

## ارائه یک مدل تصمیم گیری انتخاب سبک معماری بر اساس معیار های چندگانه

سید امین سیدی، محمد تشنه لب ، میر علی سیدی

دانشگاه آزاد اسلامی واحد ماکو - گروه کامپیوتر و عضو باشگاه پژوهشگران جوان

دانشگاه خواجه نصیرالدین طوسی - دانشکده برق

دانشکده فنی واحد تهران جنوب - گروه کامپیوتر - نرم افزار

[amseyyedi@gmail.com](mailto:amseyyedi@gmail.com)

[teshnehlabb@eetd.kntu.ac.ir](mailto:teshnehlabb@eetd.kntu.ac.ir)

[maseyyedi@gmail.com](mailto:maseyyedi@gmail.com)

چکیده - در این مقاله به ارائه یک مدل تصمیم گیری انتخاب سبک معماری<sup>۱</sup> بر اساس معیارهای چندگانه<sup>۲</sup> پرداخته شده است. این مدل بر اساس تاثیر سبک های معماری بر خصوصیات کیفی<sup>۳</sup> و موتور استنتاج فازی طراحی و پیاده سازی شده است. از مهمترین ویژگی های این روش می توان به سادگی بکارگیری، دقت اندازگیری، کمی بودن ارزیابی، قابلیت مقایسه پذیری و قابلیت پیاده سازی بصورت یک مدل اتومیک سنچش اشاره کرد.

کلمات کلیدی: خصوصیات کیفی - سبک معماری - سیستم فازی - معماری نرم افزار<sup>۴</sup>

### ۱- مقدمه

مشخص، سبک مورد نظر را انتخاب می کند، راه حل ارائه شده دارای مراحل طولانی است و به دلیل محاسبات اضافی احتمال خطای محاسباتی در این روش بالا است. این مشکلات می تواند منجر به تشخیص نادرست سبک معماری مورد نظر شود. هدف این مقاله ارائه روشی است که مشکلات روش [۱۷ و ۱۳] را نداشته باشد.

در این مقاله اثر سبک های معماری بر خصوصیات کیفی مورد بررسی قرار گرفته و براساس آن یک سیستم استنتاج فازی پیاده سازی شده است.

در فرایند توسعه نرم افزار بعد از تکمیل شدن تحلیل نیازمندیها و بدست آوردن یک سند کامل شامل مشخصات نیازمندیها، معماری نرم افزار سیستم در حال توسعه، طراحی می شود. در این مرحله فعالیت های مربوط به معماری، نقش کلیدی و اصلی را بازی می کند و بصورت پلی بین مشخصات نیازمندیها و پیاده سازی در سیستم عمل می کند. کیفیت طرح و سبک معماری تاثیر زیادی در رسیدن به نیازمندیهای غیر کارکردی<sup>۵</sup> مانند کارایی و امنیت... دارد

### ۲- سبک معماری

لغت سبک معماری برای اولین بار توسط [۲] Wolf و Perry در سال ۱۹۹۲ معرفی شد. [۳] Garlan و Shaw در سال ۱۹۹۴ در نوشته ای سبکهای معماری نرم افزار را معرفی کرده و همراه با مثالهایی، آنها را مقایسه کردند. انگیزه اصلی در انتخاب نوعی از معماری در شرایط خاص، با توجه به استدلالها و مشخصه های کیفی شکل می گیرد ولی سبکها، دانش و خرد معماران ماهر را بازنمایی کرده و معماران کم تجربه را برای طراحی معماری هایشان راهنمایی می کنند.

اگر معماری پذیرفته شده توسط معمار، نیازمندیهای موجود را برآورده نسازد، سیستم با کیفیت پایین تولید خواهد شد. در نتیجه میزان زیادی هزینه و وقت به هدر خواهد رفت. چرا که سیستم باید مجددا توسعه یابد. برای اجتناب از تلف شدن سرمایه و زمان که بدلیل انتخاب سبک معماری نامناسب می باشد، توصیف یک معماری رسمی و معتبر که قابل ارزیابی باشد، قبل از شروع به پیاده سازی یک سیستم، می تواند بسیار مفید باشد. [۱]

Astudillo [۱۳ و ۱۷]، یک روش برای ارزیابی و انتخاب سبکهای معماری ارائه کرده است. این روش بر روی حوزه

سبک معماری از دیدگاه Garlan و Shaw [۲] در سال ۱۹۹۶ سبک معماری بدین صورت تعریف شده است:

یک سبک معماری توصیفی است از انواع مولفه‌ها و نحوه چیدمان آنها و توصیفی از الگوی تعامل داده و کنترل، بین مولفه‌ها و توصیفی رسمی از مزایا و محاسن هر یک، در بکارگیری آنها.

تعاریف دیگری نیز از سبک های معماری ارائه شده است. [۳و۵]

سبک های مختلف و طبقه بندی های مختلف ارائه شده ولی در این مقاله سبک های پایه مد نظر شده است که عبارتند از:

سبک متمرکز روی داده ها<sup>۶</sup> [۶]، سبک جریان داده<sup>۷</sup> [۷]، سبک ماشین مجازی<sup>۸</sup> [۸]، سیستم های فراخوانی روال های خارجی<sup>۹</sup> [۸]، شی گرای<sup>۱۰</sup> [۹] و سبک لایه ای<sup>۱۱</sup> [۸]

دسته بندی های مختلفی [۳و۵و۸] نیز برای سبک های معماری بر اساس نوع آنها ارائه شده است که اولین دسته بندی به صورت زیر می باشد .



شکل ۱- اولین دسته‌بندی موضوعی سبکهای معماری نرم افزار [۸]

### ۳- خصوصیات کیفی

ویژگی های کیفی بخشی از نیازمندیهای غیر عملیاتی هستند که مبنای طراحی معماری نرم افزار و در نتیجه انتخاب سبک معماری می باشند.

معماری اولین مرحله تولید نرم افزار بوده و ویژگی های کیفی می بایست در نظر گرفته شوند. ویژگی های کیفی در تمام مراحل طراحی، پیاده سازی و انتقال مطرح هستند. در نتیجه اگر توسط معماری پشتیبانی شوند، قابل ردیابی خواهند بود ویژگیهای کیفیتی از نظر ارزیابی به دو دسته تقسیم می شوند: [۹]

#### ۳-۱- صفات کیفیتی قابل مشاهده در زمان اجرا

این صفات نشان می دهند که در طول مدت اجراء یک سیستم چقدر خوب می تواند نیازمندیهای رفتاری خودش را تامین کند. صفات کیفیتی قابل مشاهده در زمان اجرا شامل موارد زیر است :

کارایی<sup>۱۲</sup>

امنیت<sup>۱۳</sup>

دردسترس بودن<sup>۱۴</sup>

قابلیت عملکرد یا وظیفه مندی<sup>۱۵</sup>

قابلیت کاربرد و استفاده<sup>۱۶</sup>

#### ۳-۲- صفات کیفیتی غیر قابل مشاهده در زمان اجرا

این صفات به گونه ای هستند که با اجرای آن نمی توان تشخیص داد که به آن دست یافته ایم یا خیر و به عبارت دیگر نمی توان آن را در زمان اجرا دید و باید بعداً ارزیابی شود. همه صفات کیفی در یک مقطع خودشان را نشان نمی دهند . بعضی در مرحله تحلیل، بعضی در مرحله طراحی و ... خود را نشان می دهند.

صفات کیفیتی غیر قابل مشاهده در زمان اجرا شامل موارد می باشند :

قابلیت اصلاح پذیری<sup>۱۷</sup>

قابلیت حمل<sup>۱۸</sup>

قابلیت تجميع پذیری<sup>۱۹</sup>

قابلیت استفاده مجدد<sup>۲۰</sup>

قابلیت آزمایش<sup>۲۱</sup>

البته هر دو دسته بندی اشاره شده جزء صفات کیفی سیستمی می باشند ولی یک سری ویژگیهای کیفیتی مرتبط با حرفه هم داریم، که آنها نیز در معماری اثرگذار هستند، که از آن جمله می توان به مسائلی مانند قیمت تمام شده، زمان پایان و ... اشاره کرد.

دو نکته در رابطه با صفات کیفیتی از اهمیت زیادی برخوردار است:

• یکسری ویژگیهای کیفی مربوط به معماری می باشند در

زمان طراحی معماری مورد استفاده هستند.

• برخی از این صفات کیفیتی وابسته به معماری نیستند و

تحقق در زمینه دستیابی به این کیفیتها در معماری کار

درستی نیست و نتیجه قابل استفاده ای ارائه نمی کند.

بحث صفات کیفیتی در سیستمهای پیچیده مطرح می شود و

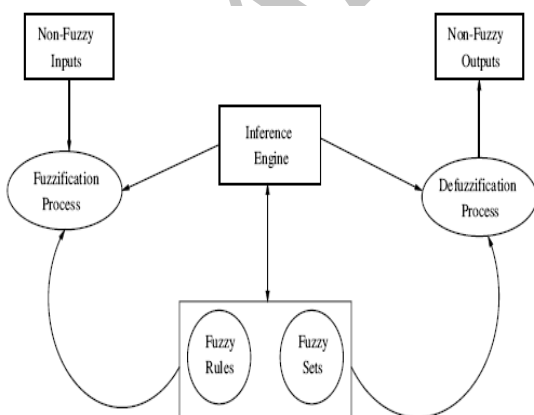
می دانند چه کاری انجام می دهند. ولی نمی توانند آن را به صورت دقیق در قالب کلمات بیان کنند [۱۱]. در توصیف دانش ناخودآگاه خبره رفتار خود را در وضعیت های مختلف نشان می دهد. در این حالت فرد خبره به عنوان یک جعبه سیاه در نظر گرفته می شود و ورودی ها و خروجی های او سنجیده می شوند. بدین ترتیب رفتار خبره به مجموعه ای از ورودی - خروجی تبدیل می شود. مسئله اصلی ساخت سیستم های فازی بر اساس زوجهای ورودی-خروجی است.

در طراحی سیستم های فازی انتخاب تعداد قواعد بسیار مهم است. انتخاب تعداد قواعد زیاد باعث پیچیده تر شدن سیستم فازی می شود که ممکن است برای حل مسئله ضروری نباشد. از طرفی دیگر انتخاب تعداد قواعد کم موجب ایجاد یک سیستم فازی ضعیف می شود

#### ۵-۲- طراحی سیستم فازی انتخاب سبک معماری

سیستم فازی ارائه شده، یک مدل تصمیم گیری برای انتخاب سبک معماری مناسب می باشد تا به معماران کم تجربه در انتخاب سبک مناسب کمک نماید. طراح درجه اهمیت ویژگیهای کیفیتی مورد نظر خود را به عنوان پارامتر ورودی به سیستم ارسال نموده و سیستم استنتاج فازی<sup>۲۲</sup> با استفاده از مجموعه قواعد فازی، در خروجی میزان مناسب بودن سبک مورد نظر را ارائه می کند. سبکی که دارای درجه تعلق بالایی باشد سبک انتخابی خواهد بود.

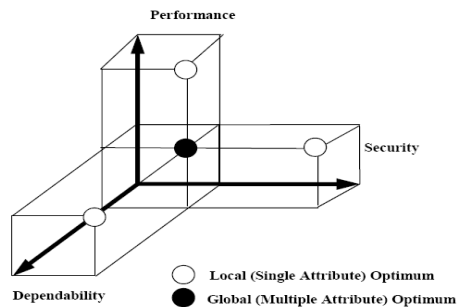
در سیستم فازی طراحی شده، ورودی های سیستم فازی خصوصیات کیفی مورد نیاز و خروجی میزان تناسب سبک های پایه معماری می باشد. در این مدل از تابع استنتاج فازی مددانی با موتور استنتاج ضرب<sup>۲۳</sup> و غیر فازی ساز میانگین مرکز<sup>۲۴</sup> استفاده شده است. [۱۲]



شکل ۳- سیستم استنتاج فازی [۱۲]

۵-۲-۱- ورودی- خروجی های سیستم فازی

برآورده کردن این صفات به صورت مجرد امکان پذیر نیست، اصولاً رسیدن به یک کیفیت روی کیفیت دیگر تاثیر می گذارد.



شکل ۲- مصالحه میان صفات کیفیتی نرم افزار [۱۰]

یکسری مدل های کیفیتی برای مشخص نمودن و ارزیابی خصوصیات کیفی ارائه شده اند که این مدل ها عبارتند از: مدل Boehm, Cavano, McCall و مدل استاندارد ISO و ... [۱۰]

#### ۴- اثر سبک ها بر صفات کیفی [۱۰ و ۷ و ۱۵ و ۱۶ و ۸]

در جدول شماره ۱ اثر سبک های متداول بر روی ویژگی های کیفی شامل کارایی، امنیت، قابلیت تغییر، قابلیت استفاده مجدد، مقیاس پذیری، قابلیت حمل و قابلیت اطمینان ارائه شده است:

| نام سبک               | Performance | Security | Modifiability | Reusability | Scalable | Portability | Reliability |
|-----------------------|-------------|----------|---------------|-------------|----------|-------------|-------------|
| Data Center           | مثبت        | منفی     | مثبت          | بی تاثیر    | مثبت     | بی تاثیر    | منفی        |
| Data Flow             | منفی        | بی تاثیر | مثبت          | مثبت        | منفی     | مثبت        | مثبت        |
| Virtual Machine       | منفی        | مثبت     | بی تاثیر      | مثبت        | مثبت     | مثبت        | بی تاثیر    |
| Remote Procedure Call | مثبت        | بی تاثیر | مثبت          | بی تاثیر    | مثبت     | بی تاثیر    | بی تاثیر    |
| Object Oriented       | بی تاثیر    | بی تاثیر | مثبت          | مثبت        | مثبت     | مثبت        | مثبت        |
| Layered               | منفی        | مثبت     | مثبت          | بی تاثیر    | مثبت     | مثبت        | بی تاثیر    |

جدول ۱- تاثیر سبک ها بر صفات کیفی

به عنوان مثال با توجه به خصوصیات سبک Data Center این سبک مقیاس پذیر، تغییر پذیر و دارای زمان پاسخ مناسب می باشد. [۶]

#### ۵- سیستم فازی انتخاب سبک های معماری

##### ۵-۱- طراحی سیستم های فازی

سیستم های فازی به منظور مدلسازی دانش بشری در مورد یک مسئله خاص استفاده می شوند. بطور کلی دانش بشری را به دو دسته خود آگاه و ناخود آگاه می توان تقسیم کرد. دانش خود آگاه بصورت صریح و روشن در قالب کلمات قابل توصیف است. دانش ناخود آگاه شامل وضعیت هایی است که انسان های خبره

## ۶- مثال

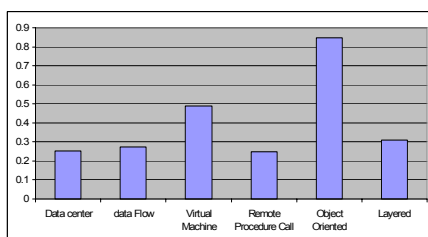
مدل تصمیم‌گیری ارائه شده در این مقاله یک مدل تصمیم‌گیری برای کمک به معماران کم تجربه برای انتخاب صحیح و درست سبک معماری براساس پارامترهای کیفی می‌باشد. همانگونه که در قسمت مقدمه ارائه شد، روش ارزیابی سبک‌های معماری Astudillo [۱۳ و ۱۷] براساس مجموع قدر مطلق تفاضلات<sup>۲۶</sup> بین صفات کیفی برنامه مورد نظر و مقادیر کیفی هر سبک پیاده‌سازی شده است، چنانچه تنها یک سبک کوچکترین مقدار مجموع قدر مطلق تفاضلات را داشته باشد سبک مورد نظر همان است، در غیر اینصورت با وجود بیش از دو مقدار مساوی، مجموع مربعات تفاضلات<sup>۲۷</sup> را بین مقادیر صفات کیفیتی برنامه مورد نظر و مقادیر صفات کیفیتی برای هر سبک محاسبه نموده، سپس ضریب همبستگی بین مقادیر صفات کیفیتی برنامه مورد نظر و مقادیر صفات کیفیتی هر سبک که مقدار SSD آن کمترین شده است را بدست آورده سبکی را انتخاب می‌کنیم که بیشترین ضریب همبستگی را با مقادیر صفات کیفیتی برنامه مورد نظر داشته باشد. اگر بیش از یک سبک، بیشترین ضریب همبستگی را داشتند، سبکی را انتخاب می‌نمائیم که مقدار صفت کیفیتی آن با صفات کیفیتی مهم برنامه مورد نظر یکسان باشد،

حال به مقایسه بین این روش و مدل ارائه شده می‌پردازیم.

در صورتی که فرض کنیم طبق نمونه ارائه شده در [۱۳ و ۱۷]، میزان اهمیت صفات کیفی یک سیستم نرم افزاری طبق جدول شماره ۲ باشد.

| Performance | Security | Modifiability | Reusability | Scalable | Portability | Reliability |
|-------------|----------|---------------|-------------|----------|-------------|-------------|
| مثبت        | بی‌تاثیر | مثبت          | مثبت        | مثبت     | بی‌تاثیر    | مثبت        |

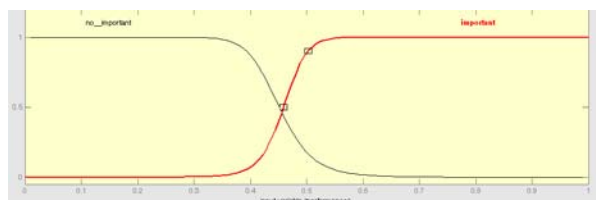
جدول ۲- میزان اهمیت صفات کیفی سیستم نرم افزاری طبق روش H. Astudillo [۱۳] سبک انتخابی، سبک سیستم‌های شی‌گرایی می‌باشد. نتایج بدست آمده از اجرا سیستم فازی ارائه شده، طبق شکل ۷ بیشترین درجه عضویت را با توجه به پارامترهای ورودی داده شده، برای سبک سیستم شی‌گرایی ارائه می‌کند.



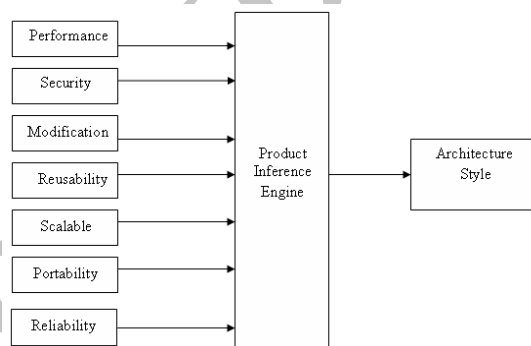
شکل ۷- نتیجه اجرا سیستم فازی بر روی نمونه

ورودی‌های سیستم فازی که شمای کلی آن در شکل ۵ نشان داده شده است، ویژگی‌های کیفی اشاره شده در قسمت ۳ هستند و دامنه مقادیر آنها بصورت بیانی<sup>۲۵</sup> "مهم" یا "غیر مهم" می‌باشند.

این ورودی‌ها نشان‌دهنده میزان اهمیت هر یک از ویژگی کیفی است. [۱۲]

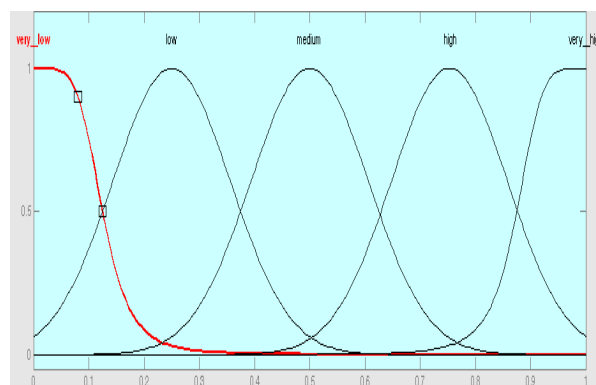


شکل ۴- دامنه مقادیر بیانی ورودی‌های سیستم فازی



شکل ۵- شمای کلی سیستم فازی

خروجی‌های سیستم فازی نشان‌دهنده درجه مناسب بودن سبک معماری کاندید است که دامنه مقادیر آن بصورت شکل ۶ است [۱۲]



شکل ۶- خروجی‌های سیستم فازی

## ۵-۲-۲- قواعد سیستم استنتاج فازی

مجموع قواعد فازی برای سیستم استنتاج فازی با توجه به تاثیر سبک‌های معماری بر روی ویژگی‌های کیفی و میزان اهمیت داشتن آن ویژگی از نظر طراح ایجاد می‌شود.

[4] F. B. Paul Clements, Len Bass, David Garlan, James Ivers, Reed Little, Robert Nord, Judith Stafford, Documenting Software Architectures: Views and Beyond, Addison Wesley, ۲۷, September, ۲۰۰۲.

[5] H. Astudillo, Five Ontological Levels to Describe and Evaluate Software Architecture 2003

[6] L. B. Paul Clements, Rick Kazman, Software Architecture in Practice, Addison Wesley, ۱۱ April ۲۰۰۳

[7] R. Land, Architecture Solutions in PAM, ۲۰۰۱.

[8] R. A. Jeff Garland, Large-Scale Software Architecture, ۲۰۰۳.

[9] M. H. K. Mario Barbacci, Quality Attributes (December 1995)

[10] H. Astudillo, Five Ontological Levels to Describe and Evaluate Software Architecture

[11] Gregory E. Kersten, Stan Szpakowicz, 1994 "Negotiation In Distributed Artificial Intelligence, IEEE "

[12] Computational Intelligence Andirs. Engelbrecht University of Pretoria South Africa 2007

[13] H. Astudillo, Five Ontological Levels to describe and Evaluate Software Architecture 2003

[14] U. P. Schultz, Architectural Patterns, (۲۰۰۴).

[15] D. P. Wen-Li Wang, Mei-Hwa Chen, Architecture-based Software Reliability Modeling, Elsevier (September ۲۰۰۵).

[16] Composing Style- Based Software Architectures From Architectural Primitives by Nikunj Rohit Kum.ar Mehta 2005

[17] پایان نامه کارشناسی ارشد دانشگاه آزاد اسلامی.

هاشمیان

در روش Astudillo [۱۳] که فقط بر روی حوزه مشخص، سبک مورد نظر را انتخاب می کند دارای مراحل طولانی می باشد و داشتن محاسبات اضافی احتمال خطای محاسباتی را افزایش می دهد که منجر به تشخیص نادرست سبک مورد نظر معمار می شود. اما مدل فازی ارائه شده سریعتر، ساده تر و با احتمال خطای کمتر انجام می شود.

## ۷- نتیجه گیری

مدل ارزیابی ارائه شده به معماران کم تجربه در انتخاب سبک معماری مطلوب کمک می نماید و این امر موجب طراحی معماری مناسب و متناسب با نیازمندیهای یک سیستم نرم افزاری خواهد بود.

در صورتی که کنترل های فازی مورد استفاده در این مدل قابلیت یادگیری و توسعه پایگاه قواعد خودشان را داشته باشند. محدودیت زمان نیز از نظر تغییر نگرش توابع عضویت و دانش موجود در پایگاه قواعد مرتفع می شود. چرا که با تغییر زمان دانش پایگاه قواعد نیز رشد می کند. این امر باعث تنظیم و بهینه نمودن قواعد و توابع عضویت و در نتیجه ارتقاء دقت مدل گردد. یکی دیگر از زمینه های بهبود این مدل بررسی تغییر پارامترهای ورودی بیانی به پارامترهای کمی است.

## مراجع

[1] Documenting Software Architecture, Paul Clements, Felix Bachmann, Len Bass, David Garlan, James Ivers, Reed Little, Robert Nord, Judith Stafford, 2002

[2] M. S. David Garlan, an Introduction to Software Architecture, (January ۱۹۹۴)

[3] M. S. Douglas Schmidt, Hans Rohnert, Frank Buschmann, Pattern-Oriented Software Architecture: A system of Patterns, John Wiley & Sons, ۱۹۹۶.

- 
- 1 Architecture Style  
2 Multi Criteria Decision Making  
3 Quality Attribute  
4 Software Architecture  
5 Nonfunctional Requirement  
6 Data Center Style  
7 Data Flow Style  
8 Virtual Machine  
9 Remote Procedure Call Style  
10 Object Oriented Style  
11 Layerd Style  
12 Performance  
13 Security  
14 Availability  
15 Functionality  
16 Useability  
17 Modification  
18 Portability  
19 Integration  
20 Reusability  
21 Testability  
22 Inference Fuzzy System  
23 Product Inference Engine  
24 Center average Defuzzifier  
25 Linguistic  
26 Sum of Absolute Differences (SAD)  
27 Sum of Squares of Differences (SID)