تاریخ دریافت: ۱۳۹۷/۰۷/۲۳
تاریخ پذیرش: ۱۳۹۷/۱۲/۱۱

ارائهٔ یک معماری برای تعامل پویا بین وب‌سرویس‌ها براساس پایگاه داده فعّال

مهدی سخائی‌نیا
استادیار گروه کامپیوتر ـ دانشکده مهندسی ـ دانشگاه بوعلی سینا ـ همدان ـ ایران
پست الکترونیکی: sakhaei@basu.ac.ir

چکیده:

معماری سرویس‌گرا به‌عنوان راه‌حلی برای یکپارچه‌سازی در سیستم و سازمان‌ها مورد توجه می‌باشد. براساس این معماری امکانی فراهم گردیده که توسعه نرم‌افزارهای فراهم کننده سرویس به‌صورت جداگانه و مستقل از توسعه نرم‌افزارهای سرویس‌گیرنده انجام می‌گیرد. اما سرویس‌های لازم در زمان توسعه نرم‌افزار و به‌صورت ایستا مشخص می‌گردد. گرچه امکان جستجو و اجرای سرویس در فراهم‌کنندگان سرویس وجود دارد، اما باید هم سرویس‌گیرنده و هم سرویس‌دهنده دانش لازم درباره همدیگر را داشته باشند. این امر احتمال تعامل پویای زمان اجرای فراهم‌کننده سرویس و استفاده کننده را کاهش می‌دهد. در این پژوهش معماری رویدادگرا برای مشکل تعامل پویای فراهم کننده و استفاده کننده از سرویس مبتنی بر پایگاه داده فعال ارائه شده است. ارزیابی کیفی براساس معیارهای مطرح شده نشان داد روش ارائه شده با انتخاب و ترکیب وب‌سرویس‌ها به‌صورت خودکار، پویا و در زمان اجرا، امکان تعامل بین سرویس‌گیرنده و فراهم‌آورنده سرویس را فراهم می‌نماید.

واژه‌های کلیدی: تعامل پویای وب سرویس‌ها، پایگاه داده فعال، معماری رویدادگرا، معماری سرویس‌گرا

* نویسندهٔ مسئول

۱- مقدمه

محاسبات مبتنی بر سرویس امکان سرهم‌کردن قطعات یک برنامه کاربردی را در قالب مجموعه‌ای از سرویس‌ها فراهم می‌کند که با اتصال سست به یکدیگر می‌توانند برای ایجاد فرایندهای منعطف و پویای کسب و کار و سیستم‌های اطلاعاتی در سازمان‌ها به‌کار گرفته شوند[۱]. فرآیندهای کسب و کار در یک سازمان، وابسته به خدماتی است که در برنامه‌های کاربردی دیگر و بعضاً در تعامل با سایر سازمان‌ها فراهم می‌گردد. مفهوم "publish-find-bind" در معماری سرویس‌گرا سبب گشته توسعه نرم‌افزارهای فراهم کننده سرویس به‌صورت جداگانه و مستقل از توسعه نرم‌افزارهای سرویس‌گیرنده انجام گردد. سرویس‌های لازم برای نرم‌افزار سرویس‌گیرنده در زمان توسعه نرم‌افزار مشخص می‌گردد. گرچه امکان جستجو و اجرای سرویس در فراهم‌کنندگان سرویس وجود دارد، اما باید هم سرویس‌گیرنده و هم سرویس‌دهنده دانش لازم درباره همدیگر را داشته باشند[۲]. این امر احتمال تعامل پویای زمان اجرای فراهم‌کننده سرویس و استفاده کننده را کاهش می‌دهد. این در حالیست که بسیاری از فرآیندهای یک سازمان ترکیبی از چند وب‌سرویس از فراهم‌کنندگان

سرویس مختلف بوده و با تغییر هر یک از این سرویس‌ها در زمان اجرا لازم است که فرآیند به‌صورت پویا تغییر نماید[۳,٤]. در این پژوهش معماری رویدادگرا برای مشکل تعامل پویای فراهم کننده و استفاده کننده از سرویس مبتنی بر پایگاه داده فعال ارائه شده است. در ادامه این بخش انگیزه و راه‌حل پیشنهادی براساس یک سناریو مطرح می‌گردد.

۱-۱- انگیزه

برای بیان انگیزه انجام این پژوهش، از توصیف یک سناریو برای کاربرد خاص استفاده می‌نماییم[۲]. فرض می‌نماییم فردی برای کار اداری تصمیم به یک سفر دارد. پس از انجام جلسه باید رزرو هتل، تاکسی و پرواز را انجام دهد. فرد باید به‌صورت دستی وب‌سرویس را اجرا نموده تا خدمات فوق را برای وی فراهم آورد. از سویی ممکن است درطی روزهای کاری قبل از سفر برنامه خودش یا پروازها و هتل‌ها تغییراتی داشته باشد.

نکاتی که در این سناریو مورد توجه قرار گرفته‌اند عبارتند از:

• برای انجام یک برنامه‌کاری(فرآیند کسب و کار) فرد ناچار است به‌صورت دستی وب‌سرویس‌ها را فراخوانی نماید، ولی تلاش می‌گردد تا اجرای این وب‌سرویس‌ها به‌صورت خودکار صورت پذیرد. برای هر فرآیند باید تعدادی وب‌سرویس براساس تقاضای کاربر انتخاب و اجرا گردد. اجرای این وب‌سرویس‌ها می‌تواند به‌صورت خودکار انجام پذیرد. اما انتخاب وترکیب وب‌سرویس‌ها برای یک فرآیندکاری مشخص، به‌صورت ایستا در زمان طراحی صورت می‌پذیرد؛ بنابراین ایجاد یک فرآیند جدید ممکن نخواهد بود.

• در صورتی که هر یک از سرویس‌های استفاده شده توسط کاربر تغییری در سرویس آن‌ها رخ دهد باید به نحوی کاربر آگاه گردد. به‌عنوان مثال در صورتی که پرواز تاخیر داشته باشد، فرد باید مطلع گردد تا تغییراتی در برنامه کاری خود بدهد. مثلاً زمان مقرر برای تاکسی کرایه شده را باید تغییر دهد. همچنین باید محل کار فرد مطلع گردد تا تغییرات لازم (مانند قرار ملاقات‌ها) در سایر برنامه‌های کاری فرد اعمال گردد. به‌عبارتی باید تعامل بین سرویس‌گیرنده و فراهم‌آورنده سرویس صورت پذیرد تا از تغییرات سرویس‌ها مطلع گردند.

۱-۲ روش پیشنهادی براساس سناریو

در سمت سرویس‌گیرنده، فرآیند کسب و کار می‌تواند تعریف گردد. مثلاً یک کار می‌تواند شرکت در سمینار باشد، برای انجام این کار قاعدتاً نیاز به رزرو پرواز، هتل و غیره می‌باشد. اغلب کارهای یک سیستم مشابه هستند، یعنی برای انجام آن‌ها باید چند وظیفه صورت بگیرد. در اینجا فرض می‌شود که برای انجام این وظایف باید وب‌سرویس‌هایی انتخاب و اجرا گردند. طرح‌کار مشخص می‌نماید که برای یک کار چه وب‌سرویس‌هایی باید فراخوانی گردد. تعریف طرح یک کار با استفاده از قواعد فعال (ECA Rule) یک پایگاه داده فعال انجام می‌گردد. پایگاه دانشی برای نگهداری این قواعد در نظر گرفته می‌شود و همچنین اعمالی مثل درج، حذف و غیره برای آن تعریف می‌گردد. یکی از نکات مهم در این بخش ثبت درخواست برای مطلع کردن سایر کارها است، در صورتی که تغییر در سرویس یک وب‌سرویس به‌کارگرفته شده صورت پذیرد. مثلاً در صورتی که یک پرواز لغوگردد، باید مشخص گردد که آیا کل برنامه سمینار را لغو نماید یا سعی بر جایگزین نمودن یک سرویس نماید. در هرصورت باید کاربر یا همان برنامه‌کاربردی (تقویم کار فرد) مطلع شود تا تغییرات کارها انجام شود. پس باید با ایجاد قواعد فعال، تعامل فراهم کننده وب‌سرویس‌ها و سرویس‌گیرنده (کاربر) انجام پذیرد. اطلاعات وب‌سرویس‌های انتخاب و اجرا شده برای یک کار در پایگاه داده ذخیره می‌باشد و قواعد فعال لازم برای اعمال تغییرات ناشی از تغییر یک وب‌سرویس در مرحله طراحی مشخص می‌شوند. پس قواعد فعال برای این پایگاه داده شامل دو دسته قواعد خواهد بود: دسته اول قواعد ایجاد شده برای طرح کار

می‌باشـــد که در زمان اجرا توسط کاربر ایجاد می‌گردد و دسته دیگر قواعدی که به هنگام طراحی پایگاه برای اِعمال کردن تغییرات حاصل از یک رویداد ایجاد می‌گردد.

به‌طور خلاصه می‌توان روند انجام کار را بدین صورت تشـــریح کرد: اگـــر کاری در تقویم کاری فرد ثبت شـــود، به‌عنوان یک رویداد سبب فعال و اجرا شدن تعدادی قواعد (قواعد تعریف طرح کار) می‌گردد. وب‌ســـرویس‌های لازم انتخاب و اجرا می‌گردند. بنابراین اطلاعاتی از هر سرویس اجرا شـــده در پایگاه داده ثبت می‌شوند. نقطه تعامل فراهم کننده سرویس باید در این زمان مشخص شود. یعنی فراهم کننده وب‌سرویس در صورتی که تغییری داشته باشد باید روی پایگاه داده آن را اعمال کند. صدور گواهی برای فراهم کننده سرویس انجام می‌شود که بتواند به پایگاه دسترسی داشـــته باشد. در صورت تغییر در سرویس امکان درج در پایگاه داده برای فراهم کننده سرویس وجود دارد. درج در پایگاه داده به‌عنوان یک رویداد و براساس قواعد فعال سبب انجام اعمالی مانند جایگزین کردن سرویس و یا هماهنگی با ســـایر وب‌سرویس‌های درگیر در این کار خواهد شد. با خاتمه کار جداول پایگاه داده بروزسانی می‌شود.

در بخـــش ۲ به شـــرح مختصری دربـــاره پایگاه داده فعـــال می‌پردازیم. در بخش ۳ ضمـــن بیان معماری روش پیشـــنهادی، اجزای پایگاه داده فعال به‌کارگرفته شرح داده می‌شود. در بخش ٤ به ارزیابی روش پیشنهادی پرداخته و در بخش ٥ نتیجه‌گیری بیان خواهد شد.

۲- پایگاه داده فعال

پایگاه‌های دادهٔ متداول دارای ماهیت ایستا هستند، یعنی اعمال پرس‌وجو، به هنگام‌سازی، درج، حذف، گزارش‌گیری و ... هر زمان که توســـط کاربر درخواســـت شـــود انجام می‌گیرد و سیستم مدیریت پایگاه داده هیچ ابتکار عملی در هنگام رخ دادن شـــرایط خاص در سیستم ندارد. بسیاری از برنامه‌های کاربردی و سیســـتم‌های دارای محاســـبات پیچیده نیاز به نظارت خـــودکار دارند تا در صورت وقوع رویداد خاصی، سیســـتم مدیریت پایگاه داده عکس العمل مقتضی را انجام دهد. برای این منظور باید سیستم پایگاه داده‌ای طراحی شود تا امکان تعریف رویدادهای مورد نظر و واکنش‌های متناظر آن‌ها در آن گنجانده شـــود. به چنین سیستمی، سیســـتم پایگاه دادهٔ فعال گفته می‌شود. کار بر روی سیستم‌های پایگاه دادهٔ فعال از اوایل دههٔ ۸۰ شروع شده است ولی در ســـال ۱۹۹٦ مشخصه‌ها و ویژگی‌های آن استاندارد شد[٥]. برای بهینه‌شدن فرآیند درک رویدادها و انجـــام واکنش مقتضی، مفهومی بـــه نام قاعده فعال[1] که از قالب رویداد-شـــرط-عمل (ECA) [٦,٧] بهره می‌برد در سیستم‌های پایگاه دادهٔ فعال استفاده می‌شود.

ایده قاعده فعال در سیستم پایگاه داده فعال، همان ایده قاعده تولید در هوش مصنوعی اســـت. در سیستم هوش مصنوعی قاعده تولید چنین است:

عمل ← شرط

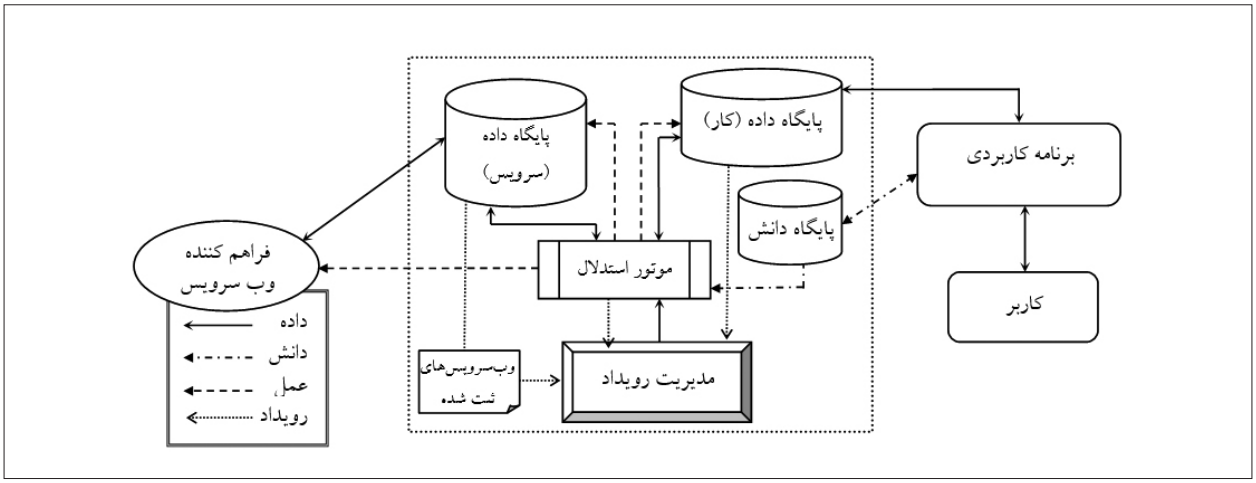
در این سیســـتم یک موتور اســـتنتاج[2]، تمام قواعد داده شده به سیستم را بررسی می‌کند. از بین تمام قواعدی که شرط داده شده در آن‌ها صحیح باشد (مجموعهٔ نامزد)، یک قاعده انتخاب و عمل تصریح شـــده در آن، انجام می‌شود. بررســـی قواعد، تا زمانی که دیگر شـــرطی صادق نباشد، ادامه می یابد.

ایده قاعده تولیـــد در هوش مصنوعی به‌صورت قاعده ECA در سیستم پایگاه داده فعال تعمیم داده شده است.

On event
If condition
Do action

ایـــن نحوه تعریف این امـــکان را فراهم می‌آورد که به جای این‌که یک موتور استنتاج داشته باشیم که به‌صورت دوره‌ای و در فواصل زمانی مشـــخص تمـــام قاعده‌ها را بررسی کند، قاعده در صورت وقوع یک رویداد فعال شود. رویداد عبارت اســـت از پدیده‌ای که در یک لحظه مشخص، در بیرون یا درون سیستم رخ می‌دهد. شرط به‌صورت یک گزاره داده می‌شـــود که پس از بروز رویداد باید ارزیابی

1- Active Rule
2- Inference engine



شکل ۱: معماری سطح بالای سیستم پیشنهادی

شود و عمل درصورت بروز رویداد و درست بودن نتیجه ارزیابی شرط انجام می‌شود.

یکی از ویژگی‌های مهم یک سیستم پایگاه داده فعال چرخه پردازش قاعده است. فرایند پردازش قاعده شامل پنج مرحلهٔ شناسایی رویداد، فعال‌سازی قاعده، انتخاب قاعده، کنترل شرط و اجرا می‌باشد. استفاده از قواعد فعال، یک سازوکار قدرتمند برای تعدادی از فعالیت‌های پایگاه داده است. از قبیل: اعمال محدودیت‌های جامعیتی، هشدارها و پیغام‌ها، نگهداری داده‌های مشتق شده، اعمال محدودیت‌های دستیابی. علاوه بر این‌ها استفاده از موتور استنتاج قواعد فعال، سیستم پایگاه داده فعال را برای ساخت یک سیستم خبره[3] و یا سیستم مبتنی بر دانش[4]، مناسب می کند[۸].

۳- معماری سیستم پیشنهادی و اجزای آن

همان‌طور که در بخش ۱-۲ بیان شد کاربر برنامه کاری خود را در یک برنامه کاربردی ثبت می‌نماید. سپس در صورتی که طرح کار مورد نظر، در پایگاه دانش مربوط به پایگاه داده فعال، قبلا ثبت شده باشد، براساس آن وب‌سرویس‌های لازم اجرا می‌گردد. سرویس‌های گرفته شده در یک پایگاه داده ثبت می‌شود. براساس قواعد فعال ثبت شده در پایگاه دانش در صورتی که رویدادی رخ دهد تغییرات لازم صورت می‌پذیرد. برای تعریف طرح‌کار برای یک کار جدید امکان افزودن و یا تغییر پایگاه دانش باید ایجاد گردد. اجزای سیستم پیشنهادی در شکل ۱ آمده است.

سیستم پیشنهاد شده قادر است براساس کار ثبت شده در پایگاه داده کار، وب‌سرویس‌های مرتبط را به‌صورت خودکار در پایگاه داده سرویس ثبت نماید. قابلیت انجام پرس وجو روی آن و تولید رویداد برای تغییرات را دارد. برای دانستن چگونگی تعامل، سیستم مجموعه‌ای از سیاست‌های احتمالی را در قالب قواعد فعال در پایگاه دانش نگهداری می‌نماید که وقتی رویدادی برسد آن را اجرا نماید. زمانی که سیاست‌های احتمالی چگونگی تعامل را برای سیستم مشخص کرد, یکسری اعمال انجام می‌گردد که می‌توانند سرویس‌های وب یا عملیات درونی پایگاه داده باشند. سیستم باید بداند به هنگام ثبت سرویس‌ها چه تعاملی(واکنشی) داشته باشد. این وب‌سرویس‌ها مجموعه‌ای از وب‌سرویس‌های ثبت شده هستند که با تغییری که در پایگاه داده می‌دهند سبب تولید رویداد می‌شوند.

در سیستم پیشنهادی کلیه رویدادها در اثر بروزرسانی پایگاه داده‌های کار و سرویس رخ می‌دهد. مثلاً کاربر از طریق برنامه کاربردی یک تغییر در برنامه را در پایگاه داده کار ثبت می‌کند که سبب ایجاد رویداد می‌شود. به

3- Expert system
4- knowledge-base system

همیـــن صورت وب‌ســـرویس‌ها هـــم اگر اعلانـــی[5] برای ســـرویس‌گیرنده داشته باشـــند، پایگاه داده سرویس‌ها را بروز کرده تا ســـبب یک رویداد گردد. براساس رویدادی که رخ می‌دهد موتور اســـتدلال قواعد فعـــال را انتخاب و اجـــرا می‌کند. نتیجـــه اجرای قواعد در قالـــب اعمالی روی پایگاه داده‌ها اعمال می‌شـــود. از طرفـــی این اعمال ممکن است درخواست سرویسی از یک فراهم کننده وب‌سرویس باشد. پایگاه داده سرویس‌ها از پایگاه داده کارها جدا گشته اســـت، چون پایگاه داده سرویس‌ها تحت دسترسی فراهم کننده وب‌سرویس‌ها می‌باشد و خود سیستم هم به‌عنوان یک وب‌سرویس باید به آن دسترسی داشته باشد.

ایجاد طرح کار، مشخص کردن وب‌سرویس‌های لازم و ترکیب آن‌ها برای انجام آن کار اســـت. زمانی که کاربر بخواهد طرح یـــک کار جدید را ایجاد نماید، با کمک برنامه کاربردی قواعـــد فعال لازم را طراحی کـــرده و در پایگاه دانش ثبت می‌نماید؛ از این قواعد برای ارتباط با پایگاه داده کار استفاده می‌نماید. قواعدی هم که مربوط به پایگاه داده سرویس‌ها می‌باشد در زمان طراحی سیستم ایجاد خواهد شد.

در ادامه این بخش به شرح بخش‌های پایگاه داده فعال به‌کار رفته می‌پردازیم.

۳-۱ زبان و مدل قواعد فعال

بخشـــی از دلایل تفاوت زبان قواعد در پایگاه داده‌های فعـــال مربوط به تفاوت مدل داده‌ای به‌کارگرفته شـــده در پایگاه می‌باشـــد[۹]. مثـــلاً در سیســـتم‌های رابطه‌ای مثل Starburst، POSTGRES، Ariel و محصولات تجاری قواعد به‌عنوان فراداده[6] در شِمای پایگاه داده‌ها تعریف می‌شود و در کنار جداول، دیدها، محدودیت‌های جامعیتی و ... ذخیره می‌شود. همانند ســـایر فراداده‌ها، دستوراتی برای اضافه کـــردن، حذف و تغییر قواعد وجود دارد. در سیســـتم‌های شـــیءگرا مانند HiPAC قواعد به‌عنوان شیءهای رده‌های اولیـــه که نمونه‌هایی از نوع قواعد تعریف شـــده در شِـــما می‌باشـــند، تلقی می‌گردد[۹]. در روش پیشـــنهادی فرض شده که مدل داده‌ای یک مدل رابطه‌ای می‌باشد. دلیل دیگر تنوع زبان‌های قواعد پیچیدگی توصیف رویدادها، شرط‌ها و عمل‌ها می‌باشـــد. در بعضـــی زبان‌ها ایجـــاد رویدادها به‌صورت ضمنی و در بعضی به‌صورت صریح می‌باشد. در روش پیشـــنهادی ما ایجاد رویدادها به‌صورت ضمنی می‌باشـــد. نحو قواعد در نظر گرفته شـــده به‌صورت زیر می‌باشد:

define rule rule-name **on** table
on event
if condition
then do action

شرط‌ها می‌توانند با عملگرهای منطقی AND و OR با هم ترکیب شوند. همان‌طور که در بخش قبلی بیان شد، ایجاد رویدادها براســـاس بروزرســـانی پایگاه داده‌ها می‌باشد. بنابراین رویدادهای لازم همان دستورات دستکاری پایگاه داده ماننـــد Delete، Insert و Update خواهـــد بود. البته دستورات در بخش بعدی بیان می‌گردد.

۳-۲ نحو زبان توصیف قواعد فعال

همان‌طـــور که بیان شـــد، کاربـــر نیـــاز دارد که برای تعریف طرح یـــک کار قواعد فعالی را ایجاد نماید. همچنین در صورت نیاز قواعـــدی را حذف و یا تغییر دهد. تعریف نحو زبان توصیف قواعد از [۱۰] اقتباس شـــده است. زبان توصیف قواعد starburst حاوی ۵ دســـتور می باشـــد که برای تعریف و دستکاری قواعد به‌کار می‌روند:

create rule, alter rule, deactivate rule, activate rule, drop rule.

همچنین امکان گروه‌بندی قواعد وجود دارد که تعریف و دســـتکاری آن‌ها بـــا دســـتورات create ruleset، alter ruleset و drop ruleset انجام می‌شود.

ایجاد یک قاعـــده جدید با دســـتور create rule انجام می‌شود. نحو این دستور به‌صورت زیر است:

create rule name on table
On triggering-operation

5- notification
6- meta data

[**if** condition]
then action-list

name نام قاعده را مشخص می کند.

table نـــام جدولی که این قاعده بـــر روی آن تعریف می‌شود.

triggering-operation رویـــدادی را کـــه قاعده در پی بروز آن فعال خواهد شد مشخص می‌کند. رویدادها در این سیستم پایگاه داده فعال، تغییرات در داده‌های پایگاه است. در سیســـتم‌های پایگاه داده رابطه‌ای این تغییرات از طریق دستورات insert,delete,update انجام می‌شود. سیستم پایگاه داده فعال، قواعدی را شامل می‌شود که با اعمال تغییر در پایگاه داده فعال می‌شوند. متناسب با رویداد مورد نظر این متغیر یکی از مقادیر updated، deleted، inserted را دارا خواهد بود. برای updated می‌توان لیست ستون‌های مورد نظر را هم مشـــخص کرد و در صورتی که مشخص نشـــود، به طور پیش فرض تمام ســـتون‌ها در نظر گرفته خواهد شـــد. هر قاعده می‌تواند حاوی یـــک یا چند رویداد باشـــد. در بخش بعدی به‌طور مفصـــل رویدادها توصیف شده است و علاوه بر رویدادهای فوق سایر رویدادها نیز مطرح شده است.

Condition شـــرطی را که باید در صورت فعال شدن قاعده ارزیابی شود مشخص می‌کند. این شرط می‌تواند یک پرسش ساده از پایگاه باشد و در این صورت شرط درست اســـت اگر و فقط اگر تعداد چندتایی برگشـــتی این پرسش حداقل یک باشد. گذاشتن شرط برای قواعد اختیاری است و می‌تواند مورد اســـتفاده قرار نگیـــرد. در این صورت با بروز رویداد عمل مورد نظر انجام خواهد شـــد و به‌عبارت دیگر شرط همواره درست است.

action-list لیست عمل‌هایی است که با بروز رویداد و درست بودن شرط باید اجرا شوند. هر عمل می‌تواند یکی از دستورات دستکاری داده‌ها (update، delete، insert و se-lect) یا دســـتورات تعریف داده‌ها (مثل drop، create table rule) و یا rollback باشد. عمل‌ها به ترتیب اجرا می‌شوند.

در قســـمت condition و action قواعـــد می‌تـــوان به جـــداول پایگاه یا به جداول گذار[۷] ارجـــاع داد. جداول گذار موجـــود عبارتنـــد از: new-updated، deleted، inserted و old-updated. اگـــر قاعده‌ای به‌عنـــوان رویداد خود (در قسمت inserted، (triggering-operation را روی جدول T تعیین کند آنگاه جدول منطقی inserted حاوی چندتایی‌های جدیدی اســـت کـــه به جدول T اضافه و باعث فعال شـــدن قاعده شده است. و به‌همین صورت برای deleted تعریف می‌گردد. new-updated حاوی مقادیر جدید چندتایی‌ها و old-updated حاوی مقادیر قبلی چندتایی‌هاست.

پس از ایجاد یک قاعده می‌توان آن را تغییر داد. برای این منظور از دستور alter rule با نحو زیر استفاده می‌شود:

alter rule name on table
[**if** condition]
[**then** action-list]

توضیح قسمت‌های مختلف آن شبیه توضیحاتی است کـــه برای ایجاد قاعده گفته شـــد. عبـــارت On قاعده، قابل تغییر نیست. برای تغییر این قسمت، یک قاعده باید حذف و دوباره ایجاد شود.

دستور حذف قاعده به‌صورت زیر است:

drop rule name **on** table

گاهی اوقات لازم اســـت که یک قاعده به‌صورت موقتی غیرفعال شـــود، برای این منظور از دســـتور deactivate rule name on table اســـتفاده می‌شود. برای این‌که آن را مجددا فعال کنیم از دستور activate rule name on table استفاده می کنیم.

امـــکان گروه‌بندی قواعـــد وجـــود دارد. از گروه‌بندی می‌توان به منظور تعریف اولویت اشتراکی یا فعال کردن و غیر فعال کردن قواعد به‌صورت دسته جمعی استفاده کرد.

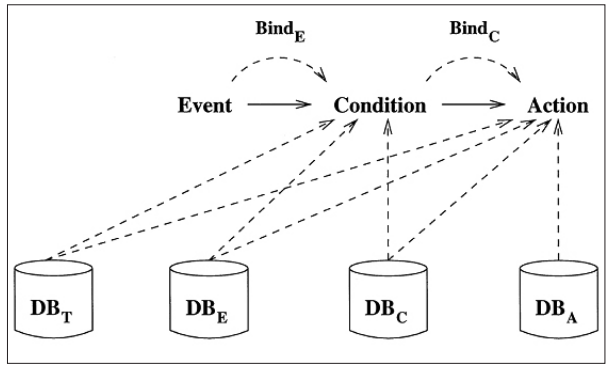
نحو دستور ایجاد یک گروه قاعده به‌صورت زیر است:

create ruleset name

برای تغییـــر تعریف یک گـــروه و افزودن و کاســـتن اعضای آن از دستور زیر استفاده می‌شود:

alter ruleset name
[**addrules** rule-list]
[**delrules** rule-list]

7- Transition Table



شکل۲ : وضعیت‌های مختلف در پردازش شرط [۱۱]

بـــرای حذف یک گروه تعریف شـــده، از دســـتور زیر استفاده می‌شود:

drop ruleset name

۳-۳- توصیف رویدادها

یک رویداد چیزی اســـت که در یـــک نقطه در زمان رخ می‌دهـــد. بنابراین تعیین یک رویداد شـــامل فراهم آوردن توصیفی از رخدادی است که باید بررسی گردد. توصیف و روشـــی که رویداد می‌تواند شناســـایی گردد، بسیار به منبع تولیدی رویداد بستگی دارد[۱۱]. در پایگاه داده فعال پیشنهادی منبع رویدادها عبارتند از:

– عملیات ساختار: رویداد می‌تواند به‌دلیل رخ دادن یک عمل روی پایگاه داده باشد. مثلاً اضافه کردن یک چندتایی، بروزرسانی صفت یا دستیابی به یک.

– تراکنش‌هـــا: فرمان‌های تراکنش می‌توانند ســـبب رخ دادن رویداد گردند مانند commit، abort و یا شروع کردن یک تراکنش.

– ساعت: در یک زمان مشـــخص رویدادی رخ می‌دهد. مثلاً درســـاعت ۳ روز ۵ اردیبهشـــت ۹۷ یا ۲ساعت پس از انجام پـــرواز. این رویداد بدین دلیل لحاظ شـــده که پس از انجام سرویس، داده‌های مربوط به آن از جداول حذف گردد.

یکی دیگر از مشخصه‌های رویدادها دانه‌بندی رویدادها می‌باشـــد[۱۱]. رویدادها می‌توانند برای یک مجموعه (مثلاً سرویس‌های یک کار)، زیر مجموعه داده‌شده (سرویس‌های یک کار که تمام شده‌اند) یا عضو مشخصی از یک مجموعه رخ می‌دهد.

نوع یـــک رویداد به‌طور کلی می‌توانـــد اولیه یا مرکب باشد. یک رویداد اولیه یک رویداد ساده مثل اضافه کردن یک چندتایی می‌باشد. رویداد مرکب ترکیبی از رویدادهای اولیه با استفاده از عملگرهایی می‌باشد. عملگرها می‌توانند ترکیب عطفـــی و فصلی از رویدادها باشـــند. عملگر توالی (Seq) تعییـــن می‌کند که یـــک رویداد باید قبـــل از رویداد دیگری انجام شود. (شاید عملگرهایی مثل closure و not بعدا به‌کار آید). در این سیســـتم رویدادها ساد فرض شده اســـت. ولی به‌نظر می رســـد که نیاز به رویدادهای مرکب برای اجرای کامل وجود داشته باشد.

۳-۴ توصیف شرط

شرط یک قاعده فعال نشـــان می‌دهد که آیا باید کاری انجام شـــود. یک قاعده فعال می‌تواند دارای شرط نباشد. هر بخـــش از قاعده، جـــدا از پایگاه داده یا ســـایر اجزای دیگـــر آن ارزیابی نمی‌گردد. پـــردازش یک قاعده می‌تواند درچهار وضعیت پایگاه داده باشـــد[۱۱]: DBT پایگاه داده در شـــروع تراکنش جاری می‌باشـــد؛ DBE پایگاه داده در زمانی کـــه رویداد رخ می‌دهد؛ DBC پایـــگاه داده وقتی که که شـــرط ارزیابی شده اســـت؛ DBA پایگاه داده وقتی که عمل[۸] انجام شـــده است. اطلاعات در دسترس برای اجزای مختلف قواعد در شـــکل ۲ بیان شـــده اســـت. شـــرط‌های به‌کار گرفته شـــده در بخش توصیف قواعد توضیح داده می‌شود.

۳-۵ توصیف عمل‌ها

اعمال ممکن اســـت بروزرسانی پایگاه داده یا مجموعه قواعد باشد. همچنین می‌تواند اطلاع دادن به کاربر یا برنامه کاربـــردی نیز به‌عنوان یـــک عمل باشـــد. در ضمن برای برقراری ارتباط با فراهم کننده سرویس برای درخواست، ثبـــت یا لغو یک ســـرویس باید از اعمالـــی در قواعد فعال تعریف نماییم. اعمال به‌کار گرفته شده در بخش توصیف قواعد توضیح داده می‌شود.

8- Action

۳-۶ پردازش و زمانبندی قواعد فعال

چرخه پردازش قواعد شامل پنج مرحله (۱): تشخیص رویدادها،(۲): فعال‌سازی قواعد مرتبط با رویدادهای رخ داده، (۳): ارزیابی بخش شرط قواعد فعال، (٤): زمان‌بندی قواعد آماده اجرا و(٥) اجرای دستورات بخش عمل قاعده‌های که در مرحله پیشین برای اجرا انتخاب شده است، می‌باشد. شایان ذکر است که مجموعه دستورات بخش عمل یک قاعده در قالب یک تراکنش اجرا می‌شوند[۱۲]. دو فاکتور بسیار مؤثر در نحوه اجرای چرخه پردازش قواعد، پیوستگی رویداد- شرط و پیوستگی شرط- عمل قواعد است. این دو پیوستگی دارای سه حالت مختلف‌اند: (۱) فوری، (۲)تعویقی و(۳) مستقل. اگر پیوستگی رویداد- شرط یک قاعده از نوع فوری باشد، شرط آن قاعده بلافاصله پس از رخ دادن رویداد متناظرش و فعال شدن قاعده، ارزیابی می‌شود. به همین ترتیب اگر پیوستگی شرط- عمل یک قاعده از نوع فوری باشد، پس از فعال شدن آن قاعده و ارزیابی شرط مربوطه، در صورت مثبت بودن نتیجه ارزیابی، دستورات بخش عمل آن قاعده فوراً اجرا می‌شوند. اگر پیوستگی رویداد- شرط (شرط-عمل) یک قاعده از نوع تعویقی باشد، ارزیابی شرط (اجرای عمل) آن قاعده تا زمان اتمام تراکنش جاری به تأخیر می‌افتد. در برخی موارد ارزیابی شرط (اجرای عمل) تا رخ دادن رویداد خاصی به تأخیر خواهد افتاد. در حالت مستقل، ارزیابی شرط (اجرای عمل) قاعده در قالب یک تراکنش مستقل و پس از اتمام تراکنش جاری انجام می‌شود. روش مورد استفاده برای زمان‌بندی قواعد به طور مستقیم در عواملی مانند زمان پاسخگویی سیستم به تراکنش‌ها، زمان بازگشت تراکنش‌ها، توان عملیاتی سیستم و به‌طور کلی در کارایی سیستم پایگاه داده فعال، بسیار مؤثر است.

در سیستم مدیریت پایگاه داده فعال به فرایند تخصیص اولویت برای اجرا به قواعد آماده اجرا زمان‌بندی قواعد گویند. قواعدی آماده اجرا هستند که اولاً در اثر وقوع رویداد متناظرشان در سیستم، فعال شده باشند و ثانیاً بخش شرط شان در لحظه ارزیابی درست باشد. روش‌ها زمان‌بندی به‌طور کلی شامل موارد زیر هستند[۱۲]:

در روش زمان‌بندی اتفاقی هرگاه که در چرخه پردازش قواعد نیاز به انتخاب قاعده جدیدی برای اجرا باشد از بین قواعد فعال در سیستم به‌صورت اتفاقی یکی برای اجرا انتخاب می‌شود. استفاده از اولویت ایستا روش دیگری است که برای انتخاب قواعد و زمان‌بندی آن‌ها استفاده می‌شود. در این روش به هر یک از قواعد یک عدد صحیح به‌عنوان اولویت نسبت داده می‌شود. سپس در هنگام انتخاب قواعد برای اجرا قاعده که در بین قواعد فعال موجود دارای کوچک‌ترین اولویت باشد برای اجرا انتخاب می‌شود. روش زمان‌بندی FCFS[9] از جمله روش‌های کلاسیک زمان‌بندی است که برای زمان‌بندی اجرای قواعد در سیستم‌های پایگاه داده فعال نیز استفاده می‌شود. در این روش هر قاعده که در سیستم در حالت فعال قرار گرفت یک برچسب زمانی که مشخص کننده زمان فعال شدن آن قاعده است همراه آن قرار می‌گیرد. درهنگام انتخاب و زمان‌بندی اجرای قواعد، قاعده‌ای که دارای کوچکترین برچسب زمانی باشد برای اجرا انتخاب خواهد شد.

روش مبتنی بر نزدیک‌ترین ضرب‌الاجل کامل‌ترین روش‌هایی است که تاکنون برای زمان‌بندی قواعد در سیستم‌های پایگاه داده فعال بیدرنگ ارایه شده است و به اختصار EDF نامیده می‌شود. در این روش، هنگام انتخاب، قاعده‌ای که دارای نزدیک‌ترین ضرب‌الاجل باشد، برای اجرا انتخاب می‌شود.

قواعدی را که در سیستم پیشنهادی استفاده می‌گردند می‌توان به سه دسته تقسیم کرد. اولین دسته قواعدی هستند در جهت تخصیص وب‌سرویس و تخصیص کار حرکت می‌نمایند. دومین دسته عکس اولین دسته می‌باشند. این دسته در جهت لغو یک کار و وب‌سرویس حرکت می‌نمایند. دسته دیگر قواعدی هستند که براساس برش زمانی باید کاری انجام دهند. ما برای زمان‌بندی قواعد

9- First Come First Serve

از روش ایستا و FCFS اســتفاده می‌نماییم. یعنی بالاترین اولویت به دســته سوم قواعد و سپس دســته دوم و اول تخصیص پیدا می‌نماید. در داخل یک دسته هم هر قاعده‌ای که زودتر فعال شده بود اجرا خواهد گردد.

۴- ارزیابی

در بخــش ۱ و توصیف ســناریو دو هدف مهم مطرح گردیــد: اولین هــدف انتخــاب و ترکیب وب‌ســرویس‌ها به‌صورت خودکار، پویا و در زمان اجرا بود. دومین هدف تامین تعامل بین سرویس‌گیرنده و فراهم‌آورنده سرویس مطرح شد. همان‌طور که در روش پیشنهادی مطرح گردید، با فراهم آوردن امکان طرح یک ســفر با استفاده از تعریف قواعد فعال خواســته مطرح شــده در هدف اول برآورده گردید. از سویی امکانی فراهم شد که وب‌سرویس‌های ثبت شــده بتوانند در پایگاه داده سرویس‌ها تغییراتی را اعمال نماینــد. عملا امکان ثبت اعلان وب‌ســرویس با این روش فراهم شد، که طبق روش پیشــنهادی با تغییر اعمال شده در پایگاه داده ســرویس‌ها با استفاده از قواعد فعال اعمال لازم انجام می‌شــود. با این امکان عملا خواســته دوم هم مرتفع گردید.

طبــق معیارهای مطــرح شــده در مرجــع [۲] روش پیشنهادی ارزیابی می‌گردد:

• باید نقطه‌ای برای ورود اطلاعات وجود داشــته باشد و اطلاعات باید فقط یکبار وارد شــود. اطلاعات یکسان چند بار وارد نشوند، به‌دلیل: الف) کاهش افزونگی ب) جلوگیری از ناســازگاری اطلاعــات وارد شــده. همان‌طــور که در معماری پیشــنهادی مشاهده می‌شود، اطلاعات فقط یکبار توســط کاربر و از طریق برنامه‌کاربردی وارد می‌شــود. برای کارهای مشابه هم پس از تعریف طرح کار، فقط نیاز به ورود کار می‌باشد و تکرار ورود اطلاعات سرویس‌های لازم حذف می‌شود.

• نیاز به یــک معماری کلی[10]. برای هر کاربرد مشــخص یک راه‌حل خاص کاربرد مورد نظر وجود خواهد داشــت. بنابراین باید یک معماری کلی ارائه گردد و بتوان برای هر کاربرد پیمانه‌های سفارشی شده افزود، دلیل: معماری قابل انعطاف و عملی برای اقتباس(تغییر شــکل). همان‌طور که بیان شد این امکان با تعریف طرح کار با استفاده از قواعد فعال فراهم گردیده است.

• مدیریت محیط مشــتری. مشتری مســئولیت نظارت بر محیــط و تغییرات رخ داده و پاســخ بــه آن و در صورت لزوم رویداد را به سیســتم واگذار می نماید. بنابراین باید امکان تعریف متن (محیط) توســط مشــتری فراهم گردد. خصوصیات محیط مشتری و اعمالی که انجام می‌دهد, به همراه حالت (state) مشــتری در حال و آینده از مواردی می‌تواند باشد که در این بخش تعریف می‌گردد. این عمل از طریق تنظیماتی که کاربر می‌تواند در برنامه‌های کاربردی انجام دهد، لحاظ شده است.

• کنترل مشــتری بر محیط. مشــتری باید به‌طور صریح مشخص نماید که کنترل چه بخش‌هایی از محیط خودش را به سیستم واگذار می نماید و سایر بخش‌ها توسط خودش یا ســایر نهادها کنترل می‌گردد. در تعریف قواعد و تغییر آن‌ها این امکان فراهم می‌شــود. مثلاً می‌توان قاعده‌ای را به‌صورتی تعریف کرد که اگر یک سرویس تغییری داشت به کاربر اطلاعات دهد. همچنین می‌توان همین قاعده را به نحوی تغییر داد که خود سیستم چایگزینی برای سرویس فراهم آورد و سایر سرویس‌ها را تنظیم نماید.

• صدور گواهی سرویس. برای ارتباط بین وب‌سرویس‌ها به‌صورت خودکار نیاز اســت که این ارتباط تعریف گردد، به‌دلیل: قابلیــت اطمینان و قابلیت اتکا بودن ســرویس‌ها. این امر تا حدودی با تهیه لیســت ثبت شده سرویس‌ها در معماری پیشنهادی قابل لحاظ کردن خواهد بود.

می‌توان روش پیشــنهادی را برای ترکیب سرویس‌ها [۱۳] توســعه داد. نکتــه دیگر این‌که در روش پیشــنهادی وب‌ســرویس موظف به اعلان تغییرات در وب‌ســرویس می‌شــود که این کار با تغییر در پایــگاه داده باعث ایجاد رویداد می‌شد. یکی از مهم‌ترین چالش‌ها زمان‌بندی قواعد

10- generic

مربوط به پایگاه داده کار اســت. چون کاربر آن را تعریف می‌نماید باید دربــاره زمان‌بندی پویــای آن طرحی ارائه نمود. دسترسی همزمان به پایگاه داده سرویس‌ها توسط فراهم کننده وب‌ســرویس‌ها از نکات دیگر میباشد که باید بررسی گردد.

۵- نتیجه‌گیری

معماری سرویس‌گرا با ایجاد یک اتصال سست نسبت به تکنیک‌های قبلی، توانایی یکپارچه‌ســازی در سیستم و سازمان‌ها را دارا می‌باشــد. مفهوم "publish-find-bind" در معماری سرویس‌گرا سبب گشته توسعه نرم‌افزارهای فراهم کننده سرویس به‌صورت جداگانه و مستقل از توسعه نرم‌افزارهای سرویس‌گیرنده انجام گردد. واضح است که نقش توســعه‌دهنده برای چگونگی تعامــل فراهم کننده و گیرنده ســرویس مهم می‌باشد. توســعه دهنده نرم‌افزار سرویس‌گیرنده، در زمان توســعه نرم‌افزار سرویس‌های لازم را مشــخص می‌نماید. گرچه امکان جستجو و اجرای سرویس در فراهم‌کنندگان ســرویس وجود دارد، اما باید هم سرویس‌گیرنده و هم سرویس‌دهنده دانش لازم درباره همدیگر را داشته باشند. این امر احتمال تعامل پویای زمان اجرای فراهم‌کننده ســرویس و اســتفاده کننده را کاهش می‌دهد. در این پژوهش مشکل در قالب یک سناریو مطرح گردید. براســاس سناریوی مزبور دو هدف مشخص شد. اولین هــدف انتخاب و ترکیب وب‌ســرویس‌ها به‌صورت خــودکار، پویا و در زمان اجرا بود. با فراهم آوردن امکان طرح یک ســفر با اســتفاده از تعریف قواعد فعال خواسته مطرح شــده در ایــن هدف برآورده گردیــد. دومین هدف تامین تعامل بین سرویس‌گیرنده و فراهم‌آورنده سرویس مطرح شد. امکانی فراهم شد که وب‌سرویس‌های ثبت شده بتوانند در پایگاه داده سرویس‌ها تغییراتی را اعمال نمایند. عملا امکان ثبت اعلان وب‌سرویس با این روش فراهم شد، که طبق روش پیشــنهادی با تغییر اعمال شــده در پایگاه داده سرویس‌ها با استفاده از قواعد فعال اعمال لازم انجام می‌شود. با این امکان عملا خواسته دوم هم مرتفع گردید. در ادامه پژوهش پیاده‌ســازی از روش پیشنهادی صورت خواهد گرفت. همچنین بخش تعریف طرح کار با ابزارهای مدیریت فرآیندهای کسب و کار جایگزین خواهد گردید.

مراجع

1. M. P. Papazoglou, P. Traverso, S. Dustdar and F. Leymann, "Service-Oriented Computing: A research roadmap," International Journal of Cooperative Information Systems, vol. 17, pp. 223-255, Jun, 2008.
2. Vijay Dheap, Paul A.S. Ward “Event-Driven Response Architecture for Event-Based Computing”, Proceedings of the 2005 conference of the Centre for Advanced Studies on Collaborative research, Toranto, 2005
3. Lane, Stephen, Qing Gu, Patricia Lago, and Ita Richardson. “Towards a framework for the development of adaptable service-based applications.” Service Oriented Computing and Applications 8, no. 3 (2014): 239-257.
4. Liu, Yaya, Jiulei Jiang, Lingyu Xu, and Lishuang Zhao. “Formalization of Business Process with Flexibility Based on Service Interaction.” In Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence & Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), 2017 IEEE 15th Intl, pp. 351-355. IEEE, 2017.
5. K. Dittrich, S.Gatziu, A.Geppert, eds.(The ACT-NET Consortium), “The active database management system manifesto: a rule base of ADBMS features”, Sigmod Record, 1996. 11
6. H. Theodore. “A survey of Active Database Systems”, available in http://www.doc.ic.ac.uk/~twh1/academic/papers/active.ps, April, 1997. 1
7. A. Vadua. “Rule Development for active database”, PhD Thesis, CS Department, University of Zurich, 1999. 12
8. Umeshwar Dayal and Eric N. Hanson and Jennifer Widom , “Active Database Systems” , 1994
9. Won Kim, “Modern Database Systems: The Object Model Interoperability and Beyond”, Addison Wesley 1994
10. Jennifer Widom , “The Starburst Active Database Rule System” , Stanford University ,
11. N. W. PATON, OSCAR DI´AZ, “Active Database Systems” ACM Computing Surveys CSUR) ,Volume 31 , Issue 1 ,Pages: 63 – 103,1999 .
12. R. Alesheykh, A. Abdollahzadeh, “Evaluation and Comparison of Rule Scheduling Approaches in Active Database Systems”, in Automation, Control and Applications (ACI ACA 2005) conference, Novosibirsk, Russia, 2005.
13. Z. Laliwala, “Event-driven Service-oriented Architecture for Dynamic Composition of Web Services", Doctoral Symposium, ACM Compute 2008, India , January 2008