

انتخاب مورد آزمون برای آزمون رگرسیون برنامه های کاربردی با استفاده از

وب سرویس ها براساس تغییر مشخصات WSDL

چکیده

اشتیاق زیادی نسبت به وب سرویس ها در جهان امروز وجود دارد. وب سرویس ها از اینترنت برای ارتباط بین دو دستگاه الکترونیکی متصل از طریق شبکه استفاده می کنند. آزمون وب سرویس چالشی است که یک درخواست کننده ی سرویس کد منبع را ندارد و به نوعی نیاز به تست کامل تاثیر تغییرات در نرم افزار دارد. تست رگرسیون یکپارچگی نرم افزار را تایید می کند و اطمینان حاصل می کند که تغییرات خطا های نرم افزار جدید معرفی شده است. روش ما شامل تجزیه فایل WSDL XML برای استخراج اطلاعات مربوط به نام عملیات، پیام های ورودی و پیام های خروجی است. هر دو فایل اصلی و تغییر یافته ی XML برای وب سرویس برای استخراج اطلاعات مربوطه ی خود از نوع پورت و عنصر پیام WSDL تجزیه شده است. پس از آن، یک جدول هش از اطلاعات استخراج شده برای هر دو WSDL اصلی و دلتا ایجاد شده است. جدول هش به یک مقایسه کننده به عنوان ورودی وارد می شود، پس از آن جدول هش مقایسه شده و تغییرات عملیات را به صورت خروجی تولید می کند. در مرحله ی آخر آزمون موارد برای تست رگرسیون از وب سرویس های تغییر یافته انتخاب خدمات بر اساس تغییرات در عملیات ارائه شده توسط مقایسه کننده انتخاب می شود.

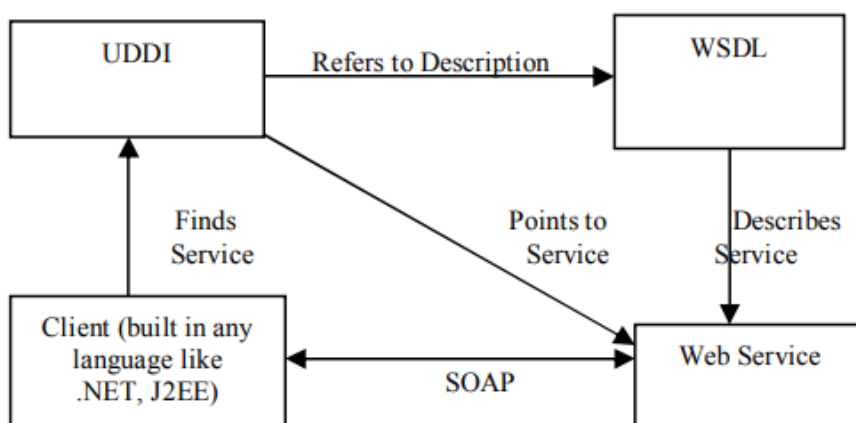
کلمات کلیدی: وب سرویس ها؛ آزمون رگرسیون؛ جدول هش. WSDL

اشتیاق زیادی برای وب سرویس ها در جهان امروز وجود دارد. وب سرویس ها از اینترنت برای ارتباط بین دو دستگاه الکترونیکی متصل از طریق شبکه استفاده می کنند. وب سرویس در اصل یک سیستم نرم افزاری است که عملکرد سازگاری برای حمایت از تعامل ماشین به ماشین برای انتقال داده ها در یک شبکه از خود نشان می دهد [1]. آن ها برنامه های کاربردی استاندارد وب هستند که مشخصات آنها در دایرکتوری UDDI که با دیگر برنامه های کاربردی وب به منظور تبادل اطلاعات ارتباط برقرار می کنند انتشار شده است. وب سرویس ها از پنج استاندارد اصلی برای برقراری ارتباط در طول شبکه استفاده می کنند: زبان توصیف وب سرویس ها (WSDL) [3]، زبان توسعه یافته ی (XML)، پروتکل انتقال متن (HTTP)، پروتکل ساده دسترسی به شی (SOAP) [2] و توضیحات جهانی، کشف و یکپارچه سازی (UDDI).

وب سرویس ها دارای یک درخواست کننده ی سرویس و ارائه دهنده ی خدمات است. نرم افزاری که داده درخواست می کند درخواست کننده ی سرویس نامیده می شود و نرم افزاری که درخواست درخواست کننده را پردازش می کند و داده را ارائه می کند ارائه دهنده ی خدمات نامیده می شود. کد منبع برای وب سرویس با ارائه دهنده ی خدمات است. درخواست کننده ی سرویس تنها WSDL دارد. بنابراین، هر زمان که یک تغییر در وب سرویس رخ می دهد، تست مجدد وب سرویس مورد نیاز است. تست یک چالش برای درخواست کننده ی سرویس است همان طور که کد منبع را ندارد و نیاز به تست کامل تاثیر تغییرات در درخواست خود را دارد. وب سرویس ها رابط گرافیکی کاربر ندارد. در عوض، آنها از یک رابط برنامه ریزی برای تبادل پیام استفاده می کنند [7].

وب سرویس ها در طول زمان مانند هر نرم افزار دیگری تکامل می یابند. به همین ترتیب، ما نیاز به انجام آزمون رگرسیون هنگام تغییر در وب سرویس داریم. تست رگرسیون انجام شده است تا اطمینان حاصل شود که نسخه ی تغییر یافته ی وب سرویس در حال کار کردن است و تمام عملیات به درستی انجام می شود. آزمون رگرسیون یکپارچگی نرم افزار را تایید می کند و اطمینان حاصل می کند که تغییرات، خطاهای جدیدی در نرم افزار به وجود نیاورده است [6]. آزمون رگرسیون وب سرویس چالش بیشتری از فراهم آوردن حداکثر آزمون پوشش برای ادغام با نرم افزار برای تضمین حداقل تعداد موارد آزمون با کمترین هزینه ارزیابی و خطر است.

[4] ها و پارک یک روش پیشنهادی از هستی شناختی دارند، به طوری که آنها کیفیت سرویس در سطح کاربر را به کار می‌برند (QoS) که دو سطح مختلف برای وب سرویس با کیفیت مناسب را ارائه می‌دهند. تا به امروز تلاش زیادی برای تحقیق در تست وب سرویس [7] برای یکپارچه سازی با برنامه‌های کاربردی دیگر انجام شده است. آزمون رگرسیون وب سرویس منطقه‌ای است که نیاز به تحقیق و بهبود بهره‌وری و زمان تست دارد. بسیاری از روش‌های موجود برای آزمون رگرسیون از خدمات وب سرویس بر اساس تست های براساس کد است. آنها محدودیت‌های خود را به‌عنوان درخواست‌کننده سرویس جهت آزمایش کد منبع برای وب سرویس دارند. یک رویکرد مبتنی بر مشخصات برای تست وب سرویس‌ها توسط مسعود و نادم پیشنهاد شده است [8]. ما رویکرد آنها را گسترش داده و برخی از تغییرات را در XML به منظور کاهش انتخاب مورد آزمون رگرسیون وب سرویس‌ها پیشنهاد داده‌ایم.



شکل 1: پیاده‌سازی وب سرویس

رویکرد مبتنی بر مشخصات با بهره‌گیری از WSDL برای وب سرویس برای انتخاب موارد آزمون جهت تست مجدد وب سرویس اجرا می‌شود. WSDL یک زبان تعریف رابط مبتنی بر XML است. که عملکرد و عملیات ارائه شده توسط یک وب سرویس را توصیف می‌کند. WSDL شامل تمام جزئیات در مورد عملیاتی است که می‌تواند توسط وب سرویس انجام شود- نام عملیات، نوع پورت، انواع پیام، پیام ورودی، پیام خروجی و صحافی. نوع پورت شامل اطلاعات مربوط به نام عملیات موجود در وب سرویس و پیام‌های مرتبط با آن است. پیام‌ها شامل عناصر داده‌ها از عملیات مانند نام پارامتر و انواع آنها و نوع خروجی عملیات است.

رویکرد ما شامل تجزیه فایل WSDL XML برای استخراج اطلاعات مربوط به نام عملیات، ورودی پیام و خروجی است. هر دو فایل XML اصلی و تغییر یافته برای وب سرویس برای استخراج اطلاعات مربوطه از نوع پورت و پیام عنصر از WSDL تجزیه می‌شود. پس از آن، یک جدول هش از اطلاعات استخراج شده برای هر دو WSDL اصلی و دلتا تولید می‌کنیم. جداول هش را به یک مقایسه‌کننده به عنوان ورودی وارد می‌کنیم، که پس از آن جداول هش مقایسه شده و تغییرات عملیات به عنوان خروجی تولید می‌شود. در آخرین مرحله موارد آزمون رگرسیون برای تست وب سرویس تغییر یافته بر اساس تغییرات در عملیات ارائه شده توسط مقایسه‌کننده انتخاب می‌شود. موارد آزمون با عملیات در وب سرویس به‌طور مستقیم رابطه برقرار می‌کند. مجموعه تست ما موارد آزمون همراه با نام عملیات را حفظ می‌کند. انتخابگر آزمون رگرسیون تنها کسانی که مورد آزمون هستند انتخاب می‌کند که در آن یک تغییر در نام عملیات و یا ارزش‌های مرتبط و یا یک عملیات جدید اضافه شده وجود دارد. موارد آزمون دیگر که در آن هیچ تغییری در سیستم وجود ندارد همچنین ممکن است برای آزمون رگرسیون انتخاب شود. این کار به‌منظور حصول اطمینان از رفتار نرم‌افزار همانند قبل است. تمام قطعات سیستم در رگرسیون تحت پوشش تست تضمین ایمنی هستند [10]. آزمون ایمنی رگرسیون تمام موارد آزمون برای سیستم را به منظور درستی آزمون پوشش می‌دهد.

بقیه مقاله برای توصیف رویکرد پیشنهادی به جزئیات نوشته شده است. سازماندهی مقاله به شرح زیر است: بخش دوم به کارهای گذشته اختصاص داده شده است و بخش سوم به تشریح جزئیات روش پیشنهادی می‌پردازد. بخش چهارم یک نتیجه‌گیری برای مقاله است و بخش پنجم شامل دیدگاه‌های آتی برای تحقیق و بحث است.

2. کارهای گذشته

وب سرویس با تغییر در تکنولوژی و تکامل شیوه‌های صنعت استنتاج شده است. هنگامی که تغییرات رخ می‌دهد، آزمون رگرسیون یک فعالیت مهم و بسیار گران قیمت به منظور حصول اطمینان از ایجاد اختلال توسط این تغییرات است که ویژگی‌های سیستم نرم‌افزار کاربردی وب را بعد از تغییرات بررسی می‌کند. آزمون رگرسیون می‌تواند:

1. براساس کد: تکنیک های بر اساس کد، تغییرات موجود در کد منبع ساخته شده در نرم افزار را مشخص می کنند. موارد آزمون بر اساس تغییرات در کد ساخته شده برای نرم افزار، انتخاب شده است. بنابراین، تکنیک های بر اساس کد برای زبان های برنامه نویسی مورد استفاده برای توسعه کد منبع بسیار خاص است.

2. براساس مدل: این تکنیک ها موارد آزمون رگرسیون را با نگاه به مدل های سیستم های مختلف تولید می کنند. بیشتر تکنیک های مبتنی بر مدل بر اساس زبان مدل سازی (UML) است.

3. براساس مشخصات: این روش براساس مشخصات و تغییرات در ویژگی های ارائه شده توسط مشتری می باشد. آزمون رگرسیون انجام می شود تا اطمینان حاصل شود که تمام مشخصات عنوان شده توسط مشتری برآورده شده است و سیستم امن است.

روت و همکارانش تحقیقات خود را در وب سرویس ساخته شده در جاوا انجام دادند. آنها یک روش آزمون رگرسیون امن برای این سرویس ها پیشنهاد دادند. نرم افزار تحت آزمون برای تحلیل ایستا و پویا از کد تجزیه و تحلیل می شود. یک گراف کنترل (CFG) بر اساس کد جاوا با نام نمودار درون جاوا (JIF) نامیده شده است. پس از آن، JIF برای هر دو کد اصلی و جدید ایجاد می شود. یک مقایسه بین JIF و یال های بالقوه خطرناک شناسایی می شود. در نهایت، موارد آزمون بر اساس یال خطرناک انتخاب می شود. آنها اعتبار رویکرد را با یک ابزار شبیه سازی مورد استفاده برای شناسایی یال خطرناک نشان داده اند [12].

روت و همکارانش به طور کلی کار خود را در وب سرویس های خاص جاوا به تمام وب سرویس ها گسترش دادند. آنها از یک چارچوب مشابه برای آزمون رگرسیون امن برای سرویس های وب عمومی استفاده کردند. گراف کنترل جریان (CFG) برای همه ی وب سرویس های تحت آزمون ایجاد شده است. سپس، یک مقایسه بین CFG برای وب سروی های قدیمی و جدید به منظور برجسته کردن تغییرات در وب سرویس انجام گرفته است. آنها همچنین پیشنهاد به انتشار موارد آزمون [13] همراه با مشخصات WSDL دادند، که می تواند در انتخاب موارد آزمون برای رگرسیون مفید واقع شود.

پنتا و همکاران، یک ابزار برای تولید موارد آزمون کدگذاری XML برای آزمون رگرسیون پیشنهاد دادند [7]. موارد آزمون به عنوان یک قرارداد بین سیستم و ارائه دهنده ی خدمات استفاده شد. برنامه های کاربردی بر اساس

سرویس‌ها پویایی را به عنوان یک عامل مهم در نظر داشتند. هدف اصلی از روش ارائه شده انجام انتخاب موارد آزمون در طول حالاتی از نرم افزار است که از خدمات استفاده می‌کند.

برای رسیدن به یک روش امن آزمون رگرسیون، برخی از اظهارات در کیفیت سرویس (QoS) و سناریوها ساخته شد. آنها این تغییرات در مشخصات را در وریکرد خود در نظر نگرفتند. جعبه ابزار برای تجزیه و تحلیل آزمون JUnit استفاده شد و پس از آن آزمون کدگذاری XML انجام شد [7].

در طول این مرحله از پژوهش، همه‌ی تکنیک‌های تست رگرسیون عمدتاً بر اساس کد بودند. تکنیک‌های تست رگرسیون براساس برای وب سرویس‌ها پیشنهاد شد که در آن مدل برای شناسایی تغییرات و تاثیر این تغییرات در نرم افزار ایجاد شده است. مدل‌ها برای توصیف رابط سرویس هستند. اتوماتا‌های با حالت محدود برای رفتار بیرونی به‌کار گرفته شدند. برای ایجاد وابستگی داده‌ها، گراف وابستگی دوبخشی ایجاد شده است. در این روش نمودار و کلاس‌ها توسط گره نشان داده است. سپس یک الگوریتم برای انتخاب مورد آزمون ارائه شده است [14]. بای و همکارانش بیشتر به تحقیق مشخصات WSDL پرداختند. آنها داده‌های آزمون را بر اساس عملیات مشخص شده در مشخصات وب سرویس WSDL تولید کردند. آنها همچنین به دنباله‌ای از زمان عملیات در وب سرویس توجه کردند. رابط WSDL برای تولید داده‌های تست برای تست مورد تحلیل قرار گرفت. رویکرد آنها در آزمون رگرسیون برای وب سرویس‌ها متمرکز نیست.

مطالعه‌ی دیگری، رویکرد مشخصات WSDL را برای انجام آزمون رگرسیون از برنامه‌های کاربردی مبتنی بر وب سرویس مورد استفاده قرار داده است. مشخصات WSDL برای انتخاب موارد آزمون برای آزمون رگرسیون مورد استفاده قرار گرفت [8]. مشخصات WSDL XML تجزیه و تحلیل شد و تغییرات نوع داده انتخاب شد. سپس درختی از XML تجزیه شده، تولید شد. درخت برای هر دو وب سرویس اصلی و تغییر یافته تولید شده است. پس‌از آن، یک مقایسه بین درختان و موارد آزمون با استفاده از انتخاب‌گر آزمون رگرسیون انجام شده است. این انتخاب‌گر از تجزیه و تحلیل مقدار مرزی برای شناسایی موارد آزمون استفاده می‌کند. آنها یک ابزار خودکار برای حمایت از رویکرد خود ساخته‌اند.

مسعود و نادیم رویکرد خود را به عناصر شامل نوع پورت برای شناسایی تغییرات در مشخصات WSDL توسعه دادند. عنصر نوع پورت شامل عملیات نام، پیام‌های ورودی و پیام خروجی از خصوصیات WSDL است. یک درخت

برای هر دو WSDL اصلی و تغییر یافته ایجاد شده است. مقایسه بین درختان به منظور انتخاب موارد آزمون در مجموعه آزمون رگرسیون انجام شده است.

رویکرد ما عنصر نوع پورت از مشخصات WSDL [8] را گرفته و سپس یک جدول هش برای هر دو WSDL اصلی و تغییر یافته ایجاد می‌کند. پس از آن، مقایسه ای بین جداول هش، تغییرات در اسامی عملیات برای وب سرویس را نشان می‌دهد. براساس شناسایی تغییرات در عملیات، موارد آزمون برای گنجانده شدن در آزمون رگرسیون انتخاب می‌شود.

3. روش ارائه شده

نرم افزار تست رگرسیون برای تعریف اشکالات غیر عمدی و خطاهایی که ممکن است در کد در نتیجه‌ی اعمال تغییرات در نرم افزار کاربردی ایجاد شود، استفاده می‌شود. سهولت توسعه‌ی وب سرویس‌ها و وابستگی‌های متقابل بین خدمات و نرم افزار، فشار زیادی بر آزمایش‌کنندگان وب سرویس به منظور حصول اطمینان از قابلیت اعتماد، قوی، مقیاس پذیر و امن بودن وب سرویس‌ها را تحمیل می‌کند. آزمون رگرسیون امن در تضمین کار وب سرویس در طول چرخه‌ی عمر نرم افزار است.

آزمون مجدد وب سرویس اصلاح یافته با استفاده از تکنیک‌های آزمون رگرسیون می‌تواند بسیار پر هزینه باشد. درخواست‌کننده‌ی سرویس کد ندارد، بنابراین نمی‌تواند تغییرات در نرم افزار وب سرویس را ادغام کند و آن را به‌طور کامل برای رفتار ایمن تست کند. هزینه آزمون رگرسیون برای یک وب سرویس را می‌توان به‌طور قابل توجهی با شناسایی تغییرات در وب سرویس و انتخاب موارد آزمون کاهش داد. این رویکرد از ایجاد هزینه در موارد آزمون جدید و بی‌فایده و نه چندان پربار اجرای دوباره‌ی آزمون موجود جلوگیری می‌کند و می‌تواند تضمین کند که کد بدون تغییر وب سرویس نتایج مشابه‌ای همانطور که قبلاً تولید می‌کرد تولید خواهد کرد. رویکرد ما از خصوصیات WSDL برای شناسایی تغییرات در وب سرویس و در نتیجه انتخاب موارد آزمون مربوط به بخش تغییر نرم افزار بهره می‌برد. مشخصات WSDL شامل توضیحات وب سرویس و تمام جزئیات عملیاتی که پشتیبانی می‌کند، است. اجزای مهم WSDL عبارتند از: نوع پورت، اتصال، انواع و پیام‌ها است.

پیام: می‌تواند ورودی یا خروجی باشد. یک داده را تعریف می‌کند که در حال منتقل شدن است.

نام عملیات: نام عملیات اجرا شده توسط وب سرویس را ارائه می‌دهد.

پیام ورودی: پارامتر ورودی انتزاعی و نوع اطلاعات را ارائه می‌دهد.

پیام خروجی: خروجی انتزاعی داده شده توسط وب سرویس را ارائه می‌دهد.

اتصال عنصر: عملیات و جزئیات پیام و فرمت پیام را ارائه می‌دهد.

نوع پورت: تمام عملیاتی که بخشی از وب سرویس هستند ارائه می‌دهد. تمام این عملیات توسط وب سرویس

انجام می‌شود.

نوع: تعریف نوع داده را فراهم می‌کند. این یک توصیف از پیام‌های رد و بدل شده‌ی استفاده شده توسط وب

سرویس را ارائه می‌دهد.

```
<wsdl:portType name="Service1Soap">
  <wsdl:operation name="AddNumbers">
    <wsdl:input message="tns:AddNumbersSoapIn"/>
    <wsdl:output message="tns:AddNumbersSoapOut"/>
  </wsdl:operation>
  <wsdl:operation name="SubtractNumbers">
    <wsdl:input message="tns:SubtractNumbersSoapIn"/>
    <wsdl:output message="tns:SubtractNumbersSoapOut"/>
  </wsdl:operation>
</wsdl:portType>
```

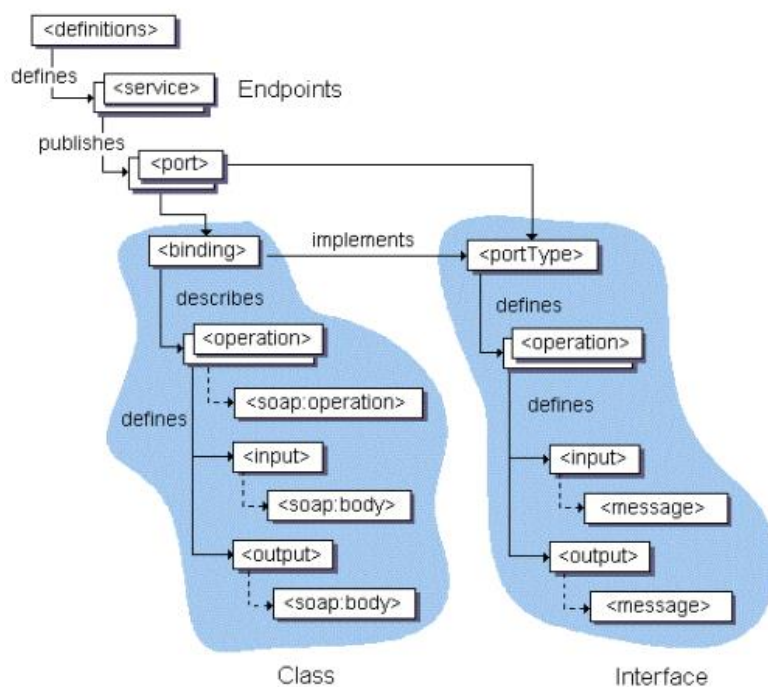
شکل 2: پیام WSDL

```
<s:element name="AddNumbers">
  <s:complexType>
    <s:sequence>
      <s:element name="number1" type="s:int" maxOccurs="1"
        minOccurs="1"/>
      <s:element name="number2" type="s:int" maxOccurs="1"
        minOccurs="1"/>
    </s:sequence>
  </s:complexType>
</s:element>
```

شکل 3: نوع پورت WSDL

ما تغییرات در نام عملیات، نام عنصر و عناصر WSDL را پیگیری می‌کنیم. وب سرویس اصلی را به عنوان خط پایه‌ی وب سرویس تعریف می‌کنیم. وب سرویس تغییر یافته با عنوان وب سرویس Delta نامیده می‌شود. رویکرد ما سه جزء، تجزیه‌کننده‌ی XML، مقایسه و انتخاب‌گر آزمون رگرسیون دارد.

پارسر XML: عملکرد تجزیه‌کننده‌ی XML به تجزیه‌ی WSDL XML برای هر دو نسخه‌ی پایه و دلتا از وب سرویس است. پارسر XML هر عنصر از WSDL را استخراج می‌کند و آن را در یک جدول هش نگه می‌دارد. نام عملیات از عنصر نوع پورت، کلید جدول هش را تشکیل می‌دهد. هر عملیات یک کلید منحصر به فرد برای جدول هش ایجاد می‌کند. تمام ارزش‌های مربوط به هر عملیات یعنی خروجی، تعداد پارامتر، نام عنصر و نوع عنصر در یک لیست ذخیره می‌شود. این فهرست ارزش هر کلید در جدول هش را درست می‌کند. هر کلید عملیات فهرست ارزش‌های خود را دارد. الگوریتم تجزیه‌کننده‌ی XML در شکل 5 نشان داده شده است. و بر اساس اصولی کار می‌کند که نام عملیات در وب سرویس منحصر به فرد است و می‌تواند به عنوان کلید در جدول هش استفاده شود.



شکل 4: روابط عناصر WSDL

جریان اطلاعات در WSDL به شرح زیر آمده است [16].

Input: WSDL Specification XML

Output: Hash table H

Algorithm: Parse WSDL XML and store it in hash table.

1. Select the <PortType> element of the WSDL specification.
2. Select the <operation> element under the <PortType> element. Select the name of the operation and assign it as the key for the hash table <key, value> pair.
Operation Name is the unique key value for the Hash table.
3. Create an empty List L which will store all the values related for a particular Key.
4. Select the <input message> element and count the number of <sequence> elements associated with the <input message>. This count will give the number of parameters for the selected operation name.
5. Add the count to the List L.
6. Select the <input message> element and using the input message name extract the parameter names and their respective types for the selected operation name.
7. Add the parameter names and their respective values to the List L.
8. Select the <output message> element and using the output message name extract the return type for the selected operation name.
9. Add the return type value to the List L.
10. Link the List L as the value part for the selected operation name key in the Hash table.
11. Repeat Steps 2-10 for all the <operation> elements in the WSDL.

شکل 5: الگوریتم برای تجزیه‌کننده‌ی XML

هنگامی که نام عملیات انتخاب شد، یک لیست خالی برای ذخیره سازی تمام اطلاعات مربوطه‌ی دیگر با توجه به عملیات: نوع خروجی، تعداد پارامتر، نام پارامتر ورودی و نوع پارامترهای ورودی ایجاد می‌شود. به این ترتیب ما قادر به ذخیره‌ی اطلاعات مربوط به هر عمل در WSDL در یک جدول هش هستیم. پارسر XML برای هر WSDL پایه و دلتا اجرا می‌شود.

Key	Value
Operation Name	Return type Parameter count Input parameter name 1 Input parameter name 1 type Input parameter name 2 Input parameter name 2 type . . . Input parameter name N Input parameter name N type

شکل 6: جدول هش

جدول هش: جدول هش تولید شده، نام عملیات را به عنوان کلید دارد و ارزش‌ها به عنوان یک لیست برای هر کلید ذخیره می‌شود. فرمت جدول هش در شکل 6 نشان داده شده است.

<p>Input: Baseline Hash table H Delta Hash table H'</p> <p>Output: List Of keys from the <key, value> pair with value changes → L1 List of keys added in the hash table H' → L2 List of keys deleted from the hash table H' → L3</p> <p>Algorithm:</p> <ol style="list-style-type: none"> 1. For each key opkey in delta hash table H', Repeat step 2. 2. If hash table H contains opkey <ol style="list-style-type: none"> 2.1 Compare the List of values associated with the opkey. 2.2 If all the values do not match, add the opkey to List L1 Else Add the opkey to List L2 3. For each key oldkey in hash table H, Repeat step 4. 4. If hash table H' does not contain oldkey <ol style="list-style-type: none"> 4.1 Add the oldkey to List L3
--

شکل 7: مقایسه‌کننده‌ی XML

مقایسه‌کننده: هنگامی که ما جداول هش برای WSDL پایه و دلتا داریم، آنها را به یک مقایسه‌کننده وارد می‌کنیم که ارزش بر اساس کلید عملیات مقایسه می‌شود. مقایسه‌کننده برای اولین بار جدول هش دلتا را اسکن می‌کند. برای هر کلید عملیاتی که در جدول هش پایه وجود دارد، مجموعه‌ای از ارزش‌ها مقایسه شده است. در موردی که در آن ارزشها مطابقت ندارد، کلید عملیات مربوطه در یک لیست ذخیره می‌شود. این فرایند تا زمانی که تمام کلیدها در جدول هش دلتا پیموده شوند تکرار می‌شود.

در موردی که، هیچ نگاشت مربوطه‌ای برای کلید عملیات در جدول هش پایه پیدا نشود، آن در یک لیست جداگانه ذخیره می‌شود.

در حال حاضر، جدول هش پایه پیموده شده است. ما در جستجوی نام عملیاتی هستیم که در حال حاضر در جدول هش دلتا نیست. این عملیات در سرویس تغییر یافته در دسترس نیستند. هر گونه اشاره‌ای به آنها باید از نرم‌افزار حذف شده و نرم‌افزار به‌طور کامل آزمایش شود. همه‌ی موارد آزمون مربوط به این عملیات باید حذف یا از مجموعه جریان تست بایگانی شوند.

انتخاب‌کننده‌ی تست رگرسیون: ما یک مجموعه تست پایه داریم که در آن هر تست اجرا شده با نام عملیات در ارتباط است. تغییرات در کلید عملیات از مقایسه‌کننده برای انتخاب می‌کند، اضافه کردن، ویرایش و حذف موارد آزمون از مجموعه آزمون پایه. ما می‌توانیم موقعیت‌های مختلف بر اساس تغییرات عملیات داشته باشیم.

1. تغییرات در ارزش‌های مربوط به عملیات‌های کلیدی: تمام موارد آزمون مربوط به کلید عملیات، بررسی و در صورت لزوم اصلاح شده است. آزمون رگرسیون انجام شده است تا اطمینان حاصل شود که رفتار وب سرویس مانند قبل از تغییرات است.

2. بدون تغییر در عملیات‌های کلیدی: مخازن تست مربوط به کلید عملیات ممکن است اجرا نشوند و در مجموعه تست دلتا وجود داشته باشند.

3. اضافه شدن کلید عملیاتی: مورد آزمون جدید باید برای کلید عملیات ایجاد شده و به‌طور کامل به عنوان رفتار مورد انتظار مورد آزمایش قرار گیرد. این موارد آزمون نیز در مجموعه تست دلتا وجود خواهد داشت.

4. حذف کلید عملیاتی: تمام موارد آزمون مربوط به کلید عملیات از مجموعه تست حذف خواهند شد. به‌عنوان وب سرویس انجام عملیات، برنامه نیاز به تغییر و تست برای هر ارجاع به عملیات منسوخ دارد.

در حال حاضر، آزمون رگرسیون می‌تواند براساس انجام مجموعه تست دلتا باشد. ما در حال حاضر لیستی از تغییرات نام عملیات داریم و هر آزمون اجرا شده با نام عملیات در ارتباط است. ما می‌توانی به‌طور انتخابی فقط آن دسته از موارد آزمون که در ارتباط با عملیات اصلاح شده در وب سرویس هستند، اجرا کنیم. بنابراین اجرای یک آزمون رگرسیون کامل و کارآمد از سیستم را مورد اطمینان قرار می‌دهد.

بنابراین یک رویکرد جامع جهت حفظ و انتخاب یک مجموعه آزمون رگرسیون است.

4. نتیجه‌گیری

در این مقاله ما یک مورد آزمون بر اساس مشخصات ارائه شده برای انتخاب آزمون رگرسیون وب سرویس ارائه می‌دهیم. رویکرد ما از مشخصات WSDL و تجزیه عناصر آن را به یک جدول هش استفاده می‌کند. تحقیقات نشان داده‌است که، در برخی شرایط، جداول هش کارآمدتر و سریع‌تر از درخت جستجوی دیگر و یا هر ساختار دیگری هستند.

Time Complexity in Big O Notation	Hash Table		Binary Tree	
	Average	Worst	Average	Worst
Space	$O(n)$	$O(n)$	$O(n)$	$O(n)$
Search	$O(1)$	$O(n)$	$O(\log n)$	$O(n)$
Insert	$O(1)$	$O(n)$	$O(\log n)$	$O(n)$
Delete	$O(1)$	$O(n)$	$O(\log n)$	$O(n)$

شکل 8: پیچیدگی زمانی جدول هش و درخت دودویی

ما رویکرد مسعود و نادیم را گسترش دادیم و از جدول هش برای ذخیره‌ی WSDL XML به جای یک درخت عملیات استفاده کردیم. استفاده از جدول هش باعث می‌شود مقایسه الگوریتم بسیار ساده‌تر باشد. همچنین، بر اساس پیچیدگی زمانی جدول هش به طور متوسط بهتر از درختان جستجو است. انتخاب‌کننده‌ی آزمون رگرسیون از شناسایی تغییرات عملیات توسط مقایسه‌کننده استفاده می‌کند. این مبنای انتخاب موارد آزمون برای وب سرویس دلتا از وب سرویس پایه را تشکیل می‌دهد. از این رو، یک روش انتخاب آزمون رگرسیون امن فراهم می‌کند.

5. کارهای آینده

در آینده قصد داریم کارمان را با خودکار کردن تولید آزمون و فرآیند انتخاب برای آزمایش رگرسیون بر روی خروجی مقایسه‌کننده ارتقا دهیم. پژوهش ما بر روی نام عملیات به صورت کلیدی برای جدول هش با فرض عملیات منحصر به فرد انجام شده است. ما می‌توانیم با استفاده از یک کلید ترکیبی برای جدول هش، بهره‌وری جدول هش را افزایش دهیم.

REFERENCES

- [1] K. Gottschalk, H. Kreger, J. Snell and S. Graham, "Introduction to Web Services Architecture", IBM Systems Journal, NO 2, VOL 41, pp. 170- 177, (2002)
- [2] Simple Object Access Protocol (SOAP) 1.2, Part 2, "World Wide Web Consortium" - Adjuncts: (2007), <http://www.w3.org/TR/soap12-part0/>
- [3] Web Services Description Language (WSDL) 2.0, part 1: "World Wide Web Consortium" - Core Language (2007) , <http://www.w3c.org/TR/wsdl20/>
- [4] . Ha and H-S. Park, "QoS based Client Information for Semantic Web Service" - International Journal of Software Engineering and Applications, Vol. 3, No. 1 (2009)
- [5] M-H. Lee, C-J.Yoo and O-B. Jang, "Embedded System Software Testing Based on SOA for Mobile Service" - International Journal of Advanced Science and Technology, Vol.1 (2008)
- [6] R. Binder, "Testing Object-Oriented Systems: models, patterns and Tools", ISBN-10: 0321700678, Addison-Wesley Professional; Edition 1, ISBN-13: 978-0321700674 (2000)
- [7] M. D. Penta, G. Esposito, M. Bruno, G. Canfora and V. Mazza, "Web Services Regression Testing and Test and Analysis of Web Services", Barresi, L., Nitto, E.D. (eds.) "Test and Analysis of web Services", pp. 205-234. Springer, Heidelberg (2007)
- [8] T. Masood, A. Nadeem and G. S. Lee, "A safe Regression Testing Technique Based on WSDL Specification" - Software Engineering Business Continuity and Education Communications in Computer and Information Science, Springer Berlin, Heidelberg (2011)
- [9] T. Masood, A. Nadeem and S. Ali, "An automated approach to regression testing of web Services based on WSDL operation changes" – Emerging Technologies(ICET), IEEE 9th International Conference 2013
- [10] XML Schemas - Part 2, "Data types - World Wide Web Consortium", <http://www.w3.org/TR/xmlschema-2/>
- [11] P. C. Jorgensen, "Software Testing: A Craftsman's Approach", second ed., LLC, CRC Press, 2002.
- [12] M. Ruth, F. Lin and S. Tu., "Applying Safe Regression Test Selection Techniques to Java Web Services", 1, 10 - (2006)
- [13] M. Ruth, S. Tu, S. Oh, A. Loup, B. Horton, O. Gallet and M. Mata, "Towards Automatic Regression Test Selection for Web Services", 31st Annual International Computer Software and Applications Conference, July 2007 24-27, Beijing, China.
- [14] T. A. Khan and R. Heckel., "A Methodology for Model based Regression Testing of Web Services", Academic and industrial Conference - Practice and Research Techniques, pp. 123-124, (2009)
- [15] X Bai, W Dong, WT Tsai, Y Chen, "WSDL-based Automatic Test Case Generation for Web Services Testing", IEEE International Workshop on Service-Oriented System Engineering (SOSE), Planned Parenthood.215- 220. IEEE Computer Society, (2005), Los Alamitos
- [16] http://download.oracle.com/otn_hosted_doc/jdeveloper/1012/web_services/ws_wsdlstructure.html