روشی برای استخراج مدل تغییرپذیری خط تولید نرم‌افزارهای سرویس‌گرا

صدیقه خوشنویس[1]، فریدون شمس علیئی[2]

۱کارشناسی ارشد مهندسی کامپیوتر (نرم افزار)، دانشکده مهندسی برق و کامپیوتر، دانشگاه شهید بهشتی، s_khoshnevis@sbu.ac.ir

۲دانشیار دانشکده مهندسی برق و کامپیوتر، دانشگاه شهید بهشتی

تار یخ دریافت: ۹۲/۹/۱۷ تاریخ پذیرش: ۹۳/۷/۵

چکیده

خط تولید نرم‌افزار عبارت از مجموعه‌ای از سیستم‌های نرم‌افزاری است که دارای شباهت بوده و با هدف برنامه‌ریزی فعالانه و افزایش قابلیت استفاده مجدد طراحی شده و از مجموعه‌ای از دارایی‌های پایه قابل استفاده مجدد تولید می‌شوند. در خط تولید سرویس‌گرا که در آن دارایی‌های پایه نرم‌افزاری را سرویس‌های نرم‌افزاری تشکیل می‌دهند، همراستایی سرویس‌ها با حرفه به عنوان یکی از اهداف سرویس‌گرایی می‌بایست مورد توجه قرار گیرد. شناسایی سرویس‌های مناسب و تغییرپذیری آنها به صورتی که همراستا با حرفه باشند، تنها با استفاده از مدل ویژگی امکان‌پذیر نیست. در نتیجه، یکی از چالش‌های عمده در خط تولید سرویس‌گرا، یافتن سرویس‌های همراستا با حرفه و تغییرپذیری آنها است. نوآوری این پژوهش، شناسایی سرویس‌های مناسب و تعیین تغییرپذیری آنها متناسب با فرایندهای حرفه از طریق یک الگوریتم تکاملی برای افراز گراف مدل خانواده فرایندهای حرفه در خط تولید سرویس‌گرا است. سرویس‌های به دست آمده از اصول طراحی تبعیت کرده و در قالب مدل تغییرپذیری سطح سرویس سازماندهی و توصیف می‌شوند. این سرویس‌ها از نظر معیارهای طراحی سرویس‌گرایی و خط تولید و نیز دارا بودن شباهت ساختاری با طراحی معماران قابل قبول هستند.

کلیدواژه

خط تولید نرم‌افزار، خط تولید سرویس‌گرا، مدیریت تغییرپذیری، معماری سرویس‌گرا، فرایند حرفه.

مقدمه

خط تولید سرویس‌گرا، خط تولید پویایی است که محصولات آن بر اساس سرویس‌ها و معماری سرویس‌گرا ساخته می‌شود[۱]و به عنوان یک خط تولید نرم‌افزار، با فراهم کردن امکان استفاده مجدد از دارایی‌های پایه مشترک و متنوع که سرویس‌های نرم‌افزاری هستند با برنامه‌ریزی فعالانه در کاهش هزینه و زمان تولید نرم-افزارهایی که به یکدیگر شباهت‌های زیادی دارند نقش مؤثری دارد. اگرچه مفهوم سرویس شباهت‌هایی با مؤلفه‌های نرم‌افزاری دارد، اما تفاوت‌های مهمی که بین این دو مفهوم وجود دارد مانع از آن می‌شود که بتوانیم از روش‌های موجود مهندسی خط تولید که برای ساختن نرم‌افزارهای مبتنی بر مؤلفه به کار می‌رود، به منظور ایجاد نرم‌افزارهای سرویس‌گرا نیز استفاده کنیم. یکی از این تفاوت های مهم، لزوم همراستایی سرویس‌ها با حرفه[۲]به عنوان یکی از اصول طراحی سرویس‌گرا است. در نرم‌افزارهای سرویس‌گرا، فرایندهای حرفه مبنای طراحی سرویس‌ها هستند و سرویس‌های شناسایی شده به فعالیت‌های درون فرایندهای حرفه مقید می‌شوند.

مانند همه خطوط تولید نرم‌افزار، مسأله مدیریت تغییرپذیری (مدل‌سازی و نگهداری تغییرپذیری و و ایجاد محصول بر اساس آن) یکی از فعالیت‌های کلیدی برای موفقیت خط تولید سرویس‌گرا است. در ادبیات حوزه مهندسی خط تولید سرویس‌گرا، نبود روشی برای مدیریت تغییرپذیری مبتنی بر حرفه کاملاً محسوس است. در این تحقیق، تمرکز ما بر جنبه مدل‌سازی تغییرپذیری در خط تولید سرویس‌گرا است و هدف زیر را دنبال می‌کنیم: «*شناسایی و توصیف سرویس‌ها و تغییرپذیری آنها به-گونه‌ای‌که مدل تغییرپذیری سطح سرویس با فرایندهای حرفه همراستا باشد.*»

روش‌های موجود توصیف تغییرپذیری در خط تولید سرویس‌گرا، اگرچه تأکید زیادی بر نقش مهم فرایندهای حرفه دارند [۳][۴][۵][۶][۷]، اما هیچ‌یک اشاره‌ای به نحوه همراستا کردن مدل تغییرپذیری و عناصر آن با فرایندهای حرفه ندارند. به‌ویژه این‌که در صورتی که ابعاد خط تولید بزرگ باشد، شناسایی سرویس‌های مناسب که اصول طراحی سرویس‌ها را پشتیبانی کنند و درجه مناسبی از اشتراک و تنوع را ایجاد نمایند، به صورت

دستی دشوار و به تجربه معمار وابسته است. در نتیجه، دقت پایین و لزوم صرف زمان و هزینه زیاد از معایب روش‌های موجود است و ضرورت وجود یک روش سامانمند[1] با هدف فوق به چشم می‌خورد. بر این اساس، سؤال تحقیق آن است که «*آیا می‌توان مدل تغییرپذیری خط تولید سرویس‌گرا را بر مبنای فرایندهای حرفه به دست آورد به‌طوری‌که سرویس‌های شناسایی شده منطبق بر اصول طراحی سرویس‌ها و طراحی خط تولید باشد؟*»

برای پاسخ به این سؤال، روشی بر پایه الگوریتم ژنتیک NSGA-II ارائه شد که ورودی‌های آن «مدل خانواده فرایندهای حرفه» (BPFM[2]) و نوعی ماتریس CRUD[3] داده- فرایند خاص و خروجی آن مدل تغییرپذیری سرویس (SVM[4]) می‌باشد. همچنین الگوریتم‌هایی برای تعیین واسط سرویس‌ها و تغییرپذیری آنها در این تحقیق پیشنهاد شده است.

مقاله حاضر به صورت زیر سازمان‌دهی شده است: کارهای پیشین و سوابق تحقیق ارائه می‌شود. سپس کلیات و جزئیات راه‌حل را بیان خواهیم کرد. و در پایان به ارزیابی روش پیشنهادی و نتیجه‌گیری از تحقیق می‌پردازیم.

کارهای پیشین

خط تولید سرویس‌گرا به عنوان یک رویکرد جدید، در چند سال اخیر مورد توجه پژوهشگران قرار گرفته است. کارهای ارائه شده در[۱]،[۸]،[۹]،[۱۰]و[۱۱]از جمله این موارد هستند. در هیچ یک از پژوهش‌های مطرح شده، به نحوه شناسایی مدل تغییرپذیری سرویس با توجه به فرایندهای حرفه اشاره نشده است.

در زمینه مدل‌سازی تغییرپذیری در خط تولید سرویس‌گرا، کارهای متنوعی انجام شده است. تعدادی از این تحقیقات برای تعیین سرویس‌ها و تغییرپذیری آنها و نوع ارتباط سرویس‌ها با فرایندهای حرفه هیچ راهکار سامانمندی ارائه نداده‌اند و تنها به ارائه نماد یا راهکار مدل‌سازی بسنده کرده‌اند؛ از جمله، در[۱۲]و[۱۳]که مستقل از فرایندهای حرفه به تعریف نقاط تغییر برای ترکیب سرویس‌ها می‌پردازند.

در برابر این پژوهش‌ها، مواردی هم هستند که بر لزوم نگاشت سرویس‌ها با فرایندهای حرفه تأکید دارند. از جمله این پژوهش‌ها می‌توان به کار «سان» و همکارانش در[۴]و[۵]و مدل «ابومطر»[۳]اشاره کرد؛اما نحوه شناسایی سرویس‌های مناسب و تعیین سامانمند تغییرپذیری سرویس‌ها در این روش‌ها وجود ندارد و این فعالیت‌ها کاملاً وابسته به تجربه معمار هستند. شاید مهمترین تحقیق در این دسته از پژوهش‌ها، کار تیم «مون» و «پارک» باشد که در[۶]و[۷]با ارائه مفهوم مدل خانواده فرایندهای حرفه (BPFM)، تغییرپذیری فرایندهای حرفه را نمایش داده و تأکید می‌کنند که مدل تغییرپذیری سرویس باید به مدل فوق قابل ردیابی باشد. اما به نحوه این ردیابی و روش شناسایی سرویس‌های مناسب پرداخته نشده است.

در[۱۴]مطالعه‌ای بر روی روش‌های شناسایی سرویس‌ها به طور کلی و امکان کاربرد آنها در خط تولید نرم‌افزار انجام شده است. ۳۰٪ روش‌های مطالعه شده در این بررسی، مبتنی بر فرایندهای حرفه است که بالاترین درصد را به خود اختصاص داده‌اند؛ اما هیچ‌یک از این روش‌ها مرتبط با خط تولید نیستند. روش‌هایی که در آنها سرویس‌ها به همراه تغییرپذیری‌شان شناسایی می‌شوند، محدود به[۱۵][۱۶][۱۷][۱۸]و[۱] هستند. اما از میان آنها هیچ‌یک مبتنی بر فرایندهای حرفه نیستند و اغلب ([۱۵][۱۷][۱۸][۱]) از طریق گروه‌بندی و یا تحلیل ویژگی‌ها به شناسایی سرویس‌ها می‌پردازند. اما نحوه این گروه‌بندی و مبنای ساختاری یا معنایی آن در این روش‌ها مبهم است و علاوه بر در نظر نگرفتن نقش فرایندهای حرفه، واسط دقیقی هم برای سرویس‌ها تعیین نمی‌شود.

خارج از بحث خط تولید سرویس‌گرا، در خط تولید عمومی، موضوع تعیین سامانمند معماری خط تولید بر اساس نیازمندی‌ها مورد توجه پژوهگشران قرار دارد و کارهایی از جمله[۱۹]در حال انجام است. در[۱۹]با استفاده از الگوریتم‌های تکاملی و بر اساس مدل ویژگی، عناصر و روابط معماری خط تولید شناسایی می‌شود.

همچنین خارج از حیطه خط تولید نرم‌افزار، برای شناسایی سرویس‌ها روش‌های متنوعی ارائه شده که در دو دسته کلی «تجویزی» و «نیمه‌خودکار» دسته‌بندی می‌شوند. از مهم‌ترین روش‌های تجویزی می‌توان به[۲۰]،[۲۱]،[۲۲]و[۲۳]اشاره کرد که اغلب مبتنی بر تجزیه فرایندهای حرفه برای شناسایی سرویس‌ها هستند. از میان روش‌های نیمه‌خودکار، می‌توان به[۲۴]، [۲۵]و [۲۶]اشاره کرد که در آنها از الگوریتم‌های شهودی یا تکاملی برای دسته‌بندی فعالیت‌های فرایند استفاده شده است.

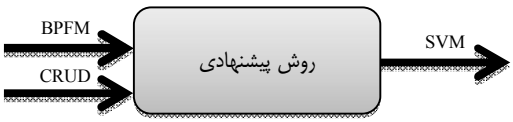
از این سوابق می‌توان این نتیجه را گرفت که هیچ‌یک از روش‌های موجود اشاره‌ای به نحوه همراستا کردن مدل تغییرپذیری خط تولید سرویس‌گرا و عناصر آن با فرایندهای حرفه ندارند؛ یافتن سرویس‌های مناسب بسیار متکی به تجربه معمار و مستعد خطا است. به ویژه وقتی اندازه خط تولید (تعداد نیازها، ویژگی‌ها، فرایندهای حرفه و...) بزرگ باشد، شناسایی سرویس‌های مناسب که اصول طراحی سرویس‌ها را پشتیبانی کنند و درجه مناسبی از اشتراک/تنوع را ایجاد نمایند به صورت دستی دشوار است. به

1. Systematic
2 .Business Process Family Model
3 .Create-Read-Update-Delete
4 .Service Variability Model

همین جهت، لزوم صرف زمان و هزینه زیاد و دقت پایین از معایب روش‌های موجود است.

کلیات روش پیشنهادی

روش پیشنهادی دارای دو ورودی و یک خروجی است (شکل ۱). ورودی‌ها عبارتند از مدل خانواده فرایندهای حرفه یا BPFM و ماتریس CRUD فعالیت حرفه- موجودیت حرفه و خروجی آن مدل تغییرپذیری سرویس (SVM) است.



شکل ۱. ورودی ها و خروجی روش مورد ارزیابی

مدل BPFM که پیشنهاد «پارک» و «مون»[۶][۷]است و جهت نمایش تغییرپذیری در سطح فرایندهای حرفه و نیز ترکیب سرویس‌ها برای خط تولیدهای سرویس‌گرا ارائه شده است، کاملترین مجموعه عناصر را جهت مدل‌سازی تغییرپذیری در سطوح نامبرده ارائه می‌کند. این مدل، مشابه یک نمودار فعالیت ساده در UML است که اطلاعات تغییرپذیری در آن گنجانده شده است. شکل ۲ یک BPFM را برای زیرسیستم ارسال نامه‌های اداری نشان می‌دهد.

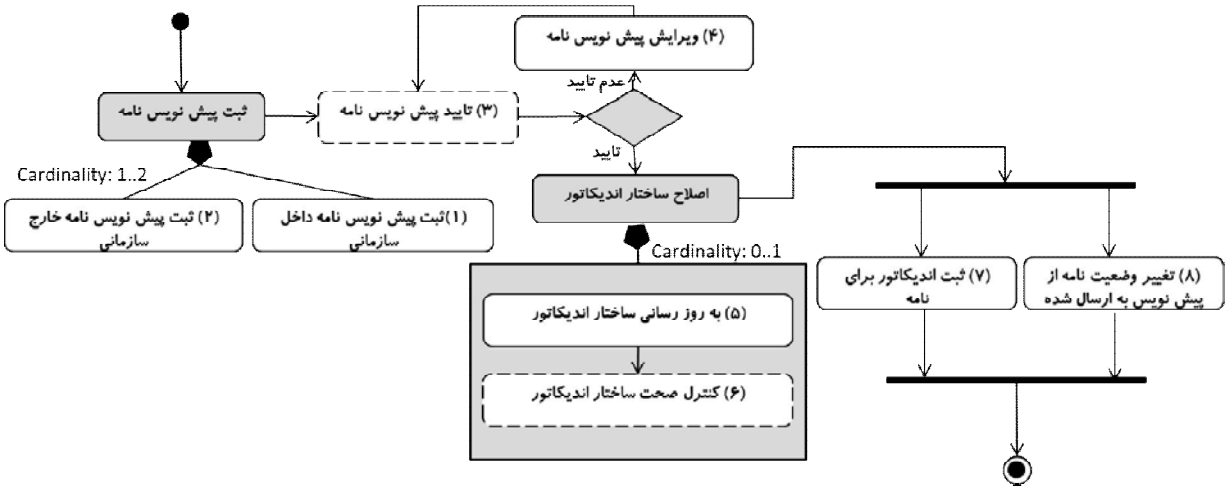
در BPFM سه نوع اطلاعات تغییرپذیری وجود دارد: اجباری یا اختیاری بودن هر فعالیت، نقاط تغییر ساده که برای نمایش گونه‌ها (بدون ناحیه گونه‌ها) به کار می‌روند و نقاط تغییرپذیری با ناحیه گونه‌ها که گونه‌ها را به صورت یک زیرنمودار نشان می‌دهد که بطور انتخابی جریان بین فعالیت‌های داخلی آن می‌تواند مشخص شود. بر این اساس تعریف ۱ را خواهیم داشت.

تعریف ۱ – مدل BPFM : یک شش‌تایی متشکل از مجموعه‌ای از فعالیت‌ها، نقاط تغییر، جریان میان فعالیت‌ها و نقاط تغییر، نقاط شروع و پایان و تعدادی رابطه بین فعالیت‌ها است.

$$BPFM = (A, VP, Flow, St, End, Rel)$$

برای جزئیات بیشتر به[۶]و[۷]مراجعه کنید.

تعریف ۲- ماتریس CRUD: ماتریسی است که سطرهای آن فعالیت‌های حرفه و ستون‌های آن موجودیت‌های حرفه هستند و در هر سلول این ماتریس در تقاطع هر فعالیت حرفه و موجودیت حرفه، نوع رابطه معنایی این دو معین می‌شود که یکی از کنش‌های «ایجاد»(C)، «خواندن»(R)، «به روز رسانی»(U) و «حذف»(D) توسط فعالیت حرفه روی موجودیت حرفه است



شکل ۲. نمودار BPFM برای زیرسیستم ارسال نامه های اداری

مزیت استفاده از ماتریس آن است که ساختار آن اجازه محاسبه و اندازه گیری معیارهای متنوع از جمله معیارهایی که برای شناسایی سرویس‌ها مورد نیاز است را فراهم می‌کند.

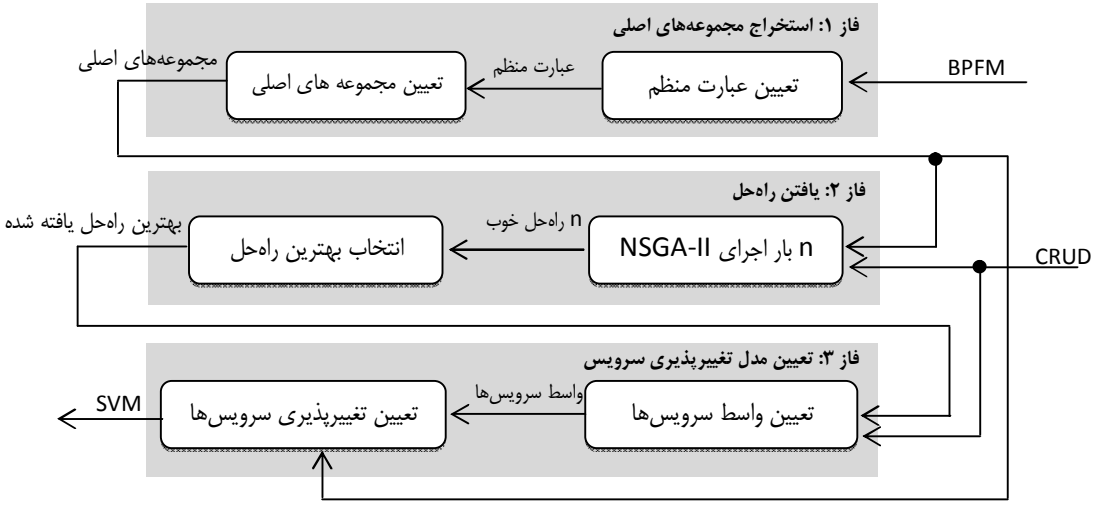
تنها خروجی روش، مدل تغییرپذیری سرویس یا SVM است. این مدل از مجموعه‌ای از سرویس‌ها به همراه واسط آنها و تغییرپذیری سرویس‌ها تشکیل شده است. واسط سرویس‌ها شامل عملیات داخل سرویس و پیام‌های ورودی و خروجی هر یک و نیز اجباری یا اختیاری بودن هریک از عملیات سرویس‌ها است. تغییرپذیری سرویس‌ها شامل تعیین اجباری یا اختیاری بودن هر سرویس و روابط جایگزینی (یای انحصاری)، یای غیرانحصاری و استلزام بین سرویس‌ها است.

تعریف ۳- SVM: این مدل مجموعه‌ای است از سرویس‌ها، واسط‌های آنها (عملیات و پیام‌ها)، اختیاری یا اجباری بودن هریک

و روابط تغییرپذیری بین آنها (شامل یای انحصاری، یای غیر انحصاری و استلزام).

شکل ۳ مراحل کلی روش پیشنهادی را نشان می‌دهد. جزئیات روش در بخش بعد بیان می‌شود.

روش پیشنهادی شامل سه فاز اصلی است. در فاز اول، بر اساس BPFM ورودی، عبارت منظمی ایجاد شده و سپس بر اساس آن مجموعه‌های اصلی یافته می‌شوند که به همراه ماتریس CRUD ورودی‌های مهم دو فاز بعدی هستند. در فاز دوم با استفاده از یک الگوریتم ژنتیک چند هدفه با امکان مرتب‌سازی غیرمغلوب و بهینه‌سازی «پارتو» به نام NSGA-II و بر اساس ماتریس CRUD گراف BPFM را افراز می‌کنیم تا راه‌حلی را بیابیم که چند معیار مهم طراحی را هم در بعد سرویس‌گرایی و هم در بعد تغییرپذیری برای همه مجموعه‌های اصلی ارضاء کند. با توجه به این که الگوریتم ژنتیک ممکن است در هر بار اجرا به یک راه‌حل بهینه محلی همگرا شود، با همان ورودی‌ها، الگوریتم را n بار تکرار می‌کنیم و پاسخ چیره (برتر) را برمی‌گزینیم و آن را با اجرای الگوریتم‌های تعیین واسط سرویس‌ها و تعیین تغییرپذیری سرویس‌ها در فاز سوم، به مدل SVM که خروجی اصلی است تبدیل می‌کنیم.



شکل۳. مراحل کلی روش پیشنهادی

جزئیات مراحل روش پیشنهادی

در این بخش هر مرحله از روش پیشنهادی با شرح بیشتر ارائه می‌شود.

تعیین عبارت منظم

پیش از هر کار، می‌بایست ساختار پیچیده نمودار BPFM ورودی را به ساختاری ساده‌تری و قابل محاسبه‌تر تبدیل کنیم. برای این منظور، از عبارت منظم[۳۳]استفاده می‌کنیم. عبارات منظم در مدل‌سازی انواع مسائل در مباحث مربوط به سیستم‌های نرم‌افزاری از جمله در خط تولید نرم‌افزار و تغییرپذیری در آن استفاده می‌شود[۳۴]و در اینجا به‌عنوان سازوکاری برای مدل کردن ساده‌تر مجموعه‌ای از پیکربندی‌ها که معادل با نمودار BPFM است به کار می‌رود. با این دیدگاه، هر پیکربندی، یک رشته و مجموعه این رشته‌ها یک زبان را تشکیل می‌دهند. الفبای این زبان با توجه به نیازهای خاص این تحقیق، تنها شامل فعالیت‌ها است (که باید صفات و روابط تغییرپذیری آنها توصیف شود) و مواردی مانند ترتیب اجرا، نقاط انشعاب و فعالیت‌های شروع و پایان مد نظر نمی‌باشد. با استفاده از عبارات منظم، توصیفی ساده‌تر و محاسبه‌پذیرتر از تغییرپذیری فرایندهای حرفه با حذف تعدادی از جزئیات غیرضروری به دست می‌آید.

تعریف ۴ – عبارت منظم (RE): در این تحقیق، عبارتی شامل تعدادی نشانه معادل با فعالیت‌های مدل BPFM است. اگر این مجموعه نشانه را الفبای زبان در نظر بگیریم و آن را با $\sum$ نشان دهیم، آن گاه:

- λ (عنصر پوچ) و هر عضو $\sum$ یک عبارت منظم هستند.
- اگر α و β دو عبارت منظم باشند، آنگاه $\alpha|\beta$ و $\alpha.\beta$ و (α) عبارات منظم هستند. عملگر (.) یا «اتصال» و عملگر (|) یا «یای انحصاری» دو عملگر جابجایی پذیر هستند. در این عبارات از پرانتز برای ایجاد اولویت استفاده می‌شود و اولویت، پس از پرانتز، با عملگر اتصال و سپس عملگر یای انحصاری است.

نشانه‌ها را برای سهولت، از شناسه‌های فعالیت‌ها انتخاب می‌کنیم. بر این اساس عبارت منظم برای BPFM مثال عبارت است از:

$$(1|2).(3|\lambda).4.(\lambda|5|5.6).(7.8)$$.

قدرت مدل‌سازی[5] عبارات منظم برای این صورت مسأله خاص، از نمودار BPFM بالاتر است. زیرا آنچه ما در مسأله به آن نیاز داریم صرفاً خود فعالیت‌ها و صفات و روابط تغییرپذیری آنهاست(مجموعه‌های A، VP و Rel). اگر نشان دهیم که این مجموعه عناصر را می‌توان با عبارات منظم پیشنهادی مدل کرد، از آنجا که به صورتی ساده‌تر و با اقلام اطلاعاتی کمتری می‌توان این عناصر را نشان داد، بنا بر این برای این مسأله، عبارات منظم از قدرت مدل‌سازی بالاتری بهره‌مند هستند. در زیر نشان می‌دهیم که چگونه مجموعه‌های فوق را با عبارات منظم پیشنهادی می‌توان مدل کرد:

(الف) هر نشانه، نمایانگر یک فعالیت است که به طور پیش فرض اجباری است. یک فعالیت اختیاری a را به صورت a|λ نشان می‌دهیم(مدل‌سازی مجموعه A). (ب) نقاط تغییر، نشان‌دهنده انتخاب میان چند «گونه» هستند؛ بنا بر این، فرضاً نقطه تغییری که سه گونه a، b، وc را ارائه می‌کند به صورت a|b|c مدل می‌شود. به همین صورت، حالت‌های پیچیده‌تر گونه‌ها را می‌توان با استفاده از پرانتز و عملگرهای اتصال و یای غیرانحصاری مدل کرد (مدل‌سازی مجموعه VP). (پ) روابط مجموعه Rel نیز شامل سه دسته رابطه است: رابطه Alt یا XOR بین a و b را با a|b، رابطه OR بین a و b را به صورت a|b|a.b و رابطه استلزام به صورت a→b را به صورت a.b|b|λ نشان می‌دهیم.

تعیین مجموعه‌های اصلی

در این تحقیق، هر مجموعه اصلی، نشان‌دهنده مجموعه‌ای از فعالیت‌ها است که می‌تواند در یک نمونه (پیکربندی) ایجاد شده از BPFM وجود داشته باشد.

تعریف ۵ – مجموعه‌های اصلی: به مجموعه فعالیت‌های موجود در هر یک از نمونه‌های قابل استخراج از عبارت منظم اطلاق می‌کنیم.

برای BPFM مثال، ۲۴ مجموعه اصلی وجود دارد که رشته‌های 1.4.7.8 و 2.3.4.5.6.7.8 نشان دهنده دو مورد از آنها هستند.

اجرای الگوریتم ژنتیک (NSGA-II)

مسأله حاضر یک مسأله بهینه‌سازی چندهدفه برای افراز گراف (BPFM) است. در واقع نیاز است مجموعه فعالیت‌های BPFM طوری افراز شود که حاصل آن مجموعه‌هایی از فعالیت‌ها (یعنی سرویس‌ها) را تشکیل دهد که مقادیر معیارهای خاصی (توابع هدف) را برای «همه مجموعه‌های اصلی» بیشینه می‌کنند. این معیارها شامل دو دسته هستند: معیارهای مربوط به اصول طراحی نرم‌افزارهای سرویس‌گرا (انسجام، اتصال، دانه‌بندی و همگرایی موجودیت‌های حرفه) و معیارهای مربوط به تغییرپذیری در خط تولید نرم‌افزار(درجه اشتراک).

از آنجا که توابع هدف این مسأله معادل یا حتی همسو با یکدیگر نیستند، آن را به صورت یک مسأله بهینه‌سازی چندهدفه در نظر می‌گیریم. از میان پنج معیار که بر اساس آنها توابع هدف تعیین شده‌اند، معیار درجه اشتراک بر خلاف سایر معیارها برای کل راه‌حل حساب می‌شود و نه برای هر سرویس منفرد در راه‌حل؛ در نتیجه حتماً باید به عنوان یک تابع هدف مجزا در نظر گرفته شود. از چهار معیار باقی‌مانده، ابتدا ارتباط معیار دانه‌بندی را با سایر معیارها بررسی می‌کنیم: اگر میزان دانه‌بندی یک سرویس بالا باشد، انسجام و اتصال، هر دو مقادیر بالایی خواهند داشت چرا که ارتباطات داخل سرویس افزایش می‌یابد و به همین نسبت هم چون عملیات زیادی در یک سرویس متمرکز شده‌اند، نیاز دیگر سرویس‌ها به آن سرویس افزایش خواهد یافت. به همین ترتیب با کاهش میزان دانه‌بندی، انسجام و اتصال سرویس نیز کاهش می‌یابد. با این وجود، از آنجا که ما به دنبال انسجام بالا و اتصال پایین و در عین حال دانه‌بندی متوسط هستیم، این اهداف را نمی‌توان با هم در قالب یک تابع هدف ادغام کرد. به علاوه، اگرچه به نظر می‌رسد انسجام و اتصال و همگرایی حرفه با هم ارتباط مستقیم یا عکس دارند، اما هیچ پژوهشی توسط نویسندگان یافته نشد که نشان‌دهنده ارتباط صریحی باشد که همواره بین آنها برقرار باشد؛ به عبارت دیگر تا به حال اثبات نشده است که همواره و همیشه با افزایش اتصال یا با کاهش همگرایی موجودیت‌های حرفه، انسجام کاهش می‌یابد یا برعکس.

برای حل مسأله بهینه‌سازی چندهدفه در این تحقیق، از یک الگوریتم ژنتیک به نام NSGA-II[۲۷]استفاده می‌کنیم. این الگوریتم برای حل مسائل بهینه‌سازی چندهدفه طراحی شده و نسبت به نسخه پیشین آن (NSGA) از پیچیدگی محاسباتی کمتری برخوردار است. بنا بر این، این الگوریتم برای صورت مسأله حاضر که در آن تعداد توابع هدف زیاد است مناسب است.

الگوریتم NSGA-II مبتنی بر یافتن راه‌حل بهینه پارتو است. بهینه پارتو به راه‌حلی گفته می‌شود که در فضای راه‌حل، راه حل دیگری بر آن چیره نباشد[۲۸]. در واقع راه‌حل x بر راه حل y چیره است اگر دو شرط زیر برقرار باشد: (۱)مقدار هیچ یک از توابع هدف x از توابع هدف y بدتر نباشد، (۲)مقدار حداقل یکی از توابع هدف برای x از y اکیداً بهتر باشد.

5. Modeling Capability

الگوریتم ژنتیک NSGA-II: این الگوریتم که در سال ۲۰۰۰ توسط «دِب» و همکارانش ارائه شده[۲۷]از نخبه‌گرایی و انتخاب تورنمنت جمعیتی[۶] استفاده می‌کند. در این الگوریتم یک رویکرد سریع مرتب‌سازی غیرمغلوب معرفی می‌شود که دارای پیچیدگی محاسباتی $O(mN^2)$ است که در آن، m تعداد توابع هدف و N اندازه جمعیت است. انتخاب تورنمنت جمعیتی، مکانیزمی برای انتخاب مبتنی بر انتخاب تورمنت است که در آن گروهی از افراد (کروموزوم‌ها) در یک تورنمنت شرکت می‌کنند و بررسی می‌شود که برنده در چه سطحی از تناسب[۷] قرار دارد. سطوح تناسب بر اساس رتبه اختصاص داده شده به راه‌حل و فاصله جمعیتی[۸] محاسبه می‌شود. فاصله جمعیتی به عنوان تخمینی از محیط مستطیل ایجاد شده توسط نقاط نماینده نزدیک‌ترین راه‌حل‌های همسایه ایجاد می‌شود عمل می‌کند، این فاصله بر اساس مقادیر نرمال شده توابع هدف عمل می‌کند. برای جزئیات بیشتر به[۲۷]مراجعه کنید.

کدگذاری برای مسأله: در اولین قدم باید برای حل مسأله کدگذاری خاصی را تعیین کنیم. یعنی ساختار کروموزوم‌ها و تعداد و ساختار ژن‌های آن باید مشخص شود.

می‌خواهیم مجموعه فعالیت‌های BPFM را طوری افراز کنیم که مقادیر معیارهای خاصی (توابع هدف) برای همه مجموعه‌های اصلی بیشینه شود. k فعالیت موجود در BPFM حداکثر می‌توانند به k مجموعه که در هر یک تنها یک فعالیت وجود دارد افراز شوند. در نتیجه، ساختار کروموزوم‌ها در راه‌حل پیشنهادی، دارای k متغیر (ژن) است که به هریک، یک شماره سرویس اختصاص داده می‌شود؛ پس ارزش هر متغیر، شماره سرویس تخصیص داده شده به آن است. بر این اساس ساختار ژن‌ها به صورت «(شناسه فعالیت، شناسه سرویس)» تعیین می‌شود. در این صورت واضح است که حداکثر k سرویس خواهیم داشت. اگر یک BPFM فرضی شامل ۵ فعالیت باشد، ساختار کروموزوم و مقادیر فرضی ژن‌ها می‌تواند به صورت شکل ۴ باشد.

(Act#1, Ser#3)	(Act#2, Ser#1)	(Act#3, Ser#3)	(Act#4, Ser#2)	(Act#5, Ser#1)

شکل ۴. ساختار کروموزوم برای یک BPFM فرضی با ۵ فعالیت

کرموزوم شکل ۴ نشان می‌دهد که راه‌حل به‌دست آمده، سه سرویس را در خود دارد. سرویس Ser#1 شامل فعالیت‌های Act#2 و Act#5 است، سرویس Ser#2 شامل فعالیت Act#4 و سرویس Ser#3 شامل فعالیت‌های Act#1 و Act#3 می‌باشد.

جمعیت اولیه: با تخصیص مقادیر تصادفی بین ۱ تا k به شناسه‌های سرویس‌های مربوط به هر فعالیت شکل می‌گیرد.

ارزیابی راه‌حل‌ها: ارزیابی هر راه‌حل (مجموعه سرویس) بر اساس توابع هدف انجام می‌شود. راه‌حل‌هایی بهینه هستند که مقادیر توابع هدف را که بر اساس معیارهای سرویس‌گرایی و تغییرپذیری می‌باشند برای همه مجموعه‌های اصلی بیشینه کنند. تعداد توابع هدف برابر است با «۴*تعداد مجموعه‌های اصلی +۱» است زیرا چهار معیار سرویس‌گرایی (انسجام، اتصال، دانه‌بندی و همگرایی موجودیت‌های حرفه[۲۴]) برای یک راه حل بالقوه، به ازای هر مجموعه اصلی اندازه‌گیری می‌شوند و تنها معیار تغییرپذیری (درجه اشتراک[۲۹]) برای کل راه‌حل یک مقدار منفرد به دست می‌دهد.

برای اندازه‌گیری معیارهای مورد نظر از ماتریس CRUD استفاده می‌کنیم. نحوه محاسبه توابع هدف در جدول ۱ نشان داده شده است. توضیح این که درجه اشتراک (V_{COMM}) از تقسیم تعداد عناصر مشترک بر تعداد کل عناصر (در اینجا سرویس‌ها) به دست می‌آید. همچنین عدد ۱۰۰ در معادلات مربوط به توابع هدف اتصال، دانه بندی و درجه اشتراک، به صورت تجربی انتخاب شده است. مثلا برای معیار اتصال، این مقدار هم تقریب خوبی از اتصال پایین به دست می‌دهد (اتصال ۰،۰۱) و هم به اندازه ای کوچک است که روی مقادیر توابع هدف دیگر و فاصله جمعیتی مورد استفاده در الگوریتم NSGA-II تأثیر منفی نگذارد.

جدول ۱. توابع هدف و ارتباط آنها با معیارها

نام معیار	تابع هدف مربوطه
انسجام(v_{COHE})	$Obj_{Cohe} = v_{COHE}$
اتصال(v_{COUP})	$Obj_{Coup} = \begin{cases} \frac{1}{v_{COUP}} & if\ v_{COUP} \neq 0 \\ 100 & else \end{cases}$
دانه‌بندی (v_{GRAN})	$Obj_{gran} = \begin{cases} \frac{1}{\lvert v_{GRAN} - 0.5 \rvert} & if\ v_{GRAN} \neq 0.5 \\ 100 & else \end{cases}$
همگرایی موجودیت‌های حرفه (v_{CONV})	$Obj_{Conv} = v_{CONV}$
درجه اشتراک (v_{COMM})	$Obj_{Comm} = \begin{cases} \frac{1}{\lvert v_{COMM} - 0.5 \rvert} & if\ v_{COMM} \neq 0.5 \\ 100 & else \end{cases}$

در ارتباط با دانه‌بندی لازم به ذکر است که سرویس‌های درشت دانه میان محصولات متنوع خط تولید سرویس‌گرا قابلیت استفاده مجدد بالایی دارند زیرا فعالیت‌های حرفه زیادی را دربرمی‌گیرند و در محصولات مختلف مورد استفاده قرار می‌گیرند؛ اما در عین حال

6. Crowded Tournament Selection
7 .Fitness
8 .Crowding Distance

استفاده مجدد از این سرویس‌ها در دامنه‌های مختلف و در محصولات آینده دشوارتر است و قابلیت نگهداری سرویس‌های درشت‌دانه نیز پایین خواهد بود؛ به‌ویژه به دلیل این‌که سرویس درشت‌دانه‌تر اتصال‌های بیشتر به همراه دارد. بنابراین ایده‌آل آن است که دانه‌بندی سرویس‌ها نزدیک به حد وسط (۰,۵) باشد.

به‌طور مشابه همین مسأله برای درجه اشتراک برقرار است که مقداری بین صفر و یک خواهد داشت. اگر درجه اشتراک برابر با یک باشد، به معنای آن است که تمام عناصر (در اینجا سرویس‌ها) در تمام محصولات مورد نیازند. این بالاترین درجه قابلیت استفاده مجدد سرویس‌ها را فراهم می‌کند؛ اما در عین‌حال، هیچ فضایی برای تنوع و تغییرپذیری میان محصولات باقی نمی‌گذارد.

تجربه[۲۹]نشان داده است که درجه اشتراک ایده‌آل، مقادیر نزدیک به ۵۰٪ است.

اگر تعداد مجموعه‌های اصلی برای یک BPFM فرضی کوچک، برابر با ۲ باشد، بردار توابع هدف برای آن به صورت شکل ۵ خواهد بود که دارای $4 * 2 + 1 = 9$ تابع است.

پارامترهای الگوریتم NSGA-II: پیاده‌سازی الگوریتم به زبان C# و در محیط ویژوال استودیو ۲۰۱۰ انجام شد. ورودی‌ها و خروجی الگوریتم در قالب فایل‌های متنی طراحی شد. عملگر انتخاب، انتخاب تورنمنت جمعیتی و عملگر ترکیب مورد استفاده، عملگر ترکیب تک‌نقطه‌ای بوده و سایر پارامترهای الگوریتم مطابق جدول ۲ تعیین شدند

$Obj_{Cohe}1$ $Obj_{Cohe}2$	$Obj_{Coup}1$ $Obj_{Coup}2$	$Obj_{gran}1$ $Obj_{gran}2$	$Obj_{conv}1$ $Obj_{conv}2$	Obj_{Comm}
توابع انسجام	توابع اتصال	توابع دانه بندی	توابع همگرایی موجودیت‌های حرفه	تابع درجه اشتراک

شکل ۵. بردار توابع هدف برای یک BPFM با دو مجموعه اصلی

انتخاب بهترین راه‌حل

در اغلب الگوریتم‌های تکاملی و از جمله الگوریتم ژنتیک، احتمال آن وجود دارد که الگوریتم به یک راه‌حل بهینه محلی همگرا شده و از یافتن بهینه واقعی باز بماند. به همین دلیل، این الگوریتم‌ها را اغلب n بار تکرار کرده و از بین راه‌حل‌های به دست آمده بهترین را برمی‌گزینند. در تحقیق حاضر ما تعداد اجراهای الگوریتم را به ازای ورودی‌های ثابت برابر ۳۰ در نظر گرفته‌ایم(n=۳۰).

انتخاب بهترین راه حل از میان ۳۰ راه حل به دست آمده موضوعی قابل بحث است. پیشنهاد ما این است که ابتدا از میان این راه‌حل‌ها، مجموعه بهینه پارتو انتخاب شود. اگر این مجموعه تنها دارای یک عضو باشد، راه حلی که بر همه راه‌حل‌های دیگر چیره است (راه‌حل بهینه) به دست می‌آید. در غیر این صورت، مجموعه پارتو شامل k≤۳۰ راه‌حلی است که اگرچه بر همه راه‌حل‌های دیگر چیره نیستند، اما راه‌حل‌هایی هستند که هیچ راه‌حل دیگری بر آنها چیره نیست. گزینه نهایی به دو صورت دستی (توسط مهندس یا معمار دامنه) و خودکار می‌تواند انتخاب شود. روش دستی (که روش پیشنهادی را نیمه‌خودکار می‌کند) به مهندس و یا معمار دامنه امکان می‌دهد تا بتواند پارامترهای خاص دیگری را نیز در انتخاب خود دخالت دهد.

جدول ۲. پارامترهای تعیین شده برای الگوریتم NSGA-II

پارامتر	مقدار
تعداد نسلها	۱۰۰۰
اندازه جمعیت	۵۰۰
نرخ ترکیب	۰,۹
احتمال جهش	۰,۱

در روش خودکار با الهام گرفتن از فاصله جمعیتی مطرح شده در NSGA-II[۲۷]و یا حتی با استفاده از فاصله‌های ساده‌تری مانند فاصله اقلیدسی می‌توان بهترین راه‌حل را (که بیشترین فاصله را با سایر راه‌حل‌ها در مجموعه پارتو دارد) یافت. ما در پیاده‌سازی روش، از مقایسه مجموعه بهینه پارتو به دست آمده از مجموعه ۳۰ عضوی بهترین راه‌حل‌ها بر مبنای بیشترین فاصله جمعیتی برای این منظور استفاده کردیم.

تعیین واسط سرویس‌ها

تعیین واسط سرویس‌ها خود دارای دو مرحله است: اول، رمزگشایی[9] راه‌حل‌ها است. در این مرحله مشخص می‌شود که در هر راه‌حل چه سرویس‌هایی وجود دارند و در هر یک از آنها چه فعالیت‌های حرفه‌ای پوشش داده می‌شوند. مرحله بعد، تعیین واسط هر یک از سرویس‌ها در قالب تعیین عملیات هر سرویس و پیام‌های ورودی و خروجی هر یک از این عملیات است. هر یک از فعالیت‌های حرفه در سرویس، نشان‌دهنده یک عملیات آن است که با همان نام در سرویس درج می‌شود. پیام‌های ورودی و خروجی این عملیات با استفاده از ماتریس CRUD انجام می‌شود. الگوریتم ۱ این مرحله را نشان می‌دهد.

تعیین تغییرپذیری سرویس‌ها

در این مرحله باید اجباری یا اختیاری بودن هر سرویس و وجود رابطه جایگزینی (یای انحصاری)، یای غیرانحصاری و استلزام میان

9. Decode

سرویس‌ها را تعیین کنیم. الگوریتم ۲ مربوط به تعیین تغییرپذیری سرویس‌ها است.

Algorithm1: Service Interface Specification

Input: Best Solution of NSGA-II (Chromosome), CRUD Matrix (M)

Output: Specified Service Set

```
Begin
  Service Set = Generate Service Set(Chromosome);
  For each svc in Service Set do
    For each business activity i assigned to svc do
      Begin
      row_i = find the row for business activity i in M;
      If row_i is not empty then
       Begin
       add operation op_i to svc();
       For each business entity j do
        case (M_i,j)
        Begin
          "C": add BE_j as an out_message to op_i();
          "R": add BE_j as an in_message to op_i();
          "U":Begin
              add BE_j as an in_message to op_i();
              add BE_j as an out_message to op_i();
              End
         "D": add BE_j as an in_message to op_i();
         Endcase
       Add service to Specified Service Set;
       Endif
      Endfor
End
```

Algorithm2: Service Variability Identification

Input: Specified Service Set (SSS), Prime Set (PS)

Output: Service variability Model (SVM)

```
Begin
  SVM=Empty;
  For each service in SSS do
  Begin
    For each p in PS do
      If !(IsUsed(service,p)) then//service isn't used in p
      Begin
        Add service to SVM.OptionalServiceSet;
        ServiceIsOptional=True;
        Break;
      Endif
      If (ServiceIsOptional=false) then
        Add Service to SVM.MandatoryServiceSet;
  Endfor
  For each s_i in optionalServiceSet do
    For each s_j!=s_i in optionalServiceSet do
    Begin
      For each p in PS do
      Begin
        If !((IsUsed(s_i,p)XOR(IsUsed(s_j,p))) then
          AreAlt=false;
        If !((IsUsed(s_i,p)OR(IsUsed(s_j,p))) then
          AreOR=false;
        If !(!(IsUsed(s_i,p)OR(IsUsed(s_j,p))) then
          AreIncl=false;
      End For
      If !(AreAlt=false) and (s_j,s_i) not in
          SVM.AltRelations then
          add (s_i,s_j) to SVM.AltRelations;
      If !(AreOr=false) and (s_j,s_i) not in
          SVM.OrRelations then
          add (s_i,s_j) to SVM.OrRelations;
      If !(AreIncl=false) then
          add (s_i,s_j) to SVM.IncRelations;
    End for
  End for
End
```

ما روش پیشنهادی و الگوریتم‌های ارائه شده را با استفاده از چارچوب jMetal [۳۷]با زبان C# پیاده‌سازی کردیم.

ارزیابی

رویکرد و نحوه ارزیابی

دو رویکرد عمده برای ارزیابی تحقیق‌های حوزه مهندسی نرم‌افزار انجام مطالعه موردی و آزمایش کنترل شده هستند[۳۰]. انجام مطالعه موردی با توجه به این که در فضای واقعی و صنعت انجام می‌شود به اعتبار سنجی بیرونی (قابلیت تعمیم نتایج به جامعه وسیعتر، در اینجا کل خطوط تولید سرویس‌گرا) کمک می‌کند؛ در

حالی که آزمایش کنترل شده برای اعتبار سنجی درونی (قابلیت استخراج نتیجه از داده‌ها به وسیله تحلیل داده‌ها) مفیدتر است. با توجه به این که اعتبارسنجی درونی مقدم بر اعتبارسنجی بیرونی است، از آزمایش تجربی[10] استفاده کرده و سعی کرده‌ایم موردهای آزمایش کاملاً برگرفته از صنعت باشند تا در آینده بتوان روی اعتبارسنجی بیرونی تحقیق نیز کار کرد.

همچنین برای انجام ارزیابی از رویکرد هدف-پرسش-معیار (GQM)[۳۵]استفاده کردیم که در آن هدف یا اهداف ارزیابی تعیین شده و به ازای هر هدف، مجموعه‌ای از پرسش‌های ارزیابی تعیین می‌شود. سپس جهت پاسخ به هر پرسش یک یا چند معیار جهت اندازه‌گیری معین می‌گردد.

هدف ارزیابی(G): *تحلیل* روش ارائه شده *به منظور* ارزیابی روش *از نقطه‌نظر* معیارهای کیفی *در زمینه* خط تولید سرویس‌گرا.

پرسش‌های ارزیابی(Q): پرسشهای ارزیابی عبراند از:

پ۱- آیا سرویس‌های شناسایی شده توسط روش با سرویس‌های شناسایی شده توسط خبرگان حوزه سرویس‌گرایی و خط تولید نرم‌افزار شباهت دارند؟

پ۲ – آیا مدل‌های SVM به دست آمده توسط روش با توجه به اهداف خط تولید مناسب هستند؟

معیارهای ارزیابی برای پرسشهای ارزیابی عبارتند از:

معیارهای ارزیابی(M) برای پرسش پ۱: م۱- میانگین درصد شباهت سرویس‌های شناسایی شده توسط روش با سرویس‌های شناسایی شده توسط خبرگان از نظر مقادیر توابع هدف(شباهت مقداری)؛ **م۲**- میانگین درجه شباهت ساختاری سرویس‌های فوق.

معیارهای ارزیابی(M) برای پرسش پ۲: م۳- نسبت مدل‌های سازگار (بدون تناقض) ایجاد شده به کل مدل‌های ایجاد شده توسط روش(درجه سازگاری)؛ **م۴**- میانگین درجه قابلیت استفاده مجدد سرویس‌ها در خط تولید (درجه اشتراک).

به علاوه، ویژگی‌های آزمایش به شرح زیر است:

پروتکل آزمایش: سه خط تولید برای آزمایش انتخاب شدند که هریک دارای BPFM مجزایی هستند که بر اساس فرایندهای حرفه موجود در شرکت‌های تولید مواد شوینده، پخش و فروش و نیز تولید مواد غذایی و با همراهی مهندسی دامنه و خبرگان حرفه طراحی شده‌اند. ماتریس‌هایCRUD مربوط به فرایندها نیز طراحی شده و در کنار مدل‌های BPFM ورودی‌های آزمایشی روش را تشکیل دادند. همچنین این نمودارها و ماتریس‌ها در اختیار تعدادی از خبرگان حوزه سرویس‌گرایی و خط تولید نرم‌افزار قرار گرفت.

شرکت کنندگان: پنج نفر از افراد واجد شرایط زیر به عنوان شرکت کننده در آزمایش (خبره) انتخاب شدند: (۱) دارای تجربه عملی در زمینه مهندسی نرم‌افزار، (۲) دارای دانش نظری یا تجربه عملی در زمینه سیستم‌های سرویس‌گرا و مهندسی خط تولید نرم‌افزار. کلیه افراد، دارای مدرک کارشناسی ارشد از دانشگاه‌های شهید بهشتی، تهران و تربیت مدرس بوده و علاوه بر دانش نظری در هر دو حوزه، دارای تجربه عملی در زمینه سرویس‌گرایی بین ۱ تا ۶ سال بودند.

مواد آزمایش: سه مدل خانواده فرایند حرفه مربوط به فرایندهای خرید (مورد الف)، پردازش سفارش (مورد ب) و ثبت دارایی ثابت (مورد پ) به همراه ماتریس‌های CRUD مربوطه و نیز یک مستند آموزشی شامل مفاهیم و روش آزمایش و موارد خواسته شده از خبرگان از مواد این آزمایش بودند.

روال آزمایش: روال آزمایش دارای دو فاز بود: پیش آزمایش و آزمایش. در مرحله پیش آزمایش مستندی آموزشی تهیه شده و در اختیار شرکت کنندگان قرار گرفت و به مدت یک هفته به آنها زمان داده شد تا آن را مطالعه کرده و سؤالات و ابهام‌های خود را مطرح کنند. در فاز آزمایش، ورودی‌های مربوطه یعنی نمودارهای BPFM و ماتریس‌های CRUD به همراه مستند آموزشی در اختیار شرکت‌کنندگان قرار گرفت تا در مدت دو ساعت در شرایطی مشابه، سرویس‌های مورد نظر خود را طراحی کنند. جدول۳ مشخصات موردهای آزمایش را نشان می‌دهد.

نتایج ارزیابی

برای پاسخ به *پرسش ارزیابی* پ۱، باید معیارهای م۱ و م۲ را بسنجیم.

برای محاسبه معیار م۱، برای هر تابع هدف از فرمول رابطه (۱) استفاده می کنیم:

$$scalarSimilarity = \frac{\sum_{i=1}^{n} 1-|\frac{\bar{x}_m-\bar{x}_e}{\bar{x}_e}|}{n} \quad (۱)$$

که در آن ابتدا برای هر تابع هدف، فاصله میانگین مقادیر به دست آمده توسط روش ($\bar{x}_m$) با میانگین مقادیر به دست آمده توسط خبرگان ($\bar{x}_e$) محاسبه می‌شود. این فواصل به صورت قدر مطلق محاسبه می شوند تا اثر یکدیگر را خنثی نکنند(در این رابطه، n تعداد موردهای آزمایش است که در اینجا برابر با ۳ می باشد). شباهت مقداری این مقادیر برابر است با یک منهای فاصله آنها. میانگین مقادیر شباهت مقداری، نشان‌دهنده متوسط شباهت مقداری میان مقادیر تابع هدف مورد نظر به دست آمده توسط

10. Empirical

روش و توسط خبرگان است. میزان میانگین شباهت مقداری برای توابع هدف به همراه جزئیات بیشتری از مقادیر توابع هدف به دست آمده توسط روش پیشنهادی و نیز توسط خبرگان در جدول ۴ ارائه شده است. میانگین کل میزان شباهت‌های مقداری میان نتایج روش با نتایج خبرگان، برابر با ۸۰,۴۴٪ می‌باشد.

جدول ۳. مشخصات موردهای آزمایش

مورد آزمایش	تعداد موجودیت‌های حرفه	تعداد فعالیت‌ها	تعداد مجموعه‌های اصلی	کروموزوم بهترین راه‌حل
الف	۸	۱۵	۵	((۱,۳), (۲,۹), (۳,۸), (۴,۷), (۵,۱۴), (۶,۶), (۷,۱۲), (۸,۱۳), (۹,۸), (۱۰,۱۱), (۱۱,۱۰), (۱۲,۴), (۱۳,۱۵), (۱۴,۱۴), (۱۵,۱))
ب	۶	۱۲	۴۴	((۱,۳), (۲,۴), (۳,۹), (۴,۱۰), (۵,۴), (۶,۱۰), (۷,۱۰), (۸,۴), (۹,۱), (۱۰,۴), (۱۱,۵), (۱۲,۳))
پ	۷	۱۰	۱۴	((۱,۱), (۲,۵), (۳,۲), (۴,۷), (۵,۴), (۶,۱۰), (۷,۱۰), (۸,۳), (۹,۴), (۱۰,۱))

جدول ۴. مقایسه متوسط مقادیر توابع بین روش پیشنهادی و خبرگان

مورد آزمایش		میانگین تابع هدف انسجام	میانگین تابع هدف اتصال	میانگین تابع هدف دانه بندی	میانگین تابع هدف همگرایی...	تابع هدف درجه اشتراک
مورد الف	روش	۰,۹۶	۰,۰۵۶	۲۰,۳۰	۱,۱۵	۲,۸۸
	خبرگان	۰,۸۱	۰,۰۴۰	۱۷,۱	۱,۵۵	۳,۱۶
مورد ب	روش	۰,۷۸	۰,۱۷	۱۱,۸	۱,۳۳	۶
	خبرگان	۰,۸۴	۰,۱۵	۱۱,۱	۰,۸۸	۶
مورد پ	روش	۰,۸۷	۰,۱۷	۱۳	۰,۹۵	۴,۶۶
	خبرگان	۰,۸۶	۰,۲۵	۹,۶۶	۱,۲	۵,۵
میانگین شباهت مقداری		٪۹۱,۱	٪۷۱,۶	٪۸۰,۱	٪۶۷,۴	٪۹۲

جهت بررسی معیار م۲ و مقایسه شباهت ساختاری نتایج نیز از یک معیار مشابهت ساختاری مبتنی بر معیار فاصله ارائه شده در[۳۱](رابطه ۲) جهت مقایسه دو مجموعه از مجموعه‌ها بهره گرفتیم، زیرا هر خروجی روش، مجموعه‌ای است از سرویس‌ها که هریک، خود مجموعه‌ای از عملیات هستند و با استفاده از روش مقایسه مجموعه‌ای از مجموعه‌ها می‌توان آنها را مقایسه نمود). قبل از انجام مقایسه، گرافی دوبخشی از دو مجموعه سرویس S_i و S_j تحت مقایسه تشکیل می‌شود که در آن هر مجموعه در یکی از دو بخش گراف قرار می‌گیرد. هر یال میان یک سرویس از بخش اول با سرویس دیگر از بخش دوم گراف، دارای وزنی است که نشان دهنده شباهت آن دو سرویس است. از آنجا که سرویس‌ها خود، مجموعه‌ای از عملیات هستند، شباهت دو سرویس منفرد m و n (یعنی SSD(m, n)) از رابطه ۲ به دست می‌آید که در آن SO_i نشان دهنده مجموعه عملیات سرویس i است.

$$SSD(m,n) = \frac{|SO_m \cap SO_n|}{|SO_m \cup SO_n|} \quad (۲)$$

شباهت یک سرویس (سرویس m) از مجموعه سرویس اول با کل سرویس‌های مجموعه سرویس دوم (S_n) را از رابطه (۳) به دست می‌آوریم. این مقدار برابر با حداکثر شباهت میان سرویس m با هر یک از k سرویس موجود در مجموعه S_n است.

$$SC(m,S_n) = Max\left(SSD\left(m,S_{n,j}\right)\right) \quad \forall j = 1,2,\dots,k \quad (۳)$$

در نهایت، در رابطه ۴، $SS(S_i,S_j)$ به دست می‌آید که نشان‌دهنده شباهت ساختاری میان دو سرویس Si و Sj است (در این رابطه، E مجموعه یال‌های گراف دوبخشی و p تعداد سرویس‌های مجموعه سرویس S_i است).

$$SS(S_i,S_j) = \frac{\sum_{n=1}^{p} SC(S_{i,n},S_j)}{|E|} \quad (۴)$$

نتایج این مقایسه که در جدول ۵ ارائه شده است، نشان می‌دهد که به‌طور میانگین، هر مورد آزمایش با مدل‌های طراحی شده توسط خبرگان، ٪۵۶,۶۶ شباهت ساختاری دارد.

بنا بر این پاسخ به *پرسش ارزیابی* پ ۱ آن است که میانگین درجه شباهت مقداری توابع هدف میان روش پیشنهادی با خبرگان، (م۱) برابر با ٪۸۰,۴۴ و میزان شباهت ساختاری سرویس‌های به دست آمده میان روش و خبرگان(م۲)، نزدیک به متوسط و برابر با ٪۵۶,۶۶ می‌باشد.

جدول ۵. مشابهت ساختاری خروجی روش برای موردهای آزمایش با مدل ارائه شده توسط خبرگان

مورد	خبره ۱	خبره ۲	خبره ۳	خبره ۴	خبره ۵	میانگین
الف	٪۷۳	٪۵۸	٪۶۵	٪۵۶	٪۳۹	٪۵۸,۲
ب	٪۶۵	٪۶۶	٪۵۶	٪۴۷	٪۴۳	٪۵۵,۴
پ	٪۵۲	٪۶۴	٪۵۹	٪۵۰	٪۵۷	٪۵۶,۴

جهت پاسخ به پرسش *ارزیابی* پ۲، باید معیارهای م۳ و م۴ را اندازه گیری نماییم.

معیار م۳، عبارت از تعداد مدل‌های سازگار (#consistentSVM) نسبت به کل مدل‌های ایجاد شده توسط روش (#SVM) است (رابطه ۵).

$$ConsistencyDegree = \frac{\#consistentSVM}{\#SVM} \quad (۵)$$

منظور از سازگاری، عدم وجود تناقض در مدل به دست آمده توسط روش (SVM) است که یکی از معیارهای کیفی مهم در مدل‌های تغییرپذیری می‌باشد. ما برای تعیین سازگاری مدل‌های SVM، از ابزار ارائه شده توسط SPLOT[۳۲]استفاده کردیم. نتیجه این بررسی نشان داد که ٪۱۰۰ موردهای آزمایش سازگار بودند (م۳) و الگوریتم ارائه شده در این روش، سازگاری مدل‌ها را تضمین می‌نماید.

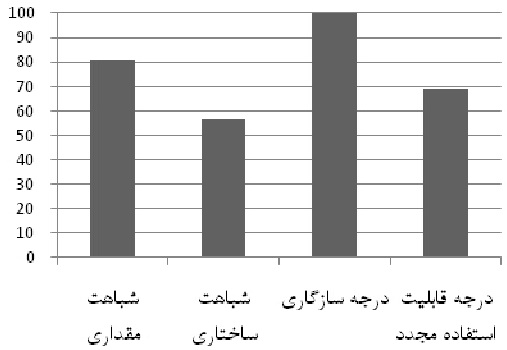
همچنین، از درجه اشتراک به عنوان معیار کیفی مهمی که نشان-دهنده درجه قابلیت استفاده مجدد سرویس‌ها در خط تولید است برای محاسبه م۴ استفاده شد. درجه اشتراک در سه مورد آزمایش به ترتیب ٪۸۴، ٪۶۷ و ٪۵۷ به دست آمد که میانگین قابلیت استفاده مجدد ٪۶۹,۳۳ را برای سرویس‌های شناسایی شده در خط تولید مربوطه به دست می‌دهد(م۴).

بنا بر این پاسخ به پرسش *ارزیابی* پ۲ آن است که ٪۱۰۰ مدل-های به دست آمده از روش سازگارهستند (م۳) و درجه قابلیت استفاده مجدد سرویس‌های شناسایی شده در خط تولید، به‌طور میانگین، ٪۶۹,۳۳ می‌باشد(م۴) یعنی هر سرویس به طور متوسط در ٪۶۹,۳۳ محصولات خط تولید مربوطه مورد استفاده واقع می-شود.

جدول ۶ و شکل ۶ به ترتیب، خلاصه نتایج ارزیابی با رویکرد GQM و مقایسه میزان معیارهای چهارگانه مطرح شده را ارائه می‌دهند.

جدول ۶. نتایج به دست آمده با رویکرد GQM

G	Q	M	
هدف	پرسش پ۱	م۱ - شباهت مقداری	٪۸۰,۴۴
		م۲- شباهت ساختاری	٪۵۶,۶۶
	پرسش پ۲	م۳- درجه سازگاری	٪۱۰۰
		م۴- درجه قابلیت استفاده مجدد	٪۶۹,۳۳



شکل ۶. مقادیر معیارهای چهارگانه (به درصد)

ویژگی‌ها و محدودیت‌های روش پیشنهادی

ویژگی‌های روش پیشنهادی که نشان‌دهنده سهم علمی تحقیق نیز هست عبارتند از: شناسایی (نیمه)خودکار سرویس‌ها برای خط تولید به همراه ویژگی‌ها و روابط تغییرپذیری آنها، مبتنی بودن روش بر فرایندهای حرفه در خط تولید و تغییرپذیری آنها، لحاظ کردن معیارهای طراحی سرویس‌ها و معیارهای مربوط به طراحی خط تولید در شناسایی سرویس‌ها.

روشی که به طور مشابه با روش پیشنهادی بر اساس تغییرپذیری فرایندهای حرفه اقدام به شناسایی (نیمه) خودکار سرویس‌های خط تولید و تغییرپذیری آنها نماید، توسط نویسندگان مقاله حاضر یافته نشد و لذا مقایسه میان روش‌های مشابه منتفی است، اما تحقیقات دیگری روی مدل‌سازی سرویسها و یا شناسایی عناصر خط تولید وجود دارند که می‌توان روش پیشنهادی را بر اساس تعدادی شاخص با آنها مقایسه نمود. این شاخص‌ها به همراه نتایج مقایسه در جدول ۷ نشان داده شده‌اند. در واقع روش‌های موجود به دو دسته تقسیم می‌شوند: روش‌هایی که به شناسایی سرویس‌ها مستقل از خط تولید و تغییرپذیری آنها می‌پردازند (این روش‌ها بر اساس اصول سرویس‌گرایی و مطابق نیازی که واقعاً در صنعت وجود دارد، اغلب مبتنی بر فرایندهای حرفه هستند). دسته دوم متعلق به روش‌هایی است که برای شناسایی سرویس‌ها و تغییرپذیری آنها جهت خط تولید نرم‌افزار ارائه شده‌اند. این روش‌ها اغلب مبتنی بر مدل ویژگی هستند؛ چرا که در حیطه خط تولید نرم افزار مدل ویژگی به عنوان یک مدل تغییرپذیری سطح

نیازمندی‌ها، مدلی شناخته شده و بسیار پرکاربرد است. تحقیقات زیادی در حیطه ارائه مدل تغییرپذیری برای خط تولید سرویس‌گرا بر اهمیت جایگاه فرایندهای حرفه تأکید دارند ولی هیچ‌یک از روش‌های شناسایی سرویس‌ها برای خط تولید سرویس‌گرا که توسط نویسندگان بررسی شد، مبتنی بر فرایندهای حرفه نیستند.

شاید بتوان گفت محدودیت اصلی روش پیشنهادی آن است که برای ردیابی مدل خانواده فرایندهای حرفه به مدل ویژگی و ردیابی فعالیت‌های درونی آن به ویژگی‌ها سازوکاری پیشنهاد نشده است. ما این بخش از کار را به تحقیقات آینده موکول می‌کنیم.

جدول ۷. مقایسه روش‌های شناسایی سرویس‌ها یا شناسایی معماری برای خط تولید نرم‌افزار

شاخص	[۲۶]	[۳۶]	[۱۵]	[۱۶]	[۱۷]	[۱]	[۱۹]	روش پیشنهادی
پشتیبانی از معماری سرویس گرا	✓	✓	✓	✓	✓	✓	_	✓
پشتیبانی از فرایندهای حرفه	✓	✓	_	_	_	_	_	✓
پشتیبانی از مدل ویژگی	_	_	✓	✓	_	✓	✓	_
شناسایی تغییرپذیری	_	_	✓	✓	_	✓	✓	✓
شناسایی عناصر معماری بر اساس معیارهای کیفی	✓	✓	_	_	✓	_	✓	✓
ارائه الگوریتم یا مراحل دقیق پیاده‌سازی	✓	✓	_	✓	✓	_	✓	✓

نتیجه گیری و کارهای آتی

در این مقاله، روشی را برای استخراج مدل تغییرپذیری سرویس برای خط تولید سرویس‌گرا مبتنی بر الگوریتم تکاملی NSGA-II ارائه کردیم. نتیجه این تحقیق، روشی نوین است برای شناسایی سرویس‌ها و تغییرپذیری آنها برای خط تولید سرویس‌گرا، به‌طوری که با فرایندهای حرفه همراستا بوده و اهداف حرفه را در کنار معیارهای کیفی طراحی برآورده سازد.

نتایج ارزیابی تجربی با رویکرد GQM نشان داد که سرویس‌های شناسایی شده توسط این روش از نظر معیارهای طراحی اغلب بهتر از خبرگان طراحی این حوزه بوده و در بدترین حالت‌ها بسیار به آنها نزدیک است؛ در واقع میان معیارهای کیفی طراحی سرویس-های حاصل از روش و سرویس‌های به دست آمده توسط خبرگان مشابهت مقداری معناداری دیده می‌شود. همچنین شباهت ساختاری نسبتاً بالایی (کمی بیشتر از متوسط ۵۰٪) میان سرویس‌های شناسایی شده توسط روش و سرویس‌های شناسایی شده توسط خبرگان مشاهده می‌گردد که با توجه به لحاظ نکردن بررسی مشابهت معنایی سرویس‌ها در روش پیشنهادی، و وجود آن در عملکرد خبرگان، قابل قبول به نظر می‌رسد.

از نظر معیارهای خط تولید نرم‌افزار، دو معیار سازگاری و درجه قابلیت استفاده مجدد در خط تولید بررسی شد و نتایج نشان داد که کلیه مدل‌های ایجاد شده توسط روش سازگار بوده و درجه قابلیت استفاده مجدد آنها دارای میزان بالایی است. در نتیجه روش ارائه شده می‌تواند برای نیل به هدف اصلی خط تولید نرم افزار یعنی قابلیت استفاده مجدد برنامه‌ریزی شده کمک کننده باشد بدون این که تناقض و ناسازگاری در مدل‌های خروجی در سیدن به این هدف مانعی ایجاد نماید.

این روش اولین روش ارائه شده برای شناسایی سرویس‌های متغیر نزدیک به بهینه بر اساس فرایندهای حرفه متغیر جهت خط تولید سرویس‌گرا و با توجه به معیارهای کیفی است که حتی اگر به طور مطلق جایگزین فعالیت معمار دامنه خط تولید سرویس‌گرا نشود می‌تواند مدل اولیه مناسبی را برای او فراهم کند تا او با اعمال تغییرات مورد نظر و محاسبه معیارهای کیفی بتواند طراحی بهتری انجام دهد.

از محدودیت‌های روش می‌توان به عدم ردیابی فعالیت‌های درون مدل خانواده فرایندهای حرفه به ویژگی‌های درون مدل ویژگی اشاره کرد که با توجه به پرکاربرد بودن مدل ویژگی لازم به نظر می‌رسد. همچنین تعدّد مجموعه‌های اصلی و در نتیجه توابع هدف اشاره کرد که در صورت افزایش بسیار زیاد ممکن است روی نتایج، اثر منفی داشته باشد؛ این مورد در مطالعات آینده مورد بررسی بیشتری قرار خواهد گرفت. اگرچه از مطالعات موردی انتخابی در این مرحله تنها برای اعتبارسنجی درونی روش استفاده کردیم، اما همگی برگرفته از صنعت هستند و در آینده سعی خواهیم کرد برای ارزیابی اعتبار بیرونی این روش با انجام مطالعات موردی صنعتی در فضای واقعی اقدام کنیم.

مرجع‌ها

[1] J. Lee,D. Muthig and M. Naab, "A Feature-Oriented Approach for Developing Reusable Product Line Assets of Service-Based Systems", The Journal of Systems and Software, 2010, vol. 83, no. 7, pp. 1123-1136.

[2] T. Erl,"Service-Oriented Architecture: Concepts, Technology, and Design", Prentice Hall,2005.

[3] M. Abu-Matar, and H. Gomaa, "Feature Based Variability for Service Oriented Architectures", IEEE WICSA, Australia, 2011, pp.302-309.

[4] C. Sun, T. Xue, and M. Aiello, "ValySeC - A Variability Analysis Tool for Service Compositions Using VxBPEL", IEEE APSCC, China, 2010, pp. 307-314.

[5] M. Koning, C. Sun, M. Sinnema, and P. Avgeriou, "VxBPEL: Supporting Variability for Web Services in BPEL", Information and Software Technology, Elsevier, 2009, vol.5, no.1, pp. 258-269.

[6] J.Park, M. Moon, and K. Yeom, "Variability Modeling to Develop Flexible Service-Oriented Application", Journal of Systems Science and Systems Engineering, 2011,Vol.20, no.2,pp.193-216.

[7] M. Moon, M. Hong, and K. Yeom, "Two-Level Variability Analysis for Business Process with Reusability and Extensibility", IEEE International Computer Software and Applications Conference, 2008, pp.263-270.

[8] R.Capilla, N.Topaloglu,"Product Lines for Supporting the Composition and Evolution of Service Oriented Applications",IEEE IWPSE, Portugal,2005,pp53-56.

[9] S. Adam, J. Doerr, "The Role of Service Abstraction and Service Variability and its Impact on Requirements Engineering for Service Oriented Systems", IEEE COMPSAC, Finland, 2008, pp. 631-634.

[10] S. Apel, C. Kastner, and C. Lengauer, "Research challenges in the tension between features and services", ACM SDSOA, Germany, 2008, pp.53-58.

[11] J. Lee, K. Kang, D. Muthig, and M. Naab, "An Approach for Developing Service Oriented Product Lines", IEEE SPLC, Ireland, 2008, pp.275-284.

[12] Hadayatullah, K. Koskimies and T. Systa, "Using Model Customization for Variability Management in Service Compositions", ICWS, U.S., 2009, pp687-694.

[13] R. Mietzner, A. Metzger, F. Leymann, and K. Pohl, "Variability Modeling to Support Customization and Deployment of Multi-Tenant-Aware Software as a Service Applications", IEEE PESOS, Canada, 2009, pp. 18-25.

[14] T. Vale et al., "A Study on Service identification Methods for Product Lines", SPLC 2012, Brazil, 2012.

[15] F. Chen, S., Li, and W. C. C. Chu, "Feature analysis for service-oriented reengineering". Asia-Pacific Software Engineering Conference (APSEC), IEEE Computer Society, 2005, pp. 201-208.

[16] D. Kang, C. Y., Song, and D. K. Baik, "A method of service identification for product line". International Conference on Communications and Information Technology, IEEE Computer Society., 2008, pp.1040-1045.

[17] S. Kim, M. Kim, and S. Park, "Service identification using goal and scenario in service oriented architecture", Asia-Pacific Software Engineering Conference (APSEC), 2008, pp. 419-426.

[18] J. Lee, D. Muthig, and M. Naab, "An approach for developing service oriented product lines". International Software Product Line Conference (SPLC), 2008, pp. 275-284.

[19] T.E. Colanzi," Search based design of software product lines architectures", ICSE12, Switzerland, 2012, pp. 1507-1510.

[20] M. Kim, S. Kim, "Service Identification Using Goal and Scenario in Service Oriented Architecture," IEEE APSCC08, Taiwan, 2008, pp. 419-426.

[21] L. Ackerman, et al., "Building SOA Solutions Using the Rational SDP," IBM Corp, 2007.

[22] S. Ghosh, et al., "SOMA: A method for developing service-oriented solutions," IBM System Journal, vol. 47, no.3, 2008, pp. 377-396.

[23] G. K. Behara, S. Inaganti, S, "Service Identification: BPM and SOA Handshake," Business Process Trends, 2007.

[24] N. Zhou, Y. Zhu, Y, H. Wang, Q. Ma, "Evaluating Service Identification with Design Metrics on Business Process Decomposition," IEEE SCC, India, 2009, pp. 160–167.

[25] H. Jain, H. Zhao, N. R. Chinta, "A Spanning Tree Based Approach to Identifying Web Services," International Journal of Web Services Research, 2004, pp. 272-277.

[26] P. Jamshidi, M. Sharifi, S. Mansour, "To Establish Enterprise Service Model from Enterprise Business Model", Proc. of IEEE International Conference on Services Computing, Honolulu, 2008, pp.93-100.

[27] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm:NSGA-II", IEEE Transactions on Evolutionary Computation, 2008, Vol. 6, No.,2,pp.182-197.

[28] G. B. Lamont, V. Veldhuizen, D. A. C. Coello, "Evolutionary Algorithms for Solving Multi-Objective Problems," Second Edition, Springer, 2007.

[34] P. Pietu. “Feature Oriented Domain Analysis Expressions”, Nordic Workshop on Model Driven Software Engineering (NW-MoDE'08), Reykjavik, Iceland, 2008.

[35] V. Basili, G. Caldiera, and H.D. Rombach, “The Goal Question Metric Approach”, Encyclopedia of Software Engineering, vol.2, John Wiley & Sons, pp. 528-532, 1994.

[36] A. Kazemi, H. Haghighi, and F. Shams, “ABSIM: An Automated Business Service Identification Method”, International Journal of Software Engineering and Knowledge Engineering Vol.23, No.09, pp. 1303-1342, 2013.

[37] Nebro, A. J., Durillo, J. J., “jMetal 3.0 User Manual”, 2010. http://*jmetal.sourceforge.net*.

[29] C. Berger, et al., “Product Line Metrics for Legacy Software in Practice”, SPLC, South Korea, 2010, Volume 2, pp. 247- 250.

[30] M. Perepletchikov, C. Ryan and Z. Tari, “The Impact of Service Cohesion on the Analyzability of Service-Oriented Software”, vol.3, no.2, 2010, pp.89-103.

[31] I.Behattacharya,L. Getoor, “Iterative Record Linkage for Cleaning and Integration”, DMKD'04, France, 2004.

[32] Software Product Line Online Tools (SPLOT), at: *http://www.splot-research.org*.

[33] J.E.Hopcroft and JD. Ullman, “Introduction to automata theory, languages, and computation”, Addison Wesley, 1979.