

سبک‌های معماری نرم‌افزار

سپیده عظیمی^{۱*}، گلناز آقایی قزوینی^۲

دانشجویی کارشناسی ارشد، مهندسی نرم‌افزار، دانشگاه آزاد اسلامی واحد

اصفهان (خوراسگان)، ایران

Sepidehazimi@yahoo.com

دکتر، عضو هیات علمی، دانشگاه آزاد اسلامی واحد دولت آباد، ایران

golnazghazvini@yahoo.com

چکیده

پیچیدگی نرم‌افزارها روز به روز در حال افزایش است و تقاضا برای نرم‌افزارهای قدرتمندتر بیشتر شده است. معماری نرم‌افزار، یکی از مباحث مهم حوزه مهندسی نرم‌افزار است و تحلیل معماری نرم‌افزار، یکی از موضوعات اصلی در این حوزه است. در فرایند طراحی معماری، تصمیمات متعددی اتخاذ می‌شود که تاثیر این تصمیمات در سطح سیستم نرم‌افزاری می‌باشد. یکی از این تصمیم‌ها، انتخاب سبک معماری مناسب است. برخی از روش‌ها یا معماری‌ها، توسط معماران نرم‌افزار، بارها برای سیستم‌های خاص ارائه شده‌اند و به دفعات مورد استفاده قرار گرفته‌اند و کاربردها و توانائی‌های خاص آن‌ها تأیید شده است و مجموعه‌ای از معماری‌ها یا الگوهای معماری تشکیل شده است که معماران را در ارائه معماری‌ها، یاری می‌کنند. هر یک از معماری‌های این مجموعه را یک سبک معماری می‌گویند. در فرایند طراحی، از سبک‌های معماری نرم‌افزار، که راه‌حل‌های آزموده شده‌ای می‌باشند، زیاد استفاده می‌شود. با توجه به اهمیتی که شاخه نسبتاً نوظهور معماری نرم‌افزار در مهندسی نرم‌افزار روز دنیا پیدا کرده است، در این مقاله روش‌ها و سبک‌های موجود و متداول معماری نرم‌افزار مورد بررسی قرار می‌گیرد.

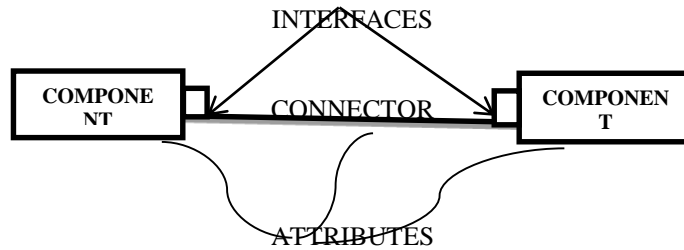
کلمات کلیدی: معماری نرم‌افزار، روش‌های ارزیابی نرم‌افزار، تحلیل نرم‌افزار، سبک‌های معماری نرم‌افزار

۱. مقدمه

معماری مجموعه‌ای از طرح‌ها و تولیدات متفاوت است که سیستم را از جنبه‌های مختلف نشان می‌دهند و هزاران تصمیم ریز و درشت که باید گرفته شوند. برخی از این تصمیم‌ها در آغازین مراحل طراحی گرفته می‌شوند و بر روی سایر عملیات تاثیر گزار خواهند بود. تصمیم‌گیری در مورد برخی دیگر به بعد موقوف می‌شود. معماری نرم‌افزار شاخه‌ای از مهندسی نرم‌افزار است که با استفاده از آن ساخت نرم‌افزارهای بزرگ و پیچیده ساده شده، هزینه‌های تولید و نگهداری کاهش یافته، کیفیت محصول ارتقاء می‌یابد و زمینه‌های مناسب برای توسعه نرم‌افزار ایجاد می‌شود. معماری نرم‌افزار از کلیدی‌ترین بخش‌های تولید نرم‌افزار است. در واقع انتخاب یک ساختار کلی برای پیاده‌سازی یک پروژه نرم‌افزاری بر مبنای مجموعه‌ای از نیازهای کاربری یک سیستم نرم‌افزاری است تا هم بتوان کاربردهای مورد نظر را پیاده‌سازی کرد و هم بتوان کیفیت نرم‌افزار، تولید و نگهداری آن را بهینه کرد و سرعت بخشید [۴-۵].

۲. اجزاء معماری نرم‌افزار

مهمترین عناصر و اجزاء معماری نرم‌افزار، مؤلفه‌ها و اتصال‌دهنده‌ها هستند که هر کدام از مؤلفه‌ها و اتصال‌دهنده‌ها با خصوصیات خاص خود، با یکدیگر در تعاملند و در قالب یک پیکربندی مشخص، معماری یک سیستم نرم‌افزاری را شکل می‌دهند [۱].



شکل (۱) اجزاء معماری نرم افزار

• مؤلفه^۱: مؤلفه‌ها به عنوان بلوک‌های اولیه و موجودیت‌هایی محاسباتی در ساخت سیستم شرکت کرده و از طریق محاسبات داخلی و ارتباطات خارجی خود کارها را انجام می‌دهند. یک مؤلفه از طریق یک یا بیشتر درگاه با محیط ارتباط برقرار می‌کند. این درگاه می‌تواند یک واسط کاربر، یک متغیر مشترک، یک نام رویه که از یک مؤلفه دیگر صدا زده شده است، یک مجموعه از رویدادها که می‌توانند از یک مؤلفه ایجاد گردند و مکانیزم‌های دیگر باشد. خصوصیات یک مؤلفه، اطلاعاتی را برای تحلیل و پیاده‌سازی نرم‌افزار مشخص می‌کند.

• اتصال‌دهنده^۲: اتصال‌دهنده‌ها تعاملات بین مؤلفه‌ها را تعریف کرده و قواعد حاکم بر آن‌ها را توصیف می‌نمایند. یک اتصال‌دهنده درگاه‌های دو یا چند مؤلفه را بهم متصل می‌نماید. خصوصیات مختلفی نیز به آن‌ها نسبت داده می‌شود، به عنوان مثال سرعت، ظرفیت و میزان عکس‌العمل اتصال‌دهنده از جمله این ویژگی‌ها هستند.

• واسط^۳: زمانی که اتصال‌دهنده بین دو مؤلفه ارتباط برقرار می‌کند، مؤلفه یک واسط تعریف می‌نماید و هر مؤلفه می‌تواند چند واسط داشته باشد. یک واسط تنها به یک مؤلفه مربوط می‌شود و هر واسط یک مؤلفه، می‌تواند به چندین واسط در مؤلفه‌های دیگر وصل شود.

• پیکربندی^۴: پیکربندی که گاهی تحت عنوان توپولوژی نیز از آن یاد می‌شود گراف همبندی است که از مؤلفه‌ها و اتصال‌دهنده‌ها تشکیل شده است و ساختار معماری را توصیف می‌نماید.

۳. سبک معماری نرم‌افزار

واژه سبک معماری برای اولین بار توسط پری و ولف در سال ۱۹۹۲ معرفی شد [۸]. یک سبک معماری یک خانواده از سیستم‌های نرم‌افزاری را با توجه به سازماندهی ساختاری آن‌ها تعریف می‌کند. همچنین مؤلفه‌ها و ارتباطات میان آن‌ها، به همراه محدودیت‌های استفاده و قواعد طراحی و ترکیب آن‌ها را بیان می‌نماید و انواع مؤلفه‌ها و نحوه چیدمان آن‌ها که شامل توصیف الگوی تعامل داده و کنترل بین مؤلفه‌ها است. همچنین یک توصیف غیررسمی از مزایا و معایب خود را در صورت استفاده از آن ارائه می‌نماید.

۱-۳- معرفی برخی از سبک‌های متداول در معماری نرم‌افزار

سبک‌های معماری نرم‌افزار به دو گروه یک‌ریختی و چندریختی تقسیم می‌شوند که در ادامه هر یک به طور جداگانه بررسی می‌شود [۷-۳].

¹ Component

² Connector

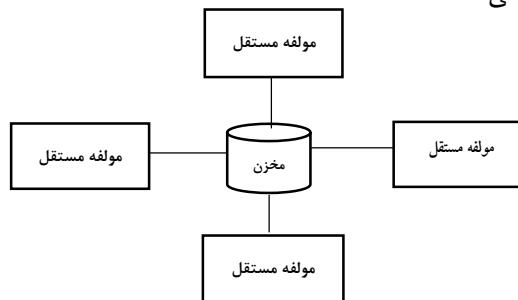
³ Interfaces

⁴ Configuration

۱-۱-۳- سبک های یکریختی

❖ سبک مخزن: (RPS) ۱

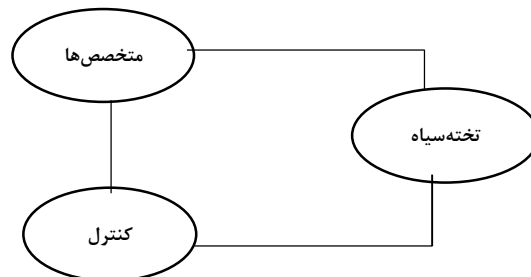
سبک مخزن شامل یک انباره مرکزی به نام مخزن و مجموعه ای از مولفه ها است که اعمال ذخیره بازیابی و بروزرسانی اطلاعات را بر روی انباره مرکزی انجام می دهند.



شکل ۲) سبک مخزن

❖ سبک تخته سیاه ۲ (BKB)

مولفه های این سبک، تخته سیاه، منابع دانش^۳ (متخصص ها) و مولفه کنترل است. مولفه کنترل یک حلقه را اجرا و ضمن نظارت بر تغییرات تخته سیاه و ارزیابی متخصص ها، یکی از آن ها را برای اجرا فعال می کند. بنابراین انجام یک تراکنش در این سبک مستلزم اجرای برنامه مولفه کنترل و بررسی وضعیت مولفه تخته سیاه توسط مولفه کنترل و انتخاب متخصص مناسب و تعامل متخصص با تخته سیاه است. منابع دانش به طور مستقیم با یکدیگر ارتباط برقرار نمی کنند و انتقال اطلاعات میان آن ها از طریق تخته سیاه ممکن می شود. معمولاً نحوه به کارگیری تخته سیاه توسط KS ها، کاملاً فرصت طلبانه است و به محض اینکه تخته سیاه به حالت آزاد رفت هر کدام از KS ها که بخواهند از روی آن خوانده، و یا بر روی آن اطلاعاتی را بنویسند، می کوشند تا تخته سیاه را به خدمت بگیرند، که از این میان تنها یکی موفق می شود و بقیه می بایست تا آزاد شدن مجدد تخته سیاه صبر کنند.



شکل ۳) سبک تخته سیاه

اما در مورد مدل فعال سازی، کلا سه جزء از سیستم ممکن است بدانند کنترل اجرا به چه ترتیبی است :

- خود تخته سیاه: تخته سیاه نحوه انتقال کنترل را به عنوان یک داده کنترلی مشترک در اختیار همه منابع دانش گذاشته و آن ها هم طبق توافق از این مدل کنترلی پیروی می کنند.
- منابع دانش: هر منبع می داند زمانی که با تخته سیاه کار دارد و همچنین تخته سیاه در وضعیت آزاد به سر می برد باید آن را در اختیار بگیرد.

¹ Repository style

² Blackboard style.

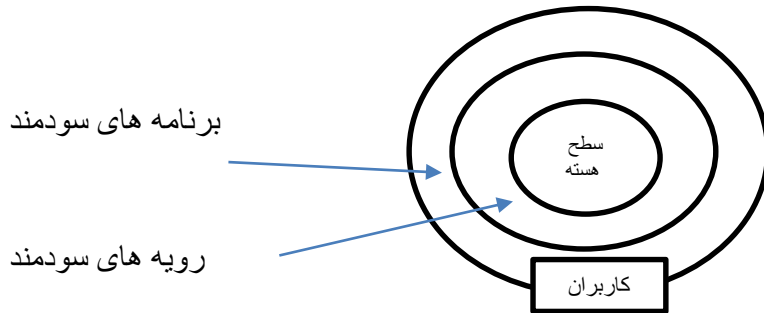
³ Knowledge Source (KS)

سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

- یک منبع خارجی به سیستم اضافه می شود که وظیفه آن اعمال مکانیزم کنترلی روی KS هاست.

❖ سبک لایه ای (LYD) ۱

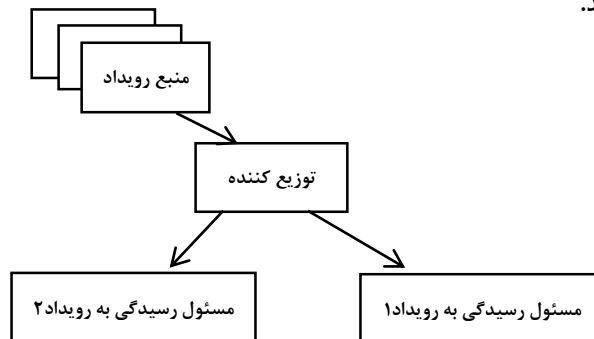
در این سبک، تمرکز روی سطوح مختلف انتزاع در نرم افزار است. سبک لایه ای بصورت سلسله مراتبی سازمان دهی می شود. هر لایه خدمتی را برای لایه بالاتر فراهم و از خدمت لایه پایین تر استفاده می کند. در انجام یک تراکنش کلیه لایه ها موثر است.



شکل ۴) سبک لایه ای

❖ سبک فراخوانی ضمنی (I/I) ۲

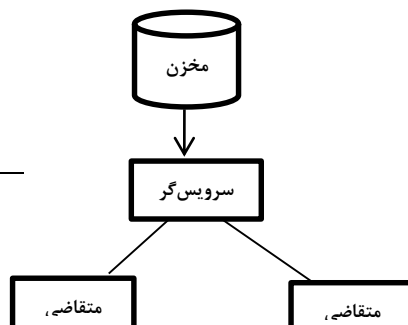
این سبک شامل تعدادی مولفه است و یک مولفه در نقش هماهنگ کننده یا توزیع کننده، با وقوع رویداد، از طریق جدول نگاشت رویداد به مولفه علاقه مند مراجعه، ضمن تعیین مولفه علاقه مند، پیامی به او ارسال یا رویه ای از آن مولفه را فراخوانی می کند. تراکنش در این سبک با وقوع رویداد شروع شده و مولفه توزیع کننده، مولفه علاقه مند را فعال می کند. با اتمام فعالیت مولفه علاقه مند، تراکنش خاتمه می یابد.



شکل ۵) سبک فراخوانی ضمنی

❖ سبک سرویس دهنده / سرویس گیرنده (C/S) ۳

مولفه های این سبک، سرویس دهنده، سرویس گیرنده و مخزن است. متقاضیان برای استفاده از خدمات سرویس گرها، باید از نام و خدمات ارائه شده توسط آن ها اطلاع داشته باشند. تراکنش در این سبک با ارسال درخواست متقاضی به سرویس گر آغاز می شود و به طور معمول نیاز به مراجعه سرویس گر به مخزن دارد و در نهایت با ارسال نتیجه توسط سرویس گر به متقاضی، به پایان می رسد.

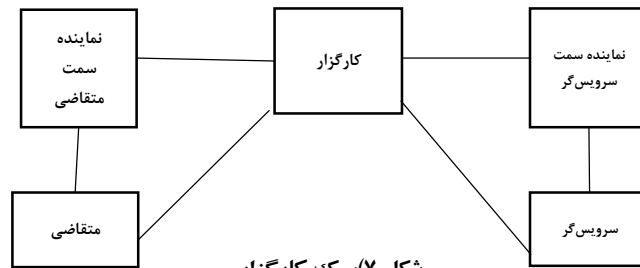


¹ Layered style
² Implicit invocation style
³ Server / client Style

شکل ۶) سبک متقاضی / سرویس گر

❖ سبک کارگزار (BRK) ۱

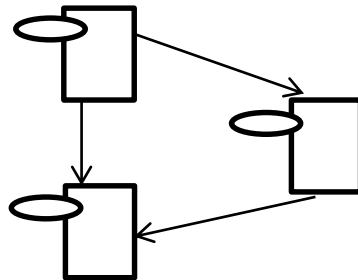
مولفه های این سبک، متقاضیان، سرویس گرها، کارگزار، نماینده سمت متقاضی، نماینده سمت سرویس گر و مخزن است. کارگزار مسئول هماهنگی ارتباط و مجزا کردن متقاضیان از سرویس گرها می باشد. سرویس گرها خود را به کارگزار معرفی و سرویس های خود را از طریق واسطه متد، در دسترس متقاضیان قرار می دهند. متقاضیان از خدمات سرویس گرها از طریق کارگزار استفاده می کنند. کارگزار، محل سرویس گر مناسب را تعیین، درخواست را به او ارسال، و نتیجه را به متقاضی برمی گرداند. تراکنش در این سبک مشابه سبک متقاضی / سرویس گر است، اما مولفه های بیشتری در آن موثر هستند.



شکل ۷) سبک کارگزار

❖ سبک شی گرا: (OO) ۲

در سبک شی گرا نمایش داده و عملیات مرتبط با آن در یک شی محصور می شود. مولفه های این سبک، اشیاء و تعامل اشیاء از طریق فراخوانی تابع است. تراکنش در سبک شی گرا به سناریو معروف است. هر سناریو زنجیری از اجرای متدی از تعدادی شی را شامل می شود.



شکل ۸) سبک شی گرا

❖ سبک جریان داده: ۳

در این سبک، به سیستم به این دید نگاه می شود که یک سری از تبدیلات روی قطعاتی متوالی از داده های ورودی انجام می شود. داده، به سیستم وارد می شود و در میان مؤلفه ها جریان می یابد. تا زمانی که به مقصد نهایی برسد و این مقصد نهایی می تواند ذخیره ای از اطلاعات، یا یک خروجی باشد.

¹ Broker style

² Object oriented style

³ Light streams style

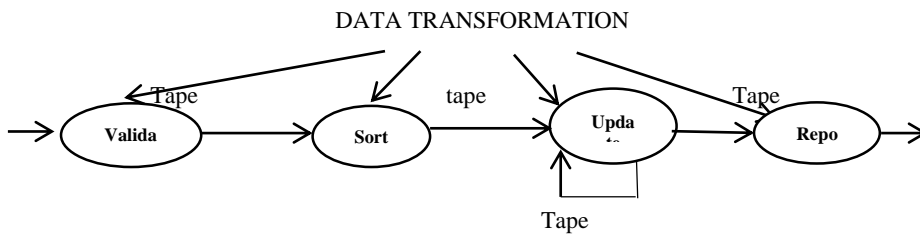
سومین همایش ملی مهندسی کامپیوتر، داده کاوی و داده های حجیم

این سبک شامل دو زیر سبک است: سبک ترتیبی و سبک لوله و فیلتر .

❖ سبک ترتیبی: ۱

در این سبک، مولفه‌ها به صورت ترتیبی اجرا می‌شوند، به طوری که در هر لحظه‌ای از زمان، تنها یک مولفه در حال اجرا است و پس از تمام شدن کار آن مولفه به مولفه بعدی منتقل می‌شود .

در سبک ترتیبی گام‌های پردازش یا همان مؤلفه‌ها، برنامه‌های مستقلی هستند و فرض بر این است که هرگام قبل از شروع گام بعد، اجرائیش به تکامل می‌رسد. هر دسته‌ای از داده‌ها، بین گام‌ها انتقال داده می‌شوند. فرم کلی برنامه در این سبک، فرم کلاسیک پردازش داده است.

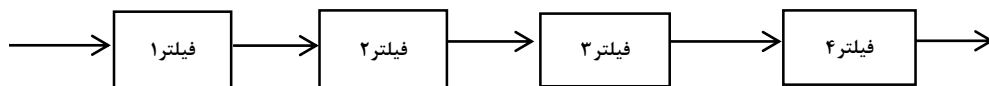


شکل ۹) سبک ترتیبی

❖ سبک لوله و فیلتر: (P/F) ۲

این سبک از تعدادی مولفه تشکیل شده است. هر مولفه در نقش یک فیلتر است و تعدادی ورودی و تعدادی خروجی دارد. خروجی هر مولفه، ورودی مولفه بعدی است. در انجام یک تراکنش کلیه فیلترها موثر است .

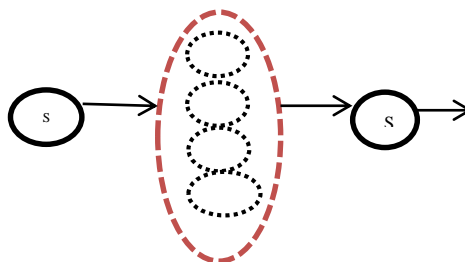
سبک لوله و فیلتر تأکید روی تبدیل و پردازش تدریجی روی داده بوسیله مؤلفه‌های متوالی دارد. این سبک یک سبک متداول در خانواده سیستم‌های عامل UNIX است. فیلترها در این سبک همان مؤلفه‌ها هستند و داده‌ها را به طور تدریجی تبدیل می‌کنند لوله‌ها همان اتصال دهنده‌ها هستند و حالتی به خود نمی‌گیرند و فقط برای حرکت دادن بین فیلترها قرار می‌گیرند قواعدی که روی این سبک حاکم‌اند، نحوه و چگونگی بستن لوله‌ها و فیلترها به هم را مشخص می‌کنند . یک لوله یک Source End دارد که به درگاه خروجی یک فیلتر متصل است و یک Sink End که به درگاه ورودی یک فیلتر متصل است.



شکل ۱۰) سبک لوله‌ها و فیلترها

❖ سبک موازی: ۳

معمولاً برای دستیابی به سرعت بالاتر در محاسبات، از سبک موازی استفاده می‌شود. این سبک دارای مولفه‌های متعدد در حال اجرا به صورت همزمان است. این مولفه‌ها، وقتی کار خود را گروهی و همزمان انجام دادند و به پایان رساندند، ادامه کار انجام می‌شود و به مولفه بعدی می‌روند.



¹ Batch Sequential style

² Pipe & Filter style

³ Parallel Style

شکل (۱۱) سبک موازی

۳-۱-۲- سبک‌های چندریختی

به غیر از سبک‌های یکریختی که مطرح شد، دسته دیگری از سبک‌ها وجود دارند که سبک‌های چندریختی نامیده می‌شوند. در اغلب سیستم‌ها، نمی‌توان گفت که از یک سبک منفرد برای ساخت استفاده شده و معمولاً سیستم‌ها چندریختی‌اند. امروزه با پیچیده‌تر شدن و بزرگتر شدن سیستم‌های نرم‌افزاری در عمل از بیش از یک نوع سبک خاص استفاده می‌شود و طراحی معماری شامل چندین سبک مختلف می‌شود. سبک‌های چندریختی در میان سبک‌های معماری از جایگاه ویژه و قابل توجهی برخوردارند، لذا ارزیابی معماری‌هایی که شامل سبک‌های چندریختی هستند، از اهمیت ویژه‌ای برخوردار است، زیرا هزینه تغییرات و اصلاحات در این سیستم‌ها بیشتر از سیستم‌های کوچک است. این چندریختی‌ها را می‌توان در انواع زیر بررسی کرد:

❖ چندریختی مکانی ۱

یعنی با داشتن شکل ساختار اجرایی، الگوهایی از سبک‌های مختلف در مکان‌های مختلف به چشم می‌خورند یا به عبارت دیگر سبک‌های مختلف در قسمت‌های متفاوتی از سیستم وجود دارند. این سبک متداول‌ترین نوع سبک‌های چندریختی به شمار می‌رود، اگر طراحی معماری مبتنی بر این سبک باشد، به این معنا است که الگوهایی از سبک‌های مختلف در مکانهای مختلف تعبیه شده است.

❖ چندریختی سلسله مراتبی ۲

یعنی یک مؤلفه از یک سبک با تقسیم شدن، طبق قواعد سبک‌های مختلفی ساخته شود. اگر طراحی سیستم مبتنی بر این سبک باشد، به این معنا است که معماری به صورت سلسله مراتبی از چندین سبک یکریختی ایجاد شده است. یک مؤلفه معماری در بالاترین سطح به صورت سلسله مراتبی و براساس اعداد سبکی مشخص ساخته می‌شود.



❖ چندریختی همزمان ۳

یعنی هر یک از انواع مختلف سبک‌ها را بتوان با تعریفی از سیستم منطبق ساخت. بدین معنا است که سبک‌ها، انواع معماری‌های نرم‌افزار را نمی‌توانند به دسته‌های کاملاً مجزا و بدون همپوشانی تقسیم کنند. به عبارتی دیگر مؤلفه‌های یک سبک ممکن است وجه مشترکی با سبک‌های دیگر داشته باشد.



شکل (۱۲) سبک چندریختی سلسله مراتبی

۴. نتیجه‌گیری

در این مقاله کلیاتی از تعاریف و مقدمات مربوط به معماری نرم‌افزار، بیان مفهوم و تعریف سبک معماری نرم‌افزار و دسته‌بندی‌های مختلف موجود از سبک‌ها و همچنین ارائه چند سبک معروف و متداول در جهت بیشتر آشنا شدن با مفهوم

¹ Locationally heterogeneous style

² Hierarchically Heterogeneous style

³ Simultaneously Heterogeneous style

سبک بررسی شد. از این توضیحات میتوان نتیجه گرفت که ارتباط نزدیکی میان مفاهیم طراحی معماری نرم افزار و سبکهای معماری وجود دارد. مطالعه سبکهای معماری و ساختار آنها این موضوع را روشن می کند که به کارگیری سبکها در طراحی، نیازمندیهای سیستم را تضمین می کند و باعث بهبود کارایی سیستم می شود. حال معمار به منظور استفاده صحیح سبکها در طراحی معماری نیاز به ارزیابی و تحلیل دقیق اثر هر سبک روی کیفیت سیستم دارد، این ارزیابی و تحلیل منجر به انتخاب سبک مناسب می شود و موفقیت طراحی معماری را تضمین میکند.

۵. مراجع

- [۱] هاشمیان ح. ۱۳۸۷. " مدلی برای بررسی ویژگیهای کیفی در سبکهای معماری چندریختی "، پایان نامه کارشناسی ارشد، دانشگاه آزاد اسلامی واحد علوم و تحقیقات.
- [۲] علی اکبری ع، رضایی ر. ۱۳۹۵. " مروری بر روشهای ارزیابی معماری نرم افزار "، مقاله کارشناسی ارشد مهندسی نرم افزار دانشگاه آزاد اسلامی واحد ساوه.
- [۳] شاه محمدی غ، جلیلی س. ۱۳۸۷. " ارزیابی کمی سبکهای معماری نرم افزار از دید قابلیت اعتماد در نرم افزارهای حساس به ایمنی "، نشریه علمی پژوهشی انجمن کامپیوتر ایران.
- [4] S. M. Sharafi, G. Aghaee Ghazvini, S. Emadi." An Analytical Model for Performance Evaluation of Software Architectural Styles", 2nd International Conference on Software Technology and Engineering, 2010, vol.1, pp. 394-398
- [5] S.S.Jalali, H.Rashidi and E.Nazemi." A New Approach to Evaluate Performance of Component-based Software Architecture", UKSim 5th European Symposium on Computer Modeling and Simulation, 2011, pp. 451– 456
- [6] X. Wu, L. Huang Chen Li, L. Chen." Estimating Software System Reliability: An Architecture-Based Approach", 9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 2015, pp. 184- 191
- [7] J. M. Franco, R. Barbosa, M. Zenha-Rela." Reliability Analysis of Software Architecture Evolution", Sixth Latin-American Symposium on Dependable Computing, 2013, pp.10-20
- [8] G.ShahMohammadi." EVALUATION OF THE SOFTWARE ARCHITECTURE STYLES FROM MAINTAINABILITY VIEWPOINT", Computer Science & Information Technology (CS & IT), 2014, pp. 183–197
- [9] N. Eftekhari, M. Poyan Rad, H. Alinejad-Rokny."Evaluation and Classifying Software Architecture Styles Due to Quality Attributes", Australian Journal of Basic and Applied Sciences, 2011, pp.1251-1256
- [10] Y.We, X.H.Shen1." Heterogeneous Architecture-Based Software Reliability Estimation: Case Study", International Conference on Convergence and Hybrid Information Technology, 2008, vol. 7, pp. 286- 290
- [11] N.Eftekhari, M.Poyan Rad, H.Alinejad-Rokny." Evaluation and Classifying Software Architecture Styles Due to Quality Attributes", Australian Journal of Basic and Applied Sciences, 2011, vol. 11, pp. 1251-1256