابل استفاده باشد.

۳,۵ مقایسه روش ها

همان طور که در بخش پیشین گفتیم روش ها را به سه دسته تقسیم کردیم ، دسته اول روش های مبتنی بر چارچوب SOMA که عبارت بودند از SOMA-ME و SOAF که هر دو راه حل های تجویزی و مبتنی بر دامنه ای داشتند، که از نظر درجه خودکارسازی ، روش SOMA-ME از درجه خودکارسازی بالاتری برخوردار بود، درحقیقت نیمه خودکار بود و مستندات خود را به صورت خودکار تولید می کرد. ولی در چارچوب SOAF در مورد محقق سازی سرویس ها ، صرفا توضیحات متنی جهت وضوح مسئله آمده است و نه دستورالعمل هایی اجرایی . سایر روش های یاده سازی معماری سرویس گرا نیز یکی روش M4SOD و دیگری روش MDBT ست که روش اول روشی فازبندی و پشت سر هم است که با وجود تمام خودکار نبودن و دخالت ذینفعان نتایج قابل قبولی در سازمان به دنبال دارد و شباهت هایی با روش های SOMA یIBM دارد. این روش رویکردی مدل رانه دارد ولی از فاز های سه گانه مدلسازی سرویس در آن به صورت صریح سخنی نیامده است .

۶,۳ نتیجه گیری

در این فصل به دسته بندی روش های موجود جهت تحقق سرویس در مدلسازی سرویس گرا پرداختیم .

پس از دسته بندی و مروری بر روش ها به مقایسه آن ها باهم پرداختیم . روش های ارائه شده در این زمینه اکثرا به صورت تجویزی بوده و مبتنی بر دامنه ای خاص می باشند و عمومی سازی و خودکارسازی آن ها میسر نیست . در این میان روش های شناسایی مولفه ها روش های فنی تر و مبسوط تری هستند ولی روش هایی که به پیاده سازی معماری سرویس گرا در سطح سازمان برای مثال ها و نمو نه های موردی خاص می پردازند از ارزش کمتری برای هدف ما برخوردارند.

روش های موجود در این فصل به سه دسته کلی تقسیم شدند، دسته اول روش های مبتنی بر چارچوب SOMA که از جمله آن ها سه روش SOMA-ME و IaaS هستند. روش SOMA-ME یک توسعه روی

SOMA ست ، که برای هر سازمان با توجه به دارایی های موجود و دامنه حرفه آن سازمان ، راه حل سرویس گرای سازمان را در قالب سرویس های مرکب ارائه می کند. چارچوب SOAF نیز بر پایه SOMA بنا نهاده شده و یک راه حل سیستماتیک و تکرار پذیر ارائه می دهد. این چارچوب برای فاز محقق سازی دو روش تهاجمی و غیرتهاجمی ارائه داده است که در نهایت منجر به توسعه سرویس های جدید و یا تطبیق سرویس های موجود با سرویس های شناسایی شده جدید می شود. در روش IaaS به اطلاعات و داده های سازمان به عنوان سرویس پیاده سازی می شوند ، مدلسازی اطلاعات در این روش نیازمند تغییراتی در فازهای شناسایی ، مشخص سازی و محقق سازی سازمان می شود. این تغییرات عبارتند از تحلیل و مدلسازی اطلاعات در فاز شناسایی ، توصیف اطلاعات در فاز مشخصه سازی و رسیدگی به داده های سازمان ، مدیریت داده و تولید محصولاتی چون مدل داده منطقی و فیزیکی در فازهای محقق سازی و پیاده سازی . دسته دوم روش های این فصل چارچوب هایی هستند که معماری سرویس گرا را در یک سازمان پیاده میکنند و لزوما مبتنی بر SOMA نیستند، از جمله آن ها روش M4SOD ی م باشد که دارای دو فاز شناسایی و محقق سازیست . در این روش برای فاز محقق سازی یک روش الگوریتمیک اراده شده که بر مبنای نظر ذینفعان پروژه در مورد نحوه پیاده سازی سرویس ها تصمیم - گیری میکند و استراتژی مناسب را انتخاب می کند. دسته سوم روش های شناسایی مولفه ها هستند ، این روش ها اکثرا مولفه گرا بوده و می توان با ایده گرفتن از آن ها، آن ها را در حیطه معماری سرویس گرا به کاربرد.

دسته چهارم روش هایی هستند که صرفا به محقق سازی سرویس می پردازند. این روش ها از اهمیت کمتری برخوردارند ، فقط در یکی از آن ها اشاره ای به روش های فرمال محقق سازی سرویس شده است که استفاده از این روش در پیاده سازی راه حل نهایی می تواند سودمند باشد.

در این تحقیق پس از بررسی روش های موجود به نتایج زیر دست یافتیم :

- پیش بینی بخشی در راه حلمان به منظور شناسایی و توصیف مولفه ها (برای این شناسایی ی م توان از ایده های [۱۶] و[۱۷] استفاده کرد.
- استفاده از روش های مربوط به اتخاذ تصمیمات خودکار در زمینه پیاده سازی سرویس ها ، از چارچوب

M4SOD و روشی که در آن جهت محقق سازی ارائه شده می توان در این زمینه ایده گرفت . علاوه بر این روش های دیگر نیمه خودکار در زمینه اتخاذ تصمیمات در سطح معماری یز قابل بررسی اند.

فصل چهارم ویژگی های راه حل های مورد انتظار

مقدمه

در فصل قبل راه حل های موجود در زمینه محقق سازی مدل سرویس را بررسی کردیم در این فصل با ارائه نقاط ضعف روش های موجود و ایده گرفتن از نقاط قوت و انتخاب گزینه های مناسب و بررسی ایده های آن ها، دور نما و ویژگی های کلی راه حل خود را با نام روش خودکار محقق سازی سرویس یا Automated Service Realization(ASRM) Methodارائه می دهیم .

۱,۴ نقاط ضعف روش های موجود

شیوه های ارائه شده جهت محقق سازی سرویس ها به صورت تجویزی بوده و خودکارسازی آنها میسر نیست و تحلیلگر می بایست آنرا شخصاً به انجام برساند، این شیوه ها در محدوده های با مقیاس وسیع نظیر سازمان دچار مشکلات عملیاتی نظیر عدم بهینه بودن محصولات تولیدی با توجه به معیارهای مربوطه است .

این روشها قادر به پشتیبانی از تکامل مدلها۲۴ با توجه به تغییرات بوجود آمده در نیازمندیها نیستند و ارائه راه حلهای ب ینه وابسته به مهارت تحلیلگر در استفاده از این شیوه ها است .

۲,۴ ویژگی های راه حل مورد انتظار

تعریف فاز محقق سازی در این تحقیق بدینگونه بیان شد که این فاز دارای دارای دو بخش پشت سر هم شناسایی و توصیف مولفه ها و گرفتن تصمیمات پیاده سازی می باشد، ما نیز در ارائه راه حل خود به این دو بخش به صورت مجزا نگاه خواهیم کرد.

۱,۲,۴ شناسایی و مشخصه سازی مولفه ها

شناسایی مولفه ها در این بخش باید به کمک روشی که ارائه خواهیم داد انجام شود، ولی ورودی و خروجی این بخش چیست ؟ و راه حل مورد نظر در این بخش باید چه ویژگی هایی را برآورده کند؟

ورودی این بخش مدل سرویس است که خروجی فاز دوم ، یعنی توصیف مولفه هاست ، در این بخش می توان به کمک ماتریس CRUD در چارچوب ASOMF به ارائه روشی برای شناسایی مولفه ها پرداخت ، ورودی هایی که از روش ASSM

چارچوب ASOMF در یافت می کنیم عبارتند از: عملیات بین سرویس ، نمودار تعامل سرویس ها، افرازها، در حقیقت پس از مشخصه سازی مدل سرویس ، برای هر سرویس دهنده که هیچیک از سرویسهای آن به دروازه ها ارتقاء نیافته اند یک مولفه معین می کنیم . برای سرویس دهندگانی که حداقل یکی از سرویسهای آنها ارتقاء یافته است ، سرویسهایی که ارتقاء نیافته اند همه روی یک مولفه مجزا محقق می شوند و هر سرویس ارتقاء یافته نیز روی یک مولفه مجزا مختص خود محقق

می شود. اما مولفه های شناسایی شده توسط این روش باید چه ویژگی هایی را برآورده کنند؟ هدف از روش آن است که مولفه های شناسایی شده تا حد ممکن دارای اتصال سست و چسبندگی بالا باشند.

۲,۲,۴ تصمیمات محقق سازی

در این بخش از راه حل لازم است تصمیمات پیاده سازی اتخاذ شود، راه حل ارائه شده در این بخش باید به گونه ای باشد که با خودکارسازی باعث پررنگ شدن نقش معمار شود و تصمیمات اتخاذ شده در این مرحله از نظر اقتصادی به صرفه باشد.

ورودی های این مرحله مولفه ها و سرویس های شناسایی شده و خروجی آن تصمیماتی ست که روی چگونگی پیاده سازی سرویس ها گرفته می شود، تصمیماتی نظیر اینکه یک سرویس توصیف شده در مدل سرویس با دارایی های موروثی سیستم پیاده شود ویا از نو توسعه داده شود.

در نهایت این بخش از راه حل به خودکار سازی فعالیتهای تحلیل سیستمهای مبتنی بر سرویس ، افزایش کیفیت ، افزایش سطح بهره وری ، بهبود یکپارچگی در سطح معماری ، ایجاد مدل سرویس به عنوان مبنایی جهت ساخت راه حلهای باکیفیت مطلوب (پشتیبانی از تکامل مدلها، بهینه با توجه به متریک ها) می پردازد.

نتیجه گیری

در این بخش به بررسی ویژگی های روش پیشنهادی پرداختیم و با تعریف محقق سازی از دیدگاه خود و تقسیم راه حل محقق سازی به دو بخش شناسایی مولفه ها و توصیف آن ها و نیز بخش تصمیمات پیاده سازی و بررسی ورودی و خروجی هر یک از این بخش ها و بیان ویژگی های مورد انتظار هر یک از آن ها ، نمایی از راه حل خود را ارائه دادیم . در زیر بخش اول روش (شناسایی و توصیف مولفه ها) انتظار می رود روش ارائه شده به مولفه های با اتصال سست و چسبندگی بالا به طوری که availability ، Securityو performance خود را حفظ کنند بینجامد و زیر بخش دوم نیز به تصمیمات پیاده سازی سرویس ها می انجامد.

فصل پنجم نتیجه گیری

۵,۱ نتیجه گیری

در ابتدای این تحقیق دورنمای خود را در باب جایگاه کاری که می خواهیم انجام دهیم و موقعیت آن در چارچوب ASOMF، بیان کردیم ، سپس به بررسی جایگاه محقق سازی سازی سرویس ها در چرخه حیاط مدلسازی سرویس گرا و فعالیت های این فاز پرداختیم ، گفتیم که فعالیت های این فاز را می توان به دو دسته شناسایی و مشخصه سازی سرویس ها و فعالیت های تصمیم گیری در حیطه پیاده سازی سرویس ها پرداختیم در ادامه ما به بررسی و مقایسه روش های موجود در این زمینه پرداختیم . روش های موجود هیچ کدام خودکار نبوده و برای فاز سوم الگوریتم ارائه نداده اند، برخی از آن ها مانند SOD۴M روش خود را در قالب گام هایی یان کرده - اند و برخی دیگر نیز مانند SOAF به توضیحات متنی در زمینه محقق سازی پرداخته اند. هدف ما از این تحقیق در نهایت رسیدن به روشی خودکار در زمینه محقق سازی مدل سرویس است که با

چارچوب ASOMF سازگار باشد که برای رسیدن به این هدف سوالات زیر مطرح است : • درجه خودکارسازی یسر برای ادامه این تحقیق چیست ؟

- آیا امکان استفاده از دانش سازمانی برای تصمیمات پیاده سازی سرویس ها وجود دارد؟

منابع

2019-01. Rob van Eijk (UL),Comparing and Aligning Process Representations

2019-02. Emmanuelle Beauxis Aussalet (CWI, UU), Statistics and Visualizations for Assessing Class Size Uncertainty

2019-03. Eduardo Gonzalez Lopez de Murillas (TUE), Process Mining on Databases: Extracting Event Data from Real Life Data Sources

2019-04. Ridho Rahmadi (RUN), Finding stable causal structures from clinical data

2019-05. Sebastiaan van Zelst (TU/e), Process Mining with Streaming Data

2019-06. Chris Dijkshoorn (VUA), Niche sourcing for Improving Access to Linked Cultural Heritage Datasets

2019-07. Soude Fazeli (OUN), Recommender Systems in Social Learning Platforms

2019-08. Frits de Nijs (TUD), Resource-constrained Multi-agent Markov Decision Processes2018-01. Han van der Aa (VUA), Comparing and Aligning Process Representations

2018-09. Felix Mannhardt (TUE), Multi-perspective Process Mining

2018-10. Steven Bosems (UT), Causal Models For Well-Being: Knowledge Modeling, Model-Driven Development of Context-Aware Applications, and Behavior Prediction

2018-11. Jordan Janeiro (TUD), Flexible Coordination Support for Diagnosis Teams in Data-Centric Engineering Tasks

2018-12. Hugo Huurdeman (UVA), Supporting the Complex Dynamics of the Informatio

2018-11. Mahdi Sargolzaei (UVA), Enabling Framework for Service-oriented Collaborative Networks2018-12. Xixi Lu (TUE), Using behavioral context in process mining 2018-13. Seyed Amin Tabatabaei (VUA), Computing a Sustainable Future 2018-14. Bart Joosten (UVT), Detecting Social Signals with Spatiotemporal Gabor Filters 2018-15. Naser Davarzani (UM), Biomarker discovery in heart failure 2018-16. Jaebok Kim (UT), Automatic recognition of engagement and emotion in a group of children2018-17. Jianpeng Zhang (TUE), On Graph Sample Clustering 2018-18. Henriette Nakad (UL), De Notaris en Private Rechtspraak 2018-19. Minh Duc Pham (VUA), Emergent relational schemas for RDF 2018-20. Manxia Liu (RUN), Time and Bayesian Networks 2018-21. Aad Slootmaker (OUN), EMERGO: a generic platform for authoring and playing scenario-based serious games 2018-22. Eric Fernandes de Mello Araujo (VUA), Contagious: Modeling the Spread of Behaviours, Perceptions and Emotions in Social Networks

.