علوم و مهندسی کامپیوتر
نشریه علمی پژوهشی انجمن کامپیوتر ایران
مجلد ۷، شماره ۱ و ۳ (الف)، ۱۳۸۸
صفحات ۲۲-۴۰
مقاله عادی

ارائه مدلی قابل اجرا از معماری نرم‌افزار به منظور ارزیابی کارایی

سیما عمادی[۱] فریدون شمس[۲]

[۱]دانشکده فنی و مهندسی، دانشگاه آزاد اسلامی واحد میبد، میبد، ایران
[۲]دانشکده مهندسی برق و کامپیوتر، دانشگاه شهید بهشتی، تهران، ایران

چکیده

با افزایش روزافزون استفاده از نمودارهای زبان مدلسازی یکپارچه برای توصیف معماری نرم‌افزار و همچنین اهمیت ارزیابی نیازهای غیروظیفه‌مندی در سطح معماری نرم‌افزار، ایجاد یک مدل قابل اجرا از این نمودارها ضروری است. از طرفی، نمودارهای زبان مدلسازی یکپارچه، قابلیت ارزیابی نیازهای غیروظیفه‌مندی سیستم را بطور مستقیم ندارند، لذا بایستی با حفظ سادگی و حفظ ویژگی‌های این نمودارها، قابلیت ارزیابی این نیازها نیز به آن‌ها اضافه شود. برای رسیدن به این هدف بایستی مدلی قابل اجرا از این نمودارها ایجاد شود. این مدل‌های قابل اجرا می‌توانند مدل‌های شبکه صف، شبکه پتری، جبر فرآیندی، فرآیندهای اتفاقی، و غیره باشند. در این مقاله، ابتدا فرض شده که معماری یک سیستم با استفاده از سه نمودار مورد کاربری، ترتیب و مولفه زبان مدلسازی یکپارچه توصیف شده است. سپس نقش این نمودارها در زمینه ارزیابی کارایی و حاشیه‌نویسی‌های مربوط به آن بررسی شده و در نهایت، طی الگوریتمی به یک مدل قابل اجرا، یعنی شبکه پتری رنگی تصادفی تعمیم‌یافته، تبدیل گردیده است.

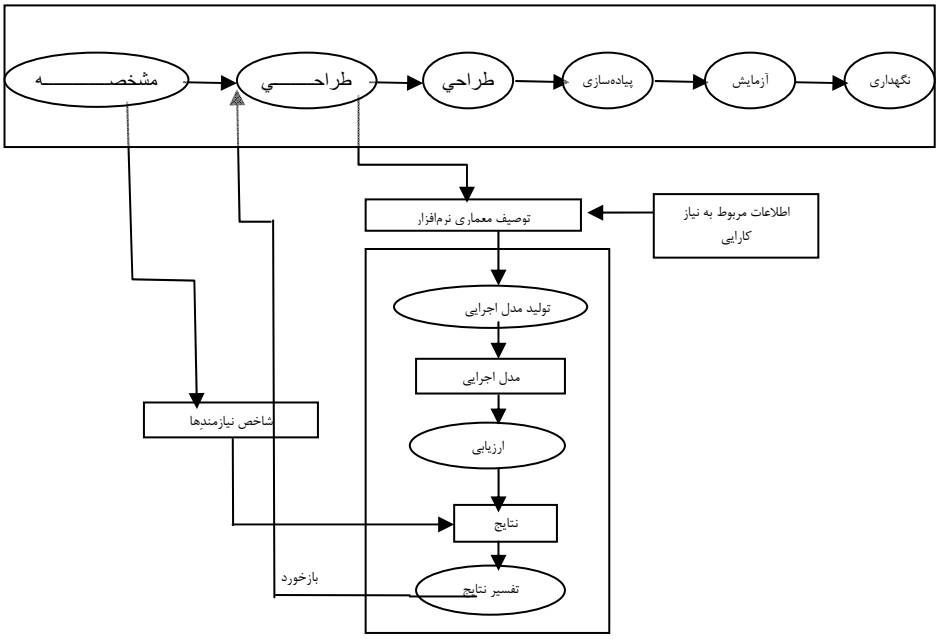
کلمات کلیدی: معماری نرم‌افزار، ارزیابی کارایی، نمودار مورد کاربری، نمودار مؤلفه، نمودار ترتیب، شبکه پتری تصادفی تعمیم‌یافته، نیازهای غیروظیفه‌مندی، مدل قابل اجرا.

۱- مقدمه

با افزایش کاربرد سیستم‌های نرم‌افزاری پیچیده، توسعه آن‌ها بر مبنای اصول و متدولوژی‌هایی که ضمن کاهش هزینه سرمایه‌گذاری شده، کلیه نیازهای مورد انتظار صاحبان سهام (اعم از نیازهای وظیفه‌مندی و غیروظیفه‌مندی) را برآورده کند، ضروری است.

مراحل اولیه طراحی، تاثیر بسیار زیادی بر روی توسعه نرم‌افزار و کیفیت محصول نهایی دارد. بنابراین تصمیمات غیردقیق در فازهای اولیه ممکن است منجر به کارهای مجدد وسیعی گردد، از طرفی بررسی این نیازها قبل از مرحله طراحی و پیاده‌سازی، هزینه و زمان کمتری را صرف می‌کند. روش‌های معمولی توسعه نرم‌افزار، تنها متمرکز بر روی بررسی صحت[۱] نرم‌افزار هستند و به مسائلی همچون ارزیابی کارایی، در مراحل بعدی توسعه، توجه می‌نمایند. بنابراین بایستی به سمت روش‌های جدید توسعه حرکت نمود که قادر باشیم در حین مشخص‌سازی نیازها، فاز معماری و طراحی، عواملی که سیستم را تحت تاثیر قرار می‌دهند یعنی نیازهای غیروظیفه‌مندی را نیز شناسایی نمائیم. با ارزیابی معماری می‌توان تحلیل کرد که معماری پیشنهادی تا چه اندازه، نیازهای مورد انتظار به ویژه نیازهای غیروظیفه‌مندی را برآورده می‌کند و نسبت به رفع مشکلات این نیازها به دلیل معماری نامناسب آن‌ها قبل از صرف هزینه و زمان جهت پیاده سازی، در همان ابتدای فرآیند توسعه اقدام نمود.

اما نیازهای غیروظیفه‌مندی به دلایل زیر به ندرت در طی توسعه نرم‌افزار بخصوص در مراحل اولیه فرآیند توسعه مورد توجه قرار می‌گیرند [۱]: بالا بودن سطح انتزاع، بیان رسمی و پیچیدگی آن‌ها، مشکل‌بودن ارزیابی و درنظرگرفتن آن‌ها در طی فرآیند توسعه، برخورد بعضی از نیازهای غیروظیفه‌مندی مثل دردسترس پذیری و کارایی با یکدیگر و تعامل قوی آن‌ها با نیازهای وظیفه‌مندی و در نهایت فاصله دانشی مابین معماران و تحلیلگران نیازهای غیروظیفه‌مندی.



شکل ۱- اجتماع ارزیابی کارایی در سطح معماری نرم‌افزار

علیرغم دلایل فوق، بهترین زمان برای سنجش رفتار قابل ارزیابی سیستم، موقعی است که معماری نرم‌افزار آن سیستم ایجاد شده‌است. معماری نرم‌افزار، بعنوان اولین محصول طراحی، خروجی‌های مناسبی ارائه می‌دهد که با استفاده از آن‌ها می‌توان رفتار قابل ارزیابی سیستم، یعنی نیازهای غیروظیفه‌مندیی مثل امنیت، قابلیت اطمینان، قابلیت استفاده، قابلیت تغییر، ایستایی و کارایی را سنجید [۲، ۳ و ۳].

با استفاده از دو روش فرآیندگرا[۲] و محصول‌گرا[۳] می‌توان نیازهای غیروظیفه‌مندی را صریحاً در سطح توسعه نرم‌افزار نشان داد. روش فرآیندگرا، نیازهای غیروظیفه‌مندی را در کنار نیازهای وظیفه‌مندی به صورت عناصر مؤثر در فرآیند توسعه نرم‌افزار در نظر می‌گیرد، در حالیکه روش محصول‌گرا، نیازهای غیروظیفه‌مندی را در محصول نهایی درنظر می‌گیرد. در این روش، نیازهای وظیفه‌مندی در محصول، اندازه‌گیری و سپس محصول، استفاده می‌شود. سپس صفات کیفیتی نرم‌افزار با هم مقایسه می‌شوند [۱]. اغلب روش‌های پیشنهادشده از نوع محصول‌گراست و بر تعریف علامتگذاری‌هایی متمرکزند که برای بیان نیازهای غیروظیفه‌مندی محصول نهایی استفاده می‌شوند. این علامتگذاری‌ها، نیازهای غیروظیفه‌مندی را به شکل گزاره‌هایی بیان می‌کنند. از مشکلات این روش‌ها می‌توان به مواردی همچون مستقل بودن آن‌ها، استفاده از علامتگذاری‌های مخصوص آن روش و ترکیب نشدن آن با مراحل دیگر توسعه نرم‌افزار اشاره‌نمود. همچنین دانش ناچیز توسعه‌دهندگان و معمار نرم‌افزار موجب می‌شود علامتگذاری‌های مبتنی بر سیستم‌های منطقی به راحتی برای آنان قابل استفاده نباشد. به‌علاوه بعضی از این روش‌ها، یکپارچه‌نمودن ارزیابی نیازهای غیروظیفه‌مندی را درسطح توسعه نرم‌افزارها خیلی ضعیف و بیشتر آن را در انتهای توسعه انجام می‌دهند. نهایتاً اینکه یکپارچه‌نمودن علامتگذاری‌های مورد استفاده در روش‌های فرآیندگرا با علامتگذاری‌های استفاده‌شده برای توصیف نیازهای غیروظیفه‌مند مشکل است.

یکی از راه‌حل‌ها برای رفع مشکل ارزیابی نیازهای غیروظیفه‌مندی و کم‌نمودن فاصله دانشی بین معمار و تحلیلگران، ایجاد یک مدل قابل‌اجرا از معماری است. یک مدل قابل‌اجرا از معماری، توصیفی رسمی از معماری محسوب می‌شود که با آن می‌توان قبل از پیاده‌سازی معماری، رفتار سیستم نهایی را بررسی نمود. از مزایای معماری قابل‌اجرا، می‌توان به مواردی همچون، بدست آوردن میزان کارایی و قابلیت اطمینان سیستم، بدست آوردن گلوگاه‌ها و نقاط بن‌بست در سیستم و محاسبه میزان موثربودن معماری قبل از پیاده‌سازی، اشاره نمود. خروجی‌های مدلسازی با انعکاس بعضی مشخصات صفات کیفیتی به معماران و طراحان سیستم‌های پیچیده، قدرت نمایش بصری کل سیستم را می‌دهند، زیرا گرامر و معنای[۴] تعریف‌شده در آن‌ها رسمی است. همچنین امکان ارزیابی و بررسی دقیق آن‌ها را با کمک ابزارهای تحلیل مناسب، قبل از ساخت واقعی آن‌ها فراهم می‌کنند [۵ و ۶]. البته باید توجه داشت که مدلسازی سیستم‌های پیچیده به تحویل به موقع سیستم به بازار خللی وارد نکند.

مدل‌های اجرایی معماری از روی مولفه‌های عملیاتی سیستم، کارها، پیغام‌ها، رویدادها، زمانبندی و مکانیزم انتقال پیغام‌ها استخراج می‌گردند. این مدل‌ها قابلیت بررسی تعامل مولفه‌ها، چگونگی تخصیص کارها به مولفه‌ها، کنترل کارها و خصوصیات زمانی و مکانی سیستم را به معماری می‌دهند.

همانطور که شکل ۱ نشان می‌دهد، از اهداف دیگر ایجاد مدل اجرایی، اجتماع فرآیند تحلیل نیازهای غیروظیفه‌مندی و نیازهای وظیفه‌مندی در سطح معماری نرم‌افزار است. در این شکل بیضی‌ها، فعالیت‌های انجام شده توسط فرآیندها و مستطیل‌ها، خروجی هر فرآیند را نشان می‌دهند. مطابق این شکل معمار بایستی به توصیفات معماری نرم‌افزار، پارامترهای مورد نیاز برای ارزیابی نیاز غیروظیفه‌مندی را اضافه‌نماید و سپس به یک مدل قابل اجرا، تبدیل گردد. سپس مدل اجرایی حاصل با استفاده از شاخص‌های نیاز غیروظیفه‌مندی موجود، ارزیابی و نتایج حاصل از این ارزیابی، تجزیه و تحلیل می‌شود. در نهایت، نتایج حاصل به طراحی معماری نرم‌افزار، پسخورد می‌شود.

یکی از مهمترین نیازهای غیروظیفه‌مندی که توسعه نرم‌افزار را تحت تاثیر قرار می‌دهند، کارایی است. زیرا مشکلات کارایی می‌تواند باعث تغییرات قابل توجهی در هر مرحله از چرخه زندگی نرم‌افزار بخصوص مراحل اولیه توسعه (یعنی معماری، طراحی، و یا فاز نیازمندی‌ها) گردد. تحقیقات مختلفی صورت گرفته‌است که نشان می‌دهد هر یک از صفات کیفیتی تحت تاثیر کدام مرحله از فرآیند توسعه نرم‌افزار قرار می‌گیرند [۷]. در این راستا نشان داده‌شده که در مرحله معماری می‌توان کارایی را ارزیابی نمود.

کارهای مختلفی در رابطه با ارزیابی نیازهای غیروظیفه مند، بخصوص کارایی در سطوح اولیه توسعه نرم‌افزار و معماری نرم‌افزار انجام شده‌است [۸، ۹ و ۱۰].

مرجع [۱۱] روش‌های مختلف ارزیابی کارایی مبتنی بر نمودارهای زبان مدلسازی یکپارچه[۵]، در سطح معماری نرم‌افزار را بطور کامل مرور نموده‌است. در [۱۲] نمودار ترتیب با نمودار وضعیت، به منظور ارزیابی کارایی، ترکیب شده‌است. در [۱۳] الگوریتمی ارائه‌شده‌است که به منظور ارزیابی کارایی معماری، یک مدل شبکه صف از MSC[۶]ها، ایجاد نموده‌است.

در [۱۴] نمودار ترتیب، استقرار و موارد کاربری با هم به یک مدل شبکه صف تبدیل شده‌اند. در این الگوریتم از نمودار ترتیب، یک گراف اجرایی، حاصل می‌شود، سپس با استفاده از نمودار استقرار، محدودیت‌های سخت افزار، نمایش داده‌می‌شود. در هر دو مرجع [۱۳] و [۱۴] مدل قابل تحلیل حاصل، یک شبکه صف است. در [۱۵] نمودار مورد کاربری و ارتباطات میان آن‌ها به OSAN تبدیل شده‌است. در [۱۶] ساده‌ترین ساختار نمودار موردکاربری و تنها ارتباط "شامل‌شدن" به شبکه پتری رنگی تبدیل شده‌است. همچنین روش‌های مختلفی برای تبدیل نمودار ترتیب به شبکه‌های پتری صورت‌گرفته‌است. در [۱۷] نمودار ترتیب و کلیه ساختارهای آن به شبکه پتری تصادفی تعمیم‌یافته تبدیل شده است. در [۱۸] نیز ساده‌ترین ساختارهای موجود در یک نمودار ترتیب را به شبکه پتری تبدیل‌کرده‌است. در [۱۹] نمودار ترتیب به شبکه پتری تصادفی خوش فرم تبدیل شده‌است.

اما کاربرد هر یک از این روش‌ها در مثال‌های کاربردی پیچیده و واقعی، محدود است و هیچ کدام از روش‌ها بطور کامل این نیازها را ارزیابی نمی‌کنند. همچنین فاصله موجود بین معمار و تحلیل‌گر نیازها بطور کامل از بین نرفته‌است. از طرفی در اغلب روش‌ها، مدل اجرایی حاصل برای ارزیابی کارایی، مدل شبکه صف است و در صورت نیاز به استفاده از آن در ارزیابی نیازهای دیگر بایستی مدلی جدید از سیستم بهمراه فاکتورهای احتمالاتی جدید درنظر گرفته شود.

تفاوت روش ارائه‌شده در این مقاله با کارهای دیگر این است که معمار، نرم‌افزار را در سطح معماری نرم‌افزار بدون دخالت تحلیلگر ارزیابی نموده و اطلاعات نیاز کارایی توسط او به توصیفات معماری حاشیه‌نویسی می‌شود.

در این مقاله، روشی برای ارزیابی بهینه ارائه نمی‌شود بلکه سعی خواهدشد بهترین مشخصات روش‌های موجود برای ارزیابی نیاز کارایی در سطوح دیگر انتخاب و در ارزیابی معماری استفاده گردد تا فاصله دانشی بین معمار و تحلیل‌گر کاهش یابد. از طرف دیگر چگونگی جمع‌آوری اطلاعات مربوط به نیاز کارایی و چگونگی ارزیابی و تحلیل نتایج آن نیز بررسی نمی‌شود.

در این مقاله، به توصیفات معماری پارامترهای مربوط به کارایی اضافه می‌شود و طی الگوریتمی به مدل اجرایی نگاشت می‌یابند که توصیف معماری و الحاق اطلاعات مربوط به نیاز کارایی توسط معمار صورت می‌گیرد و نگاشت آن‌ها به یک مدل رسمی نیز بطور خودکار یا توسط معمار انجام می‌شود. تمرکز اصلی این تحقیق بر روی تعریف یک سطح میانی بین مدل توصیفات معماری و مدل اجرایی است که برای رسیدن به آن، روش جدیدی ارائه‌شده و یک مدل رسمی و اجرایی از معماری استخراج می‌گردد.

همانطور که مشخص است، در سطح معماری نرم‌افزار هیچ‌گونه اطلاعاتی در رابطه با بستر سخت‌افزاری که سیستم نهایی بر روی آن اجرا خواهد شد، وجود ندارد. لذا در این مقاله، ابتدا این نکات را درنظرگرفته و سپس با تبدیل خودکار علامتگذاری‌های نرم‌افزاری به مدل نهایی به پرنمودن فاصله میان معمار و تحلیل‌گران نیازهای غیروظیفه‌مندی کمک می‌نماید، ضمن اینکه بدون تاخیر در فرآیند توسعه نرم‌افزار، ارزیابی نیاز کارایی را در سطح معماری نرم‌افزار انجام می‌دهد.

همانطور که بیان شد، هدف این مقاله، اجتماع فرآیند ارزیابی نیاز غیروظیفه‌مندی کارایی در سطح فرآیند توسعه معماری نرم‌افزار است. به همین منظور، ابتدا بایستی معماری نرم‌افزار را توصیف نموده و سپس طی الگوریتم پیشنهادی به یک مدل اجرایی تبدیل نمود. با مروری بر کارهای انجام‌شده، مشاهده می‌شود که محققان در محیط‌های علمی و مراکز صنعتی به دنبال زبان‌های توصیفی برای معماری هستند. از جملهٔ این زبان‌ها می‌توان از ARTek، Caps، MetaH، Rapide، UniCon و Wright نام برد. با بررسی این زبان‌های توصیفی، مشاهده می‌شود که همگی این زبان‌ها فقط بر روی خصوصیات اتصال مولفه‌ها، مانند نوع و سبک اتصال‌ها، تکیه دارند. از آنجایی‌که معماری نرم‌افزار به عنوان یک واسط بین تمامی صاحبان سهام می‌باشد، در سیستم‌های پیچیده نیاز است که علاوه بر خصوصیات ساختاری و عملیاتی مولفه‌ها، رفتار مولفه‌های اصلی در کل سیستم نیز مشخص گردد. برای توصیف معماری، می‌توان از زبان‌های توصیف معماری[۷]، MSC یا زبان مدلسازی یکپارچه نیز استفاده کرد. UML، یکی از بهترین انتخاب‌ها برای توصیف معماری است [۳و ۴]. به این دلیل که زبان‌های توصیف معماری تنها قابل استفاده در سطح توسعه معماری هستند، اما UML در تمام سطوح توسعه نرم‌افزار، قابل استفاده‌است. همچنین این زبان، توسط OMG، بعنوان یک زبان استاندارد برای توصیف معماری، معرفی شده‌است. در این توصیفات، مولفه‌های سیستم، ویژگی‌های قابل رؤیت آن‌ها و روابط بین مولفه‌ها به نمایش گذاشته می‌شوند. از طرفی، هیچ کدام از این توصیفات، مستقیما، قادر به ارزیابی صفات غیروظیفه مند نیستند و بایستی به یک مدل قابل اجرا، تبدیل شده و با کمک این مدل قابل اجرا، صفات غیروظیفه مند مورد ارزیابی قرار بگیرند.

مدل‌های قابل‌اجرای متفاوتی، جهت ارزیابی صفات غیر وظیفه‌مند، وجود دارد که مهمترین آن‌ها، شبکه‌های صف، شبکه‌های پتری، جبر فرآیندی، فرآیندهای تصادفی و غیره است. شبکه صف، جهت استفاده در تحلیل تصادفی مناسب بوده، اما برای تحلیل صفات کیفی[۸] مناسب نیستند.

تفاوت روش پیشنهادی، با فعالیت‌های انجام شده در این است که از نمودارهای مورد کاربری، ترتیب و مؤلفه، جهت ارزیابی کارایی معماری نرم‌افزار، استفاده‌شده و مدل قابل اجرای حاصل، مبتنی بر شبکه‌های پتری، است و برای ارزیابی کارایی توسط شبکه‌های پتری برخلاف شبکه‌های صف تنها به یک مدل اجرایی نیاز است و با تغییر مقدار و نوع مهره‌های موجود در این شبکه‌ها، اضافه‌نمودن عبارات به کمان‌ها برای محاسبه نیاز موردنظر و یا با اضافه‌نمودن مکان و یا گذار اضافی به شبکه می‌توان نیازهای غیروظیفه‌مندی دیگری را ارزیابی نمود. لذا با این شبکه‌ها یک چارچوب عمومی برای ارزیابی معماری نرم‌افزار از نظر نیازهای غیروظیفه‌مندی مختلف فراهم می‌شود.

در ادامه، ساختار مقاله بدین‌شرح است: در بخش ۲ نقش نمودارهای توصیف‌کننده معماری در ارزیابی کارایی و حاشیه‌نویسی‌های کارایی مربوط به آن‌ها توضیح داده می‌شود. در بخش ۳ توصیفات معماری به همراه پارامترهای کارایی مربوطه به مدل قابل اجرای مبتنی بر شبکه‌های پتری رنگی تصادفی تعمیم‌یافته تبدیل می‌شوند. در بخش ۴ به منظور اثبات صحت الگوریتم پیشنهادی یک مثال کاربردی ارائه می‌شود و نهایتا، در بخش ۵ نتیجه گیری بیان می‌گردد.

۲- نقش نمودارهای UML در ارزیابی کارایی

معماری نرم‌افزار با مشخص‌نمودن وجوه ساختاری و رفتاری، سیستم را در سطوح بالایی از انتزاع توصیف می‌کند. در این مقاله به منظور توصیف معماری نرم‌افزار از نمودارهای موردکاربری، ترتیب و مولفه موجود در UML 2.0، استفاده شده است. UML 2.0، کیفیت یادداشت‌های مورد استفاده در نمودارها و قدرت تعریف نمودارهای ترتیب و مشخصات نمودار مولفه را بهبود بخشیده‌است. در توصیف معماری نرم‌افزار، نمودار مولفه، ساختار ایستای سیستم، نمودار ترتیب رفتار

سیستم نرم‌افزاری و نمودار موردکاربری نیز سرویس‌های فراهم‌شده بوسیله سیستم را مدلسازی می‌کنند. به همین منظور در این بخش، این سه نمودار بهمراه نقش آن‌ها در ارزیابی کارایی و حاشیه‌نویسی‌های کارایی مورد نیاز آن‌ها بطور کامل توضیح داده می‌شود.

۲-۱- نقش نمودار موردکاربری در ارزیابی کارایی و حاشیه‌نویسی نمایه‌ها به آن

نمودار مورد کاربری، تعامل میان سیستم و محیط را توصیف‌می‌کند. این نمودار، شامل مجموعه‌ای از عامل‌ها[۹]، مواردکاربری و روابط میان آن‌ها است. روابط استفاده‌شده در این نمودار، روابط "انجمنی"[۱۰]، "تعمیم"[۱۱]، "توسعه‌یافته"[۱۲]، و "شامل‌شدن"[۱۳] است. بین عامل‌ها و مواردکاربری رابطه انجمنی، بین دو عامل، رابطه تعمیم و بین دو مورد کاربری رابطه توسعه‌یافته، تعمیم، و شامل‌شدن برقرار است. یک عامل نیز هر چیزی را که با سیستم در تعامل‌است و بخشی از سیستم نیست، نشان‌می‌دهد. یک مورد کاربری، وظیفه‌مندی فراهم‌شده بوسیله سیستم، زیرسیستم یا یک کلاس را نشان می‌دهد و با یک یا چند نمودار ترتیب توصیف می‌شود [۹].

زیرمجموعه‌ای از مواردکاربری برای فرآیند ارزیابی کارایی استفاده‌می‌شود، بگونه‌ایکه برای هر یک از آن‌ها بایستی یک مدل کارایی ایجاد شود [۲۰، ۲۱ و ۲۲].

در نمودار موردکاربری، به هر ارتباط بین کاربر و موردکاربری، یک احتمال، نسبت داده‌می‌شود که این، احتمال اجرای موردکاربری، توسط کاربر را نشان‌می‌دهد [۲۲]. فرض کنید، یک نمودار موردکاربری با m کاربر و n موردکاربری وجوددارد. همچنین $P_i \ (i=1,...,m)$، تناوب استفاده کاربر i ام از سیستم نرم‌افزاری، $P_{ij} \ (j=1,...,n)$، احتمال استفاده کاربر i ام از موردکاربری jام را نشان‌دهد، بگونه‌ایکه $\sum_{i=1}^{m} P_i = 1$ و $\sum_{j=1}^{n} P_{ij} = 1$ است. درنتیجه احتمال اجرای نمودار ترتیب مربوط به موردکاربری x، عبارت است از [۲۲]:

$$P(x) = \sum_{i=1}^{m} P_i \,.\, P_{ix} \qquad (۱)$$

از آنجایی که نمودار موردکاربری، یک یا چند، سناریو را نشان می‌دهد، بنابراین، این نمودار، بعنوان یک زمینه کارایی، محسوب می‌شود و با کلیشه <<PAcontext>>، نمایش داده‌می‌شود. هر موردکاربری مورد استفاده برای ارزیابی کارایی، یک مرحله را نشان می‌دهد و برای آن از کلیشه <<PAstep>>، استفاده می‌شود. همچنین، برای نمایش بار کاری سیستم از کلیشه <<PAopenload>> یا <<PAclosedload>>، استفاده می‌گردد [۲۲، ۲۳ و ۲۴].

۲-۲- نقش نمودار ترتیب در ارزیابی کارایی و حاشیه‌نویسی نمایه‌ها به آن

نمودار ترتیب، وظیفه‌مندی یک موردکاربری (نمودار مورد کاربری) را نشان می‌دهد; ضمن اینکه روی تعامل میان نمونه‌ها، تاکید می‌کند و با توجه به زمان ارسال پیام‌ها، رسم می‌شود [۲۰]. از نقطه نظر کارایی، برای مدل‌نمودن بار سیستم در نمودار ترتیب از عناصر و ساختارهای خاصی استفاده‌می‌شود. در اولین حالت، فرض می‌شود که زمان سپری‌شده، برای ارسال پیغام‌ها، چندان مهم نیست. البته اعمالی که در پاسخ به یک پیغام، انجام می‌شوند، نیاز به صرف زمان دارند که در نمودار حالت، مدلسازی‌می‌شوند. در حالت دوم، پیغام‌هایی که به ماشین‌های مختلف، ارسال‌می‌شوند، نیاز به صرف زمان دارند و باری را به سیستم، تحمیل‌می‌کنند که بایستی در این نمودار، مدلسازی‌شوند [۲۳]. به هر پیغام موجود در نمودار، می‌توان یک شرط، الصاق نمود، که این شرط، احتمال ارسال پیغام را بیان می‌کند. همچنین، چند پیغام که به آن‌ها شرط، الصاق شده‌است، می‌توانند از یک نقطه، ارسال و بطور موازی اجرا شوند. همچنین در صورتیکه، پیغام‌ها در یک ساختار تکرار، قرارگرفته‌باشند، می‌توانند، چندین مرتبه، ارسال شوند.

چون نمودار ترتیب برای بدست‌آوردن مدل کارایی، استفاده می‌شود، بعنوان یک زمینه کارایی درنظر گرفته می‌شود و کلیشه <<PAcontext>>، به آن حاشیه‌نویسی می‌شود. به هر پیغام در این نمودار، کلیشه <<PAstep>>، با برچسب‌های PAprob، PArep و size پیغام به همراه سرعت شبکه اضافه می‌شود تا به ترتیب، بیانگر احتمال اجرای پیغام‌ها، تکرار پیغام‌ها و زمان سپری‌شده برای ارسال یک پیغام از یک ماشین به ماشین دیگر باشند. البته برای حالت آخر، بجای برچسب size و سرعت شبکه از برچسب‌های PAdelay و PAdemand نیز استفاده نمود. به اولین پیغام موجود در نمودار ترتیب برای نمایش شدت بار کاری، کلیشه، <<PAclosedload>> یا <<PAopenload>>، به همراه برچسب‌های PApopulation و PAextdelay، اضافه می‌شود تا به ترتیب، تعداد کاربران و زمان مورد نیاز برای انجام یک درخواست را نشان دهند [۲۳ و ۲۴].

۲-۳- نقش نمودار مؤلفه در ارزیابی کارایی و حاشیه‌نویسی نمایه‌ها به آن

نمودار مؤلفه‌ها، شامل مجموعه‌ای از مؤلفه‌ها و ارتباطات میان آن‌ها است. ارتباطات بین مؤلفه‌ها، تعریف‌کننده مجموعه‌ای از عملیات است که بوسیله مؤلفه، انجام می‌شود. یک مؤلفه نیز یک واحد ماژولار است که دارای مجموعه‌ای از رابط‌های خوش‌تعریف است. این رابط‌ها به دو نوع رابط فراهم کننده یک عملیات[۱۴] و رابط درخواست کننده یک عملیات[۱۵] تقسیم می‌شوند [۶].

اطلاعات اضافه‌شده به نمودار مؤلفه‌ها، نوع مراکز سرویس (بعنوان مثال سرورهایی با صف انتظار یا تاخیر)، نرخ سرویس‌های فراهم‌شده توسط آن‌ها و سیاست زمان‌بندی آن‌ها برای استخراج کارها از صف انتظار را مشخص می‌نماید. به همین منظور برای مولفه‌ها از کلیشه <<PAhost>>، برای مدلسازی منابع فعال، استفاده می‌شود. با این فرض که هر مؤلفه، بر روی یک وسیله فعال منطقی، مستقر است. در این کلیشه، برای مشخص‌نمودن سیاست زمان‌بندی، از برچسب <<PAschedpolicy>>، استفاده می‌شود. برای مشخص نمودن زمان مورد نیاز برای اجرای یک مؤلفه می‌توان از کلیشه <<PAstep>>، بهمراه برچسب <<PAdelay>> یا <<PAdemand>>، استفاده نمود. استفاده از برچسب <<PAdemand>>، برای هر مولفه یعنی مرکز سرویس، یک مرکز انتظار است و در استفاده از برچسب <<PAdelay>> ، یعنی مرکز سرویس، یک مرکز تاخیری است [۲۴ و ۲۵].

۳- تبدیل توصیفات معماری به مدل قابل اجرا

همانطور که در مقدمه بیان‌شد، هدف این مقاله، استخراج یک مدل قابل اجرا از توصیفات معماری برای ارزیابی کارایی است. در این بخش، قوانینی برای تبدیل

نمودارهای توصیف‌کننده معماری به شبکه پتری معرفی می‌شود. البته در این تبدیلات، پارامترهای کارایی به نمودارها اضافه شده و الگوریتم‌هایی برای تبدیل این توصیفات به شبکه پتری ارائه می‌شود و از یکی از بسط‌های شبکه پتری یعنی شبکه پتری رنگی تصادفی تعمیم‌یافته استفاده‌می‌شود.

در یک تعریف کلی، شبکه‌های پتری، مدل‌هایی از جنبه‌های ساختار و رفتار یک سیستم رویداد گسسته را نشان می‌دهند. این شبکه‌ها، نشانگر الگوهای ارتباطی، الگوهای کنترل و جریان‌های اطلاعاتی و ارائه‌دهنده چارچوبی برای تحلیل، اعتبارسنجی و ارزیابی کارآیی، هستند. شبکه‌های پتری، گراف‌های چندگانهٔ جهتدار بوده و شامل دو قسمت مکان[۱۶] و گذار[۱۷] هستند. مکان با O، گذار با ▮ نمایش داده می‌شوند که با کمان به هم متصل می‌شوند. کمان‌ها، تنها می‌توانند مکان‌ها را به گذارها و یا گذارها را به مکان‌ها، متصل کنند. یک گذار، یک پردازشگر، رویداد، گام محاسباتی، الگوریتم، وظیفه و ... را نشان دهد. مکان‌ها نیز، بافر، پیش‌شرط، پس‌شرط، دادهٔ ورودی- خروجی، منابع مورد نیاز یا آزادشده، شرایط یا نتایج را نشان می‌دهند. شبکه پتری، با مهره[۱۸] نشانه‌گذاری می‌شود که با ● نمایش داده‌می‌شود و در مکان‌ها قرار می‌گیرد. یک علامتگذاری از یک شبکه پتری، نگاشتی است که یک عدد نامنفی صحیح (تعداد مهره‌ها) را به هر مکان در شبکه، انتساب می‌دهد و وضعیت شبکه پتری را توصیف می‌کند. در یک شبکه پتری، مهره‌ها، غیر قابل تمیز هستند و وجود یک یا چند مهره، در دسترس بودن منبع یا تکمیل شرط و عدم حضور مهره، برقراری یا عدم برقراری شرط و آزاد بودن منبع را نشان می‌دهد.

شبکه‌های پتری، برای آنالیز کیفی خواص منطقی سیستم‌ها، مورد استفاده قرار می‌گیرند. جهت استفاده از آن‌ها در آنالیز کمی، می‌توان به گذارهای این شبکه‌ها، زمان را نیز اضافه نمود. در اینصورت، این شبکه‌ها، شبکه‌های پتری تصادفی[۱۹]، نام می‌گیرند. SPNها، شبکه‌های پتری هستند که تاخیر آتش شدن گذارها، بوسیله متغیرهای تصادفی با توزیع نمایی، مشخص می‌شود. در این شبکه‌ها به هر گذار، یک تاخیر آتش شدن، بطور تصادفی، نسبت داده می‌شود که تابع چگالی احتمال آن یک نمایی منفی با نرخ λ_i است. در صورتیکه زمان به بعضی از گذارها در یک شبکه پتری تصادفی، نسبت داده‌شود، شبکه پتری تصادفی تعمیم یافته[۲۰]، نام می‌گیرد. در اینصورت گذارها در این شبکه به دو گروه گذارهای آنی[۲۱] و زمانی[۲۲]، تقسیم می‌شوند. گذارهای آنی با ▮ و گذارهای تاخیری با ▯ نشان داده می‌شوند. همچنین این شبکه یک تابع اولویت دارد که سطوح اولویت را به گذارها نسبت می‌دهد و در حالت پیش فرض گذارهای زمانی دارای اولویت صفر هستند. این نوع شبکه یک تابع وزن نیز دارد که نرخ را به گذارهای زمانی و وزن را به گذارهای آنی نسبت می‌دهد [۲۶].

شبکهٔ پتری رنگی، نسخهٔ توسعه‌یافته شبکه پتری است. علاوه‌بر مکان‌ها، گذارها و مهره‌ها در این شبکه‌ها، مفاهیم رنگ، محافظ و عبارات معرفی می‌شوند. مقادیر داده‌ای در این شبکه‌ها توسط مهره‌ها حمل‌می‌شوند. شبکه‌های پتری رنگی ارائه‌دهنده مدل‌های دقیقتری از سیستم‌های پردازشی ناهمگام پیچیده هستند. در این شبکه‌ها برخلاف شبکه‌های پتری، مهره‌ها از یکدیگر قابل‌تمیز هستند. زیرا هر یک از مهره‌ها دارای صفاتی به نام رنگ هستند و تنها می‌توانند مقادیر تعریف‌شده توسط همان مجموعه رنگ را اختیار کنند. در شبکه‌های پتری رنگی تصادفی تعمیم‌یافته به گذارهای این شبکه‌ها، زمان با توزیع نمایی منفی اضافه‌می‌گردد. در این نوع شبکه هم گذارهای آنی و هم زمانی وجوددارد. به همین علت در این تحقیق، بهترین مدل اجرایی برای ارزیابی کارایی و قابلیت اطمینان محسوب‌می‌شود.

در ادامه، نمودارهای UML استفاده‌شده برای توصیف معماری نرم‌افزار، طی الگوریتمی به یک مدل قابل اجرای مبتنی بر شبکه پتری رنگی تصادفی تعمیم‌یافته تبدیل می‌شوند. در هر یک از تبدیلات، ابتدا الگوریتم بطور کلی بیان شده و سپس هر یک از بخش‌های الگوریتم، بطور جزئی‌تر توصیف می‌گردد.

۳-۱- تبدیل نمودار موردکاربری به شبکه پتری

همانطور که بیان‌شد، نمودار موردکاربری، شامل مجموعه‌ای از عامل‌ها، موارد‌کاربری و ارتباطات میان آن‌ها است. یک موردکاربری، نیز شامل چندین مولفه بوده و ضمن توصیف کامل از یک تراکنش، عامل یا موردکاربری دیگری از آن استفاده می‌کند؛ همچنین توسط یک یا چند نمودار ترتیب توصیف می‌شود.

در این بخش الگوریتم پیشنهادی برای تبدیل نمودار مورد کاربری بهمراه حاشیه‌نویسی‌های اضافه‌شده به آن به شبکه پتری رنگی تصادفی تعمیم‌یافته ارائه می‌شود. در این مقاله، برای این تبدیل از ایده ارائه شده در [۱۶] با تفاوت‌هایی استفاده‌شده‌است. بدینصورت که برخلاف [۱۶] در این مقاله، کلیه ارتباطات بین عامل و مواردکاربری و همچنین ارتباطات مواردکاربری با یکدیگر یعنی ارتباطات توسعه‌یافته، تعمیم و شامل‌شدن به شبکه پتری تبدیل می‌شوند.

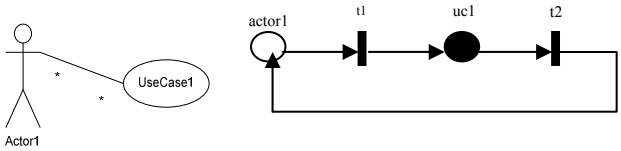
در تبدیل نمودار موردکاربری به شبکه پتری رنگی تصادفی، مهره‌ها و صفات انتسابی به آن‌ها نقش مهمی‌دارد. در این تبدیل، هر مورد کاربری که توسط عامل یا مورد کاربری دیگری فراخوانی می‌شود، بعد از اتمام اجرا به فراخوان خود بازمی‌گردد و صفت انتسابی به مهره‌های مکان‌ها، مسیر حرکت و نحوه فعال شدن گذارها را مشخص می‌کند و تبدیل انواع رابط‌های بین موارد کاربری ساده‌تر صورت می‌گیرد. ساختارهای مختلفی که نمودار موردکاربری می‌تواند داشته باشد بصورت زیر به شبکه پتری رنگی تصادفی تعمیم‌یافته تبدیل می‌شود [۲۷ و ۲۸]:

الف- تبدیل هر یک از موارد کاربری و عامل‌ها، بطور جداگانه به شبکه پتری: موارد‌کاربری، عملیاتی هستند که سیستم فراهم‌می‌کند. این نمودارها نشان‌می‌دهند که چه عامل‌هایی به موارد‌کاربری مقدار اولیه می‌دهند. همچنین آن‌ها نشان می‌دهند که چه موقع یک عامل، اطلاعات را از یک موردکاربری دریافت می‌کند. رابطه بین مواردکاربری و عامل‌ها، از نوع ارتباط[۲۳] است. در تبدیل مواردکاربری به شبکه پتری، هر عامل و موردکاربری به یک مکان، نگاشت می‌یابد؛ بین این دو مکان گذاری با یک محافظ قرار می‌گیرد که این محافظ[۲۴]، شرط مربوط به صدازدن موردکاربری توسط کاربر را نشان می‌دهد. از طرفی چون هر موردکاربری، باید به موردکاربری یا کاربر صدازننده آن برگردد، یک گذار دیگر برای برگشت در نظرگرفته می‌شود. در مرحله بعد، مکان مربوط به هر موردکاربری، با شبکه پتری حاصل از نمودار ترتیب آن جایگزین می‌شود. به همین دلیل این مکان‌ها با دایره توپر نمایش داده می‌شوند تا از مکان‌های دیگر قابل تفکیک باشند [۲۷ و ۲۸].

وضعیت‌های مختلفی که ارسال پیغام می‌تواند دنبال کند به شرح زیر است:

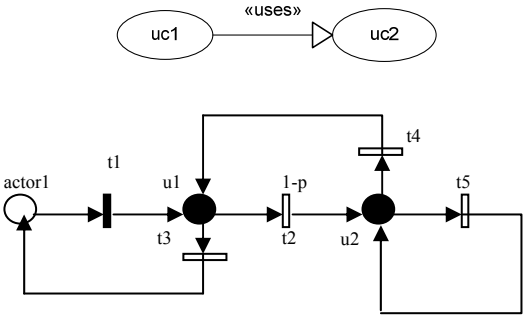
جایگزین‌شدن مهره در مکان مربوط به عامل بدین معنی است که یک عامل در حال تعامل با سیستم است، شلیک شدن گذار بعد از آن به معنی فراخوانی یک مورد کاربری توسط عامل است، جایگزین‌شدن یک مهره در مکان مربوط به موردکاربری به معنی آمادگی این موردکاربری برای انجام وظیفه خود و شلیک‌شدن گذار بعد از آن به معنی اتمام وظیفه موردکاربری و ارسال نتیجه و یا بازگشت به عامل فراخوان است.

شکل ۲ یک موردکاربری و تبدیل آن به شبکه پتری را نشان می‌دهد. در این شکل به گذار t1 زمان، نسبت داده‌می‌شود تا مدت زمان انتخاب و اجرای uc1 توسط کاربر را نشان دهد. زمان الصاق‌شده به گذار t2 مدت زمانی است که سیستم نیازدارد تا به وضعیتی برگردد تا اجازه شروع را به عامل بدهد.



شکل ۲- نمودار موردکاربری و شبکه پتری معادل آن

ب- تبدیل رابطه شامل‌شدن، بین دو موردکاربری به شبکه پتری: روابط موجود بین موارد‌کاربری، برای مدلسازی برخی عملیات موردنیاز بین دو یا چند موردکاربری، به کار می‌روند. رابطه شامل‌شدن به یک مورد کاربری، اجازه استفاده از عملیات مهیا شده توسط یک مورد کاربری دیگر را می‌دهد [۲۷ و ۲۸].



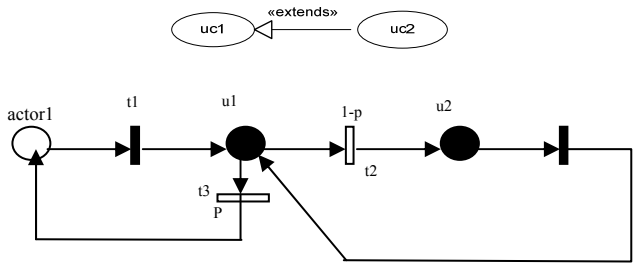
شکل ۳- رابطه شامل‌شدن بین دو مورد کاربری و تبدیل آن به شبکه پتری

شکل ۳ تبدیل این رابطه به شبکه پتری رنگی را نشان می‌دهد. در این تبدیل موردکاربری دوم اجبارا بایستی اجرا شود. در این رابطه چندین سناریو برای اجرای مواردکاربری uc1 و uc2 وجود دارد. در حالت اول ابتدا uc1 و سپس uc2 بطور کامل اجرا می‌شود. در حالت دوم، ابتدا، uc2 و سپس uc1، بطور کامل اجرا می‌شود و در حالت سوم، این مواردکاربری، به چند قسمت شکسته شده و طی مراحلی اجرا می‌شوند. مطابق شکل، شرط انتخاب هریک از موارد کاربری uc1 و uc2 و ترتیب و تعداد دفعات اجرای آن‌ها را محافظ گذارها، صفت انتخاب شده برای مهره‌ها در شبکه‌های پتری رنگی و یا زمان نسبت داده شده به مهره‌ها و گذارها مشخص می‌کنند. در این شکل، به هر مهره در مکان u1، صفت، نسبت داده می‌شود که حداقل یک مرتبه موردکاربری u2 را اجرا کند [۲۷ و ۲۸].

وضعیت‌های مختلفی که ارسال پیغام می‌تواند دنبال کند به شرح زیر است [۲۷ و ۲۸]:

جایگزین‌شدن مهره در مکان مربوط به عامل به معنی تعامل یک عامل با سیستم است، شلیک‌شدن گذار بعد از آن به معنی فراخوانی موردکاربری u1 توسط عامل است، جایگزین شدن یک مهره در مکان مربوط به u1 به معنی آمادگی آن برای انجام وظیفه خود و انتخاب اجرای موردکاربری u2 یا انتخاب ارسال نتیجه به عامل است. انتخاب این دو حالت به گذارهای مربوطه الصاق می‌شود. درصورتیکه گذار t2، شلیک شود به معنی فراخوانی u2 است و قرارگرفتن مهره در مکان u2 به معنی آمادگی آن برای انجام وظیفه خود و انتخاب برای برگشت به حالت قبل برای اجرای بخش دیگری از u1 و یا برگشت به عامل است. شرط این انتخاب به گذارهای مربوطه الصاق می‌شود. در صورتیکه گذار t4 شلیک شود بدین معنی است که بخشی از u1 بایستی اجرا شود و شلیک‌شدن گذار t3 به معنی اتمام وظیفه هر دو موردکاربری و ارسال نتیجه و یا بازگشت به عامل فراخوان است.

ج- تبدیل رابطه توسعه‌یافته، بین دو موردکاربری به شبکه پتری: یک رابطه توسعه‌یافته به یک موردکاربری اجازه می‌دهد که بطور دلخواه عملیات مهیاشده‌ء دیگر مواردکاربری را بسط دهد و بسیار شبیه رابطه شامل‌شدن عمل می‌کند. تبدیل این رابطه به شبکه پتری، در شکل ۴ نشان داده شده است. در این تبدیل، وقتی u1 در حال اجرا شدن است، اجرای u2 بصورت اختیاری است که شرط این انتخاب به گذارهای t_2 و t_3 نسبت داده می‌شود [۲۷ و ۲۸].

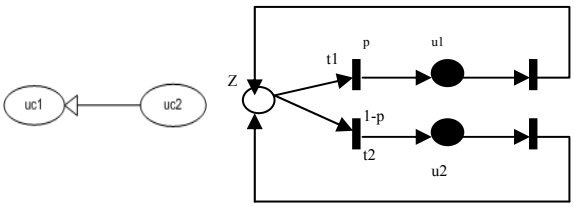


شکل ۴- رابطه توسعه‌یافته بین دو مورد کاربری و شبکه پتری معادل آن

وضعیت‌های مختلفی که ارسال پیغام می‌تواند دنبال کند به شرح زیر است:

جایگزین شدن مهره در مکان مربوط به عامل به معنی تعامل یک عامل با سیستم است، شلیک شدن گذار بعد از آن به معنی فراخوانی مورد کاربری u1 توسط عامل است، جایگزین‌شدن یک مهره در u1 به معنی آمادگی این موردکاربری برای انجام وظیفه خود و انتخاب اجرای مورد کاربری u2 یا انتخاب ارسال نتیجه و بازگشت به عامل است. انتخاب این دو حالت به مهره‌های مکان‌ها و گذارهای مربوطه الصاق می‌شود. در صورتیکه گذار t2 شلیک شود به معنی فراخوانی موردکاربری u2 است و قرارگرفتن مهره در مکان دوم به معنی آمادگی موردکاربری u2 برای انجام وظیفه خود است. در نهایت شلیک شدن گذار بعد از این مکان به معنی اتمام وظیفه موردکاربری و ارسال نتیجه به عامل فراخوان است.

د- تبدیل رابطه تعمیم، بین دو موردکاربری به شبکه پتری: مواردکاربری می‌توانند از یکدیگر ارث ببرند. بعنوان مثال مورد کاربری فرزند، رفتار و معنی را از موردکاربری پدر به ارث می‌برد و آن را به رفتارهای خود اضافه می‌کند. هر کجا که از پدر استفاده می‌شود، می‌توان از فرزندان آن استفاده نمود. تفاوت این رابطه با رابطه توسعه‌یافته این است که در رابطه تعمیم، بدون شکستن جریان اجرا، رابطه فرزند، جایگزین رابطه پدر می‌شود. شکل ۵ این تبدیل را نشان می‌دهد. در این شکل از یک مکان اضافی استفاده می‌شود و به گذار مربوطه و ارتباطات موجود، احتمال انتخاب موردکاربری پدر یا فرزند نسبت داده می‌شود [۲۷ و ۲۸].



شکل ۵- رابطه عام بین دو مورد کاربری و شبکه پتری معادل آن

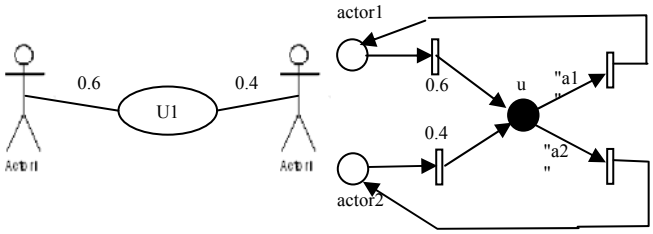
وضعیت‌های مختلفی که ارسال پیغام می‌تواند دنبال کند به شرح زیر است [۲۷ و ۲۸]:

جایگزین‌شدن مهره در مکان مربوط به عامل به معنی تعامل یک عامل با سیستم و انتخاب یکی از گذارهای پدر و فرزند (t1 و t2) است که شرط انتخاب به گذارهای مربوطه الصاق می‌شود. قرارگرفتن مهره بر روی یکی از مکان‌ها به معنی آمادگی موردکاربری برای انجام وظیفه خود است و شلیک شدن گذارهای بعد از آن به معنی اتمام وظیفه موردکاربری و ارسال نتیجه و یا بازگشت به عامل فراخوان است.

از طرفی، ممکن است یک موردکاربری، توسط چند کاربر، صدا زده‌شود یا یک کاربر، چند موردکاربری را صدا بزند. در اینصورت احتمال هر یک از این‌ها بایستی به گذارهای مربوطه و ارتباط بین آن‌ها اضافه شود.

تبدیل کلیشه PAstep به شبکه پتری: در این نمودار اگر از کلیشه <<PAstep>> با برچسب PAprob استفاده‌شود، مشابه رابطه توسعه‌یافته، شرط انتخاب به گذار مربوطه داده می‌شود [۲۹].

مطابق شکل ۶ درصورتیکه دو عامل قادر به انتخاب یک موردکاربری باشند و احتمال انتخاب آن‌ها نیز با برچسب PAprob مشخص شود، شرط انتخاب آن‌ها به گذارها داده می‌شود. همچنین نام عامل یکی از صفات انتسابی به مهره است تا بعد از اتمام اجرای u1 گذاری که باید شلیک شود را تعیین کند.



شکل ۶- نمودار مورد کاربری با احتمال انتخاب عوامل آن و شبکه پتری رنگی تصادفی تعمیم‌یافته معادل آن

تبدیل کلیشه PAopenload یا PAclosedload به شبکه پتری: درصورتیکه برچسب PAopenload یا PAclosedload به موردکاربری حاشیه‌نویسی شود، نوع زمان‌بندی ورود و خروج وظایف را تعیین می‌کند. لذا باید به مهره‌ها علاوه بر صفت، زمان نیز بگونه‌ای نسبت داده شود که نحوه ورود و خروج مهره‌ها در آن کنترل شود. درصورتیکه برچسب PAclosedload استفاده شود، تعداد کاربران برای آن مشخص است و این تعداد، با تعداد مهره‌های موجود در مکان مشخص می‌شود. درصورتیکه برچسب PAopenload استفاده شود، بایستی احتمال ورود مشتری به سیستم با کمک تابع توزیعی که در برچسب مشخص شده است، تعیین شود [۲۹].

۳-۲- تبدیل نمودار ترتیب به شبکه پتری

در طی فرآیند توسعه نرم‌افزار، چگونگی تعامل میان مولفه‌ها، جهت انجام وظایف، توسط یکی از نمودارهای ترتیب یا همکاری، توصیف می‌شود.

تفاوت این دو نمودار، در این است که، نمودار همکاری، بر روی شرکت‌کنندگان، نقش و روابط میان آن‌ها تاکید دارد، در حالیکه، نمودار ترتیب، اجرای موقت الگوهای ارتباطی را نشان می‌دهد. اگرچه در این مقاله، الگوریتم تبدیل نمودار ترتیب به شبکه پتری بیان گردیده ولی قابل استفاده برای نمودار همکاری نیز هست [۱۲ و ۱۹]. این نمودار، ترتیب پیغام‌های ارسالی بین مولفه‌ها را براساس زمان رخ دادن آن‌ها نشان می‌دهد.

در [۱۲] تبدیل بر روی پیغام‌ها و نحوه انتقال آن‌ها متمرکز است، درحالیکه در [۱۸] تبدیل بر روی شی فرستنده و دریافت‌کننده پیغام متمرکز است، که پیاده‌سازی این روش ساده‌تر و فهم آن آسان‌تر است. به همین دلیل برای تبدیل نمودار ترتیب به شبکه پتری در این مقاله از ایده ارائه شده در [۱۸] استفاده می‌شود. تفاوت آن با روش پیشنهادی این است که شبکه پتری موردنظر شبکه پتری رنگی تصادفی تعمیم‌یافته است و انواع پیغام‌ها (همگام، ناهمگام، آنی و زمانی) و انواع ساختارهای موجود در نمودار ترتیب به شبکه پتری تبدیل می‌شوند.

درصورتیکه ارتباطات از نوع ناهمگام[۲۵] باشد، عمل، تنها با ارسال پیغام، به اتمام می‌رسد، درحالیکه، در ارتباطات همگام[۲۶]، ارسال‌کننده، بعد از ارسال پیغام، منتظر پاسخ می‌ماند و در صورتی عمل، به اتمام می‌رسد که ارسال‌کننده پیغام، اجرایش را از سر بگیرد [۱۲].

با درنظرگرفتن فرضیات فوق، نمودار ترتیب، طی مراحل زیر به یک شبکه پتری، تبدیل می‌شود [۲۷ و ۲۸]:

- به ازای هر پیغام موجود در نمودار ترتیب، مولفه‌های فرستنده و دریافت‌کننده آن به یک زیرسیستم شبکه پتری تبدیل شوند.
- شبکه‌های پتری حاصل از مرحله قبل، مطابق با ترتیب و ارتباط بین پیغام‌ها، با هم ترکیب شوند.
- در نهایت برای شبکه پتری حاصل، نشانه‌گذاری اولیه تعریف شود.

در ادامه، دو مرحله اول بطور کامل‌تر، توضیح داده می‌شود.

۳-۲-۱- نمایش شبکه پتری مولفه‌های فرستنده و دریافت کننده پیغام در نمودار ترتیب و ترکیب آن‌ها با یکدیگر

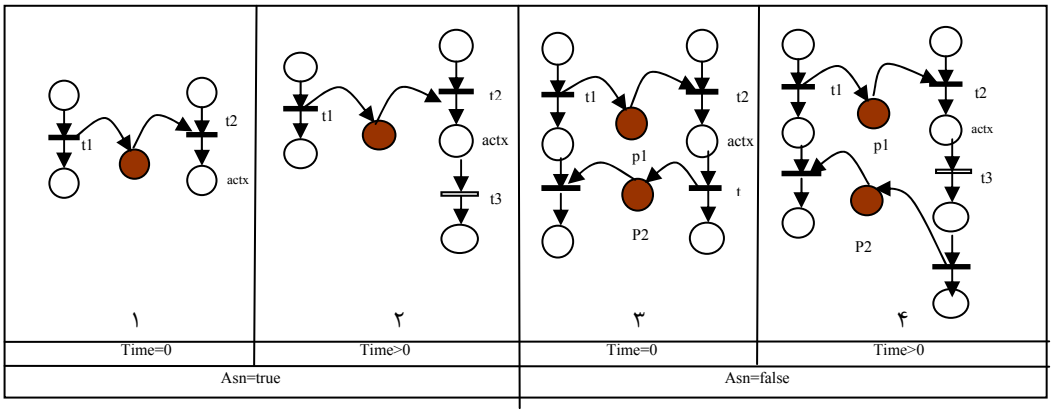
هدف این مرحله، تبدیل مولفه‌های فرستنده و دریافت‌کننده پیغام به شبکه پتری است. نمودار ترتیب در ارزیابی کارایی نشان می‌دهد که چگونه یک مشتری از یک نوع خاص در بین مراکز سرویس حرکت می‌کند. برای هر مجموعه‌ای از سناریوها که به یک مشتری نسبت داده‌شده می‌توان رنگی را برای مهره در شبکه پتری درنظرگرفت. در ادامه هر یک از مولفه‌های فرستنده و دریافت‌کننده پیغام و پیغام‌های بین آن‌ها به شبکه پتری تبدیل می‌شود.

الف- تبدیل پیغام‌های ناهمگام و آنی به شبکه پتری: در این حالت، هر یک از مولفه‌های دریافت‌کننده و ارسال‌کننده پیغام، به مکان_گذار_مکان، تبدیل می‌شوند. ارتباط بین این دو مولفه از طریق یک مکان اشتراکی، صورت می‌گیرد و این مکان برای مولفه ارسال‌کننده پیغام، نقش مکان خروجی و برای مولفه دریافت‌کننده پیغام، نقش مکان ورودی را دارد. این مکان مشترک را بعنوان یک صف انتظار نیز می‌توان در نظر گرفت [۲۷ و ۲۸].

شکل ۷-۱ این نوع پیغام و شبکه پتری معادل آن را نشان می‌دهد. در این شکل مکان اشتراکی بین دو مولفه C1 و C2 با دایره توپر مشخص شده‌است.

وضعیت‌های مختلفی که ارسال پیغام می‌تواند دنبال کند به شرح زیر است [۱۲]:

- از نظر مولفه ارسال‌کننده، جایگزین شدن مهره روی اولین مکان به معنی آغاز ارسال پیغام است، شلیک‌نمودن گذار به معنی ارسال آن و جایگزین شدن مهره، روی مکان دوم به معنی انتهای ارسال پیغام است.
- از نظر مولفه دریافت‌کننده، یک مهره روی اولین مکان به معنی انتظار برای قرارگرفتن یک مهره بر روی مکان مشترک است، شلیک نمودن گذار به معنی دریافت پیغام و جایگزین‌شدن مهره بر روی مکان دوم به معنی انتهای دریافت پیغام و اتمام اجرای عمل است.
- از نظر پیغام، وقتی گذار مربوط به فرستنده، شلیک می‌شود، به مکان مشترک، دسترسی دارد و وقتی گذار مربوط به دریافت‌کننده، شلیک می‌شود، این پیغام خاتمه می‌یابد.
- از نظر سیستم نیز وقتی مولفه فرستنده، یک مهره را در مکان مشترک، قرار می‌دهد، یعنی به عمل actx، نیاز دارد. وقتی مولفه دریافت‌کننده در وضعیتی باشد که اجازه استفاده از مهره موجود در مکان مشترک را داشته باشد، فراخوانی را تائید می‌کند.



شکل ۷- تبدیل پیغام m به شبکه پتری تصادفی تعمیم یافته

ب- تبدیل پیغام ناهمگام و زمانی به شبکه پتری: برای این نوع پیغام از برچسب PAdemand و کلیشه PAstep استفاده می‌شود.

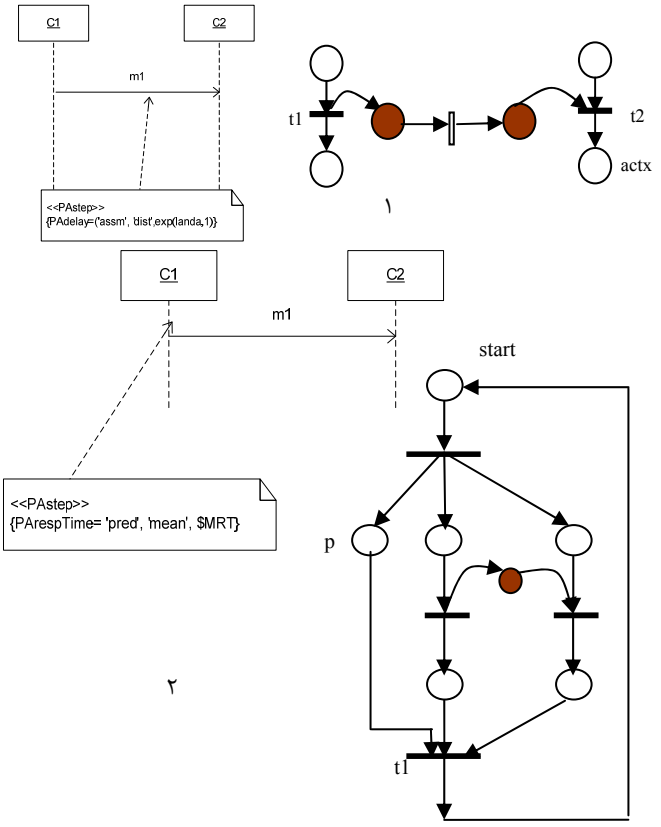
این کلیشه و برچسب‌های مربوط به آن نیز اطلاعاتی را در رابطه با احتمال اجرای هر پیغام و مدت زمان اجرای آن نشان می‌دهد. این کلیشه نشان می‌دهد که مولفه دریافت‌کننده برای اجرای عمل actx، توسط پردازنده به زمان نیاز دارد و این برچسب مدت زمان پردازش را نشان می‌دهد.

در تبدیل این پیغام به شبکه پتری رنگی تصادفی، مشابه پیغام ناهمگام و آنی، مولفه ارسال‌کننده پیغام، به مکان_گذار_مکان، تبدیل می‌شود. اما مولفه دریافت‌کننده پیغام، به مکان_گذار_مکان_گذار_مکان، تبدیل می‌شود که گذار دوم آن، یک گذار زمانی است و به آن نرخ شلیک شدن، نسبت داده می‌شود. شکل ۷-۲ این نوع پیغام و شبکه پتری معادل آن را نشان می‌دهد [۲۷ و ۲۹].

ج- تبدیل پیغام همگام و آنی به شبکه پتری: در اینصورت هر دو مولفه ارسال‌کننده و دریافت‌کننده پیغام، به مکان_گذار_مکان_گذار_مکان، تبدیل می‌شوند. مکان میانی در مولفه دریافت‌کننده پیغام، مکان انتظار و در مولفه دریافت‌کننده، فراهم‌کننده عمل actx است. همچنین ارتباط بین آن‌ها با دو مکان مشترک، صورت می‌گیرد که مکان مشترک p1 در شکل ۷-۳ عمل و داده مورد نظر را از مولفه فرستنده به مولفه دریافت‌کننده منتقل می‌کند و مکان مشترک p2 نتایج عملیات را بر می‌گرداند. مشابه قبل مکان p1 شبیه صف عمل می‌کند [۲۷ و ۲۸].

وضعیت‌های مختلفی که ارسال پیغام در این‌حالت می‌تواند دنبال کند به شرح زیر است:

- از نظر مولفه ارسال‌کننده، جایگزین شدن مهره روی اولین مکان به معنی آغاز ارسال پیغام است، شلیک‌نمودن گذار اول به معنی ارسال آن و جایگزین شدن مهره، روی مکان دوم به معنی انتظار برای قرار گرفتن یک مهره بر روی مکان مشترک دوم است. شلیک نمودن گذار دوم به معنی دریافت نتیجه عمل actx و جایگزین شدن مهره بر روی مکان سوم به معنی انتهای ارسال پیغام است.
- از نظر مولفه دریافت‌کننده، یک مهره روی اولین مکان به معنی انتظار برای قرارگرفتن یک مهره بر روی مکان مشترک اول است، شلیک نمودن گذار اول به معنی دریافت پیغام و جایگزین شدن مهره بر روی مکان دوم به معنی اتمام اجرای عمل actx است. شلیک شدن گذار دوم به معنی ارسال نتیجه پیغام و جایگزین شدن مهره بر روی مکان سوم به معنی انتهای دریافت پیغام است.
- از نظر پیغام، وقتی گذار اول مربوط به فرستنده، شلیک می‌شود، به مکان مشترک، دسترسی دارد و وقتی گذار دوم آن شلیک می‌شود به معنی دریافت نتیجه و انتهای اجرا است. وقتی گذار اول مولفه دریافت‌کننده، شلیک می‌شود، به معنی خاتمه اجرای عمل actx و وقتی گذار دوم شلیک می‌شود به معنی انتهای اجرای پیغام است.
- از نظر سیستم نیز وقتی مولفه فرستنده، یک مهره را در مکان مشترک اول، قرار می‌دهد، یعنی به عمل actx نیاز دارد و اگر مولفه دریافت‌کننده در وضعیتی باشد که اجازه استفاده از مهره موجود در مکان مشترک را داشته باشد، فراخوانی را تائید می‌کند. وقتی مولفه دریافت‌کننده، یک مهره را در مکان مشترک دوم قرار می‌دهد، یعنی اجرای عمل actx به اتمام رسیده است و مولفه فرستنده اگر در موقعیت استفاده از مهره موجود در مکان مشترک دوم باشد، از آن استفاده می‌کند.



شکل ۸- نمودار ترتیب ۱- با برچسب PAdelay و شبکه پتری معادل آن ۲- با برچسب PAresptime و شبکه پتری معادل آن

د- تبدیل پیغام همگام و زمانی به شبکه پتری: در این نوع پیغام برای نمایش زمان موردنیاز برای پردازش عمل actx با برچسب PAdemand حاشیه‌نویسی می‌شود. در تبدیل این نوع پیغام به شبکه پتری رنگی تصادفی، مولفه ارسال‌کننده پیغام، مثل پیغام همگام و آنی به مکان_گذار_مکان_گذار_مکان، تبدیل می‌شود. مولفه دریافت‌کننده نیز به مکان_گذار_مکان_گذار_مکان_گذار_مکان_گذار_مکان، تبدیل می‌شود که گذار مرکزی، از نوع زمانی است. شکل ۷-۴ این نوع پیغام و شبکه پتری معادل آن را نشان می‌دهد.

تبدیل برچسب PAdelay به شبکه پتری : در صورتیکه پیغام با برچسب PAdelay حاشیه‌نویسی شود یعنی پیغام m با تاخیر به مولفه دریافت کننده پیغام می‌رسد و مطابق شکل ۸-۱ به شبکه پتری تبدیل می‌شود. در این شکل گذار زمانی، تاخیر انتقال پیغام را به مولفه دریافت‌کننده پیغام نشان می‌دهد [۲۹].

تبدیل برچسب PAresptime به شبکه پتری: اگر به اولین مرحله در نمودار ترتیب، برچسب PAresptime از کلیشه PAstep حاشیه‌نویسی شود، معیارهای کارایی را با میانگین مقادیر بیان می‌کند در واقع این برچسب، زمان کل موردنیاز برای اجرا یا زمان پاسخ یک سناریو را نشان می‌دهد و تاخیر کل برای اجرای یک مرحله که شامل کلیه موارد مثل انتظار منابع و تمام زمان‌های اجرا است را دربرمی‌گیرد [۲۹ و ۱۹].

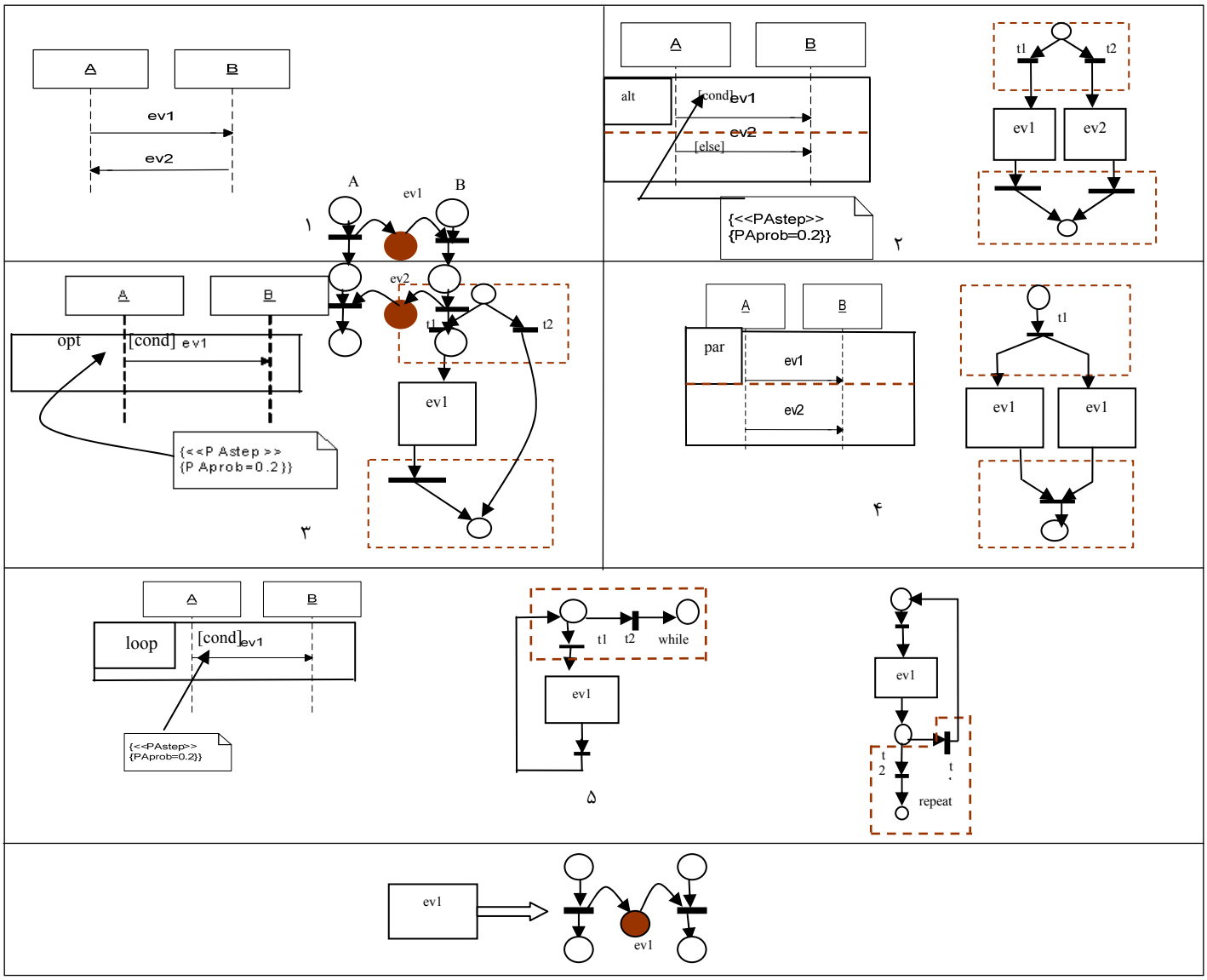
شکل ۸-۲ شبکه پتری معادل پیغام با برچسب PAresptime را نشان می‌دهد. در تبدیل این برچسب به شبکه پتری، مکان خنثی p و گذار t به شبکه پتری حاصل از نمودار ترتیب اضافه می‌شود. نقش مکان در اینجا نگهدارنده تعداد دفعات اجرای سناریو و نقش گذار، انتقال به وضعیت شروع در شبکه پتری است [۱۹].

تبدیل برچسب PAprob به شبکه پتری: در صورتیکه برچسب PAprob به پیغام‌ها حاشیه‌نویسی شود، یعنی پیغام‌ها به ترتیب حالت انتخاب و ساختار تکرار دارند که در ساختارهای مختلف نمودار ترتیب توضیح داده می‌شود [۲۹].

در ادامه انواع ساختارهای موجود در نمودار ترتیب یعنی ساختار ترتیب، انتخاب، توازی و تکرار، بررسی و نگاشت آن‌ها نیز به شبکه پتری بیان می‌شود.

ساختار ترتیب: در این ساختار، پیغام‌ها با توجه به ترتیب زمانی ارسال پیغام بین مولفه‌ها، به شبکه پتری تبدیل می‌شوند. بعنوان مثال شکل ۹-۱ ساختار ترتیب و شبکه پتری معادل آن را نشان می‌دهد. مطابق این شکل، ابتدا پیغام ev1 و سپس ev2 اجرا می‌شود [۲۷ و ۲۸].

ساختار انتخاب: این ساختار در نمودار ترتیب با دو عملگر alt و opt بیان می‌شود. عملگر alt، انتخاب یک عملوند را از بین مجموعه‌ای از عملوندها نشان می‌دهد. عمل نیز بایستی یک عبارت محافظ بصورت ضمنی یا صریح داشته باشد تا در این نقطه از تعامل، برای انتخاب به مقدار "درست" ارزیابی‌شود.



شکل ۹- تبدیل ساختارهای مختلف نمودار ترتیب به شبکه پتری

شکل ۹-۲ نیز ساختار انتخاب و شبکه پتری معادل آن را نشان می‌دهد. در این شکل بین دو پیغام ev1 و ev2 انتخاب، صورت می‌گیرد. جهت انتخاب یکی از این دو پیغام به شبکه پتری حاصل، بخشی اضافه شده که در کادر مشخص شده و ساختار انتخاب را نمایش می‌دهد. همانطور که این شکل نشان می‌دهد، احتمال انتخاب هر یک از پیغام‌های ev1 و ev2 به گذارهای t_1 و t_2، الصاق شده است. احتمال نسبت داده شده به این دو گذار، در قالب وزن، از کلیشه PAstep و برچسب PAprob بدست می‌آید [۲۷ و ۲۸].

عملگر opt، از نظر معنایی مشابه عملگر alt است و وقتی در نمودار ترتیب استفاده می‌شود که یک عمل، انتخاب یا رد می‌شود. شکل ۹-۳ ساختار انتخاب با استفاده از عملگر opt و شبکه پتری معادل آن را نشان می‌دهد.

مطابق این شکل، پیغام ev1 یا اجرا می‌شود و یا درصورت عدم برقراری شرط بدون اجرا شدن آن، پیغام‌های بعد از آن اجرا می‌گردند. احتمال انتخاب پیغام ev1 به گذار مربوط به آن الصاق می‌شود [۲۷ و ۲۸].

ساختار توازی: این ساختار در نمودار ترتیب با عملگر par بیان می‌شود و رفتارهای موازی اعمال را نشان می‌دهد. در اینحالت با یک رخداد، اعمال مختلفی به موازات هم انجام می‌شوند. شکل ۹-۴ ساختار توازی و شبکه پتری معادل آن را نشان می‌دهد. در این شکل دو پیغام ev1 و ev2 به موازات هم اجرا می‌شوند. مثل حالت قبل، به شبکه پتری، بخشی اضافه شده که در کادر قرار گرفته و اجرای موازی دو پیغام را نشان می‌دهد [۲۷ و ۲۸].

ساختار تکرار: این ساختار در نمودار ترتیب با استفاده از عملگر loop بیان می‌شود و تکرار یک یا چند پیغام و عمل را در این نمودار نشان می‌دهد. دفعات تکرار یا شرط تکرار حلقه توسط محافظ‌های عملگر مشخص می‌شود و می‌تواند شامل حد پائین و بالای دفعات تکرار باشد. شکل ۹-۵ ساختار حلقه و شبکه پتری معادل آن را در حالت تکرار بصورت while و repeat_until نشان می‌دهد. در این شکل پیغام ev1 در ساختار حلقه، قرار گرفته است. این ساختار، مشابه ساختار انتخاب، برای شرط خروج از حلقه، احتمالی را در نظر می‌گیرد که احتمال مربوطه به گذارهای t_1 و t_2 الصاق می‌شود [۲۷ و ۲۸].

۳-۳- سازگاری بین شبکه‌های پتری حاصل از دو نمودار ترتیب و مورد کاربری

وقتی که چندین نمودار زبان مدلسازی یکپارچه برای بدست‌آوردن مدل‌های رسمی استفاده می‌شوند، لازم است که سازگاری اطلاعات این نمودارها بررسی شود. در این تحقیق این سازگاری رعایت شده است، بدین صورت که تعامل بین مولفه‌ها با نمودار ترتیب و وظیفه‌مندی سیستم با نمودار مورد کاربری توصیف می‌شود. همچنین مولفه‌های کل سیستم، ارتباط بین آن‌ها و وظایف قابل مشاهده آن‌ها با نمودار مولفه توصیف می‌شود. در این بخش، شبکه‌های پتری حاصل از دو نمودار موردکاربری و ترتیب، با هم ترکیب می‌شوند تا یک شبکه پتری کامل از آن‌ها حاصل‌شود و جهت ارزیابی، استفاده شود. در [۱۷] این دو نمودار بصورت خیلی ساده به شبکه پتری رنگی تبدیل شده‌اند و سپس با هم ترکیب شده‌اند.

همانطور که بیان شد، هر مورد کاربری، با یک سناریو، توصیف می‌شود و در شبکه پتری بصورت یک مکان نمایش داده شد. بنابراین در ترکیب این نمودارها، کافی است، به جای مکان نشان دهنده موردکاربری، شبکه پتری معادل آن که از نمودار ترتیب آن، حاصل‌می‌شود، جایگزین شود. در حالت محدود، اگر برای هر مورد کاربری یک یا چند نمودار ترتیب رسم گردد، احتمال یا شرط انتخاب هر یک از نمودارهای ترتیب باید به کمان‌ها و گذارهای مربوطه در شبکه پتری اضافه شود [۲۷].

بعنوان مثال شکل ۱۰-۱ یک نمودار مورد کاربری، شکل ۱۰-۲ نمودار ترتیب مورد کاربری u1 و شکل ۱۰-۳ ترکیب شبکه پتری حاصل از این دو نمودار را نشان می‌دهد.

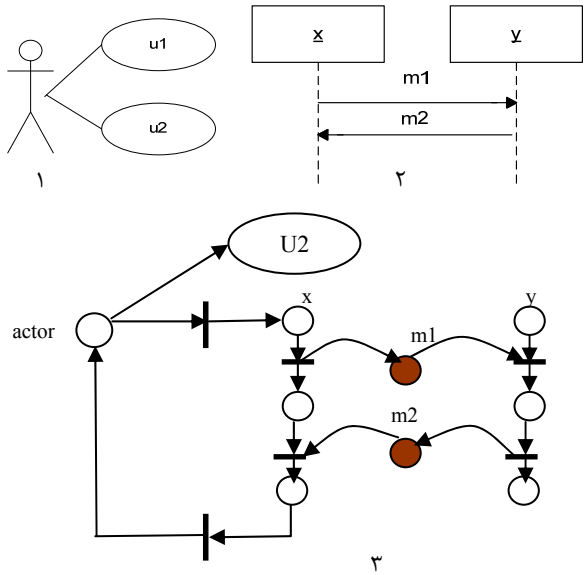
۳-۴-تبدیل نمودار مؤلفه به شبکه پتری

همانطور که بیان‌شد، نمودار مؤلفه‌ها، شامل مجموعه‌ای از مؤلفه‌ها و ارتباطات میان آن‌ها است. هر مؤلفه، وظایفی به عهده دارد که این وظایف بصورت عملیاتی [۲۷] برای آن مشخص می‌شوند. بعضی از این عملیات به تنهایی توسط مؤلفه قابل انجام هستند، اما بعضی دیگر به عملیات مؤلفه‌های دیگر نیاز داشته و یا نیازهای مؤلفه‌های دیگر را برآورده می‌نمایند .این نوع عملیات توسط ارتباطات تعریف شده بین مؤلفه‌ها مشخص می‌شود.

در [۳۰] نقش‌ها به همراه عبارات مسیر و نقش‌های همکار آن‌ها به شبکه پتری تبدیل‌شده‌است. در این تحقیق نیز از آن ایده به منظور تبدیل مولفه‌ها و نمودار مولفه به شبکه پتری استفاده شده است. بنابراین نمودار مؤلفه طی مراحل زیر به شبکه پتری تبدیل می‌گردد [۲۷ و ۳۱]:

- هر یک از مولفه‌های نمودار مولفه به منظور نگاشت به شبکه‌های پتری پالایش شوند.
- هر یک از مؤلفه‌ها بطور جداگانه به شبکه پتری، تبدیل شود.
- شبکه‌های پتری حاصل از مرحله قبل، مطابق با نوع ارتباط بین مؤلفه‌ها با هم ترکیب شوند.
- در نهایت برای شبکه پتری حاصل، نشانه‌گذاری اولیه تعریف شود.

در ادامه سه مرحله اول بطور کامل توضیح داده‌می‌شود.



شکل ۱۰- (۱) نمودار کاربری نمونه، (۲) نمودار ترتیب مورد کاربری u1، (۳) ترکیب این دو نمودار در شبکه پتری معادل

۳-۴-۱- پالایش مولفه‌های نمودار مولفه

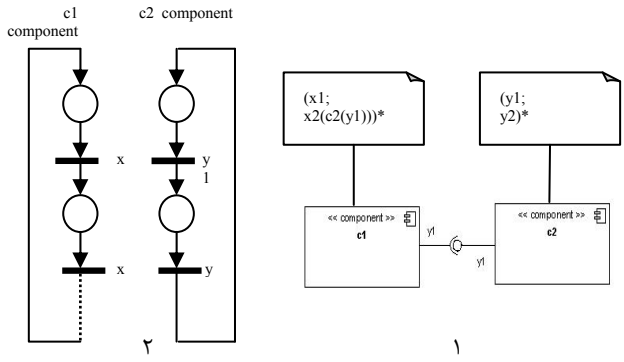
در این بخش ابتدا مولفه‌ها، پالایش می‌شوند تا بتوان الگوی رفتاری مولفه‌های درگیر و همکاران آن را بصورت شبکه‌های پتری نشان داد. در این پالایش، ترتیب عملیات مربوط به یک مولفه باید مشخص شود. یعنی نقطه آغاز برای تسهیل در مرحله پالایش، مشخص نمودن ترتیب مسئولیت‌های هر مولفه است. از آنجایی که

توصیف رفتار مولفه با نمایشات غیررسمی مشکل است، در این تحقیق از عبارات مسیر [۳۲]، استفاده می‌شود تا محدودیت‌هایی را بر روی ترتیب رفتار مولفه‌ها اعمال کند. عبارات مسیر، معنای دقیق ترتیب میان عملیات یک مولفه را نشان می‌دهند.

این ترتیب، رفتار ممکن یک مولفه را توصیف می‌کند و آنچه را که باید به شکل عملیات پایه مثل ترتیب، انتخاب و تکرار عملیات انجام شود، بطور واضح بیان می‌کند. گرامر رسمی عبارات مسیر از عبارات باقاعده[۲۸] استخراج می‌شود تا عملیات‌هایی مثل ترتیب، انتخاب، تکرار و توازی را نشان‌دهد. مولفه‌های همکار نیز با اضافه نمودن عبارات مسیر به مسئولیت‌های یک مولفه، محدود می‌شوند. یعنی مولفه‌های همکار به اعمال محدودیت‌های جداگانه بعنوان همکار نیاز ندارند؛ زیرا فعالیت مولفه‌های همکار به فعالیت مسئولیت‌های آن‌ها وابسته است. لذا ترتیب اجرای مسئولیت‌ها بایستی با استفاده از ساختارهای پایه مثل ترتیب، انتخاب، تکرار و همزمانی، مشخص‌شود.

در این تحقیق فرض می‌شود که کلیه عملیات انجام شده بوسیله هر مؤلفه، ترتیب فراخوانی آن‌ها و دفعات اجرا و انتخاب این عملیات مشخص است. همچنین عملیاتی که جهت انجام وظیفه خود، عملیاتی در مؤلفه دیگر را فراخوانی می‌کنند یا توسط مؤلفه دیگر فراخوانی‌می‌شوند، معلوم‌است. در هر مؤلفه برای مشخص‌نمودن این ویژگی‌ها از عملگرهای تعریف شده در عبارات مسیر، استفاده می‌شود.

به کمک این عملگرها می‌توان انتخاب یک عملیات از بین مجموعه‌ای از عملیات، تکرار صفر، یک یا چند مرتبه عملیات و ترتیب اجرای آن‌ها را مشخص نمود. در عبارات مسیر معرفی شده در این تحقیق، فرض می‌شود که عملگر "؛" ترتیب اجرای عملیات، عملگر "،" عمل انتخاب یک عملیات از بین مجموعه‌ای از عملیات، عملگر "+"، تکرار یک یا چند مرتبه عملیات و عملگر "*"، تکرار صفر یا چند مرتبه آن‌ها را نشان‌می‌دهد. با استفاده از عملگر "()" نیز عملگر موردنظر، برای کل عملیات داخل پرانتز اعمال می‌شود. در واقع با اضافه نمودن عبارات مسیر، رفتار دقیق یک مولفه مشخص می‌شود [۲۷ و ۳۱]. شکل ۱۱-۱ یک نمودار مولفه پالایش شده با عبارات مسیر را نشان می‌دهد. مطابق این شکل مولفه C1، دو عملیات دارد که ابتدا x1، یک مرتبه و سپس x2 چند مرتبه اجرا می‌شوند [۲۷ و ۳۱].



شکل ۱۱- (۱) نمودار مؤلفه، (۲) شبکه پتری معادل هر یک از مؤلفه‌ها

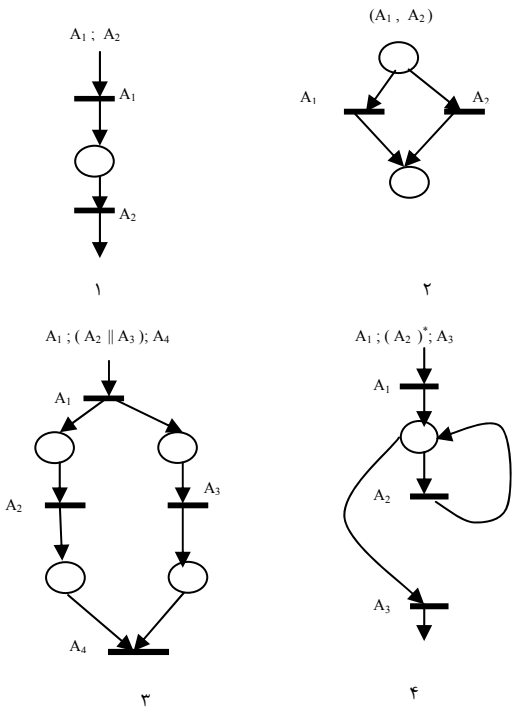
بعد از اضافه‌نمودن عبارات مسیر به عملیات داخل مولفه، مسئولیت‌های هر مولفه بطور کامل مشخص می‌شود. در مرحله بعد بایستی، مولفه‌ها و عملیات همکار آن‌ها مشخص شود. هر مولفه برای انجام بعضی از عملیات خود ممکن است به عملیات مولفه‌های دیگر نیازمند باشد؛ لذا برای نشان‌دادن مولفه‌های همکار، جلوی هر عملیاتی که به عملیات دیگر نیازمند است، نام عملیات و مولفه همکار نوشته می‌شود [۲۷ و ۳۱].

Laur و Campbell در [۳۲]، نگاشت عملیات پایه را در عبارت مسیر به معادلشان در شبکه پتری نشان داده‌اند. همچنین آن‌ها نشان دادند که "یک شبکه پتری، یک عبارت مسیر را شبیه‌سازی می‌کند اگر و تنها اگر مجموعه اعمالی[۲۹] که آن تولید می‌کند، دقیقا همان اعمالی باشد که از عبارات مسیر حاصل می‌شود." در اینجا منظور از رشته اعمال، همان وظایف یا ویژگی‌های قابل رویت مولفه است که با عبارات مسیر، پالایش شده‌اند. الگوریتم و تبدیل عبارات مسیر به شبکه‌های پتری در [۳۲] بیان شده است.

۳-۴-۲- تبدیل هر یک از مؤلفه‌ها به شبکه پتری

در تبدیل هر مولفه به شبکه پتری، فرض می‌شود که گذارها، فعالیت یک مسئولیت داخل مولفه و مکان‌ها پیش‌شرط شلیک‌شدن یک گذار را نشان می‌دهند. بنابراین برای تبدیل یک مؤلفه به شبکه پتری، بازای هر عملیات در مؤلفه، یک مکان-گذار، رسم می‌گردد؛ که این گذار، عملیات مربوط به مولفه را نشان می‌دهد. سپس براساس ترتیب صدازدن عملیات در مؤلفه، به هم مرتبط می‌شوند. البته در صورتیکه تعامل بین مولفه‌ها همگام باشد، عملیاتی که با مؤلفه دیگر در ارتباط است (به عملیاتی در مؤلفه دیگر نیاز دارد)، با نقطه‌چین به عملیات بعدی وصل می‌شود تا در مرحله بعدی این ارتباط، تکمیل‌گردد. کمان‌های رسم‌شده بین مکان‌ها و گذارها در یک مولفه، انتقال مولفه از یک وضعیت به وضعیت دیگر را نشان می‌دهد [۲۷ و ۳۱]. بعنوان مثال، شکل ۱۱-۲ شبکه پتری معادل هر یک از مؤلفه‌های شکل ۱۱-۱ را جداگانه، نمایش می‌دهد. در این شکل مولفه c1، دارای دو عملیات است که به ترتیب، ابتدا x1 و سپس x2 فراخوانی می‌شوند. عملیات x2 به عملیات y1 از مولفه c2 نیاز دارد، به همین منظور، گذار مربوط به آن باید به مؤلفه c2 متصل شود. در این مرحله، شبکه پتری بخشی از نمودار مولفه‌ها یعنی مسئولیت‌ها یا ویژگی‌های قابل رویت آن‌ها حاصل‌گردید.

با توجه به انواع عبارت مسیر که بین عملیات، وجود دارد، شبکه پتری حاصل برای هر یک در ادامه توضیح داده می‌شود [۲۷ و ۳۱]:



شکل ۱۲- شبکه پتری عبارات مسیر مختلف برای عملیات یک مؤلفه

ساختار ترتیب: در صورتیکه بین دو عملیات یک مولفه از عملگر ";" استفاده شود، یعنی این عملیات بایستی به ترتیب اجرا شوند. بنابراین شبکه پتری معادل آن‌ها نیز باید اجرای یک به یک و به ترتیب آن‌ها را نشان دهد. شکل ۱۲-۱ اجرای ترتیبی دو عملیات A_1 و A_2 را نشان می‌دهد. در این شکل، ابتدا عملیات A1 و سپس A2 انجام می‌شود. در واقع یک مسیر ترتیبی، با دو مسئولیت، دقیقا رفتاری مشابه با شبکه پتری با دو گذار دارد که با یک مکان بهم متصل شده‌اند.

ساختار انتخاب: در صورتیکه بین دو عملیات یک مولفه از عملگر "،" استفاده‌شود، یعنی یکی از این دو عملیات بایستی انتخاب و اجرا گردد. شکل ۱۲-۲ انتخاب یکی از عملیات A_1 و A_2 را در شبکه پتری، نشان می‌دهد. مطابق این شکل، یکی از عملیات A_1 و A_2 بایستی انتخاب و اجرا شوند. در شبکه پتری معادل آن، شرط انتخاب عملیات به کمان‌ها و گذارهای مربوطه اضافه می‌شود.

ساختار موازی: شکل ۱۲-۳ اجرای موازی دو عملیات و شبکه پتری حاصل از آن را نشان می‌دهد. در این شبکه، ابتدا A_1 اجرا می‌شود و با شلیک‌شدن گذار آن، هر دو عملیات A_2 و A_3 به موازات هم اجرا می‌شوند. بعد از خاتمه اجرای A_2 و A_3، گذار مربوط به عملیات A_4 اجرا می‌شود.

ساختار تکرار: در این ساختار، یک یا چند عملیات به دفعات و یا تحت شرایطی تکرار می‌شوند. شکل ۱۲-۴ ساختار تکرار (while) عملیات A_2 و شبکه پتری معادل آن را نشان‌می‌دهد. در این شکل، بعد از شلیک‌شدن گذار مربوط به عملیات A_1 گذار مربوط به عملیات A_2 تا زمان برقراری شرط، شلیک می‌شود و درصورتیکه شرط برقرار نباشد، گذار مربوط به عملیات A_3 شلیک می‌شود. شرط مربوط به انتخاب عملیات A_2 و A_3 روی کمان ورودی و گذار مربوط به آن‌ها مشخص می‌شود.

۳-۴-۳- ترکیب شبکه پتری حاصل از هر یک از مولفه‌ها در نمودار مولفه

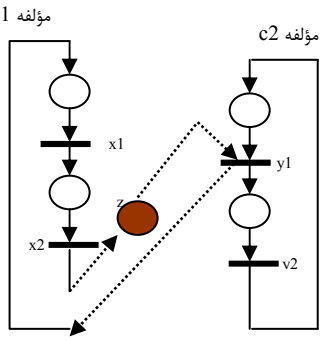
در این مرحله، ارتباطات بین مولفه‌ها به شبکه پتری تبدیل می‌شوند. در واقع شبکه پتری حاصل از هر یک از مولفه‌ها براساس نوع ارتباط میان آن‌ها، با هم ترکیب می‌شوند. همانطور که بیان‌شد، دو نوع ارتباط، بین مؤلفه‌ها، وجود دارد. یکی از نوع ارتباط "نیازدارد" و دیگری از نوع ارتباط "فراهم می‌کند" است، که معنای این تعاملات بطور ضمنی در عبارات مسیر و مولفه‌های همکار هر مولفه بیان شده‌است و تنها باید بطور صریح در شبکه پتری نشان داده شود. از نقطه‌نظر همگامی، تعامل بین مولفه‌ها برای هر مولفه در نمودار مولفه‌ها به دو گروه تقسیم می‌شود. در تعاملات ناهمگام، مولفه درخواست‌کننده یک عملیات، منتظر پاسخ مولفه همکار خود نمی‌ماند و به اجرای عملیات خود ادامه می‌دهد. در تعاملات همگام، مولفه درخواست‌کننده، منتظر پاسخ همکار خود باقی می‌ماند.

در تبدیل ارتباطات بین مولفه‌ها، گذار مربوط به مولفه درخواست‌کننده عملیات، با استفاده از یک مکان مشترک به گذار مربوط به مولفه فراهم‌کننده عملیات، متصل می‌گردد و سپس درصورت همگام بودن ارتباط، خروجی گذار فراهم‌کننده عملیات به خروجی گذار درخواست‌کننده عملیات، متصل می‌شود. در صورتیکه، ارتباط بین دو مولفه از نوع ناهمگام باشد، نتیجه‌ای به مولفه درخواست‌کننده عملیات بازنمی‌گردد. این مکان مشترک، تبادل پیغام بین دو مولفه را نشان می‌دهد و بعنوان پیش‌شرط برای شلیک شدن یک گذار در نظرگرفته می‌شود و کمان‌هایی که این مکان را به گذارهای دو مولفه متصل می‌کند، تبادل پیغام بین دو مولفه را انجام می‌دهد. مکان مشترک در اینجا نیز، یک صف انتظار است.

در این رابطه وضعیت‌های مختلفی که ارسال پیغام می‌تواند دنبال کند به شرح زیر است [۲۷ و ۳۱]:

- از نظر مولفه درخواست‌کننده یک عملیات، شلیک نمودن گذار مربوط، به معنی ارسال درخواست و قرار گرفتن مهره بر روی مکان مشترک است و جایگزین شدن مهره روی مکان بعد از آن گذار به معنی انتهای انجام درخواست است.
- از نظر مولفه انجام‌دهنده یک درخواست، جایگزین‌شدن مهره روی مکان مشترک به معنی انتظار برای برقراری شرط آن گذار، شلیک‌نمودن گذار به معنی انجام درخواست و اگر تعاملات همگام باشد به معنی ارسال نتایج است.
- از نظر پیغام درخواستی نیز، وقتی گذار مربوط به مولفه فرستنده شلیک می‌شود به مکان مشترک دسترسی دارد و وقتی گذار مربوط به مولفه دریافت‌کننده شلیک می‌شود، یعنی عملیات درخواستی انجام شده است.
- از نظر سیستم نیز وقتی فرستنده یک مهره را در مکان مشترک قرار دهد یعنی به عملیاتی در مولفه همکار نیازمند است و وقتی مولفه همکار در وضعیتی باشد که اجازه استفاده از مهره موجود در مکان مشترک را داشته باشد، فراخوانی درخواست را تائید می‌کند.

بعنوان مثال شکل ۱۳ شبکه پتری کامل نمودار مؤلفه شکل ۱۱ را نشان می‌دهد. همانطور که این شکل نشان می‌دهد، خروجی گذار x2 به ورودی گذار y1 متصل شده‌است. در این شکل مکان z، نقش مکان مشترک را دارد. این مکان برای عملیات درخواست‌کننده، نقش مکان خروجی و برای عملیات فراهم‌کننده، نقش مکان ورودی را بازی‌می‌کند [۲۷ و ۳۱].

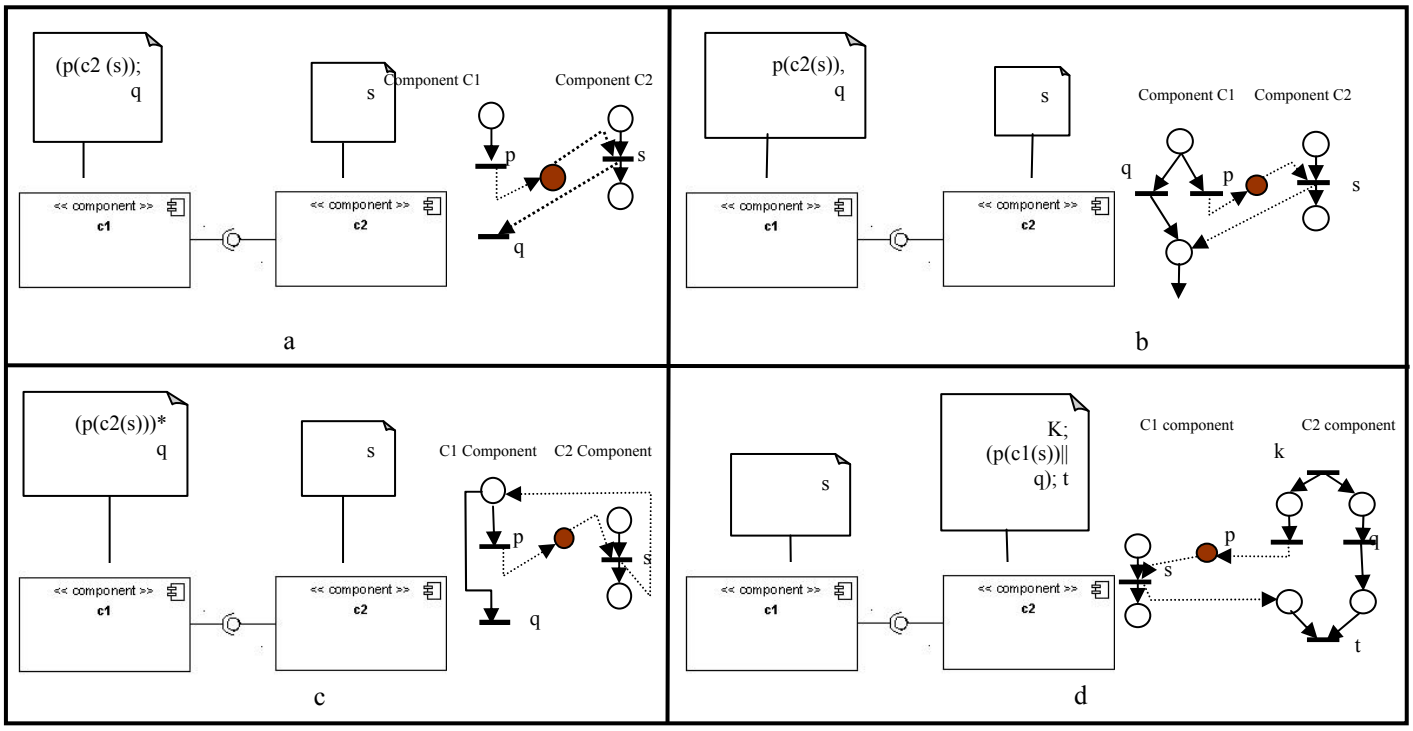


شکل ۱۳- شبکه پتری نمودار مؤلفه شکل ۱۱

همانطور که بیان شد، وظایف مولفه‌ها با تعریف عبارات مسیر برای مسئولیت‌های هر مولفه مشخص می‌شوند. در ادامه ساختارهای مختلفی که با این عبارات مسیر به همراه تعامل با همکاران ایجاد می‌شود به شبکه پتری تبدیل می‌شوند.

شکل ۱۴-۱ یک عبارت مسیر ترتیبی و تعامل همگام مولفه C1 و همچنین شبکه پتری معادل آن را نشان می‌دهد. همانطور که قبلا بیان شد، برای هر عملیاتی که در یک مولفه به عملیاتی در مولفه دیگر نیازمند است، باید نام عملیات مولفه همکار آن نیز ذکر شود. بنابراین همانطور که شکل ۱۴-۱ نشان می‌دهد، وقتی مولفه C1 و C2 با هم ارتباط برقرار می‌کنند، عملیات s از مولفه C2 استفاده می‌شود. شکل ۱۴-۲ نگاشت یک مولفه را با عملیات انتخاب به شبکه پتری معادلش نشان می‌دهد. برحسب انتخاب عملیات p یا q از مولفه C1 گذار مربوط به آن‌ها شلیک می‌شود که شرط انتخاب به گذارهای مربوطه الصاق می‌شود.

شکل ۱۴-۳ نگاشت یک مولفه را با عملیات تکرار به شبکه پتری معادل نشان می‌دهد. همانطور که شکل نشان می‌دهد، نقطه بازگشت بعد از شلیک‌شدن گذار مربوط به عملیات s از مولفه C2 مکانی است در مولفه C1 که عملیات s را فراخوانی نموده است.



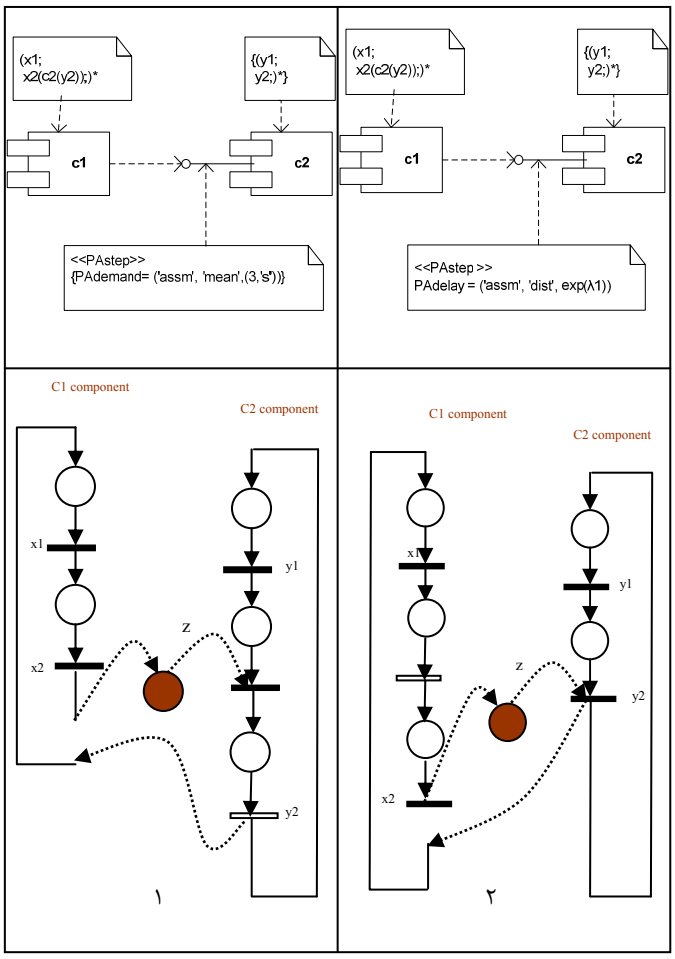
شکل ۱۴- عبارات مسیر مختلف بین مولفه‌ها در نمودار مولفه و شبکه پتری معادل آن‌ها

شکل ۱۴-۴ نگاشت یک مولفه را با عملیات همزمانی به شبکه پتری معادلش نشان می‌دهد. تکمیل دو مسیر همزمان، پیش‌شرط فعال نمودن عملیاتی است که بعد از مسیرهای همزمان قرارمی‌گیرد.

تبدیل برچسب‌های PAdelay و PAdemand به شبکه پتری: مراکز سرویس و مشخصات آن‌ها بوسیله نمودار مولفه مشخص می‌شود. هر مولفه یک مرکز سرویس تاخیری را نشان می‌دهد. رابط بین مولفه‌ها نیز مجموعه وظایف[۳۰] قابل اجرا توسط مرکز سرویس را نشان‌می‌دهد.

چون مولفه می‌تواند تعداد زیادی رابط داشته‌باشد بنابراین مرکز سرویس مربوطه می‌تواند چندین کلاس سرویس داشته‌باشد که در شبکه پتری رنگی با انتساب چندین رنگ به مهره‌های آن مولفه می‌توان نشان‌داد، که هر رنگ، مربوط به یک رابط مولفه است. زمان سرویس، از برچسب PAdelay / PAdemamd کلیشه PAstep استخراج می‌شود. درصورتیکه از برچسب PAdelay برای روابط یک مولفه استفاده‌شود، یعنی مولفه یک مرکز سرویس تاخیری است و با تاخیر وظیفه خود را انجام می‌دهد که نرخ تاخیر آن در برچسب بیان می‌شود. در صورتیکه از برچسب PAdemand برای روابط یک مولفه استفاده شود، یعنی مولفه یک مرکز سرویس انتظار است و باید منتظر پاسخ بماند و نرخ تاخیر به گذار مربوط به آن وظیفه اعمال می‌شود.

شکل ۱۵ نمودار مولفه و شبکه پتری معادل آن را در صورتیکه ارتباطات از نوع همگام باشد، نشان‌می‌دهد. در این شکل، ارتباط بین مولفه‌ها که با برچسب PAdemand و PAdelay حاشیه‌نویسی شده‌اند به شبکه پتری معادلشان تبدیل شده‌اند. در شکل ۱۵-۱ ارتباط بین دو مولفه با برچسب PAdemand حاشیه‌نویسی‌شده است. یعنی پردازش عمل x توسط مولفه فراهم‌کننده با تاخیر انجام می‌شود. در شکل ۱۵-۲ ارتباط بین دو مولفه با برچسب PAdelay حاشیه‌نویسی‌شده است که بدین معنی است که عمل x توسط مولفه درخواست‌کننده با تاخیر فراخوانی می‌شود [۲۷].



شکل ۱۵- نمودار مولفه با برچسب ۱- PAdemand و ۲- PAdelay و شبکه پتری معادل آن‌ها

۴- مثال کاربردی

در این بخش، مثالی کاربردی از سیستم تجارت الکترونیکی، معرفی می‌شود و الگوریتم ارائه‌شده در این مقاله بر روی آن اعمال می‌شود.

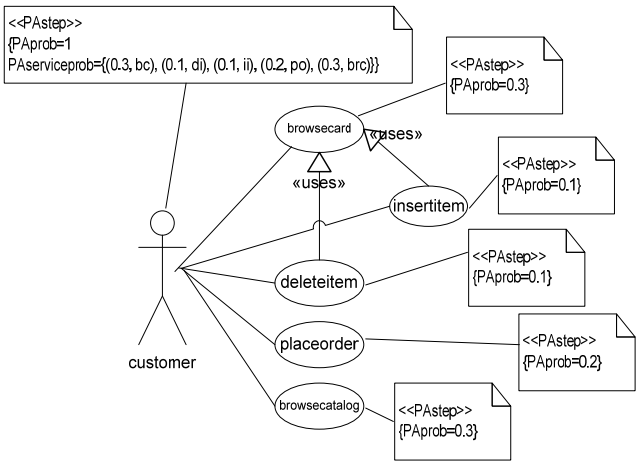
جریان رخداد اصلی سیستم تجارت الکترونیکی بدینصورت است که در این سیستم، یک کارپرداز[۳۱]، کاتالوگ‌هایش را روی وب منتشر می‌کند و سفارش مشتریان را دریافت و آن‌ها را به مشتری تحویل می‌دهد. برای رسیدن به این هدف، سیستم بایستی اطلاعات مربوط به داده‌ها، کاتالوگ و سفارش خرید را برای مشتریان نگهداری کند. هر مشتری کارتی دارد که با کمک آن می‌تواند گزینه‌های خود را اضافه و یا حذف نماید. مشتری نیز در صورتی می‌تواند درخواست سفارش کالا کند که کارت اعتباری او خالی نباشد. همچنین مشتری می‌تواند گزینه‌ای را به کارت خود اضافه و یا از کارت حذف نماید. در صورتی عمل حذف انجام می‌گیرد که گزینه موردنظر در کارت باشد. همچنین سیستم به مشتری اجازه می‌دهد تا نظارت کاملی بر وضعیت سفارش و تحویل کالای خود داشته باشد [۳۳].

شکل ۱۶ نمودار موردکاربری سیستم تجارت الکترونیکی را بهمراه کلیشه‌های مربوط به کارایی نشان می‌دهد که نمودار کاربری از مرجع [۳۳] استخراج شده‌است و پارامترهای کارایی به آن حاشیه‌نویسی شده‌است. مطابق این شکل، کاربر می‌تواند درخواست نماید که محتویات کارت او یا کاتالوگ نمایش داده‌شود، یا اینکه به کارت خود داده‌ای را وارد و یا حذف نماید و یا کالای موردنظر خود را سفارش دهد. همانطور که در این شکل مشخص است، برای هر عامل، احتمال استفاده از سیستم و احتمال استفاده از موارد کاربری تعیین شده است. همچنین احتمال استفاده از هر یک از موارد کاربری موردنیاز در ارزیابی کارایی نیز مشخص شده است. این حاشیه‌نویسی‌ها با هم ترکیب می‌شوند تا احتمال رفتار مورد کاربری x در سیستم را در طی اجرای سیستم نشان دهند.

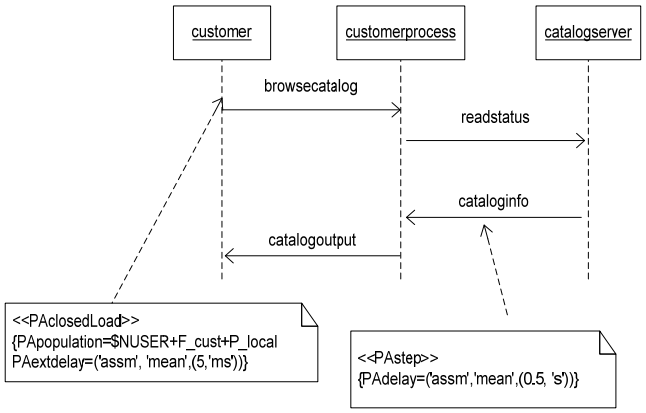
شکل ۱۷ نمودار ترتیب مورد کاربری نمایش کاتالوگ را بهمراه کلیشه‌های مربوط به کارایی نشان می‌دهد. طبق این نمودار، مشتری درخواست نمایش کاتالوگ را به مولفه customerprocess ارسال می‌نماید. این مولفه نیز پیغام readstatus را به catalogserver ارسال می‌نماید و درخواست خواندن وضعیت کاتالوگ مشتری را می‌نماید. در نهایت مولفه catalogserver، کاتالوگ داده‌ها را به مولفه customerprocess ارسال می‌کند.

شکل ۲۰ شبکه پتری رنگی نمودار ترتیب نمایش کاتالوگ را نشان می‌دهد. در این نمودار، برچسب PAdelay به یک گذار زمانی تبدیل می‌شود و تاخیر انتقال پیغام را نشان می‌دهد. به این نمودار بارکاری با کلیشه <<PAclosedload>> حاشیه‌نویسی شده‌است. در این کلیشه تعداد کاربران و زمان موردنیاز برای تولید یک درخواست مشخص شده‌است. البته تعداد کاربران می‌تواند تابعی از یک متغیر خارجی و بعضی مقادیر دیگر برای حالت‌های خاص باشد.

مثلا مطابق این شکل، $NUSER یک متغیر خارجی است و مقادیر f_cust و p_bs به ترتیب احتمال ورود مشتری به سیستم و احتمال درخواست مشتری از سرویس نمایش کارت را نشان می‌دهند. بارهای کاری منظورشده از نوع بسته است زیرا تعداد افرادی که اسامی آن‌ها در سیستم ثبت شده است، مشخص است و لذا جمعیت ثابت است. اما browsecatalog سرویسی است که برای تمام کاربران اینترنت در دسترس است و برای آن بایستی بار کاری باز درنظر گرفته شود اما برای ساده‌ب ودن مدل این بار کاری هم بسته منظور می‌شود [۳۳]. همچنین در این نمودار به پیغام cataloginfo کلیشه <<PAstep>> به همراه برچسب PAdelay حاشیه‌نویسی شده است. این حاشیه‌نویسی زمانی که customerprocess بعد از دریافت این دو پیغام و قبل از ارسال به مولفه بعدی صرف‌می‌کنند را نشان‌می‌دهد. این زمان از نوع تاخیری است.



شکل ۱۶- نمودار موردکاربری سیستم تجارت الکترونیکی به همراه کلیشه‌های کارایی

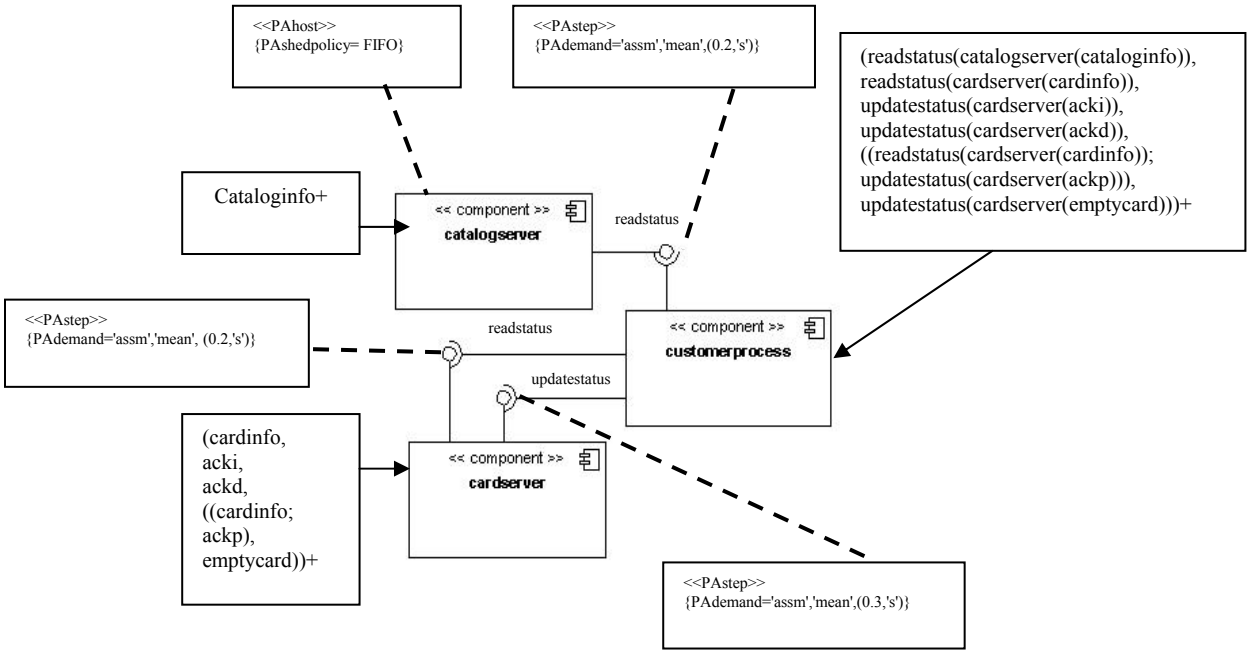


شکل ۱۷- نمودار ترتیب موردکاربری نمایش کاتالوگ به همراه کلیشه‌های کارایی [۳۳]

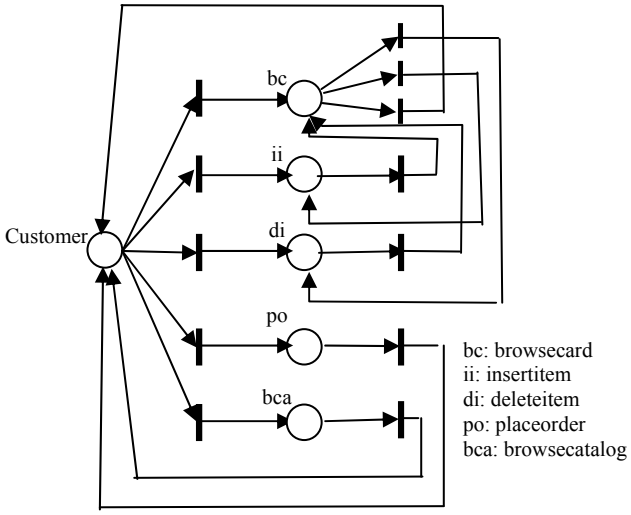
شکل ۱۸ بخشی از نمودار مولفه سیستم فوق را با کلیشه‌های مربوط به کارایی نشان می‌دهد. در این نمودار، ترتیب و دفعات اجرای مسئولیت‌های هر یک از مولفه‌ها بهمراه همکاران آن‌ها در کنار هر مولفه مشخص شده است. در ضمن این ارتباط با ارتباطات "نیاز دارد" و "فراهم می‌کند" موجود در مولفه نیز مشخص شده است. این نمودار بخشی از نمودار مولفه در مرجع [۳۳] بوده و عبارات مسیر به آن اضافه شده است.

در این شکل، اطلاعات اضافه‌شده به نمودار مؤلفه، نوع مراکز سرویس (بعنوان مثال سرویس‌دهنده‌هایی با صف انتظار یا تاخیر)، نرخ سرویس‌های فراهم‌شده توسط مولفه‌ها و سیاست زمان‌بندی مورداستفاده آن‌ها برای استخراج کارها از صف انتظار را مشخص می‌نماید. به همین منظور از کلیشه <<PAhost>>، برای مؤلفه‌ها، استفاده می‌شود تا منابع فعال، مدلسازی شوند. با این فرض که هر مؤلفه، بر روی یک وسیله فعال منطقی، مستقر است. در این کلیشه، برای مشخص نمودن سیاست زمان‌بندی، از برچسب PAschedpolicy ، استفاده می‌شود.

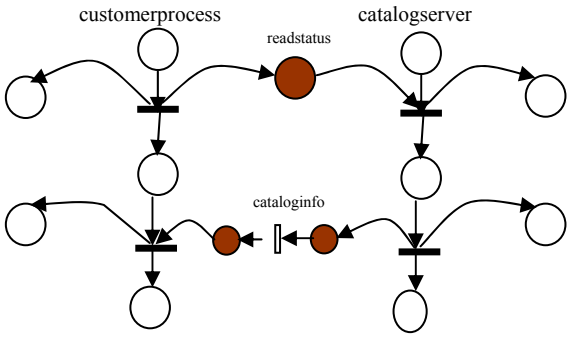
برای مشخص‌نمودن زمان مورد نیاز برای اجرای یک مؤلفه از کلیشه <<PAstep>>، بهمراه برچسب <<PAdelay>> یا <<PAdemand>>، استفاده‌شده‌است. برچسب <<PAdemand>>، نشان می‌دهد که مولفه، یک مرکز انتظار است و برچسب <<PAdelay>>، نشان می‌دهد که مولفه یک مرکز تاخیری است.



شکل ۱۸- بخشی از نمودار مولفه سیستم تجارت الکترونیکی به همراه اطلاعات کارایی و عبارات مسیر حاشیه‌نویسی شده به آن



شکل ۱۹- شبکه پتری حاصل از نمودار موردکاربری سیستم تجارت الکترونیکی

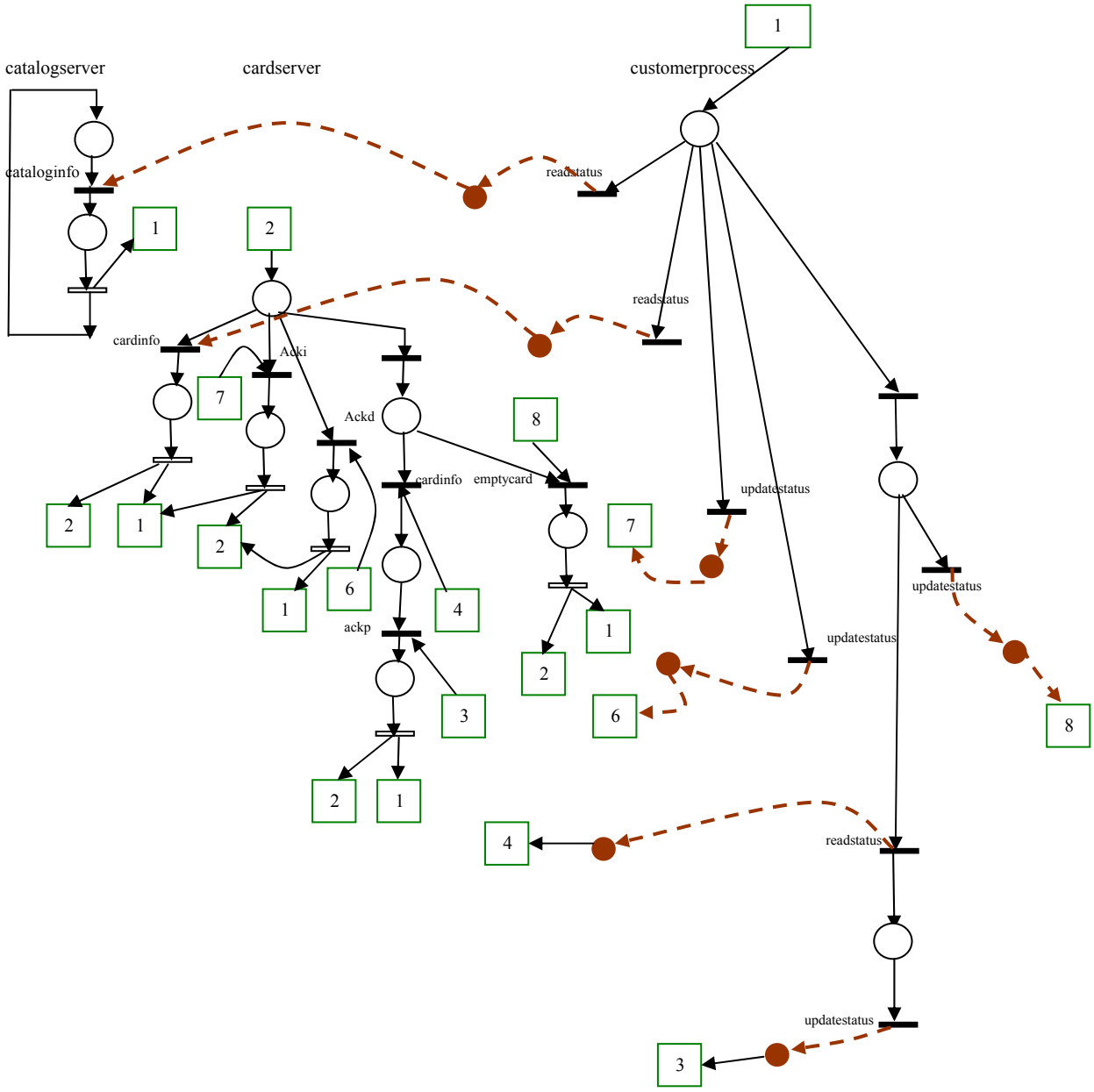


شکل ۲۰- شبکه پتری رنگی مبتنی بر زمان نمودار ترتیب نمایش کاتالوگ

شکل ۱۹ شبکه پتری نمودار مورد کاربری سیستم تجارت الکترونیکی را نشان می‌دهد. در این شکل احتمال انتخاب هر یک از موارد کاربری به گذارهای مربوط به آن‌ها در شبکه پتری نسبت داده‌شده‌است.

شکل ۲۰ شبکه پتری رنگی نمودارترتیب نمایش کاتالوگ را نشان می‌دهد. در این نمودار، برچسب PAdelay به یک گذار زمانی تبدیل می‌شود و تاخیر انتقال پیغام را نشان می‌دهد.

شکل ۲۱ شبکه پتری بخشی از نمودار مولفه سیستم تجارت الکترونیکی را نشان می‌دهد. در این مثال نوع ارتباطات برای همه مولفه‌ها از نوع همگام است. یعنی مولفه فراخوان بایستی تا دریافت نتایج منتظر بماند. در این شکل هر یک از مولفه‌ها در یک ستون نشان داده شده است که در هر ستون مکان‌ها وضعیت مولفه‌ها را و کمان‌ها، پیغام‌های ارسالی بین آن‌ها را نشان می‌دهند.



شکل ۲۱- شبکه پتری کامل نمودار مولفه سیستم تجارت الکترونیکی

۵- نتیجه‌گیری

هدف اصلی این مقاله، فراهم نمودن امکان تخمین و پیش‌بینی کارایی نرم‌افزار توسط معمار در سطح معماری نرم‌افزار بود. با این انگیزه که معمار این وظیفه را بدون هیچگونه یادگیری در رابطه با قراردادهای ریاضی که معمولا برای نمایش مدل‌های کارایی نیاز است و بدون صرف هزینه و زمان، انجام دهد، ضمن اینکه با تطبیق طراحی معماری با نتایج حاصل از تحلیل و ارزیابی مدل اجرایی، معمار می‌تواند کیفیت معماری و در نهایت کیفیت سیستم را با تصحیح مشکلات رفتاری قبل از پیاده‌سازی، بهبود بخشد.

در واقع این مقاله ابتدا بر روی ارزیابی کارایی در سطح معماری نرم‌افزار بدون درنظرگرفتن ویژگی‌های بستر سخت‌افزاری و سپس بر روی ارائه مدلی قابل اجرا و مبتنی بر شبکه‌های پتری از توصیفات معماری متمرکزگردید. هدف اصلی این مقاله، ارائه مدلی قابل اجرا به منظور ارزیابی کارایی در سطح معماری نرم‌افزار بود.

برای رسیدن به این هدف بایستی پلی مابین معمار و مدل اجرایی ایجاد شود تا فرآیند ارزیابی در این سطح انجام پذیرد. در این مقاله برای توصیف معماری از نمودارهای مورد کاربری، ترتیب و مولفه استفاده گردید که پارامترهای کارایی به آن حاشیه‌نویسی شدند.

سپس برای ایجاد مدل اجرایی از توصیفات معماری یکی از بسط‌های شبکه پتری به نام شبکه پتری رنگی تصادفی تعمیم‌یافته انتخاب گردید. در نهایت الگوریتم‌هایی برای تبدیل این توصیفات به شبکه پتری ارائه‌گردید. در این تبدیل هر یک از ارتباطات موجود در نمودارهای موردکاربری و ترتیب به شبکه‌پتری تبدیل‌شدند. همچنین به نمودار مولفه، عبارات مسیر اضافه‌گردید تا ترتیب اجرای هر یک از مسئولیت‌های مولفه‌ها به همراه همکاران آن‌ها مشخص‌شود. عبارات مسیر بکاررفته در این نمودار، پیش‌شرط‌های لازم برای فعال‌نمودن مسئولیت‌های مختلف یک مولفه را نشان‌می‌دهد. این عبارات مسیر علاوه بر همگام‌نمودن مسئولیت‌های یک مولفه و همچنین همگام‌نمودن یک مولفه با مولفه‌های دیگر،

اصول اولیه ساختیافته مدلسازی را نیز مشابه روش‌های برنامه‌نویسی ساخت‌یافته فراهم می‌کند.

در این مقاله سعی شده‌است که یک روش گام به گام ارائه شود تا تیم معماری مجبور نشود پس از آماده شدن مدل معماری، وقت زیادی را صرف ارزیابی نماید، بلکه فقط با رعایت نمودن نکاتی از همان ابتدا و در حین مستندکردن معماری، وقتی که نوبت به مرحلهٔ ارزیابی رسید در زمان مناسب و با صرف انرژی اندک، فرآیند ارزیابی را با دقت و صحت قابل قبولی انجام‌دهد.

تعدادی از فعالیت‌ها که می‌تواند باعث بهبود تحقیقات انجام‌شده در این مقاله گردد و در واقع نوعی محدودیت برای روش پیشنهادی در این تحقیق محسوب می‌شود، به شرح زیراست:

۱- نمودارهای زبان مدلسازی یکپارچه توصیف کننده معماری افزایش یابد. اگرچه در این تحقیق از نمودارهای زبان مدلسازی یکپارچه مناسبی برای توصیف معماری استفاده گردید. اما واقعیت این است که استفاده از نمودارهای زبان مدلسازی یکپارچه باعث بهبود کار می‌شود. بعنوان مثال نمودار کلاس برای استخراج بعضی جنبه‌های بار جمعیتی در سیستم می‌تواند استفاده شود. از نمودار استقرار در کنار نمودار مولفه برای اضافه نمودن جنبه‌های سخت‌افزاری استفاده می‌شود. زیرا آن‌ها توزیع مولفه‌ها در بسترهای سخت‌افزاری، توزیع بسترهای سخت‌افزاری بر روی منابع عملیاتی سیستم نشان‌می‌دهند.

۲- پیش‌بینی نیازهای کیفیتی دیگر در سطح معماری نرم‌افزار. در این فعالیت بایستی نمایه‌های این نیازها معرفی‌گردد تا به نمودارهای زبان مدلسازی یکپارچه حاشیه‌نویسی شوند. البته ابتدا بایستی بررسی شود هر نیاز غیروظیفه‌مندی در چه سطحی از توسعه باید ارزیابی‌شود و اینکه این نیازها در چه دامنه‌ای از کاربردها اهمیت دارند.

۳- بررسی روش‌هایی که امکان تبدیل توصیفات معماری را به مدل قابل اجرا بدون دخالت انسان و به صورت کاملا خودکار فراهم آورد. بگونه‌ای که بتوان مدل معماری را در قالب یک فایل ورودی به ابزار موردنظر داده و مدل اجرایی حاصل را تحلیل و ارزیابی نمود.

۴- در این تحقیق، شبیه بسیاری از روش‌های ارائه‌شده فرض شده است که اطلاعات اولیه در رابطه با پارامترهای کارایی موجود است. اما یکی از مسائل موجود برای ارزیابی در سطح معماری، بدست‌آوردن نمایه‌های عملیاتی و اطلاعات لازم در رابطه با پارامتر کارایی است. از فعالیت‌های بعدی این تحقیق، راه‌حلی برای بدست آوردن ساده‌تر این اطلاعات است.

مراجع

[1] N. Rosa, G. Justo, and P. Cunha, "A Framework for Building Non-Functional Software Architectures," *Proc. of the ACM symposium on Applied computing*, pp. 141-147, 2001.

[2] P. Clements, R. kazman, and K. Klein, *Evaluating Software Architecture Methods and Case Studies*, Addison Wesely, 2002.

[3] P. Avgeriou1, N. Guelfi1, and N. Medvidovic, "Software Architecture Description and UML," *Proc. of the Modeling Languages and Applications*, pp. 23-32, 2004.

[4] S. Emadi, and F. Shams, "A New Executable Model for Software Architecture Based on Petri Net," *Indian Journal of Science and Technology*, vol. 2, no. 9, pp. 15-25, 2009.

[5] N. Medvidovic, D. S. Rosenblum, D. F. Redmiles, and J. E. Robbins, "Modeling Software Architectures in the Unified Modeling Language," *ACM Transactions on Software Engineering and Methodology*, vol. 11, no. 1, pp. 2–57, 2002.

[6] S. Emadi, and F. Shams, "From UML Component Diagram to an Executable Model Based on Petri Nets," *Proc. IEEE the 3rd international symposium on information technology*, pp. 2780-2787, 2008.

[7] S. Emadi and F. Shams, "Mapping annotated use case and sequence Diagrams to a Petri Net Notation for Performance Evaluation", *Proc. of the 2nd International Conference on Computer and Electrical Engineering*, pp. 68-71, 2009.

[8] S. Emadi, and F. Shams, "Transformation of Usecase and Sequence Diagrams to Petri Nets", *Proc. of the International Conference on Communication Systems, Networks and Applications*, pp. 399-403, 2009.

[9] H. Miao, and J. Liu, "Modeling Architecture Based Development in UML," *Proc. of the 10th IEEE International Conference on Engineering of Complex Computer Systems*, pp. 56-65, 2005.

[10] G. Cernosek, and E. Naiburg, *The Value of Modeling*, a technical discussion of software modeling, IBM, June 2004.

[11] D. Milojicic, M. Breugst, I. Busse, J. Campbell, S. Covaci, B. Friedman, K. Kosaka, D. Lange, K. Ono, M. Oshima, C. Tham, S. Virdhagriswaran, and J. White. MASIF, "the OMG Mobile Agent System Interoperability Facility," *Proc. of the International Symposium on Mobile Agents*, pp. 50-67, 1998.

[12] S. Balsamo, A. D. Marco, P. Inverardi, and M. Simeoni, "Model-Based Performance Prediction in Software Development: A Survey," *IEEE Transaction on Software Engineering*, vol 30, nO. 5, pp. 295-310, 2004.

[13] Object Management Group, "Unified Modeling Language 2.0", http://www.omg.org/uml/, 2003.

[14] S. Balsamo, and M. Simeoni, "On Transforming UML Models into Performance Models," *Proc. of the Workshop on Transformations in UML*, pp. 1-6, 2001.

[15] S. Bernardi, S. Donatelli, and J. Merseguer, "From UML Sequence Diagrams and Statecharts to Analysable Petri Net Model", *Proc. of the 3rd International workshop on Software and performance* , pp. 35-45, 2002.

[16] F. Andol, F. Aquilani, S. Balsamo, and P. Inverardi. "Deriving Performance Models of Software Architectures from Message Sequence Charts," *Proc. of the 2nd International Workshop on Software and Performance*, pp. 47-57, 2000.

[17] V. Cortellessa, and R. Mirandola, "Deriving a Queueing Network Based Performance Model from UML Diagrams," *Proc. of the 2nd International Workshop on Software and Performance*, pp. 58-70, 2000.

Analysis," *Proc. of the Workshop on Software Architecture Description & UML*, 2004.

[30] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modeling with Generalized Stochastic Petri Nets*, J. Wiley, 1995.

[31] F. Shams, *Modelling the Behaviour of Processes Using Collaborating Objects*, PHD Thesis, University of Manchester, 1996.

[32] P. E. Lauer, and R. H. Campbell, "A Description of the Path Expression by Petri Nets," *Proc. of the 2nd ACM Symposium on Principles of Programming Languages*, pp. 95-105, 1975.

[33] A. D. Marco, *Model-based Performance Analysis of Software Architectures*, Ph.D thesis, university of L'Aquila, Italy, 2004.

سیما عمادی دریافت مدرک کارشناسی کامپیوتر - نرم‌افزار از دانشگاه آزاد اسلامی واحد میبد با اخذ رتبه اول در سال ۱۳۷۴. دریافت مدرک کارشناسی‌ارشد کامپیوتر ـ نرم‌افزار از دانشگاه آزاد اسلامی واحد نجف‌آباد با اخذ رتبه اول در سال ۱۳۷۶. دریافت مدرک دکتری کامپیوتر ـ نرم‌افزار از دانشگاه آزاد اسلامی واحد علوم و تحقیقات تهران با اخذ رتبه اول در سال ۱۳۸۷. مربی دانشگاه آزاد اسلامی واحد میبد از سال ۱۳۷۶ الی ۱۳۸۷ و استادیار از سال ۱۳۸۸ تاکنون. همچنین مدیر گروه کامپیوتر دانشگاه آزاد اسلامی واحد میبد از سال ۱۳۷۶ الی ۱۳۸۳و مدیر گروه کامپیوتر گرایش‌های نرم‌افزار و معماری دانشگاه آزاد اسلامی واحد علوم و تحقیقات یزد از سال ۱۳۸۹ تاکنون. زمینه‌های تحقیقاتی: معماری نرم‌افزار ـ ارزیابی کارایی و قابلیت اطمینان در سطح معماری نرم‌افزار - توسعه مبتنی بر مؤلفه و طراحی نرم‌افزارهای تحمل‌پذیر خطا.

آدرس پست‌الکترونیکی ایشان عبارت است از:

emadi@maybodiau.ac.ir

فریدون شمس مدرک کارشناسی مهندسی نرم‌افزار را از دانشکده مهندسی برق و کامپیوتر دانشگاه شهید بهشتی در سال ۱۳۶۶و کارشناسی‌ارشد مهندسی کامپیوتر از دانشگاه صنعتی شریف در سال ۱۳۶و مدرک دکتری مهندسی نرم‌افزار از دانشگاه منچستر انگلستان در سال ۱۳۷۵ اخذ نموده‌اند. زمینه تحقیقاتی ایشان معماری نرم‌افزار، معماری سازمانی، معماری سرویس‌گرا، متدولوژی‌های چابک، سیستم‌های مقیاس وسیع و مهندسی هستان‌شناسی است. ایشان در حال حاضر معاون فناوری اطلاعات و ارتباطات دانشگاه شهید بهشتی و نیز دانشیار دانشکده مهندسی برق و کامپیوتر این دانشگاه هستند. همچنین ایشان مدیریت دو گروه تحقیقاتی بنام‌های خودکارسازی مهندسی نرم‌افزار (aser.sbu.ac.ir) و معماری سیستم‌های اطلاعاتی (isa.sbu.ac.ir) را در دانشگاه شهید بهشتی بر عهده دارند.

آدرس پست‌الکترونیکی ایشان عبارت است از:

f_shams@sbu.ac.ir

[18] M. Abdollahi Azgomi, and A. Movaghar, "Towards an Object-Oriented Extension for Stochastic Activity Networks," *Proc. of the 10th Workshop on Algorithms and Tools for Petri Nets*, pp. 144-155, 2003.

[19] M. Elkoutbi, and K. Rodulf, "Modelling Interactive Systems With Hierarchical Coloured Petri Nets," *Proc. of the International Conference on Advanced Simulation Technologies*, pp. 432-437, 1998.

[20] J. M. Fernandes, S. Tjell, J. B. Jørgensen, and Ó. Ribeiro, "Designing Tool Support for Translating Use Cases and UML 2.0 Sequence Diagrams into a Coloured Petri Net," *Proc. of the IEEE 6th International Workshop on Scenarios and State Machines*, pp. 1-10, 2007.

[21] A. Ouardani, P. Esteban, M. Paludetto, and J. C. Pascal, "A Meta Modeling Approach for Sequence Diagram to Petri Nets Transformation within the Requirements Validation Process," *Proc. of the Annual European Simulation and Modelling*, pp. 345-349, 2006.

[22] S. Bernardi, and J. Merseguer, "Performance Evaluation of UML Design with Stochastic Well-Formed Nets," *Journal of Systems and Software*, vol. 80, no. 11, PP. 1843–1865, 2007.

[23] S. Balsamo, M. Marzolla, and R. Mirandola "Efficient Performance models in Component-Based Software Engineering," *Proc. of the 32nd Euromicro Conference on Software Engineering and Advanced Applications*, pp. 64-71, 2006.

[24] J. Merseguer, J. Campos, and E. Mena, "Analysing Internet Software Retrieval Systems: Modeling and Performance Comparison," *Wireless Networks: The Journal of Mobile Communication, Computation and Information*, vol. 9, no. 3, pp. 223-238, 2003.

[25] J. Merseguer, S. Bernardi, J. Campos, and S. Donatelli, "A Compositional Semantics for UML State Machines Aimed at Performance Evaluation," *Proc. of the 6th International Workshop on Discrete Event Systems*, pp. 295-302, 2002.

[26] J. Merseguer, and J. Campos, "Exploring Roles for the UML Diagrams in Software Performance Engineering," *proc. of the International Conference on Software Engineering Research and* Practice, pp. 43-47, 2003.

[27] O. M. GROUP, "UML Profile, for Schedulability, Performance, and Time," OMG document http:/www.omg.org, 2005.

[28] V. Cortellessa, A. D. Marco, P. Inverardi, F. Mancinelli, and P. Pelliccione, "A Framework for the Integration of Functional and Non-Functional Analysis of Software Architectures," *Electronic Notes in Theoretical Computer Science journal*, vol. 116, pp. 31-44, 2005.

[29] V. Cortellessa, A. D. Marco, P. Inverardi, H. Muccini, and P. Pelliccione, "Using UML for SA-Based Modeling and

اطلاعات بررسی مقاله:

تاریخ ارسال: ۸۷/۹/۱۴
تاریخ اصلاح: ۸۹/۵/۲۶
تاریخ قبول شدن: ۸۹/۸/۲۴
نویسنده مرتبط: دکتر سیما عمادی، دانشکده فنی و مهندسی، دانشگاه آزاد اسلامی واحد میبد، میبد، ایران.

---

[1] Correctness
[2] Process Oriented
[3] Product Oriented
[4] Semantic
[5] Unified Modeling Language (UML)
[6] Message Sequence Chart
[7] Architecture Description Language (ADL)
[8] Qualitative
[9] Actors
[10] Association
[11] Generalization
[12] Extend
[13] Include
[14] Provided Interface
[15] Required Interface
[16] Place
[17] Transition
[18] Token
[19] Stochastic Petri Net (SPN)
[20] Generalized Stochastic Petri Net (GSPN)
[21] Immediate Transition
[22] Timed Transition
[23] Communication
[24] Guard
[25] Asynchronous
[26] Synchronous
[27] Functional
[28] Regular Expression
[29] Action
[30] Job
[31] Supplier