



رصد حافظه ماشین مجازی به عنوان یک شیء اطلاعاتی با بهره‌گیری از یک طرح جدید شماره‌گذاری نگارش‌ها

مجید تجملیان*^۱، محمد قاسم‌زاده^۱

دانشکده مهندسی کامپیوتر، دانشگاه یزد، یزد، ایران^۱

چکیده

دهه‌های متمادی است که طرح‌های شماره‌گذاری برای ردیابی نگارش‌ها و ویرایش‌های مختلف پرونده‌ها، بسته‌های نرم‌افزاری، مستندات و سایر انواع اشیاء اطلاعاتی استفاده می‌شوند. عدم وجود یک استاندارد همه‌منظوره، تطابق‌پذیر، جامع و کارآمد مشکلات مختلفی را در کاربردهای مختلف فناوری اطلاعات ایجاد می‌کند. جهت کمک به رفع این مشکلات، در این مقاله یک طرح شماره‌گذاری چهارگانه و تطابق‌پذیر برای شماره‌گذاری نگارش‌ها ارائه می‌شود. در این طرح، شناسه نگارش با استفاده از چهار عدد صحیح ساخته می‌شود. این چهار عدد را به ترتیب از چپ به راست «شماره ترتیب نشر»، «شماره نسل»، «شماره لیست قابلیت‌ها» و «شماره لیست تصحیحات» نامیده‌ایم. مقدارهای خاص برای شماره‌های چهارگانه مذکور و معانی آنها را تعریف کرده‌ایم. علاوه بر آن معیارهای تغییر شماره‌های چهارگانه نیز به تفکیک نوع کاربردهای مختلف مشخص شده‌اند. این یک طرح «تطابق‌پذیر» است، زیرا این قابلیت را دارد که بدون تغییر ساختار، برای ردیابی نگارش‌ها و ویرایش‌های مختلف پرونده‌ها، بسته‌های نرم‌افزاری، مستندات خروجی پروژه‌ها، طرح‌ها، قوانین، آئین‌نامه‌ها، شیوه‌نامه‌ها، نقشه‌ها، گرافیک‌ها، مستندات اداری و حقوقی و سایر موارد مشابه در محیط‌های مختلف به‌خوبی بکار رود. همچنین شگرد ابداعی جدیدی به نام «پیش‌صفری توسعه یافته» ارائه شده که برای دستیابی به «مرتب‌سازی لغوی» در اسامی مشتمل بر شناسه نگارش کاربرد دارد. از این طرح می‌توان برای رصد تغییرات انواع اشیاء اطلاعاتی مانند حافظه ماشین‌های مجازی در فرآیند مهاجرت زنده استفاده کرد. نتایج آزمایشی و تحلیلی حکایت از سزاواری و مؤثر بودن طرح پیشنهادی در برآورده سازی انتظارات مورد نظر دارند. گرچه ما از طرح پیشنهادی برای پژوهش در حوزه مهاجرت ماشین‌های مجازی استفاده کرده‌ایم، اما این طرح می‌تواند به عنوان یک طرح جامع و کارا در کلیه محیط‌های اداری، قانون‌گذاری، حقوقی، دانشگاهی، مهندسی، تولیدی، صنعتی، تولید مستندات، عملیاتی، توسعه نرم‌افزار و سایر موارد مشابه تبدیل به یک استاندارد همگانی شده و با موفقیت به کار گرفته شود. همچنین استاندارد شدن طرح پیشنهادی و استفاده فراگیر از آن می‌تواند کمک بزرگی به یکنواخت شدن درک همگان از شماره‌گذاری نگارش‌ها و ویرایش‌های اشیاء اطلاعاتی باشد.

کلمات کلیدی: حافظه ماشین‌های مجازی، شیء اطلاعاتی، طرح چهارگانه تطابق‌پذیر برای شماره‌گذاری نگارش‌ها، مهاجرت زنده ماشین‌های مجازی، پیش‌صفری توسعه یافته



تاریخچه مقاله:

تاریخ ارسال: ۹۷/۱۲/۱۰

تاریخ اصلاحات: ۹۸/۳/۱۵

تاریخ پذیرش: ۹۸/۵/۲۰

تاریخ انتشار: ۹۸/۵/۱۵

Keywords:

VM Memory
Informational object
QAVNS
VM live migration
E-zero Leading

Observation of VM Memory as an Informational Object using a New Version Numbering Scheme

Majid Tajamolian^{*1}, Mohammad Ghasemzadeh¹

¹Yazd University, Yazd, Iran

Abstract

For decades, numbering schemes have been used to track different versions and revisions of files, software packages, documents, and other types of informational objects. The absence of an all-purpose, adaptive, comprehensive and efficient standard makes various problems in different applications of information technology. To help address these issues, this article presents a quadruple and adaptive scheme for version numbering. In this scheme, the version ID is created using four integers. We call these four numbers from left to right "Release Sequence Number", "Generation Number", "Features List Number" and "Corrections List Number." We have defined specific values for the quadrilateral numbers and their meanings. In addition, the criteria for changing the quadruple numbers are specified in different application types. This is an "adaptive" scheme, as it truly has the ability to track, various versions of files, software packages, project output documents, plans, rules, bylaws, style sheets, maps, graphics, administrative and legal documents, and other similar things in different environments without need of restructuring. Also, a new inventive method called "Extended Zero Leading" is introduced which is used to achieve "lexicographical ordering" in names containing the version ID. The scheme can be used to observe changes in the many types of informational objects such as virtual machine memory in the live migration process. Experimental and analytical results indicate the appropriateness and effectiveness of the proposed scheme in meeting the expectations. Although we have used the proposed scheme for research in virtual machine live migration, it can be used as a comprehensive and effective scheme in all administrative, regulatory, legal, academic, engineering, manufacturing, industrial, document production, operational, and software development environments. It can become a universal standard and be applied successfully. The standardization of the proposed scheme and its widespread usage can also be a great help to harmonize everyone's understanding of the version and revision numbering of informational objects.

روش ارجاع به مقاله : م. تجملیان، م. قاسم‌زاده، رصد حافظه ماشین مجازی به عنوان یک شیء اطلاعاتی با بهره‌گیری از یک طرح جدید شماره‌گذاری نگارش‌ها، دوفصلنامه محاسبات و سامانه‌های توزیع شده، سال دوم، شماره اول، شماره پیاپی

۳، سال ۱۳۹۸، ص ۶۵-۸۳



۱ - مقدمه

بخشی از نتایج مقدماتی این پژوهش قبلاً منتشر شده‌اند [۱]. در این مقاله ضمن بیان مجدد نکات کلیدی، نسخه توسعه یافته نتایج پژوهش ارائه می‌گردد. یکی از مباحث مهم در مهاجرت زنده ماشین‌های مجازی نحوه تغییرات حافظه ماشین مجازی در مراحل مهاجرت و تصمیم‌گیری پیرامون نحوه مهاجرت بر مبنای آن است. در این راستا وجود طرحی که بتواند تغییرات حافظه ماشین مجازی را به عنوان نگارش‌های یک شیء اطلاعاتی به صورت مؤثر نمایش داده و رهگیری نماید، مفید است.

در دنیای فناوری اطلاعات برای ردیابی نگارش‌ها و ویرایش‌های^۱ مختلف پرونده‌ها، بسته‌های نرم‌افزاری و مستندات، از «طرح‌های شماره‌گذاری»^۲ مختلفی استفاده می‌شود. ساختار بسیاری از این طرح‌ها تابع سلیقه مدیران پروژه‌های مربوطه است.

به دلیل استاندارد نبودن طرح‌های شماره‌گذاری، ابهامات زیادی در ذهن افراد (بخصوص مخاطبین بیرون از پروژه‌ها) بروز می‌کند. همچنین مفاهیم موجود در بسیاری از این طرح‌ها برای نوع خاصی از خروجی پروژه‌ها مناسب و برای انواع دیگر نامناسب است.

از طرف دیگر مفاهیم نگارش، ویرایش و شماره‌گذاری تنها در فضای تولید نرم‌افزار کاربرد ندارد، بلکه تقریباً در هر محیطی که با «اشیای اطلاعاتی»^۳ سروکار داشته باشیم، این مفاهیم نیز مطرح و مورد استفاده هستند. این اشیای اطلاعاتی طیف وسیعی را شامل می‌شوند که به عنوان نمونه می‌توان به بسته‌های

نرم‌افزاری، مستندات خروجی پروژه‌ها، طرح‌ها، قوانین، آئین‌نامه‌ها، شیوه‌نامه‌ها، نقشه‌ها، گرافیک‌ها و مستندات اداری اشاره نمود.

در این مقاله یک طرح شماره‌گذاری جدید ارائه می‌شود که ضمن برخورداری از نقاط قوت طرح‌های موجود، نقاط ضعف آن‌ها را نیز پوشش می‌دهد و ضمناً تطابق‌پذیر^۴ (دارای قابلیت کاربرد چندمنظوره) نیز است. این طرح در حوزه پژوهشی مربوط به فعالیت‌های نویسندگان این مقاله (مهاجرت زنده ماشین‌های مجازی) کاربرد مؤثری می‌تواند داشته باشد.

ادامه این مقاله بدین شرح سازمان‌دهی شده است: در بخش بعدی انواع طرح‌های موجود برای شماره‌گذاری نگارش‌ها معرفی می‌گردند. در بخش سوم ساختار و نحوه عمل‌کرد طرح پیشنهادی به طور مشروح ارائه شده است. در بخش چهارم مرتب‌سازی لغوی در شناسه نگارش‌ها مورد بررسی قرار گرفته است و برای رفع مشکل آن شگرد جدیدی به نام «پیش‌صفری توسعه یافته» پیشنهاد شده است. در بخش پنجم ساختار کامل شناسه نگارش در طرح پیشنهادی تدوین شده است. در بخش ششم عملکرد طرح پیشنهادی با تطبیق آن بر کاربردهای مختلف مورد توجه قرار گرفته است. در بخش هفتم نتایج حاصله از اجرای طرح پیشنهادی در دو محیط کاری واقعی گزارش و ارزیابی شده است. در انتها (بخش هشتم) نتیجه‌گیری و پیشنهادها آمده‌اند.

3 Informational Object
4 Adaptive

1 Version and Revision
2 Numbering Scheme



۲ - طرح‌های موجود

در حال حاضر طرح‌های متعدد و مختلفی توسط تولیدکنندگان نرم‌افزار و سایر محتواهای دیجیتال مورد استفاده قرار می‌گیرند که می‌توان آنها را به چهار دسته کلی تقسیم کرد:

- ا) شماره شناسه‌های مبتنی بر ترتیب
- ب) شناسه‌های مبتنی بر تاریخ نشر
- ج) کدهای حرفی-عددی
- د) سایر طرح‌های متفرقه

الف) طرح‌های شماره شناسه مبتنی بر ترتیب:

این طرح‌ها رایج‌ترین نوع در عرصه صنعت نرم‌افزار هستند. در این طرح‌ها هر شماره نگارش مبتنی بر یک یا چند سطح عددی است که به ترتیب افزایش می‌یابند. در بیشتر اوقات هرچه از سمت چپ در شماره نگارش به سمت راست حرکت کنیم ارقام همچون یک عدد اعشار معمولی کاهش می‌یابد. البته به طور قطعی در اینجا همچون یک عدد اعشار فقط با دو سطح که با نقطه اعشار از هم جدا شده‌اند مواجه نیستیم؛ بلکه ممکن است چندین سطح وجود داشته باشند و جداکننده نیز به جای نقطه، نویسه‌های دیگری همچون dash (-) باشد. اگر تغییرات نسبت به نگارش قبل زیاد باشد چپ‌ترین رقم‌ها تغییر می‌کنند؛ اگر تغییرات متوسط باشند رقم‌های میانی، اگر تغییرات کم باشند رقم‌های بعدی و اگر خیلی کم باشند راست‌ترین رقم‌ها تغییر می‌کنند. میزان تغییر ارقام بستگی به نظر منتشرکننده دارد و قطعی نیست که فقط به میزان یک واحد باشد. به عنوان مثال هنگامی که پس از نگارش

۵,۱ نرم‌افزار Internet Explorer نگارش ۵,۵ ارائه گردید، بیانگر آن بود که تغییرات نسبت به نگارش قبلی نسبتاً بالا بوده است. نکته دیگری که در این طرح‌ها بایستی مورد توجه قرار گیرد آن است که سطوح مختلف الزاماً تک‌رقمی نیستند. به عنوان نمونه، در نرم‌افزار Drupal با شماره نگارش ۷,۶۱ مواجه هستیم [۲]. رویکرد دیگر در این دسته آن است که از دو شماره ترتیب اصلی و فرعی استفاده شده و به همراه آن رشته حرفی-عددی همچون a و b^۵ و rc^۶ را برای نوع انتشار به کار ببرند. یک «دنباله انتشار نرم‌افزار» که از این رویکرد استفاده می‌کند، می‌تواند به این شکل نمایش داده شود:

0.7 → 0.8 → 0.9 → 1.0a1 → 1.0b1 →
1.0b2 → 1.0rc1 → 1.0

البته بدیهی است که هرکدام از مراحل alpha, beta و rc در انتشار می‌توانند تا جایی که نیاز باشد با شماره‌های ترتیب بعدی ادامه یابند و در نهایت به نوع انتشار بعدی تبدیل شوند. ضمناً این احتمال هم وجود دارد که در این روش بعضی از افراد برای نمایش حجم بالای تغییرات از شماره‌هایی با گام‌های بزرگ‌تر استفاده کنند؛ به عنوان مثال از نگارش ۱,۳ به نگارش ۱,۷ جهش کنند.

رویکردهای دیگری همچون طرح‌های ذیل نیز در این دسته وجود دارند (موارد مندرج در [] به معنی اختیاری بودن آنهاست):

major.minor[.build[.revision]]
major.minor[.maintenance[.build]]



مبتنی بر ساختاری سه بخشی به شکل `version.revision.change` یا آن‌گونه که در نگارش ۲,۰,۰ نوشتار خود آورده، `major.minor.patch` است [۵]. گرچه در طرح وی بسیاری از مشکلات دیگر طرح‌های مبتنی بر ترتیب حل شده است، لیکن طرح پیشنهادی وی هنوز از ضعف‌هایی رنج می‌برد که ذیلاً به پاره‌ای از آن‌ها اشاره می‌گردد:

- نبود توصیف کاربرد طرح در محیط‌های غیر از تولید نرم‌افزار
- شفاف نبودن حد و مرز تغییر عددهای `minor` و `patch`
- عدم تباین بین حالتی که در یک نشر، تغییراتی از دو جنس `minor` و `patch` به طور همزمان اتفاق افتاده است، با حالتی که فقط تغییراتی از جنس `minor` رخ داده است.
- فاقد توانایی جهت اطلاع‌رسانی کافی به مخاطبین در مورد میزان بلوغ محتوایی بسته نرم‌افزاری مربوطه
- عدم امکان «مرتب‌سازی لغوی»^۷ نام نرم‌افزارهای حاوی شناسه نگارش آنها

به نظر می‌رسد طرح «نگارش‌دهی معنایی» فقط مناسب برای کاربرد در تولید نرم‌افزارهایی است که «رابط برنامه‌نویسی نرم‌افزار» (API)^۸ دارند و آن را به صورت دقیق تعریف کرده‌اند. البته لازم به ذکر است که طرح مذکور در رابطه با «مدیریت وابستگی‌ها»^۹ و

به عنوان نمونه‌ای دیگر از این‌گونه طرح‌ها می‌توان به پروژه عظیم و معروف GNU اشاره کرد که از رویکرد فوق‌الذکر استفاده می‌کند. البته توصیه این پروژه به توسعه‌دهندگان آن است که حتی‌الامکان فقط از دو شماره اول (یعنی `major` و `minor`) استفاده کنند [۳]. همچنین پروژه معروف هسته سیستم عامل Linux از طرح سه بخشی مشابهی استفاده می‌کند که به ترتیب از چپ به راست `major.minor.patch` نامیده می‌شوند. به دلیل حجم زیاد تقریباً همیشگی پیاده‌سازی و تغییرات در حال انجام بر روی این پروژه، کار به طور موازی در دو شاخه مستقل از هم به نام‌های «نشر پایدار» و «نشر ناپایدار» انجام می‌گیرد. علامت نشرهای پایدار زوج بودن شماره دوم (`minor`) در شناسه نگارش مربوطه، و علامت نشرهای ناپایدار فرد بودن شماره دوم (`minor`) در شناسه نگارش مربوطه است [۴].

در این رویکردها نیز همچون بقیه موارد فوق‌الذکر، تعریف اینکه چه تغییری «اصلی» محسوب شده و چه تغییری «فرعی»، کاملاً به نظر منتشرکننده بستگی دارد. اینکه چه چیزی یک «ساخت» (`build`) را تعریف می‌کند و اینکه یک «ویراست» (`revision`) چه تفاوتی با یک تغییر فرعی (`minor`) دارد هم باز به نظر منتشرکننده بستگی دارند. لذا مشکل عدم وجود یک درک استاندارد و مشترک بین منتشرکننده و مخاطب در این طرح‌ها مشهود است.

آخرین تلاش در این دسته طرح‌ها را آقای «پرستون ورنر» تحت عنوان «نگارش‌دهی معنایی» یا `Semantic Versioning` ارائه داده است. طرح وی

9 Dependency Management

7 Lexicographical Ordering

8 API: Application Programming Interface



نگارش‌ها استفاده می‌شود، غالباً دلایل بازاریابی و تجاری دلیل اصلی هستند و در پشت صحنه (داخل گروه‌های کاری شرکت‌ها) از شناسه‌های نگارش واقعی‌تری استفاده می‌شود. به عنوان مثال در داخل شرکت مایکروسافت MS-Windows 95 به نام‌های MS-DOS 7.0 و MS-Windows 4.00، و MS-Windows 2000 Server به نام MS-Windows NT 5.0 شناخته می‌شوند.

ج) شناسه‌های مبتنی بر کدهای حرفی-عددی

در این دسته از طرح‌ها، از کدهای حرفی-عددی برای ساخت شناسه نگارش استفاده می‌شود. به عنوان مثال می‌توان به تعدادی از محصولات سامانه‌عامل معروف شرکت مایکروسافت تحت عناوین MS-Windows XP، MS-Windows Me، MS-Windows Vista و محصول Flash MX از شرکت Macromedia اشاره نمود.

نمونه‌ای دیگر از این دسته طرح‌ها را می‌توان در توزیع معروف و محبوب سامانه‌عامل گنو/لینوکس به نام Debian مشاهده کرد. این توزیع گرچه از شماره‌های ترتیب اصلی و فرعی برای نگارش‌های پایدار خود استفاده می‌کند، لیکن در کنار آنها از نام‌های رمز متناسب با شخصیت‌های پویانمایی معروف «داستان اسباب‌بازی» (Toy Story) نیز بهره می‌برد. به عنوان مثال در زمان تدوین این مقاله آخرین نگارش پایدار توزیع دبیان ۱۰,۰ است که با عنوان Buster شناخته می‌شود. نگارش‌های آزمایشی این توزیع بر مبنای تاریخ انتشار شماره نگارش دریافت می‌کنند، و نگارش ناپایدار

جلوگیری از افتادن به دام «جهنم وابستگی»^{۱۰} تأثیر مثبتی بر پروژه‌های نرم‌افزاری دارد و به همین دلیل هم علیرغم مدت کمی که از ارائه آن گذشته است، مورد توجه بسیاری از فعالان این حوزه قرار گرفته است.

ب) طرح‌های شناسه مبتنی بر تاریخ انتشار

در این طرح‌ها متناسب با تاریخ انتشار، شناسه‌ها درج می‌شوند. به عنوان مثال پروژه معروف WINE اوایل از ترکیب شماره سال، ماه و روز (مثل Wine 20040505) به عنوان شناسه نگارش استفاده می‌کرد [۶]. توزیع معروف سامانه‌عامل گنو لینوکس به نام Ubuntu نیز از طرح شناسه نگارش مشابهی استفاده می‌کند. مثلاً Ubuntu 18.04 در آوریل ۲۰۱۸ منتشر شده است [۷]. نمونه دیگر بازی رایانه‌ای Street Fighter EX است که هنگام شروع شناسه نگارش خود را به صورت یک تاریخ بعلاوه یک کد منطقه جغرافیایی نمایش می‌دهد (مانند ASIA 961219).

در پاره‌ای از موارد از تاریخ‌های کد شده به عنوان شناسه نگارش استفاده می‌شود. به عنوان مثال شرکت مایکروسافت در مورد مجموعه نرم‌افزاری MS-Office از شناسه‌های چهار رقمی استفاده می‌کند که دو رقم سمت راست شماره روز در ماه تولید و دو رقم سمت چپ تعداد ماهی است که از ژانویه سالی که محصول شروع به تولید شده، گذشته است. بنابراین ۲۱۰۵ به معنی پنجمین روز از بیست و یکمین ماه از ژانویه سالی که محصول شروع به تولید شده، است.

در بیشتر مواقع هنگام استفاده از تاریخ در شناسه نگارش، از طرح مندرج در استاندارد [8] ISO 8601 تبعیت می‌شود [۹]. هنگامی که از تاریخ در شماره



و میزان تغییرات هر نگارش به نسبت نگارش قبلی آگاه نمی‌سازند.

۳ - طرح پیشنهادی

جهت غلبه بر این مشکلات، طرحی چهارگانه برای شماره‌گذاری شناسه نگارش‌ها با تطابق‌پذیری (قابلیت استفاده چندمنظوره) ارائه می‌شود. «تطابق‌پذیری» به معنی آن است که این طرح قابلیت آن را دارد که بدون تغییر ساختارش برای ردیابی نگارش‌ها و ویرایش‌های مختلف پرونده‌ها، بسته‌های نرم‌افزاری، مستندات خروجی پروژه‌ها، طرح‌ها، قوانین، آئین‌نامه‌ها، شیوه‌نامه‌ها، نقشه‌ها، گرافیک‌ها، مستندات اداری و حقوقی و سایر موارد مشابه در محیط‌های مختلف به خوبی بکار رود. بنابراین از این به بعد در این مقاله در جهت تأکید بر قابلیت استفاده چندمنظوره، برای همه موارد فوق‌الذکر از واژه «شیء» استفاده خواهیم کرد. ما طرح پیشنهادی‌مان را QAVNS نامیده‌ایم که سرواژه عبارت **Quadruple Adaptive Version Numbering Scheme** به معنی «طرح چهارگانه تطابق‌پذیر برای شماره‌گذاری نگارش‌ها» است.

در این طرح پیشنهادی، در حالت عادی شماره‌گذاری با استفاده از چهار فیلد به شکل عدد صحیح انجام می‌گیرد که با ممیز (نویسه ممیز «٫») در فارسی و نویسه نقطه «.» در زبان‌های اروپایی) از یکدیگر جدا می‌شوند و به ترتیب از چپ به راست «شماره ترتیب نشر» یا RSN^{۱۱}، «شماره نسل»^{۱۲}، «شماره لیست قابلیت‌ها»^{۱۳} و «شماره لیست تصحیحات»^{۱۴} نامیده می‌شوند. البته دو فیلد اختیاری

آن همواره Sid (همنام پسر شرور همسایه در پویانمایی فوق‌الذکر) نامیده می‌شود [۱۰].

در اغلب این طرح‌ها، قاعده مشخص و یکنواختی استفاده نمی‌شود و جنبه‌های تبلیغی و بازرگانی محصول به منطق علمی، یکنواخت و استاندارد غلبه دارند.

(د) سایر طرح‌های متفرقه

البته بعضاً با طرح‌هایی غیرعادی و متفرقه نیز مواجه می‌شویم که بیشتر جنبه‌های فانتری را روایت می‌کنند. به عنوان مثال واژه‌پرداز معروف TeX از روشی منحصربفرد استفاده می‌کند. پس از انتشار نگارش ۳ که نرم‌افزار مذکور تا حد زیادی به بلوغ و پایداری رسید، روزآمدسازی‌های بعدی با اضافه کردن یک رقم به انتهای شماره نگارش قبلی با هدف نزدیک شدن مجانبی به مقدار عدد پی (II) نمایش داده می‌شود. در حقیقت شماره نگارش واقعی نرم‌افزار TeX همان تعداد ارقام عدد شماره نگارش آن است! در زمان تدوین این مقاله عدد شماره نگارش این نرم‌افزار ۳,۱۴۱۵۹۲۶۵ است.

به طور مشابهی شماره نگارش‌های نرم‌افزار مفسر زبان **Metafont** که همانند TeX توسط پروفیسور دونالد کنوت منتشر شده است، به صورت مجانبی به مقدار عدد e نزدیک می‌شود. در حال حاضر عدد شماره نگارش این نرم‌افزار ۲,۷۱۸۲۸۱۸ است. بدیهی است که این گونه طرح‌ها نیز با وجود جذابیت و بدیع بودن از نظر مفهومی، بیشتر جنبه فانتری داشته و مخاطب را از نوع



اینکه میزان و نوع تغییرات ایجاد شده در اشیاء اطلاعاتی متوالی چگونه تشخیص داده شده و چه دسته‌بندی بر روی آنها انجام شود، تا حد زیادی بستگی به حوزه مربوط به استفاده از شیء اطلاعاتی دارد. به عنوان نمونه آقای بامل و همکارانش راهکاری را برای تشخیص خودکار و تعیین نگارش اجزای نرم‌افزاری که طبق استاندارد OSGi توسعه داده شده باشند پیشنهاد داده‌اند [۱۲]. طبیعتاً در این مقاله به دلیل رویکرد کلی‌نگر و تطابق‌پذیری که ما داریم، وارد این مقوله نمی‌شویم و به بیان معیارهای کلی برای تشخیص میزان و نوع تغییرات ایجاد شده در اشیاء اطلاعاتی و تأثیر آن‌ها بر روی شناسه نگارش شیء بسنده می‌کنیم.

چنانچه هیچکدام از تغییرات اعمال شده در شیء از جنس افزودن و/یا تغییر قابلیت یا محتوی نباشند بلکه همگی از جنس تصحیح و رفع اشکال باشند، به جای عدد شماره لیست قابلیت‌ها، فقط عدد لیست تصحیحات یک واحد افزایش خواهد یافت. مثلاً اگر پس از نگارش ۱۱,۰,۷,۰ تعدادی تغییرات همگی از جنس تصحیح و رفع اشکال در شیء انجام شوند، نگارش جدید ۱۲,۰,۷,۱ خواهد بود و نه ۱۲,۰,۸,۰ (که نشانگر تغییر تنها در قابلیت‌ها و محتوی است). به همین منوال اگر کلیه تغییرات نگارش بعدی هم فقط از جنس تصحیح و رفع اشکال باشند، مجدداً شماره لیست تصحیحات افزایش یافته و نگارش بعدی ۱۳,۰,۷,۲ خواهد بود.

اگر تغییرات اعمال شده در شیء به نسبت نگارش قبلی تلفیقی از «افزودن و/یا تغییر قابلیت یا محتوی» و «تصحیح و رفع اشکال» باشند، در این صورت شماره لیست قابلیت‌ها یک واحد افزایش یافته و شماره لیست

دیگر به نام‌های PRI و BMD نیز در این طرح وجود دارند که کاربرد اصلی آن‌ها در اشیای اطلاعاتی مورد استفاده در مهندسی نرم‌افزار است و در بخش پنجم مقاله مورد اشاره و توضیح قرار گرفته‌اند.

کار بر روی تولید یک شیء با تخصیص شناسه نگارش ۰,۰,۰,۰ آغاز می‌شود. عدد صفر در جایگاه شماره ترتیب نشر نشانگر آن است که این نطفه نخستین ایجاد شیء است. عدد صفر در جایگاه شماره نسل اشاره به آن دارد که شیء مذکور هنوز در مراحل اولیه ایجاد است و قابلیت ارائه (حتی به صورت آزمایشی) را ندارد. عدد صفر در جایگاه شماره لیست قابلیت‌ها نشانگر آن است که شیء حاوی حداقل محتوی یا قابلیتی که محل اعراب باشد (هرچند ناقص) نیست. عدد صفر در جایگاه شماره لیست تصحیحات نشانگر آن است که هنوز بر روی این نگارش شیء تصحیحاتی انجام نگرفته است. با افزودن محتوی یا قابلیت به آن (بدون انجام تصحیحات)، نگارش‌های بعدی به ترتیب ۱,۰,۱,۰ و ۲,۰,۲,۰ و ۳,۰,۳,۰ و ۴,۰,۴,۰ بوده و به همین ترتیب جلو خواهند رفت. هیچکدام از شماره‌های چهارگانه فوق‌الذکر الزامی به تک رقمی بودن ندارند؛ لذا مثلاً پس از نگارش ۱۵,۰,۹,۰ می‌توان به نگارش ۱۶,۰,۱۰,۰ رسید.

تغییرات در هر نگارش نسبت به سایر نگارش‌ها را «دلتای جهت‌دار»^{۱۵} یا به طور خلاصه «دلتا» می‌نامند که به معنی «سلسله عملیات تغییر پایه‌ای که وقتی به یکی از نگارش‌ها اعمال شود، نگارش دیگر به دست می‌آید» است. نوع دیگری از دلتا به نام «دلتای متقارن»^{۱۶} نیز تعریف می‌شود که در اینجا مورد نظر ما نیست [۱۱].



قابلیت یا محتوی است و هم تعدادی تصحیح و رفع اشکال را داراست.

تغییر شماره نسل از صفر به یک، زمانی اتفاق می‌افتد که اولین نسل قابل ارائه ایجاد شده باشد. اینکه منظور از «قابل ارائه بودن» چیست بسته به تعاریف سازمانی دارد. البته به طور کلی می‌توان معیارهایی همچون موارد ذیل را به طور مشترک در همه محیط‌های کاری پیشنهاد نمود:

- فاقد ایراد و خطای آشکار
- دارای ظاهر منظم و منطقی
- فاقد توضیحات و محتواهای مربوط به مراحل تولید/پایاده‌سازی

پس از آن متناظر با توضیحات فوق، چنانچه تغییرات اعمال شده در شیء زیاد و بنیادی باشند، منجر به تغییر در شماره نسل می‌شوند. مثلاً از شماره نگارش ۲۳۵,۱,۱۴,۷۲ به شماره نگارش ۲۳۶,۲,۰,۰ می‌رسیم. توجه به این نکته ضروری است که اگر شماره نسل افزایش یابد، شماره‌های لیست قابلیت‌ها و لیست تصحیحات بایستی به عدد صفر باز نشانی شوند.

تغییراتی که منجر به تغییر در شماره نسل می‌شوند را می‌توان بدین شرح پیشنهاد نمود:

- تغییراتی که ساختار کلی شیء را تحت تأثیر قرار می‌دهند
- تغییراتی که ظاهر شیء را دگرگون می‌سازند
- تغییراتی که زیرساخت‌ها یا قالب‌های مورد استفاده در شیء را کاملاً متفاوت می‌سازند

تصحیحات به مقدار «یک» بازنشانی خواهد شد. بنابراین برای چنین حالتی در مثال فوق پس از نگارش ۲۷,۰,۷,۱۵ به نگارش ۲۸,۰,۸,۱ خواهیم رسید. توجه کنید که در همین مثال اگر تغییرات نگارش جدید نسبت به نگارش قبلی فقط از جنس افزودن و/یا تغییر قابلیت یا محتوی به شیء باشد و هیچگونه تغییری از جنس تصحیح و رفع اشکال در آن نباشد، آنگاه پس از نگارش ۲۷,۰,۷,۱۵ به نگارش ۲۸,۰,۸,۰ خواهیم رسید؛ یعنی شماره لیست قابلیت‌ها یک واحد افزایش یافته و شماره لیست تصحیحات به مقدار «صفر» بازنشانی خواهد شد. با رعایت این قاعده، به راحتی با قیاس شناسه نگارش جدید با شناسه نگارش قبلی می‌توان دریافت که نگارش جدید شیء حاوی چه نوع تغییراتی نسبت به نگارش قبلی است:

- اگر فقط شماره لیست تصحیحات افزایش یافته باشد، به معنی آن است که نگارش جدید شیء نسبت به نگارش قبلی فقط تعدادی تصحیح و رفع اشکال را داراست.
- اگر شماره لیست قابلیت‌ها افزایش یافته باشد و شماره لیست تصحیحات به عدد صفر بازنشانی شده باشد، به معنی آن است که نگارش جدید شیء نسبت به نگارش قبلی فقط حاوی تعدادی تغییر از جنس افزودن و/یا تغییر قابلیت یا محتوی است.
- اگر شماره لیست قابلیت‌ها افزایش یافته باشد و شماره لیست تصحیحات به عدد یک بازنشانی شده باشد، به معنی آن است که نگارش جدید شیء نسبت به نگارش قبلی هم حاوی تعدادی تغییر از جنس افزودن و/یا تغییر



اعمال استراتژی مناسب ذخیره‌سازی را ممکن می‌سازد [۱۳].

نکته‌ای بدیهی که باید مدنظر قرار گیرد آن است که وقتی یک نگارش از شیء ارائه گردید، دیگر به هیچ وجه نباید در محتوای شیء تحت آن نگارش تغییری انجام گیرد؛ بلکه کلیه تغییرات باید در نگارش جدید شیء اعمال شود.

تغییر «شماره ترتیب نشر»، به صورت افزایش ساده یک واحدی در هر نگارش صورت می‌پذیرد. این شماره در حقیقت یک شماره سریال ترتیبی است که وجود آن در ابتدای شناسه نگارش مزایای زیر را در پی خواهد داشت:

- بیانگر تعداد نگارش‌های ارائه شده از شیء مربوطه از ابتدای ایجاد آن تاکنون است.
- در نگاه ترکیبی با فیلد دوم (شماره نسل) می‌تواند بیانگر میزان بلوغ محتوای شیء باشد. به عنوان مثال یک شیء را با شناسه نگارش ۹,۵,۰,۳ با شیء دیگری با شناسه نگارش ۴۱۷,۵,۰,۳ مقایسه کنید. گرچه هر دو از نسل پنجم بوده و فیلدهای «شماره لیست قابلیت‌ها» و «شماره لیست تصحیحات» یکسانی دارند، اما مقایسه فیلد اول دو شیء نشان می‌دهد که شیء دوم از بلوغ بسیار بیشتری برخوردار است.
- در هنگام روزآمدسازی، فیلد «شماره ترتیب نشر» به استفاده‌کننده می‌فهماند که چند انتشار (Release) بین آنچه که او دارد و آنچه

تغییراتی در جهت رفع اشکالات یا افزودن قابلیت‌ها که «سازگاری با نگارش قبل»^{۱۷} شیء را نقض می‌کند (مثلاً تغییرات ناسازگار با گذشته در «رابط برنامه‌نویسی نرم‌افزار» یا API)

برخلاف بعضی از طرح‌های تجاری معرفی شده در بخش دوم مقاله، در طرح پیشنهادی ما میزان تغییرات اعمال شده در شیء تأثیری در گام‌های تغییرات شماره‌های چهارگانه ندارند. بنابراین هرقدر هم که تعداد ایرادهای تصحیح شده بین دو نگارش پیاپی زیاد باشد، شماره لیست تصحیحات فقط یک واحد افزایش می‌یابد. متناظر با آن، هرقدر هم که تعداد قابلیت‌ها/محتوی اضافه شده یا تغییر یافته بین دو نگارش پیاپی زیاد باشد، شماره لیست قابلیت‌ها فقط یک واحد افزایش می‌یابد. همچنین هرقدر هم که تغییرات بنیادی انجام شده بین دو نگارش پیاپی شیء زیاد باشد، شماره نسل فقط یک واحد افزایش می‌یابد. ما معتقدیم بیان میزان تغییرات توسط گام‌های تغییر شماره‌ها ایده مناسب و دقیقی نیست. در عوض، ارائه گزارش‌های «رویدادنگاری تغییرات»^{۱۸} به همراه هر نگارش از شیء می‌تواند مخاطبین را به طور شفاف از میزان تغییرات نسبت به نگارش قبلی آگاه سازد.

نوع و حجم تغییرات بین نگارش‌ها (موسوم به دلتا) برای نحوه عمل‌کرد خودکار «سامانه‌های کنترل نگارش» مهم است. در این رابطه پروفیسور بنات‌الله و همکارانش یک طرح تطابق‌پذیر برای مدیریت نگارش مستندات ارائه کرده‌اند که تشخیص خودکار نوع و حجم تغییرات و تصمیم‌گیری خودکار سامانه VCS برای



نمایند. به این منظور علاوه بر آنکه بر روی نگارش‌های شاخه جدید منشعب از نگارش LTS (مثلاً نگارش ۳۸,۲,۰,۰ و فرزندان آن در (شکل-۱) کار می‌کنند، به طور موازی بر روی ارائه فرزندان شاخه اصلی خود LTS هم فعالیت دارند. به انتقال قابلیت‌های جدید و رفع اشکالات، از نگارش‌های شاخه(های) جدید به نگارش‌های شاخه اصلی LTS، عمل «سازگاری برگشتی»^{۲۱} می‌گویند. لازم به توضیح است که هر نگارش LTS و شاخه مربوط به آن فقط تا مدت معینی (که معمولاً توسط تولیدکننده اعلام می‌گردد) فعال و تحت پشتیبانی است. البته این بدان معنا نیست که تعداد LTS‌های فعال در یک درخت نگارش الزاماً فقط یکی است؛ بلکه در عمل ممکن است بنا به علل مختلفی شاهد چندین LTS فعال در یک برهه زمانی باشیم.

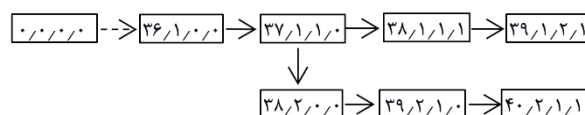
با توجه به مطالب فوق، گرچه RSN (شماره ترتیب نشر) در مسیر بین یک برگ تا ریشه درخت نگارش‌ها یکتا بوده و هر دفعه یک واحد افزایش می‌یابد، لیکن به غیر از RSN‌های موجود در تنه اصلی درخت (قبل از اولین نقطه انشعاب)، بقیه RSN‌ها الزاماً در کل درخت یکتا نیستند؛ بلکه ممکن است در شاخه‌های مختلف تکرار شوند. به بیان دیگر پس از وقوع انشعاب، «RSN اولین عضو هر دو شاخه» عبارت است از «RSN گره انشعاب بعلاوه یک». همانگونه که در (شکل-۱) مشاهده می‌شود، RSN نگارش‌های ۱,۱,۱ و ۲,۰,۰ هر دو برابر ۳۸ هستند که یک واحد بیشتر از RSN نگارش ۱,۱,۰ (گره انشعاب) است.

۴ - مرتب‌سازی لغوی و پیش‌صفری توسعه یافته

که می‌خواهد به آن روزآمد شود وجود داشته است. (به عنوان مثال روزآمدسازی از نگارش ۹,۵,۰,۳ به ۳۴,۵,۱,۱ که بیانگر یک فاصله ۲۵ تایی بین نگارش موجود و نگارش روزآمدسازی است)

- این فیلد در صورت رعایت نکات مندرج در بخش بعدی مقاله می‌تواند نقش تسهیل‌ساز برای «مرتب‌سازی لغوی» را نیز ایفا کند.

توجه به این نکته مهم است که توسعه و انتشار یک شیء اطلاعاتی (بخصوص بسته‌های نرم‌افزاری) الزاماً به شکل خطی اتفاق نمی‌افتد. بدین معنی که معمولاً در عمل شاهد آن هستیم تنه اصلی درخت نگارش‌های شیء از یک نقطه شروع به انشعاب می‌کند و نگارش‌های مختلف بعضاً به شکل موازی ارائه می‌شوند. یک نمونه ساده از این شرایط را می‌توانید در (شکل-۱) مشاهده نمائید:



(شکل-۱): یک نمونه ساده از انشعاب در درخت نگارش‌ها

دلیل این موضوع غالباً تصمیم تولیدکنندگان به ارائه نگارش‌های «تحت پشتیبانی طولانی مدت» موسوم به LTS^{۱۹} یا ESR^{۲۰} هستند. این نگارش‌ها در حقیقت نگارش‌هایی از شیء هستند که به صورت گره انشعاب در درخت نگارش‌ها دیده می‌شوند (نگارش ۳۷,۱,۱,۰ در (شکل-۱)). تولیدکنندگان خود را متعهد می‌دانند که نگارش‌های LTS را برای مدت طولانی پشتیبانی



منجر شود. به عنوان مثال به محل ترتیب قرار گرفتن شناسه نگارش ۱۰۰۰,۸,۶,۲۱۹ در لیست مرتب شده زیر توجه کنید:

۰۹۹,۴,۹۲,۸۱۱

۱۰۰۰,۸,۶,۲۱۹

۱۰۰,۵,۲۲,۱۱۱

۱۰۱,۵,۲۳,۰

دلیل اینکه در لیست فوق شناسه ۱۰۰۰,۸,۶,۲۱۹ حتی بالاتر از شناسه ۱۰۰,۵,۲۲,۱۱۱ قرار گرفته است، آن است که کد استاندارد نویسه «۰» (ممیز فارسی) از کد استاندارد نویسه «e» (صفر) بزرگ‌تر است.

در شگرد ابداعی و پیشنهادی ما تحت عنوان «پیش‌صفری توسعه یافته»، هنگامی که تعداد ارقام پیش‌بینی شده تکافوی نیاز را ندهد، یک نویسه e (مخفف کلمه «توسعه یافته»^۳) به ابتدای رشته عددی، قبل از رقم پرارزش جدید، افزوده می‌شود. این کار باعث می‌شود که حتی با افزوده شدن ارقام عدد، مرتب‌سازی لغوی نقض نشده و ترتیب صحیح حفظ شود. با بکار بردن این شگرد در شرایطی همچون مثال قبل خواهیم داشت:

۰۹۹,۴,۹۲,۸۱۱

۱۰۰,۵,۲۲,۱۱۱

۱۰۱,۵,۲۳,۰

...

۹۹۹,۸,۶,۲۱۸

در بسیاری از محیط‌های عملیاتی، نیاز به «مرتب‌سازی لغوی» اشیاء و فایل‌های مربوطه آنها است. این خاصیت منجر به آن می‌شود که هر نگارش به طور صحیح بعد از نیاکان و قبل از اعقاب خود قرار بگیرد [۱۴]. برای این منظور لازم است که دو موضوع در مورد فیلد «شماره ترتیب نشر» مد نظر قرار گیرد:

- اول، در هر سازمان متناسب با شرایط و نیازهای محلی، بایستی تعداد ارقام فیلد «شماره ترتیب نشر» پیشاپیش تعیین گردد.

- دوم، با رعایت طول عدد، شماره مذکور به شکل «پیش‌صفری توسعه یافته»^۲ درج شود.

«پیش‌صفری توسعه یافته» (به طور خلاصه e0)

شگرد ابداعی ما برای چیره شدن بر محدودیت روش معمول پیش‌صفری است. در روش معمول پیش‌صفری برای حفظ مرتب‌سازی لغوی، به تعداد باقیمانده تا تکمیل طول رشته، قبل از ارقام یک عدد را با صفر پر می‌کنند. اما اشکال این روش آن است که چنانچه طول رشته (رقم‌های پیش‌بینی شده برای عدد) پس از مدتی تکافوی نیاز را نکند و پر شود، با افزوده شدن رقم پرارزش جدید، «مرتب‌سازی لغوی» نقض می‌شود. به عنوان مثال اگر ظرفیت سه‌رقمی را برای عدد در نظر بگیریم، از ۰۰۰ شروع کنیم و به ترتیب به ۰۰۱، ۰۰۲، الی ۹۹۹ برسیم، پس از آن مجبوریم ۱۰۰۰ را داشته باشیم که در مرتب‌سازی لغوی بین ۱۰۰ و ۱۰۱ قرار می‌گیرد! این رفتار موقعی که عدد در یک شناسه نگارش مورد استفاده قرار می‌گیرد با توجه به کد نویسه‌هایی مثل ممیز و نقطه (در جدول استاندارد نویسه‌ها)، می‌تواند به رفتارهای ناخواسته دیگری نیز



بنابراین چنانچه به طور خلاصه بخواهیم در طرح QAVNS ساختار یک شناسه نگارش را به صورت الگویی بیان کنیم، خواهیم داشت:

$e0RSN.G.FF.CCC[-PRI][-BMD]$

که در آن:

- $e0RSN$ بیانگر «شماره ترتیب نشر» با استفاده از شگرد «پیش‌صفری توسعه یافته» است. پیشنهادی که به نظر می‌رسد برای اغلب محیط‌های کاری مناسب باشد، پیش‌بینی عددی سه رقمی برای شماره ترتیب نشر است.
- G بیانگر «شماره نسل» است و در اغلب محیط‌های کاری و کاربردهای مختلف، بیش از یک رقم نخواهد بود.
- FF بیانگر «شماره لیست قابلیت‌ها» است و در اغلب محیط‌های کاری و کاربردهای مختلف، بیشتر از دو رقم نخواهد شد.
- CCC بیانگر «شماره لیست تصحیحات» است و در اغلب محیط‌های کاری و کاربردهای مختلف، از سه رقم فراتر نخواهد رفت.
- PRI بیانگر «شناسه پیش‌انتشار» است که یک فیلد اختیاری بوده، با نویسه - از سایر فیلدهای سمت چپ جدا می‌شود و ساختاری به شکل زیر دارد:

$(a|b|rc)[1-9]$

نویسه a مخفف رشته α ، نویسه b مخفف رشته β و rc مخفف عبارت $release\ candidate$

$e1000,8,6,219$

$e1001,8,7,0$

$e1002,9,0,0$

در شگرد «پیش‌صفری توسعه یافته» چنانچه در شرایط خاصی، پر شدن ظرفیت ارقام بیش از یک دفعه رخ دهد، می‌توان با اعمال شگرد به طور مجدد و افزودن نویسه e به ابتدای رشته به صورت تکراری، ضمن افزایش تعداد ارقام عدد، همچنان حفظ ترتیب لغوی را تضمین نمود.

به عنوان مثال:

$e9999,14,5,72$

$ee10000,14,5,73$

$ee10001,14,6,1$

...

$ee99999,84,30,114$

$eee100000,84,30,115$

$eee100001,84,30,116$

البته بدیهی است که چنانچه در یک سازمان متناسب با شرایط و نیازهای محلی، تعداد ارقام عدد «شماره ترتیب نشر» پیشاپیش به طور واقع‌بینانه‌ای تعیین گردد، احتمال بروز شرایط فوق به صورت مکرر کاهش خواهد یافت.

۵ - ساختار کامل شناسه نگارش در QAVNS



جدید، یک واحد افزایش یابد. فیلد PRI نیز در کلیه کاربردها مفهوم و روند یکسانی را داراست و بیانگر وضعیت ارائه شیء اطلاعاتی است که می‌تواند در یکی از حالات «ارائه داخلی» (alpha)، «ارائه خصوصی» (beta)، و «نامزد برای ارائه عمومی» (release candidate) باشد. فیلد BMD نیز همان‌گونه که در بخش قبل مورد اشاره قرار گرفت، حاوی ابر داده‌هایی است که سازنده شیء اطلاعاتی به تشخیص خود لازم می‌داند آن‌ها را در شناسه نگارش شیء درج نماید.

با توجه به موارد مندرج در فوق، در کاربردهای ذیل تنها به توضیح پیرامون مفهوم و روند تغییر سه فیلد دیگر خواهیم پرداخت.

۱) بسته‌های نرم‌افزاری

فیلد CCC در صورت رفع اشکالات نرم‌افزاری (موسوم به Bug) یک واحد افزایش می‌یابد. همچنین اصلاحاتی در ظاهر نوشتاری کد منبع که جنبه تزئینی، توگذاری، فاصله‌دهی بین کلمات و/یا خطوط، تغییر در بزرگی و کوچکی حروف و موارد مشابه دیگر (بدون هیچ‌گونه تغییر محتوایی) داشته و برای کمک به خوانایی بیشتر کد منبع انجام می‌شود، جزو این دسته محسوب می‌گردد.

فیلد FF در صورت هرگونه تغییرات محتوایی که منجر به افزوده شدن قابلیت‌های جدید به نرم‌افزار شود، یک واحد افزایش می‌یابد. البته به شرط آنکه تغییرات مذکور بنیادی نبوده و در ضمن منجر به تغییر در رابط برنامه‌نویسی نرم‌افزار (API) هم نشود. چنانچه علاوه بر این تغییرات محتوایی، نسبت به نگارش قبل تغییرات اصلاحی/رفع اشکال نیز در نرم‌افزار رخ داده باشد، فیلد

است. هرکدام از آن‌ها به طور اختیاری می‌تواند با یک رقم ۱ تا ۹ دنبال شود.

به عنوان مثال:

137.4.1.28-b3

- BMD بیانگر «آبر داده‌های مربوط به ساخت» است که یک فیلد اختیاری بوده، با یک نویسه - از سایر فیلدهای سمت چپ جدا می‌شود و بیشتر برای بسته‌های نرم‌افزاری کاربرد دارد. این ابر داده‌ها می‌تواند شامل تاریخ ساخت، شناسه شخص سازنده و سایر موارد مورد نیاز باشد. به عنوان مثال:

137.4.1.28-rc2-20161123

البته بدیهی است با توجه به اینکه مرتب‌سازی لغوی کل رشته توسط فیلد اول (eORSN) تأمین می‌گردد، سایر فیلدها از لحاظ اندازه محدودیتی ندارند.

۶ - عملکرد طرح پیشنهادی

همانگونه که در ابتدای بخش قبل آمد، QAVNS قابلیت آن را دارد که بدون تغییر ساختار برای ردیابی نگارش‌ها و ویرایش‌های مختلف پرونده‌ها، بسته‌های نرم‌افزاری، مستندات خروجی پروژه‌ها، طرح‌ها، قوانین، آئین‌نامه‌ها، شیوه‌نامه‌ها، نقشه‌ها، گرافیک‌ها، مستندات اداری و سایر موارد مشابه در محیط‌های مختلف به خوبی بکار رود. در این بخش به عنوان نمونه تطبیق طرح مذکور بر تعدادی از کاربردهای مختلف را بیان می‌کنیم.

لازم به توضیح است که فیلد eORSN در کلیه کاربردها مفهوم و روند یکسانی را داراست و بایستی به عنوان یک شماره ترتیب، به ازای هر ویرایش/نگارش



- *CCC* به «یک» بازنشانی می‌شود؛ در غیر این صورت فیلد *CCC* به «صفر» بازنشانی می‌شود.
- تغییر در بزرگی و کوچکی حروف
- فیلد *G* در صورت اعمال هرگونه تغییرات بنیادی که ساختار نرم‌افزار، یا ظاهر رابط کاربری (*UI*) آن و یا رابط برنامه‌نویسی نرم‌افزار (*API*) را متحول کند، یک واحد افزایش می‌یابد. در این حالت فیلدهای *CCC* و *FF* به «صفر» بازنشانی خواهند شد.
- کلید تغییرات مربوط به قالب‌بندی و اعمال سبک‌ها
- فیلد *FF* به «صفر» بازنشانی خواهند شد.
- کلید تغییرات مربوط به صفحه‌بندی
- تغییرات در پاراگراف‌بندی متن بدون تغییر در محتوای جملات
- تغییر در اندازه و رنگ کل یا بخشی از جداول، نمودارها و یا تصاویر
- تغییرات در ظاهر لیست‌های علامت یا شماره‌دار یا تبدیل پاراگراف‌های عادی به اینگونه لیست‌ها
- افزودن یا حذف انواع شکست به/از متن
- منظور از «مستندات» کلید پرونده‌هایی هستند که حاوی متن، جدول، نمودار و موارد مشابه دیگر هستند. لذا انواع نامه‌ها و مکاتبات، گزارش‌ها، کتاب‌ها، پایان‌نامه‌ها، مقالات و همچنین انواع متون حقوقی از قبیل قراردادهای تفاهم‌نامه‌ها، قوانین، آئین‌نامه‌ها و شیوه‌نامه‌ها/دستورالعمل‌ها و سایر موارد مشابه مشمول آن هستند.
- در مستندات، فیلد *CCC* در صورت رفع اشکالات یا اصلاح ظاهری (و نه محتوایی) یک واحد افزایش می‌یابد. پاره‌ای از اینگونه تغییرات ظاهری عبارتند از:
 - تغییر قالب ذخیره‌سازی در پرونده
 - اصلاح اشتباهات تایپی یا املایی
 - تغییراتی که جنبه تزئینی یا خوانایی بیشتر دارند
 - تغییر در علائم سجاوندی
 - تغییر قلم‌ها و اندازه آنها
 - تغییر رنگ‌های متن، روزمینه و پس‌زمینه
 - توگذاری متن
 - فاصله‌دهی بین کلمات و/یا خطوط
- تغییر در اندازه و رنگ کل یا بخشی از جداول، نمودارها و یا تصاویر
- تغییرات در ظاهر لیست‌های علامت یا شماره‌دار یا تبدیل پاراگراف‌های عادی به اینگونه لیست‌ها
- افزودن یا حذف انواع شکست به/از متن
- در مستندات، فیلد *FF* در صورت هرگونه تغییرات محتوایی که منجر به دگرگونی، یا افزوده شدن یا حذف بخشی از متن/جدول/نمودار شود، یک واحد افزایش می‌یابد. البته به شرط آنکه تغییرات مذکور ساختاری نباشد. چنانچه علاوه بر این تغییرات محتوایی، نسبت به نگارش قبل تغییرات اصلاحی/رفع اشکال نیز در ظاهر محتوا رخ داده باشد، فیلد *CCC* به «یک» بازنشانی می‌شود؛ در غیر این صورت فیلد *CCC* به «صفر» بازنشانی می‌شود.
- فیلد *G* در صورت اعمال هرگونه تغییرات بنیادی که ساختار محتوا را متحول کند، یک واحد افزایش می‌یابد. در این حالت فیلدهای *CCC* و *FF* به «صفر» بازنشانی خواهند شد. نمونه‌ای از تغییرات بنیادی در مستندات عبارتند از:
 - تغییر، افزایش یا کاهش فصل‌بندی متن



تغییرات محتوایی، نسبت به نگارش قبل تغییراتی از جنس اصلاح سطحی/رفع اشکال نیز در محتوا رخ داده باشد، فیلد CCC به «یک» بازنشانی می‌شود؛ در غیر این صورت فیلد CCC به «صفر» بازنشانی می‌شود. نمونه‌ای از تغییراتی که منجر به افزایش مقدار فیلد FF می‌شوند عبارتند از:

- تغییر شکل، افزودن یا کاستن عناصر بصری از گرافیک

- تغییر، افزایش یا کاهش لایه‌های گرافیکی (در تصاویر) یا اطلاعات مکانی (در نقشه‌ها)

- تغییرات بصری در نمودارها ناشی از تغییر مقادیر مربوطه

- تغییر، افزایش یا کاهش اطلاعات توصیفی در نقشه‌ها

فیلد G در صورت اعمال هرگونه تغییرات بنیادی که ساختار محتوا را متحول کند، یک واحد افزایش می‌یابد. در این حالت فیلدهای CCC و FF به «صفر» بازنشانی خواهند شد. به عنوان نمونه‌ای از تغییرات بنیادی در گرافیک‌ها:

- طراحی مجدد گرافیک (به شرط ابقای ارتباط مفهومی آن با نگارش قبلی)

- تغییر بیش از نیمی از لایه‌های بصری/اطلاعاتی

۷ - نتایج آزمایشی

در این مقاله یک طرح تطابق‌پذیر شماره‌گذاری نگارش‌ها برای اشیای اطلاعاتی جهت استفاده در مهاجرت زنده ماشین‌های مجازی ارائه گردید.

- تغییر، افزایش یا کاهش ورق‌ها در پرونده‌های صفحه‌گستره

- بازنویسی بیش از نصف متن

- هرگونه تغییری در متن که منجر به تغییر محتوایی بیش از نصف «فهرست مطالب» شود

- هرگونه تغییری در متن که منجر به کاهش یا افزایش بیش از پنجاه درصد حجم متن گردد

۳) گرافیک‌ها

منظور از «گرافیک‌ها» کلیه پرونده‌هایی هستند که نمایانگر عناصر بصری ساکن هستند. لذا انواع نقشه، عکس، نقاشی، طراحی، نمودار و سایر موارد مشابه مشمول آن هستند.

در گرافیک‌ها، فیلد CCC در صورت رفع اشکالات یا اصلاح سطحی (و نه عمیق) یک واحد افزایش می‌یابد. پاره‌ای از اینگونه تغییرات سطحی عبارتند از:

- تغییر قالب ذخیره‌سازی یا تغییر مقادیر تنظیمی پرونده

- تغییراتی که جنبه تزئینی یا خوانایی بیشتر دارند

- تغییر قلم‌ها و اندازه آنها

- تغییر رنگ‌ها

- تغییر دقت تفکیک تصویر

- تغییر عناصر گرافیکی مورد استفاده در راهنما

در گرافیک‌ها، فیلد FF در صورت هرگونه تغییرات غیرسطحی، یک واحد افزایش می‌یابد. البته به شرط آنکه تغییرات مذکور ساختاری نباشد یا منجر به تغییر کلی در ظاهر گرافیک نشود. چنانچه علاوه بر این



علاوه بر این، طرح پیشنهادی (QAVNS) می‌تواند به عنوان یک طرح جامع و کارا در کلیه محیط‌های اداری، عملیاتی، هنری، توسعه نرم‌افزار، تولید مستندات، قانون‌گذاری، انتشار کتاب‌ها و سایر موارد مشابه تبدیل به استاندارد همگانی شده و با موفقیت به کار گرفته شود. استاندارد شدن این طرح کمک بزرگی به یکنواخت شدن درک همگان از شماره‌گذاری نگارش‌ها و ویرایش‌ها خواهد بود. پیشنهاد می‌شود ادارات، سازمان‌ها، شرکت‌ها، نهادهای آموزشی و مؤسسات تولیدی، روش QAVNS یاد شده را به عنوان یک استاندارد فراگیر معرفی و در کلیه سطوح سازمانی به کار گیرند.

۹ - مراجع

- [1] Tajamolian, Majid, and Mohammad Ghasemzadeh. "A Version Numbering Scheme for Informational Objects Used in VM Live Migration." *Journal of Advances in Computer Engineering and Technology* 5, no. 4 (2019): 245-254.
- [2] Tajamolian, Majid, and Mohammad Ghasemzadeh. "A Version Numbering Scheme for Informational Objects Used in VM Live Migration." *Journal of Advances in Computer Engineering and Technology* 5, no. 4 (2019): 245-254.
- [3] Stallman, Richard. "GNU coding standards." (1992).
- [4] Erenkrantz, Justin R. "Release management within open source projects." *In Proc. 3rd. Workshop on Open Source Software Engineering*. 2003.
- [5] Tajamolian, Majid, and Mohammad Ghasemzadeh. "A Version Numbering Scheme for Informational Objects Used in VM Live Migration." *Journal of Advances in Computer Engineering and Technology* 5, no. 4 (2019): 245-254.

طرح پیشنهادی در یک محیط پژوهشی دانشگاهی و یک شرکت دانش‌بنیان با موفقیت اجرا شده است. بکارگیری طرح جدید در هر دو محیط، منجر به رفع مشکلات موجود در نظام‌مند کردن شناسه ویرایش‌ها و نگارش‌ها گردید.

در جریان تحقیق همچنین QAVNS را به تدریج توسعه دادیم تا تمامی نیازهایمان را به عنوان یک طرح واحد برای نظام‌مند کردن شناسه ویرایش‌ها و نگارش‌های و خط سیر آن‌ها در انواع اشیاء اطلاعاتی پوشش دهد. اشیائی که در دو محیط مذکور برای آن‌ها از QAVNS استفاده کردیم عبارت بودند از: محصولات نرم‌افزاری، مستندات فنی، گزارش‌های خروجی پروژه‌ها، مستندات اداری، قراردادهای آئین‌نامه‌ها، شیوه‌نامه‌ها و طرح‌های گرافیکی.

نتایج آزمایشی و تحلیلی حکایت از سزاواری و مؤثر بودن طرح پیشنهادی در برآورده سازی انتظارات داشتند. همچنین مشخص گردید که آموزش QAVNS به کارکنان و افراد همکار در فعالیت‌های مختلف در مدت زمانی کوتاه امکان‌پذیر است. ضمناً این طرح بخوبی می‌تواند برای خودکارسازی انتخاب روش مناسب در مهاجرت زنده ماشین‌های مجازی به کار رود.

۸ - نتیجه‌گیری و پیشنهادها

طرح پیشنهادی (QAVNS) ابزاری مفید برای رصد نوع و میزان تغییرات در حافظه یک ماشین مجازی (به عنوان یک شیء اطلاعاتی) با هدف خودکارسازی انتخاب روش مناسبتر جهت مهاجرت زنده ماشین‌های مجازی است.



[14] Keller, Arthur M., and Jeffrey D. Ullman. "A version numbering scheme with a useful lexicographical order." *In Proceedings of the Eleventh International Conference on Data Engineering*, pp. 240-248. IEEE, 1995.



مجید تجملیان در سال ۱۳۹۷ دکترای خود را در رشته مهندسی کامپیوتر از دانشگاه یزد دریافت کرد. وی همچنین یک مشاور معروف فناوری اطلاعات در استان یزد است. او دوره

کارشناسی ارشد خود را در مهندسی نرم افزار رایانه، در سال ۱۳۷۸ در دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران) گذراند و لیسانس خود را در مهندسی نرم افزار رایانه، در سال ۱۳۷۳ از دانشگاه صنعتی اصفهان اخذ کرد. حوزه‌های پژوهشی مورد علاقه ایشان شامل نرم افزارهای آزاد و متن‌باز (FLOSS)، رایانش ابری، ماشین‌های مجازی، امنیت فناوری اطلاعات، سامانه‌های عامل و مدیریت شبکه‌های ترکیبی است.



محمد قاسم زاده دانشیار علوم کامپیوتر و مهندسی کامپیوتر در دانشگاه یزد است. در سال ۱۳۶۸ لیسانس خود را در رشته علوم کامپیوتر و مهندسی از

دانشگاه شیراز و کارشناسی ارشد رشته هوش مصنوعی و رباتیک را از دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران) در سال ۱۳۷۴ دریافت کرد. برای برنامه پژوهش دکتری خود، در دانشگاه‌های تریور و پوتسدام در آلمان، از فوریه ۲۰۰۲ تا فوریه ۲۰۰۶ کار کرد. وی مدرک

[6] Tajmolian, Majid, and Mohammad Ghasemzadeh. "A Version Numbering Scheme for Informational Objects Used in VM Live Migration." *Journal of Advances in Computer Engineering and Technology* 5, no. 4 (2019): 245-254.

[7] Tajmolian, Majid, and Mohammad Ghasemzadeh. "A Version Numbering Scheme for Informational Objects Used in VM Live Migration." *Journal of Advances in Computer Engineering and Technology* 5, no. 4 (2019): 245-254.

[8] Intergovernmental Oceanographic Commission. "Ocean Data Standards Volume 2. Recommendation to adopt ISO 8601: 2004 as the standard for the representation of dates and times in oceanographic data exchange." (2011).

[9] Kuhn, Markus. "A summary of the international standard date and time notation." (1995).

[10] Tajmolian, Majid, and Mohammad Ghasemzadeh. "A Version Numbering Scheme for Informational Objects Used in VM Live Migration." *Journal of Advances in Computer Engineering and Technology* 5, no. 4 (2019): 245-254.

[11] Conradi, Reidar, and Bernhard Westfechtel. "Version models for software configuration management." *ACM Computing Surveys (CSUR)* 30, no. 2 (1998): 232-282.

[12] Bauml, Jaroslav, and Premek Brada. "Automated versioning in OSGi: A mechanism for component software consistency guarantee." *In 2009 35th Euromicro Conference on Software Engineering and Advanced Applications*, pp. 428-435. IEEE, 2009.

[13] Benatallah, Boualem, Mehregan Mahdavi, Phuong Nguyen, Quan Z. Sheng, Lionel Port, and Bill McIver. "An adaptive document version management scheme." *In International Conference on Advanced Information Systems Engineering*, pp. 46-62. Springer, Berlin, Heidelberg, 2003.



دکترای خود را در "علوم نظری رایانه"، در نوامبر ۲۰۰۵ به اتمام رساند. او همچنین فرصت مطالعاتی خود را به عنوان مهمان-پژوهشگر پسادکتر در HPI آلمان از فوریه تا نوامبر ۲۰۱۶ گذراند. حوزه‌های پژوهشی مورد علاقه ایشان شامل طراحی و تحلیل الگوریتم، روشهای هوش مصنوعی، پردازش زبان طبیعی، داده‌های حجیم و امنیت نرم‌افزار است.