



مرکز توسعه فناوری نیرو (مئن)



انجمن کامپیوتر ایران
Computer Society of Iran

ارائه روشی خودکار به منظور شناسایی سرویس های نرم افزاری

فریدون شمس
دانشگاه شهید بهشتی

دانشکده مهندسی برق و کامپیوتر
گروه پژوهشی اتوماسیون مهندسی نرم افزار
f_shams@sbu.ac.ir

پویان جمشیدی
دانشگاه شهید بهشتی

دانشکده مهندسی برق و کامپیوتر
گروه پژوهشی اتوماسیون مهندسی نرم افزار
p_jamshidi@sbu.ac.ir

معماری نه تنها محصولات و استانداردهای مربوطه را در بر داشته، بلکه به سایر جنبه های توسعه سیستم، نظیر تحلیل و طراحی راه حل نیز مرتبط است. موضوع اصلی این مقاله دربرگیرنده مراحل اولیه چرخه حیات راه حل مبتنی بر سرویس، که تحلیل و طراحی سرویس ها و یا مدلسازی سرویس ها [۳] نامیده میشود، میباشد.

تاکنون فعالیتهای تحقیقاتی و کاربردی زیادی جهت ارائه فناوری ها، محصولات و استانداردهای محاسبات مبتنی بر سرویس انجام شده است. ولی کارهای گزارش شده بسیار محدودی در زمینه های تحلیل و طراحی سرویسها موجود میباشد [۲] که همین کارهای محدود نیز دربرگیرنده مشکلاتی هستند [۴،۵]. روشهای ارائه شده در این حوزه قابلیت های خودکار سازی و بکار گیری شاخصهای کیفیتی در سطوح مدیریتی و فنی را نداشته، و مدلهای بنیادین و رسمی در این زمینه ارائه نشده، بنابراین بکارگیری آنها در سطوح وسیع نظیر سازمانها باعث کاهش کیفیت راه حل، عدم یکپارچگی در سطح معماری و کاهش سطح بهره وری تولید آن خواهد شد. با توجه به کاستی های ذکر شده این پژوهش مبتنی بر هدف زیر پایه گذاری شده است:

«ارائه یک روش به نام ASIM (Automated Service Identification Method) جهت شناسایی خودکار سرویسها به عنوان جزء اصلی معماری با توجه به شاخص های کیفیتی تبیین شده و نیز ارائه مدلی جهت تسهیل تصمیم گیری در سطح معماری به منظور خودکار سازی»

یکی از تصمیمات اصلی سبک معماری مبتنی بر سرویس، شناسایی اجزاء آن از مدل کاری با توجه به یکسری از معیارهای تبیین شده است [۲]. اگرچه شباهتهایی مابین مولفه ها و سرویس های نرم افزاری وجود دارد، اما استفاده از روشهای ارائه شده در حوزه مولفه ها جهت شناسایی سرویس ها به دلیل تعدد و قدمت آنها باعث طراحی غیر بهینه، کارکرد ضعیف، کاهش مقیاس پذیری و ارتباطات پیچیده بین

چکیده: یکی از فعالیتهای کلیدی به منظور ساخت راه حل های مبتنی بر سرویس، شناسایی اجزاء معماری با توجه به شاخص های کیفیتی است. به منظور کارآمد نمودن راه حلهای مبتنی بر سرویس، انتخاب روشی مناسب جهت شناسایی سرویس ها از مدلهای کاری یک سازمان بسیار حیاتی تلقی می شود. روشهای کنونی در این محدوده بدون مد نظر داشتن شاخص های کیفیتی و یکپارچگی معنایی در سطح سرویس ها، بیشتر بر روی شناسایی آنها از موجودیتهای کاری پرداخته و فرآیندهای سازمانی را مد نظر قرار نمی دهند. همچنین این روشها نیاز به تجربه و هوش انسانی داشته و قابلیت مکانیزه شدن را ندارند. این مقاله به ارائه یک روش نوین به نام ASIM جهت شناسایی خودکار سرویس های نرم افزاری از روی مدل های کاری سازمانی میپردازد. در این روش، شناسایی سرویس ها از طریق بهینه نمودن تابع هدف متشکل از شاخص های دانه بندی، انسجام، پیوستگی، استفاده مجدد، قابلیت نگهداری و انطباق با سازمان، انجام می پذیرد.

واژه های کلیدی: تحلیل و طراحی سرویس گرا، مدلسازی سرویس گرا، مدل سرویسهای سازمانی، مدل کاری سازمانی، شناسایی سرویس.

۱. مقدمه

به منظور توسعه سیستمهای کاربردی سازمانی سطح انتزاع اجزاء معماری میبایست به اندازه سطح انتزاع دامنه کاری سازمان ارتقاء یافته باشد [۱]. رویکردهای سنتی شیء گرای و مبتنی بر مولفه برای ساخت چنین سیستمهایی ناکافی بوده و منجر به ایجاد رویکرد نوین مبتنی بر سرویس که از سطح انتزاع بالاتری برخوردار می باشد، شده است [۲،۳]. معماری مبتنی بر سرویس به عنوان یکی از سبک های معماری [۴] پیشرو در راه حلهای سازمانی نیز مطرح میباشد. این سبک

سرویسها میگردد [۶،۷]. مدلی که سرویسها از روی آن به طور خودکار شناسایی میگردد میبایست در سطحی باشد که تصمیمات معماری به صورت خودکار اخذ و در مدل مقصد اعمال گردد، همچنین تعریف معیارهای کمی روی آن امکان پذیر باشد. به علاوه این مدل میبایست معادل با مدل کاری که با سبک های مختلفی ایجاد میگردد، باشد. مدلسازی میتواند منجر به تسهیل ساخت راه حل مبتنی بر سرویس از نیازمندی های کاری و ایجاد ارتباط مستحکمتر بین آنها گردد [۵]. توسعه مبتنی بر مدل، رویکردی شامل کاربرد مدل ها و فناوری های مربوطه جهت ارتقاء سطح انتزاع به منظور خودکار سازی فعالیتهای مختلف چرخه حیات نرم افزار میباشد. این مقاله به عنوان اولین قدم در راستای توسعه مبتنی بر مدل کاربردهای سرویس گرا با رویکرد خودکار سازی فعالیتهای ابتدایی چرخه حیات راه حل ارائه می گردد. سهم علمی این مقاله شامل ارائه یک مدل ماتریسی که پیوند دهنده اجزاء کلیدی مدل کاری سازمانی بوده و نیز روشی جهت شناسایی خودکار سرویس ها با تعریف الگوریتمی بر روی این مدل می باشد. در بخش ۲ این مقاله، پژوهشهای مرتبط انجام شده با اهداف ترسیم شده تحلیل گشته است. در قسمت ۳ مفاهیم بنیادین مورد استفاده در حوزه تحلیل و طراحی سرویس گرا ارائه گشته است. روش ASIM، در بخش چهارم این مقاله معرفی گشته و در ادامه کاربردهای آن در بخش بعدی ارائه شده است. اعتبار سنجی روش به همراه نتیجه گیری و اقدامات آتی پیشبینی شده در خاتمه ذکر گردیده است.

۲. مرور ادبیات و پژوهشهای پیشین

امسدن [۵] بر روی جمع آوری نیازمندی ها، ساخت مدل سرویس انعکاس دهنده نیازمندی ها، همچنین ساخت و استقرار راه حل محقق کننده مدل طراحی پرداخته است. در این پژوهش به تکنیکهای شناسایی سرویسها و ارزیابی مدل سرویس [۸-۱۰] پرداخته نشده و صرفاً به توصیف مدل سرویس و اجزاء آن اشاره شده است. اگرچه این کار به ترویج رویکرد مبتنی بر مدل برای ساخت راه حل پرداخته، ولی به یک رویکرد نیمه خودکار بسنده کرده است. تنها مرحله ای که کاملاً به صورت خودکار در آمده، انتقال مدل سرویس به مدل پیاده سازی و در مرحله انتقال مدل کاری به مدل سرویس نیاز به عامل انسانی می باشد.

هابرز [۷] ده روش عملی شناسایی سرویس را ارائه داده است. هر یک از آنها با مزایا و معایب آن تدوین گشته است. اگرچه اکثر این روشها مبتنی بر مدل میباشد، ولی به دلیل عدم معرفی شاخصهای اندازه گیری، قابلیت خودکار سازی نداشته و با دخالت عامل انسانی مواجه می شوند.

اینگانتی [۱۱] با معرفی روش شناسایی سرویسهای سازمانی بر پایه تحلیل زنجیره ارزش، روشی کاربردی در سطح سازمان ارائه داده است. در این روش نیز با وجود اینکه بر روی اعتبار سنجی سرویسهای شناسایی شده تاکید گردیده، ولی بدلیل عدم بکارگیری شاخصهای کمی روشی خودکار ارائه نشده است.

زنگ [۱۲] روشی مبتنی بر بازمهندسی معماری راه حل ارائه داده، به طوری که سرویسها از طریق بسته بندی شدن در مدل دامنه شناسایی میگردند. این روش عملاً هیچ کدام از ویژگی های سه گانه را دارا نیست.

مروری اجمالی بر روی متدولوژی های تحلیل و طراحی سرویس گرا توسط زیرمن [۱] ارائه شده است. در این کار یک چارچوب شناسایی سرویس با محوریت مدل های ساختاری و رفتاری زبان مدلسازی UML ارائه گشته است. اگرچه امکان تبدیل مدل کاری به مدل سرویس عنوان شده، ولی به صورت تصریح شده ارائه نگردیده است.

قابلیتهای خودکار سازی، بکارگیری شاخص های کیفی و استفاده از اصول مبتنی بر مدل مهمترین ویژگی های روش ارائه شده در مقاله حاضر هستند. اگرچه فعالیت شناسایی سرویس نقش پررنگی در چرخه حیات راه حل دارد ولی تمامی کارهای ارائه شده سه ویژگی یاد شده را ندارد [۱۳-۱۷]. اکثر آنها قابلیت خودکار سازی نداشته و وابسته به عامل انسانی معمار نرم افزار میباشند. تمامی آنها تنها به قسمتی از شاخص های کیفی موجود توجه داشته که باعث کارا نبودن مولفه های شناسایی شده میگردند. همچنین کارهای یاد شده از مدل های کاری مختلف با استانداردها و زبانهای مدلسازی متفاوت جهت مدل ورودی استفاده کرده که به دلیل عدم یکپارچگی باعث عدم یکپارچگی روشهای یاد شده میگردند. مقاله حاضر با بکارگیری تمامی سه ویژگی یاد شده، به ارائه یک روش خودکار شناسایی سرویس بر پایه شاخص های دانه بندی، انسجام، پیوستگی، استفاده مجدد، قابلیت نگهداری و انطباق با سازمان با بکارگیری مدل یکپارچه ورودی خواهد پرداخت.

۳. تعاریف و مفاهیم اصلی

در این بخش با ارائه تعاریف و مفاهیم پایه ای که به صورت رسمی ارائه گشته، سعی شده تا پایه ای سازگار جهت سایر بخش های این پژوهش بنا نهاده شود.

تعریف ۱: مدل کاری سازمانی [۱۸] به صورت $EBM = \{EBP, EDM\}$ تعریف گشته، به طوری که EBP مجموعه فرآیندهای کاری و EDM مجموعه موجودیتهای کاری را در بر می گیرد.

به منظور استفاده از مدل کاری در روش پیشنهادی، میبایست پایین ترین سطح هر یک از دو بعد که در سطح مفهومی قابل تعریف باشد،

پیامهای تشکیل دهنده ساختار آن و RS مجموعه روابط بین این سرویس و سایر سرویسها میباشد. همچنین $EBP \subset I$ و $BE \subset Msg$.

تعریف ۷:

اگر داشته باشیم:

$$Cluster_x = \{(EBP_i, BE_j), i=1 \dots h1, j=1 \dots h2\},$$

$$Cluster_y = \{(EBP_{i'}, BE_{j'}), i'=1' \dots h'1, j'=1' \dots h'2\},$$

و M یک ماتریس CRUD و S یک سرویس نرم افزاری باشد، آنگاه:

- $\forall x, y \text{ Cluster}_x \cap \text{Cluster}_y = \{\}$
- $\forall x, y \text{ if } l'1 = h1, l'2 = h2, \text{ Cluster}_x \cup \text{Cluster}_y = \{(EBP_i, BE_j), i=1 \dots h'1, j=1 \dots h'2\}$
- اگر M در قالب n خوشه، خوشه بندی شده باشد، آنگاه:

$$\bigcup_{x=1}^n \text{Cluster}_x = M$$

$$\forall \theta \exists x \text{ Cluster}_x = S$$

مدل سرویس های سازمانی، مدل پایه ای راه حل های مبتنی بر سرویس است که به عنوان فرآورده ورودی فعالیتهای طراحی و پیاده سازی راه حل بکار گرفته میشود. این مدل به صورت مدل انتزاعی راه حل پیاده سازی شده به عنوان الگویی جهت پیاده سازی چندین راه حل مختلف مبتنی بر سرویس بکار میرود و به عنوان مدلی ترکیبی شامل مجموعه سرویس ها، سرویس دهندگان، مشخصه های سرویس، افرازگرها، پیامها، و تعاملات و ارتباطات مابین آنها میباشد [۲۲، ۲۳].

تعریف ۸: مدل سرویسهای سازمانی به صورت زیر تعریف میشود، بطوریکه S مجموعه سرویس ها، Pr مجموعه سرویس دهندگان، $Spec$ مجموعه مشخصه های سرویسها، Prt به عنوان مجموعه افرازها، Msg به عنوان مجموعه پیامها، و R به عنوان مجموعه روابط بین این اجزاء میباشد.

$$ESM = \{S, Pr, Spec, Prt, Msg, R\}$$

۴. روش پیشنهادی شناسایی سرویس

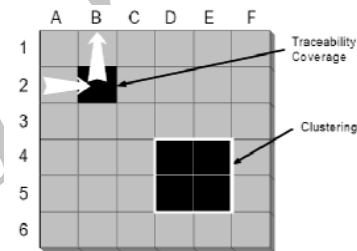
مطابق تعریف ۸، مجموعه S شامل سرویس های نامزد، به عنوان جزء اصلی مدل سرویسهای سازمانی، در مرحله شناسایی بدست آمده، و مجموعه سرویس دهندگان، مجموعه مشخصه های سرویس ها، و مجموعه افرازها، مجموعه پیامها، مجموعه مولفه های سرویس، و همچنین مجموعه روابط بین این اجزاء و سایر اجزاء مدل سرویس، در مرحله مشخصه سازی تعریف میگردند.

در این پژوهش، جهت تسهیل خودکار سازی تصمیم گیری در سطح معماری راه حل، به خصوص خودکار سازی فرآیند شناسایی سرویس، مدلی محاسباتی مبتنی بر ماتریس عینی تر از مدل کاری سازمانی و

اتخاذ گردد. بدین منظور "فعالیتی که توسط یک نقش در یک مکان و زمان مشخص انجام میپذیرد و ارزش افزوده مشخصی تولید نماید" به عنوان پایین ترین سطح فرآیند کاری، فرآیند کاری پایه ای نامیده میشود. به علاوه موجودیت کاری به عنوان پایین ترین سطح مدل دامنه میبایست تنها بوسیله یک فرآیند کاری پایه ای ایجاد گردد و سایر فرآیندها صرفاً از آن استفاده نمایند [۱۹].

تعریف ۲: یک موجودیت کاری به صورت $BE = \{n, A, R\}$ تعریف میشود، بطوریکه n نام آن، A مجموعه ویژگی های آن و R مجموعه روابط بین این موجودیت با سایر موجودیتهای مدل دامنه میباشد.

تعریف ۳: یک فرآیند کاری پایه ای به صورت $EBP = \{n, (BE_j, sr)\}$ تعریف می شود، به طوریکه n نام آن، BE_j به عنوان ز امین موجودیت کاری که ارتباط معنایی با فرآیند کاری پایه ای مربوطه دارد و $\{ "C", "R", "U", "D" \} \in sr$ به عنوان نوع رابطه معنایی مطرح است.



شکل ۱. تحلیل ماتریسی [20]

مدل ماتریسی جهت نمایش ارتباط بین دو محور متفاوت مناسب میباشد. همچنین علاوه بر این ویژگی میتوان بر روی آن تحلیلهای محاسباتی و عملیات بررسی متقابل را انجام داد [۲۰] (شکل ۱). به هر حال، در این مقاله تمرکز اصلی بر روی عملیات خوشه بندی است. یک خوشه در ماتریس نمایش دهنده ارتباط نزدیک مابین عناصر سطرها و ستونهای آن خوشه بوده و ماهیت آن خوشه بستگی به ماهیت عناصر سطری و ستونی دارد. در این پژوهش فرآیندهای کاری پایه ای در سطر و موجودیتهای کاری در ستون ماتریس قرار میگیرند.

تعریف ۴: ماتریس CRUD به صورت زیر تعریف می شود، به طوریکه عناصر معنایی "C" به صورت قطری [۲۱] در آن واقع گشته است.

$$M = \{(EBP_i, BE_j) \mid i=1 \dots \#row, j=1 \dots \#column\}$$

تعریف ۵: خوشه به صورت زیر تعریف میگردد، به طوریکه $\#column+1 \leq l2 < h2 \leq 1$ و $\#row+1 \leq l1 < h1 \leq 1$

$$Cluster = \{(EBP_i, BE_j), i=1 \dots h1, j=1 \dots h2\}$$

تعریف ۶: یک سرویس نرم افزاری به صورت $S = \{n, I, Msg, RS\}$ تعریف میشود، به طوریکه n نام آن، I نمای رفتاری آن، Msg مجموعه

به منظور بهبود فرآیند توسعه نرم افزار، شاخص های مختلفی تعریف و در سطوح عملیاتی به کار برده شده است. شاخصهای مختلف در سطح مدیریت فرآیند توسعه نظیر هزینه استفاده مجدد و در سطح فنی نظیر انسجام و پیوستگی در پژوهشهای پیشین مطرح گشته است [۲۳-۲۸]. در این پژوهش، با ارائه شاخص های کیفی (جدول ۱) به عنوان معیاری جهت بهینه نمودن راه حل مبتنی بر سرویس گامی در جهت بهبود فرآیندهای چرخه حیات راه حل های مبتنی بر سرویس برداشته شده است. این شاخص ها بر اساس تحقیقات گذشته در حوزه های سرویس گرایی و مبتنی بر مولفه و همچنین تجربیات بدست آمده از پروژه های سرویس گرایی که توسط نگارندگان این مقاله اجرا گشته، حاصل شده است.

جدول ۱. شاخص های کیفیتی

حد مطلوب	فرمول	معنا	نماد	عامل / شاخص
-	$\sum_{j=1}^{h2} \sum_{i=1}^{h1} SR_{ij}$	عامل نزدیکی	Aff(S)	Affinity
بیشینه	G(S)	سهم سرویس در ساخت سیستم کاربردی	RE(S)	Reuse Efficiency
بیشینه	Cohesion(S) / Coupling(S)	سهولت تغییرات سرویس متناسب با نیازمندی های جدید	M(S)	Maintainability
بیشینه	$G(S) = \alpha * (h1 - l1), \alpha = Aff(S) / ((h1 - l1) * (h2 - l2))$	دانه بندی سرویس	G(S)	Granularity
بیشینه	G(S)	سهم سرویس در محدوده کاری سازمان	BV(S)	Business value
بیشینه	Aff(S)	نزدیکی معنایی مابین عملیات سرویس	Cohesion(S)	Cohesion
کمینه	$1 + \sum_{i=1}^{h1} \sum_{j=1}^{\#column} SR_{ij} - Aff(S)$	نزدیکی معنایی بین عملیات بین سرویس های مختلف	Coupling(S)	Coupling

به منظور استفاده از شاخص های کیفی در فرآیند شناسایی سرویس ها، قاعده شاخص میبایست بر اساس اجزاء مدل ماتریس CRUD تعریف گردد. هر یک از قواعد مربوطه معرفی تابعی (جدول ۱) است که مقدار مناسب شاخص را بر اساس قواعد تجربی محاسبه مینماید. بنابراین، هر یک از آن ها به عنوان تابعی از فضای مدل کاری سازمانی به اعداد حقیقی مثبت میباشند (شکل ۴ و جدول ۲).

	BE1	BE2	BE3	BE4	BE5	BE6
EBP1	1	0.75			0.75	
EBP2	0.25	0.25				
EBP3			1	0.75		0.25
EBP4			0.25	1		
EBP5			0.75			
EBP6			0.75	0.25		
EBP7					1	
EBP8					0.75	0.25
EBP9		0.25				0.75

انتزاعی تر از مدل سرویسهای سازمانی تعریف میگردد. این ماتریس که از آن با نام ماتریس CRUD نام برده میشود، در شکل ۲ نشان داده شده است. سطرهای این ماتریس را فرآیندهای کاری پایه ای، ستون های آنرا موجودیتهای کاری پایه ای، و خانه های آنرا روابط معنایی (C, D, R, U) تشکیل میدهند که به ترتیب اولویت معنایی آنها به این صورت تعبیر میگردد:

- نشان "C" به معنی ساختن موجودیت کاری پایه ای توسط فرآیند کاری پایه ای مربوطه است.
- نشان "U" به معنی به روز رسانی موجودیت کاری پایه ای توسط فرآیند کاری پایه ای مربوطه است.
- نشان "D" به معنی حذف موجودیت کاری پایه ای توسط فرآیند کاری پایه ای مربوطه است.
- نشان "R" به معنی استفاده از موجودیت کاری پایه ای توسط فرآیند کاری پایه ای مربوطه است.

	BE1	BE2	BE3	BE4	BE5	BE6
EBP1	C	U			U	
EBP2	R	R				
EBP3			C	U		R
EBP4			R	C		
EBP5			U			
EBP6			U	R		
EBP7					C	
EBP8					U	R
EBP9		R				U

شکل ۲. ماتریس CRUD

به منظور انجام محاسبات بر روی مدل ماتریسی میبایست این مدل از وضعیت مبتنی بر برچسب به وضعیت مبتنی بر مقدار تغییر یابد. بنابراین، برچسبها متناسب با اولویتشان با مقادیری حقیقی بین صفر و یک جایگزین میگرددند ($0 \leq R \leq D \leq U \leq C \leq 1$). هر قدر عدد مربوطه بزرگتر باشد نشان دهنده ارتباط قویتر مابین فرآیند کاری پایه ای و موجودیت مربوطه میباشد. در این مقاله به دلیل هماهنگ سازی بین مثالها و ساده سازی محاسبات از جایگزینی ($C=1, U=0.75, D=0.5, R=0.25$) همانند شکل ۳ استفاده میگردد.

	BE1	BE2	BE3	BE4	BE5	BE6
EBP1	1	0.75			0.75	
EBP2	0.25	0.25				
EBP3			1	0.75		0.25
EBP4			0.25	1		
EBP5			0.75			
EBP6			0.75	0.25		
EBP7					1	
EBP8					0.75	0.25
EBP9		0.25				0.75

شکل ۳. ماتریس شکل ۲ با جایگزینی عددی نمادها

۴.۱. شاخص های کیفیتی

شکل ۴. ماتریس خوشه بندی شده

جدول ۲. مقادیر شاخص های کیفی S

شاخص / سرویس	Aff(S)	G(S)	Cohesion(S)	Coupling(S)	M(S)	BV(S)	RE(S)
S1	2.25	1.125	2.25	1.75	1.28	1.125	1.125
S2	4.75	2.375	4.75	1.25	3.8	2.375	2.375
S3	2.75	1.375	2.75	1.25	2.2	1.375	1.375

۴.۲. توابع هدف و امید

شاخص های کیفی مطرح شده رابطه ای متقابل با یکدیگر دارند به طوری که همه آنها در یک زمان نمیتوانند به مقدار بهینه برسند [۲۵]. به طور مثال سرویسهای با دانه بندی درشت کارایی بالاتری در میزان قابلیت استفاده دارند ولی از قابلیت استفاده مجدد آنها کاسته میشود. بنابراین مساله شناسایی سرویس یک مساله بهینه سازی چند هدفه میباشد. فرض نماییم مجموعه ای از n سرویس از مدل کاری سازمانی شناسایی شده است. تابع هدف Z با دامنه مجموعه سرویس ها و برد فضای حقیقی مثبت بدین صورت تعریف میگردد:

$$Z(\text{Service Set}) = \frac{\sum_{i=1}^n RE(S_i) * \sum_{i=1}^n M(S_i) * \sum_{i=1}^n BV(S_i) * \sum_{i=1}^n Cohesion(S_i) * \sum_{i=1}^n G(S_i)}{\sum_{i=1}^n Coupling(S_i)}$$

$$= \frac{\sum_{i=1}^n G(S_i) * \sum_{i=1}^n Cohesion(S_i) * \sum_{i=1}^n G(S_i) * \sum_{i=1}^n Cohesion(S_i) * \sum_{i=1}^n G(S_i)}{\sum_{i=1}^n Coupling(S_i) * \sum_{i=1}^n Coupling(S_i)}$$

$$= \frac{\sum_{i=1}^n G(S_i)^3 * \sum_{i=1}^n Cohesion(S_i)^2}{\sum_{i=1}^n Coupling(S_i)^2}$$

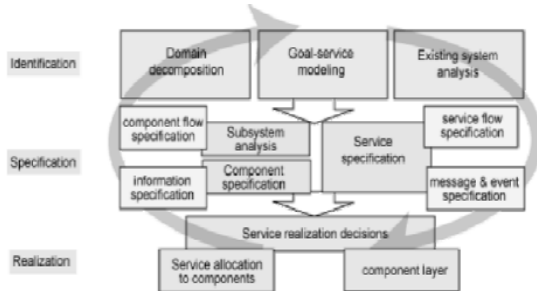
همچنین تابع امید $Pr(S)$ جهت اندازه گیری احتمال شناسایی سرویسی با تابع هدف مناسب، با فرمول ذیل تعریف میگردد.

$$Pr(S) = 1 - \frac{\sum_{i=h1}^{\#row} \sum_{j=2}^{h2} sr_{ij}}{\sum_{i=h1}^{\#row} \sum_{j=2}^{h2} 1}$$

در میان این شاخص ها، دانه بندی به عنوان مهمترین شاخص مطرح است به طوری که اهمیت و تاثیر آن بر کارایی راه حل ها توسط محققین اعلان گشته است. با رشد مهندسی نرم افزار، دانه بندی اجزاء الگوهای مختلف به منظور توسعه راه حل های نرم افزاری به تدریج از دانه بندی ریز به دانه بندی درشت تغییر نموده و به طوری که این رویکرد به عنوان مهمترین موضوع در مهندسی نرم افزار مطرح بوده است [۱۷].

۴.۳. فعالیت شناسایی سرویس

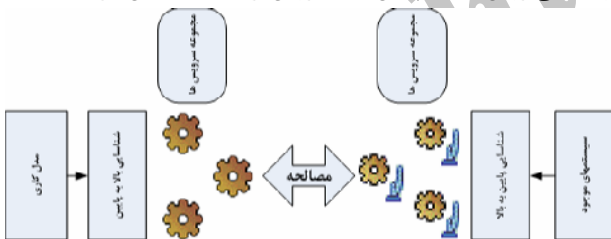
مهمترین مراحل تحلیل و طراحی سرویس گرا (مدلسازی سرویس گرا) مطابق چارچوب SOMA [۲] شامل شناسایی، مشخصه سازی و محقق سازی (شکل ۵) میباشد.



شکل ۵. فرآیندهای شناسایی، مشخصه سازی و محقق سازی در

چارچوب SOMA [۲]

شناسایی سرویس (همانند شناسایی کلاس و مولفه در معماری های مربوطه) اولین و چالش برانگیزترین مرحله در این چارچوب است. بر اساس این چارچوب، رویکردهای مختلفی (شکل ۶) جهت شناسایی سرویس ها بکار گرفته میشود. در روش بالا به پایین با تجزیه نمودن عناصر کاری و بهبود آنها سرویس ها شناسایی میگردد. در روش پایین به بالا با تحلیل سیستمهای موجود، کارکردی که قابل ارائه به صورت سرویس باشد شناسایی میگردد [۹]. در روش مبتنی بر هدف که ترکیبی از دو روش پیشین میباشد، با مصالحه نیازمندی ها و دارایی های موجود، سرویس ها حاصل میگردد. در این کار پژوهشی، رویکرد اول یعنی روش بالا به پایین جهت روش ارائه شده تعیین گردیده است.



شکل ۶. رویکردهای مختلف شناسایی سرویس

شناسایی سرویس به عنوان فرآیندی جهت منتج کردن مجموعه سرویس های نامزد با بهینه نمودن شاخص های کیفیتی با توجه به رویکرد و راهبردی مشخص مطرح است. از آنجایی که این فرآیند به عنوان یکی از اولین فعالیتهای چرخه حیات راه حل میباشد، خطاهای ایجاد شده در این مرحله در سایر مراحل جاری میگردد [۲۲]. در این پژوهش مساله شناسایی سرویس بدین گونه تنظیم میگردد: "چگونه مجموعه سرویس های بهینه با شاخص های کیفیتی مطلوب از نیازمندی های کاری منتج می شود؟"

چرخه حیات نرم افزار است. معماری مبتنی بر مدل به عنوان الگویی مدل رانه در جهت تعبیه اطلاعات طراحی و وابستگی بر روی مدلها و تبدیلها مطرح شده است [۳۰]. نگاهی خودکار مدلها در سطوح مختلف مانند مدل های مستقل از محاسبات، مدلهای مستقل از بستر و مدلهای وابسته به بستر محاسباتی از جمله فعالیتهای این گونه معماری است. ایجاد مدلها با سطح انتزاع بالا و تبدیل خودکار آنها به راه حل نهایی از جمله اهداف توسعه مبتنی بر مدل محسوب میشود [۳۱]. این رویکرد به منظور توسعه راه حلهای مبتنی بر سرویس نیز قابل اجرا میباشد. مدلهای کاری که توسط تحلیلگر سیستم تهیه میگردد با این رویکرد میتواند به صورت خودکار به مدل سرویس تبدیل شود. این تبدیل با توجه به اینکه یک نگاشت، از مدل مستقل از محاسبات، به مدل مستقل از بستر است (شکل ۷)، با رویکرد معماری مبتنی بر مدل منطبق بوده و باعث افزایش کیفیت راه حل، کاهش هزینه کل مالکیت، یکپارچگی در سطح معماری و افزایش سطح بهره وری تولید آن خواهد شد.

۶. ارزیابی و اعتبار سنجی روش

در این بخش، اثربخشی تابع هدف Z که نقش اصلی را در روش شناسایی سرویس ارائه شده ایفا مینماید، بررسی میگردد. ماتریسهای M و M' (شکل ۸ و ۹) دو نمونه از ماتریسهایی است که به عنوان مطالعه موردی در این بخش مورد بررسی قرار میگیرند. جهت بررسی هر مورد، الگوریتم شناسایی با ۳ پارامتر (θ) متفاوت اجرا میگردد. خوشه بندیهای حاصل شده در جداول ۳ و ۵ و مقادیر متناظر تابع هدف و امید هر سرویس و نیز مقدار تابع هدف مجموعه سرویسها در جداول ۴ و ۶ آورده شده است.

M	BE1	BE2	BE3	BE4	BE5	BE6
EBP1	1	0.75			0.75	
EBP2	0.25	0.25				
EBP3			1	0.75		0.25
EBP4			0.25	1		
EBP5			0.75			
EBP6			0.75	0.25		
EBP7					1	
EBP8					0.75	0.25
EBP9		0.25				0.75

شکل ۸. مطالعه موردی ۱

جدول ۳. خوشه بندیهای مطالعه موردی ۱

خوشه بندی	خوشه ۱	خوشه ۲	خوشه ۳
M1	سطر: ۲-۱ ستون: ۲-۱	سطر: ۶-۳ ستون: ۴-۳	سطر: ۹-۷ ستون: ۶-۵
M2	سطر: ۳-۱ ستون: ۳-۱	سطر: ۶-۴ ستون: ۴-۴	سطر: ۹-۷ ستون: ۶-۵
M3	سطر: ۶-۱ ستون: ۴-۱	سطر: ۹-۷ ستون: ۶-۵	ندارد

جهت رسیدن به راه حلی برای مساله ذکر شده میبایست تابع هدف Z پیشینه گردد. بنابراین مساله شناسایی سرویس بدین گونه فرمول بندی میگردد:

SI: EBM → ESM

$$S = \{S_1, S_2, \dots, S_n\}$$

$$\text{Max } Z(S), Z =$$

$$\frac{\sum_{i=1}^n G(S_i)^3 * \sum_{i=1}^n \text{Cohesion}(S_i)^2}{\sum_{i=1}^n \text{Coupling}(S_i)^2}$$

۴.۴. الگوریتم شناسایی سرویس

به منظور حل مساله شناسایی سرویس، الگوریتم ذیل جهت خودکارسازی فرآیند حل مساله ارائه میگردد. این الگوریتم مدل کاری که به فرم مدل ماتریسی درآمده را به عنوان ورودی گرفته و با توجه به تابع هدف Z که در آن شاخصهای کیفیتی لحاظ گشته، آنرا خوشه بندی کرده و به عنوان مجموعه سرویسهای سازمانی منتج میسازد.

Algorithm 1: identification of services from CRUD matrix

Input:

- M: CRUD matrix
- θ_1 : Objective threshold
- θ_2 : Prospect threshold ($0 \leq \theta_2 \leq 1$)

Output:

- $S = \{S_1, S_2, \dots, S_n\}$

Begin

ServiceSet: = {}; i: =1;

FirstLocation = (1, 1);

NC: = NumberOfColumns (M);

While FirstLocation[2] <= NC

S_i (Start): = FirstLocation;

S_i (End): = FindTheNextCreateLocation();

If $Z(\{S_{ij}\}) > \theta_1$ and $Pr(\{S_{ij}\}) > \theta_2$ Then

ServiceSet: = ServiceSet + $\{S_i\}$;

FirstLocation: = S_i (End);

i: = i+1;

End If

End While

End

۵. کاربردهای روش پیشنهادی

در مهندسی نرم افزار، افزایش پیچیدگی محصولات، کوتاه بودن چرخه های توسعه، و انتظارات افزایش بهره وری و کیفیت، چالشهای اساسی را در تمامی مراحل چرخه حیات نرم افزار ایجاد کرده است. توسعه مبتنی بر مدل رویکردی در راستای پاسخگویی به این نیازها است [۲۹]. این رویکرد شامل کاربرد مدلها و فناوریهای مربوطه در جهت افزایش سطح انتزاع به منظور خودکارسازی فعالیتهای موجود در

جدول ۴. نتایج مطالعه موردی ۱

	Z1	Z2	Z3	Pr1	Pr2	Pr3	Z
M1	2.35	193.44	12.58	0.99	1	1	609.71
M2	1.5	0.41	12.58	0.89	1	1	58.25
M3	65.53	12.58		0.98	1		273.33

M'	BE1	BE2	BE3	BE4	BE5	BE6
EBP1	1					
EBP2	0.5					
EBP3		0.25				
EBP4	0.25					
EBP5	0.25					
EBP6	0.75					
EBP7	0.75					
EBP8	0.25	1				
EBP9						
EBP10	0.25		1			
EBP11		0.25		1	1	1

شکل ۹. مطالعه موردی ۲

جدول ۵. خوشه بندی های مطالعه موردی ۲

خوشه بندی	خوشه ۱	خوشه ۲	خوشه ۳
M'1	سطر: ۷-۱ ستون: ۱-۱	سطر: ۱۰-۸ ستون: ۳-۲	سطر: ۱۱-۱۱ ستون: ۶-۴
M'2	سطر: ۱۰-۱ ستون: ۳-۱	سطر: ۱۱-۱۱ ستون: ۶-۴	ندارد
M'3	سطر: ۹-۱ ستون: ۲-۱	سطر: ۱۱-۱۰ ستون: ۶-۳	ندارد

جدول ۶. نتایج بررسی موردی ۲

	Z1	Z2	Z3	Pr1	Pr2	Pr3	Z
M'1	336.14	1.77	5.76	0.88	0.88	1	751.28
M'2	353.21	5.76	-	0.91	1	-	495.42
M'3	390.62	7.11	-	0.88	1	-	555.66

به منظور بررسی روش ارائه شده، راه حل بهینه بدست آمده (با توجه به جداول ۴ و ۶) با نتایج حاصل از راه حل پیشنهادی ۵ خبره منتخب، مقایسه میگردد. افراد خبره همگی نقش های متفاوتی در چرخه حیات راه حل داشته و دارای تبحر در معماری سیستمهای سازمانی با انواع سبک های معماری به خصوص رویکرد مبتنی بر سرویس هستند. راه حل پیشنهادی از سوی هر یک از افراد خبره و راه حل بهینه منتج شده از طریق روش ASIM در جدول ۷ نمایش داده شده است. همانطور که مشاهده میگردد، راه حل بهینه بدست آمده از طریق روش پیشنهادی بسیار نزدیک به راه حل های افراد خبره بوده، به طوریکه در مثال اول ۳ فرد خبره عینا راه حل بهینه پیشنهادی، ۲ نفر هم دومین راه حل بهینه و در مثال دوم ۴ فرد راه حل بهینه، و ۱ فرد دومین راه حل بهینه را ارائه داده اند.

جدول ۷. مقایسه راه حل های بهینه با راه حل های پیشنهادی خبرگان

راه حل بهینه	راه حل خبره ۱	راه حل خبره ۲	راه حل خبره ۳	راه حل خبره ۴	راه حل خبره ۵	راه حل بهینه
M	M1	M1	M3	M1	M3	M1
M'	M'1	M'3	M'1	M'1	M'1	M'1

۷. نتیجه

در این مقاله، یک روش نوین جهت شناسایی سرویس های نرم افزاری به عنوان اصلی ترین جزء راه حل مبتنی بر سرویس ارائه شده است. این روش به عنوان مهمترین رکن تبدیل مدل کاری به مدل سرویس، نه تنها معیارهای کیفی نظیر دانه بندی، انسجام، پیوستگی، استفاده مجدد، قابلیت نگهداری و انطباق با سازمان را مدنظر دارد، بلکه یک فرآیند خودکار را ارائه داده است. فرآیند شناسایی از دو تابع هدف و امید که بر روی ماتریس معادل مدل کاری تعریف گشته استفاده مینماید. تابع هدف جهت اندازه گیری مناسب بودن سرویس با توجه به شاخص های تعریف شده و تابع امید جهت اندازه گیری امید شناسایی سرویس مناسب در ادامه کار فرآیند هستند. اعتبار سنجی روش از طریق مقایسه جواب بهینه حاصل شده از فرآیند ارائه شده با جواب های داده شده توسط افراد خبره انجام پذیرفته است.

فرآیندهای مشخصه سازی و محقق سازی، مراحل بعدی ساخت راه حل های مبتنی بر سرویس میباشند. ارائه روشهایی به منظور خودکار سازی این فرآیندها، توسعه ابزارهایی جهت پشتیبانی از آنها و تجمع آنها در قالب یک چارچوب مدل رانه جهت ایجاد خودکار مدل سرویس در سطح طراحی از فعالیتهای آتی این پژوهش خواهد بود. خودکار سازی فعالیتهای شناسایی، مشخصه سازی و مدلسازی اجزاء معماری، معمار نرم افزار را به تمرکز روی فعالیتهای مهمی نظیر اعتبار سنجی و تصمیم گیری مدلهای سوق داده و در نهایت باعث افزایش کیفیت راه حل، کاهش هزینه کل مالکیت، یکپارچگی در سطح معماری و افزایش سطح بهره وری تولید آن خواهد شد.

تقدیر و تشکر

این پژوهش با استفاده از اعتبارات پژوهشی دانشگاه شهید بهشتی به دانشکده مهندسی برق و کامپیوتر، گروه اتوماسیون مهندسی نرم افزار (<http://aser.sbu.ac.ir>) انجام شده است.

مراجع

- [1] Zimmermann, O., Schlimm, N., Waller, G., and Pestel, M., *Analysis and Design Techniques for Service-oriented Development and Integration*, IBM Deutschland, 2005.
- [2] Arsanjani, A., *Service-Oriented Modeling and Architecture (SOMA)*, IBM® developerWorks®, 2004.
- [3] Zimmermann, O., Krogh, P., and Gee, C., *Elements of Service-Oriented Analysis and Design*, IBM® developerWorks®, 2004.
- [4] Fielding, R.T., *Architectural Styles and the Design of Network based Software Architectures*, PhD Thesis, University of California, Irvine, 2000.
- [5] Amsden, J., *Modeling SOA: Part 1. Service identification*, IBM® developerWorks®, October 2007.

[23] Fayad, M.E., "Accomplishing Software Stability," *J. Commun. ACM*, Vol. 45, no. 1, pp. 111–115, 2002.

[24] Lee, J.K., Jung, S.J., and Kim, S.D., *Component Identification Method with Coupling and Cohesion*, Proc. of the 8th Asia-Pacific Software Engineering, Macau, China, 79–86, 2001.

[25] Mili, H., Mili, A., Yacoub, S., and Addy, E., *Reuse-Based Software Engineering: Techniques, Organization, and Controls*. New York, John Wiley and Sons Ltd, 2002.

[26] Szyperski, C., *Component software: Beyond Object-Oriented Programming*. Addison-Wesley, 1998.

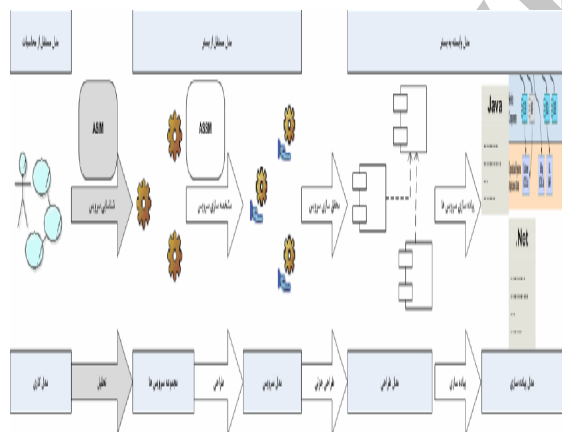
[27] Vitharana, P., Jain, H., and Zahedi, F., "Strategy-Based Design of Reusable Business Components," *IEEE Trans. Systems, Man, and Cybernetics-Part C: Applications and Reviews*. Vol. 34, no. 4, pp. 460–474, 2004.

[28] Xu, W., Yin, B.L., and Li, Z.Y., "Research on the business component design of enterprise information system," *J. of Software*. Vol. 14, no. 7, pp. 1213–1220, 2003.

[29] Hailpern, B., Tarr, P., "Model-driven development: the good, the bad, and the ugly", *IBM Systems Journal*, Vol. 45, Issue 3, pp. 451–461, July 2006.

[30] Hearnden, D., *Deltaware: Incremental Change Propagation for Automating Software Evolution in the Model-Driven Architecture*, PhD thesis, University of Queensland, 2007.

[31] Brown, A.W., Delbaere, M., Eeles, P., Johnston, S., and Weaver, R., "Service-oriented solutions with the IBM Rational Software Development Platform," *IBM System Journal*, Vol. 44, No. 4, 2005.



شکل ۷. مدلها و فرآیندها در چرخه حیات راه حل

[6] Feuerlicht, G., Meesathit, S., *Towards Software Development Methodology for Web Services*, Proc. of the 4th SoMet_W05 International Conference: New Trends in Software Methodologies, Tools and Techniques, Tokyo, Japan, ISBN 1-58603-556-8, pages 263-277, September 28-30, 2005.

[7] Hubbers, J.W., Ligthart, A., Terlouw, L., "Ten Ways to Identify Services," *SOA Magazine*, December 2007.

[8] Zimmermann, O., Koehler, J., and Leymann, F., *Architectural decision models as micro-methodology for Service-Oriented Analysis and Design*, Proceedings of the SEMSOA Workshop 2007 on Software Engineering Methods for Service-Oriented Architecture, pp. 46-60, Hanover, Germany, May 10-11, 2007.

[9] Portier, B., *SOA terminology overview, Part 3: Analysis and design*, IBM® developerWorks®, 2007.

[10] Johnson, S., *Modeling service-oriented solutions*, IBM® developerWorks®, 2005.

[11] Inaganti, S., Behara, G.K., *Service Identification: BPM and SOA Handshake*, Technical Report. Business Process Trends, www.bptrends.com, 2007.

[12] Zhang, Zh., Liu, R., and Yang, H., *Service Identification and packaging in Service-oriented Reengineering*, www.cse.dmu.ac.uk/STRL/research/publications/pubs/2005/2005-8.pdf, 2005.

[13] Teale Ph., Jarvis, R., "Business Patterns for Software Engineering Use, Part2," *The Architecture Journal*, 2004.

[14] Lee, S.D., Yang, Y.J., *COMO: A UML-based component development methodology*, Proc. of 6th Asia Pacific Software Engineering Conf. Takamatsu, pp.54–63, 1998.

[15] Ganesan, R., Sengupta, S., *O2BC: A technique for the design of component-based applications*, Proc. of 39th Int. Conf. and Exhibition on Technology of Object-Oriented Language and Systems, pp.46–55, 2001.

[16] Somjit, A., Dentcho, B., "Development of industrial information systems on the Web using business components," *computer in Industry*, vol.50, no.2, pp.231–250, 2003.

[17] Wang, Z.J., Xu, X.F., Zhan, D.C., "A Survey of Business Component Identification Methods and Related Techniques," *International Journal of Information Technology*, 2, 4 (2005), ISSN 1305-239X, 2005.

[18] Ambler, S.W., Nalbene, J., Vizdos, M.J., *Enterprise Unified Process: Extending the Rational Unified Process*. ISBN: 0131914510, Prentice Hall, chapter 7, 2005.

[19] Martin, J., *Information Engineering*, Prentice Hall, 1990.

[20] Lonsdale systems, *matrix analysis*, <http://www.iinet.net.au/~lonsdale>.

[21] Jamshidi, P., Sharifi, M., Mansour, S., *To Establish Enterprise Service Model from Enterprise Business Model*, Proc. of IEEE International Conference on Services Computing, 2008.

[22] Johnston, S., Smith, J., *RUP Plug-In for SOA V1.0*, IBM® developerWorks®, 2005.