

ارزیابی کمی سبک‌های معماری نرم‌افزار از دید قابلیت اعتماد در نرم‌افزارهای حساس به ایمنی

غلامرضا شاه محمدی سعید جلیلی

دانشکده مهندسی برق و کامپیوتر، دانشگاه تربیت مدرس، تهران، ایران

چکیده

در فرایند طراحی معماری تصمیمات متعددی اتخاذ می‌شود که تاثیر این تصمیمات در سطح سیستم نرم‌افزاری می‌باشد. یکی از این تصمیم‌های مهم، انتخاب سبک معماری مناسب است. با توجه به نقش حیاتی قابلیت اعتماد در سیستم‌های نرم‌افزاری حساس به ایمنی، یکی از معیارهای اصلی در انتخاب سبک معماری نرم‌افزار قابلیت اعتماد بالا می‌باشد. در فرایند طراحی از سبک‌های معماری نرم‌افزار، که راه حل‌های آزموده شده‌ای می‌باشند، زیاد استفاده می‌شود. اما با توجه به اینکه در هر سیستمی ترکیب خاصی از صفات کیفی مورد توجه است، عدم تحلیل کمی تاثیر سبک‌های معماری نرم‌افزار بر روی صفات کیفی مانع استفاده موثر از سبک‌های معماری می‌گردد. این تحقیق گامی به سوی کمی سازی تاثیر سبک‌های معماری نرم‌افزار بر روی صفت کیفی قابلیت اعتماد می‌باشد. به این ترتیب با استفاده از نتایج این تحقیق می‌توان تعیین کرد که در سیستمی که قابلیت اعتماد از اهمیت بالایی برخوردار است، بکارگیری چه سبک‌هایی قابلیت اعتماد بالاتری را بدنبال خواهد داشت.

در این مقاله با استفاده از دو رویکرد دیاگرام بلوکی قابلیت اعتماد و رویکرد مدل مبتنی بر حالت، مارکوف، قابلیت اعتماد سبک‌های معماری نرم‌افزار محاسبه می‌گردد. آنگاه نتایج حاصل از دو رویکرد روی سبک‌های معماری نرم‌افزار، مقایسه می‌گردد. دستاورد اصلی این مقاله کمی سازی و رتبه‌بندی ارزش سبک‌های معماری نرم‌افزار از دید صفت کیفی قابلیت اعتماد در مرحله انتخاب سبک معماری نرم‌افزار می‌باشد.

کلمات کلیدی: معماری نرم‌افزار، ارزیابی قابلیت اعتماد سبک‌های معماری نرم‌افزار، دیاگرام بلوکی قابلیت اعتماد، مدل مارکوف

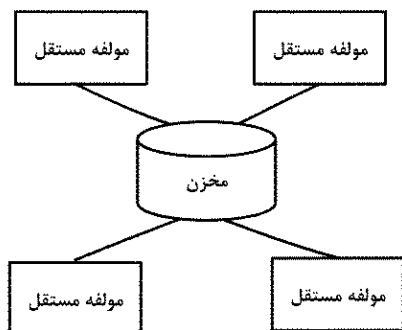
۱- مقدمه

سیستم می‌باشد [۲]. یکی از تصمیم‌های مهم مرحله طراحی نرم‌افزار، انتخاب سبک معماری مناسب است. از آنجا که وظیفه‌مندی‌های نرم‌افزار با استفاده از ساختار(های) متعددی قابل تامین است [۳]، بنابراین عامل تعیین کننده در انتخاب سبک معماری نرم‌افزار، میزان حمایت از صفات کیفی مورد نظر می‌باشد. به همین دلیل تحقیقات متعددی در این زمینه انجام شده است [۴، ۵، ۶، ۷، ۸]. با توجه به نقش اساسی قابلیت اعتماد در سیستم‌های نرم‌افزاری، خصوصاً سیستم‌های نرم‌افزاری حساس به ایمنی، یکی از ملاک‌های اصلی در انتخاب سبک معماری نرم‌افزار، میزان حمایت سبک معماری از قابلیت اعتماد به نرم‌افزار می‌باشد. از مشکلات عمده در استفاده از سبک‌های معماری عدم بررسی تاثیر کمی آنها روی صفات کیفی می‌باشد [۹]. این تحقیق گامی به سوی کمی سازی

نقش نرم‌افزارهای حساس به ایمنی در زندگی بشر آنقدر مهم است، که شکست آنها باعث بروز خسارت‌های مالی و جانی جبران ناپذیری می‌گردد. در نتیجه توجه به صفات کیفی نرم‌افزار در مراحل اولیه چرخه حیات نرم‌افزار اهمیت فراوانی دارد. قابلیت اعتماد یکی از صفات کیفی حیاتی نرم‌افزار تلقی می‌شود و در نرم‌افزارهای حساس به ایمنی تامین آن یک ضرورت اجتناب ناپذیر می‌باشد. طراحی معماری نرم‌افزار اولین مرحله در توسعه نرم‌افزار است که ارزیابی چگونگی تامین نیازهای نرم‌افزار، خصوصاً نیازهای غیروظیفه‌ای، در آن انجام می‌شود [۱]. در فرایند طراحی معماری تصمیمات متعددی اتخاذ می‌شود که تاثیر این تصمیمات در سطح

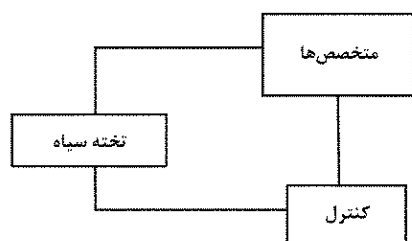
می‌دهد [۱۶].

سبک مخزن (RPS): سبک مخزن (شکل ۱) شامل یک انبار مرکزی به نام مخزن (C_{RPS}) و مجموعه‌ای از مولفه‌ها (C_i) است که اعمال ذخیره، بازیابی و بروز رسانی اطلاعات را بر روی انبار مرکزی انجام می‌دهند. پایگاه داده، نمونه‌ای از این سبک می‌باشد [۱۵]. در سبک مخزن تعامل هر مولفه با مخزن یک تراکنش محسوب می‌شود.



شکل ۱- سبک مخزن

سبک تخته سیاه^۸ (BKB): مولفه‌های این سبک (شکل ۲)، تخته‌سیاه (C_{BKB})، منابع دانش^۹ (متخصص‌ها) (C_{KT}) و مولفه کنترل (C_C) است. مولفه کنترل، یک حلقه را اجراء و ضمن نظارت بر تغییرات تخته‌سیاه و ارزیابی متخصص‌ها، یکی از آنها را برای اجراء، فعال می‌کند [۱۴]. بنابراین انجام یک تراکنش در این سبک مستلزم اجراء برنامه مولفه کنترل و بررسی وضعیت مولفه تخته‌سیاه توسط مولفه کنترل و انتخاب متخصص مناسب و تعامل متخصص با تخته‌سیاه می‌باشد.



شکل ۲- سبک تخته سیاه

سبک لوله و فیلتر^{۱۰} (R/F): این سبک (شکل ۳) از تعدادی مولفه (C_i) تشکیل شده است. هر مولفه در نقش یک فیلتر است و تعدادی ورودی و تعدادی خروجی دارد. خروجی هر مولفه، ورودی مولفه بعدی است [۱۴]. در انجام یک تراکنش کلیه فیلترها موثر می‌باشند.



شکل ۳- سبک لوله و فیلتر

سبک لایه‌ای (LYD): در سبک لایه‌ای (شکل ۴)، تمرکز روی سطوح مختلف انتزاع در نرم‌افزار است. سبک لایه‌ای بصورت سلسله مراتبی سازماندهی می‌شود. هر لایه (C_i) خدمتی را برای لایه بالاتر فراهم و از خدمت لایه پایین‌تر استفاده می‌کند [۱۴]. در انجام یک تراکنش کلیه لایه‌ها موثر می‌باشند.

تأثیر سبک‌های معماری نرم‌افزار بر صفت کیفی قابلیت اعتماد می‌باشد.

روش‌های متعددی برای ارزیابی قابلیت اعتماد نرم‌افزار ارائه شده است [۱۰، ۱۱، ۱۲]. یادآوری می‌گردد در مرحله انتخاب سبک معماری نرم‌افزار در فرایند ساخت نرم‌افزار، وظیفه‌مندی‌های مولفه‌های سبک‌های معماری نرم‌افزار مشخص نیست. در نتیجه تنها برخی از این روش‌ها مانند دی‌اگرام بلوکی قابلیت اعتماد^۱ (RBD) [۱۲] و روش مبتنی بر حالت مارکوف [۱۰] برای ارزیابی قابلیت اعتماد سبک‌های معماری نرم‌افزار، قابل استفاده می‌باشند. در [۱۳]، سبک‌های معماری نرم‌افزار را از دید قابلیت اعتماد براساس روش RBD ارزیابی کردیم و بازای اندازه‌های مختلف نرم‌افزار (تعداد مختلف مولفه‌ها)، قابلیت اعتماد سبک‌های معماری نرم‌افزار را محاسبه کردیم. در این تحقیق: (۱) به جای اندازه نرم‌افزار، با توجه به مفهوم تراکنش و تعداد مولفه‌های موثر در انجام تراکنش در هریک از سبک‌های معماری نرم‌افزار، قابلیت اعتماد سبک‌های معماری نرم‌افزار ارزیابی شده است. (۲) مولفه سرویس‌گر در سبک‌های متقاضی/سرویس‌گر و کارگزار به جای مولفه واحد پیچیده، مرکب از m مولفه در نظر گرفته شده‌اند. (۳) ارزیابی سبک‌های معماری نرم‌افزار از دید قابلیت اعتماد با رویکردهای RBD و مبتنی بر حالت مارکوف انجام و ضمن مقایسه نتایج دو رویکرد، سبک‌های معماری نرم‌افزار رتبه‌بندی شده‌اند.

روش RBD، یک نمایش بلوکی از مولفه‌های نرم‌افزار و وضعیت قابلیت اعتماد آنها ارائه می‌دهد. با توجه به اینکه سبک‌های معماری نرم‌افزار، مدل‌های مختلفی برای حل مسایل معماری ارائه می‌دهند، روش مناسبی برای تعیین قابلیت اعتماد سبک‌های معماری و تخمین قابلیت اعتماد آنها و در نتیجه رتبه‌بندی سبک‌های معماری از دید قابلیت اعتماد در مرحله انتخاب سبک معماری برای نرم‌افزار می‌باشد. در رویکرد مبتنی بر حالت، معماری نرم‌افزار به صورت زنجیر مارکوف زمان گسسته مدل شده و قابلیت اعتماد نرم‌افزار به صورت تحلیلی، تخمین زده می‌شود. بنابراین با تبدیل هر سبک معماری به ماشین حالت، امکان ارزیابی قابلیت اعتماد آن به صورت تحلیلی وجود دارد.

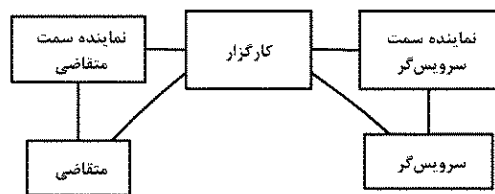
ما در این مقاله با استفاده از دو رویکرد RBD و روش مبتنی بر حالت مارکوف سبک‌های معماری نرم‌افزار را از دید قابلیت اعتماد بطور کمی ارزیابی می‌کنیم. این مقاله در بخش‌های زیر تنظیم شده است. در بخش دوم، سبک‌های معماری نرم‌افزار، در بخش سوم مدل‌های کیفی نرم‌افزار و در بخش چهارم قابلیت اعتماد نرم‌افزار و روشهای ارزیابی آن مطرح می‌گردد. در بخش پنجم ارزیابی کمی و در بخش ششم رتبه‌بندی سبک‌های معماری نرم‌افزار و در بخش هفتم بررسی موردی انجام شده، ارائه می‌گردد. در بخش هشتم نتیجه‌گیری می‌گردد.

۲- سبک‌های معماری نرم‌افزار

سبک‌های معماری نرم‌افزار، مدل‌هایی برای حل مسایل طراحی نرم‌افزار ارائه می‌دهند. هر مدل، مولفه‌های تشکیل‌دهنده، مسئولیت مولفه‌ها، روابط و نحوه همکاری آنها را توصیف می‌کند [۱۴]. استفاده از الگوها در بسیاری از اصول مهندسی رایج است. الگوها براساس مقیاس به دو دسته سبک‌های معماری^۲ و الگوهای طراحی^۳ تقسیم می‌شوند. Shaw و همکارش [۱۵] هفت سبک معماری رایج شامل لوله و فیلتر^۴، شی گراء یا نوع داده انتزاعی، فراخوانی ضمنی^۵، لایه‌ای^۶، مخزن^۷، مفسر و کنترل فرایند را معرفی کرده‌اند. Buschmann و همکارانش [۱۴] نیز در کتاب خود، الگوها را در سطوح مختلف بررسی کرده‌اند.

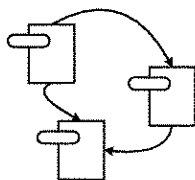
در این بخش با توجه به اینکه قابلیت اعتماد سبک‌های معماری نرم‌افزار را با توجه به مولفه‌های موثر در انجام تراکنش بررسی می‌کنیم، ضمن اشاره به مفهوم تراکنش، هشت سبک معماری بطور اجمالی معرفی می‌شود. یادآوری می‌گردد، تراکنش، مجموعه‌ای از اعمال است که یک واحد منطقی از کار را تشکیل

معرفی و سرویس‌های خود را از طریق واسطه متد، در دسترس متقاضیان قرار می‌دهند. متقاضیان از خدمات سرویس‌گرها از طریق کارگزار استفاده می‌کنند. کارگزار، محل سرویس‌گر مناسب را تعیین، درخواست را به او ارسال، و نتیجه را به متقاضی برمی‌گرداند [۱۴]. تراکنش در این سبک مشابه سبک متقاضی/سرویس‌گر است، اما مولفه‌های بیشتری در آن موثر می‌باشند.



شکل ۷- سبک کارگزار

سبک شی‌گرا^{۱۴} (OO): در سبک شی‌گرا (شکل ۸)، نمایش داده و عملیات مرتبط با آن در یک شی محصور می‌شود. مولفه‌های این سبک، اشیاء و تعامل اشیاء از طریق فراخوانی تابع می‌باشد [۱۵]. تراکنش در سبک شی‌گرا به سناریو معروف است. هر سناریو زنجیری از اجرای متدی از تعدادی شی را شامل می‌شود.

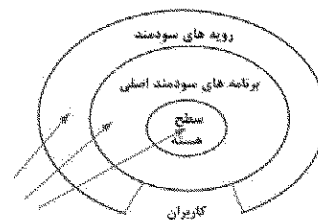


شکل ۸- سبک شی‌گرا

۳- مدل‌های کیفی نرم‌افزار

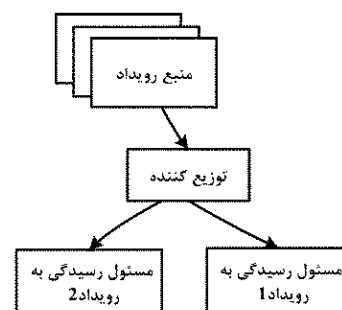
هدف اصلی از ساخت نرم‌افزار، پیاده‌سازی نیازهای وظیفه‌ای آن است. بایستی توجه داشت که در فرایند ساخت نرم‌افزار، کیفیت نرم‌افزار براساس معماری آن پایه‌گذاری می‌شود. یادآوری می‌گردد که طراحی مجدد نرم‌افزارها اغلب به سبب نقص در انجام وظایف محوله به آنها نیست، بلکه به دلیل عدم برخورداری از کیفیت لازم می‌باشد [۱۷]. بنابراین نیرو محرکه‌ی پشتیبان صفات کیفی مهم نرم‌افزار، تقسیم وظایف میان ساختارهای (مولفه‌ها) معماری است.

معماری نرم‌افزار باید دو گروه از نیازهای کیفی را تأمین کند. نیازهای کیفی توسعه و نیازهای کیفی عملیاتی. نیازهای کیفی توسعه، مانند قابلیت نگهداری و قابلیت استفاده مجدد، مورد توجه تولیدکننده نرم‌افزار می‌باشد. نیازهای کیفی عملیاتی، مانند کارایی و قابلیت اعتماد، مورد توجه کاربران سیستم می‌باشند. مدل‌های کیفی مختلفی برای نرم‌افزار ارائه شده که مهم‌ترین آنها مدل MacCall [۱۹]، مدل Boehm [۲۰] و مدل ISO9126 [۲۱] می‌باشد. MacCall یک مدل کیفی سلسله مراتبی سه سطحی از صفات کیفی نرم‌افزار ارائه می‌دهد که قابلیت اعتماد در اولین سطح آن قرار دارد. مدل کیفی Boehm نیز کیفیت نرم‌افزار را مشابه مدل MacCall به صورت سلسله مراتبی از مشخصه‌ها^{۱۵}، در سه سطح تعریف می‌کند و قابلیت اعتماد در سطح دوم این مدل قرار دارد. مدل ISO 9126 که توسط سازمان بین‌المللی استاندارد، برای سنجش کیفیت نرم‌افزار ارائه شده است نیز سلسله مراتبی از مشخصه‌ها در دو سطح است و قابلیت اعتماد در سطح اول آن قرار دارد. با توجه به اینکه مدل کیفی ISO 9126 براساس مدل های MacCall و Boehm تعریف شده است و بسیاری از مشخصه‌های آنها را دارد، بنابراین در این تحقیق صفت کیفی قابلیت اعتماد نرم‌افزار بر مبنای تعریف این مدل مورد سنجش



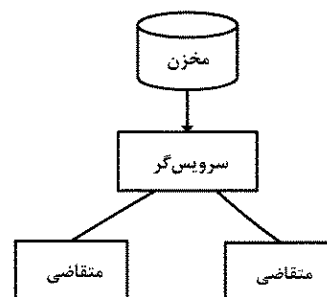
شکل ۴- سبک لایه‌ای

سبک فراخوانی ضمنی (I/I): این سبک (شکل ۵) شامل تعدادی مولفه (C_i) (که مولد رویداد، علاقه‌مند به رویداد و یا هم مولد و هم علاقه‌مند به رویداد می‌باشد) است و یک مولفه در نقش هماهنگ‌کننده یا توزیع کننده (C_d)، با وقوع رویداد، از طریق جدول نگاشت رویداد به مولفه علاقه‌مند مراجعه، ضمن تعیین مولفه علاقه‌مند، پیامی به او ارسال یا رویه‌ای از آن مولفه را فراخوانی می‌کند [۱۵]. تراکنش در این سبک با وقوع رویداد شروع شده و مولفه توزیع کننده، مولفه علاقه‌مند را فعال می‌کند. با اتمام فعالیت مولفه علاقه‌مند تراکنش خاتمه می‌یابد.



شکل ۵- سبک فراخوانی ضمنی

سبک متقاضی/سرویس‌گر (C/S): مولفه‌های این سبک (شکل ۶) متقاضیان (C_c)، سرویس‌گر (ها) (C_s) و مخزن (C_{ms}) می‌باشند. متقاضیان برای استفاده از خدمات سرویس‌گرها، باید از نام و خدمات ارائه شده توسط آنها اطلاع داشته باشند. تراکنش در این سبک با ارسال درخواست متقاضی به سرویس‌گر آغاز می‌شود و به طور معمول نیاز به مراجعه سرویس‌گر به مخزن دارد و در نهایت با ارسال نتیجه توسط سرویس‌گر به متقاضی به پایان می‌رسد.



شکل ۶- سبک متقاضی/سرویس‌گر

سبک کارگزار^{۱۶} (BRK): مولفه‌های این سبک (شکل ۷) متقاضیان (C_c)، سرویس‌گر (ها) (C_s)، کارگزار (C_{brk})، نماینده سمت متقاضی^{۱۷} (C_{esp})، نماینده سمت سرویس‌گر^{۱۸} (C_{ssp}) و مخزن (C_{ms}) است. کارگزار مسئول هماهنگی ارتباط و مجزاکردن متقاضیان از سرویس‌گرها می‌باشد. سرویس‌گرها خود را به کارگزار

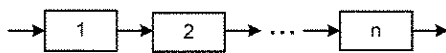
قرار می‌گیرد.

نرم‌افزار می‌توان استفاده کرد. روش دیاگرام بلوکی قابلیت اعتماد (RBD) به عنوان یک روش مبتنی بر ساختار نرم‌افزار، برای ارزیابی قابلیت اعتماد سیستم‌های نرم‌افزاری ارائه شده است. Tripathi و همکارش [۲۳] از RBD برای تخمین قابلیت اعتماد سیستم‌های نرم‌افزاری مرکب از سلسله مراتبی از پیمانه‌ها استفاده کرده‌اند. Leblanc و همکارش [۲۴] براساس RBD مدلی برای نمایش مسایل دنیای واقعی و الگوریتمی برای تحلیل این مدل در مرحله اولیه توسعه نرم‌افزار ارائه کرده‌اند.

در ادامه با استفاده از روش RBD و روش مدل مبتنی بر حالت مارکوف، سبک‌های معماری نرم‌افزار مورد ارزیابی قرار می‌گیرند.

۴-۱- مدل سازی و ارزیابی قابلیت اعتماد سبک‌های معماری نرم‌افزار با روش دیاگرام بلوکی قابلیت اعتماد (RBD)

با توجه به اینکه هر سبک معماری، مولفه‌های تشکیل دهنده معماری و نحوه تعامل آنها با یکدیگر را تعیین می‌کند، روش RBD برای ارزیابی سبک‌های معماری نرم‌افزار مناسب می‌باشد. روش دیاگرام بلوکی قابلیت اعتماد، قابلیت اعتماد سبک‌های معماری نرم‌افزار را براساس ساختار آنها تعیین می‌کند. پیش‌بینی قابلیت اعتماد سبک معماری نرم‌افزار از طریق بررسی قابلیت اعتماد مولفه‌های تشکیل دهنده آن سبک معماری، قابل انجام است. برای ساخت یک RBD باید پیکربندی مولفه‌های سبک معماری نرم‌افزار را تعیین کرد (شکل ۹).



شکل ۹- دیاگرام بلوکی پیکربندی ترتیبی

یک پیکربندی ترتیبی از مولفه‌ها را نشان می‌دهد. در پیکربندی ترتیبی، شکست هر مولفه منجر به شکست نرم‌افزار می‌گردد. قابلیت اعتماد نرم‌افزار با پیکربندی ترتیبی برابر احتمال این است که همه مولفه‌های سبک معماری آن نرم‌افزار درست عمل کنند. در این صورت، قابلیت اعتماد نرم‌افزار از رابطه (۱) محاسبه می‌گردد.

$$R_s = P(X_1 \cap X_2 \cap \dots \cap X_n) = P(X_1) P(X_2|X_1) P(X_3|X_1, X_2) \dots P(X_n|X_1, X_2, \dots, X_{n-1}) \quad (1)$$

در این رابطه R_s قابلیت اعتماد نرم‌افزار، X_i رخداد فعال بودن مولفه i -ام و $P(X_i)$ احتمال فعال بودن مولفه i -ام می‌باشد. اگر مولفه‌ها مستقل از یکدیگر در نظر گرفته شوند رابطه (۱) به رابطه (۲) تبدیل می‌شود. در این رابطه R_i قابلیت اعتماد مولفه i -ام می‌باشد.

$$R_s = \prod_{i=1}^n P(X_i) = \prod_{i=1}^n R_i \quad (2)$$

در نرم‌افزارهای با پیکربندی موازی (شکل ۱۰)، برای موفقیت نرم‌افزار حداقل باید یکی از مولفه‌ها درست عمل کند. احتمال شکست نرم‌افزاری از n مولفه موازی برابر احتمال این است که همه مولفه‌های نرم‌افزار دچار شکست شوند. به عبارت دیگر، اگر حداقل یک مولفه درست عمل کند، آنگاه نرم‌افزار درست عمل خواهد کرد. قابلیت اعتماد نرم‌افزار از رابطه (۳) محاسبه می‌گردد.

برطبق مدل ISO 9126 قابلیت اعتماد، مجموعه‌ای از زیر مشخصه‌ها است که قابلیت نرم‌افزار را برای حفظ کارایی نرم‌افزار تحت شرایط بیان شده در دوره زمانی معین، مشخص می‌کند [۲۱]. به عبارت دیگر قابلیت اعتماد، مدت زمانی را نشان می‌دهد که نرم‌افزار برای استفاده در دسترس است. زیرمشخصه‌های قابلیت اعتماد در مدل ISO 9126 به شرح زیر است.

- بلوغ^{۱۶}: فرکانس شکست نرم‌افزار برآثر عیوب در نرم‌افزار را مشخص می‌کند. هر قدر فرکانس شکست کمتر باشد، میزان بلوغ نرم‌افزار بیشتر است.
 - تحمل خطا: قابلیت حفظ سطح کارایی مشخص شده در صورت بروز عیب در نرم‌افزار را نشان می‌دهد.
 - قابلیت ترمیم: برقراری مجدد سطح کارایی و ترمیم داده‌ها در موقع بروز شکست، و زمان و تلاش مورد نیاز برای آن را نشان می‌دهد.
 - بررسی زیرمشخصه‌های قابلیت اعتماد، موارد زیر را نشان می‌دهد.
 - فرکانس شکست نرم‌افزار با تعداد مولفه‌های حیاتی سبک معماری، ارتباط مستقیم دارد. زیرا هر قدر تعداد مولفه‌های حیاتی سبک معماری بیشتر باشد، امکان بالقوه شکست نرم‌افزار بیشتر می‌گردد.
 - تحمل خطای نرم‌افزار نیز با تعداد مولفه‌های حیاتی سبک معماری، ارتباط معکوس دارد، زیرا هر قدر تعداد مولفه‌های حیاتی سبک معماری بیشتر باشد، تحمل خطای نرم‌افزار کمتر می‌باشد.
 - قابلیت ترمیم با تعداد مولفه‌های حیاتی سبک معماری، ارتباط معکوس دارد. زیرا اصولاً قابلیت ترمیم برای مولفه‌هایی مطرح است که در صورت شکست آنها، کارایی نرم‌افزار دچار تنزل جدی می‌شود.
- با توجه به مطالب فوق‌الذکر، رویکرد RBD و رویکرد مدل مبتنی بر حالت مارکوف که ارزیابی قابلیت اعتماد را براساس مولفه‌های نرم‌افزار و نحوه تعامل آنها انجام می‌دهند، روش‌های مناسبی برای ارزیابی قابلیت اعتماد و زیر مشخصه‌های آن در سطح سبک‌های معماری نرم‌افزار می‌باشند.

۴-۲- قابلیت اعتماد نرم‌افزار و روش‌های ارزیابی آن

در [۱]، روش‌های ارزیابی قابلیت اعتماد در یک انتزاع سطح بالا، به روش‌های کمی و کیفی تقسیم شده‌اند. گونه‌های مختلفی از روش‌های کمی وجود دارد که برخی قبل و برخی بعد از پیاده‌سازی سیستم نرم‌افزاری قابل استفاده می‌باشند. روش‌های مبتنی بر سنجش که روی شکست و از کار افتادگی سیستم متمرکز می‌باشند، قبل و بعد از پیاده‌سازی سیستم نرم‌افزاری قابل استفاده می‌باشند.

روش‌های مدل سازی و ارزیابی قابلیت اعتماد معماری نرم‌افزار [۲۲] به سه رویکرد مبتنی بر حالت، مبتنی بر مسیر و افزایشی دسته بندی می‌شود. در رویکرد مبتنی بر حالت، معماری نرم‌افزار به صورت زنجیر مارکوف زمان گسسته یا پیوسته، یا فرایند نیمه مارکوف مدل شده و قابلیت اعتماد نرم‌افزار به صورت تحلیلی روی معماری با رفتار شکست، تخمین زده می‌شود. در رویکرد مبتنی بر مسیر، برای محاسبه قابلیت اعتماد نرم‌افزار، مسیرهای ممکن اجراء، به صورت تجربی، توسط آزمون یا به صورت الگوریتمی، در نظر گرفته می‌شود. در رویکرد افزایشی، معماری نرم‌افزار بطور صریح در نظر گرفته نمی‌شود، بلکه تمرکز این روش بر تخمین چگالی شکست وابسته به زمان سیستم کاربردی، با استفاده از داده‌های شکست مولفه، می‌باشد. با توجه به این که سبک‌های معماری نرم‌افزار انتزاعی هستند و در نتیجه فاقد مسیرهای ممکن اجراء و تخمین چگالی شکست وابسته به زمان هستند، روش‌های مبتنی بر رویکرد مسیر و رویکرد افزایشی برای ارزیابی قابلیت اعتماد سبک‌های معماری نرم‌افزار، قابل استفاده نمی‌باشند. اما از مدل مبتنی بر حالت مارکوف برای ارزیابی قابلیت اعتماد سبک‌های معماری

۲. یک ماتریس تصادفی $T=(P_{ij})$ ، با ابعاد $n \times n$ ، که $P_{ij} \geq 0$ و $\sum_{j=1}^n P_{ij} = 1$ ، با فرض اینکه سیستم در حالت s_i باشد، احتمال گذار به حالت s_j است.
 ۳. بردار $\pi^0=(\pi_1^0, \dots, \pi_n^0)$ که نشانگر احتمال آن است که سیستم ابتدا در حالت s_i باشد.

تعریف ۱- حالت s_i گذرا است اگر گذار $s_i \rightarrow s_j$ موجود باشد اما گذار $s_j \rightarrow s_i$ موجود نباشد، یعنی s_i به بعضی از s_j ها دسترسی دارد، اما s_j ها به s_i دسترسی ندارند. در غیر اینصورت حالت s_i ارگودیک است.

تعریف ۲- کلاس یک زنجیر مارکوف، کلاس هم ارزی استنتاج شده^{۱۷} توسط رابطه ارتباط روی مجموعه‌ای از حالت‌ها است. یک کلاس α به کلاس β دسترسی دارد اگر برای بعضی از $s_i \in \alpha$ و $s_j \in \beta$ ، $s_i \rightarrow s_j$ باشد. کلاسی که به کلاس دیگری دسترسی نداشته باشد، کلاس نهایی نامیده می‌شود. یک کلاس هم ارزی ارگودیک، یک کلاس نهایی است که همه حالت‌های آن ارگودیک باشد.

قضیه ۱- اگر $a_{ij}^{(q)}$ عنصر (i, j) از A^q را مشخص کند، ماتریس A کاهش‌ناپذیر است اگر و فقط اگر برای هر (i, j) ، یک عدد طبیعی q موجود باشد، بطوریکه $a_{ij}^{(q)} > 0$ باشد.

مدل قابلیت اعتماد نرم‌افزار Cheung [۱۰] که به مدل کاربر-گرا معروف است، و روی مدل مارکوف ارائه شده است، در ادامه معرفی می‌گردد.

۴-۲-۱- مدل Cheung

این مدل از یک ماتریس تصادفی استاندارد $n \times n$ بنام T تشکیل شده است (رابطه ۵). $n=k+2$ به صورت رابطه (۶) است. در ماتریس T ، یک ماتریس داخلی بنام M قرار دارد. این ماتریس، یک ماتریس گذار حالت $k \times k$ است که s_1 حالت ورود و s_k حالت خروج می‌باشد. $M(i, j)$ عنصر i -امین ردیف و j -امین ستون، R_i قابلیت اعتماد مولفه نرم‌افزاری c_i و $P_{i,j}$ احتمال گذار از مولفه c_i به c_j می‌باشد که $1 \leq i, j \leq k$ دو حالت ارگودیک S و F بترتیب حالت موفق و حالت شکست نرم‌افزار را نشان می‌دهند. بنابراین، اجرای موفق نرم‌افزار در نهایت به حالت ارگودیک S و در غیر این صورت به حالت F ختم می‌شود.

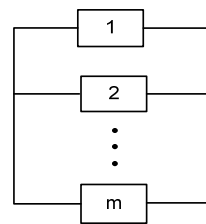
از آنجا که حالات S و F ارگودیک می‌باشند، احتمال گذار از حالت S به S و احتمال گذار از حالت F به F هر یک معادل ۱ می‌باشند. دو بردار داخلی B_1 و B_2 نیز وجود دارد که B_1 فقط یک عنصر مخالف ۰ دارد که احتمال رسیدن به حالت S از حالت خروج s_k است. B_2 احتمال‌های شکست از حالت‌های s_1 تا s_k به حالت F را ذخیره می‌کند.

براساس معادله‌های (۵) و (۶)، این ماشین حالت محدود یک مدل مارکوف زمان گسسته است که مجموع هر ردیف ماتریس استاندارد T برابر ۱ می‌باشد.

$$T = \begin{matrix} & S & F & s_1 \dots s_k \\ \begin{matrix} S \\ F \\ s_1 \dots s_k \end{matrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ B_1 & B_2 & M \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ R_K \end{bmatrix} & \begin{bmatrix} 1-R_1 \\ \vdots \\ 1-R_{K-1} \\ 1-R_K \end{bmatrix} \end{matrix} \quad (5)$$

$$M(i, j) = \begin{cases} R_i P_{ij} & \text{state } s_i \text{ reaches } s_j \text{ and } i \neq k \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

for $1 \leq i, j \leq k$



شکل ۱۰- دیاگرام بلوکی پیکربندی موازی

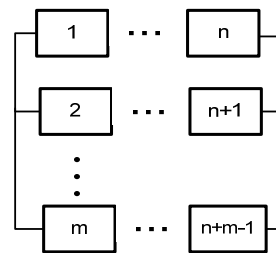
$$R_s = 1 - P(X_1 \cap X_2 \cap \dots \cap X_n) = 1 - [P(X_1)P(X_2|X_1)P(X_3|X_1X_2) \dots P(X_n|X_1(X_2 \dots X_{n-1}))] \quad (3)$$

در این رابطه R_s قابلیت اعتماد نرم‌افزار، X_i پیشامد شکست مولفه i -ام و $P(X_i)$ احتمال شکست مولفه i -ام می‌باشد. اگر مولفه‌ها از یکدیگر مستقل باشند، رابطه (۳) به رابطه (۴) تبدیل می‌شود. در این رابطه R_i قابلیت اعتماد مولفه i -ام می‌شود.

$$R_s = 1 - P(X_1)P(X_2) \dots P(X_n) = 1 - \prod_{i=1}^n P(X_i) = 1 - \prod_{i=1}^n Q_i = 1 - \prod_{i=1}^n (1 - R_i) \quad (4)$$

در این رابطه Q_i عدم اطمینان مولفه i -ام می‌باشد. در پیکربندی موازی، مولفه با بیشترین قابلیت اعتماد، بیشترین اثر را روی قابلیت اعتماد نرم‌افزار دارد، زیرا قابل اعتمادترین مولفه، آخرین مولفه‌ای است که احتمال شکست دارد.

شکل ۱۱، پیکربندی ترتیبی- موازی را نشان می‌دهد. برای محاسبه قابلیت اعتماد نرم‌افزار با پیکربندی ترتیبی- موازی، ابتدا قابلیت اعتماد بخش‌های ترتیبی را محاسبه کرده و در رابطه (۴)، یعنی رابطه قابلیت اعتماد نرم‌افزار با پیکربندی موازی، درج می‌کنیم. به این ترتیب قابلیت اعتماد نرم‌افزار با پیکربندی ترتیبی- موازی بدست می‌آید.



شکل ۱۱- دیاگرام بلوکی پیکربندی ترتیبی- موازی

۴-۲-۲- مدل سازی و ارزیابی قابلیت اعتماد سبک‌های معماری نرم‌افزار با روش مبتنی بر حالت مارکوف

مدل مارکوف، یک ماشین حالت محدود است که احتمال گذار به حالت بعدی فقط به حالت جاری بستگی دارد. در مدل مارکوف زمان گسسته، گذارها در فواصل زمانی گسسته رخ می‌دهند و احتمال گذار، از یک توزیع گسسته تبعیت می‌کند [۲۵]. مدل مارکوف زمان گسسته شامل:

۱. مجموعه محدودی از n حالت $S = \{S_1, \dots, S_n\}$

سبک‌های معماری نرم‌افزار محاسبه و در پایان، نتایج حاصل از دو روش با یکدیگر مقایسه می‌شود.

۵-۱- سنجش سبک‌های معماری نرم‌افزار با روش RBD

در این بخش براساس روش RBD، سبک‌های معماری نرم‌افزار مورد سنجش قرار می‌گیرد و نتایج ارزیابی در جدول ۱ ارائه می‌شود. یادآوری می‌گردد که قابلیت اعتماد مولفه‌ها مستقل از یکدیگر در نظر گرفته می‌شود.

سبک مخزن (RPS): در سبک مخزن انجام تراکنش مستلزم تعامل مولفه مستقلی مانند C_i با مخزن است. در نتیجه قابلیت اعتماد سبک مخزن از رابطه (۸) محاسبه می‌گردد. در این رابطه R_{RPS} ، قابلیت اعتماد مولفه مخزن و R_i قابلیت اعتماد هریک از مولفه‌های مستقل می‌باشد.

$$R = R_i \cdot R_{RPS} \quad (۸)$$

سبک تخته سیاه (BKB): در سبک تخته‌سیاه، انجام تراکنش مستلزم اجرای برنامه مولفه کنترل (C_c) و بررسی وضعیت مولفه تخته‌سیاه (C_{bkb}) توسط مولفه کنترل و انتخاب متخصص مناسب (C_{kr}) و تعامل متخصص با تخته‌سیاه می‌باشد. در نتیجه قابلیت اعتماد سبک تخته‌سیاه از رابطه (۹) محاسبه می‌گردد. در این رابطه R_c ، قابلیت اعتماد مولفه کنترل، R_{bkb} ، قابلیت اعتماد مولفه تخته‌سیاه و R_{kr} قابلیت اعتماد متخصص می‌باشد.

$$R = R_c \cdot R_{bkb} \cdot R_{kr} \quad (۹)$$

سبک لوله و فیلتر (P/F): در سبک لوله و فیلتر (شکل ۳) برای انجام یک تراکنش، همه مولفه‌ها باید فعال باشند. با توجه به اینکه در هر تراکنش m مولفه موثر می‌باشد، قابلیت اعتماد این سبک از رابطه (۱۰) محاسبه می‌شود.

$$R = \prod_{i=1}^m R_i \quad (۱۰)$$

سبک لایه‌ای (LYD): در سبک لایه‌ای (شکل ۴) برای تامین یک سرویس مشخص، همه مولفه‌ها باید فعال باشند. بنابراین قابلیت اعتماد این سبک نیز از رابطه (۱۰) محاسبه می‌شود. در این رابطه R_i ، قابلیت اعتماد هر لایه می‌باشد.

سبک فراخوانی ضمنی (I/I): در این سبک تراکنش با وقوع رویداد شروع شده و منجر به فعال‌سازی مولفه علاقه‌مند (C_i) توسط مولفه توزیع کننده رویداد (C_d) شده و با اتمام فعالیت مولفه علاقه‌مند خاتمه می‌پذیرد. بنابراین قابلیت اعتماد این سبک از رابطه (۱۱) محاسبه می‌شود. در این رابطه، R_d ، قابلیت اعتماد مولفه توزیع کننده رویداد و R_i ، قابلیت اعتماد مولفه علاقه‌مند به رویداد می‌باشد.

$$R = R_d \cdot R_i \quad (۱۱)$$

سبک متقاضی/سرویس‌گر (C/S): در این سبک انجام تراکنش با ارسال درخواست مولفه متقاضی به سرویس‌گر آغاز شده و مولفه‌های سرویس‌گر و مخزن در آن موثر می‌باشند. پس از انجام فرایند مورد نظر توسط سرویس‌گر که معمولاً به تعامل با مولفه مخزن نیاز دارد، نتیجه به مولفه متقاضی ارسال می‌شود. نکته قابل

در مدل Cheung [۱۰]، قابلیت اعتماد نرم‌افزار (R) از رابطه (۷) محاسبه می‌شود.

$$R = (-1)^{k+1} \cdot R_k \frac{|E|}{|I-M|} \quad (۷)$$

$|I-M|$ دترمینان ماتریس ($I-M$) و $|E|$ دترمینان ماتریس حاصل از حذف آخرین ردیف و اولین ستون ماتریس ($I-M$) می‌باشد. از آنجا که T استاندارد است، $|I-M| \neq 0$ است. بنابراین قابلیت اعتماد نرم‌افزار دارای جواب می‌باشد. R_k ، قابلیت اعتماد آخرین مولفه در اجرا می‌باشد.

برای ساخت مدل مارکوف برای ارزیابی قابلیت اعتماد، باید سبک معماری نرم‌افزار به یک ماشین حالت تبدیل شود. از قابلیت اعتماد مولفه و احتمال گذار میان مولفه‌ها برای محاسبه احتمال گذار میان حالت‌ها استفاده می‌شود. در موقع نگاشت سبک معماری نرم‌افزار به ماشین حالت، یک حالت، مجموعه غیر تهی از مولفه‌های نرم‌افزاری در حال اجرا در مدت زمان معینی از شروع تا انتهای اجرای مولفه را نشان می‌دهد. گذار حالت زمانی رخ می‌دهد که مدت زمان اجراء مولفه پایان یابد.

۴-۲-۲- محدودیت مدل‌سازی قابلیت‌اعتماد مارکوف

فرض مدل مارکوف این است که انتقال کنترل میان مولفه‌های نرم‌افزاری یک فرایند مارکوف است. به عبارت دیگر این مدل برای مدل‌سازی نرم‌افزار با رفتار احتمالی مستقل از تاریخچه قابل استفاده می‌باشد. رفتار قطعی نرم‌افزار به معنی زنجیر اجراء با احتمال گذار ۱ بین هر دو مولفه متوالی می‌باشد. زنجیر اجرای قطعی یکی از خصوصیات زیر را دارد.

- استقلال از تاریخچه اجراء. در این مورد هر مولفه در زنجیر دقیقاً یکبار ظاهر می‌شود.
- وابستگی به تاریخچه اجراء. در این مورد حداقل یک مولفه بیشتر از یکبار اجراء می‌شود.

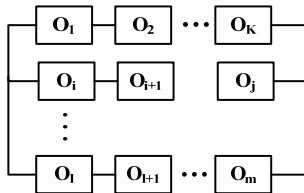
مورد اول، یک فرایند مارکوف است که احتمال همه گذارهایش ۱ است. این مورد با روش‌های معمول قابل مدل‌سازی است. در مورد دوم یک مولفه بیشتر از یک جانشین یا مقدم بلافاصل دارد، و گذار از این مولفه به مولفه بعدی به مولفه قبلی، یعنی تاریخچه اجراء آن بستگی دارد. ویژگی ماتریس تصادفی مارکوف این است که مجموع احتمال گذار از هر مولفه به همه مولفه‌های بعدی بلافاصل آن برابر ۱ می‌باشد. در مورد دوم مولفه‌هایی وجود دارند که این ویژگی را نقض می‌کنند، و زنجیر اجراء، یک فرایند مارکوف نیست. در صورتیکه زنجیر، مستقل از تاریخچه اجراء باشد، برای اجرای یک مولفه همیشه از حالت واحدی استفاده می‌شود. بنابراین تعداد حالت‌ها برای فرایند مارکوف، بدون توجه به تعداد دفعات اجرای هر مولفه، برابر تعداد مولفه‌های اجراء شده می‌باشد. اما در صورتی که زنجیر به تاریخچه اجراء وابسته باشد، برای هر حضور مولفه در زنجیر، به یک شاخص مجزا برای تفکیک نتیجه اثر تاریخچه نیاز است. بنابراین برای نمایش اجراء‌های مختلف مولفه باید از حالت‌های مجزا استفاده کرد [۲۶].

۵- ارزیابی کمی قابلیت اعتماد سبک‌های معماری

در بخش ۴-۱، روش دیگرام بلوکی قابلیت اعتماد (RBD) و در بخش ۴-۲، مدل مبتنی بر حالت مارکوف برای محاسبه قابلیت اعتماد سبک‌های معماری نرم‌افزار معرفی شد. در این بخش به تفکیک این دو روش، قابلیت اعتماد هر یک از

شی گراء شامل موردهای کاربری متعددی است و هر مورد کاربری مسیر اجراء خاص خود را دارد. با شکست یک مورد کاربری، سیستم به کار خود با توان عملیاتی کمتر ادامه می دهد. بنابراین پیکربندی سبک شی گراء مطابق شکل ۱۳ می باشد که هر مسیر، اجراء یک مورد کاربری را نشان می دهد. بنابراین قابلیت اعتماد سبک شی گراء از رابطه (۱۵) محاسبه می شود.

$$R_u = \prod_{i=1}^k R_i \quad (14)$$



شکل ۱۳- دیاگرام بلوکی اجراء تراکنش های مختلف

$$R = 1 - \prod_{u=1}^p (1 - R_u) \quad (15)$$

در این رابطه، p تعداد موردهای کاربری و R_u قابلیت اعتماد هر مورد کاربری می باشد. با جایگذاری R_u از رابطه (۱۴)، رابطه (۱۵) به رابطه (۱۶) تبدیل می شود.

$$R = 1 - \prod_{u=1}^p (1 - \prod_{j=1}^k R_j) \quad (16)$$

از آنجا که تعدادی از کلاس ها در اغلب موردهای کاربری مشترک می باشند، قابلیت اعتماد سبک شی گراء مطابق رابطه (۱۷) می باشد که c تعداد کلاس های مشترک در موردهای کاربری می باشد.

$$R = \prod_{l=1}^c R_l \cdot (1 - \prod_{u=1}^p (1 - \prod_{j=1}^k R_j)) \quad (17)$$

بررسی قابلیت اعتماد بخش موازی $(1 - \prod_{j=1}^k R_j)$ ، با تعداد مختلف کلاس ها نشان داد که قابلیت اعتماد این بخش نزدیک به ۱ است. در نتیجه قابلیت اعتماد سبک شی گراء معادل قابلیت اعتماد بخش ترتیبی $(\prod_{l=1}^c R_l)$ می باشد. بنابراین قابلیت اعتماد این سبک از رابطه (۱۸) محاسبه می شود.

$$R = \prod_{l=1}^c R_l \quad (18)$$

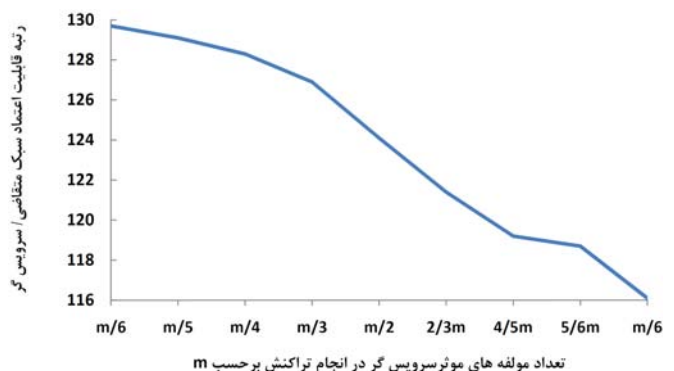
شکل ۱۴، نمودار تغییر رتبه قابلیت اعتماد سبک شی گراء برحسب درصد کلاس های مشترک در موردهای کاربری برای $n_0=15$ است. مطابق نمودار با افزایش این درصد، رتبه قابلیت اعتماد سبک کاهش می یابد. مولفین براساس تجارب خود در تولید سیستم های نرم افزاری، تعداد کلاس های مشترک در موردهای کاربری را ۲۰ درصد تعداد کلاس های سیستم $(0.2n_0)$ در نظر گرفتند.

توجه این است که سرویس گر، یک مولفه مرکب است و خود شامل چند مولفه است. به عبارت دیگر اجراء سرویس گر به منزله اجراء m مولفه می باشد. در نتیجه در انجام تراکنش، $m+1$ مولفه، یعنی یکبار مخزن و m مولفه در سرویس گر، هر یک در تراکنش موثر می باشند. بنابراین قابلیت اعتماد این سبک از رابطه (۱۲) محاسبه می شود. در این رابطه، R_{tps} قابلیت اعتماد مولفه مخزن و $R_1 \dots R_m$ قابلیت اعتماد مولفه های تشکیل دهنده سرویس گر می باشد.

$$R = R_m \dots R_2 R_1 R_{tps} \quad (12)$$

در مولفه سرویس گر سبک های متقاضی/سرویس گر و کارگزار، در انجام تراکنش بخشی از m مولفه موثر می باشند. شکل ۱۲، نمودار تغییر رتبه قابلیت اعتماد سبک متقاضی/سرویس گر را برحسب تعداد مولفه هایی از سرویس گر که در انجام تراکنش موثر می باشند را برحسب m برای $m=5$ نشان می دهد. مطابق نمودار با افزایش تعداد این مولفه ها، رتبه قابلیت اعتماد سبک کاهش می یابد. این مقدار نیز براساس تجارب مولفین در تولید سیستم های نرم افزاری، $m/2$ در نظر گرفته شد.

تولیدکنندگان نرم افزار براساس تجارب گذشته خود در تولید نرم افزار، پارامترهای درصد کلاس های مشترک در موردهای کاربری و تعداد مولفه های موثر سرویس گر در انجام تراکنش را تعیین می کنند.



شکل ۱۲- نمودار تغییر رتبه قابلیت اعتماد سبک متقاضی/سرویس گر برحسب تعداد مولفه های موثر سرویس گر در انجام تراکنش برحسب m

سبک کارگزار (BRK): در سبک کارگزار انجام تراکنش با ارسال درخواست مولفه متقاضی، آغاز شده و مولفه های کارگزار (C_{brk}) ، نماینده سمت متقاضی (C_{esp}) و نماینده سمت سرویس گر (C_{ssp}) ، سرویس گر (C_s) و مخزن (C_{tps}) در انجام تراکنش موثر می باشند.

مشابه سبک متقاضی/سرویس گر اجراء سرویس گر به منزله اجراء $m/2$ مولفه است. در نتیجه در انجام تراکنش، $m/2+7$ مولفه، موثر می باشند. بنابراین قابلیت اعتماد این سبک از رابطه (۱۳) محاسبه می شود. در این رابطه R_{esp} ، قابلیت اعتماد مولفه نماینده سمت متقاضی، R_{brk} قابلیت اعتماد مولفه کارگزار، R_{ssp} قابلیت اعتماد مولفه نماینده سمت سرویس گر و R_{tps} قابلیت اعتماد مولفه مخزن می باشد.

$$R = R_{esp} R_{brk} R_{ssp} R_{m/2} \dots R_2 R_1 R_{tps} \quad (13)$$

سبک شی گراء (OO): در سبک شی گراء (شکل ۸)، به ازای k کلاس موثر در مورد کاربری، قابلیت اعتماد مورد کاربری از رابطه (۱۴) محاسبه می شود. سبک

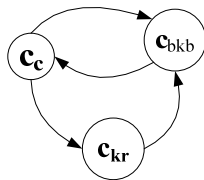
احتمال گذار از هر حالت به حالت بعدی (P_{ij}) قطعی و برابر ۱ است. ماتریس گذار M و $|I - M|$ به صورت زیر می‌باشند.

$$M = \begin{bmatrix} 0 & R_i \\ 0 & 0 \end{bmatrix}, \quad |I - M| = \begin{vmatrix} 1 & -R_i \\ 0 & 1 \end{vmatrix} = 1$$

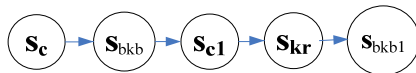
رابطه (۱۹) محاسبه می‌شود. $R_K = R_{tps}$ می‌شود. براساس رابطه (۷)، قابلیت اعتماد سبک مخزن از

$$R = R_i \cdot R_{tps} \quad (19)$$

سبک تخته سیاه (BKB): مطابق شکل ۱۶، مولفه کنترل براساس وضعیت تخته سیاه، متخصص مناسب را فراخوانی می‌کند. متخصص فراخوانی شده با تخته سیاه تعامل می‌کند. احتمال از مولفه کنترل به مولفه تخته سیاه و متخصص ها هریک برابر ۱ می‌باشد. مولفه بعدی مولفه کنترل از نظر اجراء، در موقع بررسی وضعیت مولفه تخته سیاه، مولفه تخته سیاه و بعد از بررسی وضعیت تخته سیاه، یک متخصص است. بنابراین ماشین حالت معادل این سبک، شرط مدل مارکوف را ندارد، و رفتار وابسته به تاریخچه دارد.



شکل ۱۶- تعامل مولفه‌ها در سبک تخته سیاه



شکل ۱۷- ماشین حالت معادل سبک تخته سیاه

در نتیجه باید برای هر احتمال گذار از هر حالت به حالت بعدی (P_{ij}) قطعی و برابر ۱ می‌باشد. ماتریس گذار M و $|I - M|$ به صورت زیر می‌باشند.

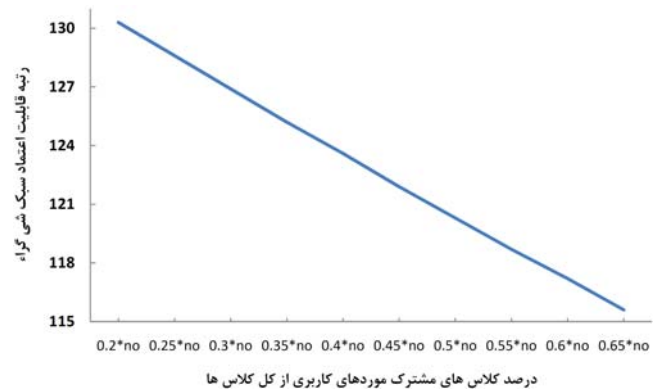
$$M = \begin{bmatrix} 0 & R_c & 0 & 0 & 0 \\ 0 & 0 & R_{bkb} & 0 & 0 \\ 0 & 0 & 0 & R_{c1} & 0 \\ 0 & 0 & 0 & 0 & R_{kr} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad |I - M| = \begin{vmatrix} 1 & -R_c & 0 & 0 & 0 \\ 0 & 1 & -R_{bkb} & 0 & 0 \\ 0 & 0 & 1 & -R_{c1} & 0 \\ 0 & 0 & 0 & 1 & -R_{kr} \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix} = 1$$

سبک از رابطه زیر محاسبه می‌شود. $R_K = R_{bkb1}$ و $|E| = R_c R_{bkb} R_{c1} R_{kr}$ براساس رابطه (۷)، قابلیت اعتماد این

$$R = (-1)^{5+1} \cdot R_{bkb1} \frac{R_c R_{bkb} R_{c1} R_{kr}}{1} = R_c R_{bkb} R_{c1} R_{kr} R_{bkb1}$$

با توجه به اینکه مقادیر R_{bkb} و R_{bkb1} و R_c و R_{c1} یکسان فرض می‌شود، قابلیت اعتماد این سبک از رابطه (۲۰) محاسبه می‌شود.

$$R = R_c^2 R_{bkb}^2 R_{kr} \quad (20)$$



شکل ۱۴- نمودار تغییر رتبه قابلیت اعتماد سبک شی گراء برحسب درصد کلاس های مشترک مورد های کاربری برای $no=15$

جدول ۱، قابلیت اعتماد هریک از سبک‌های معماری نرم‌افزار را براساس روش RBD نشان می‌دهد.

جدول ۱- قابلیت اعتماد سبک‌های معماری نرم‌افزار براساس روش RBD

| نام سبک | نماد | قابلیت اعتماد |
|------------------|------|---|
| مخزن | RPS | $R_i \cdot R_{tps}$ |
| تخته سیاه | BKB | $R_c R_{bkb} R_{kr}$ |
| لوله و فیلتر | P/F | $\prod_{i=1}^m R_i$ |
| لایه‌ای | LYD | $\prod_{i=1}^m R_i$ |
| فراخوانی ضمنی | I/I | $R_d R_i$ |
| متقاضی/ سرویس گر | C/S | $R_{m/2} \dots R_2 R_1 R_{tps}$ |
| کارگزار | BRK | $R_{csp} R_{brk} R_{ssp} R_{m/2} \dots R_2 R_1 R_{tps}$ |
| شی گراء | OO | $\prod_{l=1}^c R_l$ |

۵-۲- سنجش سبک‌های معماری نرم‌افزار براساس مدل مارکوف

در این بخش با استفاده از روش Cheung [۱۰]، که مبتنی بر مدل مارکوف است، قابلیت اعتماد سبک‌های معماری نرم‌افزار را مورد سنجش قرار می‌دهیم.

سبک مخزن (RPS): در سبک مخزن، مولفه‌های مستقل با مخزن از طریق خواندن/نوشتن از/در مخزن، تعامل دارند. درخواست مولفه c_i به مولفه مخزن ارسال و مولفه مخزن درخواست را انجام می‌دهد. ماشین حالت معادل این سبک با توجه به تعامل یک مولفه در انجام یک تراکنش به شکل ۱۵ است. s_i حالت مولفه c_i در موقع ارسال درخواست به مولفه مخزن و s_{tps} حالت معادل مولفه مخزن می‌باشد.



شکل ۱۵- ماشین حالت معادل سبک مخزن

سبک لوله و فیلتر (P/F): شکل ۱۸، ماشین حالت معادل این سبک را با ۳ مولفه نشان می‌دهد. هر حالت متناظر با اجرای یک مولفه (فیلتر) است. یک گذار حالت زمانی رخ می‌دهد که اجرای یک فیلتر کامل شده و کنترل به فیلتر بعدی



شکل ۱۸- ماشین حالت معادل سبک لوله و فیلتر

محول می‌گردد. گذار از هر حالت به حالت بعدی (P_{ij}) قطعی و برابر ۱ می‌باشد، در نتیجه ماتریس گذار M و $|I - M|$ به صورت زیر می‌باشد.

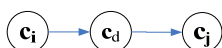
$$M = \begin{bmatrix} 0 & R_1 & 0 \\ 0 & 0 & R_2 \\ 0 & 0 & 0 \end{bmatrix}, \quad |I - M| = \begin{bmatrix} 1 & -R_1 & 0 \\ 0 & 1 & -R_2 \\ 0 & 0 & 1 \end{bmatrix} = 1$$

رابطه (۲۱) محاسبه می‌شود. $|E| = R_1 R_2$ و $R_K = R_3$ براساس رابطه (۷)، قابلیت اعتماد این سبک از رابطه

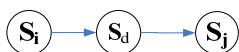
$$R = (-1)^{3+1} \cdot R_3 \frac{R_1 R_2}{1} = R_1 R_2 R_3 \quad (21)$$

در حالت کلی و به ازاء m مولفه موثر در تراکنش، قابلیت اعتماد این سبک از رابطه $\prod_{i=1}^m R_i$ محاسبه می‌شود.

سبک فراخوانی ضمنی (I/I): با توجه به مفهوم تراکنش در این سبک، وقوع رویداد توسط مولفه C_i ، منجر به فعال‌سازی مولفه علاقه‌مند (C_j) توسط مولفه توزیع کننده رویداد (C_d) شده و با اتمام فعالیت مولفه علاقه‌مند تراکنش خاتمه می‌یابد. بنابراین تعامل مولفه‌ها به شکل ۲۱ و ماشین حالت معادل آن به شکل ۲۲ می‌باشد.



شکل ۲۱- تعامل مولفه‌ها در سبک فراخوانی ضمنی



شکل ۲۲- ماشین حالت معادل سبک فراخوانی ضمنی

احتمال گذار از هر حالت به حالت بعدی (P_{ij}) قطعی و برابر ۱ می‌باشد. ماتریس گذار M و $|I - M|$ به صورت زیر می‌باشد.

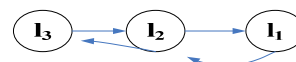
$$M = \begin{bmatrix} 0 & R_i & 0 \\ 0 & 0 & R_d \\ 0 & 0 & 0 \end{bmatrix}, \quad |I - M| = \begin{bmatrix} 1 & -R_i & 0 \\ 0 & 1 & -R_d \\ 0 & 0 & 1 \end{bmatrix} = 1$$

محاسبه می‌شود. $|E| = R_i R_d$ و $R_K = R_j$ براساس رابطه (۷)، قابلیت اعتماد از رابطه (۲۴)

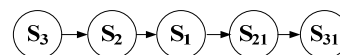
$$R = R_i R_d R_j \quad (24)$$

سبک متقاضی/سرویس‌گر (C/S): تعامل مولفه‌ها در این سبک به شکل ۲۳ می‌باشد. احتمال گذار از مولفه سرویس‌گر (S) به مولفه متقاضی (C) و مخزن (RPS) هریک برابر ۱ می‌باشد. مطابق شکل، مولفه بعدی سرویس‌گر از نظر اجراء یکبار مولفه مخزن و یکبار مولفه متقاضی است. بنابراین این سبک، شرط مدل مارکوف را نداشته و رفتار وابسته به تاریخچه دارد. در نتیجه برای هر اجرای مولفه، باید یک حالت مجزا در نظر گرفت. بنابراین ماشین حالت معادل آن به شکل ۲۴ می‌باشد.

سبک لایه‌ای (LYD): مطابق شکل ۱۹ که نحوه تعامل ۳ مولفه در این سبک را نشان می‌دهد، احتمال گذار از مولفه I_2 به مولفه‌های I_1 و I_3 هریک برابر ۱ می‌باشد. با توجه به اینکه مولفه بعدی مولفه I_2 از نظر اجراء، یکبار مولفه I_1 و یکبار مولفه I_3 است. بنابراین ماشین حالت معادل این سبک، شرط مدل مارکوف را نداشته و رفتار وابسته به تاریخچه دارد. در نتیجه برای هر اجرای مولفه‌ها، باید یک حالت مجزا در نظر گرفت. شکل ۲۰، ماشین حالت معادل شکل ۱۹ است. حالت S_{21} بعد از برگشت کنترل از مولفه I_1 به مولفه I_2 و حالت S_{31} بعد از برگشت کنترل از مولفه I_2 به مولفه I_3 می‌باشد.



شکل ۱۹- سبک لایه‌ای با سه لایه



شکل ۲۰- ماشین حالت معادل سبک لایه‌ای

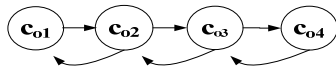
گذار از هر حالت به حالت بعدی (P_{ij}) قطعی و برابر ۱ می‌باشد، در نتیجه ماتریس گذار M و $|I - M|$ به صورت زیر می‌باشد.

$$M = \begin{bmatrix} 0 & R_3 & 0 & 0 & 0 \\ 0 & 0 & R_2 & 0 & 0 \\ 0 & 0 & 0 & R_1 & 0 \\ 0 & 0 & 0 & 0 & R_{21} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad |I - M| = \begin{bmatrix} 1 & -R_3 & 0 & 0 & 0 \\ 0 & 1 & -R_2 & 0 & 0 \\ 0 & 0 & 1 & -R_1 & 0 \\ 0 & 0 & 0 & 1 & -R_{21} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} = 1$$

مطابق نتایج سنجش این سبک به روش RBD، $m/2$ از مولفه‌های سرویس‌گر در تراکنش موثر می‌باشند، قابلیت اعتماد این سبک از رابطه (۲۸) محاسبه می‌شود.

$$R = R_c^2 R_{csp}^2 R_{brk}^2 R_{ssp}^2 R_{m/2}^2 \dots R_2^2 R_1^2 R_{rps} \quad (28)$$

سبک شی گراء (OO): تعامل ۴ کلاس موثر در انجام مورد کاربری به شکل ۲۷ است. مطابق شکل، کلاس بعدی کلاس C_{03} از نظر اجراء یکبار کلاس C_{04} و یکبار کلاس C_{02} است. بنابراین این سبک نیز شرط مدل مارکوف را نداشته و رفتار وابسته به تاریخچه دارد. با توجه به اینمورد و تحلیل قابلیت اعتماد این سبک به روش RBD، کلاس مشترک در انجام مورد کاربری موثر می‌باشند و براساس رابطه (۷)، قابلیت اعتماد آن از رابطه (۲۹) محاسبه می‌شود.



شکل ۲۷- تعامل کلاس‌ها در انجام تراکنش

$$R = R_1^2 R_2^2 \dots R_{c-1}^2 R_c \quad (29)$$

جدول ۲، رابطه قابلیت اعتماد سبک‌های معماری براساس مدل مارکوف را نشان می‌دهد.

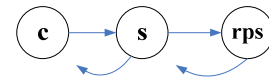
جدول ۲- قابلیت اعتماد سبک‌های معماری نرم‌افزار براساس مدل مارکوف

| نام سبک | نماد | قابلیت اعتماد |
|------------------|------|---|
| مخزن | RPS | $R_i R_{rps}$ |
| تخته سیاه | BKB | $R_c^2 R_{bkb}^2 R_{kr}$ |
| لوله و فیلتر | P/F | $\prod_{i=1}^m R_i$ |
| لایه‌ای | LYD | $R_{m-1}^2 \dots R_2^2 R_1^2$ |
| فراخوانی ضمنی | I/I | $R_i R_d R_j$ |
| متقاضی/ سرویس‌گر | C/S | $R_c^2 R_{m/2}^2 \dots R_2^2 R_1^2 R_{rps}$ |
| کارگزار | BRK | $R_c^2 R_{csp}^2 R_{brk}^2 R_{ssp}^2 R_{m/2}^2 \dots R_2^2 R_1^2 R_{rps}$ |
| شی گراء | OO | $R_1^2 R_2^2 \dots R_{c-1}^2 R_c$ |

۵-۳- مقایسه نتایج ارزیابی دو رویکرد

در بخش ۵-۱ و ۵-۲، سبک‌های معماری نرم‌افزار از دید قابلیت اعتماد براساس روش دیاگرام بلوکی قابلیت اعتماد و روش مبتنی بر حالت مارکوف ارزیابی شدند و قابلیت اعتماد آنها محاسبه شد. بررسی ارزش قابلیت اعتماد در دو روش نشان می‌دهد که:

۱- در رابطه قابلیت اعتماد سبک‌های متقاضی/سرویس‌گر، کارگزار و فراخوانی ضمنی، در روش مبتنی بر حالت مارکوف، قابلیت اعتماد مولفه متقاضی نیز موثر است. یادآوری می‌گردد، منشاء مدل مبتنی بر حالت مارکوف، روش Cheung است، که به مدل کاربرگرا معروف است، یعنی دریافت نتیجه برای کاربر اهمیت دارد. با این دیدگاه، مولفه متقاضی نیز در قابلیت اعتماد اثر دارد.



شکل ۲۳- تعامل مولفه‌ها در سبک متقاضی/ سرویس‌گر



شکل ۲۴- ماشین حالت معادل سبک متقاضی/ سرویس‌گر

احتمال گذار از هر حالت به حالت بعدی (P_{ij}) قطعی و برابر ۱ است. بنابراین ماتریس گذار M و $|I-M|$ به صورت زیر می‌باشد.

$$M = \begin{bmatrix} 0 & R_c & 0 & 0 & 0 \\ 0 & 0 & R_s & 0 & 0 \\ 0 & 0 & 0 & R_{rps} & 0 \\ 0 & 0 & 0 & 0 & R_s \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad |I-M| = \begin{bmatrix} 1 & -R_c & 0 & 0 & 0 \\ 0 & 1 & -R_s & 0 & 0 \\ 0 & 0 & 1 & -R_{rps} & 0 \\ 0 & 0 & 0 & 1 & -R_s \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} = 1$$

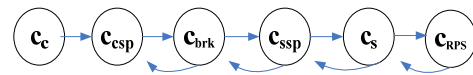
(۷)، از رابطه (۲۵) محاسبه می‌شود.

$$R = (-1)^{5+1} \cdot R_c \frac{R_c R_s R_{rps} R_s}{1} = R_c^2 R_s^2 R_{rps} \quad (25)$$

مطابق نتایج سنجش این سبک به روش RBD، $m/2$ از مولفه‌های سرویس‌گر در تراکنش موثر می‌باشند. بنابراین قابلیت اعتماد این سبک از رابطه (۲۶) محاسبه می‌شود.

$$R = R_c^2 R_{m/2}^2 \dots R_2^2 R_1^2 R_{rps} \quad (26)$$

سبک کارگزار (BRK): در سبک کارگزار، تعامل مولفه‌ها به شکل ۲۵ می‌باشد. احتمال گذار از مولفه کارگزار به سرویس‌گر و متقاضی، هریک برابر ۱ می‌باشد. مطابق این شکل و مشابه سبک متقاضی/سرویس‌گر، این سبک، شرط مدل مارکوف را نداشته و رفتار وابسته به تاریخچه دارد. در نتیجه برای هر اجرای مولفه، باید یک حالت مجزا در نظر گرفت. بنابراین ماشین حالت این سبک به شکل ۲۶ می‌باشد.



شکل ۲۵- تعامل مولفه‌ها در سبک کارگزار



شکل ۲۶- ماشین حالت معادل سبک کارگزار

براساس رابطه (۷)، قابلیت اعتماد این سبک از رابطه (۲۷) محاسبه می‌شود.

$$R = R_c^2 R_{csp}^2 R_{brk}^2 R_{ssp}^2 R_s^2 R_{rps} \quad (27)$$

جدول ۴- مقادیر قابلیت اعتماد سبک‌های معماری نرم‌افزار براساس مدل مارکوف

| نماد | m=2 | m=3 | m=4 | m=5 | m=6 | m=7 |
|------|-------------------|-------------------|--------------------|--------------------|--------------------|--------------------|
| سبک | n ₀ =6 | n ₀ =9 | n ₀ =12 | n ₀ =15 | n ₀ =18 | n ₀ =21 |
| RPS | ۰.۹۶ | ۰.۹۶ | ۰.۹۶ | ۰.۹۶ | ۰.۹۶ | ۰.۹۶ |
| BKB | ۰.۹۲ | ۰.۹۲ | ۰.۹۲ | ۰.۹۲ | ۰.۹۲ | ۰.۹۲ |
| P/F | ۰.۹۶ | ۰.۹۴ | ۰.۹۲ | ۰.۹ | ۰.۸۹ | ۰.۸۷ |
| LYD | ۰.۹۵ | ۰.۹۲ | ۰.۸۹ | ۰.۸۷ | ۰.۸۴ | ۰.۸۲ |
| I/I | ۰.۹۴ | ۰.۹۴ | ۰.۹۴ | ۰.۹۴ | ۰.۹۴ | ۰.۹۴ |
| C/S | ۰.۹۲ | ۰.۹۱ | ۰.۸۹ | ۰.۸۸ | ۰.۸۷ | ۰.۸۶ |
| BRK | ۰.۸۴ | ۰.۸۳ | ۰.۸۲ | ۰.۸۱ | ۰.۷۹ | ۰.۷۸ |
| OO | ۰.۹۷ | ۰.۹۵ | ۰.۹۴ | ۰.۹۳ | ۰.۹۲ | ۰.۹۱ |

برای رتبه‌بندی گزینه‌ها از دید یک معیار در روش تصمیم‌گیری چند معیاره فرآیند تحلیل سلسله‌مراتبی^{۱۸}، از نرمال سازی مقادیر معیار استفاده می‌شود. در اینجا نیز برای محاسبه رتبه سبک‌های معماری از دید قابلیت اعتماد از این شیوه استفاده می‌کنیم. بنابراین رتبه قابلیت اعتماد سبک‌های معماری، در دو رویکرد، از رابطه (۳۰) محاسبه و در جدول‌های ۵ و ۶ ارائه شده است.

در این رابطه، x_{ij} ارزش قابلیت اعتماد سبک در ردیف i -ام و ستون j -ام جدول 3×4 ، و مخرج رابطه مجموع ارزش قابلیت اعتماد ستون j -ام است. از ضریب ۱۰۰۰ در رابطه (۳۰) برای وضوح رتبه سبک‌ها استفاده شده است.

$$R_{ij} = \frac{x_{ij}}{\sum_{i=1}^s x_{ij}} * 1000 \quad (30)$$

ضمن اینکه رتبه قابلیت اعتماد سبک‌های معماری در دو رویکرد اندکی متفاوت است، و این اختلاف ناشی از تفاوت رابطه قابلیت اعتماد سبک‌های معماری در دو رویکرد است، با اینحال روند تغییر رتبه سبک‌های معماری با توجه به تغییر تعداد مولفه‌های موثر در انجام تراکنش در دو رویکرد یکسان است. روش مبتنی بر حالت مارکوف تمایز بیشتری را میان رتبه قابلیت اعتماد سبک‌های معماری نشان می‌دهد.

جدول ۵- رتبه سبک‌های معماری نرم‌افزار از دید قابلیت اعتماد براساس روش RBD

| نماد | m=2 | m=3 | m=4 | m=5 | m=6 | m=7 |
|------|-------------------|-------------------|--------------------|--------------------|--------------------|--------------------|
| سبک | n ₀ =6 | n ₀ =9 | n ₀ =12 | n ₀ =15 | n ₀ =18 | n ₀ =21 |
| RPS | ۱۲۶ | ۱۲۷.۳ | ۱۲۸.۳ | ۱۳۱.۷ | ۱۳۰.۴ | ۱۳۱.۷ |
| BKB | ۱۲۳.۴ | ۱۲۴.۷ | ۱۲۵.۷ | ۱۲۸.۹ | ۱۲۷.۷ | ۱۲۸.۹ |
| P/F | ۱۲۶ | ۱۲۴.۷ | ۱۲۳ | ۱۲۳.۵ | ۱۲۰.۹ | ۱۱۹.۳ |
| LYD | ۱۲۶ | ۱۲۴.۷ | ۱۲۳ | ۱۲۳.۵ | ۱۲۰.۹ | ۱۱۹.۳ |
| I/I | ۱۲۶ | ۱۲۷.۳ | ۱۲۸.۳ | ۱۳۱.۷ | ۱۳۰.۴ | ۱۳۱.۷ |
| C/S | ۱۲۶ | ۱۲۶ | ۱۲۵.۷ | ۱۲۷.۶ | ۱۲۵ | ۱۲۴.۸ |
| BRK | ۱۱۸.۱ | ۱۱۸ | ۱۱۹ | ۱۲۰.۷ | ۱۱۸.۲ | ۱۱۸ |
| OO | ۱۲۸.۶ | ۱۲۷.۳ | ۱۲۷ | ۱۲۷.۵ | ۱۲۶.۴ | ۱۲۶.۲ |

شکل ۲۸، نمودار تغییر رتبه قابلیت اعتماد سبک‌های معماری را برحسب تعداد مولفه‌های موثر در انجام تراکنش در رویکرد مبتنی بر حالت مارکوف نشان می‌دهد. مطابق این نمودار، با افزایش تعداد مولفه‌های موثر در انجام تراکنش رتبه

۲- روش RBD، قابلیت اعتماد سبک‌های معماری را براساس مولفه‌های موثر در انجام تراکنش محاسبه می‌کند، اما به تعداد دفعات حضور این مولفه‌ها در انجام تراکنش توجه ندارد. روش مبتنی بر حالت مارکوف، ماشین حالت معادل هر سبک معماری را در نظر گرفته و براساس آن قابلیت اعتماد هر سبک معماری محاسبه می‌شود.

در نتیجه به تعداد دفعات حضور هر مولفه در انجام تراکنش توجه دارد. در اغلب سبک‌های معماری مانند سبک لایه‌ای، حضور مجدد مولفه در انجام تراکنش اغلب به سبب برگشت از فراخوانی است و در عمل مولفه کار خاصی انجام نمی‌دهد و زمان اندکی صرف برگشت از فراخوانی می‌شود. در نتیجه قابلیت اعتماد مولفه‌ها در این حالت ۰.۹۹ در نظر می‌گیریم.

۴-۵- ارزیابی سیستم‌های بزرگ

در دسته‌ای از سیستم‌های نرم‌افزاری نه تنها از یک سبک معماری به عنوان مبنای اصلی ساختاردهی به سیستم استفاده می‌شود، بلکه هر یک از اجزاء آنها نیز از سبک معماری خاصی استفاده می‌کنند. با توجه به تعیین قابلیت اعتماد سبک‌های معماری در این تحقیق، امکان ارزیابی و تعیین قابلیت اعتماد چنین سیستم‌هایی نیز وجود دارد.

۶- رتبه‌بندی سبک‌های معماری نرم‌افزار

در بخش پنجم، قابلیت اعتماد سبک‌های معماری نرم‌افزار محاسبه گردید. در رابطه قابلیت اعتماد اغلب سبک‌های معماری نرم‌افزار، پارامتر m که نشانگر تعداد مولفه‌های موثر در انجام تراکنش است، وجود دارد. بنابراین به منظور بررسی تاثیر تعداد مولفه‌های موثر در انجام تراکنش در قابلیت اعتماد سبک‌های معماری نرم‌افزار، مقدار m ، ۲، ۳، ۴، ۵، ۶ و ۷ در نظر گرفته می‌شود. مولفه‌های سبک شی‌گرا، یعنی کلاس‌ها نسبت به مولفه‌های سایر سبک‌های معماری ریزدانه‌تر می‌باشند، با این تقریب که به طور متوسط در هر مولفه ۳ کلاس در انجام تراکنش موثر باشند، متناظر با تعداد مولفه‌های موثر در انجام تراکنش، در سایر سبک‌های معماری، تعداد کلاس‌های موثر در انجام تراکنش ۶، ۹، ۱۲، ۱۵، ۱۸ و ۲۱ در نظر گرفته می‌شود. قابلیت اعتماد هر مولفه/کلاس را ۰.۹۸ در نظر می‌گیریم.

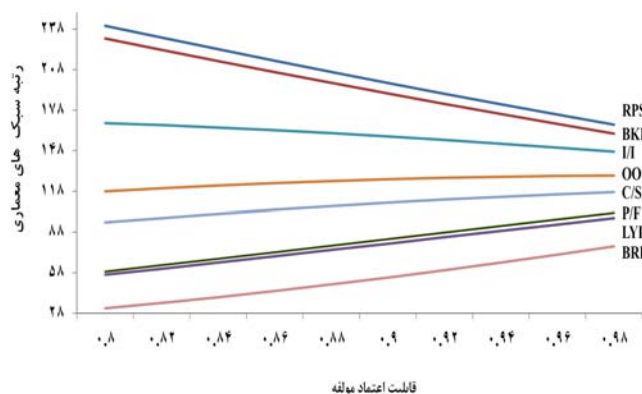
به ازاء تعداد مختلف مولفه‌های موثر در انجام تراکنش و براساس رابطه قابلیت اعتماد سبک‌های معماری در جدول‌های ۱ و ۲، و با توجه به مقدار قابلیت اعتماد هر مولفه/کلاس، مقادیر قابلیت اعتماد هریک از سبک‌های معماری نرم‌افزار در دو رویکرد محاسبه و در جدول‌های ۳ و ۴ درج شده است.

جدول ۳- مقادیر قابلیت اعتماد سبک‌های معماری نرم‌افزار براساس روش RBD

| نماد | m=2 | m=3 | m=4 | m=5 | m=6 | m=7 |
|------|-------------------|-------------------|--------------------|--------------------|--------------------|--------------------|
| سبک | n ₀ =6 | n ₀ =9 | n ₀ =12 | n ₀ =15 | n ₀ =18 | n ₀ =21 |
| RPS | ۰.۹۶ | ۰.۹۶ | ۰.۹۶ | ۰.۹۶ | ۰.۹۶ | ۰.۹۶ |
| BKB | ۰.۹۴ | ۰.۹۴ | ۰.۹۴ | ۰.۹۴ | ۰.۹۴ | ۰.۹۴ |
| P/F | ۰.۹۶ | ۰.۹۴ | ۰.۹۲ | ۰.۹ | ۰.۸۹ | ۰.۸۷ |
| LYD | ۰.۹۶ | ۰.۹۴ | ۰.۹۲ | ۰.۹ | ۰.۸۹ | ۰.۸۷ |
| I/I | ۰.۹۶ | ۰.۹۶ | ۰.۹۶ | ۰.۹۶ | ۰.۹۶ | ۰.۹۶ |
| C/S | ۰.۹۶ | ۰.۹۵ | ۰.۹۴ | ۰.۹۳ | ۰.۹۲ | ۰.۹۱ |
| BRK | ۰.۹ | ۰.۸۹ | ۰.۸۹ | ۰.۸۸ | ۰.۸۷ | ۰.۸۶ |
| OO | ۰.۹۸ | ۰.۹۶ | ۰.۹۵ | ۰.۸۲ | ۰.۹۳ | ۰.۹۲ |

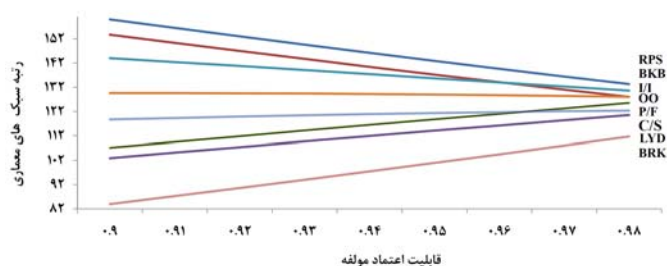
در نظر می‌گیریم.

در این بخش با تغییر قابلیت اعتماد مولفه‌های طراحی شده از ۰.۸ تا ۰.۹۸، تاثیر این تغییر را بر قابلیت اعتماد سبک‌های معماری به ازاء $m=5$ بررسی می‌کنیم. مطابق نمودار شکل ۲۹، با افزایش قابلیت اعتماد مولفه‌های طراحی شده، تفاوت رتبه قابلیت اعتماد سبک‌های معماری کمتر می‌شود. بازاء $R=0.8$ ، سبک کارگزار کمترین و سبک مخزن بالاترین رتبه را از نظر قابلیت اعتماد دارد. با افزایش قابلیت اعتماد مولفه‌های طراحی شده، روند تغییر رتبه قابلیت اعتماد سبک‌های مخزن، تخته سیاه و فراخوانی ضمنی کاهشی و روند تغییر رتبه قابلیت اعتماد سایر سبک‌های معماری افزایشی است.



شکل ۲۹- نمودار تغییر رتبه سبک‌های معماری برحسب تغییر قابلیت اعتماد مولفه‌های طراحی شده

به منظور نمایش واضح تر تغییر رتبه سبک‌های معماری، در نمودار شکل ۳۰، روند تغییر قابلیت اعتماد مولفه‌های طراحی شده در محدوده ۰.۹ تا ۰.۹۸ نشان داده شده است.



شکل ۳۰- نمودار تغییر رتبه سبک‌های معماری برحسب تغییر قابلیت اعتماد مولفه‌های طراحی شده

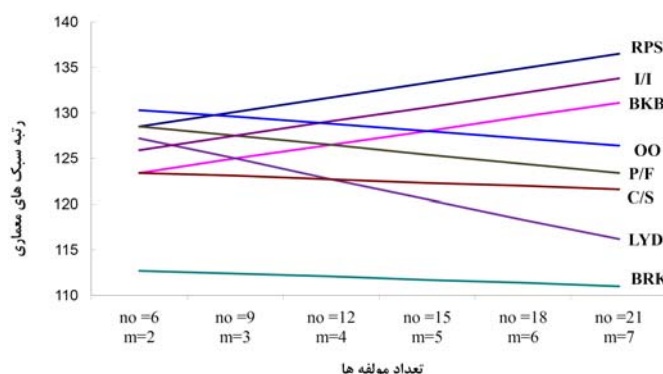
۷- بررسی موردی

سیستم نرم‌افزاری مورد بررسی، سیستم کنترل ترافیک هوایی^{۱۹} (ATC) [۳] می‌باشد. سیستم کنترل ترافیک هوایی تضمین حرکت ایمن هواپیما در خطوط هوایی را بر عهده دارد. هواپیماها در هر پرواز با تعدادی موجودیت ATC سرو کار دارند که اطلاعات خود را به مرکز کنترل ترافیک هوایی ارسال می‌کنند. این سیستم باید در فاصله زمانی تعیین شده واکنش نشان دهد. بنابراین از نوع بیدرنگ سخت^{۲۰} است و حساس به ایمنی تلقی می‌شود، زیرا زندگی انسان‌ها به آن بستگی دارد. بنابراین چنین سیستمی باید از قابلیت اعتماد بالایی برخوردار باشد. با فرض آنکه تعداد مولفه‌های موثر در انجام هر تراکنش را ۴ مولفه در نظر بگیریم، و با توجه به اینکه قابلیت اعتماد مولفه‌ها در چنین سیستمی باید بالا باشد، مطابق

سبک‌های مخزن (RPS)، فراخوانی ضمنی (I/I)، و تخته سیاه (BKB) افزایش و رتبه سایر سبک‌های معماری کاهش می‌یابد. از نظر شهودی این نتایج درست است. در سبک‌های مخزن (RPS)، فراخوانی ضمنی (I/I)، و تخته سیاه (BKB) تعداد مولفه‌های موثر در انجام تراکنش، ثابت است. در سایر سبک‌های معماری مانند سبک‌های لایه‌ای (LYD)، لوله و فیلتر (P/F)، با افزایش تعداد مولفه‌های موثر در انجام تراکنش، رتبه قابلیت اعتماد سبک‌ها کاهش می‌یابد. در سبک کارگزار (BRK) و سبک متقاضی/سرویس‌گر (C/S) علاوه بر m مولفه، مولفه‌های ثابت دیگری در تراکنش موثر می‌باشند که باعث کاهش رتبه قابلیت اعتماد آنها می‌گردد.

جدول ۶- رتبه سبک‌های معماری نرم‌افزار از دید قابلیت اعتماد براساس مدل مارکوف

| نماد سبک | m=2 | m=3 | m=4 | m=5 | m=6 | m=7 |
|----------|---------|---------|----------|----------|----------|----------|
| | $n_0=6$ | $n_0=9$ | $n_0=12$ | $n_0=15$ | $n_0=18$ | $n_0=21$ |
| RPS | ۱۲۸.۵ | ۱۳۰.۱ | ۱۳۱.۷ | ۱۳۳.۳ | ۱۳۴.۹ | ۱۳۶.۵ |
| BKB | ۱۲۳.۴ | ۱۲۵ | ۱۲۶.۵ | ۱۲۸ | ۱۲۹.۶ | ۱۳۱.۱ |
| P/F | ۱۲۸.۵ | ۱۲۷.۵ | ۱۲۶.۵ | ۱۲۵.۴ | ۱۲۴.۴ | ۱۲۳.۴ |
| LYD | ۱۲۷.۲ | ۱۲۵ | ۱۲۲.۷ | ۱۲۰.۵ | ۱۱۸.۳ | ۱۱۶.۲ |
| I/I | ۱۲۵.۹ | ۱۲۷.۵ | ۱۲۹.۱ | ۱۳۰.۶ | ۱۳۲.۲ | ۱۳۳.۸ |
| C/S | ۱۲۳.۴ | ۱۲۳.۱ | ۱۲۲.۷ | ۱۲۲.۳ | ۱۲۲ | ۱۲۱.۶ |
| BRK | ۱۱۲.۷ | ۱۱۲.۴ | ۱۱۲.۱ | ۱۱۱.۷ | ۱۱۱.۴ | ۱۱۱ |
| OO | ۱۳۰.۳ | ۱۲۹.۶ | ۱۲۸.۸ | ۱۲۸ | ۱۲۷.۲ | ۱۲۶.۴ |



شکل ۳۱- نمودار قابلیت اعتماد سبک‌های معماری نرم‌افزار برحسب تعداد مولفه‌های موثر در تراکنش در رویکرد مبتنی بر حالت مارکوف

۶-۱- تحلیل رتبه قابلیت اعتماد سبک‌ها با توجه به قابلیت اعتماد مولفه‌های طراحی شده

در اغلب سبک‌های معماری، وظایف برخی از مولفه‌ها بدون توجه به نرم‌افزار خاص مشخص است. سایر مولفه‌ها با توجه به وظایف نرم‌افزار خاص طراحی می‌شوند. به این ترتیب مولفه‌های هر سبک معماری به دو نوع با وظایف مشخص و مولفه‌های طراحی شده تقسیم می‌شوند. بطور مثال در سبک کارگزار، مولفه‌های کارگزار، نماینده سمت متقاضی و نماینده سمت سرویس‌گر و مخزن از نوع با وظایف مشخص و سایر مولفه‌ها از نوع طراحی شده می‌باشند. قابلیت اعتماد مولفه‌های با وظایف مشخص را ۰.۹۸ در نظر می‌گیریم. همانگونه که در بخش ۵-۳ اشاره گردید، قابلیت اعتماد هر مولفه در زمان برگشت از فراخوانی مولفه دیگر، را ۰.۹۹

[2] A. G. Jansen, and J. Bosch, "Software Architecture as a set of Architectural Design Decisions," *Proc. 5th IEEE/IFIP Working Conf. on Software Architecture*, pp. 109–119, 2005.

[3] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, 2nd Edition, Addison-Wesley, 2003.

[4] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, *Non-Functional Requirements in Software Engineering*, Kluwer Academic Publishing, 2000.

[5] F. Losavio, L. Chirinos, A. Matteo, N. Lévy, and A. Ramdane-Cherif, "ISO Quality standards for measuring architectures," *The Journal of System and Software*, Vol. 72, No. 2, pp. 209–223, 2004.

[6] J. Shao, and B. H. Far, "Development of an Intelligent System for Architecture Design and Analysis," *Proc. Of the Canadian Electrical and Computer Engineering Conference*, PP.539-542, 2004.

[7] H. Reza, and E. Grant, "Quality-Oriented Software Architecture," *Proc. of the IEEE Conf on Information Technology: Coding and Computing*, pp. 140–145, 2005.

[8] M. AlSharif, W. P. Bond, and T. Al-Otaiby, "Assessing the complexity of software architecture," *Proc. of the 42nd annual ACM Southeast regional conference*, pp. 98–103, 2004.

[9] N. B. Harrison, and P. Avgeriou, "Leveraging Architecture Patterns to Satisfy Quality Attributes," *First European Conf. on Software Architecture*, pp. 263–270, 2007.

[10] R. C. Cheung, "A user-oriented software reliability model," *IEEE Trans. on Software Engineering*, Vol.6, No. 2, pp. 118–125, 1980.

[11] Y. Meng-Lai, C. L. Hyde, and L. E. James, "A Petri-net approach for early-stage system-level software reliability estimation," *Proc. of the IEEE Annual Reliability and Maintainability Symposium*, pp. 100–105, 2000.

[12] W. Farr, *Software Reliability Modeling Survey*, Chapter 3, *Handbook of Software Reliability Engineering*, McGraw-Hill, New York, 1996.

[۱۳] غ. ر. شاه محمدی، س. جلیلی، "ارزیابی کمی سبک‌های معماری نرم‌افزار از دید قابلیت اعتماد با رویکرد مدل‌سازی دیگرام بلوکی،" *مجموعه مقالات سیزدهمین کنفرانس بین‌المللی انجمن کامپیوتر ایران*، ۱۳۸۶.

[14] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal., *Pattern-Oriented Software Architecture-A system of Patterns*, John Wiley & Sons, 1996.

[15] M. Shaw, and D. Garlan, *software architecture: Perspectives on an Emerging Discipline*, Prentice Hall, 1996.

[16] A. Silberschatz, H. F. Korth, and S. Sudarshan, *Database System Concepts*, McGraw-Hill Series in

جدول ۶ سبک‌های شی گراء (OO)، مخزن (RPS)، و فراخوانی ضمنی (I/I) برای این سیستم توصیه می‌شود.

۸- نتیجه‌گیری

در این مقاله ابتدا مروری اجمالی بر سبک‌های معماری نرم‌افزار، دسته‌بندی‌ها و مدل‌های کیفی نرم‌افزار، داشتیم. آنگاه صفت کیفی قابلیت اعتماد در مدل ISO 9126 تبیین گردید. با توجه به فقدان بررسی کمی تاثیر سبک‌های معماری روی صفات کیفی خصوصاً قابلیت اعتماد و نظر به اهمیت سنجش قابلیت اعتماد سبک‌های معماری نرم‌افزار، به منظور انتخاب سبک معماری مناسب، روش‌های ارزیابی قابلیت اعتماد نرم‌افزار بررسی و مشخص گردید که در سطح سبک‌های معماری نرم‌افزار، دو رویکرد RBD و رویکرد مبتنی برحالت مارکوف برای ارزیابی سبک‌های معماری نرم‌افزار از دید قابلیت اعتماد قابل استفاده می‌باشند. آنگاه قابلیت اعتماد سبک‌های معماری با استفاده از این دو رویکرد محاسبه و نتایج آنها با هم مقایسه گردید.

نتایج رتبه‌بندی سبک‌های معماری در دو رویکرد نشان داد که ضمن اینکه مقادیر قابلیت اعتماد سبک‌های معماری در دو رویکرد اندکی متفاوت است، و این اختلاف ناشی از تفاوت رابطه قابلیت اعتماد سبک‌های معماری در دو رویکرد است، با اینحال روند تغییر رتبه سبک‌های معماری با تغییر تعداد مولفه‌های موثر در انجام تراکنش در دو رویکرد یکسان است. ضمن اینکه تمایز بیشتری بین رتبه سبک‌های معماری در رویکرد مبتنی بر مدل مارکوف مشاهده شد. به منظور بررسی تاثیر تعداد مختلف مولفه‌های موثر در انجام تراکنش روی رتبه صفت کیفی قابلیت اعتماد سبک‌های معماری، با تغییر تعداد مولفه‌ها از ۲ تا ۷، ارزش قابلیت اعتماد سبک‌های معماری محاسبه گردید.

مهمترین دستاورد این مقاله، کمی کردن صفت کیفی قابلیت اعتماد در سطح سبک‌های معماری نرم‌افزار و در مرحله انتخاب سبک معماری نرم‌افزار است. کار مشابهی که به ارزیابی، رتبه‌بندی و مقایسه قابلیت اعتماد سبک‌های معماری نرم‌افزار پرداخته باشد، مشاهده نگردیده است. Wang و همکاران [۲۷]، با اصلاح مدل قابلیت اعتماد مبتنی بر مارکوف پیشنهادی Cheung [۱۰]، ساخت یک ماشین حالت را که قادر به بررسی مدل‌های همگن و ناهمگن باشد، ارائه داده‌اند. روش آنها برای ارزیابی سبک‌های معماری ناهمگن قابل استفاده می‌باشد.

Shao و همکارش [۶] یک سیستم هوشمند برای انتخاب سبک معماری نرم‌افزار ارائه داده‌اند، اما چگونگی ارزیابی و رتبه‌بندی سبک‌های معماری نرم‌افزار از دید صفات کیفی را مشخص نکرده‌اند.

سپاسگزاری

این تحقیق با حمایت مالی مرکز تحقیقات مخابرات ایران طرح شماره ۸۵-۱۲-۸۵ TMU انجام شده است.

مراجع

[1] A. Immonen, and E. Niemelä, "Survey of reliability and availability prediction methods from the viewpoint of software architecture," *Journal of Software and Systems Modeling*, Vol. 7, No.1, pp. 49–65, 2008.



سعید جلیلی کارشناسی‌ارشد خود را در سال ۱۳۶۴ از دانشگاه صنعتی شریف و درجه دکترای خود را در سال ۱۳۷۰ از دانشگاه برادفورد انگلستان اخذ نموده است. زمین‌های تحقیقاتی مورد علاقه ایشان، آزمون نرم‌افزار، واریسی حین اجرای نرم‌افزار، ارزیابی کمتی معماری نرم‌افزار و یادگیری ماشین می‌باشد.

آدرس پست‌الکترونیکی ایشان عبارت است از:

sjalili@modares.ac.ir

اطلاعات بررسی مقاله:

تاریخ ارسال: ۸۷/۰۶/۲۶

تاریخ اصلاح: ۸۹/۰۴/۲۹

تاریخ قبول شدن: ۸۹/۰۵/۱۱

نویسنده مرتبط: دکتر سعید جلیلی، دانشکده مهندسی برق و کامپیوتر، دانشگاه تربیت مدرس، تهران، ایران.

¹ Reliability Block Diagrams

² Architecture Styles

³ Design Patterns

⁴ Pipe and Filter

⁵ Implicit Invocation

⁶ Layered

⁷ Repository

⁸ Blackboard

⁹ Knowledge Resources

¹⁰ Dispatcher

¹¹ Broker

¹² Client-Side Proxy

¹³ Server-Side Proxy

¹⁴ Object Oriented

¹⁵ Characteristics

¹⁶ Maturity

¹⁷ Induced

¹⁸ Analytic Hierarchy Process

¹⁹ Air Traffic Control

²⁰ Hard Real-Time

Computer Science, 1997.

[17] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, Addison-Wesley, 1998.

[18] IEEE std 610.12-1990 (n.d.), IEEE Standard Glossary of Software Engineering Terminology, 1990.

[19] J. A. McCall, P. K. Richards, and G. F. Walters, "Factors in Software Quality," *Nat'l Tech. Information Service*, Vol. 1, 2 and 3, 1977.

[20] B. W. Boehm, J. R. Brown, H. Kaspar, M. Lipow, G. McLeod, and M. Merritt, *Characteristics of Software Quality*, North Holland, 1978.

[21] ISO, "ISO 9126-1:2001, Software engineering—Product quality, Part 1: Quality model," 2001.

[22] K. Goseva-Popstojanova, and K. S. Trivedi, "Architecture-based approach to reliability assessment of software systems," *Performance Evaluation*, Vol 45, pp. 179–204, 2001.

[23] R. Tripathi, and R. Mall, "Early Stage Software Reliability and Design Assessment," *Proc. of the 12th Asia-Pacific Software Engineering Conference*, pp. 619–628, 2005.

[24] S. P. Leblanc, and P. A. Roman, "Reliability Estimation of Hierarchical Software Systems," *Proc. of Annual Reliability and Maintainability Symposium*, pp. 249–253, 2002.

[25] A. Berman, and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press Inc., New York, 1979.

[26] W. Wang, and M. Chen, "Heterogenous Software Reliability Modeling," *Proc. of the 13th International IEEE Symposium on Software Reliability Engineering*, pp. 41–52, 2002.

[27] W. Wang, D. Pan, and M. Chen, "Architecture-based software reliability modeling," *Journal of Systems and Software*, Vol. 79, No. 1, PP. 132–146, 2006.



غلامرضا شاه محمدی دانشجوی دوره دکترای مهندسی کامپیوتر گرایش نرم‌افزار دانشکده مهندسی برق و کامپیوتر دانشگاه تربیت مدرس می‌باشد. نامبرده در طراحی و ارزیابی معماری نرم‌افزار مبتنی بر سبک‌های معماری پژوهش می‌کند. وی کارشناسی‌ارشد و کارشناسی خود را در رشته مهندسی کامپیوتر گرایش نرم‌افزار به ترتیب از دانشگاه تربیت مدرس و دانشگاه فردوسی مشهد اخذ نموده است. زمینه‌های تحقیقاتی مورد علاقه ایشان ارزیابی کمتی معماری نرم‌افزار، طراحی مبتنی بر سبک معماری نرم‌افزار، روش‌های مبتنی بر سنجش برای ارزیابی سیستم، کاربرد روش‌های رسمی در مهندسی نرم‌افزار و تخمین هزینه سیستم می‌باشد.

آدرس پست‌الکترونیکی ایشان عبارت است از:

shahmohamadi@modares.ac.ir