دانشگاه کاشان

دانشکده مهندسی

گروه مهندسی نرم افزار کامپیوتر

پایان نامه

جهت اخذ درجه کارشناسی ارشد در رشته مهندسی کامپیوتر

عنوان:

فعال سازی UDDI در معماری سرویس گرا بوسیله پایگاه داده فعال و ارزیابی کارآیی آن

استاد راهنما:

دکتر سید مرتضی بابامیر

استاد مشاور:

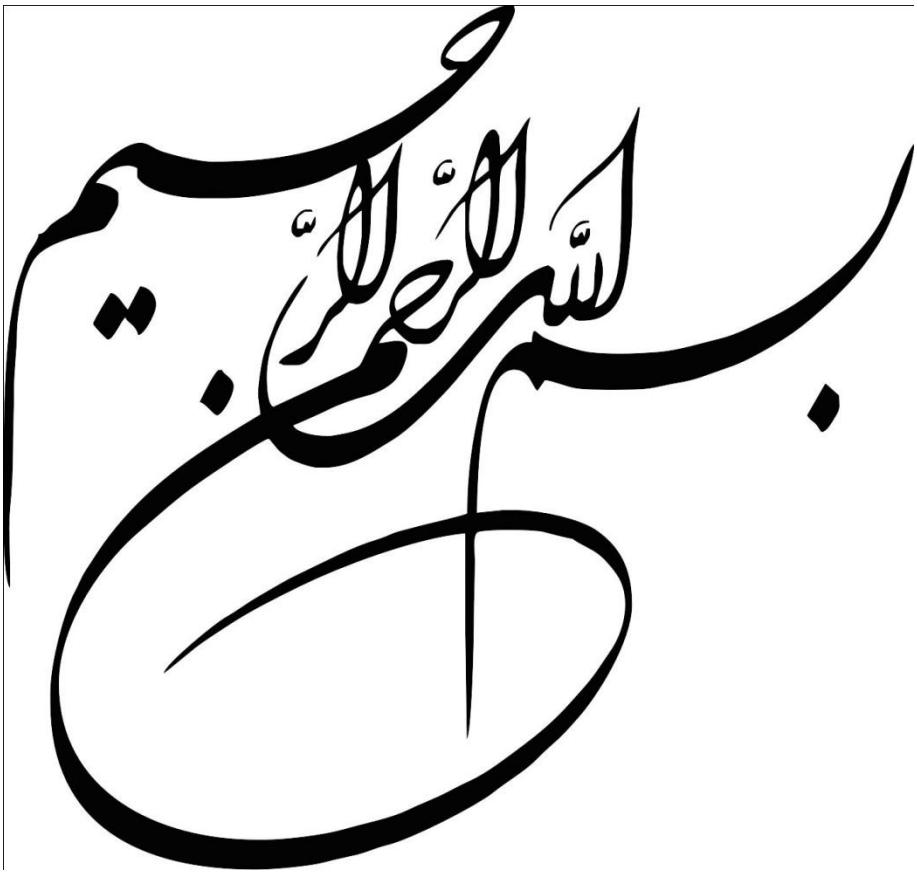
دکتر محمد ابراهیم شیری احمد آبادی

دکتر حسین ابراهیم پور کومله

به وسیله:

مسعود عرب فرد

شهریور ماه ۱۳۹۰


بسم الله الرحمن الرحيم

تقدیم به....

همه

پدر و مادر

هایی که از خود گذشتند تا عصاره های وجودشان پر بگیرند و به اوج برسند.

و تقدیم این پایان نامه تنها کاری است که من حقیر می توانم برای آنان انجام دهم.

باشد تا کمی از الطا ف آنها را پاسخگو باشم.

و در آخر.....

اللهم عجل لولیك الفرج

تشکر و قدردانی

سپاسگزارم

از استاد راهنمای عزیزم جناب آقای دکتر بابامیرکه با تلاش ها و زحمات بی دریغ شان مرا در ارائه پایان نامه یاری کردند واین پایان نامه با زحمات بی دریغ ایشان به سرانجام رسید.

سپاسگزارم

از جناب آقای دکتر شیری به دلیل راهنمایی هایی که در راستای انجام پایان نامه به بنده دادند.

از جناب آقای دکتر ابراهیم پور به دلیل راهنمایی هایی که در راستای انجام پایان نامه به بنده دادند

از جناب آقای دکتر شیخ زاده به دلیل حضور در جلسه دفاع به عنوان استاد ناظر

سپاسگزارم

از پدر مهربانم و مادر عزیزتر از جانم که تمام تلاش خود را در جهت ارتقای علمی بنده کرده اند و تا عمر دارم مدیون زحمات آنها هستم و هرکجا که باشم بوسه ای بر دستان عزیزشان می زنم.

چکیده

مخزن معماری سرویس گرا که با نام UDDI[1] شناخته می شود در اکثر معماری های توزیع شده بر پایه سرویس مورد استفاده قرار می گیرد. استفاده از این مخزن به مصرف کنندگان سرویس کمک می کند تا به راحتی سرویسی که نیاز دارند را پیدا کنند. اما این مخزن یک مخزن منفعل است بدین معنی که بعد از پیدا کردن سرویس، مصرف کننده دیگر هیچ تعاملی را با مخزن ندارد و این عدم تعامل باعث بروز مشکلاتی برای مصرف کنندگان خواهد شد. ما در این تحقیق قصد داریم تا تمرکز اصلی را بر روی مخزن سرویس گذاشته و عیب منفعل بودن آن را بوسیله تکنیک پایگاه داده فعال مرتفع کنیم. از جمله مشکلاتی که ممکن است برای مصرف کنندگان سرویس اتفاق بیافتد اینست که در صورتی که تولید کننده ای آدرس سرویس خودش را تغییر دهد مصرف کننده از این تغییر مطلع نمی شود و یا اگر اطلاعات مربوط به سرویسی ویرایش شود باز هم مصرف کننده از آن مطلع نمی شود مگر اینکه به مخزن سرکشی کند. برای برقراری ارتباط بین مصرف کننده و مخزن ما از محرک[2] های پایگاه داده استفاده می کنیم تا مشکل منفعل بودن مخزن را برطرف کنیم و بعد از فعال سازی مخزن کارایی مخزن فعال را ارزیابی کنیم.

به منظور ارائه عملی بودن مدل پیشنهادی، ما آن را برای یک سیستم اطلاعاتی بیمارستانی نمونه با بیست ایستگاه کاری به کار گرفتیم و کارایی آن را ارزیابی کردیم. مدل پیشنهادی ما از تعدادی سرویسگر ناظر و تعدادی متقاضی تشکیل شده است که در آن سرویسگر ناظر وظیفه اطلاع رسانی در خصوص تغییرات مخزن به متقاضی یا مصرف‌کنندگان سرویس را به عهده دارد. علاوه بر این، مدل طراحی‌شده اجازه افزودن قابلیت‌هایی مانند تحمل پذیری خطا، آمارگیری و دسته‌بندی مصرف-کنندگان سرویس وکنترل متمرکز و تعامل بین مصرف کنندگان سرویس را به مخزن می‌دهد.

[1] Universal Description Discovery and Integration
[2] Trigger

کلمات کلیدی: معماری سرویس گرا – ثبات سرویس – پایگاه داده فعال - اطلاع رسانی به مصرف کننده سرویس .

فهرست مطالب:

عنوان	صفحه
فصل ۱ : کلیات	۱
۱-۱ مقدمه	۲
۱-۲ مروری بر معماری سرویس گرا	۳
۱-۳ سرویس	۷
۱-۴ نتیجه	۲۱
۱-۵ ساختار پایان نامه	۲۱
فصل۲ : ادبیات و پیشینه تحقیق	۲۳
۲-۱ مقدمه	۲۴
۲-۲ پیشینه تحقیق	۲۴
۲-۲-۱ حوزه فعال سازی مخزن	۲۴
۲-۲-۲ حوزه پایش سرویس	۴۵
۲-۲-۳ حوزه کنترل کیفیت سرویس	۶۲
۲-۳ لزوم تحقیق	۷۴
فصل۳ : پایگاه داده فعال و قوانین فعال سازی	۷۶
۳-۱ مقدمه	۷۷

۳-۲ تعریف پایگاه داده ۸۰
۳-۳ تعریف جامعیت ۸۲
۳-۴ عوامل نقص جامعیت در پایگاه داده ۸۳
۳-۵ کنترل جامعیت ۸۴
۳-۶ تعریف پایگاه داده فعال ۸۴
۳-۷ تعریف قواعد فعال ۸۶
۳-۸ معماری سیستم فعال ۸۸
۳-۹ پیاده سازی معماری سیستم فعال ۹۱
۳-۱۰ مدل اجرای سیستم فعال ۹۲
۳-۱۱ مزایای سیستم فعال و کاربرد آن ۹۴
۳-۱۲ مکانیزم پیاده سازی قواعد فعال ۹۶
۳-۱۳ ابزارهای پیاده سازی قواعد فعال ۹۷
۳-۱۴ شرح واژگان کلیدی ۱۰۳
فصل ۴ : مدل پیشنهادی فعال سازی مخزن UDDI ۱۰۶
۴-۱ مقدمه ۱۰۷
۴-۲ مدل پیشنهادی ۱۰۹
۴-۳ مدل پیشرفته ۱۱۱
۴-۴ مقایسه بین مدل ها ۱۱۲

۴-۵ بیان تفاوت های بین مدل ها ۱۱۷

۴-۶ تفاوت تحمل پذیری خطا بین مدل ها ۱۱۹

۴-۷ پیاده سازی مدل پیشنهادی ۱۲۰

۴-۷-۱ برنامه متقاضی ۱۲۳

۴-۷-۲ برنامه سرور ۱۲۴

۴-۷-۳ محرک های مورد استفاده ۱۲۵

۴-۸ روند اجرای قابلیت ها در مدل پیشنهادی ۱۲۷

۴-۹ شکل کلی مدل ۱۳۲

۴-۱۰ چند سناریو مربوط به مدل ۱۳۲

۴-۱۱ طرحی از مدل پیاده سازی شده ۱۳۶

فصل ۵ : پیاده سازی سیستم HIS ۱۳۸

۵-۱ مقدمه ۱۳۹

۵-۲ معرفی سیستم HIS ۱۳۹

۵-۳ شماره گذاری زیر سیستم ها ۱۴۳

۵-۴ تشریح سرویس دهنده های مهم مرتبط با زیر سیستم های انتخابی ۱۴۵

۵-۵ تعیین تعاملات زیر سیستم ها و سرویس دهنده ها ۱۴۶

۵-۶ سرویس ها و خصیصه هاو عملیات هر سرویس ۱۴۹

۵-۷ نحوه پیاده سازی ۱۵۳

فصل ۶ : ارزیابی کارایی ۱۵۸

۶-۱ مقدمه ۱۵۹

۶-۲ معیارهای کارایی ۱۶۱

۶-۳ ارزیابی زمان پاسخ گویی ۱۶۴

۶-۴ ارزیابی توان عملیاتی ۱۶۷

۶-۵ نتیجه ۱۶۸

فصل ۷ : نتیجه گیری ۱۶۹

۷-۱ نتیجه گیری کلی ۱۷۰

۷-۲ پیشنهادات و کارهای آتی ۱۷۳

فهرست مراجع ۱۷۵

واژه نامه ۱۷۹

پیوست یک:کد معماری پیشنهادی

پیوست دو: مقالات پذیرفته شده

فهرست جدول ها:

عنوان صفحه

جدول ۱.مقایسه تکنیک فعال و غیر فعال ۳۹

جدول ۲. مقایسه مدل ها بر طبق معیارهای معرفی شده ۱۱۳

جدول ۳.نقاط قوت و ضعف مدل ها ۱۱۴

جدول ۴ : زمان پاسخ گویی سرویس های سیستم HIS ۱۶۴

جدول ۵ :توان عملیاتی مربوط به عملیات اضافه سازی سرویس ها ۱۶۷

فهرست نمودارها:

عنوان صفحه

نمودار ۱. زمان پاسخ گویی عملیات اضافه سازی ۱۶۵

نمودار ۲. زمان پاسخ گویی عملیات ویرایش ۱۶۵

نمودار ۳. زمان پاسخ گویی عملیات حذف ۱۶۶

نمودار ۴. زمان پاسخ گویی عملیات لیست گیری ۱۶۶

نمودار ۵. توان عملیاتی سرویس ها ۱۶۷

فهرست شکل ها:

عنوان **صفحه**

شکل ۱ .اطلاعات درون مخزن UDDI ۱۱

شکل ۲ . قالب یک پیام SOAP ۱۸

شکل ۳.شکل ساختاری UDDI ۲۹

شکل ۴.چارچوب فعال سازی ۳٤

شکل ۵.تعامل دلال سرویس فعال با تولید کننده و مصرف کننده ٤۰

شکل ۶. معماری پایش وب سرویسها ٤۲

شکل ۷. فرایندهای بر پایه مدل ٤٤

شکل ۸. زمان پاسخ وب سرویس ٤۹

شکل ۹.گزارش دهی استثنائات عمومی ۵۱

شکل ۱۰.گزارش دهی عملیات عمومی ۵۳

شکل ۱۱.معماری کنترل کیفیت سرویس ۵۵

شکل ۱۲.معماری سیستم پایش ٦۲

شکل ۱۳.معماری کنترل کیفیت سرویس ٦٦

شکل ۱۴.ثبت وب سرویس ۷۰

شکل ۱۵.جستجوی وب سرویس ۷۱

شکل ۱۶.فرایند پایش و ارزیابی ۷۳
شکل ۱۷.معماری سیستم فعال ۹۰
شکل ۱۸.پایگاه داده ایجاد شده ۱۰۰
شکل ۱۹.نمونه یک محرک ۱۰۱
شکل ۲۰.اضافه سازی یک رکورد ۱۰۲
شکل ۲۱. نتیجه اجرای محرک ۱۰۲
شکل ۲۲.مدل گوش دهنده ۱۰۹
شکل ۲۳.مدل پیشرفته مخزن فعال ۱۱۲
شکل ۲۴.نحوه کار برنامه کلاینت ۱۲۳
شکل ۲۵.نحوه کار برنامه سرور ۱۲٤
شکل ۲۶.نمای کلی مدل پیشنهادی ۱۳۲
شکل ۲۷.طرح پیاده سازی سیستم ۱۳٦
شکل ۲۸.چارت سیستم اطلاعات بیمارستانی ۱٤۱
شکل ۲۹.زیر سیستم های انتخابی ۱٤۲
شکل ۳۰.شماره گذاری زیر سیستم های انتخابی ۱٤۳
شکل ۳۱.نمایی از تعاملات سرویس های استخراجی ۱٥۲
شکل ۳۲.نمایی از انتشار سرویس ها بر روی شبکه ۱٥۳
شکل ۳۳.لیست عملیات سرویس ترخیص ۱٥٤

شکل ۳۴.عملیات اضافه سازی سرویس ترخیص ۱۵۵

شکل ۳۵.عملیات بازگردانی اینکه آیا بیمار در لیست سیاه وجود دارد یا خیر ۱۵۶

شکل ۳۶.نمایی از برنامه HIS ۱۵۶

شکل ۳۷.فرم طراحی شده برای سرویس ترخیص ۱۵۷

شکل ۳۸.نمودار زمان پاسخ گویی ۱۶۲

شکل ۳۹.روند تغییر توان عملیاتی ۱۶۳

فهرست اختصارات:

AOSIS	Advancing open standards for the information society
ASB	Active service Broker
BPMS	Business Process Management System
DBMS	Database Management System
DDL	Data Definition Language
ECA	Event Condition Action
FIFO	First Input First Output
HIS	Health Information System
HTTP	Hypertext Transfer Protocol
IIS	Internet Information Services
JVM	Java virtual Machine
OWL	Web Ontology Language
QOS	Quality Of Service
SOAP	Simple Object Access Protocol
TCP	Transmission Control Protocol
UDDI	Universal Description Discovery and Integration
UDP	User Datagram Protocol
URL	Uniform Resource Locator
WSDL	Web Service Definition Language
XML	Extensible Markup Languages
XSD	XML Schema Definition

فصل۱:

کلیات

۱–۱: مقدمه:

معماری سرویس‌گرا مدلی برای توسعه برنامه های نرم افزاری است و از آن به عنوان یکی از معماری های مهندسی نرم افزار نام می برند که در آن استفاده و سازماندهی منابع گسترده اعم از برنامه و داده به نحوی صورت می گیرد که بکارگیری این قابلیت ها به شکل یکسان و با تعاریف مشخص صرف نظر از سکو[1]،مشخصه ی شی و دامنه امکان پذیر می شود[۱]. در این فصل به معرفی کلی معماری سرویس‌گرا می پردازیم و آن را به طور کامل مورد بررسی قرار می دهیم.

[1] Platform

۱-۲: مروری بر معماری سرویس‌گرا

معماری سرویس‌گرا یک مدل و رهیافت برای ساخت نرم افزارهای توزیع شده است که تمام وظایف نرم افزار را در قالب سرویس بیان می کند. از ویژگی های این سرویس ها قابلیت استفاده و فراخوانی توسط نرم افزار های دیگر است. همانطور که می دانید برای توسعه نرم افزار های کاربردی روش ها و مدل های گوناگونی از زمان بدو بوجود آمدن مهندسی نرم افزار ارائه شده است. اولین مدل های تولید نرم افزار مدل هایی بر پایه برنامه سازی ساخت یافته[1] و پیمانه ای[2] بود و بعد از آن و با تحول در گسترش فناوری های جدید مهندسی نرم افزار روش و مدلی با نام شی گرا[3] بوجود امد. در این روش با تجزیه و تقسیم صورت مسئله به یکسری اشیا سعی می شد تا صورت مسئله به مسائل کوچکتر تقسیم شود تا شی مستقل و مشخص عمل کند. پس با تعریف دقیق و کامل از شی و اشیا و ارتباط بین آن ها نرم افزار شکل می گرفت. با ظهور مدل شی گرا در حیطه مهندسی نرم افزار راه برای ایجاد نرم افزار های بزرگ و عظیم هموار شد چرا که با تجزیه صورت مسئله به اشیا مستقل،یک پروژه می توانست بین شرکت های مختلف تقسیم شود و هر شرکت می توانست به طور مستقل روی قسمت مورد

[1] Structured
[2] Modular
[3] Object Oriented

نظر خود کار کند. در نهایت هر یک از بخش ها به طور مجزا مورد آزمایش قرار می گرفت و پس از اطمینان از نحوه عملکرد، آن جز با نرم افزار اصلی تلفیق می شد. اما با پیشرفت کاربردهای نرم افزار در حیطه تجارت های کسب و کار لزوم برقراری ارتباط بین نرم افزار ها با ساختارهای مختلف اهمیت یافت که در این زمان بود که بحث معماری سرویس‌گرا مورد توجه قرار گرفت.

اما برای معماری سرویس‌گرا تعاریف متعدد وگوناگونی بیان شده است که ما به چند تعریف آن بسنده می کنیم.

- معماری سرویس‌گرا روشی برای ساخت سیستم های توزیع شده است که در آن سیستم ها عملکرد سیستم بصورت سرویس در اختیار مصرف کنندگان و سایر سرویس ها قرار می گیرد[۲].
- معماری سرویس‌گرا روشی برای طراحی و پیاده سازی نرم افزار های بزرگ و گسترده است که بوسیله ارتباط میان سرویس ها با هم در تعامل اند[۳] .
- معماری سرویس‌گرا مجموعه قوانین،سیاست ها و چارچوب هایی است که نرم افزارها را قادر می سازد تا عملکرد خود را از طریق مجموعه سرویس های مجزا و در عین حال مربوط به هم در اختیار سایر مصرف کنندگان قرار دهد[۲].