



یک چارچوب توسعه برای کاربردهای رایانش ابری مبتنی بر الگو و زبان الگو

علیرضا روحی

استادیار دانشکده‌ی فناوری اطلاعات و مهندسی کامپیوتر، دانشگاه شهید مدنی آذربایجان*

چکیده

رایانش ابری، روش‌های استفاده از منابع فناوری اطلاعات را به‌طور چشم‌گیری دگرگون ساخته است. با پیچیده‌تر شدن توسعه‌ی نرم‌افزارهای مستقل، این پیچیدگی در توسعه‌ی کاربردهای مبتنی بر ابر نیز بیش از پیش خودنمایی می‌کند. یکی از بهترین مشق‌های حل یک مسئله در حالت کلی، اعمال الگوها به‌شکل مجموعه‌ای از الگوهای مرتبط در قالب یک زبان الگوست. هرچند با رشد روزافزون تعداد الگوها و روابط میان آن‌ها در زبان‌های الگو، و دشواری ذاتی کار بررسی دستی زبان‌های الگوی اعمال‌شده، تدارک ابزار پشتیبانی طراحی مبتنی بر زبان الگو ضروری به‌نظر می‌رسد.

این مقاله به‌دنبال ارائه‌ی یک چارچوب توسعه برای اعمال الگوها و زبان‌های الگو در کنار واری‌آنها در زمینه‌ی رایانش ابری است.

کلیدواژه‌ها: رایانش ابری، الگو، زبان الگو، واری‌آنها، و طراحی مبتنی بر الگو و زبان الگو

تاریخچه مقاله:

تاریخ ارسال: ۹۷/۳/۱۵

تاریخ اصلاحات: ۹۷/۴/۳۰

تاریخ پذیرش: ۹۷/۵/۵

تاریخ انتشار: ۹۷/۵/۱۵

Keywords:

Cloud computing

Pattern

Pattern language (PL)

Pattern language verifier

Pattern and PL-based design

A development framework for pattern and pattern language-based cloud computing applications

Alireza Rouhi

Faculty of Information Technology and Computer Engineering, Azarbaijan Shahid Madani University

Abstract

Cloud computing has dramatically changed the way of information technology usage. With increased complexity of stand-alone application developments, this complexity is also evolving in the development of cloud-based applications. One of the best practices to solve a problem as a whole is to apply patterns as a set of related patterns in the form of a pattern language (PL). However, with the growing number of patterns and their inter-relationships in PLs and the inherent difficulties of manual verification of the applied PLs, it seems necessary to provide tools for supporting a PL-based design.

This paper aims to present a development framework for applying patterns and PLs as well as their verification in the cloud computing context.

ع.روحی، یک چارچوب توسعه برای کاربردهای رایانش ابری مبتنی بر الگو و زبان الگو، دوفصلنامه محاسبات و سامانه‌های توزیع شده، سال اول، شماره اول، ص ۹۰-۷۲، سال انتشار ۱۳۹۷.

روش ارجاع به مقاله:

* Corresponding author. Tel: +98 41 31452094

E-mail address: rouhi@azaruniv.ac.ir (A. Rouhi)



۱- مقدمه

توسعه‌دهندگان کاربردهای مبتنی بر ابر

به‌منظور بهره‌برداری از ظرفیت کامل چنین محیطی لازم دارند تا از اصول معماری و کارکردهای خاص کاربردهای مبتنی بر ابر استفاده کنند. لذا، یک کاربرد مستقل موجودی که روی یک ماشین منفرد اجرا دارد، با مهاجرت ساده به روی ابر به حداقل منافع منجر شده و حتی ممکن است دسترس‌پذیری کاربردها را نیز در صورت عدم رعایت درست اصول معماری پایین بیاورد. در صورتی که، با توزیع منابع و مدیریت خودکار آن‌ها، قابلیت ارتجاعي و مدل‌های هزینه‌ای پرداخت در ازای استفاده، بیشترین سود را برای توسعه‌دهنده‌ی کاربرد به بار می‌آورند. لازم به ذکر است که این خودکارسازی به‌ناچار بایستی در خود کاربرد یا با استفاده از کارکرد مدیریت فراهم‌کننده به‌وجود بیاید. بنابراین، ضرورت دارد که توسعه‌دهندگان کاربرد، رابط‌های مدیریتی ابرهای استفاده شده را به‌کار بگیرند تا ماشین‌ها را شروع و متوقف ساخته و با نظارت بر استفاده از منابع، کارکرد مهیا شده توسط فراهم‌کننده را دسترسی نمایند. به‌رغم تلاش گسترده‌ی بخش صنعت در جهت استانداردسازی رابط‌های مدیریت ابر، تکیه بر استفاده از کارکرد یک فراهم‌کننده‌ی به‌خصوص نیز به وابستگی سفت و سخت^۶ کاربرد توسعه‌دهنده به یک فراهم‌کننده‌ی خاص ابر منجر می‌شود. تکامل رابط‌ها و کارکرد ابری که صنعت جلودار آن است مقایسه‌ی محصولات عرضه شده توسط ابرها را نیز مشکل می‌سازد. مفاهیم مشترک ذیربط نیز در بیشتر موارد پنهان مانده یا در دسترس عموم قرار داده نمی‌شوند. لذا، توسعه‌ی کاربرد ابر در مدیریت

رایانش ابری^۱، شیوه‌های استفاده از منابع فناوری اطلاعات (IT^۲) را به‌طرز گسترده‌ای متحول ساخته است. در اینجا، منابع لازم می‌توانند بر اساس نیاز رزرو شده و در صورت عدم نیاز، رها شوند. مطابق تعریف مؤسسه‌ی ملی استانداردها و فناوری (NIST^۳) [۱]، «رایانش ابری، یک مدل کسب‌وکار است که روش‌ها و فناوری‌ها را برای تدارک منابع IT به‌شکل انعطاف‌پذیر از طریق یک شبکه با یک رابط سرویس‌دهی یکپارچه می‌سازد». در حال حاضر به‌نظر می‌رسد که استفاده از فناوری‌های ابر به یک استاندارد از مجموعه‌های نرم‌افزاری مبدل شده است: تصاویر ماشین و خدمات مهیا شده برای استفاده، توسط فراهم‌کننده‌ی ابر ارائه می‌شوند و یا بوسیله‌ی توسعه‌دهندگان کاربرد به‌منظور استفاده‌ی مجدد در چندین کاربرد و پیکربندی محیطی ایجاد می‌شوند. می‌توان مهم‌ترین خواص ابرها را به‌صورت زیر خلاصه کرد [۲]:

- **قابلیت ارتجاع یا کشسانی^۴:** تعداد منابع در دسترس کاربرد می‌تواند به‌صورت انعطاف‌پذیر تغییر یابد تا با نیاز فعلی جفت‌وجور شود.
- **پرداخت در ازای/استفاده:** هزینه‌ی منابع، مطابق طول زمان استفاده از آن‌ها محاسبه می‌شود.
- **استانداردسازی:** استفاده از مجازی‌سازی^۵ ساخت‌افزایی و خدمات سکوی تهیه شده توسط فراهم‌کننده، محیط‌های زمان اجرای کاربردها را در ابر، استاندارد می‌سازد.

⁴ Elasticity

⁵ Virtualization

⁶ Tight coupling

¹ Cloud Computing

² Information Technology (IT)

³ National Institute of Standards and Technology (NIST)



برداشتن الگوها و اعمال آن‌ها، مدل طراحی خود را به تدریج کامل کند [۴، ۸، ۹].

بر اساس اهمیت درک مفهوم الگوها، روابط میان الگوها و زبان‌های الگو در زمینه‌ی رایانش ابری به‌منظور استفاده و کاربرد آن‌ها در توسعه‌ی کاربردهای مبتنی بر ابر، این مقاله با گسترش نقشه‌ی راه ارائه شده در [۱۰]، به‌دنبال ارائه‌ی چارچوبی برای اعمال الگوها و زبان‌های الگو توسط طراح مدل مبتنی بر ابر می‌باشد. موضوعی که در اینجا اهمیت دارد، انتخاب الگوی مناسب، به‌لحاظ ارتباط میان الگوهای یک زبان مشخص و شیوه‌ی واریسی^{۱۰} الگوهای اعمال‌شده و ارتباط بین الگوهای مد نظر زبان است. این روال که «واریسی زبان الگو»^{۱۱} [۸] نامیده می‌شود با رویکردهای مختلف در قالب چارچوب پیشنهادی بحث و بررسی می‌شود.

سایر مطالب مقاله به این صورت سازمان‌دهی می‌شود. بخش ۲، به معرفی کارهای مرتبط در زمینه‌ی هدایت و راهنمایی طراح مدل‌های مبتنی بر الگو و زبان الگو می‌پردازد. بخش ۳، ضمن ارائه‌ی مفهوم الگو و زبان الگو، به ارائه‌ی خط‌مشی‌هایی برای شناسایی آن‌ها می‌پردازد. بخش ۴، یک دسته‌بندی از الگوها و روابط میان الگوهای رایانش ابری ارائه می‌دهد. بخش ۵، ضمن تشریح واریسی دستی اعمال یک زبان الگو، چارچوب توسعه و واریسی کاربردهای مبتنی بر الگوها و زبان الگو رایانش ابری را با تأکید بر رویکردهای ممکن پیاده‌سازی معرفی می‌کند. نهایتاً، بخش ۶، ضمن جمع‌بندی، به نتیجه‌گیری و مروری بر کارهای آتی می‌پردازد.

نیازمندی‌ها با مشکل مواجه شده و به یک فراهم‌کننده‌ی ابر خاصی محدود می‌شود. برای مثال، توسعه‌ی Azure ویندوز با توسعه‌ی App Engine گوگل یا سرویس‌های وب Amazon (AWS^۷) متفاوت است.

مهم‌تر این که، مسئله‌ی جابجایی فراهم‌کنندگان ابر در حالت کلی قابل حل نیست؛ چون، برای مثال، سرویس‌های عرضه شده‌ی سکوها و زیرساخت‌های زمان اجرا در کارکرد و پشتیبانی زبان‌های برنامه‌نویسی فرق دارند. با این وجود، استفاده از الگوها^۸ و زبان‌های الگو می‌تواند در حل نیازمندی‌های مکرر کاربردها به توسعه‌دهندگان کمک کنند. الگوها عمدتاً وابستگی خاصی به فناوری خاص مستندات و خط‌مشی‌های فراهم‌کننده، همین‌طور، کاربردهای موجود ابر ندارند [۳]. لذا، الگوها تجریدی^۹ از مفاهیم معماری، مستقل از فناوری فراهم می‌کنند. الگوها به‌صورت اسناد خوانای انسانی هستند که دانش تثبیت‌شده‌ی خیرگان را به مبتدیان در زمینه‌های مختلف، اعم از توسعه‌ی کاربردهای مبتنی بر ابر انتقال می‌دهند [۴-۶].

خیلی کم پیش می‌آید که الگوها به‌صورت تکی به‌کار گرفته شوند. اعمال الگوها به‌شکل مجموعه‌ای از الگوهای مرتبط، در قالب یک زبان الگو، به‌عنوان یکی از بهترین مشق‌های حل یک مسئله به‌صورت کلی در یک زمینه‌ی مشخص مطرح است [۷، ۸]. زبان الگو، شبکه‌ای از الگوهای مرتبط است که فرایندی را برای حل یک مسئله فراهم می‌کند. در حقیقت، این زبان است که اعمال الگوهای مناسب را در زمینه‌ی یکدیگر دیکته و اجبار می‌کند، طوری که توسعه‌دهنده می‌تواند با

¹⁰ Verification

¹¹ Pattern language verification (PLV)

⁷ Amazon Web Services (AWS)

⁸ Patterns

⁹ Abstraction



۲- کارهای مرتبط

به‌عبارت دیگر، این ابزار صرفاً اعمال الگوهای زبان PofEAA و روابط آن‌ها را واریسی می‌کند.

روحی و زمانی [۹، ۱۲]، در راستای تعمیم مفهوم واریس زبان الگو^{۱۷} که توسط زمانی و باتلر [۸] مطرح شده بود، چارچوبی به‌نام PLVGen [۱۳] توسعه داده‌اند که با پیش‌فرض نمایش راه‌حل الگوهای زبان با نمودارهای کلاس UML و قابلیت توصیف با زبان‌های توصیف الگو و زبان الگو، یعنی PSL^{۱۸} و PLSL^{۱۹} [۱۴]، واریس زبان الگوی ورودی را به‌صورت خودکار تولید می‌کند. واریس تولیدی که به زبان EVL^{۲۰} [۱۴] تولید می‌شود طراح مدل را به‌صورت محاوره‌ای هدایت می‌کند تا مدل طراحی خود را گام‌به‌گام با پیروی از ساختار الگوهای زبان و لحاظ روابط میان الگوهای آن، ضمن مشاهده‌ی عواقب الگوهای اعمال شده توسعه بدهد.

فیلینگ^{۲۱} و همکاران [۱۵] که عمدتاً روی الگوهای رایانش ابری کار کرده‌اند، برای هدایت طراح مدل از یک جدول تصمیم استفاده کرده‌اند. در خانه‌های این جدول، نوع رابطه‌ی الگوها که می‌توانند از نوع وابستگی قوی (یا همان/استفاده) یا تضاد باشند، مشخص می‌شود.

۳- شناسایی الگوها و زبان‌های الگو

در این بخش، برای فهم مفهوم الگو و زبان الگو، تعریف ساده‌ای از این مفاهیم و نمونه‌هایی ارائه می‌شود. سپس، یک ساختاری برای الگوهای رایانش ابری ارائه می‌شود.

هرچند که نمونه‌ی کارهای مرتبط در ادبیات تحقیق مربوط به راهنمایی و هدایت طراح در اعمال الگوها، به‌شکل زبان الگو محدود است. در اینجا، به معرفی مختصر برخی از کارهای مهمی می‌پردازیم که در این زمینه انجام شده‌اند.

زدان^{۱۲} [۱۱] برای حل مسائل پیچیده‌ای که انتخاب الگوی راه‌حل مناسب از میان الگوهای نامزد، به دلیل تعدد آن‌ها دشوار است، از یک روش سامان‌مند انتخاب الگو استفاده می‌کند. در اینجا، برای نمایش زبان الگو از یک گرامری شبیه BNF^{۱۳} (که برای توصیف بیشتر زبان‌های برنامه‌نویسی مرسوم استفاده می‌شود) حاشیه‌نویسی شده استفاده می‌کند. با ناکافی بودن گرامر و حاشیه‌های آن، برای رفع پیچیدگی انتخاب الگوی مناسب از تکنیک تحلیل فضای حالت طراحی و رویکرد QOC^{۱۴} استفاده می‌کند.

زمانی و باتلر^{۱۵} [۸]، به‌منظور هدایت طراح یک مدل UML بر اساس زیرمجموعه‌ای از مجموعه‌ی الگوهای Fowler [۷] در قالب زبانی به‌نام PofEAA^{۱۶}، ابزاری به نام ArgoPLV توسعه داده‌اند که خود از امکانات خاص ابزار ArgoUML بهره می‌برد. کار این ابزار مشابه یک کامپایلر نوعی یک زبان برنامه‌نویسی است که طی سه مرحله، ساختار، گرامر و معانی الگوهای اعمال شده را واریسی می‌کند. محدودیت اصلی ابزار ارائه شده به تک‌زبانی بودن آن بر می‌گردد.

¹⁷ Pattern Language Verifier (PLV)

¹⁸ Pattern Specification Language (PSL)

¹⁹ Pattern Language Specification Language (PLSL)

²⁰ Epsilon Validation Language (EVL)

²¹ Fehling

¹² Zdan

¹³ Backus-Naur Form (BNF)

¹⁴ Questions, Options, Criteria (QOC)

¹⁵ Butler

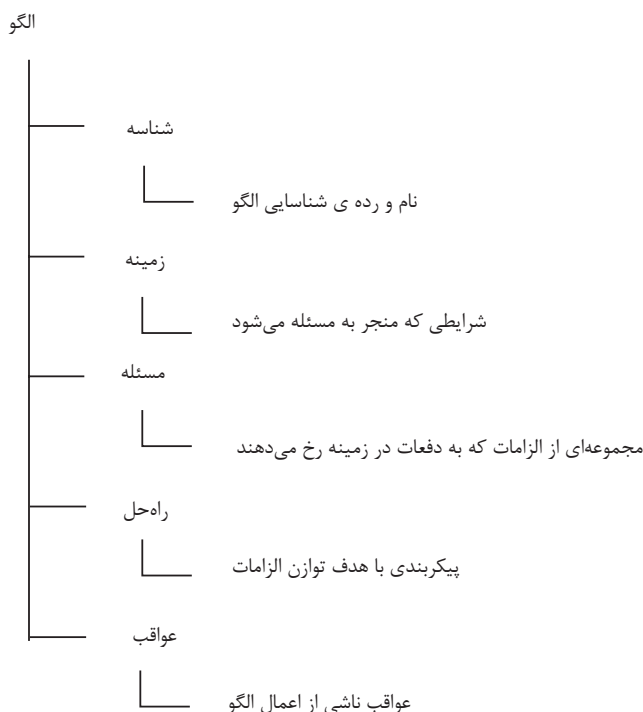
¹⁶ Patterns of Enterprise Application Architecture (PofEAA)



۳-۱- تعریف الگو

به زبان ساده، الگو راه‌حلی برای یک مسئله است که به دفعات در یک زمینه‌ی مشخصی رخ می‌دهد [۴، ۵، ۷، ۱۶]. الگوها در قلمروهای زیادی، اعم از معماری ساختمان و نرم‌افزار شیء‌گرا نیز استفاده دارند [۵، ۱۶]. لذا، به‌رغم داشتن شباهت‌هایی، ساختار و شکل الگو در زمینه‌های مختلف، متفاوت است. شکل رایجی از ساختار الگو در شکل ۱ نشان داده شده است. در اینجا، ساختار

الگو، شامل شناسه (برای شناسایی الگو که بایستی حتی‌الامکان با معنی انتخاب شود تا انتقال مفهوم آن به استفاده‌کنندگان مبتدی الگو و در میان توسعه‌دهندگان آسان‌تر شود)، زمینه^{۲۲} (شرایطی که منجر به یک مسئله می‌شود)، مسئله، راه‌حل (توصیف چگونگی رفع‌ورجوع مسئله عمدتاً با نمودارهایی همچون نمودارهای کلاس و توالی UML) و عواقب^{۲۳} (که خود را در دو شکل، منافع^{۲۴} یا دیون باقی‌مانده^{۲۵} نشان می‌دهد) الگوست.



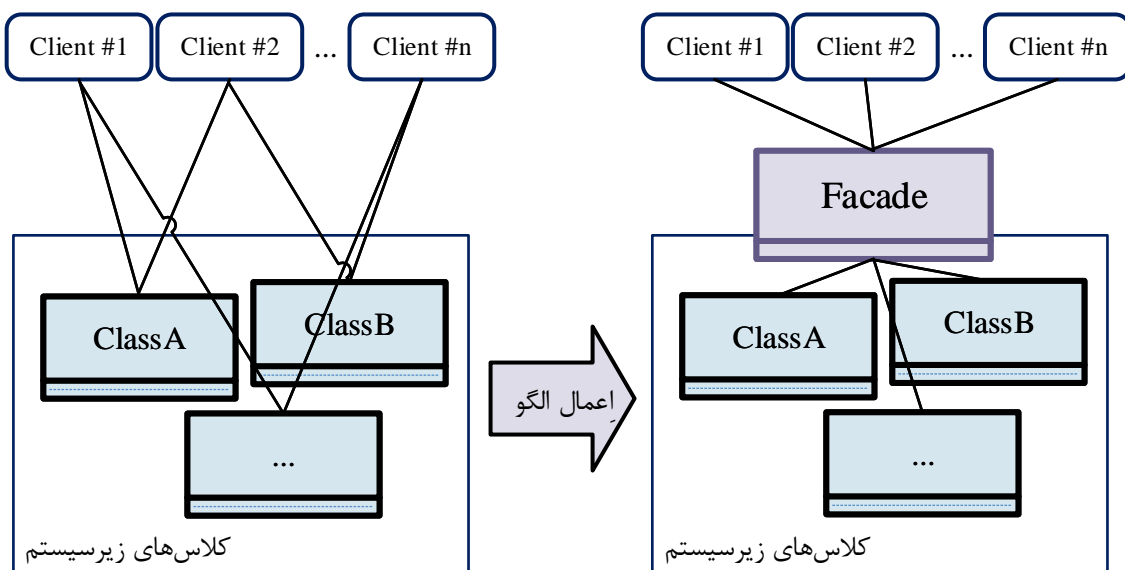
شکل ۱- شکل و ساختار الگو (برگرفته از [۱۴])

اعمال یک الگوی Facade می‌تواند به‌عنوان راه‌حلی مطرح شود. شکل ۲، استفاده از این الگو را نشان می‌دهد. در حقیقت، در اینجا، الگوی Facade به‌عنوان رابطی عمل می‌کند تا پیچیدگی‌های داخلی کلاس‌های کامپایلر را از دید کارفرماها و استفاده‌کنندگان آن پنهان کند.

به‌عنوان مثالی از الگو و کاربرد آن، اعمال الگوی Facade [۵] را در یک سیستم آشنای کامپایلر یک زبان برنامه‌نویسی در نظر بگیرید. در این سیستم، کلاس‌های مختلفی اعم از اسمبلر و لینکر وجود دارند. چون الزامی نیست که همه‌ی کاربران (کارفرماهای) یک کامپایلر، برای استفاده با این زیرسیستم‌ها آشنایی داشته باشند،

²⁴ Benefits
²⁵ Liabilities

²² Context
²³ Consequences



شکل ۲- اعمال الگوی Facade برای دسترسی آسان کارفرماها به یک سیستم کامپایلر (برگرفته از [۱۴])

چنانچه پیشتر عنوان شد، الگوها معمولاً در کاربردشان رفتار اجتماعی دارند [۴]. به عبارت دیگر، الگوها در متن و چارچوب یکدیگر اعمال می‌شوند. بنابراین، آنچه لازم است لحاظ رابطه‌ی میان الگوها در انتخاب و اعمال آنهاست که باعث کارآمدی و اثربخشی یک طراحی در مقایسه با اعمال تکی الگوها بدون در نظر گرفتن روابط آنها نیز می‌گردد. روابط میان الگوها بستگی به زمینه و الگوهای استفاده شده در آن دارد. برای مثال، زبان الگوی Broker که در ساخت سیستم‌های توزیع شده و همروند استفاده می‌شود روابط «استفاده» و «رقابت» را میان الگوها تعریف می‌کند [۱۷]. نمودار ریلی مجموعه‌ی الگوهای این زبان در شکل ۳ نشان داده شده است. در این زبان، الگوی Broker به عنوان الگوی آغازین زبان محسوب می‌شود. به عبارت دیگر، این الگو بدون هیچ الگوی پیش‌نیازی قابل اعمال است. برای مثالی از روابط میان الگوها، رابطه‌ی میان الگوهای Broker و Layers را در نظر بگیرید. الگوی Broker به منظور ایجاد ساختار پیمانه‌ای و سادگی طراحی از الگوی Layers استفاده می‌کند. همه‌ی روابط این زبان

الگوهای رایانش ابری، مسائلی را پوشش می‌دهند که موقع طراحی، ساخت، و مدیریت کاربردهای ابری پیش می‌آیند. اسناد مربوط به الگوها برای راهنمایی ترتیب اعمال الگوهای دیگر، ترکیب الگوهایی که استفاده‌ی رایجی دارند و مشخص کردن الگوهای مرتبط به‌همدیگر ارجاع می‌دهند. الگو در مقایسه با مستندات فنی یا خط‌مشی‌های توسعه که فناوری‌های به‌خصوصی را در نظر می‌گیرند، در حقیقت، اصل یک راه‌حلی را اقتباس می‌کند که برای یک مسئله‌ی مکرر کارگر است. الگوهای رایانش ابری از کاربردهای موجود، سرویس‌های فراهم‌کننده، و سایر منابع بدست می‌آیند که سبک‌های معماری مرسوم، مؤلفه‌های محیط‌های زمان اجرای ابری، و مؤلفه‌های کاربردهای ابر را بیان می‌کنند. قالب این اسناد الگو برای ارائه‌ی یک ظاهر و حس مشترک، همگن ساخته می‌شود [۱۵].

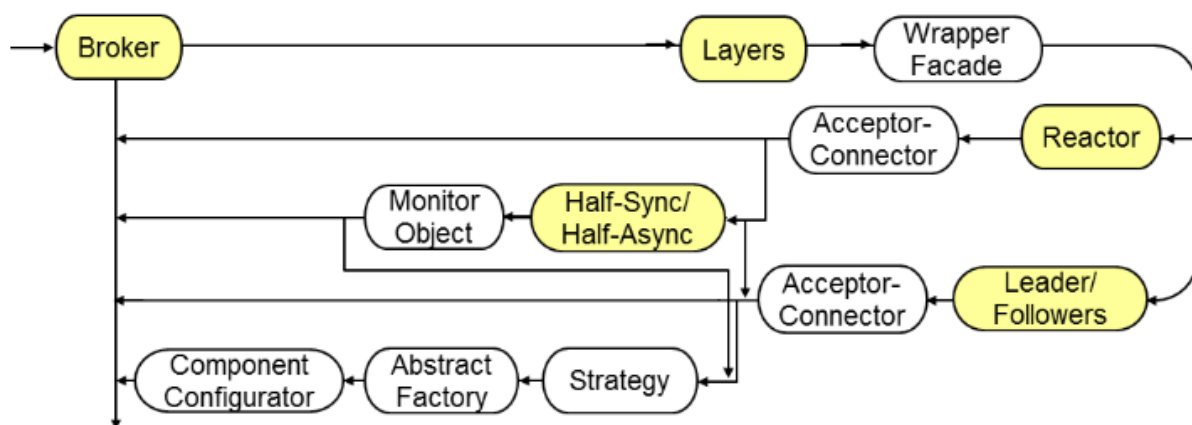
۳-۲- زبان الگو



حل یک مسئله‌ی خاص در یک زمینه‌ی مشخص فراهم می‌کند. در نمونه‌ی زبان الگوی نشان داده شده در شکل ۳، بر اساس نظر و خواست طراح، به لحاظ اهداف کیفی طراحی می‌توان مسیرهای مختلفی را با شروع از الگوی آغازین «Broker» انتخاب کرده و الگوهای آن را با در نظر گرفتن روابط بین آن‌ها اعمال کرد. یکی از معیارهای مهم در انتخاب این مسیرها عواقبی است که به شکل منافع و دیون در اعمال الگوها در زمینه‌ی یکدیگر عاید می‌شوند [۹، ۱۲].

به استثنای رابطه‌ی بین الگوهای Reactor و Leader/Followers که «رقابت» است از نوع «استفاده» هستند. رابطه‌ی «رقابت» به این معنی است که دو الگو برای استفاده در یک زمینه با همدیگر رقابت دارند و این نظر طراح است که بسته به اهداف کیفی که مد نظر دارد، با این فرض که الگوی Wrapper Facade به عنوان زمینه‌ی جاری است، یکی از آن‌ها، یا هر دو را انتخاب و اعمال نماید.

به بیان ساده، زبان الگو، شبکه‌ای از الگوهای مرتبط با یک الگوی آغازین است که فرایندی را برای



شکل ۳- زبان الگوی Broker (برگرفته از [۱۸])

البته، مشابه مفهوم الگویی که در رایانش ابری مطرح است، تعاریف دیگری از زبان الگو نیز در منابع برای حوزه‌های مختلف وجود دارد [۹، ۱۱].

برای شناسایی الگوها و اقتباس آن‌ها در قالب کاتالوگ الگو و زبان الگو، روش‌ها و منابع اطلاعاتی مختلفی وجود دارد که در اینجا رویکرد مرجع [۱۵] به صورت خلاصه ارائه می‌شود:

- همکاری با صنعت مرتبط با رایانش ابری که در این زمینه موفق بوده‌اند. البته، مفاهیم و اصولی که این کاربردها دارند انتزاعی ساخته می‌شوند

زبان الگوی یک دامنه‌ی خاص، مجموعه‌ای از الگوهای دامنه، روابط بین آن‌ها، و قوانین ترکیب آن‌هاست. لذا، زبان الگو به مسائل مشترک در یک دامنه می‌پردازد تا فرایند طراحی را هدایت کند. بر این اساس، الگوهای رایانش ابری بایستی در این حوزه، مشاوره‌ی کافی را برای هدایت کاربر، به منظور حصول هدف کلی طراحی ایجاد یک کاربرد ابری یا تجدید ساختار یک کاربرد موجود فراهم کنند. به علاوه، این الگوها بایستی مفاهیم را تا حدی انتزاعی سازند که تضمین شود اسامی الگوها برای ارتباط بهتر میان طراحان استفاده می‌شوند [۱۵].



تا فارغ از هرگونه محدودیت فناوری و محیطی باشند.

- بررسی ادبیات تحقیق و منابع موجود زبان‌های الگو و مستندات فراهم‌کنندگان ابر می‌تواند منبع قابل توجهی برای استخراج الگوها و زبان‌های الگو در هر زمینه‌ای اعم از رایانش ابری باشد. لازم به ذکر است که بایستی الگوهای موجود در زمینه‌های دیگر، همچون کاربردهای مستقل، کاربردهای گرید ۲۶، کاربردهای مبتنی بر پیغام و غیره نیز بررسی شوند تا در صورتی که در کاربردهای ابری کارگر بودند جمع‌آوری و به‌کار گرفته شوند. یک قالب یکسانی نیز باید برای نمایش الگوها ارائه شود تا الگوهایی که از منابع مختلف استخراج می‌شوند همگن و یک‌شکل شوند. جدول ۱، یک ساختار مشخصی را برای قالب الگوهای رایانش ابری نشان می‌دهد.
- تجربه‌ی خبرگانی که از کار با سیستم‌های توسعه‌ی کاربردهای ابری به ما منتقل می‌شود.

جدول ۱- یک ساختار و قالب الگو برای الگوهای رایانش ابری [۶]

جزء	توصیف کوتاه
نام الگو	این اسم برای شناسایی هر الگو استفاده می‌شود. بهتر است مختصر و گویا انتخاب شود تا به‌خاطر سپاری آن آسان‌تر شود.
هدف ۲۷	خلاصه‌ای از الگو، یعنی مسئله و راه‌حل در یک یا دو جمله.
آیکن	برای هر الگو، یک آیکنی هم‌اندازه با سایر الگوها طراحی

می‌شود که به‌صورت تصویری نشان‌گر الگو و کاربرد آن باشد. از این آیکن‌ها در نمودارسازی و ارتباط ساده‌تر الگوها استفاده می‌شود.

مسئله‌ای که توسط الگو حل می‌شود به‌صورت یک سؤال بیان می‌شود. خواننده می‌تواند با مقایسه‌ی این پرسش با مسئله‌ی طراحی که مد نظرش است به الگو پی ببرد.

شرایطی را بیان می‌کند که منجر به مسئله شده است. در حالی که سؤال پیش‌برنده، یک درک سریعی از مسئله‌ی حل شده توسط الگو را فراهم می‌کند، یک توصیف مفصلی از تمامی چالش‌ها در این بخش نیز ارائه می‌شود.

این که چگونه مسئله توسط الگو حل می‌شود را به‌صورت کوتاه بیان می‌کند.

در اینجا، راه‌حل، حداقل با یک طرح گرافیکی نشان داده می‌شود که کارکرد راه‌حل یا معماری حاصل را بعد از اعمال الگو به‌تصویر می‌کشد.

عواقب و آثار اعمال الگو را بیان می‌کند.

غالباً الگوها می‌توانند به شیوه‌های مختلفی اعمال شوند که به شرایط و چالش‌های متفاوتی بستگی دارد. این اختلاف‌ها چندان زیاد نیستند که در اسناد مجزایی بیان شوند. لذا، در اینجا، این تغییرات اندک

پرسش

پیش‌برنده ۲۸

زمینه ۲۹

چالش‌ها

راه‌حل

طرح

نتیجه

گونه‌ها

²⁸ Driving question

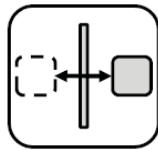
²⁹ Context

²⁶ Grid

²⁷ Intent



- **نام الگو:** الگوی دروازه‌ی مؤلفه‌ی ابر
- **هدف:** چگونه یک مؤلفه‌ی کاربرد در یک ابر یا مرکز داده در دسترس قرار می‌گیرد، موقعی که در یک ابر یا مرکز داده‌ی متفاوتی میزبانی می‌شود و ارتباط میان این محیط‌ها محدود است؟
- **آیکن:**



- **زمینه:** کاربردهای متعدد یا مؤلفه‌های آن‌ها مابین ابرها یا مراکز داده‌ای متفاوتی توزیع شده‌اند. ارتباط همگام یا ناهمگام بین این محیط‌ها محدود است، برای مثال، از طریق استفاده از فایروال. در اکثر کاربردهای این الگو، ارتباط به‌طرف داخل از یک محیط خارجی^{۳۳} به یک محیط داخلی^{۳۴} محدود می‌شود. در حالی که مؤلفه‌های داخلی مقیم در محیط داخلی ممکن است مؤلفه‌های خارجی را دسترسی نمایند، هیچ‌گونه دسترسی خارجی برای مؤلفه‌های داخلی مجاز نیست.
- **چالش‌ها:** کانال‌های ارتباطی در دسترس میان محیط‌های محاسباتی مورد استفاده توسط یک شرکت غالباً محدود می‌شوند. در حالی که ارتباط به‌طرف بیرون ممکن است، ارتباط به‌طرف داخل از یک محیط خارجی ممکن است کاملاً محدود شود. با این وجود، مؤلفه‌های کاربردی که در خارج مستقرند

زمینه و راه‌حل، یعنی گونه‌های الگو پوشش داده می‌شوند. الگوهایی که با این الگو در ارتباط هستند را مشخص می‌کند، مثلاً در زمینه‌ی خود از این الگو استفاده کرده یا آن را پالایش می‌کنند. این ارتباط‌ها هستند که امکان ترکیب و ساخت یک زبان الگو را میسر می‌سازند.

استفاده‌های شناخته‌شده
نمونه‌های عملی کاربرد این الگو را نشان می‌دهد.

حاشیه‌نویسی‌ها
۳۰
با هدف قابلیت گسترش ممکن است حاشیه‌هایی با مصنوعات اضافی مربوط به نمونه‌سازی الگوها در سکوها‌ی عینی، رفتار آن‌ها، کنترل وضعیت مؤلفه‌های مشمول آن‌ها و غیره لحاظ شوند. البته، این حاشیه‌ها برای همه‌ی الگوها به‌شیوه‌ای یکسان اعمال نمی‌شوند.

۴- دسته‌بندی الگوها

در اینجا، نخست یک الگو به‌عنوان مثال ارائه می‌شود. سپس، دسته‌بندی‌های مختلف الگوهای رایانش ابری در منابع مختلف مطرح می‌شوند.

۴-۱- الگوی دروازه‌ی مؤلفه‌ی ابر

الگوی دروازه‌ی مؤلفه‌ی ابر^{۳۱} در طی تحلیل Windows Azure App Fabric عرضه شده توسط مایکروسافت و گذرگاه سرویس کلان‌شرکت WSO2، محصولی از WSO2^{۳۲} کشف شد.

³³ Off-premise

³⁴ On-premise

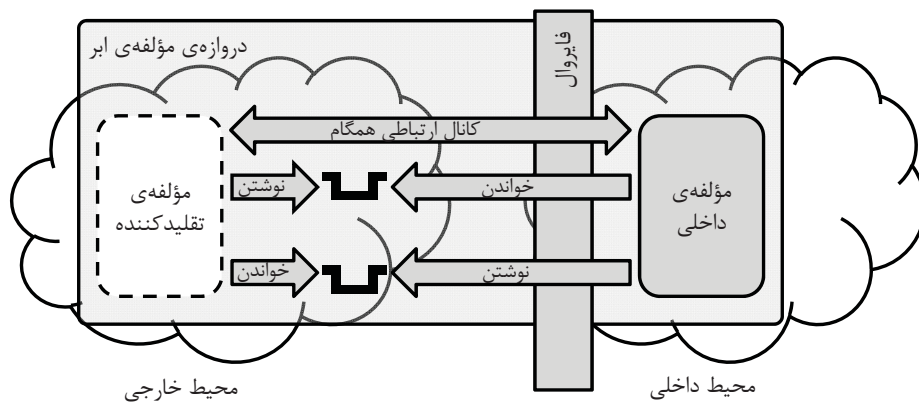
³⁰ Annotations

³¹ Cloud Component Gateway Pattern

³² <https://wso2.com/products/enterprise-service-bus/>



- **طرح:** طرح این الگو در شکل ۴ نشان داده می‌شود.
 - **نتیجه:** از آنجایی که از دسترسی مستقیم به مؤلفه‌ی داخلی اجتناب می‌شود، داده‌ها و کارکردهای آن می‌توانند از طریق کپی آن دسترسی شوند. کانال‌های ارتباطی ممکن است همگام یا ناهمگام باشند و همواره به‌واسطه‌ی محدودیت‌های دسترسی، توسط مؤلفه‌ی داخلی برپا شده و حفظ می‌شوند.
- ممکن است نیاز به دسترسی داده‌ها و کارکرد مؤلفه‌های کاربردهای داخلی داشته باشند که در داخل مستقرند، اما دسترسی مستقیم به این مؤلفه‌ها ناممکن است.
- **راه‌حل:** رابط مؤلفه‌ی داخلی در محیط خارجی تکثیر می‌شود. این تکثیر، رابط مؤلفه‌ی داخلی را تقلید کرده و دسترسی‌های مؤلفه‌ی داخلی را به پیش می‌فرستد.



شکل ۴- طرح الگوی دروازه‌ی مؤلفه‌ی ابر [۱۵]

- **الگوهای مرتبط:** الگوی دروازه‌ی مؤلفه‌ی ابر می‌تواند در ابرهای ترکیبی برای رفع و رجوی محدودیت‌های ارتباطی بین محیط‌های محاسباتی که یکپارچه هستند استفاده شود. این کار برای مجتمع‌سازی کاربردهای متعدد، در کنار یکپارچه‌سازی مؤلفه‌های کاربردی که یک کاربرد مرکب را می‌سازند کارگر است. ارتباط ناهمگام می‌تواند با اعمال الگوی پیغام‌رسانی مطمئن^{۳۵}، تحویل دقیقاً یک‌بار یا تحویل حداقل یک‌بار را تضمین کند. به‌علاوه، الگوی دروازه‌ی مؤلفه‌ی ابر، مبتنی بر مفاهیم الگوی proxy [۵] و الگوی facade [۵] می‌باشد که چالش‌های مشابهی را در کاربردهای مستقل مرتفع می‌سازند.
- **گونه‌ها:** این الگو در اصل، یکپارچه‌سازی بین محیط‌های متفاوت را مطرح می‌کند. برای مثال، می‌توان محیط‌های خارجی را یکپارچه کرد در صورتی که محدودیتی مشابه بین آن‌ها

³⁵ Reliable messaging pattern



دسترس‌پذیری سرویس‌ها در هر دو محیط استفاده شود.

- حاشیه‌نویسی‌ها: استفاده از پیاده‌سازی ویندوز Azure در آدرس <https://docs.microsoft.com/en-us/azure/service-bus-relay/service-bus-relay-tutorial> و مستندات دروازه‌های سرویس WSO2 در آدرس <http://wso2.org/project/esb/java/30.1/docs>. ارائه شده است.

۴-۲- فهرست الگوها

در اینجا، کلیاتی از الگوهای کشف‌شده‌ی رایانش ابری در قالب مطرح شده، شامل لیستی از نام الگو، آیکن، و سؤال پیش‌برنده ارائه می‌شود. این کاتالوگ الگوها شامل رده‌بندی‌های مدل‌های سرویس ابر، انواع ابر، عرضه‌های ابر، و معماری کاربردهای ابر می‌باشد.

• مدل‌های سرویس ابر

زیرساخت به‌عنوان سرویس:



چگونه یک زیرساخت IT می‌تواند به‌صورت پویا از طریق یک شبکه عرضه شود؟

سکو به‌عنوان سرویس:



چگونه سکوهایی IT می‌توانند به‌صورت پویا از طریق یک شبکه عرضه شوند؟

نرم‌افزار به‌عنوان سرویس:



چگونه نرم‌افزار می‌تواند به‌صورت پویا از طریق یک شبکه عرضه شود؟

ترکیب به‌عنوان سرویس:



وجود داشته باشد. صف‌هایی هم که برای تحقق ارتباط ناهمگام میان مؤلفه‌ی داخلی و مؤلفه‌ی رابط آن لازم هستند ممکن است توسط فراهم‌کننده‌ی متفاوتی میزبانی شود، به نام فراهم‌کننده‌ی یکپارچه‌سازی کاربرد کلان‌شرکتی^{۳۶} (EAI) به‌عنوان یک سرویس، یا در یک سگمنت شبکه‌ی تحت کنترل خاص در محیط داخلی، یک منطقه‌ی غیرنظامی^{۳۷} (DMZ).

- استفاده‌های شناخته‌شده: یک پیاده‌سازی از این الگو و سرویس‌هایی به‌عنوان مؤلفه‌های کاربردی یکپارچه‌سازی توسط مایکروسافت عرضه شده است که از کانال ارتباطی همگام استفاده می‌کند. این پیاده‌سازی بخشی از ویندوز Azure App Fabric می‌باشد. در صورتی که سرویس داخلی با استفاده از ابزار توسعه‌ی مایکروسافت توسعه یابد، مؤلفه‌ی تقلیدکننده می‌تواند خودکار تولید شود. در شرایطی که نیاز شود سرویس‌های داخلی به‌صورت خارجی در دسترس قرار بگیرند، یک سرویس داخلی اضافه‌تری باید توسعه داده شود که سرویس موجود را دستیابی می‌کند. WSO2، دسترسی ناهمگام مؤلفه‌های داخلی را به‌عنوان بخشی از گذرگاه سرویس کلان‌شرکتی WSO2 عرضه می‌کند. در صورتی که این گذرگاه سرویس کلان‌شرکتی، داخلی نصب شود، در یک محیط خارجی، به نام «دروازه‌های سرویس» ممکن است برای

³⁷ Demilitarized zone (DMZ)

³⁶ Enterprise Application Integration (EAI)



چگونه سرویس‌های محاسبه با هزینه‌های پایینی می‌تواند عرضه شود در صورتی که دسترس‌پذیری آن‌ها قید و بند نداشته باشد؟

گره محاسبه با دسترسی بالا:

چگونه سرویس‌های محاسبه می‌توانند عرضه شوند در صورتی که دسترس‌پذیری آن‌ها اهمیت حیاتی داشته باشد؟

سازگاری محض:

چگونه دسترس‌پذیری یک راه‌حل ذخیره‌سازی می‌تواند افزایش یابد در حالی که سازگاری هم در تمامی زمان‌ها تضمین شود؟

سازگاری احتمالی^{۳۹}:

چگونه دسترس‌پذیری و کارایی یک راه‌حل ذخیره‌سازی توزیع شده می‌تواند افزایش یابد در صورتی که نیازمندی سازگاری خفیف‌تر شود؟

ذخیره‌ی داده‌ای رابطه‌ای:

چگونه عناصر داده‌ای می‌توانند ذخیره شوند چنان که رابطه‌های میان صفات آن‌ها و عناصر دیگر بتوانند بیان شده و پرس‌وجوهای پیچیده بر اساس این صفات امکان‌پذیر باشند؟

ذخیره‌ی Blob:

چگونه عناصر داده‌ای بزرگ می‌توانند به صورت متمرکز ذخیره و سازمان‌دهی شده و از طریق یک شبکه در دسترس قرار بگیرند؟

ذخیره‌ی بلاکی:

چگونه ذخیره‌ی مرکزی شبیه به درایوهای سخت محلی می‌توانند دستیابی شوند؟

چگونه یک کاربرد ترکیبی می‌تواند به صورت پویا از طریق یک شبکه عرضه شود؟

• انواع ابر

ابر عمومی:

چگونه سرویس‌های الاستیک IT می‌توانند به صورت همروند برای شرکت‌های متفاوتی عرضه شوند؟

ابر خصوصی:

چگونه سرویس‌های الاستیک IT به صورت انحصاری برای استفاده‌ی داخلی یک شرکت عرضه می‌شود؟

ابر ترکیبی:

چگونه سرویس‌های الاستیک IT می‌توانند برای شرکت‌های متعددی عرضه شوند در حالی که برخی سرویس‌ها توسط یک شرکت به صورت انحصاری استفاده می‌شوند و حتی ممکن است توسط آن فراهم شوند؟

ابر مجموعه‌ای^{۳۸}:

چگونه سرویس‌های الاستیک IT می‌توانند به صورت همروند برای مجموعه‌ی ویژه‌ای از شرکت‌ها عرضه شوند؟

• عرضه‌های ابر

زیرساخت الاستیک:

چگونه منابع IT می‌توانند به صورت همروند و مطابق نیاز عرضه شوند؟

گره محاسبه با دسترسی پایین:

³⁹ Eventual Consistency

³⁸ Community Cloud



ذخیره‌ی NoSQL:



چگونه یک پایگاه داده می‌تواند یک ساختمان داده را در مقیاس گسترده و انعطاف پذیر پشتیبانی کند؟

میان افزار پیغام‌گرا:



چگونه کاربردها (یا مؤلفه‌های کاربردها) می‌توانند از طریق پیغام‌هایی از دور در ارتباط باشند در حالی که به لحاظ مکان و قالب پیغامشان، توزیع نرمی^{۴۰} داشته باشند؟

پیغام‌رسانی مطمئن:



چگونه پیغام‌ها می‌توانند مبادله شوند در حالی که تضمین شود پیغام‌ها در شرایط شکست‌های سیستمی یا ارتباطی گم نمی‌شوند؟

تحویل دقیقاً یک بار:



چگونه یک میان‌افزار پیغام‌گرا می‌تواند مطمئن شود که یک پیغامی که فرستاده می‌شود فقط یک بار به گیرنده تحویل می‌شود؟

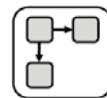
تحویل حداقل یک بار:



چگونه کارایی یک سیستم پیغام‌رسانی می‌تواند افزایش یابد در صورتی که پیغام‌های تکراری قابل قبول باشند؟

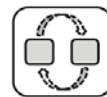
• معماری کاربردهای ابر

کاربرد ترکیبی:



چگونه کارکرد کاربرد، توزیع شده و از منابع مختلف ترکیب می‌شود؟

توزیع نرم:



چگونه می‌توان وابستگی‌های میان کاربردها و مؤلفه‌های آن‌ها را کاهش داد؟

مؤلفه‌ی بدون حالت:



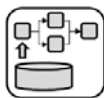
چگونه می‌توان از فقدان داده‌ها اجتناب کرد، در صورتی که مؤلفه‌ای از یک کاربرد از کار بیفتد یا از کاربرد برداشته شود؟

مؤلفه‌ی Idempotent:



چگونه مؤلفه‌ای که پیغام‌هایی را دریافت می‌کند به پیغام‌های تکراری رسیدگی می‌کند؟

کاهش نگاهت:



چگونه کارایی پرس‌وجوهای پیچیده بر روی مجموعه‌های داده‌ای بزرگ افزایش یابد، در صورتی که راه‌حل ذخیره‌سازی مورد استفاده، این قبیل پرس‌وجوها را به صورت بومی پشتیبانی نکند؟

مؤلفه‌ی الاستیک:



چگونه تعداد مؤلفه‌های کاربردی که گسترش داده می‌شوند بر اساس بهره‌مندی سیستم، خودکار تطبیق پیدا کنند؟

موازنه‌گر بار الاستیک^{۴۱}:



چگونه تعداد مؤلفه‌های کاربردی که گسترش داده می‌شوند بر اساس تعداد درخواست‌ها، خودکار تطبیق پیدا کنند؟

صف الاستیک:



چگونه تعداد مؤلفه‌های کاربردی که گسترش داده می‌شوند بر اساس تعداد درخواست‌های ناهمگام به شیوه‌ای بهینه، به صورت خودکار تطبیق پیدا کنند؟

سگ پاسبان^{۴۲}:



چگونه یک کاربرد با دسترس پذیری بالا با استفاده از گره‌های محاسبه‌ی نامطمئن محقق شود؟

⁴² Watchdog

⁴⁰ Loosely coupled

⁴¹ Elastic Load Balancer



نخست مسئله را در بین الگوهای کشف شده جستجو می‌کنیم تا بهترین الگویی را انتخاب کنیم که راه حل آن کامل تر باشد. همان گونه که در مقدمه هم عنوان شد، عمده‌ی مسئله‌ها با اعمال الگوهای ترکیبی در قالب زبان الگو طراحی و مدل سازی می‌شوند. برای الگوهای مطرح شده در بخش قبل، دو رابطه به صورت زیر قابل شناسایی است [۱۵]:

- ۱- **رابطه‌ی استفاده:** به این معنی که یک الگو از الگوی دیگر در زمینه‌ی خود استفاده می‌کند.
 - ۲- **رابطه‌ی تضاد:** به این معنی که یک الگو نمی‌تواند با الگوی دیگر در یک زمینه استفاده شود.
- لازم به ذکر است که رابطه‌های بین الگوها بیشتر بستگی به زمینه و الگوهای زبان الگو دارد [۱۲]. برای نمونه رابطه‌های «استفاده» در شکل ۵، روابط میان الگوی *زیرساخت به‌عنوان سرویس* و سایر الگوها، یعنی الگوهای *ابر عمومی*، *ابر خصوصی*، *ابر ترکیبی*، *ابر مجموعه‌ای*، *زیرساخت الاستیک*، *گره محاسبه‌ی با دسترسی پایین*، *گره محاسبه‌ی با دسترسی بالا* و *مؤلفه‌ی الاستیک* را نشان می‌دهد.

به‌عنوان مثالی از رابطه‌ی تضاد میان الگوهای رایانش ابری می‌توان به رابطه‌ی *ابر خصوصی* و *ابر عمومی* اشاره کرد که در شکل ۶ با یک پیکان دوسر در میان دو الگو نشان داده شده است. به عبارت دیگر، نمی‌توان هم‌زمان، یک ابر را به صورت عمومی و خصوصی در نظر گرفت. در اینجا، به منظور حفظ اختصار از پرداختن به روابط *استفاده* و *تضاد* دیگری که میان الگوهای رایانش ابری وجود دارد اجتناب می‌کنیم (برای جزئیات بیشتر [۶، ۱۵] را ببینید).

انتقال به روزرسانی:

چگونه یک کاربرد مؤلفه‌ای به صورت یک جا به روزرسانی می‌شود، موقعی که نسخه‌های جدید مؤلفه‌های کاربرد یا میان افزار استفاده شده، سیستم عامل و غیره در دسترس قرار بگیرند؟
مؤلفه‌ی تک نمونه:



چگونه یک مؤلفه‌ی کاربرد می‌تواند میان کرایه‌کنندگان متعددی به اشتراک گذاشته شود در صورتی که پیکربندی تکی ضرورتی نداشته باشد؟



مؤلفه‌ی تک نمونه‌ی قابل پیکربندی:
چگونه یک مؤلفه‌ی کاربرد می‌تواند میان کرایه‌کنندگان متعددی به اشتراک گذاشته شود در صورتی که پیکربندی تکی لازم باشد؟



مؤلفه‌ی با تعدد نمونه:

چگونه یک مؤلفه‌ی کاربرد برای کرایه‌کنندگان متعددی که آن را پیکربندی می‌کنند فراهم شود، در صورتی که اشتراک امکان پذیر نباشد؟



۵- چارچوب توسعه و واریسی کاربردهای

مبتنی بر الگوها و زبان الگوی رایانش

ابری

در این بخش، نخست روش دستی شناسایی، انتخاب و اعمال الگوها و زبان‌های رایانش ابری را مطرح می‌کنیم. سپس، چارچوب پیشنهادی توسعه و واریسی خودکار خود را برای پرهیز از پیچیدگی‌های ذاتی این روال ارائه می‌کنیم.

۵-۱- اعمال دستی الگوها و زبان الگو

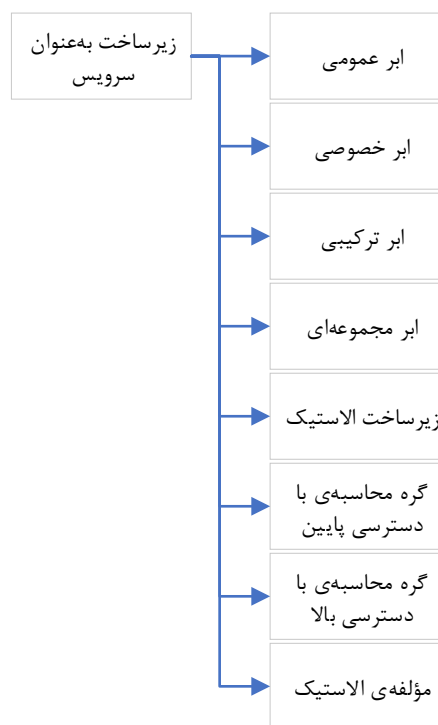


در ادامه، بر اساس روابطی که الگوهای اعمال شده دارند سایر الگوها به ترتیب در زمینه‌ی آنها اعمال می‌شوند. موضوع دیگری که اهمیت دارد، توجه به عواقبی است که از اعمال الگوها ناشی می‌شود [۱۲]. بسته به نظر طراح ممکن است مصالحه‌هایی نیز در حین اعمال الگوها با لحاظ روابط دیکته شده توسط زبان الگو صورت گیرد. برای مثال ممکن است طراح بخواهد برای امنیت بیشتر کاربرد، الگوهای را اعمال کند که موجب تنزل کارایی و سادگی مدل سیستم می‌شوند. برای جزئیات بیشتر در رابطه با عواقب مثبت و منفی الگوها، به عبارتی منافع و دیون الگوهای اعمال شده به [۱۲] مراجعه کنید. روشن است که اعمال دستی الگوهای یک زبان، مستلزم بررسی دستی صحت الگوهای تکی اعمال شده در کنار روابط میان الگوهای استفاده شده است.

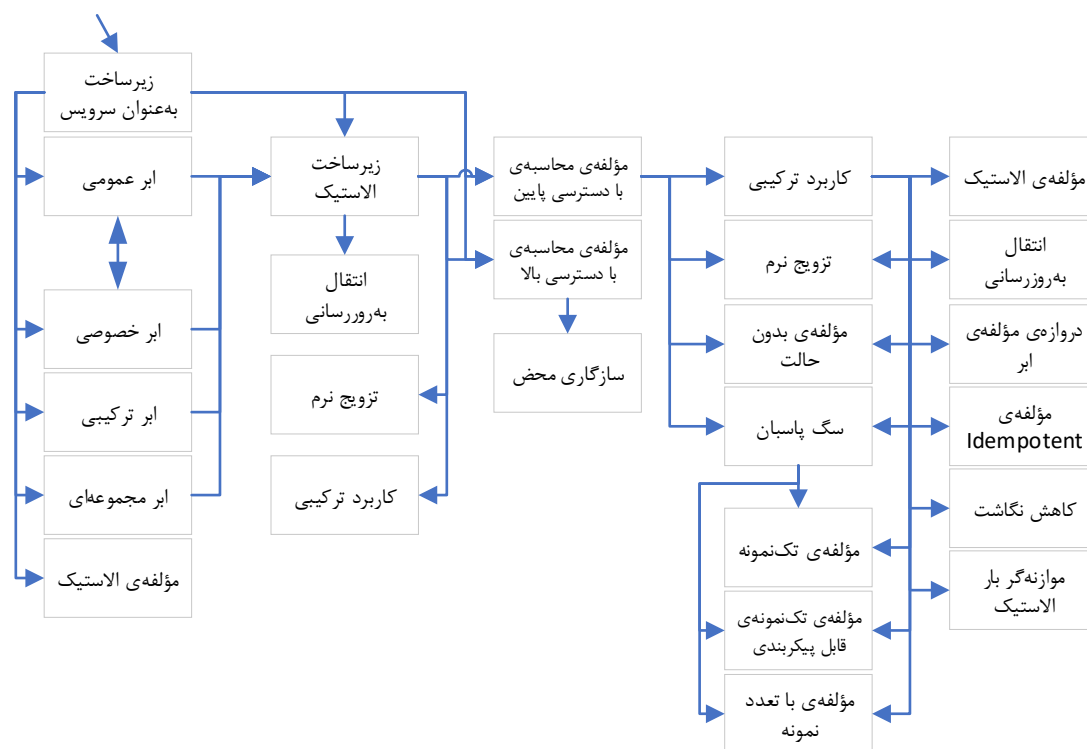
این فرایند می‌تواند با زبانی که تنها تعداد معدودی الگو دارد کارگر باشد. با این وجود، با رشد تعداد الگوها و متعاقباً پیچیدگی روابط میان الگوها، واریسی دستی زبان‌های الگوی اعمال شده، دشوار، مستعد خطا و عملاً ناممکن خواهد بود [۸، ۱۲]. لذا، توسعه‌ی یک چارچوب انتخاب و اعمال درست و واریسی خودکار الگوها و روابط میان آنها برای یک زبان الگو ضروری به نظر می‌رسد.

شکل ۶، به دلیل محدودیت فضا، تنها زیرمجموعه‌ای از الگوهای رایانش ابری را در قالب یک زبان الگو با الگوی آغازین «زیرساخت به‌عنوان سرویس» نشان می‌دهد (برای جزئیات بیشتر مربوط به الگوهای دیگر و روابط میان آنها به [۱۵] مراجعه کنید).

برای اعمال الگوها با پیروی از زبان الگو، لازم است که نخست آن دسته از الگوها اعمال شوند که هیچ پیش‌نیازی ندارند که در زبان الگو، اصطلاحاً الگوهای آغازین نامیده می‌شوند [۴].



شکل ۵- رابطه‌ی «استفاده‌ی» میان الگوی زیرساخت به‌عنوان سرویس و الگوهای دیگر رایانش ابری [۱۵]



شکل ۶- بخشی از زبان الگوی رایانش ابری و روابط میان الگوهای آن با الگوی آغازین «زیرساخت به‌عنوان سرویس» [۱۵]

۵-۲- چارچوب توسعه و واریسی

در اینجا، برای پرهیز از هرگونه ابهام و سردرگمی خواننده در رابطه با مفهوم واریسی که در این مقاله مد نظر است، تعریف ساده‌ای از این مفهوم ارائه می‌شود. واریسی (که در اینجا می‌توان با اعتبارسنجی هم مترادف دانست)، در حقیقت بررسی درستی الگو یا زبان الگوی اعمال‌شده توسط یک طراح روی مدل طراحی است. یکی از عوامل بازدارنده در توسعه‌ی ابزار خودکارسازی اعمال الگوها و زبان‌های الگو، نبود یک زبان مشترک و توصیف واحدی از الگوها و زبان‌های الگوست [۸، ۹]. لذا، در گام اول لازم است یک مدل یا فرامدل^{۴۳} واحدی برای الگوها و زبان‌های الگو تعریف شود [۹]. بر این

اساس، گام بعدی، واریسی و اعتبارسنجی این الگوها و زبان‌های الگو روی مدل‌های طراحی است که با استفاده از این زبان‌ها توسعه داده می‌شوند [۱۲]. تا جایی که بررسی ادبیات موضوع نشان می‌دهد دو رویکرد کلی برای واریسی الگو و زبان الگو وجود دارد که عبارتند از:

- رویکرد گرامری و کامپایلری [۸، ۱۹]: در اینجا، نخست الگوها و زبان الگو با یک گرامری مشابه BNF توصیف می‌شود. در ادامه، واریسی زبان الگو، همانند یک کامپایلر نوعی زبان‌های برنامه‌نویسی همچون C یا جاوا در نظر گرفته می‌شود. لذا، درستی الگوهای تکی اعمال‌شده، روابط میان آن‌ها و معانی ذیربط با گام‌های متناظر کامپایلر،

⁴³ Metamodel

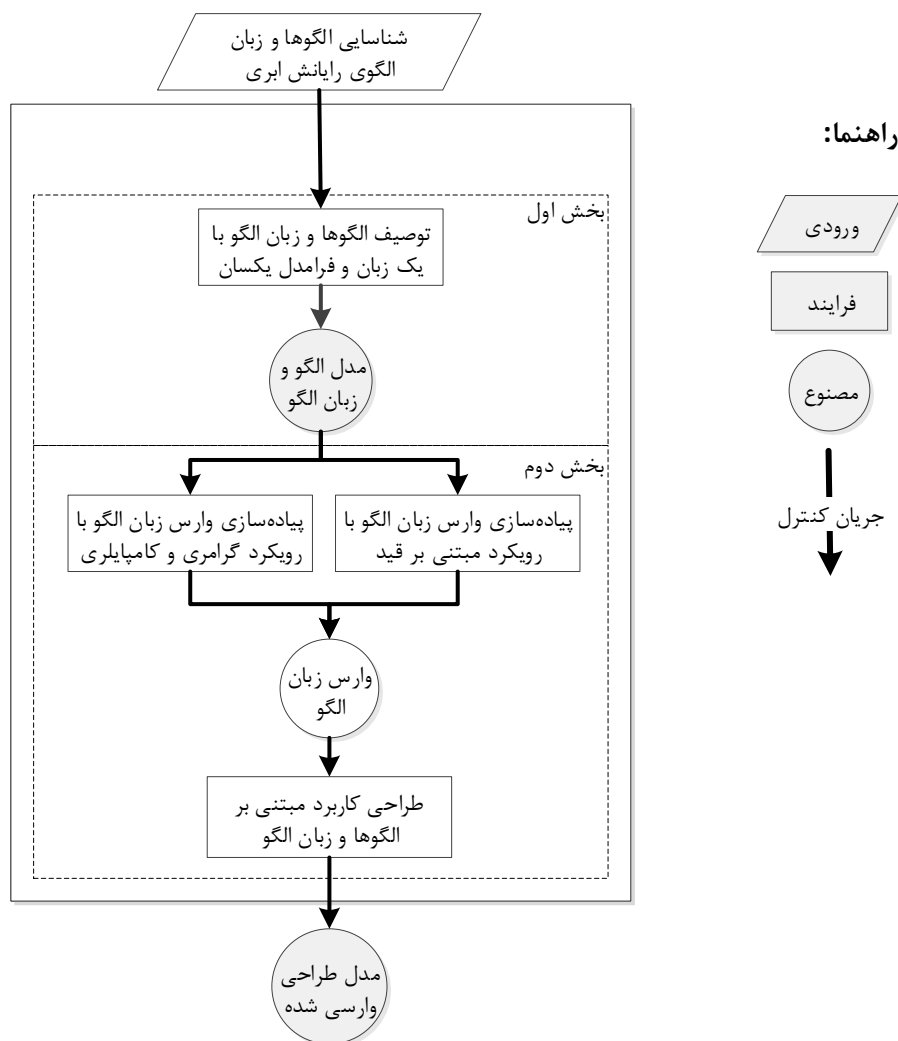


روابط میان آن‌ها و در مجموع، زبان الگوی مد نظر است.

شکل ۷، چارچوب پیشنهادی توسعه و واریسی کاربردهای رایانش ابری مبتنی بر الگوها و زبان‌های الگو را نشان می‌دهد. این چارچوب، در حقیقت، دارای دو بخش اساسی است. بخش اول، ارائه‌ی یک زبان یا فرامدل برای توصیف الگوها و زبان‌های الگوست.

یعنی بررسی لغوی (درستی الگوهای تکی)، نحوی (درستی روابط میان الگوها) و معنایی (درستی الگوهای اعمال شده در زمینه‌ی مورد نظر) واریسی و اعتبارسنجی می‌شود.

- رویکرد مبتنی بر قید [۹، ۱۲، ۲۰]: الگوها، روابط میان الگوها، و در کل، زبان الگوی مطلوب، به‌صورت مجموعه‌ای از قیود تعریف می‌شود که درستی و ارضای آن‌ها در یک مدل طراحی به‌معنی اعمال درست الگوها،



شکل ۷- چارچوب توسعه و واریسی کاربردهای رایانش ابری مبتنی بر الگو و زبان‌های الگو



یک دسته‌بندی از الگوهای رایانش ابری مطرح شد. در ادامه، روابط بین الگوها و چگونگی اعمال آن‌ها در قالب زبان الگو تشریح شدند.

این مقاله سعی داشت یک نقشه‌ی راهی برای طراحان مدل‌های کاربردی مبتنی بر ابر ارائه کند. در همین راستا، یک چارچوبی برای شناسایی، توصیف و واری الگوها و زبان‌های رایانش ابری ارائه شد. این چارچوب، توسعه و پیاده‌سازی ابزار طراحی و واری کاربردهای مبتنی بر زبان الگو را تسهیل می‌بخشد. لازم به ذکر است که در تداوم این پژوهش، ارائه‌ی یک مدل رسمی برای الگوها و زبان‌های الگوی رایانش ابری یا تطبیق مدل‌های رسمی موجود (لازمه‌ی بخش اول در توسعه‌ی چارچوب پیشنهادی) [۹] می‌تواند تولید ابزارهای لازم را برای اعتبارسنجی اعمال این الگوها و زبان‌ها بر روی مدل‌های کاربردی مبتنی بر ابر تسهیل نماید.

بخش دوم، بر اساس انتخاب یکی از رویکردهای مطرح‌شده، واری یا کامپایلر زبان الگوی مد نظر توسعه و پیاده‌سازی می‌شود. سپس، واری بدست آمده برای واری و اعتبارسنجی زبان الگو در طراحی یک مدل مبتنی بر زبان الگوی مذکور توسط طراح استفاده می‌شود تا مدل واری شود. این چارچوب می‌تواند کارایی و کارآمدی طراحی مدل را با فراهم‌ساختن تعامل طراح با واری زبان الگو از طریق پیشنهاد و توصیه‌ی الگوهای مطلوب ارتقا ببخشد [۸، ۱۲، ۱۳].

۶- نتیجه‌گیری

با توجه به پیچیدگی توسعه‌ی کاربردهای مبتنی بر ابر، استفاده از الگوها و زبان‌های الگوی این زمینه برای غلبه بر این پیچیدگی‌ها توصیه می‌شود. به همین منظور، کارهای موجود در رابطه با هدایت و راهنمایی طراح مبتنی بر الگو و زبان الگو بحث و بررسی شدند. سپس،

۷- مراجع

- [8] B. Zamani and G. Butler, "Pattern language verification in model driven design," *Information Sciences*, vol. 237, pp. 343-355, 2013.
- [9] A. Rouhi and B. Zamani, "Towards a formal model of patterns and pattern languages," *Inf. Softw. Technol.*, vol. 79, no. C, pp. 1-16, 2016.
- [۱۰] ع. روحی، «توسعه‌ی کاربردهای رایانش ابری مبتنی بر الگو و زبان الگو: یک نقشه‌ی راه»، چهارمین کنفرانس ملی محاسبات توزیعی و پردازش داده‌های بزرگ، دانشگاه شهید مدنی آذربایجان، تبریز، ۱۳۹۷، صفحات ۳۶۹-۳۶۲.
- [11] U. Zdun, "Systematic pattern selection using pattern language grammars and design space analysis," *Software: Practice and Experience*, vol. 37, no. 9, pp. 983-1016, 2007.
- [12] A. Rouhi and B. Zamani, "A model-based framework for automatic generation of a pattern language verifier," *Softw. Pract. Exper.*, vol. 47, no. 12, pp. 1945-1980, 2017.
- [13] A. Rouhi and B. Zamani, "PLVGen Setup Guide v1.0," UI-SE-MDSERG-2016-08, Model-Driven Software Engineering Research Group, University of Isfahan, Isfahan, Iran, 2016. <http://www.bahmanzamani.com/TR/UI-SE-MDSERG-2016-08.pdf>.
- [14] A. Rouhi, "Presenting a process for generating a pattern language verifier," Ph.D. dissertation, University of Isfahan, 2017.
- [15] C. Fehling, F. Leymann, R. Retter, D. Schumm, and W. Schupeck, "An architectural pattern language of cloud-

- [1] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of internet services and applications*, vol. 1, no. 1, pp. 7-18, 2010.
- [2] C. Fehling, F. Leymann, R. Retter, W. Schupeck, and P. Arbitter, *Cloud computing patterns: fundamentals to design, build, and manage cloud applications*. Springer, 2014.
- [3] F. Amato and F. Moscato, "Exploiting cloud and workflow patterns for the analysis of composite cloud services," *Future Generation Computer Systems*, vol. 6, pp. 255-265, 2017.
- [4] F. Buschmann, K. Henney, and D. C. Schmidt, "Pattern-oriented software architecture: On patterns and pattern languages, vol. 5," *Chichester, West Sussex, England: John Wiley & Sons*, 2007.
- [5] E. Gamma, *Design patterns: elements of reusable object-oriented software*. Pearson Education India, 1995.
- [6] C. Fehling, "Cloud computing patterns: identification, design, and application," Ph.D. Dissertation, University of Stuttgart 2015.
- [7] M. Fowler, *Patterns of enterprise application architecture*. Addison-Wesley Longman Publishing Co., Inc., 2002.



- based applications," presented at the Proceedings of the 18th Conference on Pattern Languages of Programs, Portland, Oregon, USA, 2011 .
- [16] C. Alexander *A pattern language: towns, buildings, construction*. Oxford university press, 1977.
- [17] D. C. Schmidt and C. Cleeland, "Applying patterns to develop extensible ORB middleware," *IEEE Communications Magazine*, vol. 37, no. 4, pp. 54-63, 1999.
- [18] B. Frank ,H. Kevlin, and C. Douglas, "Pattern Oriented Software Architecture "A pattern Language for Distributed Computing", " *Pattern-Oriented Software Architecture, A Pattern Language for Distributed Computing*, 2007.
- [19] H. C. Lovatt, A. M. Sloane, and D. R. Verity, "A pattern enforcing compiler (PEC) for Java: using the compiler," in *Proceedings of the 2nd Asia-Pacific conference on Conceptual modelling-Volume 43*, 2005, pp. 69-78: Australian Computer Society, Inc.
- [20] D. Kolovos, L. Rose, A. García-Domínguez, and R. Paige, *The Epsilon book*. 2018.

علیرضا روحی:



ایشان مدرک کارشناسی را از دانشگاه خوارزمی در سال ۱۳۷۹، ضمن انتخاب به‌عنوان دانش‌آموخته‌ی ممتاز و رتبه‌ی اولی دریافت کرد. سپس، کارشناسی ارشد

را از دانشگاه صنعتی شریف در سال ۱۳۸۳ و دکتری را در سال ۱۳۹۶، ضمن انتخاب به‌عنوان دانش‌جوی برگزیده‌ی پژوهشی دانشکده‌ی مهندسی کامپیوتر دانشگاه اصفهان، اخذ نمود. هر سه مقطع تحصیلی نامبرده، رشته‌ی مهندسی کامپیوتر (نرم‌افزار) است. در حال حاضر، ایشان به‌عنوان عضو هیئت علمی دانشگاه شهید مدنی آذربایجان مشغول به‌کار هستند. نشانی رایانامه‌ی ایشان عبارت است از:

rouhi@azaruniv.ac.ir