
بسم الله
الرحمن الرحيم

دانشگاه صنعتی اصفهان
دانشکده برق و کامپیوتر

ارائه روشی مبتنی بر مدل برای شناسایی و انتخاب تطبیق‌پذیر سرویس‌ها در معماری سرویس‌گرا

پایان نامه کارشناسی ارشد مهندسی کامپیوتر- نرم‌افزار

سید علیرضا حاجی‌میرصادقی

استاد راهنما
دکتر محمدعلی منتظری

۱۳۹۰

دانشگاه صنعتی اصفهان
دانشکده برق و کامپیوتر

پایان نامه کارشناسی ارشد رشتهٔ مهندسی کامپیوتر سید علیرضا حاجی‌میرصادقی

تحت عنوان

ارائه روشی مبتنی بر مدل برای شناسایی و انتخاب تطبیق‌پذیر سرویس‌ها در معماری سرویس‌گرا

در تاریخ ۱۳۹۰/۳/۹ توسط کمیته تخصصی زیر مورد بررسی و تصویب نهایی قرار گرفت.

۱- استاد راهنمای پایان نامه — دکتر محمد علی منتظری

۲- استاد مشاور پایان نامه — دکتر محمد داورپناه جزی

سرپرست تحصیلات تکمیلی دانشکده — دکتر سید محمود مدرس هاشمی

تشکر و قدردانی

سپاسگزارم

از پدر و مادر عزیزم که محبت‌های ایشان را هیچگاه نمی‌توانم جبران کنم.

سپاسگزارم

از استاد گرامی، دکتر محمد علی منتظری که همواره با راهنمایی‌ها و نظرات ارزشمندشان در طول این تحقیق گره‌گشا بوده‌اند.

فهرست مطالب

چکیده

با گسترش سیستم‌های اطلاعاتی پیچیده که در محیط‌های عملیاتی پویا مستقر می‌شوند، نیاز به سیاست‌های مدیریتی کلان برای مدیریت و کنترل سیستم‌ها بیش از پیش به چشم می‌آید. هم‌چنین چالش‌های مهندسی سیستم‌های نرم‌افزاری جدید از برطرف کردن نیازمندی‌های وظیفه‌مندی و غیروظیفه‌مندی ساده به موارد پیچیده‌تری تغییر پیدا کرده است. مواردی چون مقاوم‌سازی نرم‌افزار در برابر تغییرات ناگهانی در محیط‌های غیرقطعی و تطبیق‌پذیری سیستم در حال اجرا. در راستای برآورده ساختن نیازهای جدید و پوشش چالش-های ذکر شده، سبک معماری سرویس‌گرا ارائه شده است که با معرفی مفاهیم نوینی در مهندسی نرم‌افزار مانند سرویس، ارتباط سست بین سرویسی، واسط‌های خوش‌تعریف و انقیاد پویا سعی دارد یک معماری قابل انعطاف برای کنترل کیفیت خدمات ارائه دهد. به منظور تضمین کیفیت سرویس‌ها در زمان اجرای سیستم، معماری سرویس‌گرا باید خود را در برابر تغییرات ناگهانی محیط عملیاتی و یا درخواست سرویس‌های جدید از سوی سرویس‌گیرندگان تطبیق دهد. در دهه اخیر کارهای زیادی در زمینه معماری‌های سرویس‌گرای خودتطبیق انجام گرفته است. این تحقیق نیز می‌کوشد چهارچوبی مبتنی بر مدل برای به کارگیری قابلیت تطبیق‌پذیری در معماری سرویس‌گرا ارائه دهد، به گونه‌ای که انتخاب سرویس‌ها در زمان اجرا و نیز شناسایی سرویس‌های حرفه در زمان مدل‌سازی با در نظر گرفتن شرایط محیط اجرا و نیازهای مشتریان قابل تطبیق باشد. از این رو در این چهارچوب، مدل جدیدی برای واسط سرویس و مصرف-کننده سرویس در معماری سرویس‌گرا مطرح شده است. هم‌چنین با استفاده از مفهوم حلقه بازخورد، سازوکاری طراحی شده است که اطلاعات زمان اجرا از فاز استقرار و نگهداری معماری سرویس‌گرا به فاز مدل‌سازی و به طور خاص گام شناسایی سرویس‌ها منتقل شود. علاوه بر این یک روش نیمه‌خودکار و مبتنی بر مدل برای شناسایی سرویس‌های کسب و کار معرفی شده است که قادر است با در اختیار داشتن مدل‌های فرآیندهای کسب و کار، موجودیت‌های کسب و کار و نیز با کمک اطلاعاتی که از محیط اجرایی دریافت می‌کند، سرویس‌های مورد نیاز حرفه را شناسایی کند.

کلمات کلیدی: معماری سرویس‌گرا، سیستم‌های نرم‌افزاری خودتطبیق، شناسایی سرویس، انتخاب پویای سرویس‌ها

فصل اول
کلیات تحقیق

۱-۱ مقدمه

امروزه با گسترش سیستم‌های اطلاعاتی پیچیده[۱] که در محیط‌های عملیاتی پویا[۲] مستقر می‌شوند، نیاز به سیاست‌های مدیریتی کلان برای مدیریت، کنترل و هدایت سیستم‌ها بیش از پیش به چشم می‌آید. چالش‌های مهندسی سیستم‌های نرم‌افزاری از برطرف کردن نیازمندی‌های وظیفه‌مندی و غیروظیفه‌مندی ساده به موارد پیچیده-تری تغییر پیدا کرده‌اند. از آن جمله می‌توان به میزان تحمل نرم‌افزار در برابر تغییرات ناگهانی در محیط، میزان انطباق[۳] نرم‌افزار با نیازمندی‌های جدید، عملیات در محیط‌های غیرقطعی[۴] و متفاوت بودن توسعه‌دهندگان اجزای یک سیستم اطلاعاتی اشاره کرد[۱].

در راستای برآورده ساختن نیازهای جدید و پوشش چالش‌های ذکر شده، سبک معماری سرویس‌گرا[۵] ارائه شده است که مبنای آن ساخت برنامه‌های حرفه[۶] از طریق پیوند دادن سرویس‌ها و برنامه‌های مستقلی است که هر یک توسط یک توسعه‌دهنده‌ی مجزا پیاده‌سازی شده است[۲]. معماری سرویس‌گرا با معرفی مفاهیم نوینی در مهندسی نرم‌افزار مانند سرویس، ارتباط سست[۷]، واسط‌های خوش‌تعریف[۸] و انقیاد پویا[۹] سعی دارد یک معماری قابل انعطاف

---

[1] Complex Information Systems
[2] Dynamic Operation Environments
[3] Adaptation
[4] Uncertain Environments
[5] Service Oriented Architecture (SOA)
[6] Business Application
[7] Loose Coupling
[8] Well Defined Interfaces
[9] Dynamic Binding

برای کنترل کیفیت خدمات[1] ارائه داده که در عین حال از سطح انتزاع[2] مناسب برای استفاده در سازمان‌های بزرگ نیز برخوردار باشد. بی تردید در این راه وجود سازوکارهایی برای تضمین کیفیت سرویس‌ها امری ضروری است. علاوه بر آن سیستم‌های نرم‌افزاری سرویس‌گرا به دلیل ماهیت منعطف خود که ناشی از به کارگیری سرویس‌های شخص ثالث[3] می‌باشد، باید همواره در انتظار وقوع تغییرات ناگهانی در ارائه خدمات و یا بروز اختلال در کیفیت خدمات باشند[3]. این موضوع به قدری کلیدی است که منجر شده است تحقیقات گسترده‌ای در سالیان اخیر در زمینه‌هایی چون کنترل پویای کیفیت سرویس، نظارت[4] بر کیفیت سرویس‌ها، تطبیق معماری سرویس‌گرا در زمان اجرا و طرح-ریزی مجدد[5] ترکیب سرویس‌ها آغاز شود. در این پایان‌نامه کوشش شده است روش نوینی برای تطبیق‌پذیر کردن معماری سرویس‌گرا ارائه شود تا در نهایت کیفیت سرویس‌های خروجی افزایش یابد.

۱-۲ طرح مسئله

سبک معماری سرویس‌گرا بر روی ترکیب پویای سرویس‌های ارائه شده توسط سرویس دهندگان مختلف که ارتباط سست با یکدیگر داشته باشند تاکید دارد. این نوع معماری از ترکیب سرویس‌ها برای ساخت شبکه‌ای از سرویس‌ها و در نهایت سیستم نرم‌افزاری استفاده می‌کند. از این رو چنین سیستمی باید آمادگی مواجهه با مشکلاتی نظیر از دسترس خارج شدن هر یک از سرویس‌ها، تغییر کیفیت خدماتی که هر یک از سرویس‌ها ارائه می‌کنند و یا حتی خاموش شدن کامل یک سرویس را داشته باشد. این مشکلات تاثیر مستقیم بر قابلیت اعتماد[6] سیستم چه از لحاظ دسترس‌پذیری[7] (قابلیت پذیرش یک درخواست سرویس) و چه از لحاظ قابلیت اطمینان[8] (قابلیت انجام درخواست صادر شده به صورت موفقیت آمیز) دارند[۳]. با این حساب تضمین قابلیت اعتماد در سیستم‌های نرم‌افزاری سرویس-گرا یک شرط اساسی به شمار می‌آید.

برای رسیدن به یک سیستم سرویس‌گرای قابل اعتماد که کیفیت خدمات آن تضمین شده باشد، لازم است:

۱- بر کیفیت خروجی سرویس‌ها نظارت شده و این کیفیت اندازه‌گیری شود.

۲- بازخورد مشاهدات انجام شده از کیفیت سرویس‌ها منجر به ایجاد تصمیماتی برای تغییر در انتخاب سرویس‌دهنده، انتخاب سرویس و یا تغییر در معماری سرویس‌ها شود به گونه‌ای که کیفیت خدمت ارائه شده در مجموع افزایش یابد.

---

[1] Quality of Service (QoS)
[2] Abstraction Level
[3] Third Party Services
[4] Monitoring
[5] Re-planning
[6] Dependability
[7] Availability
[8] Reliability

۳- تصمیماتی که در گام دوم اتخاذ می‌شوند به سیستم نرم‌افزاری در حال اجرا اعمال شود به گونه‌ای که خدشه‌ای به وظیفه‌مندی‌های ارائه شده توسط سیستم وارد نشده و سیستم به حالت ناپایدار وارد نشود.

سه گامی که در بالا ذکر شد در واقع تعریف ضمنی نوعی از توسعه نرم‌افزار است که به آن مهندسی نرم-افزارهای قابل تطبیق[1] گفته می‌شود. نرم‌افزارهای قابل تطبیقی که دخالت عامل انسانی در انجام عمل انطباق در آنها به حداقل رسیده باشد نرم‌افزارهای خودتطبیق[2] نامیده می‌شوند. مدل پایه‌ایِ ارائه شده برای نرم‌افزارهای خود تطبیق بر مبنای دو گام اصلی مدیریت تغییرات[3] و مدیریت تکامل[4] بنا نهاده شده است [۴]. روش‌های متداول کنونی هم با ایده گرفتن از این مدل، با طراحی یک حلقه‌ی کنترلی، چهار وظیفه‌ی مشاهده داده‌های محیطی، تحلیل داده‌ها، انجام تصمیم‌گیری و اعمال تغییرات را انجام می‌دهند[۴]. با وجود تحقیقات متعددی که در زمینه نرم‌افزارهای خودتطبیق صورت گرفته است اما این تحقیقات بیشتر بر روی سبک معماری مبتنی بر مولفه[5] انجام شده است[۵]. کارهای محدودی هم که در زمینه معماری سرویس‌گرا صورت گرفته عموما تنها به بررسی یک مورد خاص پرداخته‌اند و سعی در ارائه راه‌حل‌های خاص برای آن مورد داشته‌اند. اما از سوی دیگر کارهای گسترده‌ای در زمینه کیفیت سرویس در سطح مدل‌سازی، طراحی و پیاده‌سازی سرویس‌ها انجام شده است. روشن است که مهم‌ترین عامل تغییر در سیستم‌های نرم‌افزاری سرویس‌گرا تغییر در کیفیت سرویس است. در نتیجه به نظر می‌رسد خودتطبیقی در نرم-افزارهای مبتنی بر معماری سرویس‌گرایی، اگر نه همیشه در اکثر اوقات منتهی به تطبیق سرویس‌ها و معماری برای رسیدن به کیفیت سرویس مطلوب می‌شود.

این پایان‌نامه به دنبال پاسخگویی به این پرسش اساسی است که آیا می‌توان یک مدل یا در نگاه کلی‌تر چهارچوب خودتطبیقی برای تضمین و کنترل کیفیت خدمات در معماری سرویس‌گرا ارائه داد؟ این مدل چه ویژگی‌هایی باید داشته باشد و در چه سطح انتزاعی از معماری سرویس‌گرا قرار گیرد؟ به دنبال پاسخگویی به این پرسش اصلی، کوشش می‌شود تا ضمن ارائه مدلی برای به کارگیری قابلیت خود تطبیقی در معماری سرویس‌گرا در جهت تامین کیفیت خدمات، روش‌ها و الگوهایی هم برای محقق‌سازی آن ارائه شود تا هم به نوعی اعتبارسنجی چهارچوب پیشنهاد شده باشند و هم به نشان دادن مسیر نهایی چهارچوب کمک کند.

---

[1] Adaptive Software Engineering
[2] Self Adaptive Software
[3] Change Management
[4] Evolution Management
[5] Component Based Software Engineering (CBSE)

۱-۳ اهداف تحقیق

هدف اصلی این پایان‌نامه ارائه یک چهارچوب عمومی برای معماری سرویس‌گرای خودتطبیق است که قادر باشد کیفیت سرویس‌ها را در زمان اجرا تضمین کند. برای دستیابی به این هدف موارد زیر در این پایان‌نامه ارائه شده‌اند:

۱. یک چهارچوب کاری برای الف) تضمین کیفیت سرویس در زمان مدل‌سازی و ب) تطبیق معماری سرویس‌گرا برای کنترل کیفیت سرویس در زمان اجرا
۲. یک روش خودکار مدل‌رانه[1] برای شناسایی سرویس[2] در فاز مدل‌سازی سرویس‌گرا در راستای محقق سازی[3] بند الف چهارچوب
۳. یک سازوکار انطباق در زمان اجرا جهت انتخاب سرویس[4] مناسب یا معماری مناسب در راستای محقق‌سازی بند ب چهارچوب

۱-۴ دامنه تحقیق

این تحقیق همان‌طور که در بخش قبل اشاره شد به دنبال معرفی یک چهارچوب کاری برای معماری سرویس‌گرا است به گونه‌ای که طی آن کیفیت سرویس ارائه شده کنترل شود. بر این اساس در زمان مدل‌سازی، سرویس‌ها به کمک برخی توابع مکاشفه‌ای طوری انتخاب می‌شوند که پتانسیل بالاتری برای برخورداری از کیفیت بالا در زمان پیاده‌سازی داشته باشند. در زمان اجرا هم با گسترش مدل‌های موجود سعی شده است تا سازوکاری برای انطباق مدل سرویس با تغییرات ایجاد شده جهت حفظ کیفیت خدمات ارائه شود.

لازم به ذکر است موارد مربوط به نظارت بر سرویس‌ها، مشاهده و جمع‌آوری داده‌های آماری جهت برآورد کیفیت سرویس خارج از محدوده‌ی این پایان‌نامه می‌باشد. هم‌چنین در راستای هدف شماره دو پایان‌نامه، صرفا به ارائه روشی برای شناسایی سرویس‌ها می‌پردازیم و روش‌های طراحی آن‌ها بررسی نمی‌شوند. بنابراین خروجی این بخش از تحقیق مجموعه‌ای از سرویس‌های کاندید بوده و جزئیات ارتباطات میان سرویس‌ها (پیام‌های ارسالی و دریافتی سرویس‌ها)، جزئیات عملیات هر سرویس و سایر جزئیات طراحی سرویس‌ها را مد نظر قرار نمی‌دهد. از سوی دیگر در راستای هدف شماره سه نیز ذکر این نکته ضروری است که الگوریتم‌های ارائه شده مستقل از محدودیت‌های وب‌سرویس‌های موجود و به طور کل مستقل از فناوری پیاده‌سازی ارائه شده‌اند.

---

[1] Model Driven
[2] Service Identification
[3] Realization
[4] Service Selection

۱-۵ مراحل انجام تحقیق

برای پاسخگویی به مسئله طرح شده در این تحقیق همان‌طور که در شکل ۱-۱ مشاهده می‌شود، از مطالعات کتابخانه‌ای و شرکت در جلسات بحث و گفتگو پیرامون معماری نرم‌افزار جهت شناسایی مفاهیم مورد نیاز تحقیق استفاده شده است. سپس روش‌های مطرح در زمینه نرم‌افزارهای سرویس‌گرای خود تطبیق مورد بررسی و مقایسه قرار گرفتند. کاستی‌ها و معایب این روش‌ها در کنار نقاط قوت و ایده‌های به کار رفته در آنها به دقت مورد بررسی قرار گرفت. در نتیجه این بررسی در این پایان‌نامه چهارچوبی جدید برای به کارگیری قابلیت خود تطبیقی در معماری سرویس‌گرا ارائه شده است که باعث تضمین کیفیت در زمان مدل‌سازی و کنترل کیفیت در زمان اجرا می‌شود. در ادامه کار تحقیق به دو بخش مجزا تقسیم شد.

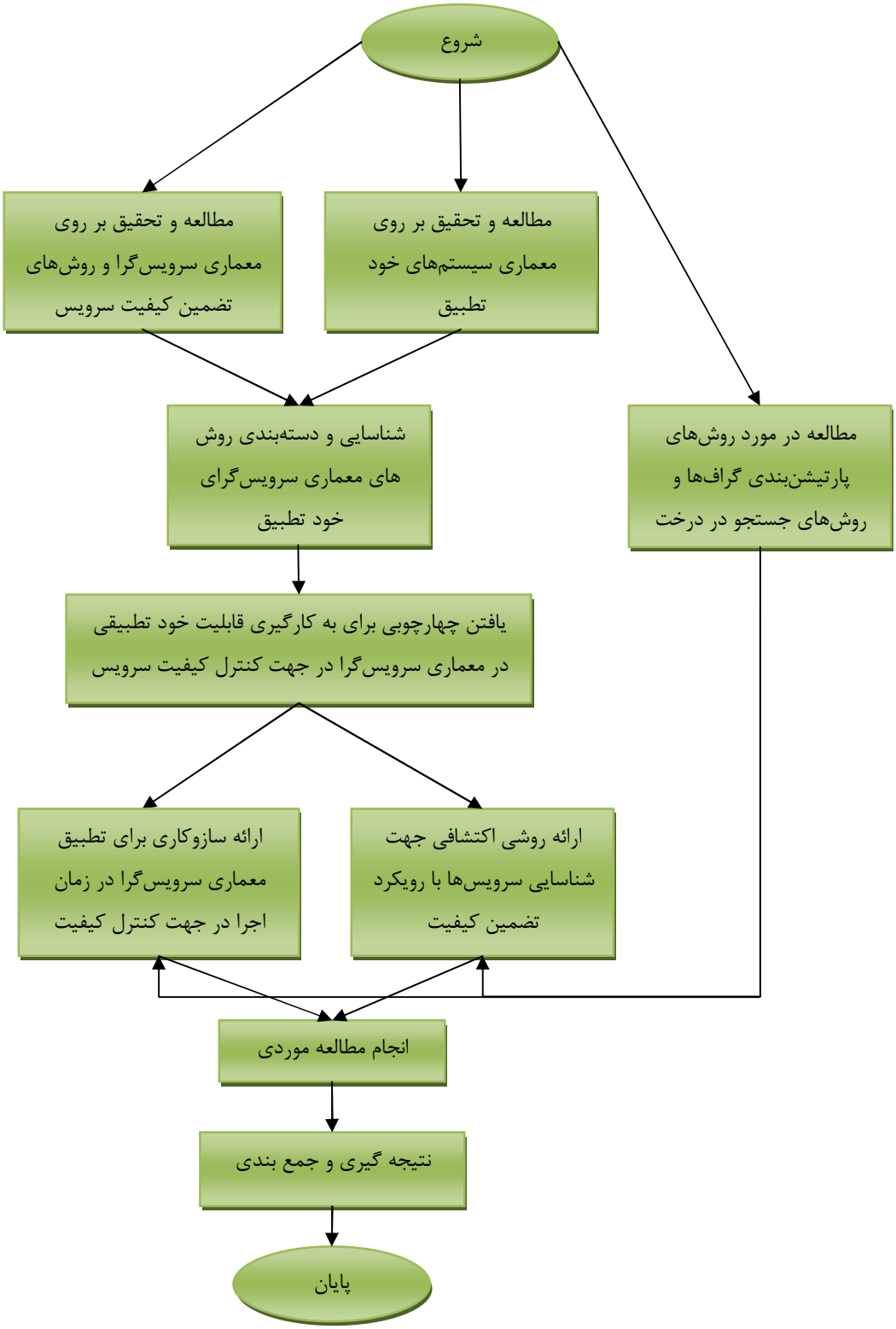
بخش نخست به یافتن روشی برای شناسایی سرویس‌های کاندید برای حرفه اختصاص یافت که طی آن با استفاده از برخی توابع مکاشفه‌ای و نیز الگوریتم‌های پارتیشن‌بندی گراف، روشی خودکار جهت شناسایی سرویس-های مناسب برای حرفه معرفی شد. در بخش دوم سازوکاری برای اعمال ویژگی خود تطبیقی در زمان اجرا برای معماری‌های سرویس‌گرا ارائه شده است که در آن خصوصیات جدیدی برای مفاهیم پایه سرویس‌گرایی مثل سرویس‌گیرنده و واسط سرویس[1] تدوین شده است. در آخر نیز ضمن انجام یک مطالعه موردی امکان‌سنجی چهارچوب پیشنهادی بررسی شده و مورد ارزیابی قرار گرفته است.

۱-۶ ساختار پایان نامه

در فصل دوم مفاهیم پایه و ادبیات تحقیق در دو بخش معماری سرویس‌گرا و نرم‌افزارهای خودتطبیق مورد بررسی قرار می‌گیرند. در فصل سوم کارهای مرتبط انجام شده در زمینه روش‌های ارائه شده برای به کارگیری ویژگی خودتطبیقی در معماری‌های سرویس‌گرا بررسی و مقایسه می‌شوند. کاستی‌های روش‌های موجود بیان شده و برای پوشش نواقص چهارچوب نوینی برای معماری سرویس‌گرای خودتطبیق ارائه می‌شود. در این فصل هم‌چنین سازوکار انجام تطبیق در زمان اجرا نیز به تفصیل توصیف می‌شود. فصل چهارم به معرفی روش شناسایی سرویس‌های حرفه و الگوریتم‌های پیاده‌سازی آن اختصاص دارد. این روش به گونه‌ای طراحی شده است که قابل استفاده شده توسط چهارچوب ارائه شده در فصل سوم باشد. فصل پنجم به ارزیابی و امکان‌سنجی الگوریتم‌ها و چهارچوب ارائه شده از طریق به کارگیری آن‌ها در یک مطالعه موردی و سپس مقایسه با روش‌های موجود می‌پردازد. و در نهایت در فصل ششم ضمن جمع‌بندی و نتیجه‌گیری از مطالب ارائه شده، کارهای آتی که می‌تواند چهارچوب پیشنهادی را تقویت کند بیان می‌شوند.

---

[1] Service Broker



شکل ۱-۱ مراحل انجام تحقیق

فصل دوم
ادبیات تحقیق

۲-۱ مقدمه

در فصل پیش مسئله مورد مطالعه در این پایان‌نامه معرفی شد و گستره آن تعیین گردید. بر همین اساس در این فصل مفاهیم پایه و تعاریفی که در راستای هدف پایان‌نامه بوده و در ادامه راه از آنها استفاده خواهد شد، مورد بررسی قرار می‌گیرند. از آن‌جایی که هدف این پایان‌نامه دستیابی به یک چهارچوب نرم‌افزاری برای سیستم‌های سرویس‌گرا با قابلیت خودتطبیقی می‌باشد، لازم است تعاریف و مفاهیم ابتدا در دو حوزه‌ی مستقل مورد تحقیق یعنی حوزه سرویس‌گرایی و حوزه خودتطبیقی از نظر گذرانده شوند و سپس ارتباط این دو حوزه مورد بررسی قرار گیرد.

بر این اساس ابتدا مفهوم سرویس معرفی شده و متدولوژی‌های مبتنی بر معماری سرویس‌گرایی، مفاهیم سرویس‌گرایی، چرخه حیات توسعه سیستم‌های نرم‌افزاری مبتنی بر معماری سرویس‌گرا و اهمیت و کاربرد آنها را تشریح می‌کنیم. در بخش دوم این فصل به بررسی ویژگی خودتطبیقی به عنوان یک نیازمندی غیروظیفه‌مندی سیستم‌های نرم‌افزاری می‌پردازیم و معماری، گام‌های مهندسی نرم‌افزار و الگوهای به دست آمده برای سیستم‌های نرم‌افزاری خودتطبیق بررسی می‌شوند. در انتها نیز اصول به کارگیری قابلیت خودتطبیقی در معماری‌های نرم‌افزاری و به خصوص معماری سرویس‌گرا بررسی خواهد شد.

۲-۲ سرویس

برای سرویس تعاریف مختلفی ارائه شده است که هرکدام در حیطه و شرایط خاصی به کار می‌روند. در حیطه نرم‌افزارهای مقیاس وسیع در سطح سازمان‌ها استفاده از سطح تجرید کلاس و شیء به علت گستردگی حیطه نرم‌افزار باعث افزایش پیچیدگی می‌گردد، به همین علت از مفهوم سرویس به عنوان ابزاری جهت بالا بردن سطح تجرید و در نتیجه کاهش پیچیدگی استفاده می‌کنیم. به عبارت دیگر سرویس‌ها نسبت به اشیاء سطح بالاتری از تجرید را معرفی می‌کنند [۲]. در سازمان‌های بزرگ که از چندین حرفه مرتبط با هم تشکیل شده‌اند، می‌توان برای پاسخ به نیازهای هر حرفه، سرویسی ارائه نمود و این سرویس‌ها را به منظور پوشش تمامی نیازهای سطح سازمان با هم مجتمع کرد. با توجه به تفاوت در دیدگاه‌های متخصصین کسب و کار و متخصصین نرم افزار، تعریف سرویس در این دو حیطه متفاوت است.

۲-۲-۱ سرویس کسب و کار در برابر سرویس نرم افزاری

مفهوم سرویس کسب و کار[1] برای توصیف مجموعه مشخصی از فعالیت‌هایی که توسط یک سازمان انجام می‌شوند به کار می‌رود. بنابراین برای شناسایی نرم‌افزاری سرویس های کسب و کار می توان مجموعه فعالیت های سازمان را بررسی و به چند گروه تقسیم نمود. به هرکدام از این گروه‌ها یک سرویس کسب و کار گفته می‌شود. البته باید توجه داشت برای گروه‌بندی فعالیت‌های سازمان در سرویس‌ها باید به فاکتورهای مختلفی از جمله انسجام[2] فعالیت‌های درونی سرویس‌ها توجه داشت. در تعریف فوق به نحوه پیاده‌سازی سرویس‌های سازمان توجه نشده است. از این رو سرویس‌های کسب و کار تنها از نظر سازمانی قابل بررسی هستند.

از سوی دیگر مفهوم سرویس نرم افزاری[3] برای توصیف قسمتی از یک برنامه کاربردی[4] که به صورت جداگانه توسط موجودیت‌های متفاوت مورد استفاده قرار می‌گیرد، به کار می رود. طبق این تعریف سرویس نرم-افزاری از یک قرارداد سرویس[5]، حداقل یک واسط سرویس[6]، پیاده‌سازی داده‌ها و منطق سازمان تشکیل شده است [۶].

---

[1] Business service
[2] Coherence
[3] Software service
[4] Application
[5] Service contract
[6] Service interface

قرارداد سرویس بیان کننده وظیفه‌مندی سرویس، عملیات آن، پیام‌های ورودی و خروجی هر یک از این عملیات، و قواعد و مقررات استفاده از سرویس است. واسط سرویس عملیات قابل فراخوانی سرویس را برای استفاده کنندگان آن مشخص می‌کند. بنابراین منطق حرفه و داده‌های آن از طریق واسط سرویس در دسترس استفاده کنندگان آن قرار می‌گیرند [۷]. به بیان دیگر سرویس‌های نرم افزاری علاوه بر مشخص‌سازی دقیق منطق سازمان، به جنبه‌های پیاده سازی این منطق و داده‌های آن نیز می‌پردازند.

ایده اصلی مفاهیم سرویس‌های کسب و کار و سرویس‌های نرم‌افزاری بسیار به هم شبیه است. هر دوی این مفاهیم منطق سازمان را به بخش‌های کوچک‌تری تقسیم کرده و از نوعی قرارداد برای مشخص‌سازی ویژگی‌های سرویس استفاده می‌کنند. اما از آنجا که سرویس‌های کسب و کار حیطه بزرگ‌تری از فعالیت‌های سازمان را پوشش می‌دهند، ارتباطات میان سرویس‌های کسب و کار نسبت به ارتباطات میان سرویس‌های نرم‌افزاری پیچیده‌تر است. هم‌چنین اطلاعاتی که در قراردادهای سرویس نرم‌افزاری گنجانده می‌شود حاوی جزئیات مربوط به نحوه فراخوانی سرویس‌ها توسط مشتریان و سرویس‌های دیگر است (اطلاعات مربوط به قواعد و معنای[1] ارتباطات میان سرویس‌ها) در صورتی که قرارداد سرویس‌های کسب و کار اطلاعات مربوط به ارتباطات میان موجودیت‌های سازمان را توصیف می‌کند [۷].

از این‌رو می‌توان گفت برای شناسایی سرویس‌های نرم‌افزاری مورد نیاز جهت پوشش دهی به نیازهای سازمان می‌توان ابتدا با استفاده از مدل‌های سازمانی سرویس‌های کسب و کار را تعیین نمود و سپس با افزودن ویژگی ها و جزئیات نرم‌افزاری به آن‌ها، به مجموعه سرویس‌های نرم‌افزاری دست یافت.

۲-۲-۲ ویژگی های کیفی سرویس

منابع مختلف برای سرویس‌های نرم‌افزاری ویژگی‌های کیفی مختلفی را ارائه کرده‌اند. به طور کلی موارد زیر را می‌توان به عنوان ویژگی‌های کیفی سرویس‌های نرم‌افزاری در نظر گرفت:

۱. داشتن قرارداد رسمی [۲ و ۷]: سرویس‌های نرم‌افزاری و کسب و کار باید دارای حداقل یک قرارداد رسمی باشند. قرارداد رسمی سرویس حداقل باید حاوی اطلاعاتی در مورد وظیفه‌مندی سرویس، ویژگی‌های کیفی، و پیام‌های ورودی و خروجی آن باشد.

[1] Syntax & Semantic

۲. افزایش سطح انتزاع [۲ و ۷]: سرویس‌ها با پوشش دادن جزئیات درونی منطق سطح پایین سازمان، سطح انتزاع را افزایش می‌دهند [۲]. میزان سطح انتزاع سرویس‌ها پس از تعریف قرارداد رسمی مشخص می‌گردد [۷].

۳. قابلیت استفاده مجدد [۲ و ۷]: سرویس‌ها را باید بتوان در پروژه‌های دیگر درون سازمانی، یا پروژه‌های مشابه برون سازمانی در توسعه سیستم‌های نرم‌افزاری دوباره استفاده کرد.

۴. اتصال سست[1] [۲ و ۷]: سرویس‌ها باید حداقل وابستگی را به هم داشته باشند. منظور از وابستگی میان سرویسی استفاده یک سرویس از یک یا چند عملیات سرویس‌های دیگر برای انجام وظیفه‌مندی خود است.

۵. قابلیت ترکیب [۲ و ۷]: سرویس‌ها را باید بتوان با صرف هزینه کم با هم ترکیب نمود.

۶. خودمختاری [۲ و ۷]: برای خودمختاری سرویس‌ها دو تعریف وجود دارد. در تعریف اول منظور از سرویس خودمختار سرویسی است که بتوان آن را به صورت مستقل و بدون استفاده از سرویس‌های دیگر اجرا نمود [۷]. این تعریف از خودمختاری تا حد زیادی شبیه ویژگی اتصال سست سرویس‌هاست. تعریف دوم خودمختاری را به عنوان عدم رویهم افتادگی[2] محدوده سرویس‌ها بیان می‌دارد [۲ و ۷].

۷. بی وضعیتی[3] [۲ و ۷]: مقصود از بی وضعیت بودن سرویس‌ها عدم مدیریت اطلاعات مربوط به حالت (state) اجرای فرآیند توسط سرویس‌هاست. این امر به دلیل کاهش ویژگی اتصال سست سرویس‌ها در صورت مدیریت اطلاعات حالت فرآیندهاست. بنابراین سرویس‌ها باید به گونه‌ای طراحی شوند که حداقل اطلاعات ممکن را در مورد وضعیت فرآیند نگهداری کنند [۲].

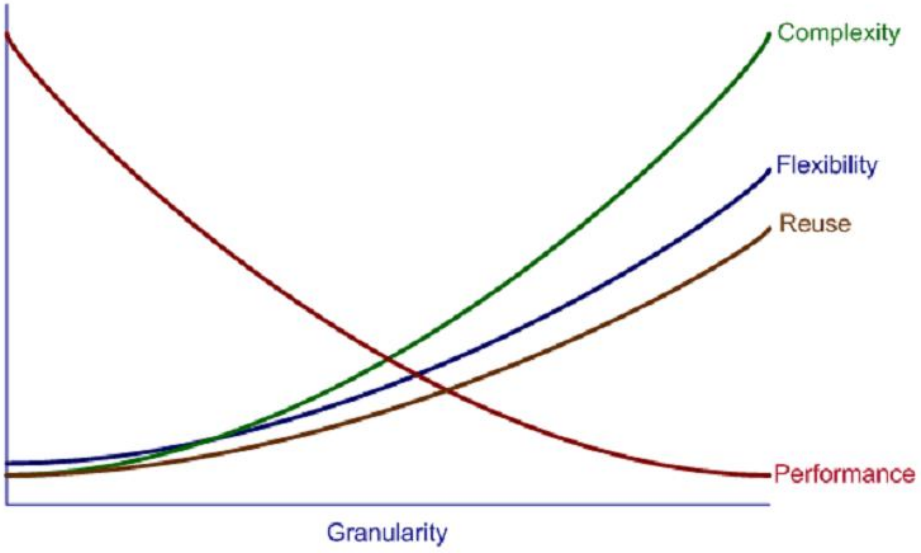
۸. قابلیت کشف[4] [۲ و ۷]: سرویس‌ها باید به گونه‌ای طراحی و پیاده‌سازی شوند که افراد و تقاضاکنندگان سرویس بتوانند از روی توصیف آنها از منطق درونی‌شان آگاه شوند. این ویژگی را می‌توان تا حد زیادی با ویژگی قرارداد رسمی سرویس‌ها مرتبط دانست.

۹. انسجام [۷]: منظور از انسجام سرویس میزان ارتباط معنایی و اجرایی عملیات درونی آنهاست. به عبارت دیگر سرویس‌ها باید طوری طراحی شوند که فعالیت‌های درونی آنها بیشترین میزان ارتباط منطقی، معنایی، و عملیاتی را با هم داشته باشند.

---

[1] Loose coupling
[2] Overlapping
[3] Stateless
[4] Discoverability

علاوه بر ویژگی‌های فوق، دانه‌بندی[1] را می‌توان به عنوان یکی دیگر از ویژگی‌های اساسی سرویس‌ها به شمار شمار آورد [۷]. منظور از دانه‌بندی سرویس‌ها بزرگی و کوچکی محدوده پوشش داده شده توسط هر سرویس است. برخلاف ویژگی‌های قبلی، برای دانه‌بندی سرویس‌ها نمی‌توان کاهش یا افزایش را تجویز کرد. به عبارت دیگر نمی‌توان گفت دانه‌بندی سرویس‌ها را بهتر است افزایش دهیم یا کاهش. این امر به خاطر آنست که دانه‌بندی سرویس‌ها بر روی چندین ویژگی کیفی سیستم نهایی تأثیر می‌گذارد. در شکل ۲- ۱ این موضوع نمایش داده شده است.



شکل ۲-۱ تاثیر دانه بندی سرویس ها بر سایر ویژگی های کیفی

در شکل ۲-۱ تأثیر دانه‌بندی بر ویژگی‌های کارآیی، انعطاف‌پذیری، پیچیدگی، و قابلیت استفاده مجدد نشان داده شده است که از میان آن‌ها سه ویژگی اول مربوط به کیفیت سیستم نهایی و ویژگی چهارم مربوط به کیفیت سرویس است. البته باید توجه داشت در شکل ۲- ۱ محور افقی ریزدانگی سرویس‌ها را نشان می‌دهد به این معنی که هر چه در محور افقی به سمت راست حرکت کنیم سرویس ها کوچک‌تر می‌شوند.

۲-۳ معماری سرویس گرا

در منابع برای معماری سرویس‌گرا تعریف‌های مختلفی ارائه شده است که هر کدام با دیدگاه خاصی به این مفهوم توجه کرده‌اند. به عبارت دیگر، افراد مختلف با توجه به نقش فنی خود، معماری سرویس‌گرا را تفسیر کرده‌اند. این تعریف‌ها را می‌توان به دو دسته تعریف‌های آکادمیک و تجاری تقسیم‌بندی کرد. برخی تعاریف آکادمیک عبارتند از:

[1] Granularity