

آزمون سیستم‌های اطلاعات سلامت با استفاده از شبکه پتری

محمد بیاتی^۱، رامین نصیری^۲

^۱ کارشناس ارشد مهندسی کامپیوتر نرم‌افزار، دانشگاه آزاد اسلامی واحد علوم و تحقیقات.
^۲ عضو هیئت علمی گروه مهندسی کامپیوتر، دانشگاه آزاد اسلامی تهران مرکز.

نام نویسنده مسئول:

محمد بیاتی

چکیده

در معماری‌های سیستم‌های اطلاعاتی سلامت از استانداردها و چارچوب‌هایی مختلفی مانند HL7 برای تبادل، اشتراک‌گذاری و بازیابی اطلاعات استفاده می‌شود، HL7 بیشترین پیاده‌سازی و کاربردی‌ترین استاندارد را در بین سیستم‌های اطلاعات سلامت دارد اما به دلیل پیاده‌سازی‌های متفاوت در کامپوننت‌های مختلف و تفاوت بسیار زیاد نسخه دوم با سوم HL7 از وب سرویس معنایی برای تعامل‌پذیری بیشتر سیستم‌های اطلاعات سلامت استفاده می‌شود. یکی از چالش‌های موجود در سیستم‌های اطلاعاتی سلامت که با استفاده از استاندارد HL7 و وب سرویس معنایی تعامل‌پذیر شده، آزمون آن است. در این مقاله آزمون سیستم‌های اطلاعاتی از دیدگاه کاربر نهایی صورت گرفته و به عنوان یک اصل برای آزمون است، بر همین اساس WSMO به دلیل وجود المان توصیف هدف به عنوان روش پیاده‌سازی وب سرویس معنایی انتخاب شده است، نمونه‌های آزمون بر اساس دیدگاه کاربران نهایی پیاده‌سازی و طراحی می‌شود، به این صورت که نگاشت‌هایی بین المان توصیف هدف، زبان WSMO و المان‌های شبکه پتری صورت می‌گیرد و شبکه پتری ایجاد شده به گراف تبدیل می‌شود و در نهایت نمونه‌های آزمون توسط گراف تهیه می‌شود. داده‌های ورودی برای ایجاد نمونه آزمون توسط روش MCDC و شرایط مرزی تولید شده و در نهایت نمونه آزمون‌های ایجاد شده توسط روش Mutant که مستقل از روش تولید نمونه آزمون است ارزیابی خواهد شد تا میزان اثربخشی و کارایی نمونه‌های آزمون بدست آید.

واژگان کلیدی: وب سرویس معنایی - شبکه پتری - سیستم‌های اطلاعات سلامت -

WSMO-HL7

مقدمه

رشد روز افزون فناوری اطلاعات که از آن می‌توان به انقلاب فن‌آوری اطلاعات و ارتباطات یاد کرد، گستره وسیعی از صنایع را تحت پوشش قرار داده است. یکی از حوزه‌های پیشگام در امر کاربرد فناوری، حوزه بهداشت و درمان است. سیستم‌های اطلاعات بیمارستانی یا سیستم‌های اطلاعات درمان، سیستم‌های کامپیوتری هستند که برای مدیریت بهتر اطلاعات پزشکی و مدیریتی بیمارستان و حوزه درمان در جهت بهبود کیفیت مراقبت‌های بهداشتی طراحی شده‌اند. [۱]

در سیستم‌های سلامت از مشخصه‌های استاندارد HL7 به منظور تعامل‌پذیری استفاده می‌شود. اما چالشی که اینجا مطرح می‌شود این است که استانداردها هر روزه به سمت تکامل پیدا کردن در پیش هستند و به همین دلیل چالش تعامل‌پذیری همچنان به صورت یک مسئله باقی می‌ماند. به عنوان مثال می‌توان به عدم تطابق نسخه‌های ۲ و ۳ HL7 نام برد. در نسخه ۲ این استاندارد از دیتا مدلی استفاده نشده است ولی به منظور تعامل‌پذیری از آپشن‌های زیادی استفاده شده است ولی در نسخه ۳ این استاندارد از یک دیتا مدل مرجع استفاده شده است همانطور که ملاحظه می‌شود دو مدل با یکدیگر تطابق ندارند و رفتن از یک نسخه به نسخه دیگر به هزینه و زمان و کار زیادی احتیاج دارد. همچنین به دلیل اینکه این مدل در کامپیوترهای مختلف با کامپوننت‌های مختلف پیاده‌سازی می‌شود تعامل آنها با یک اینترفیس انجام می‌شود مانند وب سرویس، فایل یا دیتابیس برای توصیف این اینترفیس‌ها از تعامل معنایی می‌توان استفاده کرد که این تعامل معنایی با وب سرویس معنایی قابل انجام گرفتن است. [۲]

در معماری سرویس‌گرا ساخت سرویس مبتنی بر تکنولوژی مستقل و جدا از هم است. این معماری برای ارتباط بین سیستم‌های بیمارستانی تنها به تعریف قاعده و ساختار ارتباط بین سرویس دهنده و سرویس گیرنده بسنده کرده است و از لحاظ معنایی در نظر گرفته نشده است به همین دلیل سرویس‌ها توسط کامپیوترها به وسیله یکدیگر فهمیده نمی‌شوند و بر همین اساس کشف سرویس، انتخاب سرویس، درخواست سرویس و ترکیب سرویس انجام نمی‌شود اما با بهره‌گیری از وب سرویس معنایی می‌توان این مشکل را حل کرد. [۳]

پیش از این آزمون‌کننده برای آزمون وب سرویس‌های معنایی از روش‌های سنتی استفاده می‌کرد، به این صورت که آزمون‌کننده فقط خصوصیتی که فراهم‌کننده وب سرویس مشخص کرده، آزمون می‌کرد و این آزمون‌ها تنها از دید فراهم‌کننده مورد ارزیابی قرار می‌گرفت اما آزمون‌کننده دیدگاه کاربرنهایی رو هم در نظر داشته باشد باعث ایجاد آزمون‌های درست و کامل می‌شود زیرا سیستم نهایتاً باید در اختیار کاربر نهایی قرار گیرد و به همین دلیل مهم است وب سرویس بر اساس دیدگاه کاربر نهایی مورد ارزیابی قرار گیرد. WSMO (Web Service Modeling Ontology) به عنوان چهارچوبی مفهومی برای توصیف معناشناسی تمام جنبه‌های مربوط به وب سرویس‌های معنایی به منظور تسهیل اتوماسیون خدمات در وب شناخته شده است. به همین منظور با کمک المان‌های سطح بالا به طور صریح به بیان نیازهای کاربران می‌پردازد. [۵]

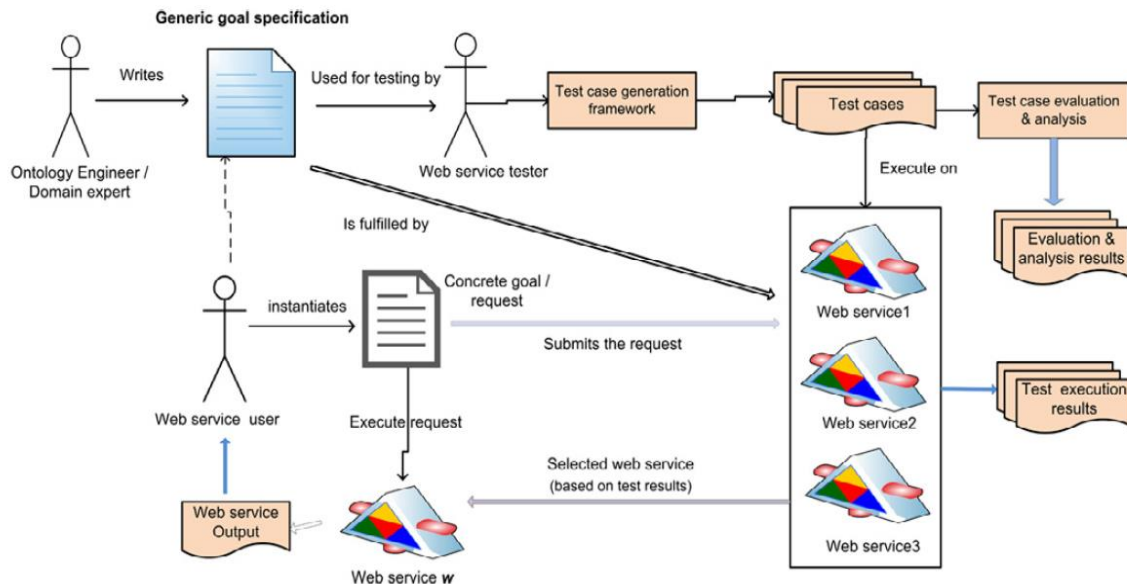
از شبکه‌های پتری نیز به طور گسترده برای تجزیه و تحلیل و تبیین سیستم‌های همروند، غیرهمزمان، توزیع شده، موازی، غیر قطعی و اتفاقی استفاده می‌شود. [۴]. شبکه‌های پتری اجازه تجزیه و تحلیل متفاوتی بر روی مدل می‌دهد تا خصوصیات در دسترس‌پذیری، محدود بودن، بن بست، زنده بودن، برگشت‌پذیری و نگهداری سیستم تجزیه و تحلیل کند. همچنین این مدل میزان پوشش مورد کاربرد را بررسی می‌کند. تحقیقات گسترده‌ای برای شبکه پتری و ابزارهای آن صورت گرفته است که منجر به ترکیب نشانگذاری‌ها با علائم ریاضی شده است و همین امر باعث اعتبارسنجی وب سرویس‌ها می‌شود و این اعتبارسنجی سرویس‌ها باعث کاهش هزینه توسعه سیستم‌های مبتنی بر سرویس می‌شود. از دلایلی که می‌توان برای انتخاب شبکه پتری به عنوان مدلی برای اعتبارسنجی معماری نام برد منطبق بود شبکه پتری همانند WSMO بر اساس ASM (Abstract State Machine) است همچنین سادگی شبکه پتری برای مدل سازی و ایجاد مدلی بدون پیچیدگی و ماهیت ریاضی آن به عنوان ابزاری برای مدل سازی در انتخاب آن تاثیر گذار بوده است. [۶-۷]

هدف نهایی ما در این مقاله آزمون سیستم‌های اطلاعاتی بیمارستانی با توجه قابلیت رویکردی آن به معماری سرویس‌گرا و استاندارد HL7 و پیاده‌سازی آن توسط وب سرویس معنایی با مشخصه WSMO است. بنابراین برای آزمون معماری مذکور نیازهای کاربران نهایی مورد توجه قرار می‌گیرد، نیازمندی‌های کاربران نهایی در المان توصیف هدف قرار گرفته و با توجه به اینکه تمرکز اصلی شبکه پتری بر روی روال‌های نیازهای کاربران نهایی پس بهترین گزینه برای تولید نمونه برنامه آزمون می‌تواند باشد.

۱- تولید نمونه آزمون بر اساس شبکه پتری

شبکه پتری یکی از ابزارهای مدل‌سازی و آزمون سیستم‌های مبتنی بر سرویس است. روش پیشنهاد شده برای تولید نمونه آزمون برای وب سرویس با توجه به ساختار شبکه پتری بر اساس پوشش خطا است. با تجزیه و تحلیل WSMO، وب سرویس‌های معنایی به شبکه

پتری سطح بالا تبدیل می‌شود. یک وب سرویس برای اجرا، نیازمند گرفتن مجموعه‌ای از ورودی‌ها، توابع اجرایی و خروجی مورد انتظار است برای تولید نمونه آزمون ابتدا تبدیل *WSMO* به شبکه پتری باید انجام شود این تبدیل حاوی نیازمندی‌های کاربران است. روش ارائه شده در این بخش بر اساس نگاشت المان توصیف هدف به شبکه پتری است. بر همین اساس که در شکل ۱ مشخص شده کل فرآیند آزمون به نمایش درآمده به این صورت که المان توصیف توسط متخصص دامنه نوشته می‌شود و آزمون کننده از این سند به عنوان تولید نمونه آزمون و ارزیابی توسط شبکه پتری استفاده می‌کند.



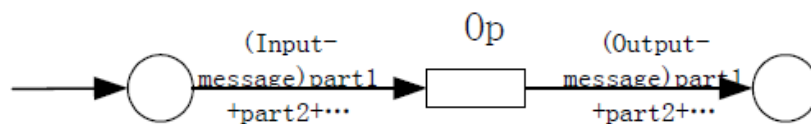
شکل ۱: فرآیند کلی راه حل ارائه شده

در این مقاله در دو بخش آزمون وب سرویس‌ها را انجام خواهد شد:

- ۱- تولید نمونه آزمون: در این قسمت با توجه به روش پیشنهاد شده در این فصل به تولید نمونه آزمون پرداخته خواهد شد.
- ۲- ارزیابی نمونه آزمون: در این قسمت با توجه به تولید نمونه آزمون به ارزیابی نمونه آزمون پرداخته تا کارایی آن سنجیده شود.

۲-۱- تبدیل *WSMO* به شبکه پتری

توابع موجود در وب سرویس‌ها برای تبدیل شامل دو سطح تجزیه و تحلیل هستند: پیغام و بخش. پیغام‌های ورودی مشخص کننده آراگومان‌های تابع هستند و پیغام‌های خروجی مشخص کننده نتایج مورد انتظار از آزمون هستند. پیغام می‌تواند شامل بخش‌ها باشد. یک بخش به عنوان یک پارامتر یک تابع نگاشت می‌شود. این نگاشت به صورت تجزیه ورودی و خروجی توابع که بصورت گذار و ورودی و خروجی به صورت کمانهای ورودی و خروجی به نمایش داده می‌شود.



شکل ۲: ورودی و خروجی در یک شبکه پتری

همانطور که در شکل ۲ مشخص است تمام بخشهای یک ورودی و خروجی توسط یک مکان همراه با توکن نمایش داده می‌شود. برای ورودی و خروجی های ترکیبی توابع از نشان + بر روی کمان ها استفاده می‌شود. از کمان‌ها برای برقراری ارتباط با مکان‌ها و توابع استفاده می‌شود که شامل پیغام‌های ورودی و خروجی است استفاده می‌شود.

با توجه به مطابق بودن UIO (Unique Input OutPut) با ماشین حالت متناهی برای تشخیص خطا و از کار افتادگی سیستم از ورودی خروجی های یکتا استفاده می‌شود. بدین منظور UIO را در این مقاله مطابق با شبکه پتری اعمال می‌کنیم به همین دلیل UIO مطابق با HPN ($High Petri Net$) باشد تعاریف زیر را بیان می‌کنیم. [۸]

HPN شامل مجموعه‌ای محدود از مکان‌ها است که قبل از توابع نمایش داده می‌شود $P = \{p_1, p_2, \dots, p_n\}$ ، همچنین مجموعه‌ای محدود از گذارها نیز برای نمایش توابع استفاده می‌شود $T = \{t_1, t_2, \dots, t_m\}$ ، مجموعه ورودی به صورت $I = \{i_1, i_2, \dots, i_n\}$ و مجموعه خروجی به صورت $O = \{o_1, o_2, \dots, o_n\}$ نمایش داده می‌شود. HPN را می‌توان به صورت زیر تعریف کرد:

$$P \times I \rightarrow P \times O \quad (1)$$

در این نگاشت HPN از ۳ تاپل تشکیل شده $PH = \{P, T, A\}$ نشان دهنده مسند یک HPN نشان دهنده گذارها یا توابع موجود در وب سرویس هستند و A نشاندهنده ورودی و خروجی‌ها هستند.

اینترفیس I در المان توصیف هدف به صورت ۳ تاپل نمایش داده می‌شود $I = \{C(O), T, \sum\}$ ، $C(O)$ در اینجا مجموعه ایی از حالات در آنتولوژی است این مجموعه می‌تواند شامل n عضو متناهی باشد. T مجموعه ایی از قوانین گذار که شامل m عضو متناهی است. \sum آرگومان های توابع است که با حالات در ارتباط هستند. همانطور که مشاهده می‌شود می‌توان تناظر یک به یکی بین تاپل‌های HPN و المان مشخصه هدف برقرار کرد:

$$C(O) \approx P \quad (2)$$

$$T \approx T \quad (3)$$

$$\sum \approx A \quad (4)$$

پس از نگاشت‌های فوق نوبت به ارائه الگوریتمی مبتنی بر UIO می‌رسد. در این الگوریتم به ازای هر تراکنش کوتاهترین ورودی و خروجی مطابق UIO در جایی که ti گذار شروع و tj گذاری است که بعد از ورودی و خروجی ساخته می‌شود. فرض کنید سیستم آزمون شده توسط یک ساختار میانی بیان می‌شود. این ساختار میانی در واقع گرافیکست که از سیستم مدل شده توسط شبکه پتری بدست می‌آید.

$$G = \{T, A\} \quad (5)$$

در این گراف T توسط گره حالات که در شبکه پتری مشخص شد نگاشت می‌شود، در واقع حالات گذار در شبکه پتری به حالت گراف تبدیل می‌شود و A در واقع همان کمان‌هایی است که در شبکه پتری وجود داشته‌اند. گراف متقارن G^* برای تولید دنباله آزمون در نظر داشته باشید، این گراف توسط مسیر اوپلر تولید می‌شود، الگوریتم تولید آن به این صورت است:

۱- محاسبه کوتاه‌ترین دنباله UIO تطابق داده شده با هر حالت در گراف G

۲- اگر مسیر اوپلری وجود داشته باشد که شروع آن از ti و پایان آن tj باشد، بنابراین دنباله آزمون $UIO(tj)$ است.

۳- برای تبدیل گراف G به گراف متقارن G^* و تبدیل شدن به مسیر اوپلری در حالی که $A^* \subset A$ کمان‌ها را به همان مداری که در G هست اضافه می‌کنیم

۴- تولید مسیر اوپلر از جایی است که گره آغازین مسیر اوپلر شروع می‌شود و دنباله آزمون انجام می‌شود.

همانطور که ملاحظه می‌شود الگوریتم مشخص شده از نمودار HPN به نمودار گراف تبدیل می‌شود، این کار باعث منجر به تولید نمونه آزمون از طریق گراف تبدیل شده می‌شود. برای اینکه آزمون مورد نظر پوشش کافی داشته باشد و مورد پذیرش باشد از ویژگی $trap$ استفاده می‌شود این ویژگی همانطور که بیان شد برای مورد قبول بودن پوشش آزمون بیان می‌شود. بدین منظور از دو معیار پوشش استفاده می‌شود:

۱- شرایط مرزی

۲- پوشش پذیری شرط اصلاح شده^۲

۲-۳- تولید Trap

شرایط مرزی رفتار یک برنامه را بر اساس مرزهای یک متغیر آزمون می‌کند. این روش برای تولید مجموعه ایی از موارد آزمون برای آزمون‌های مرزی به طور موثر و کارآمد مورد استفاده قرار می‌گیرد. بدین منظور مراحل محاسبه شرایط مرزی به صورت زیر بیان می‌شود:

۱- با استفاده المان توصیف هدف می‌توان متغیرها را شناسایی کرد و شرایط مرزی آنرا بدست آورد.

۲- با اختصاص شرایط مرزی متفاوت و مختلف و نزدیک‌تر شدن به مرزهای این متغیر محدود آزمون مرزی محاسبه می‌شود.

۳- محاسبه ویژگی *trap*

پوشش پذیری شرط اصلاح شده آزمون بر اساس معیار پوششی مبتنی بر منطق است که شرایط را به صورت تکی و کامل آزمون می‌کند. شرایط تولید این *trap* بر اساس مراحل زیر است:

۱- تعیین شرایط منطقی موجود در مشخصه

۲- محاسبه پوشش پذیری شرط اصلاح شده بر اساس شرایط منطقی موجود در آزمون.

۳- تولید ویژگی *trap*

۲-۴- ارزیابی و صحت نمونه آزمون تولید شده بر اساس Mutation

بعد از ایجاد نمونه‌های آزمون از طریق شبکه پتری، قدم بعدی ارزیابی کیفی این نمونه‌های آزمون است که آیا با توجه به این نمونه آزمون، خطاهایی که در برنامه است کشف شده است یا خیر. بدین منظور از روش *mutation* استفاده می‌شود. در این پروژه ما با این روش به اندازه‌گیری موثر بودن نمونه آزمون می‌پردازیم، به این صورت که با تزریق خطا به برنامه اصلی و مقایسه آن با خروجی برنامه اصلی می‌توان تعداد خطاها را کشف کرد.[۹]

به دو دلیل از این روش استفاده شده است

۱- این روش مستقل از روش ساخت نمونه آزمون است.

۲- با این روش می‌توان تعداد خطاهایی که منجر به از کار افتادگی سیستم می‌شود را اندازه‌گیری کرد یا به اثر بخشی روش تولید آزمون برای پیدا کردن خطاهای سیستم پرداخت.

روش کار به این صورت است که اگر یک وب سرویس معنایی S داشته باشیم همراه با مشخصه ϕ باشد و با پیاده‌سازی Δ و همراه

مجموعه آزمون T مشخص شده باشد، وب سرویس معنایی S دارای مجموعه از نسخه‌های *Mutant* است:

$$M(S) = \{S_1, S_2, \dots, S_n\}$$

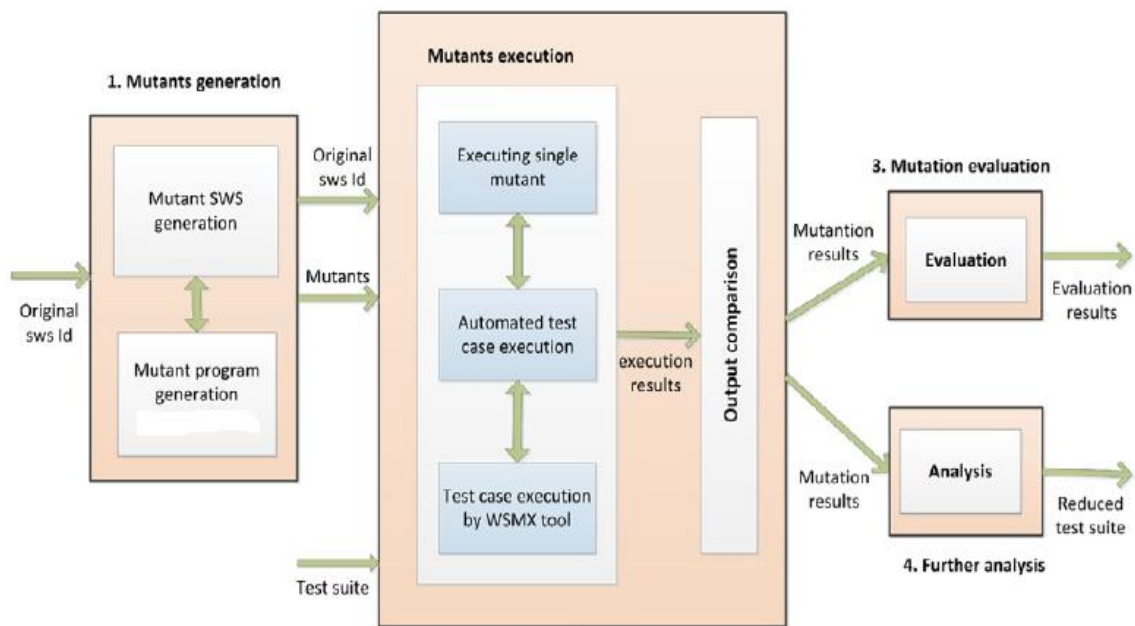
که توسط فرآیند تولید *Mutant* تولید شده است.

مجموعه آزمون T بر روی وب سرویس‌های اصلی S و وب سرویس *Mutant* اجرا می‌شود و نتایج یا خروجی این آزمون‌ها ذخیره

می‌شود. نتایج خروجی *Mutant* دسته بندی شده و آماده برای ارزیابی نتایج می‌شود در این ارزیابی، کارایی در غالب تعداد خطاها نمایش داده می‌شود و با افزایش کیفیت نمونه آزمون تعداد نمونه آزمون کاهش پیدا می‌کند.

در شکل ۳ فرآیند ارزیابی نمونه آزمون با *Mutant* مشخص شده است.

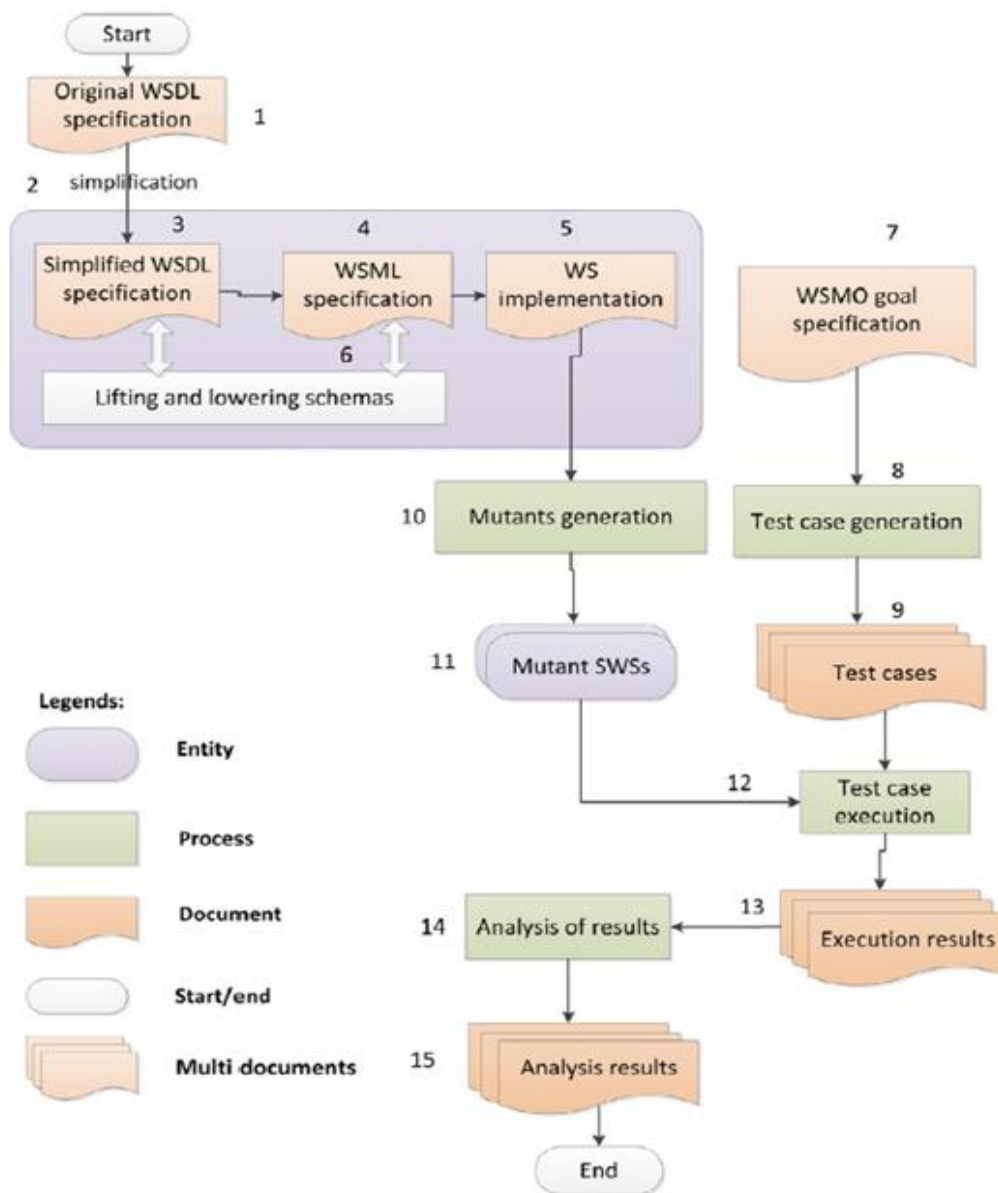
² Modified Condition/Decision Coverage(MDCCD)



شکل ۳: فرآیند ارزیابی نمونه آزمون‌های تولید شده با استفاده از روش *Mutant*

۳- مطالعه موردی *Practice Fusion*

در این مقاله از مستندات سایت www.practicefusion.com استفاده شده است، این شرکت پلتفرمی برای ارائه بهتر خدمات بین پزشکان و بیماران قرار می‌دهد. به منظور مطالعه موردی از این شرکت سرویس‌های دریافت اطلاعات از آزمایشگاه و رادیولوژی در نظر گرفته شده است. ارسال اطلاعات در این شرکت به صورت وب سرویس انجام می‌شود همچنین شرکت مذکور از استاندارد *HL7* نسخه ۲,۵,۱ استفاده کرده است تا بدین منظور اطلاعات به صورت یک استاندارد مشترک بین تمام سیستم‌هایی که از این سرویس استفاده می‌کند در نظر گرفته شده است.



شکل ۴: مراحل تولید و ارزیابی نمونه آزمون مطالعه موردی

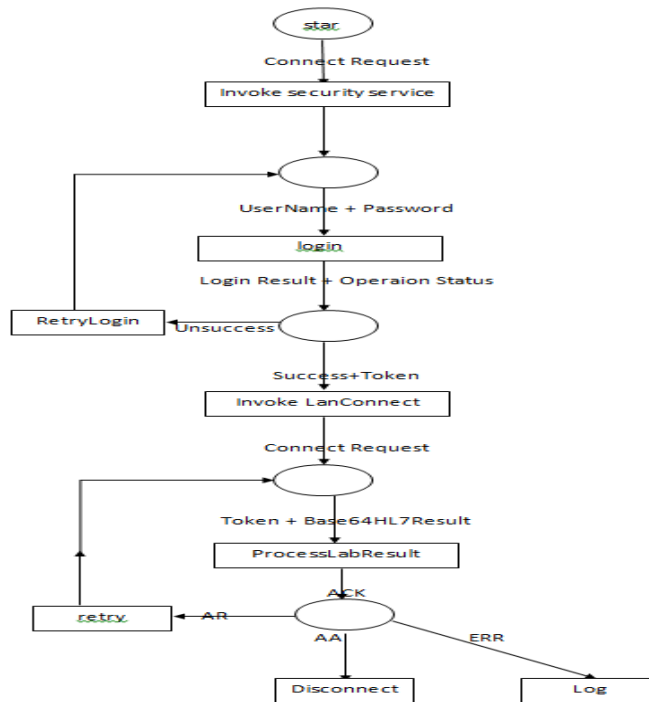
فرایند موجود برای این مطالعه موردی بدین صورت است که در شکل ۴ مشاهده می‌شود. همانطور که در شکل ۴ مشخص است *WSDL* شرکت *Practice Fusion* را بدست آورده سپس با توجه با راهنمایی‌های داده شده به *WSML* تبدیل می‌شود در قدم هفتم با توجه به راهنمایی‌های سایت شرکت *Practice Fusion* و نیازمندی‌ها و شرایطی که همانند المان توصیف هدف بیان شده به تولید نمونه آزمون با استفاده از روش مطرح شده می‌پردازیم. در قدم دهم به یافتن و تغییر وب سرویس اصلی برای ایجاد *Mutant* پرداخته شده تا نمونه آزمون مشخص شده مورد ارزیابی قرار گیرد پس از ایجاد *Mutant* به اندازه‌گیری کارایی و اثربخشی نمونه آزمون تولید شده خواهیم پرداخت.

۳-۱- تولید نمونه آزمون

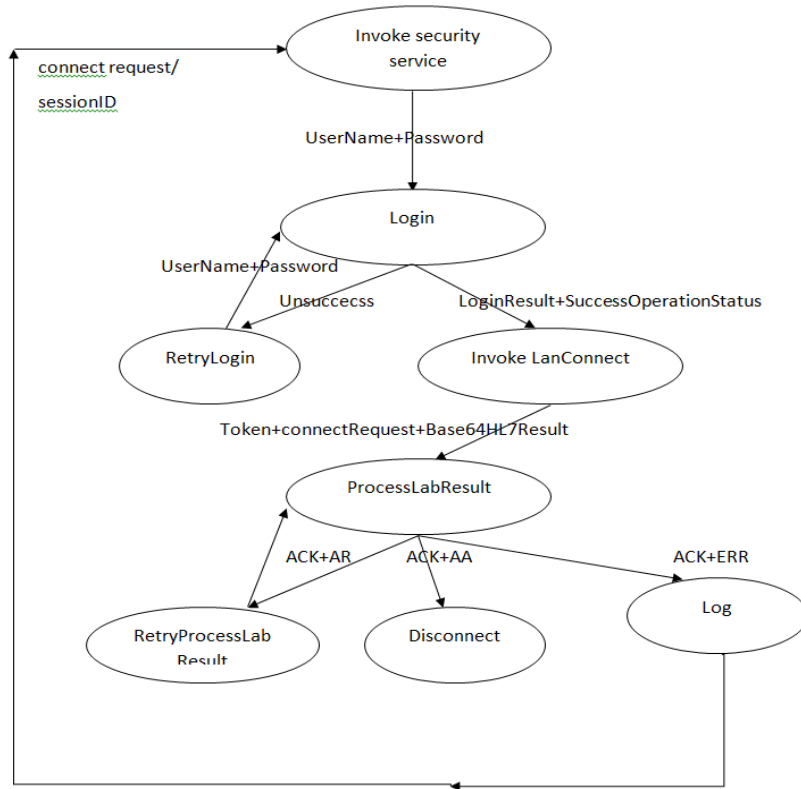
در اینجا به رسم شبکه پتری سیستم ارسال اطلاعات نتایج آزمایشگاه و رادیولوژی می‌رسیم. همانطور که در بخش قبل درباره روش پیشنهادی برای رسم شبکه پتری گفته شد و با توجه به *WSDL* وب سرویس‌های مذکور، توابع مشخص شده در این سند به عنوان گذرگاه مشخص می‌شود و آرک‌ها به عنوان ورودی و خروجی و دایره‌ها به عنوان حالات این شبکه در نظر گرفته شده است این شبکه شامل ترکیب دو وب سرویس است که در قدم اول اطلاعات هویتی اعتبار سنجی می‌شود و در قدم بعدی به نتایج اطلاعات موجود در آزمایشگاه یا

رادیولوژی وارد سیستم *EHR* می‌شود. با توجه به اینکه وب سرویس‌ها ارسال اطلاعات آزمایشگاه و رادیولوژی کاملاً شبیه هم هستند، به همین دلیل تنها اختلاف این وب سرویس در نمودار شبکه پتری *ProcessLabResult* است که به جای آن *SubmitResultsRequest* قرار می‌گیرد. در قدم بعدی برای تولید نمونه آزمون شبکه پتری با توجه به روش گفته شده به گراف تبدیل می‌شود.

همانطور که گفته شد برای تبدیل شدن به گراف تنها گذرگاه‌ها و ورودی و خروجی *UIO* در نظر گرفته می‌شود. پس از بوجود آمدن گراف که در شکل ۶ مشخص شده نمونه آزمون تولید می‌شود این نمونه آزمون بر اساس راه‌هایی است که باید از نقطه شروع تا نقطه پایان طی شود.



شکل ۵: نمودار شبکه پتری مطالعه موردی



شکل ۶: نمودار گراف تولید شده مطالعه موردی

در جدول ۱ حالت‌ها و ورودی و خروجی‌های گراف فوق را به اختصار نامگذاری می‌کنیم تا تولید نمونه آسانتر گردد.

State		Arcs	
Invoke security service	t1	Connect request/sessionID	a1
Login	t2	UserName+Password	a2
RetryLogin	t3	UserName+Password (from node RetryLogin)	a3
Invoke LanConnect	t4	Unsucccass	a4
ProcessLabResult	t5	LoginResult+SuccessOperationStatus	a5
RetryProcessLab Result	t6	Token+connectRequest+Base64HL7Result	a6
Log	t7	Token+connectRequest+Base64HL7Result (from node RetryProcessLab Result)	a7
Disconnect	t8	ACK+AR	a8
SubmitResultsRequest	t9	ACK+ERR	a9
RetrySubmitResultsRequest	t10	ACK+AA	a10

جدول ۱: اختصارسازی کمان‌ها و گذارها

با توجه به جدول فوق شروع به تولید نمونه آزمون می‌کنیم به عنوان مثال $\{a2, a5, a6, a9\}$ یکی از این نمونه‌های آزمون بر اساس گرافی است که تولید شده است.

برای تولید نمونه آزمون دو نوع *trap* را در نظر گرفته شده است، پوشش‌پذیری شرط اصلاح شده و شرایط مرزی، به این صورت که توابعی که در وب سرویس استفاده می‌شود با توجه به محدودیت‌ها و شرایط گفته شده در امان توصیف هدف مد نظر قرار داده شده است. به عنوان مثال گفته شده بعد از تولید *token* فرستنده ۲ ساعت مهلت استفاده از توکن را برای ارسال اطلاعات به سمت سیستم *EHR* دارد بنابراین یکی از شرایط مرزی برای تولید *trap* این است که در صورت استفاده این توکن دقیقاً در ساعت ۲ رفتار سیستم چگونه خواهد بود.

در جدول ۲ تعداد *trap* ها بر اساس توابع و شرایطی که در وب سرویس اعلام شده است را بیان می‌کنیم.

شماره ردیف	نام تابع	پوشش پذیری شرط اصلاح شده	شرایط مرزی
1	Login	2	0
2	RetyLogin	3	0
3	Invoke LanConnect	1	0
4	ProcessLabResult	2	12
5	RetryProcessLabResult	3	12
6	Log	2	0
7	Disconnect	2	0
8	SubmitResultsRequest	2	12
9	RetrySubmitResultsRequest	3	12

جدول ۲: تعداد *trap* طبق پوشش پذیری شرط اصلاح شده و شرایط مرزی

۲-۳- ارزیابی نمونه آزمون‌های تولید شده

برای ارزیابی نمونه آزمون‌های تولید شده نسخه اصلی برنامه را تغییر و خروجی برنامه تغییر داده شده را به خروجی برنامه اصلی مورد مقایسه قرار داده می‌شود در صورتی که خروجی این دو یکی باشد آزمون تولید شده موثر بوده و به اصطلاح خطا کشته شده در صورتی که خروجی یکی نباشد آزمون تولید شده موثر واقع نشده است.

همانطور که در جدول ۳ مشخص شده است توابع موجود در برنامه همراه با تعداد *mutant* ها و موارد آزمون نام برده شده است بعد از تزریق داده و مقایسه خروجی ها با یکدیگر تعداد *mutant* ها شناسایی شده مشخص شده است، در این بین نسبت یا امتیاز تعداد *mutant* های شناسایی شده به کل تعداد *mutant* ها به درصد در ستون آخر قرار گرفته است تا کارایی و اثر بخشی موارد آزمون مشخص شود.

نام تابع	تعداد مورد آزمون	تعداد <i>mutant</i>	تعداد <i>mutant</i> های شناسایی شده	درصد <i>mutant</i> های شناسایی شده
Login	2	3	3	100
RetyLogin	3	5	3	60
Invoke LanConnect	1	1	1	100
ProcessLabResult	14	54	41	75.6
RetryProcessLabResult	15	56	43	76.7
SubmitResultsRequest	14	54	41	75.6
RetrySubmitResultsRequest	15	56	43	76.7
Log	2	3	2	66.6
Disconnect	2	3	2	66.6

جدول ۳: امتیاز داده شده *Mutant* به ازای توابع موجود در مطالعه موردی

در این بین بعضی از موارد آزمون تعداد *mutant* های بیشتری را شناسایی کرده و به اصطلاح می‌کشد به همین دلیل بعضی از موارد آزمون اثربخش‌تر هستند و بعضی از موارد آزمون هیچ *mutant* را شناسایی نمی‌کند و اثر بخشی ندارد و می‌توان این نمونه آزمون ها را نادیده گرفت و تعداد نمونه آزمون ها را کاهش داد.

نام تابع	تعداد کل مورد آزمون	تعداد مورد آزمونی که می‌تواند نادیده گرفته شود	درصد کاهش
Login	2	0	-
RetyLogin	3	1	33.3
Invoke LanConnect	1	0	-
ProcessLabResult	14	10	71.4
RetryProcessLabResult	15	10	66.6

71.4	10	14	SubmitResultsRequest
66.6	10	15	RetrySubmitResultsRequest
3	-	2	Log
-	-	2	Disconnect

جدول ۴: درصد کاهش نمونه آزمونهای تولید شده در مطالعه موردی

همانطور که در جدول ۴ مشاهده می‌شود تعداد مورد آزمون در هر کدام از توابع مورد بررسی قرار گرفته است و آزمون‌هایی که هیچ *mutant* در آن شناسایی نشده مشخص شده است و درصد تعداد آزمونهای کارا در این مورد مشخص شده است به عنوان مثال برای تابع *ProcessLabResult* تعداد ۱۴ مورد آزمون بکار رفته است که از این بین تعداد ۴ مورد آزمون همه *mutant* ها را شناسایی کرده است، بنابراین می‌توان ۱۰ مورد آزمون را ندیده گرفت و تنها موارد آزمونی که موثر هستند مورد استفاده قرار گیرد، ستون آخر درصد کاهش مورد آزمون از کل آزمون را نمایش می‌دهد.

نتیجه‌گیری

در این مقاله به بررسی ساختار و المان‌های *WSMO* پرداختیم و برای تولید نمونه آزمون بر اساس پتری نت، *WSMO* را با به شبکه پتری تبدیل کردیم. همانطور که مطرح شد *WSMO* المانهای مختلفی برای نگاشت دارد که مهمترین المان آن توصیف هدف است. در این المان نیازهای کاربران نهایی مشخص شده و به عنوان مرجع نیازهای کاربران مطرح می‌شود. برای تبدیل کردن *WSMO* به شبکه‌های پتری چالش‌های می‌توان مطرح است به این صورت که در صورت تبدیل، روش تبدیل می‌تواند خود یک چالش باشد و در حین تبدیل مواردی از سیستم واقعی به خوبی نگاشت نشده و اثر خود را در آزمون باقی بگذارد به همین دلیل برای تولید نمونه آزمون روش *Mution* مطرح شد تا کارایی و اثربخشی نمونه آزمون مورد ارزیابی قرار گیرد. پس از آن به بررسی و ارزیابی نمونه آزمون پرداخته و این نمونه‌های آزمون را امتیاز بندی شد که متوسط امتیاز تابع‌های بررسی شده بر اساس موارد آزمون ۷۷,۵۳ است که امتیاز مطلوبی برای یک سیستم سلامت محسوب می‌شود، به عبارت دیگر مورد آزمون‌های ارائه شده بالای ۳/۴ سیستم را از جهات مختلف مورد آزمون قرار می‌دهد. همچنین این نکته نیز ضروری است که برای بهینه‌سازی و صرفه‌جویی در وقت و هزینه می‌توان بسیاری از موارد آزمون را نادیده گرفت زیرا موارد آزمون دیگر پوشش دهنده موارد آزمون نادیده گرفته شده هستند و به همین دلیل با بررسی این موارد آزمون و نتیجه این شد که با معدل ۶۱,۸۶ درصد می‌توان در هزینه و زمان آزمون طبق جدول ۴ صرفه‌جویی کرد.

منابع و مراجع

- [1] A. Winter, et al, Strategic information management in hospitals. Health Information Systems. Springer London, 2010, pp. 237-282.
- [2] Nie, Hongchao, et al. "From healthcare messaging standard to semantic web service description: generating WSMO annotation from HL7 with Mapping-Based approach." Services Computing (SCC), 2013 IEEE International Conference on. IEEE, 2013.
- [3] A. Dogac, G. B. Laleci, S. Kirbas, Y. Kabak, S. S. Sinir, A. Yildiz, et al.: "Artemis: Deploying semantically enriched Web services in the healthcare domain", Information Systems, vol. 31, 2006, pp. 321-339
- [4] Murata T. Petri nets: properties, analysis and applications. In Proceedings of the IEEE 1989; 77(4):541-580.
- [5] J. Domingue , D. Roman , M. Stollberg , Web service modeling ontology (WSMO)-an ontology for semantic web services, in: Proceedings of the W3C Workshop on Frameworks for Semantics in Web Services, 2005, pp. 9-10 .
- [6] Jokhio, M. Shaban, et al. "Goal-based testing of semantic web services." Information and Software Technology 83 (2017): 1-13.
- [7] Jokhio, Muhammad Shaban. "Goal-based testing of semantic web services." Proceedings of the 2009 IEEE/ACM International Conference on Automated Software Engineering. IEEE Computer Society, 2009.
- [8] Dong, Wen-Li, and Y. U. Hang. "Web service testing method based on fault-coverage." Enterprise Distributed Object Computing Conference Workshops, 2006. EDOCW'06. 10th IEEE International. IEEE, 2006.
- [9] M.S. Jokhio , G. Dobbie , T. Hu , J. Sun , Automated mutation-based test case evaluation for semantic web services, in: Proceedings of the 2014 23rd Australian Software Engineering Conference, in: ASWEC '14, IEEE Computer Society, Washington, DC, USA, 2014, pp. 210-219 .