دانشکده مهندسی کامپیوتر و فناوری اطلاعات

گروه هوش مصنوعی

تعیین تعداد کمینه log های مورد نیاز برای تشخیص صحیح یک مدل فرایندی بر اساس شبکه های پتری

رضا شمس

اساتید راهنما:

دکتر علی اکبر پویان

استاد مشاور:

داوود شمسی

پایان نامه جهت اخذ درجه کارشناسی ارشد

شهریور ۱۳۹۱

دانشگاه صنعتی شاهرود

دانشکده: مهندسی کامپیوتر و فناوری اطلاعات

گروه: هوش مصنوعی

پایان نامه کارشناسی ارشد آقای رضا شمس

تحت عنوان: تعیین تعداد کمینه log های مورد نیاز برای تشخیص صحیح یک مدل فرایندی بر اساس شبکه های پتری

در تاریخ ..... توسط کمیته تخصصی زیر جهت اخذ مدرک کارشناسی ارشد مورد ارزیابی و با درجه ..... مورد پذیرش قرار گرفت.

اساتید راهنما	امضاء	اساتید مشاور	امضاء
آقای دکتر علی اکبر پویان		داوود شمسی	

اساتید داور	امضاء	نماینده تحصیلات تکمیلی	امضاء
نام و نام خانوادگی:		نام و نام خانوادگی:	
نام و نام خانوادگی:			

تقدیم به

چشمان منتظر مادرم

و

دستان خسته پدرم

تشکر و قدردانی

پاس خدایی که آدمی را به نعمت تفکر آراست و اساتید فرزانه‌ای چون دکتر علی‌اکبر پویان را در مسیر راهم قرار داد تا از اندیشه نابشان بهره گیرم و دانش و بینششان را ره‌توشه خویش سازم.

آقایان پاس می‌دارم اندیشه بلندتان را و ارج می‌نهم همت والایتان را.

تشکر می‌نمایم از پدر و مادر یگانه‌ام، برادر و خواهرانم که وجودشان تکیه‌گاهی برای تمام لحظه‌های سخت من و دعاهایشان تنها سرمایه بال گشودنم بسوی خوشبختی است.

تعهد نامه

اینجانب **رضا شمس** دانشجوی دوره کارشناسی ارشد رشته **هوش مصنوعی** دانشکده دانشگاه صنعتی شاهرود نویسنده پایان نامه **تعیین تعداد کمینه log های مورد نیاز برای تشخیص صحیح یک مدل فرایندی بر اساس شبکه های پتری** تحت راهنمائی **دکتر علی اکبر پویان** متعهد می شوم .

- تحقیقات در این پایان نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است .
- در استفاده از نتایج پژوهشهای محققان دیگر به مرجع مورد استفاده استناد شده است .
- مطالب مندرج در پایان نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است .
- کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود می باشد و مقالات مستخرج با نام « دانشگاه صنعتی شاهرود » و یا « Shahrood University of Technology » به چاپ خواهد رسید .
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان نامه تأثیرگذار بوده اند در مقالات مستخرج از پایان نامه رعایت می گردد.
- در کلیه مراحل انجام این پایان نامه ، در مواردی که از موجود زنده (یا بافتهای آنها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است .
- در کلیه مراحل انجام این پایان نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری ، ضوابط و اصول اخلاق انسانی رعایت شده است .

تاریخ

امضای دانشجو

چکیده

مدیریت فرایندهای تجاری شامل برنامه‌ریزی، شبیه‌سازی و آنالیز یک سیستم تجاری است. امروزه گرایش فناوری اطلاعات به سمت مدیریت سیستم‌های گردش کار[1] است. سیستم مدیریت جریان کار، یک سیستم کامپیوتری است که به تعریف و مدیریت یکسری از فعالیت‌ها در یک سازمان، برای رسیدن به یک هدف خاص، می‌پردازد. برای شبیه‌سازی سیستم‌های گردش‌کار از مفهوم مدل فرایندی استفاده می‌شود. ساخت یک مدل فرایندی از روی ثبت‌رخدادها یکی از زمینه‌های مهم در مدیریت فرایندهای تجاری می‌باشد.

در این تحقیق به بررسی اطلاعات مفید در یک ثبت‌رخداد برای بازسازی یک مدل فرایندی پرداخته می‌شود. بعبارتی دیگر هدف از این پایان نامه بررسی حداقل تعداد دنباله رخدادهای مورد نیاز برای تشخیص یک مدل فرایندی می‌باشد. در این پایان نامه، ما شبکه‌ی پتری را بعنوان یک مدل فرایندی مورد بررسی قرار می‌دهیم. شبکه‌های پتری یکی از رایج‌ترین مدل‌های فرایندی است که مورد استفاده قرار می‌گیرد.

الگوریتم‌های زیادی برای استخراج یک مدل فرایندی از ثبت‌رخدادها وجود دارد، که هریک با توجه به مدل گردش‌کار مورد استفاده قرار می‌گیرد. هر چه دانش استخراج شده از ثبت‌رخدادها کامل‌تر باشد مدل فرایندی دقیق‌تری بدست می‌آید و مدیریت سیستم بهتر می‌شود. اما اطلاعات اضافی در یک ثبت‌رخداد نیز باعث کاهش کارایی الگوریتم‌های اکتشاف می‌شود. بنابراین ایجاد یک ثبت‌رخداد بهینه می‌تواند باعث افزایش کارایی الگوریتم‌های اکتشاف شود. در این تحقیق از الگوریتم آلفا بعنوان الگوریتم پایه برای ساخت مدل فرایندی استفاده می‌کنیم. همچنین راه حل پیشنهادی، یک ثبت‌رخداد کامل و بهینه را برای الگوریتم آلفا بدست می‌آورد.

کلمات کلیدی: شبکه پتری، فرایندکاوی، ثبت رخداد

[1] Workflow management system

مقالات استخراج شده:

۱- R.shams, Ali a. pouyan, "Number of required logs in completeness event logs to discover correct Process Model based on Workflow net". Information processing letters, elsevier , submitted

۲- R.shams, Ali a.pouyan, "Generate all possible logs from a Workflow model based on formal language", Journal of Computing Technologies, accepted

فهرست

عنوان صفحه

فهرست اشکال

عنوان صفحه

فهرست جداول

عنوان **صفحه**

فصل اول

مقدمه

۱-۱- مقدمه‌ای بر مدل‌سازی

امروزه، کامپیوتر نقش مهمی در زندگی بشر دارد و همچنین سیستم‌های مدیریتی نقش ویژه‌ای در پیشرفت فعالیت‌های بشر ایفا می‌کنند. با گسترش فعالیت‌های بشری، وجود یک سیستم مدیریتی که بتواند بر کارها نظارت کند، لازم بنظر می‌رسد. سیستمی که بتواند با خطایابی و بهینه‌سازی کارها، به فعالیت‌های یک سازمان نظارت داشته باشد. سیستم مدیریت جریان کار[۱] [۳-۱] ، یک مدل کلی را برای فرایندهای تجاری نشان می‌دهد. سیستم‌های جریان کاری در زمینه‌ی سیستم‌های پویای رخداد گسسته[۲] ایجاد می‌شوند. رفتارهای پویای DEDSs ها توسط دنباله رخدادهایی گسسته توصیف می‌شوند. این دنباله رخدادها توسط سیستم ایجاد می‌شوند.

تاکنون تکنیک‌های مختلفی برای مدل‌سازی سیستم‌های جریان فرایندکاری معرفی شده، که این تکنیک‌ها، برای تحلیل سیستم‌ها به منظور ارزیابی جنبه‌های عملیاتی و تصحیح خطا مورد استفاده قرار می‌گیرد. یکی از ابزارهای مهم مدل‌سازی، شبکه‌ی پتری[۳] [۴] می‌باشد. شبکه پتری ابزاری مناسب برای مدل‌سازی بر پایه منطق ریاضی بصورت گرافیکی می‌باشد. با وجود اینکه شبکه پتری گرافیکی است، اما پایه ریاضی قوی دارد. شبکه پتری، ابزاری گرافیکی جهت توصیف رسمی جریان فعالیت‌ها در سیستم‌های پیچیده است که یک مدل آسنکرون از سیستم ایجاد می‌کند.

۱-۲- مقدمه‌ای بر فرایندکاوی[۴]

در هنگام استفاده از تکنولوژی سیستم مدیریت جریان‌کاری، مشکلات زیادی پیش می‌آید. یکی از این مشکلات، نیازمند بودن سیستم به یک طرح جریان کاری است. به طور مثال طراح باید بتواند یک مدل با

[۱] Workflow management system
[۲] DEDSs
[۳] Petri net
[۴] Process mining

جزییات دقیقی طراحی کند، که این مدل باید جریان‌کار در این سیستم را توصیف کند. طراح باید دانش کاملی در مورد زبان جریان‌کاری و ارتباطات مدیریتی در سیستم داشته باشد. قبل از طراحی مدل مدیریت جریان-کاری، باید بتوان اطلاعاتی در مورد فرایند جریان‌کاری که در سیستم اتفاق می‌افتد، را بدست آورد.

بطور کلی می‌توان گفت که اساس فرایندکاوی، استخراج دانش از مجموعه ثبت‌هایی از رخدادهاست، که این رخدادها از سیستم اطلاعاتی بدست می‌آیند. فرایندکاوی حداقل به دو دلیل می‌تواند مفید باشد. اول اینکه، فرایندکاوی می‌تواند بعنوان ابزاری برای شناسایی و ردیابی کارهای کاربران و فعالیت‌های یک سیستم بکار رود. بعبارتی دیگر کارکرد فعالیت‌های مختلف فرایندها را در سیستم بدست آورد. گاهی اوقات امکان دارد همه اطلاعات مربوط به یک سیستم ثبت شود، اما فرایندهای فرعی (که جزیی از فرایند اصلی در نظر گرفته می‌شود) به طور کلی، نادیده گرفته شوند. بنابراین سیستم مدیریت اطلاعاتی باید مبنای کار خود را فرایندهای اصلی همچون استفاده از منابع و جریان زمان قرار دهد. دوم اینکه فرایندکاوی می‌تواند بعنوان ابزاری برای مقایسه فرایند واقعی با بعضی از فرایندهای از قبل تعریف شده به کار رود.

۱-۳- تعریف مساله

ساخت یک مدل فرایندی از روی ثبت‌رخداد[۱] به دو موضوع مهم بستگی دارد. اول به الگوریتم اکتشاف، دوم به مقدار اطلاعات نمایش داده شده در ثبت رخداد.

در سال‌های اخیر چالش‌های زیادی در زمینه‌ی فرایندکاوی مطرح شده است [۷-۵]. یکی از این چالش‌ها که کمتر به آن توجه شده، مساله کامل بودن ثبت‌رخداد[۲] [۸] است.

یکی از مسائلی است که در تشخیص یک مدل فرایندی از اهمیت ویژه‌ای برخوردار است، اطلاعاتیست که در یک ثبت رخداد نشان داده می‌شود. نشان دادن همه‌ی اطلاعات ممکن در یک ثبت رخداد تقریبا غیر ممکن

[۱] Event log
[۲] Completeness event log

است. بنابراین همه‌ی الگوریتم‌های اکتشاف، یکسری فرض‌ها و محدودیت‌هایی برای ثبت‌رخداد در نظر می‌گیرند. در بعضی از این الگوریتم‌ها، تنها اطلاعاتی در مورد ترتیب رخدادها مورد استفاده قرار می‌گیرد و در بعضی دیگر از الگوریتم‌ها تعداد تکرار رخدادها، نیز دارای اهمیت می‌باشد.

شاید یک ثبت‌رخداد شامل اطلاعاتی اضافی باشد، و برای مدل‌سازی سیستم نیازی به آن‌ها نباشد. اما پرسشی که در اینجا مطرح می‌شود این است که، آیا می توان یک ثبت‌رخداد بهینه برای یک مدل فرایندی خاص تعیین کرد. در این پایان‌نامه به این پرسش پاسخ داده می‌شود و این مساله را مورد بررسی قرار می‌دهد. با بررسی یک مدل فرایندی می‌توان به رفتارهای اساسی آن دست پیدا کرد و این رفتار اساسی را در ثبت‌رخداد نشان داد. ثبت‌رخداد بهینه، ثبت رخدادیست که با کمترین تعداد دنباله رخداد، بتواند همه‌ی رفتارهای اساسی یک سیستم را نشان دهد.

۱-۴- ساختار پایان‌نامه

در این پایان نامه در ابتدا در مورد مفاهیم و اساس سیستم‌های جریان‌کار توضیح داده شده است و سپس به معرفی شبکه‌ی پتری بعنوان ابزاری برای مدل‌سازی سیستم‌های جریان‌کار پرداخته است. انواع مختلفی از شبکه‌ی پتری وجود دارد، که بطور مختصر به توصیف آن‌ها پرداخته شده است. سپس به زمینه‌ی فرایندکاوی پرداخته شده است. الگوریتم‌های فرایندکاوی سعی بر ساخت یک مدل فرایندی از روی یک ثبت‌رخداد را دارند. ثبت‌رخداد شامل اطلاعاتی است که از سیستم گرفته می‌شود. در ادامه با تعریف ثبت‌رخداد بهینه، مساله‌ی بهینگی در یک ثبت‌رخداد کامل مورد بررسی قرار گرفته است. استفاده از ثبت رخداد بهینه تاثیر زیادی در الگوریتم‌های فرایندکاوی دارد و همچنین می‌تواند در زمان نیز صرفه‌جویی کند.

ساختار و محتوای این پایان‌نامه شامل هفت فصل است که مختصری از فصل‌های آتی در ادامه آورده شده است:

فصل ۲- بررسی سیستم‌های جریان کاری و کاربرد آن‌ها: در این فصل سیستم‌های جریان کاری و کاربرد آن‌ها در دنیای امروز مورد بررسی قرار می‌گیرد. همچنین در این فصل ابزارهای مختلف برای مداسازی آن بیان می گردد.

فصل ۳- شبکه پتری: در این فصل شبکه پتری بعنوان بک شیوه‌ی مدل‌سازی سیستم‌های جریان‌کاری مورد بررسی قرار می‌گیرد و همچنین ساختار و انواع مختلف آن به طور اجمالی بیان می‌گردد.

فصل ۴- فرایندکاوی : در این فصل، فرایندکاوی بعنوان شیوه‌ای برای ساخت یک مدل فرایندی از اطلاعات بدست آمده از سیستم، مورد بررسی قرار می‌گیرد. همچنین تکنیک‌ها و الگوریتم‌ها در این زمینه بیان می‌گردد.

فصل ۵- تعیین تعداد دنباله رخدادها در یک ثبت رخداد بهینه: در این فصل با توجه به رفتارهای اساسی یک مدل فرایندی، به بررسی وجود الگوریتم بهینه سازی ثبت‌رخداد پرداخته می‌شود و سپس تکنیک‌ها و الگوریتم‌های لازم برای تعیین تعداد دنباله رخدادها در یک ثبت‌رخداد بهینه بیان می‌شود.

فصل ۶- پیاده‌سازی وارزیابی نتایج: در این فصل نرم‌افزار PROM به عنوان نرم‌افزاری برای پیاده‌سازی الگوریتم‌های فرایند کاوی بیان می‌گردد و سپس با استفاده از این نرم‌افزار به نحوه‌ی عملکرد ثبت‌رخداد بهینه پرداخته می‌شود.

فصل ۷- نتیجه‌گیری و کارهای آتی: جمع‌بندی و کارهای آتی در این فصل بیان می‌گردد.

فصل دوم

سیستم مدیریت جریان کاری

۲-۱- مقدمه

امروزه، کامپیوتر نقش مهمی در زندگی بشر دارد. اکثر کارهای شرکت‌ها، کارخانجات، سازمان‌ها با مجهز به سیستم کامپیوتری می‌باشد. سیستم‌های کامپیوتری علاوه بر اینکه دقت کارها را زیاد می‌کند باعث افزایش سرعت فعالیت‌ها نیز می‌شود. با گذشت زمان، اعتماد مردم به سیستم‌های کامپیوتری بیشتر می‌شود و فعالیت های بشری خود را به‌سیستم‌های کامپیوتری مجهز می‌کند.

همانطور که نقش علوم کامپیوتر روز به روز در زندگی انسان‌ها افزایش پیدا می‌کند، سیستم‌های مدیریتی نقش ویژه‌ای در پیشرفت فعالیت‌های بشر ایفا می‌کند. با گسترش فعالیت‌های بشری، وجود یک سیستم مدیریتی که بتواند بر کارها نظارت کند، لازم بنظر می‌رسد. سیستمی که بتواند با خطایابی و بهینه‌سازی کارها، به فعالیت‌های یک سازمان نظارت داشته باشد. علاوه بر این، هماهنگی فعالیت‌ها و فرایندهای تجاری و سازمانی و بهینه‌سازی آن‌ها، جهت صرفه‌جویی اقتصادی، سبب شده است که سیستم‌های مدیریتی مورد توجه بسیاری از محققین در صنعت قرار گیرد.

هدف از مدیریت جریان‌کار، هماهنگی فعالیت‌ها و خودکارسازی فرایندهای تجاری توسط ترتیب مشخصی از کارها به همراه منابع اطلاعاتی مربوط به آن‌ها می‌باشد.

۲-۲- جریان‌کار

قبل از اینکه به مفهوم سیستم مدیریت جریان کار بپردازیم، ابتدا جریان‌کار را تعریف می‌کنیم.

طبق تعریف سازمان بین‌المللی استاندارد سازی سیستم مدیریت جریان‌کار [۱][۹] جریان‌کاری عبارت است از یک رویه خودکار که در آن فعالیت‌ها، سندها و اطلاعات، بر طبق یک‌سری قوانین از قبل تعیین شده، برای رسیدن به یک هدف تجاری، بین چند عامل مجزا، ردوبدل می‌شود.

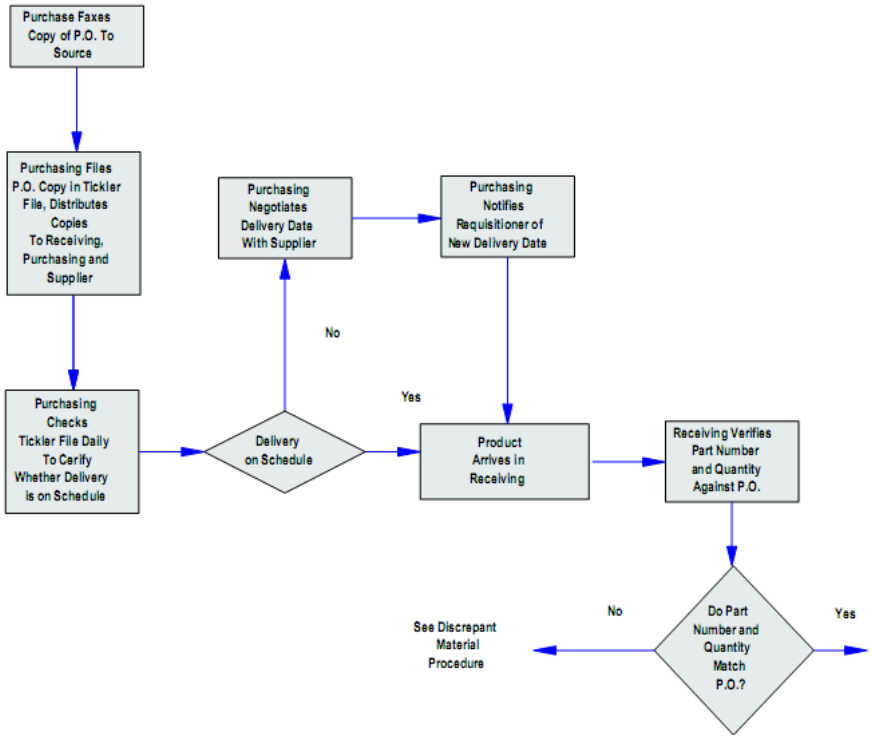
برای تعیین جزئیات مسیر و نیازمندی‌های فرایند یک جریان‌کار، از یک‌سری تکنیک‌های مدل فرایندی استفاده می‌کنیم. یک دنباله از این تکنیک‌های فرایند مدلی، مدل فرایندی تصمیمات زنجیره‌ای نام [۲]دارد. شکل (۲-۱)، مثالی از این تکنیک را نشان می‌دهد. در این تکنیک، از بخش‌های تصمیم‌گیری برای ساخت فرایند استفاده شده است. در شکل (۲-۲)، مدل فرایندی جریان رخداد، فرایند را بعنوان زنجیره‌ای از رخدادهای اتوماتیک و دستی نشان می‌دهد و همچنین جزئیات بیشتر در آن نشان داده شده است.

۲-۳- سیستم مدیریت جریان‌کار

سیستم مدیریت جریان‌کار، سیستمی است که توسط یک نرم‌افزار کامپیوتری به تعریف، اجرا و مدیریت یک جریان‌کار می‌پردازد[۱۰]. سازمان‌ها با استفاده از سیستم‌های مدیریت جریان‌کاری، فعالیت‌های مربوط به فرایندهای تجاری را کنترل می‌کنند. علاوه بر این، اکثر سیستم‌های مدیریتی، با اندازه‌گیری و آنالیز کردن اجرای فرایندها باعث بهبود پیوسته کارایی سیستم می‌شوند. این بهبودی می‌تواند کوتاه مدت(سازماندهی مجدد کارها برای تعادل سربار کار[۳] در هر زمان) یا بلند مدت (تعریف مجدد قسمت‌هایی از سیستم‌های فرایند جریان-کاری برای جلوگیری از ایجاد بن‌بست) باشد. همچنین سازمان‌ها سیستم‌های مدیریت جریان‌کاری را با دیگر سیستم‌ها همچون مدیریت اسناد، سیستم اطلاعاتی، پایگاه داده، سیستم مدیریتی و پست الکترونیکی ترکیب می‌کند. این ترکیب باعث ایجاد ساختاری می‌شود که فرایند بتواند از دیگر سیستم‌های مستقل استفاده کند.

---

[۱] WfMC
[۲] Decision -chain process model
[۳] workload



شکل (۲-۱) مدل فرایندی زنجیره تصمیم

۲-۳-۱- ویژگی‌های عمومی سیستم‌های مدیریتی جریان‌کار

سیستم‌های مدیریت جریان کار دارای ویژگی‌های مشترک زیادی می‌باشند که در ادامه به فهرستی از آن‌ها می‌پردازیم:

- ابزار تعریف فرایند: ابزاری متنی یا گرافیکی برای تعریف فرایندهای تجاری. هر فعالیت در فرایند مربوط به یک شخص یا یک برنامه کامپیوتری می‌باشد. یک‌سری قوانین برای تعیین مقدار پیشروی فرایندها در جریان کار تعریف شده است. بعضی از سیستم‌های مدیریت جریان‌کاری، توسط مدیر می‌توانند به‌صورت پویا تغییر پیدا کنند.

-شبیه‌سازی، نسخه اصلی و آزمایشی: بعضی از سیستم‌ها، نسخه اصلی و آزمایشی یک جریان‌کاری ویژه را شبیه‌سازی می‌کنند. این سیستم‌های جریان‌کاری، قبل از استفاده در قسمت‌های مختلف آزمایش و تست می‌شوند.

-راه اندازی و کنترل کارها: فرایندهای تجاری تعریف شده، راه‌اندازی می‌شوند. سپس منابع انسانی و اطلاعاتی برای کامل کردن فعالیت‌ها در فرایند تجاری، مورد استفاده قرار می‌گیرند.

-تصمیم‌گیری‌های مبتنی بر قانون: برای کنترل، ردیابی، مسیریابی داده‌های مربوط به یک جریان‌کاری، در هر قدم یک‌سری قوانین تعریف می‌شود. به‌طور مثال، می‌توان قانونی وضع کرد که وقتی یک شرایط مناسب بود یک ایمیل هشدار ایجاد کند و یا قوانین دیگر، شرایط مسیریابی سندها یا فعالیت‌ها را بر اساس محتویاتشان اجرا می‌کند.

-مسیریابی سندها: در سیستم‌های ساده، گاهی اوقات نیاز به جابجایی اطلاعات و فایل‌ها بین قسمت‌های مختلف وجود دارد. اما در سیستم‌های خبره، این کار توسط چک کردن سندها در خارج از یک صندوق مرکزی انجام می‌شود. در هر دو سیستم کاربر در فرایند می‌تواند توضیحی به سند زیاد کند. بدون این‌که در اصل سند تغییری ایجاد شود.

- فراخوانی برنامه‌ها برای دستکاری و بازبینی داده‌ها: پردازنده کلمات، صفحه گسترده‌ها[1]، سیستم‌های GIS، برنامه‌های تولیدی و غیره می‌توانند توسط کاربران جهت به‌روز رسانی و دستکاری آن‌ها فراخوانی شوند.

-فهرست کار: این فهرست به کاربر اجازه می‌دهد که فعالیت‌های جاری را با ویژگی‌هایشان همچون اولویت، تاریخ شروع و... شناسایی کند. در بعضی از سیستم‌ها سربار کار به‌خوبی قابل نمایش است. این سیستم‌ها، مکان کار را در فرایند کاری آنالیز می‌کنند و همچنین زمان فعالیت‌های گوناگون را پیش‌بینی می‌کنند.

[1] spreadsheet

-کارهای خودکار: کارهای برنامه‌ریزی شده می‌توانند به‌صورت خودکار فراخوانی شوند. این کارها می‌تواند شامل نوشتن یک نامه، ایمیل هشدار و یا اجرای یک برنامه تولیدی باشد.

-هشدار رخداد: کاربر یا مدیر می‌تواند متوجه اتفاقات مهم در جریان کار، همچون افزایش سربار شود.

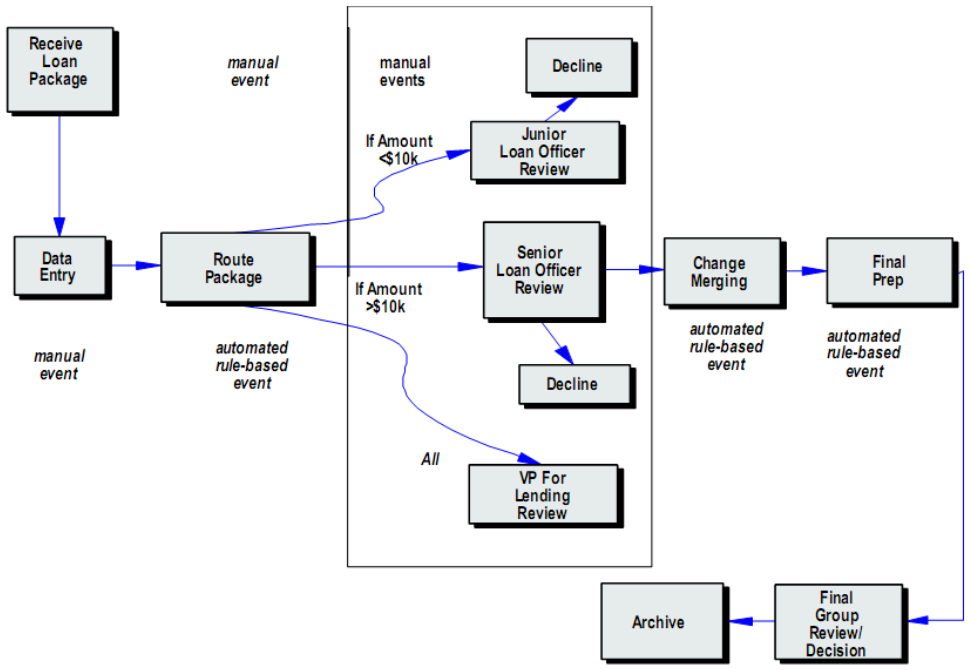
- لیست‌های توزیع شده برای پیغام‌ها: برای تبادل پیغام‌های ad-hoc بین کارمندان، لیست‌های توزیع شده می‌توانند مورد استفاده قرار گیرند.

-نظارت بر فرایندها: سیستم می‌تواند اطلاعات با ارزشی را در سربار جاری، سربار آینده، گلوگاه‌ها و... بدست آورد.

-دستیابی به اطلاعات در صفحات گسترده جهانی وب: بعضی سیستم‌ها برای دستیابی از راه دور مشتری‌ها، پشتیبان‌ها و... یک واسط مناسب درنظر می‌گیرند.

-ردیابی و ثبت فعالیت‌ها: اطلاعات هر قدم می‌تواند ثبت شود. این اطلاعات می‌تواند شامل زمان شروع و پایان فعالیت، کاربرهای منتسب به هر کار باشد. این اطلاعات می‌تواند بعدا برای آنالیز فرایند مورد استفاده قرار گیرد و زمان واقعی پایان کار را مشخص کند.

-کاربر اصلی و امنیت: هر برنامه کاربردی توابعی دارد که می‌تواند کاربران اصلی را شناسایی کند و به کاربران خاصی اجازه دستیابی دهد.



شکل (۲-۲) مدل فرایندی جریان رخداد

۲-۴- تاریخچه مدیریت جریان‌کار

در سال ۱۹۹۰، سیستم‌های مدیریتی جریان‌کار یکی از مهمترین برنامه‌های کاربردی در تکنولوژی اطلاعات شد. سیستم‌های مدیریت جریان کار به متخصصین کامپیوتر کمک کرد که بتوانند فرایندهای تجاری را بصورت خودکار در آورند. سیستم‌های مدیریت جریان کار بصورت پایه برای مکانیزه کردن اکثر برنامه‌های تجاری استفاده شد. سندسازی فرایندهای پردازش تصویر، سیستم‌های اداری و همچنین پردازش تراکنش‌ها توسط سیستم‌های مدیریت جریان کار گسترش یافتند. در ابتدا این سیستم برای ذخیره، بازیابی و ردیابی اطلاعات استفاده می‌شد. در ادامه نرم‌افزار جریان‌کار در سیستم‌های اداری مورد استفاده قرار گرفت. پست الکترونیکی به عنوان هسته‌ی ارتباطات برای جابجایی اطلاعات بین کاربران در سیستم‌های اداری قرار گرفت.

سیستم مدیریت جریان کار دارای چهار نسل می باشد، (جدول(۲-۱)) [۱۱]. در نسل اول، سیستم جریان-کار قسمتی از برنامه‌های کاربردی بود، مانند پست الکترونیکی و مدیریت سندها. معماری این سیستم‌ها

انعطاف‌پذیر نبود و برنامه‌نویس به‌سختی می‌توانست در سورس برنامه تغییری ایجاد کند و همچنین این نرم‌افزارها دارای معماری آزاد[1] نبودند. در نسل دوم، سیستم جریان‌کار از برنامه کاربردی جدا شد و بعنوان یک سیستم مجزا در نظر گرفته شد. در این نسل، جریان کاری به‌صورت یک زبان اسکریپتال در نظر گرفته شد. در نسل سوم، این سیستم‌ها به عنوان نرم‌افزاری با معماری آزاد. در این نسل، این سیستم به‌صورت مدل گرافیکی نمایش داده می‌شد. در نسل چهارم سیستم‌های مدیریت جریان‌کار با دیگر میان‌افزارها همچون پست الکترونیکی و مدیریت مسیرها ترکیب شد. همچنین در این نسل ظاهر ارتباطی[2] سیستم استانداردسازی شد.

جدول(۲-۱) نسل‌های سیستم مدیریت جریان‌کاری

نسل	ویژگی‌های متمایز کننده	ویژگی‌های اصلی
اول	برنامه‌های معین	- نرم‌افزار کاربردی با سیستم جریان‌کار یکی در نظر گرفته می‌شد. - سورس برنامه‌ها پیچیده بود. - نرم افزار دارای معماری آزاد نبود
دوم	برنامه‌های جدا شده	- ویژگی‌های جریان‌کار از نرم‌افزار جدا شد. - برنامه جریان کار به‌عنوان برنامه‌ها جداگانه در نظر گرفته شد. - فرایندها به‌صورت یک زبان اسکریپتال در نظر گرفته شد.
سوم	سرویس‌های مناسب	- سروریس‌های جریان کار در اختیار دیگر برنامه‌ها قرار گرفت - دارای معماری آزاد - دارای یک مدل گرافیکی
چهارم	افزودن سرویس‌های جدید	- سرویس‌های جریان‌کار با دیگر میان‌افزارها ترکیب شد. - استاندارد ظاهر گرافیکی - سیستمی جامع و غیر قابل مشاهده

[1] open source
[2] interface

۲-۵- مزایای سیستم‌های جریان‌کاری

ابزار مدیریت جریان‌کار باید بتواند اصول فرایندهای تجاری و ساختار سازمانی را بهبود بخشد. اگر سیستم مدیریت جریان‌کار بتواند بعنوان قسمتی از یک فرایند تجاری درنظر گرفته شود، می‌تواند مزایای زیادی را شامل شود.

- مناسب برای تغییرات سازمانی: در دنیای امروزی، تغییرات سریع و موثر می‌تواند تاثیر بسزایی در پیشرفت سازمان داشته باشد. سیستم‌های مدیریت جریان کار می‌توانند کمک زیادی در این زمینه داشته باشند. در حالی که نقش‌ها، قوانین و فعالیت‌ها در سیستم در نظر گرفته شده، اما این مشخصه‌ها کمتر در مدیریت فرایندها مداخله می‌کنند. علاوه بر این، اشتراک‌گذاری سندها و اگاهی از قسمت‌های مختلف سیستم می‌تواند باعث بهبودی در ارتباطات شود. همچنین با توجه به ردیابی فرایندها می‌توان هماهنگی بین اعضای تیم‌ها و واحدهای تجاری را بهبود بخشید. یکی از اهداف سیستم‌های مدیریت جریان کار، جمع آوری افراد، با تخصص گوناگون، در یک سازمان و هماهنگی بین آنان است.

بنابراین می‌توان نتیجه گرفت که می‌توان فرایندهای تجاری را توسط سیستم‌های مدیریت جریان کار تحت کنترل کاربران در آورد.

-مناسب برای تغییرات فرایندی: سیستم‌های مدیریت جریان کار سازمان‌ها را مجبور به بررسی و تعریف فرایندها می‌کند و باعث بهبود در فرایند تجاری می‌شود. در واقع می‌توان گفت که فرایندهای اصلی باید آنالیز شوند تا بتوان با تغییر در سیستم‌های مدیریت جریان‌کار، از کارهای اضافی و بیهوده در آینده جلوگیری کرد. می‌توان سه هدف زیر به‌عنوان اساسی‌ترین اهداف در را در بهبود فرایندها نام برد[۱۲]:

الف- کمینه کردن زمان فرایند:

- کاهش تعداد اجزاء در یک فرایند

- کاهش بیشینه زمان اجرای هر کار
- کاهش زمان برای جابجایی سندها و اطلاعات بین فعالیت‌ها
- کاهش زمان انتظار یک فعالیت در صف برای اتمام پروژه
- افزایش تعداد فعالیت‌هایی که به‌صورت موازی انجام می‌شود.

ب- بیشینه کردن ارزش محتوا (کاهش قیمت یک محصول و یا افزایش کیفیت یک محصول)

- در هر مورد جدید، مسیرهای فرایند کار، قوانین و نقش‌ها به صورت استاندارد تعریف شوند. فقط در شرایط خاص همچون شکایت مشتری و یا افزایش قیمت‌ها می‌توان از این استاندارد خارج شد.
- همه قسمت‌ها و اجزاء می‌توانند به‌صورت آنلاین به اطلاعات تمام سیستم دسترسی پیدا کنند.
- ردیابی به صورت پیوسته صورت می‌گیرد.
- حذف هزینه‌های مربوط به اسکن و پرینت سندها

ج- بیشینه کردن انعطاف‌پذیری در ارتباطات اولیه

- ایجاد دست‌یابی به قسمت‌هاب مختلف
- ثبت اطلاعات مشتری تنها در ارتیاط اولیه
- پشتیبانی از تراکنش توزیع شده
- با توجه به انعطاف‌پذیر بودن فرایند کاری می‌توان، یک فرایند را طبق نیاز مشتری تغییر داد.

-بهبود و افزایش دسترسی به اطلاعات: در واقع می‌توان گفت سیستم‌های مدیریت جریان‌کار، هماهنگی بین داده‌ها و دانش را در یک سیستم نشان می‌دهد. "فرایند کاری ایجاد کننده هوش تجاری از تجربات است". اطلاعات فرایندی که در بین اجزاء و کارکنان مختلف پراکنده شده است می‌تواند ترکیب شده و مورد استفاده همه قرار گیرد. بویژه این کار برای کارمندانی که تازه استخدام شده‌اند و اطلاعات زیادی از سیستم پیچیده سازمان ندارند مفید خواهد بود.

کارمندان می‌توانند اطلاعات بیشتری در مورد پروژه در اختیار دیگر اعضای تیم قرار دهند و در یک پروژه خاص می‌توان اطلاعات گذشته و جاری بک حالت خاص را همزمان در اختیار داشت و از آن‌ها استفاده کرد.

-بهبود امنیت و اعتبار: سیستم مدیریت جریان کاری، یک ارتباط امن برای تولید یک سرویس پایدار ایجاد می‌کند. مدیریت فرایند کاری داده‌ها را از برنامه‌های مختلف جمع‌آوری می‌کند و به این داده‌ها نظم می‌بخشد. با ایجاد مکانیزم‌هایی همچون تعیین حق دسترسی(تعیین می‌کند چه افرادی می‌توانند به داده‌ها دسترسی پیدا کنند ویا آن‌ها را تغییر دهند)، کنترل فرایند(یک سند قبل این که به مرحله بعدی برود باید تایید شود) و گرفتن نسخه پشتیبان، به سیستم اعتبار می‌بخشد.

۲-۶- انتخاب فرایند مناسب

مدیریت فرایند کاری برای فرایندهایی مفید واقع خواهد شد که بتوانند از همکاری بین قسمت‌های مختلف سیستم بهره گیرند. فرایندهایی که نیاز به جابجایی سندهای زیادی بین قسمت‌های مختلف سیستم دارند. در عین حال فرایندهای ساده نیز می‌توانند از مدیریت جریان کاری بهره ببرند. انعطاف‌پذیریه سیستم‌های مدیریت جریان‌کار می‌تواند برای همه‌ی فرایندها مفید باشد.

معیارهای زیر برای انتخاب فرایندهایی که بیشترین بهره را از سیستم‌های مدیریت جریان کار بیان شده است[۱۳]:

-سرعت: فرایندهای طولانی بیشترین بهره را از مدیریت جریان کار می‌برند. شکایت مشتری‌ها یا مدیرها (که چرا زمان این فرایند طولانی شده است) اغلب نیاز به مدیریت جریان کار را نشان می‌دهد.

-هزینه: آگاهی از فرایندهایی که مدیریت هزینه در بین قسمت‌های مختلف آن‌ها از اهمیت زیادی برخوردار می‌باشد.

-صحت: فرایندهایی که دقت در ارتباطات بسیار مهم می‌باشد.

-کیفیت: آیا کیفیت تولید نهایی مناسب است؟

-رضایت مشتری: آیا یک فرایند یک جریانی متداوم از شکایات یا مسایل اشتباه ایجاد می‌کند؟

-انعطاف‌پذیری: رویه‌های غیر قابل انعطافی که کارایی کارمندان در آن‌ها مهم می‌باشد و کارمندان باید در مورد عملکردشان توضیح دهند.

۲-۷- معیارهای موفقیت در سیستم‌های جریان‌کاری

در [۱۵-۱۴] یکسری معیار برای تعیین میزان موفقیت در یک سیستم مدیریت جریان‌کاری معرفی شده است که در زیر به تعدادی از آن‌ها می‌پردازیم:

- تمرکز بر اهداف تجاری: صرف زمان برای مطالعه بر روی سازماندهی سیستم و تعیین آن دسته از مدیریت جریان‌کاری که بیشترین سود را به سیستم می‌رساند و انتخاب پروژه‌هایی که به عنوان اهداف اصلی سیستم تعیین می‌شوند.

-تمرکز بر روی پروژه‌هایی که قابل فهم هستند: در ابتدای پیاده‌سازی پروژه‌هایی انتخاب می‌شوند که فعالیت‌های فرایندشان کاملا قابل فهم باشند.

-استفاده از معیارهای گوناگون: در نظر گرفتن معیارهایی که امکان دارد بهره‌برداری را محدود کند. وقتی که یکبار پیاده‌سازی شد، با ردیابی معیارها می‌توان سود و زیان آن‌ها را تعیین کرد. طول یک چرخه، تعداد خطاها، زمانی که طول می‌کشد یک فرایند به پایان برسد، و زمان دستیابی به داده‌ها می‌تواند مثال‌هاب خوبی از این معیارها باشد.

-بدست آوردن پشتیبانی از مدیریت قوی‌تر: وقتی که فرایندها در سیستم‌های تجاری قابل اندازه‌گیری باشند می‌توان گزارشات دقیق‌تر داد و پشتیبانی دقیق‌تری از مدیریت کرد.

-بدست‌آوردن پشتیبانی کارمندان: کارمندان باید متوجه تاثیر فعالیتشان در فرایندها باشند و باید بتوانند فرایندها را طراحی مجدد کنند. اگرچه می‌توان اکثر فعالیت‌ها را به‌طور خودکار در فرایند در نظر گرفت، اما همچنان فعالیت‌های بحرانی به تجربه و دانش کارمندان بستگی دارد. کارمندان باید شیوه‌ی کارکردن با سیستم را بدانند تا بتوانند بیشترین بهره بری را داشته باشند.

-ترکیب با سیستم جدید یا سیستم جاری: بیشتر سود و مزایای سیستم‌های مدیریت جریان کار از ترکیب با سیستم جاری در فرایند تجاری بدست می‌آید. یکی از اهداف جریان‌کاری، ترکیب این سیستم‌ها با فرایندهاست.

- اجرا در فازهای گوناگون: در فاز اول، قسمت‌های کوچکی در نظر گرفته شود با کارمندان محدود و فعالیت‌های محدود. اما در فازهای بعدی این سیستم گسترش پیدا کند.

۲-۸- انواع سیستم‌های مدیریت جریان‌کاری

می‌توان سیستم‌های مدیریت جریان‌کاری را به سه دسته تقسیم کرد: ۱- سیستم‌های جریان‌کاری تولیدی[۱]. ۲- سیستم‌های مدیریت جریان‌کاری مبتنی بر پیغام[۲] ۳- سیستم‌های مدیریت جریان‌کاری مبتنی بر وب[۳].

۲-۸-۱- سیستم‌های جریان کاری تولیدی

۲-۸-۱-۱- معرفی

[۱] production workflow
[۲] Messaging-based Workflow Systems
[۳] Web-based Workflow Systems

این سیستم‌ها، یکی از قسمت‌های اصلی فروشگاه‌ها می‌باشد. در فروشگاه‌ها از این سیستم‌ها بعنوان سیستم‌های مدیریت ذخیره‌سازی فایل‌ها، سیستم‌های سندسازی پردازش تصویر و ... استفاده می‌شود. در این سیستم‌ها فایل‌ها در یک مکان ذخیره می‌شوند، اما نحوه مسیر یابی و جابجایی داده‌ها مدیریت می‌شود[۱۴].

۲-۸-۱-۲- مزایا

این سیستم‌ها ویژگی و اطلاعات بیشتری را نسبت به ابزار پیغام‌رسانی پشتیبانی می‌کند. همچنین این سیستم‌ها در محدوده‌ی بیشتری از شبکه‌ها، اجرا می‌شود.

۲-۸-۱-۳- معایب

این سیستم‌ها نسبت به سیستم‌های دیگر گران‌تر هستند و همچنین برنامه‌هایی که از آن‌ها استفاده می‌کنند، هزینه‌های بیشتری دارند. این نرم افزارها جزء نرم‌افزارهای با معماری آزاد نیستند و وابسته به پلتفرم ماشین هستند.

۲-۸-۱-۴- رویکرد

این سیستم‌ها باعث کاهش هزینه در استفاده از جابجایی اطلاعات توسط کاغذ می‌شود. سازمان‌ها تنها یکبار سندها را از مشتری‌ها می‌گیرد و سپس آن‌ها را اسکن می‌کند. وسپس با توجه به درخواست‌ها جابجایی اطلاعات توسط فایل‌ها انجام می‌شود.

امروزه در سیستم‌های تولیدی زیادی از مدیریت جریان‌کاری تولیدی استفاده می‌کنند، همچون: مدیریت پایگاه‌داده‌ها، مدیریت سندها، مدیریت پروژه‌ها، مدیریت داده‌های تولیدی، مدیریت اشیاء و مدیریت پیغام‌های الکترونیکی.

۲-۸-۲- سیستم‌های مدیریت جریان‌کاری مبتنی بر پیغام

۲-۸-۲-۱- معرفی

این سیستم‌ها که گاهی اوقات، سیستم مدیریت جریان‌کار کاربر اصلی[۱] نامیده می‌شود، بر مبنای ابزار جابجایی اطلاعات همچون پست الکترونیکی است. برای جابجایی اطلاعات از پست الکترونیکی و فایل‌های واصل به آن استفاده می‌شود.

۲-۸-۲-۲- مزایا

این سیستم‌ها دارای هزینه پایینی می‌باشند، چون از پست الکترونیکی برای جابجایی اطلاعات استفاده می‌کند. این سیستم‌ها، با توجه به اینکه سرعت زیادی دارند، می‌توانند به‌صورت موازی در سازمان‌ها به‌کار روند.

۲-۸-۲-۳- معایب

این سیستم‌ها چون از پست الکترونیکی استفاده می‌کنند، مانند سیستم مدیریت جریان کاری تولیدی انعطاف‌پذیر نیستند. بعضی از دولت‌ها نیز به‌دلیل امنیت کم از پست‌های الکترونیکی، برای مدیریت جریان کار استفاده نمی‌کنند.

۲-۸-۲-۴- رویکرد

سیستم‌های مدیریت جریان‌کار مبتنی بر وب بیشتر برای سازمان‌هایی طراحی شده که پیچیدگی کاری آن‌ها کم است و مسیریابی کارها در سیستم آن سازمان‌ها ساده است.

سیستم‌های مدیریت کار مبتنی بر وب را می‌توان به سه قسمت تجزیه کرد: ۱- تکنولوژی پیغام الکترونیکی ۲- مدیریت فرم‌ها ۳- مدیریت پایگاه داده. پیغام الکترونیکی و جابجایی اطلاعات توسط پست الکترونیکی انجام

[۱] administrative

می‌شود. کاربران داده‌هایی که به صورت فرم هستند را بین قسمت‌های مختلف جابجا می‌کنند. همچنین کاربران می‌توانند در این فرم‌ها دستکاری کنند و در انتهای کار این فرم‌ها در پایگاه داده ذخیره می‌گردد.

۲-۸-۳- سیستم‌های مدیریت جریان‌کاری مبتنی بر وب

۲-۸-۳-۱- معرفی

این سیستم‌ها از برنامه‌های کاربردی در صفحات گسترده جهانی (www) برای تبادل اطلاعات استفاده می‌کند. با توجه به توانایی‌هایی که در چنین برنامه‌هایی وجود دارد، می توان سیستم مدیریت جریان کاری را گسترش داد. در این سیستم‌ها جابجایی اطلاعات بر پایه‌ی سرویس‌های مشتری-خدمتگذار انجام می‌شود.

۲-۸-۳-۲- مزایا

اکثر سازمان‌ها، امکانات اولیه برای استفاده از این سیستم را دارند. با توجه به اینکه این سیستم‌ها از سرویس مشتری- خدمت‌گذار استفاده می‌کنند، دارای انعطاف‌پذیری زیادی هستند. همچنین با توجه به اینکه صفحات گسترده جهانی در همه‌جا در دسترس است، می‌توان فعالیت‌های یک سازمان را گسترش بیشتری داد.

۲-۸-۳-۳- معایب

کاربران باید در زمینه اینترنت مهارت‌های کافی را داشته باشند وحتی با زبان‌های برنامه‌نویسی تحت وب، همچون جاوا آشنا باشند. تا بتوانند در مواقع ضروری تغییرات لازم را انجام دهند. مهمترین مسئله، امنیت تبادل اطلاعات در اینترنت است. چون سندها و اطلاعات با استفاده از اینترنت جابجا می‌شوند، حفظ امنیت آن‌ها بسیار مهم می‌باشد.

۲-۸-۳-۴- رویکرد

مهمترین تفاوتی که این سیستم با سیستم‌های بیان شده وجود دارد، گستردگی و نحوه دسترسی با آن و تعریف پلتفرم مناسب است. در سیستم‌های قبلی باید پروتکل‌های مناسبی برای ارتباط، تعریف کرد. در حالی که در این سیستم‌ها از پروتکل TCP/IP استفاده می‌کند.

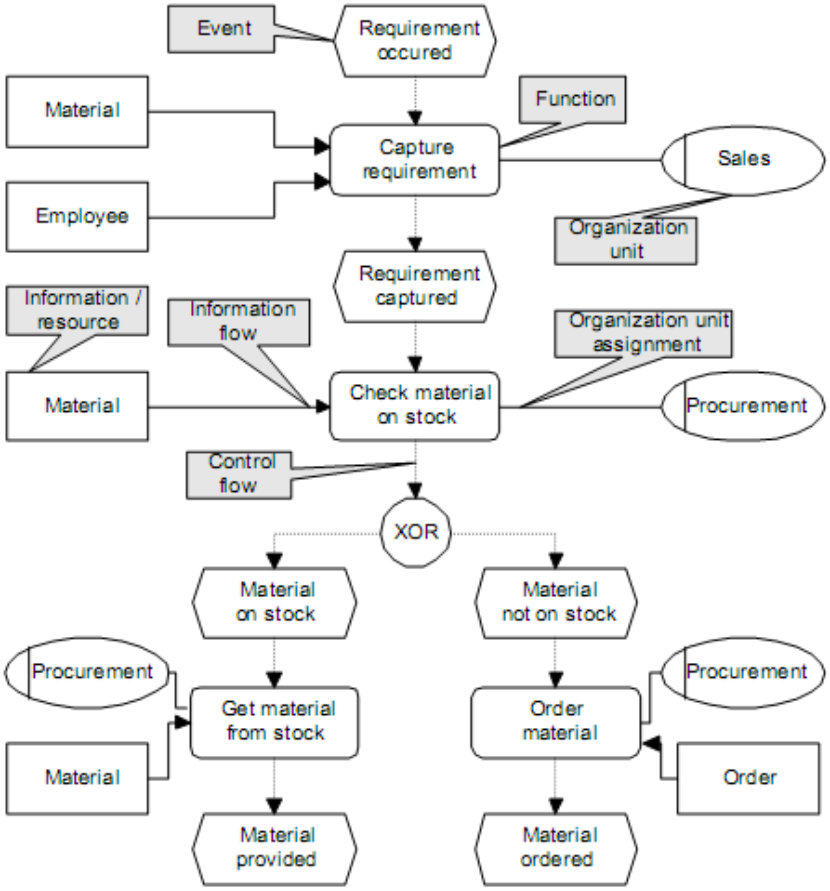
۲-۹- مدل‌سازی سیستم‌های مدیریت جریان‌کاری

برای پیاده‌سازی سیستم‌های مدیریت جریان کار شیوه‌های مختلفی دارد. سازمان‌ها با توجه به نیازشان پیاده‌سازی‌های مختلفی در نظر می‌گیرند. در این قسمت به معرفی چند شیوه برای نمایش و پیاده‌سازی این سیستم‌ها بیان می‌کنیم.

۲-۹-۱- زنجیره فرایندی مبتنی بر رخداد [۱]

زنجیره فرایندی مبتنی بر رخداد توسط Scheer، Keller و Nüttgens برای مدل کردن فرایندهای تجاری در معماری سیستم‌های اطلاعاتی مجتمع (ARIS) گسترش یافت. این شیوه به سرعت در پروژه‌های تجاری گسترش یافت. می‌توان گفت EPC گرافی مرتب از رخدادها و توابع است و دارای ارتباط دهنده‌های گوناگونی است که باعث اجرای فرایندهای موازی همزمان می‌شود [۱۶]. این شیوه از عملگرهای منطقی برای ارتباط استفاده می‌کند. مهمترین ویژگی آن ساده و قابل فهم بودنش است و به همین دلیل به‌طور گسترده در فرایندهای تجاری به‌کار می‌رود. شکل (۲-۳) مثالی از این شیوه را نشان می‌دهد.

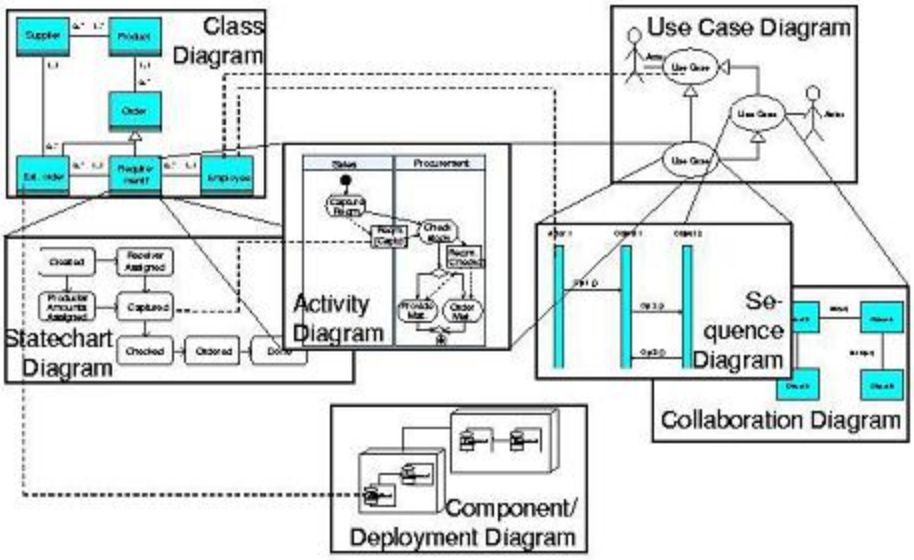
[۱] Event-driven Process Chain (EPC)



شکل(۲-۳) مثالی از EPC برای انتخاب تجهیزات نظامی

UML -۲-۹-۲

UML یک زبان مدل کردن ویژوال است که برای سندسازی در نرم‌افزارها وپروژه‌ها استفاده می‌شود. یکی از کاربردهای UML ساده‌سازی یک پروژه نرم‌افزاری می‌باشد. UML برای نمایش، از گراف استفاده می‌کند بنابراین با کاربر ارتباط خوبی برقرار می‌کند. UML خود یک زبان برنامه‌نویسی نیست، بلکه ابزاری است که برای استفاده دیگر زبان‌های برنامه نویسی، اطلاعاتی را فراهم می‌کند. UML برای نشان دادن ارتباط اجزاء مختلف سیستم، از هفت دیاگرام مختلف استفاده می‌کند که هر یک جنبه‌ای از فعالیت‌ها را زیر نظر دارد[۱۷].

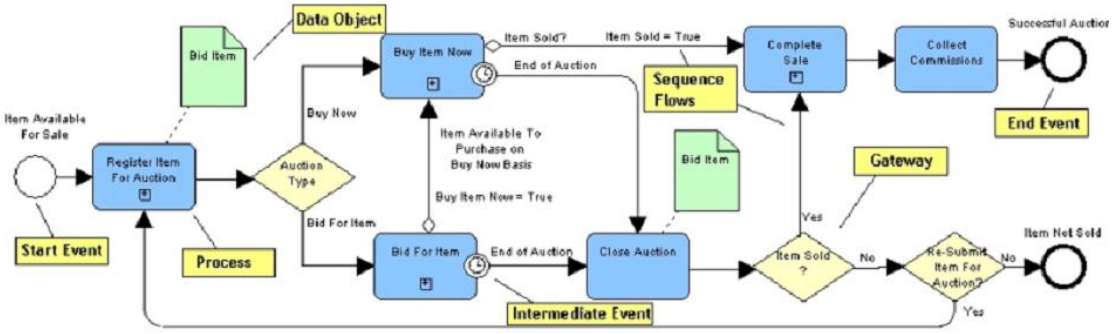


شکل(۲-۴) دیاگرام‌های یک UML وارتباط بین آن‌ها

۲-۹-۳- دیاگرام فرایند تجاری[۱]

این دیاگرام به دو منظور طراحی شده است. اول به منظور ساده کردن و قابل فهم کردن فرایند تجاری. شما به راحتی می‌توانید یک سیستم را مدل کنید بدون این‌که به دانش خاصی (مانند مدیریت) نیاز داشته باشید. دوم اینکه فرایندهای تجاری پیچیده را براحتی می‌تواند مدل کند و همچنین به سادگی می‌تواند به زبان‌های اجرایی تجاری نگاشته شود[۱۸]. برای مدل کردن یک فرایند تجاری، از یک رخداد که شروع کننده‌ی یک فرایند است آغاز می‌کنیم و سپس فرایند ادامه پیدا می‌کند و در رخداد انتهایی فرایند به اتمام می‌رسد. در شکل (۲-۵) یک دیاگرام فرایند تجاری را نشان می‌دهد.

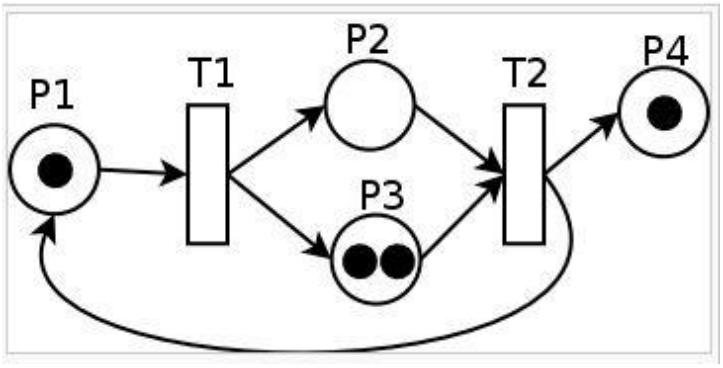
[۱] Business process Diagram



شکل (۲-۵) دیاگرام فرایند تجاری فروشگاه آنلاین

۲-۹-۴- شبکه‌های پتری

شبکه‌های پتری یکی از چند زبان مدل کردن ریاضی است که برای توصیف سیستم‌های توصیف شده، بکار می‌رود. این شبکه‌ها توسط آدام پتری [۱۹] در آگوست ۱۹۳۹ به منظور توصیف فرایندهای شیمیایی مطرح شد. شبکه پتری یک گراف دو بخشی است که دو نوع نود تشکیل شده است: ۱- نود گذر[۱] (رخدادی که امکان دارد رخ دهد). ۲- نود مکان[۲] (شرایط و حالات را نشان می‌دهد). یال‌های جهت‌دار[۳] ارتباط بین گذار و مکان‌ها را بر قرار می‌کند. شبکه‌های پتری را به دو صورت گرافیکی یا جبر ریاضی می‌توان نشان داد. که نحوه‌ی گرافیکی برای درک بیشتر کاربران است. در شکل (۲-۶) یک مثال از شبکه پتری را مشاهده می‌کنید.



شکل (۲-۶) مثالی از یک شبکه پتری

[۱] Transition
[۲] Place
[۳] Arcs

۲-۱۰- نتیجه‌گیری

در این فصل ابتدا جریان کار و مدیریت جریان کار معرفی کردیم و کاربردهای آن را در دنیای امروزی بیان کردیم. سپس تاریخچه‌ی این سیستم‌ها و تغییر و تحول آن‌ها را گفتیم. سپس مزایای این سیستم‌ها را بیان کردیم و همچنین به نحوه انتخاب فرایند مناسب و معیارهای برای این سیستم‌ها پرداختیم و سپس چهار شیوه مدل کردن سیستم‌های فرایند کاری همراه با معایب و مزایای آن‌ها بیان کردیم. در فصل آتی شبکه‌های پتری به صورت دقیق‌تر مورد بررسی قرار می‌گیرد و شیوه مدل کردن توسط شبکه‌های پتری به‌صورت کامل توضیح داده خواهد شد.

فصل سوم

شبکه پتری

۳-۱- مقدمه

محققین برای غلبه بر مسایلی همچون آنالیز، طراحی و کنترل یک سیستم، به مدل‌سازی سیستم با استفاده از مبانی ریاضی پرداختند. این مدل‌ها، مدل‌های رخداد گسسته [۱](DEMs) نامیده می‌شوند. هدف اصلی این مدل‌ها، بررسی ویژگی‌ها و رفتارهای سیستم است [۲۳-۲۰].

تئوری شبکه پتری بعنوان یک مدل رخداد گسسته و یک زبان مدل‌سازی سطح بالا، چارچوبی را برای توصیف، آنالیز و مدل‌سازی سیستم را ارائه می‌دهد. نقطه‌ی قوت تئوری شبکه پتری پایه‌ی ریاضی آن است. همچنین این تئوری، سیستم را از جنبه‌های مختلفی همچون نمایش گرافیکی، مورد بررسی قرار می‌دهد و شیوه‌های مختلفی را برای آنالیز سیستم در نظر می‌گیرد.

شبکه پتری ابزاری مناسب برای مدل‌سازی بر پایه منطق ریاضی بصورت گرافیکی می‌باشد [۲۶-۲۴]. با وجود اینکه شبکه پتری گرافیکی است، اما پایه قوی ریاضی دارد. شبکه پتری، ابزاری گرافیکی جهت توصیف رسمی جریان فعالیت‌ها در سیستم‌های پیچیده است که یک مدل آسنکرون از سیستم ایجاد می‌کند. شبکه پتری رنگی [۲۷] را می‌توان با مفهوم شی گسترش داد و به شبکه پتری شی [۲] رسید.

در عین حال، می‌توان گفت که تئوری PN [۳] یک زبان مدل‌سازی سطح بالاست که هنوز می‌تواند توسعه‌های زیادی یابد. با پیچیده شدن سیستم پیچیدگی PN نیز افزایش پیدا می‌کند. بنابراین می‌توان مساله آنالیز و محاسباتی زیادی روی سیستم تعریف کرد. مساله‌ی آنالیز مدل PN شامل بررسی ویژگی‌هایی همچون زنده بودن [۴]، محدود بودن [۵] و برگشت‌پذیری [۶] یک مدل PN است [۳۰-۲۸].

---

۱ Discrete event models
۲ Object petri net
۳ Petri net
۴ liveness
۵ boundedness
۶ reversibility

۳-۲- تعریف رسمی یک شبکه پتری

بطور کلی یک شبکه‌ی پتری عبارت است از یک ارتباط منطقی بین یکسری عناصر فعال (رخدادها) و عناصر غیرفعال (شرایط) است[۳۰].

تعریف ۳-۱- ساختار یک شبکه پتری: یک شبکه‌ی پتری یک چهارگانه‌ی $N = (P, T, F, V)$ است که:

P: مجموعه‌ای محدود از مکان‌های شبکه

T: مجموعه‌ای محدود از گذارهای شبکه

F: زیر مجموعه‌ای از $\{(P\times T)\cup(T\times P)\}$ است (مجموعه‌ای از کمان‌هاست).

V: $F\rightarrow \mathbb{N}$، که یک تابع تعداد است.

$\mathbb{N}$ مجموعه‌ای از اعداد غیر منفی است و $P\cap T = \emptyset$ و $P\cup T \neq \emptyset$

اگر در چهارگانه‌ی فوق، تابع اعداد در نظر گرفته نشود، شبکه پتری، شبکه P/T نامیده می‌شود [۴]. یک شبکه‌ی پتری شامل یکسری نود و کمان‌های جهت‌دار بین آن‌هاست. نودها به دو دسته تقسیم می‌شوند: یکی مکان‌ها (عناصر غیر فعال) و دیگری گذارها (عناصر فعال). مکان‌ها و گذارها توسط کمان‌های جهتدار به یکدیگر متصل‌شده‌اند (بعبارتی دیگر شبکه پتری یک گراف دوبخشی است). تابع تعداد، اتصالات را توصیف می‌کند. بطور مثال، اگر برای یک مکان P و گذار t داشته باشیم $V(p,t) = k\ (k \neq ۰)$ آنگاه K کمان از p به t وجود دارد (یا یک کمان با وزن K) و یا برعکس اگر $V(t,p) = k\ (k \neq ۰)$ آنگاه K کمان از t به p وجود دارد (یا یک کمان با وزن K).

مکان p، مکان ورودی گذار t است اگر $V(p,t) \neq 0$ (حداقل یک کمان از p به t وجود داشته باشد) و همچنین مکان p، مکان خروجی گذار t است اگر $V(t,p) \neq 0$ (حداقل یک کمان از t به p وجود داشته باشد). مجموعه مکان‌های ورودی و خروجی یک گذار را با $I(t)$ و $O(t)$ یا $\{•t\}$ و $\{t•\}$ نشان می‌دهند. به همین صورت نیز می‌توان گذارهای ورودی و خروجی را برای یک مکان نیز تعریف کرد. تعداد مکان‌های ورودی و خروجی p توسط تابع V مشخص می‌شود. بطور مثال، اگر برای یک مکان p و گذار t داشته باشیم: $V(p,t)=k$، آنگاه تعداد مکان p در مکان‌های ورودی t به تعداد K است. اگر تابع تعداد را $F \rightarrow \{0,1\}$ درنظر بگیریم، آنگاه همه‌ی کمان-ها دارای وزن یک می‌شوند[۳۱].

۳-۲-۱- نمایش گرافیکی شبکه‌ی پتری:

یک شبکه پتری از چهار عنصر پایه تشکیل شده است : مکان، گذار، کمان و نشانه[۱].

مکان: نمایانگر وضعیت‌های ممکن سیستم هستند. هر مکان می‌تواند شامل تعدادی نشانه باشد.

نشانه: بیان می‌کنند که سیستم یا بخشی از آن در زمان جاری در چه حالتی ممکن است قرار داشته باشند.

گذار: فعالیت‌های سیستم را نشان می‌دهند. یک گذار، زمانی فعال[۲] می‌شود که تمام مکان‌های ورودی آن نشانه داشته باشند. یک گذار فعال می‌تواند اجرا[۳]شود که در این صورت یک نشانه از هر مکان ورودی آن کم می شود و یک نشانه به هر مکان خروجی آن اضافه می‌گردد. چنانچه دو یا چند گذار همزمان فعال باشند، هر یک از آنها ممکن است اجرا شوند. انتخاب یک گذار از میان گذارهای فعال پیش بینی نشده و تصادفی است [۳۲].

کمان: مکان‌ها را به گذارها و بالعکس متصل می‌کند. بین دو مکان یا دو گذار نمی‌توان کمان داشت[۳۳].

---
[۱] Token
[۲] Enable
[۳] Fire

از کنار هم قرار دادن و اتصال این اجزاء گراف شبکه پتری تشکیل می‌شود که قوانین خاص خود را دارد.

۱. هیچ دو گذاری نمی‌توانند بطور مستقیم به هم متصل شود.

۲. هیچ دو مکانی نمی‌توانند بطور مستقیم به هم متصل شوند.

۳. بین هر دو اتصال بین گذارها حتماً می‌بایست یک مکان قرار بگیرد.

تعریف ۳-۲- یک گراف شبکه پتری، یک گراف دو بخشی جهت‌دار از ساختار شبکه پتری است.

با توجه به این‌که نودها به دو دسته تقسیم می‌شوند، گراف دو بخشی است. در شکل (۳-۱) مثالی از یک گراف شبکه پتری را مشاهده می‌کنید.

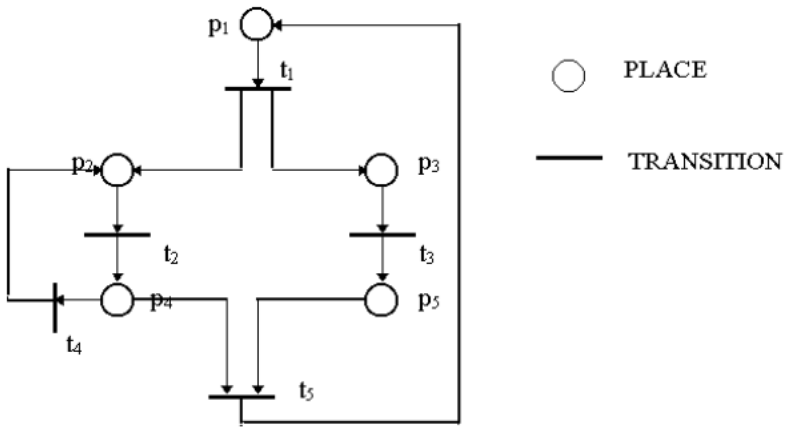
$I(t_1) = \{p_1\}$ $O(t_1) = \{p_2, p_3\}$

$I(t_2) = \{p_2\}$ $O(t_2) = \{p_4\}$

$I(t_3) = \{p_3\}$ $O(t_3) = \{p_5\}$

$I(t_4) = \{p_4\}$ $O(t_4) = \{p_2\}$

$I(t_5) = \{p_4, p_5\}$ $O(t_5) = \{p_1\}$

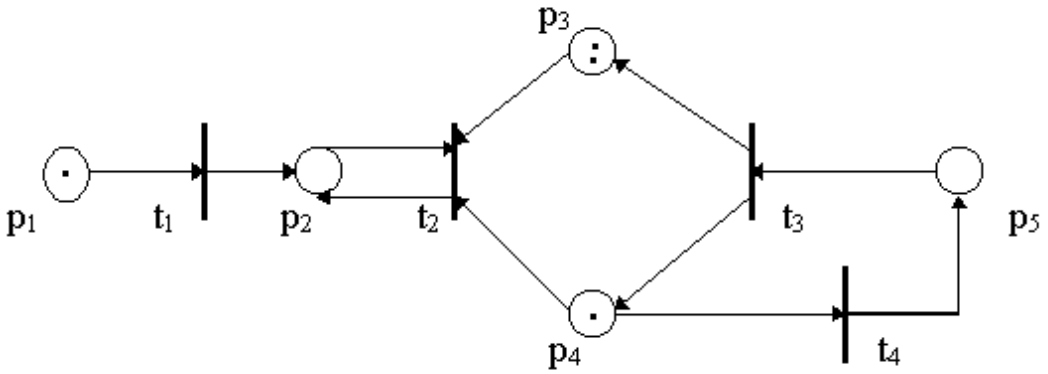


شکل (۳-۱) مثالی از یک گراف شبکه پتری

به تخصیص نشانه‌ها به مکان‌ها در یک شبکه پتری، نشانه‌گذاری[۱] گفته می‌شود. نشانه‌ها در مکان‌های شبکه-ی پتری قرار می‌گیرند. در گراف شبکه پتری ، نشانه‌ها بصورت نقاط سیاه نمایش داده می‌شوند. نشانه‌ها در بین مکان‌ها حرکت می‌کنند که این پویا بدن رفتار سیستم را نشان می‌دهد.

تعریف ۳-۳- یک نشانه‌گذاری μ از یک شبکه‌پتری $N=(P, T,F,V)$ نگاشتی است که: $\mu : P \rightarrow \mathbb{N}$

در یک شبکه‌ی پتری تعداد نشانه‌های مکان p با $\mu(p)$ مشخص می‌شود. یک مکان $p \in P$ نشانه‌دار[۲] است اگر $\mu(p)>0$ ، در غیر این صورت بی‌نشانه[۳] است. تعداد نشانه‌ها در هر مکان حالت[۴] شبکه پتری را مشخص می‌کند. حالت اویه بصورت μ. نشان داده می‌شود.



شکل (۳-۲) یک شبکه پتری نشانه‌گذاری شده با $\mu = (1,0,2,1,0)$

برای نشان دادن حالت یک شبکه پتری از یک بردار استفاده می‌کنیم، که هر مولفه‌ی آن نشان‌دهنده تعداد نشانه‌های داخل مکان‌ها است. بعبارت دیگر اگر تعداد نشانه‌های مکان p_i را با μ_i نشان دهیم آنگاه می‌توان از بردار $\mu = (\mu_1, \mu_2,..., \mu_n)$ برای نشان دادن حالت شبکه‌ی پتری استفاده کرد.

[۱] marking
[۲] marked
[۳] unmarked
[۴] state

تعریف ۳-۴- یک شبکه پتری نشانه‌گذاری شده $M = (N,\mu)$، یک شبکه پتری $N=(P, T, F, V)$ با تابع نشانه‌گذاری μ است.

در شکل (۳-۲) تصویر یک شبکه پتری نشانه‌گذاری شده را مشاهده می‌کنید. در تعریفی که از ساختار شبکه پتری شد، فرض بر این شد که تعداد نشانه‌های هر مکان نامحدود باشد. چنین شبکه‌هایی، شبکه‌های ظرفیت-نامحدود[۱] نامیده می‌شود. در حالی، معمولا برای مدل‌سازی سیستم‌های واقعی، هر مکان حداکثر تعدادی معین می‌تواند نشانه بپذیرد. این شبکه‌ها، شبکه‌های ظرفیت-محدود نامیده می‌شود.

رفتار یک شبکه‌ی پتری بر اساس قوانین اجرا[۲] مشخص می‌شود. قوانین اجرا نشان می‌دهد که یک گذار تحت چه شرایطی می‌تواند اجرا شود. بعبارتی دیگر یک گذار زمانی فعال می‌شود که تعداد نشانه‌ها در مکان‌های ورودی هر گذر بیشتر از تعداد کمان‌های بین آن گذر و مکان باشد. وقتی گذار اجرا می‌شود، از هر مکان ورودی آن گذر به تعداد کمان‌های مابین نشانه کم می‌شود و به هر مکان خروجی به اندازه‌ی کمان‌های مابین زیاد می-شود. این قوانین اجرا ساختار شبکه را نشان می‌دهند.

تعریف ۳-۵- در یک شبکه‌ی پتری نشانه‌گذاری شده‌ی $M = (N,\mu)$ گذار $t_j \in T$ فعال است اگر و فقط اگر به ازای همه‌ی مکان‌های ورودی p_i داشته باشیم : $\mu(p_i) \geq F(p_i, t_j)$.

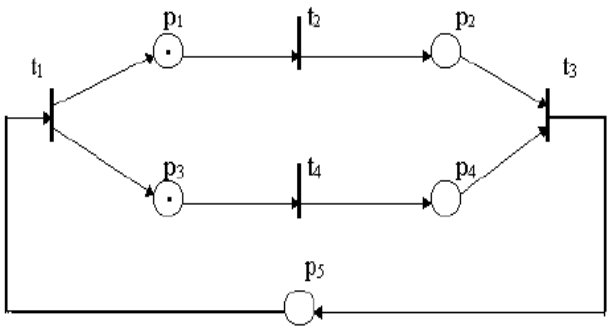
۳-۲- ویژگی‌های اصلی و ساختاری شبکه پتری

با اینکه شبکه پتری از مفاهیم ساختاری کمی تشکیل شده است. اما این مفاهیم می‌توانند سیستم‌های پیچیده را مدل‌سازی کنند. از جمله‌ی این مفاهیم ساده می‌توان به همزمانی[۳]، عدم قطعیت[۱] و توالی[۲] اشاره کرد[۳۴-۳۶].

[۱] Infinite capacity
[۲] Firing rule
[۳] concurrency

۳-۳-۱- همزمانی

فعالیت‌های همزمانی (یا موازی) در یک سیستم را می‌توان با گذارهای همزمان مدل کرد. دو گذار مستقل زمانی همزمان هستند که، همزمان فعال باشند. بعبارت دیگر، دو گذار فعال زمانی همزمان هستند که بتوانند به هر ترتیبی اجرا شوند. ویژگی همزمانی می‌تواند بسیاری از رفتارهای سیستم را مدل کند. در عمل، طراحی و پیاده‌سازی رخدادهای همزمان در یک سیستم از اهمیت ویژه‌ای برخوردار است. در یک سیستم اگر دو فرایند یا رخداد بتوانند بصورت همزمان اتفاق افتد، تاثیر زیادی در صرفه‌جویی هزینه دارد. شکل (۳-۳) دو گذار همزمان را نشان می‌دهد.

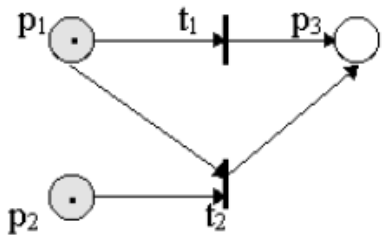


شکل (۳-۳) یک شبکه پتری که در آن گذار t_2 و t_4 در حالت (۱،۰،۱،۰،۰) همزمان هستند

دو گذار فعال ناسازگار[۳] نامیده می‌شوند اگر نتوانند همزمان با هم اجرا شوند. بعبارتی دیگر دو گذار فعال زمانی ناسازگار هستند که اجرای یکی از آن‌ها باعث غیر فعال کردن گذار دیگر شود[۳۷]. این ویژگی در شکل (۳-۴) نشان داده شده است. در این شکل گذار t_1 و t_2 ناسازگارند.

---

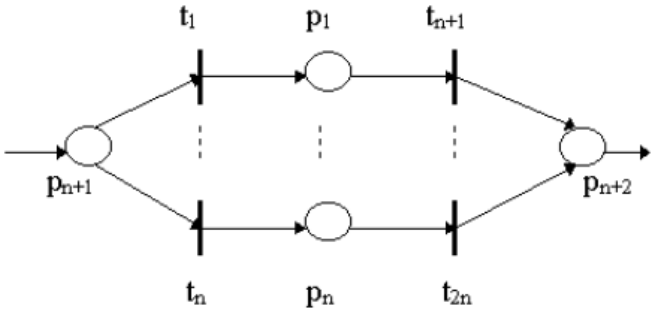
[۱] Non- determination
[۲] sequence
[۳] conflict



شکل(۳-۴) یک شبکه پتری با ویژگی ناسازگاری

۳-۳-۲- عدم قطعیت

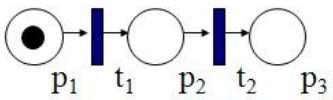
نوع دیگر ویژگی همزمانی، عدم قطعیت (انتخاب) می‌باشد. در این ویژگی گذارهای فعال برای اجرا شدن با یکدیگر رقابت می‌کنند. ویژگی عدم قطعیت با مفهوم همزمانی ارتباط نزدیکی دارد. وقتی یک شبکه‌ی پتری اجرا می‌شود، رفتارهای پویای سیستم بصورت دنباله‌ای از رخدادها قابل مشاهده است. در دنباله رخدادهای اجرا شده، گذارهای فعال به هر ترتیبی می‌توان اجرا شوند. بعبارتی دیگر اگر در یک لحظه، بیش از یک گذار فعال باشد، آنگاه هر یک از این گذارهای فعال می‌توانند اجرا شوند. بطور کلی، این انتخاب بصورت تصادفی است و انتخاب گذار در دنباله با توزیع احتمال یکنواخت صورت می‌گیرد. شکل (۳-۵) ساختار انتخاب را در شبکه‌ی پتری نشان می‌دهد.



شبکه (۳-۵) ساختار همزمانی انتخاب در شبکه پتری

۳-۳-۳- توالی

ویژگی توالی، یکی از عمومی‌ترین و پرکاربردترین ویژگی‌های شبکه‌ی پتری است. این ویژگی دارای مفهومی است که در آن ترتیب اجرا شدن گذارها نشان داده می‌شود. بعبارتی دیگر، برخلاف ویژگی عدم قطعیت، گذارها در یک تر تیب خاصی اجرا می‌شوند. بعنوان مثال در شکل (۳-۶)، گذار t_2 فقط بعد از گذار t_1 می‌تواند انجام شود. همان‌طور که در شکل (۳-۶) مشاهده می‌شود، در یک توالی، گذارها و مکان‌ها بصورت یکی در میان ظاهر می‌شوند.

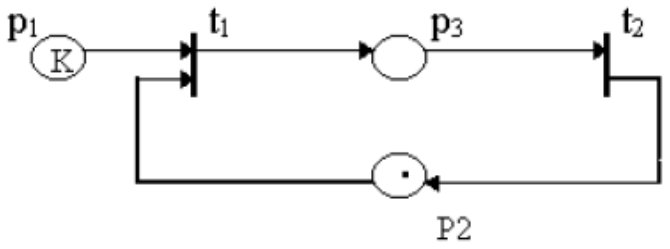


شکل (۳-۶) ساختار یک شبکه پتری با ویژگی توالی

ساختار توالی در یک شبکه‌ی پتری می‌تواند بصورت $\{t_1, t_2, \dots, t_n\}$ و $\{p_1, p_2, \dots, p_n\}$ نشان داده شود.

۳-۳-۴- ساختار مدار

گاهی اوقات شبکه پتری دارای ساختار حلقه است. ساختار حلقه زمانی اتفاق می‌افتد که در یک فرایند، تعدادی گذار به تعداد دفعات نامحدودی تکرار شوند. ساختار مدار، رفتار چرخه‌ای منابع غیر مصرفی را مدل می‌کند. برای مثال فرض کنید که یک ربات از یک منبع برای بارگذاری و تخلیه داده استفاده می‌کند. مدل مربوطه در شکل (۳-۷) قابل مشاهده است. نشانه‌گذاری اولیه‌ی این شبکه پتری بصورت $\mu=(k,1,0)$ است. این نشانه‌گذاری نشان می‌دهد که ربات دارای k داده است. وقتی گذار t_1 اجرا می‌شود، سیستم یک داده را مصرف می‌کند و به حالت $\mu=(k-1,0,1)$ می‌رود. تا زمانی که ربات داده را نگه می‌دارد، گذار t_1 غیر فعال است. وقتی که گذار t_2 اجرا می‌شود، سیستم به حالت $\mu=(k-1,1,0)$ می‌رود. این حالت مانند حالت اولیه است، با این تفاوت که یک داده مصرف شده است.



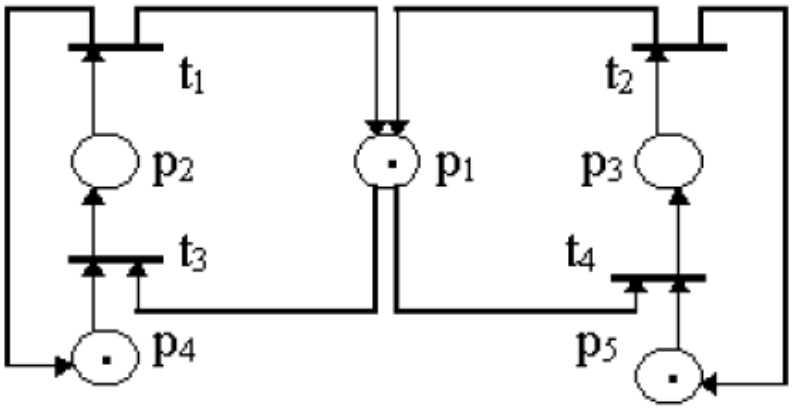
شکل (۳-۷)یک ساختار مدار با $\mu_0=(k,1,0)$

۳-۴- انحصار متقابل[1]

ناسازگاری دو جانبه نوعی مکانیزم همگام‌سازی محسوب می‌شود. این ویژگی موقعیت‌هایی را مدل می‌کند که بین چند فرایند منابعی مشترک باشند. با توجه به مشترک بودن منابع، انحصار متقابل از دسترسی همزمان چند فرایند به این منابع جلوگیری می‌کند. انحصار متقابل باعث جلوگیری از ایجاد بن بست در سیستم‌های توزیع شده می‌شود.

مثالی از انحصار متقابل را در شکل (۳-۸) مشاهده می‌کنید. در این شکل، مکان‌های p۱، p۲ و p۵ وضعیت منابع را نشان می‌دهند. وجود یک نشانه در مکان p۱ نشان دهنده‌ی در دسترس بودن این منبع است. مکان‌های p۲ و p۳ فعالیت‌هایی را نشان می‌دهد که از این منبع استفاده می‌کنند. در این شبکه، گذارهای t۳ و t۴ فعال هستند، اما فقط یکی از آن‌ها می‌تواند اجرا شود. وقتی که یکی از گذارهای t۳ یا t۴ اجرا می‌شود، مکان p۱ نشانه خود را از دست می‌دهد. بنابراین گذار دیگر غیر فعال می‌شود. بطور مثال، اجرا گذار t۳ باعث می‌شود که از مکان‌های p۱ و p۴ یک نشانه حذف شود و یک نشانه به مکان p۲ افزوده شود. تا زمانی که فعالیت p۲ در حال اجراست، فعالیت p۳ نمی‌تواند به منبع اطلاعاتی دست پیدا کند. با اتمام فعالیت p۲ گذار t۱ اجرا می‌شود و نشانه از مکان p۲ حذف می‌شود ویک نشانه به مکان‌های p۱ و p۴ افزوده می‌شود. حال، منبع مشترک در دسترس است و می‌تواند توسط p۳ یا p۲ مورد استفاده قرار گیرد.

[1] Mutual exclusion



شکل (۳-۸) مثالی از انحصار متقابل در شبکه‌ی پتری

شبکه پتری در بسیاری از کاربردها، از ترکیبی از ویژگی‌های بیان شده، تشکیل شده است[۳۸]. ترکیب این ساختارها، مدل شبکه پتری ایستا را ایجاد می‌کند و پویایی سیستم توسط قوانین اجرا و حرکت نشانه‌ها در شبکه پتری ایجاد می‌شود[۳۹،۴۰].

۳-۵- ویژگی‌های رفتاری شبکه‌ی پتری

یکی از مزایای شبکه‌ی پتری، وجود ابزاری برای آنالیز و بررسی ویژگی‌های رفتاری سیستم مدل شده توسط شبکه است [۳۶]. وقتی که یک سیستم توسط شبکه‌ی پتری مدل می‌شود، مدل ساخته شده باید مورد بررسی قرار گیرد و مشخص شود که آیا این مدل بدرستی رفتار سیستم را نشان می‌دهد یا نه. بررسی ویژگی‌های رفتاری سیستم، این اطلاعات را در بر دارد. در ادامه به بررسی چند ویژگی رفتاری مهم در شبکه‌ی پتری پرداخته می‌شود.

یکی از ویژگی‌های رفتاری مهم شبکه‌ی پتری، دسترسی[۱] است. اساس این ویژگی مربوط به پویایی رفتار شبکه‌ی پتری است. اجرا شدن گذارهای فعال موجب تغییر حالت در شبکه‌ی پتری می‌شود. به مجموعه همه‌ی حالت‌های یک شبکه‌ی پتری که از حالت اولیه قابل دسترسی هستند را فضای حالت شبکه‌ی پتری می‌گویند.

تعریف ۳-۶- حالات قابل دسترسی: فرض کنید که N یک شبکه پتری نشانه گذاری شده با حالت اولیه‌ی S. باشد. حالت S یک حالت قابل دسترس از حالت .S نامیده می‌شود، اگر و تنها اگر دنباله‌ای ازحالات فعال موجود باشد که با اجرا آن، شبکه از حالت .S به حالت S برود.

ویژگی رفتاری مهم دیگر شبکه‌ی پتری محدود به k[۲]بودن شبکه‌ی پتری است.

تعریف ۳-۷- یک مکان در یک شبکه پتری محدود به k نامیده می‌شود اگر در همه‌ی حالات قابل دسترسی از حالت اولیه، شامل حداکثر k نشانه باشد. همچنین این مکان امن[۳] نامیده می‌شود اگر در آن k=۱ باشد.

یک شبکه پتری نشانه‌گذاری شده محدود به k نامیده می‌شود اگر همه‌ی مکان‌های آن محدود به k باشند و همچنین یک شبکه پتری نشانه‌گذاری شده امن نامیده می‌شود اگر همه‌ی مکان‌های آن امن باشند. امن بودن نوع خاصی از محدود به k بودن است که در آن k=۱.

یکی دیگر از ویژگی‌های رفتاری مهم دیگر در شبکه‌ی پتری زنده بودن شبکه پتری است. مفهوم این ویژگی مربوط به وجود بن بست[۴] در سیستم برای تخصیص منابع است. در شبکه‌ی پتری، بن بست به مجموعه گذار-هایی گفته می‌شود که نمی توانند اجرا شوند. گذاری که در بن بست نباشد زنده[۵] نامیده می‌شود.

---

۱ Reachability
۲ K-bounded
۳ safe
۴ deadlock
۵ Live

تعریف ۳-۸- یک شبکه‌ی پتری زنده نامیده می‌شود اگر برای هر گذر t_j، بتوان بدون توجه به حالت اولیه، از حالت فعلی به حالتی که شامل t_j هست، انتقال یافت.

یک شبکه‌ی پتری زنده تضمین می‌کند که در همه دنباله اجراها، هیچ بن‌بستی وجود ندارد [۲۲]. در ادامه به تعریف چند ویژگی رفتاری دیگر از شبکه پتری می‌پردازیم.

تعریف ۳-۹- شبکه پتری متصل: فرض کنید $N = (P, T, F, V)$ یک شبکه پتری باشد N متصل[۱] (ضعیف) نامیده می‌شود اگر به ازای هر دو نود x و y از شبکه، داشته باشیم: $x(FUF^{-1})^*y$. که R^{-1} رابطه معکوس و R^* بستار انتقالی و بازتابی رابطه R را نشان می‌دهد. همچنین N متصل قوی نامیده می‌شود اگر و تنها اگر به ازای هر دو نود x و y از شبکه، داشته باشیم: $x(F)^*y$.

۳-۶- ساختار بلوکی DED ها

شبکه‌های پتری رخدادهای پیاپی، تصمیم‌گیری (عدم قطعیت) و همزمانی (موازی بودن) را توصیف می‌کند[۲۳]. رفتارهای پویا در سیستم فرایندها را بوجود می‌آورند. ساختارهای اولیه‌ی همچون AND-join ،همگام‌سازی[۲]، ناسازگاری، ساختارهای بلوکی هستند که برای همگام‌سازی سیستم‌های توزیع شده بکار می‌روند.

همگام‌سازی تضمین می‌کند که به منابع عمومی در یک سیستم دستیابی بطور صحیح صورت می‌گیرد. بعنوان مثال یک سناریو اتوماتیک صنعتی را در نظر بگیرید. فرض کنید یک فعالیت (گذار t') برای شروع نیاز به یک ماشین ویژه و مواد اولیه دارد. نحوه‌ی مدل کردن اتفاق در شکل(۳-۹ (a)) نشان داده شده است. وجود نشانه در مکان t' نشان دهنده‌ی آماده بودن مواد اولیه است و در این صورت گذار t' منتظر می‌ماند که منبع

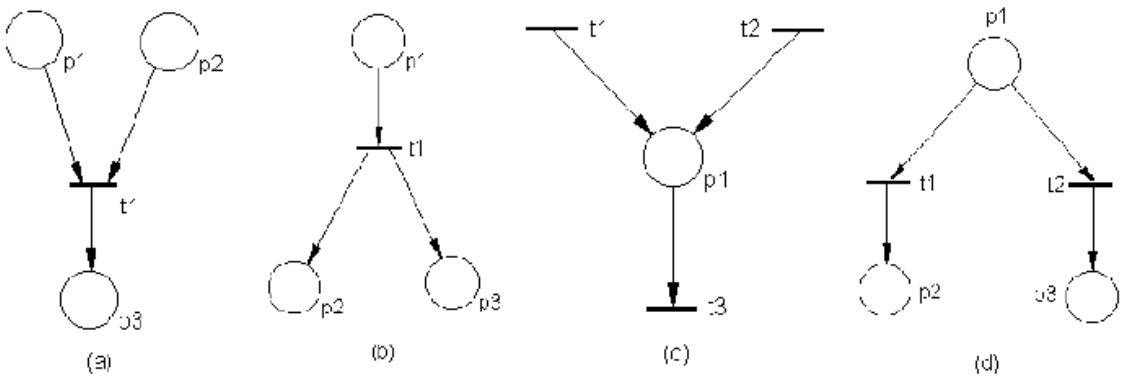
[۱] connected
[۲] synchronisation

دیگر نیز(که مکان p_2 آن را نشان می‌دهد) در دسترس باشد. وقتی در هر دو مکان p' و p۲ نشانه قرار گرفت گذار t' فعال می‌شود.

ساختار AND (یا چنگال) وضعیتی را مدل می‌کند که یک رخداد در یک موقعیت اجرا می‌شود و دو یا چند سیگنال کنترلی یا اطلاعات خروجی را ایجاد می‌کند. یک ساختار AND در شکل (۳-۹ (b)) نشان داده شده است. فرض کنید کامل شدن یک عمل، رخدادی باشد که توسط گذار t مدل شده است. آنگاه بعد از اجرا گذارt دو نوع اطلاعات تولید می‌شود، که با وجود نشانه در مکان‌های خروجی نشان داده می‌شود. این اطلاعات برای عمل‌های بعدی می‌تواند مورد استفاده قرار می‌گیرد.

یک ساختار join، چندین نوع اطلاعات یا سیگنال کنترلی را در یک وضعیت عمومی ترکیب می‌کند. ساختار یک join را در شکل (۳-۹ (c)) نشان داده شده است. مکان p۱ دو کمان ورودی و یک کمان خروجی دارد. فرض کنید یک منبع بصورت عمومی بین چند فرایند مورد استفاده قرار می‌گیرد. این منبع زمانی آزاد می‌شود که فرایندها دیگر نیازی به آن نداشته باشند (فرایندها یک نشانه در مکان p۱ قرار می‌دهند).

ساختار انتخاب، رفتارهای گوناگون ممکن را در یک سیستم نشان می‌دهد. بطور مثال فرض کنید در یک کارخانه برای انجام کاری، باید انتخابی از بین ماشین‌ها صورت گیرد. ساختار انتخاب این موقعیت را مدل می‌کند. این ساختار در شکل (۳-۹ (d)) نشان داده شده است. وجود نشانه در مکان p۱ باعث فعال شدن گذارهای t۱ و t۲ می‌شود. اما فقط یکی از آن‌ها می‌تواند اجرا شود.



شکل (۳-۹) ساختار اولیه: a- همگامسازی، b- AND، c- join، d- انتخاب

۳-۷- شبکه‌های پتری گسترش یافته

با وجود تمام نقاط قوت شبکه پتری کلاسیک دارای ضعف‌های بسیاری در موقعیت‌های عملی است و نمی‌توان با آن بسیاری از فعالیت‌های عملی را مدل‌سازی نمود. به همین دلیل شبکه پتری در جهت‌های مختلف گسترش یافته است که مهمترین توسعه‌های شبکه‌های پتری عبارتند از: ۱- شبکه‌ی پتری رنگی. ۲- شبکه‌ی پتری زمانی. ۳- شبکه‌ی پتری سلسله مراتبی. در ادامه به طور کوتاه به مروری بر این شبکه‌ها پرداخته می‌شود.

۳-۷-۱- شبکه‌ی پتری رنگی

درشبکه‌ی پتری رنگی، بعد جدیدی به نشانه‌ها افزوده می‌شود و به این ترتیب نشانه‌ها دارای صفت شده و دامنه‌ی انتخاب اجراها، بیشتر می‌شود. در شبکه‌های پتری رنگی نشانه‌ها، نماینده توابع متفاوتی هستند. بعنوان مثال می‌توان از نشانه‌های متفاوتی برای نمایش فراخوانی‌های سیستم عامل و یا رده‌های متفاوت کارها استفاده کرد. به کمک این نشانه‌های مختلف، می‌توان مشخص کرد که از میان چندین گذار فعال، کدام یک از آنها می‌توانند اجرا شوند. در اینجا منظور از رنگ یک صفت ویژه است که می‌تواند به یک نشانه اختصاص داده شود و آن را از بقیه نشانه‌ها متمایزکند.

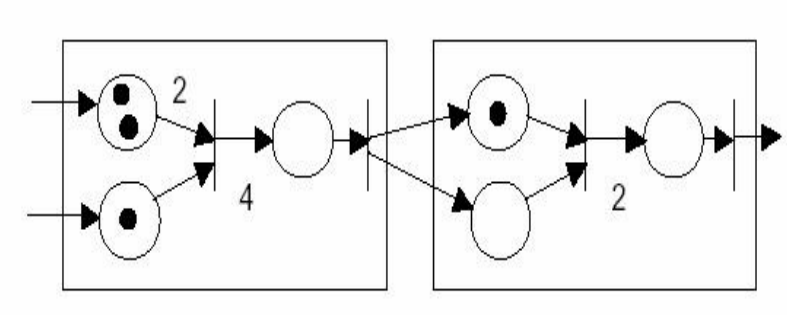
۳-۷-۲- شبکه پتری زمانی[۱] (TPN)

شبکه پتری زمانی همانند شبکه پتری است که برای هر گذر یک مشخصه زمان در نظر گرفته می‌شود [۴۱]. در شبکه پتری زمانی، هنگامی که یک گذار می‌خواهد اجرا شود، زمان بررسی خواهد شد. قوانین اجرا کردن در این مدل به این صورت است که گذار فعال، فقط در یک محدوده‌ی خاص می‌تواند اجرا می شود [۴۴-۴۲].

۳-۷-۳- شبکه پتری سلسله مراتبی[۲](PPN)

شبکه‌ی پتری سلسله مراتبی در اواخر دهه ۱۹۸۰ توسعه پیدا کرده است. همواره مشخصه‌ها و ویژگی‌هـا در یک سیستم واقعی رو به پیچیده و گسترده شدن هستند. ویژگی این شبکه باعث جداسازی، ساختارهای مختلف است. با جدا سازی ساختارهای مختلف شبکه، بررسی و اصلاح مدل آسانتر صورت خواهد گرفت. هر ساختار در این شبکه « ساب نت[۳] » نامیده می‌شود [۴۵] .

هر ساب نت، به وسیله جعبه مستطیلی شکل که در پردازنده بخشی از مدل شبکه پتری می‌باشد، نشان داده می‌شود. در شکل (۳-۱۰) یک شبکه پتری سلسله مراتبی را با دو ساب نت مشاهده می‌کنید.



شکل (۳-۱۰) ساختار یک شبکه‌ی پتری سلسله مراتبی

۳-۸- کاربرد شبکه پتری

[۱] Time petri net
[۲] Prioritized Petri nets
[۳] Sub Net

ساخت یک مدل از روی سیستم‌های صنعتی اساسی‌ترین گام در کاربر شبکه‌ی پتری است[۴۸-۴۶]. هر چند نیازی نیست که مدل همه‌ی جزییات سیستم را در بر داشته باشد، اما باید بتواند رفتارهای اساسی سیستم و ارتباطات بین آن‌ها پوشش دهد. قبل از مدل کردن سیستم، طراح باید فرایندهای صورت گرفته در این سیستم را کاملا متوجه شود. برای مدل کردن یک سیستم توسط شبکه پتری باید به مقوله‌های زیر دقت داشت:

۱- نوع سیستمی که قرار است مدل شود. همچون پروتکل‌های ارتباطی، طراحی سخت‌افزار، آنالیز نرم‌افزار، سیستم اتوماتیک کارخانه، کنترل ناظر و غیره.

۲- نوع شبکه پتری مورد استفاده[۴۹] . همچون شبکه پتری ساده، شبکه پتری زمانی، شبکه‌ی پتری رنگی، شبکه پتری فازی و غیره.

۳- محدوده‌ی کاربرد[۵۰]. یعنی این‌که شبکه پتری برای چه کاری مورد استفاده قرار می‌گیرد. همچون کنترل، زمانبندی کنترل کیفیت، بررسی کارایی و غیره.

۳-۹- نتیجه‌گیری

در این فصل شبکه‌ی پتری بعنوان یکی از ابزارهای مدل‌سازی معرفی گردید و ویژگی‌ها، رفتارها و ساختار شبکه پتری بیان شد. شبکه پتری، ابزاری گرافیکی جهت توصیف رسمی جریان فعالیت‌ها در سیستم‌های پیچیده است که یک مدل آسنکرون از سیستم ایجاد می‌کند. شبکه پتری در بسیاری از کاربردها، از ترکیبی از ویژگی-هایی همچون همزمانی، عدم قطعیت، توالی، ساختار مدار و انحصار متقابل تشکیل شده است. ترکیب این ساختارها، مدل شبکه پتری ایستا را ایجاد می‌کند و پویایی سیستم توسط قوانین اجرا و حرکت نشانه‌ها در شبکه پتری ایجاد می‌شود.

در ادامه شبکه‌های پتری گسترش یافته بیان شد. درشبکه‌ی پتری رنگی، بعد جدیدی به نشانه‌ها افزوده می‌شود و شبکه پتری زمانی همانند شبکه پتری است که برای هر گذر یک مشخصه زمان در نظر گرفته می-

شود. شبکه پتری سلسله مراتبی باعث جداسازی ساختارهای مختلف در شبکه می‌شود. با جدا سازی ساختار-های مختلف شبکه، بررسی و اصلاح مدل آسانتر صورت خواهد گرفت.

ساخت یک مدل از روی سیستم‌های صنعتی اساسی‌ترین گام در کاربرد شبکه‌ی پتری است. مدل طراحی شده برای یک سیستم باید بتواند تمام جنبه‌های اصلی سیستم را نشان دهد. وقتی که یک سناریو واقعی مدل می‌شود، باید تمام جزییات لازم بتواند بر مدل نگاشت شود. هر چند ابزارهای مدل‌سازی شاید پیچیده بنظر برسند، اما این ابزارها بسیار جامع هستند و استفاده از آن‌ها، آنالیز سیستم را آسان می‌کند.

فصل چهارم

فرایندکاوی

۴-۱- مقدمه

در چند دهه اخیر مفهوم و تکنولوژی سیستم‌های مدیریت جریان‌کار [۱] در بسیاری از سیستم‌های اطلاعاتی بکار رفته است. سیستم مدیریت جریان‌کار، یک مدل کلی را برای فرایندهای تجاری نشان می‌دهد. در هنگام استفاده از از تکنولوژی سیستم مدیریت جریان‌کاری به مشکلات زیادی پیش می‌آید. یکی از این مشکلات، نیازمند بودن سیستم به یک طرح جریان‌کاری است. به طور مثال طراح باید بتواند یک مدل با جزییات دقیقی طراحی کند، که این مدل باید جریان‌کار در این سیستم را توصیف کند. طراح باید دانش کاملی در مورد زبان جریان‌کاری و ارتباطات مدیریتی در سیستم داشته باشد. قبل از طراحی مدل مدیریت جریان‌کاری، باید اطلاعاتی در مورد فرایند جریان کاری که در سیستم اتفاق می‌افتد، را بدست آوریم.

فرض می‌شود رخدادهایی که در یک سیستم اتفاق می افتد را می‌توان طوری ذخیره سازی کرد، که:

۱- هر رخداد مربوط به یک فعالیت در سیستم باشد.

۲- هر رخداد دریک دنباله[۱] رخداد می‌افتد. (نمونه ای از یک جریان‌کار)

۳- رخداد ها به ترتیب اجرا می‌شوند. (دنباله رخداد[۲])

به طور کلی می‌توان فرایند کاوی را به شکل زیر توصیف کرد[۵۱]:

"اساس فرایند کاوی، استخراج دانش از دنباله ای از رخدادهاست، که این رخدادها از سیستم اطلاعاتی بدست می‌آیند. تا چند وقت پیش، اطلاعات موجود در دنباله رخداد، به ندرت برای آنالیز کردن فرایند ها به کار

[۱] case
[۲] Event log

می‌رفت. هدف فرایند کاوی، مهیا کردن ابزار و تکنیک‌هایی است، که بتوان فرایندها، داده‌ها، کنترل و ساختار اجتماعی یک سیستم را از دنباله‌ای از رخدادها استخراج نمود. "

۴-۲- ثبت رخداد

سیستم‌های اطلاعاتی که معمولا توسط سازمان‌ها مورد استفاده قرار می‌گیرند، فعالیت‌های خود در فرایند-های اجرایی را ثبت‌رخداد ثبت می‌کنند. هنگامی که یک فرایند شروع بکار می‌کند، یک مثالی از فرایند، راه اندازی می‌شود، که نمونه فرایند[۱] نامیده می‌شود. یک نمونه فرایندی شامل دنباله‌ی از رخدادهاست که در آن نمونه فرایند اجرا می‌شوند. هر رخداد مربوط به یک فعالیت مشخصی در طول اجرای فرایند می‌باشد. علاوه بر این رخدادها شامل صفت‌هایی همچون نشان-زمان[۲] و منبع[۳] می‌باشد. صفت زمان-نشان زمانی که رخداد اجرا می‌شود را نشان می‌دهد و صفت منبع نشان می‌دهد که این رخداد مربوط به سیستم است یا کاربر. در دنباله‌ی رخداد، رخدادها بر اساس صفت نشان-زمان مرتب می‌شوند. دنباله رخداد، معمولا به عنوان نقطه شروع در زمینه‌ی تحقیقاتی فرایند کاوی به‌کار می‌رود[۵۲].

ثبت‌رخداد بعنوان منبع اصلی داده در فرایند کاوی به‌کار می‌رود، که گاهی اوقات آن را " دنباله بازبینی[۴] " نیز می‌نامند. تعریف دقیق ثبت‌رخداد به صورت زیر می‌باشد[۵۳]:

"دنباله‌ای از فعالیت‌های یک سیستم کامپیوتری که به صورت فایل در سیستم ذخیره می‌شود. این فایل می-تواند توسط مدیر بازبینی شود و فعالیت کاربران و فرایندها در سیستم شناسایی شوند."

تعریف ۴-۱- ثبت‌رخداد کامل –ثبت رخدادی که‌دنباله‌های آن قابل نمایش باشند و همچنین تعداد آن‌ها به اندازه ای باشد که بتواند تمام رفتار سیستم را پوشش دهد، را دنباله رخداد کامل می‌نامند.

---

[۱] Process instance
[۲] timestamp
[۳] resource
[۴] Audit trail

گاهی اوقات مجموعه‌ای از دنباله رخداد، ثبت‌رخداد نامیده می‌شود. در جدول(۴-۱) نمونه ای از ثبت رخداد را مشاهده می‌کنید.

جدول(۴-۱) نمونه ای از ثبت‌رخداد

case identifier	task identifier
case 1	task A
case 2	task A
case 3	task A
case 3	task B
case 1	task B
case 1	task C
case 2	task C
case 4	task A
case 2	task B
case 2	task D
case 5	task A
case 4	task C
case 1	task D
case 3	task C
case 3	task D
case 4	task B
case 5	task E
case 5	task D
case 4	task D

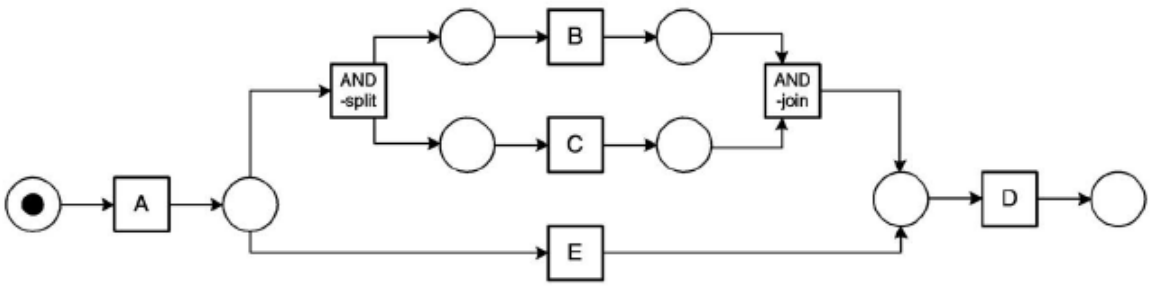
فرایندکاوی حداقل به دو دلیل می‌تواند مفید باشد. اول اینکه، فرایندکاوی می‌تواند بعنوان ابزاری برای شناسایی و ردیابی کارهای کاربران و فعالیت‌های یک سیستم بکار رود. بعبارتی دیگر کارکرد فعالیت‌های مختلف فرایندها را در سیستم بدست آورد. گاهی اوقات امکان دارد همه اطلاعات مربوط به یک سیستم ثبت شود، اما فرایندهای فرعی (که جزیی از فرایند اصلی در نظر گرفته می‌شود) به طور کلی، نادیده گرفته شوند. بنابراین سیستم مدیریت اطلاعاتی باید مبنای کار خود را فرایندهای اصلی همچون استفاده از منابع و جریان زمان قرار

دهد. دوم این‌که فرایندکاوی میتواند به‌عنوان ابزاری برای آنالیز دلتا[1] به کار رود [۵۴]. (به طور مثال مقایسه فرایند واقعی با بعضی از فرایندهای از قبل تعریف شده)

۴-۳- اصول فرایندکاوی

فرض کنید اطلاعات جدول (۴-۱)، را از یک سیستم اطلاعتی بدست آورده ایم. در بسیاری از برنامه‌های کاربردی، ثبت‌رخدادها شامل صفت نشان-زمان می‌باشد. برای توضیح اصول فرایندکاوی، ثبت رخداد جدول (۴-۱) را در نظر بگیرید. این ثبت‌ها شامل اطلاعاتی از پنج دنباله است. ثبت‌ها برای چهار دنباله نشان داده شده‌اند (۱ و۲ و۳ و ۴) و همچنین چهار فعالیت A و B و C و D در سیستم اجرا شده‌اند. برای دنباله‌ی پنجم تنها ۳ فعالیت اجرا می‌شوند: A و E وD. هر دنباله جریان‌کار با اجرای رخداد A شروع می‌شود و با اجرای رخداد D به پایان می‌رسد. در برخی از دنباله‌ها فعالیت B بعد از فعالیت C اجرا می‌شود و در بعضی دنباله‌ها فعالیت C بعد از فعالیت B اجرا می‌شود.

با توجه به اطلاعاتی که در جدول (۴-۱) نشان داده شده است و با این فرض که این ثبت رخداد کامل باشند، می‌توان یک مدل فرایندی را همچون شکل (۴-۱) بدست آورد. مدل فرایندی نشان داده شده در شکل (۴-۱) نوعی از یک شبکه پتری می‌باشد.



شکل (۴-۱) مدل فرایندی استخراج شده از جدول (۴-۱)

[1] Delta analysis

همانطور که در شکل مشاهده می‌کنید این شبکه پتری با فعالیت A شروع و با فعالیت D به پایان می‌رسد. در شبکه پتری فعالیت‌ها توسط گذارها[۱] نشان داده می‌شود. بعد از اجرای فعالیت A یا فعالیت E اجرا می‌شود یا فعالیت B و C به‌طور موازی[۲] اجرا می‌شود. برای اجرای موازی دو فعالیت B و C، باید فعالیت‌هایی را بعنوان فعالیت‌های پنهان(AND-split و AND-join) به شبکه پتری زیاد کنیم. این فعالیت‌های پنهان به منظور هماهنگ‌سازی بین دیگر فعالیت‌ها به کار می‌رود و جزء فعالیت‌های جریان کار سیستم نمی‌باشند. باید دقت شود که دو فعالیت در صورتی موازی در نظر گرفته می‌شود که به هر ترتیبی بتوانند اجرا شوند. با توجه به این-که برای هر رخداد، یک زمان شروع و یک زمان پایان در نظر گرفته می‌شود، می‌توان فعالیت‌ها را به صورت اجرای موازی در نظر گرفت. همچنین زمان شروع و پایان یک ثبت را می‌توان بعنوان معیاری برای زمان اجرای یک فعالیت در نظر گرفت. در عین حال، در فرایندکاوی، برای سادگی کار، فرض می‌شود که فعالیت‌ها به صورت یک‌پارچه[۳] هستند.

به‌طور کلی می‌توان گفت که ایده‌ی اصلی فرایندکاوی، ساخت یک مدل فرایندی همچون شکل (۴-۱) از ثبت رخدادهایی همچون جدول (۴-۱) است.

ساخت مدل فرایندی ساده (همچون مدل فرایندی شکل (۴-۱)) از ثبت رخداد آسان می‌باشد. اما برای مدل فرایندی پیچیده‌تر کار بسیار مشکل‌تر می‌باشد. به طور مثال اگر فعالیت دیگری با فرایند B موازی باشد. آنگاه ثبت رخداد جدول (۴-۱)، تمام رفتار سیستم ما را نشان نمی‌دهد. فرض کنید ۱۰ فعالیت به طور موازی می-توانند اجرا شوند. تعداد کل حالت‌هایی که امکان دارد اجرا شود (تعداد دنباله رخدادها) !۱۰ می‌باشد. ثبت رخدادی با این تعداد دنباله، واقع بینانه نیست و هزینه‌های زیادی برای ما در بر دارد. علاوه بر این، احتمال این-که بعضی از مسیرهای خاص در مدل فرایندی دنبال شود، کم می‌باشد و امکان دارد شناسایی نشود. همچنین

[۱] transitions
[۲] parallel
[۳] atomic

داده‌های نویزی (ثبت رخدادی که به طور اشتباهی ثبت شده است و یا همچنین ثبت‌هایی که شامل رخدادهایی هستند که این رخدادها به ندرت اتفاق می‌افتند) می‌تواند در آینده فرایند کاوی را دچار مشکل کند.

۴-۴- شبکه‌های جریان کار

اکثر سیستم‌های جریان‌کار، از ساختار بلوکی[۱] استانداردی همچون And-join، And-split، OR-split و OR-join تشکیل شده‌اند[۵۸-۵۵].

یک شبکه پتری می‌تواند بعنوان ابزاری برای مسیریابی دنباله‌ها به کار رود. فعالیت‌ها توسط گذار ها مدل می‌شوند و ارتباطات منطقی بین گذار ها توسط مکان و کمان بیان می‌شوند. در واقع می‌توان گفت که یک مکان یک وضعیت[۲] را مشخص می‌کند. این وضعیت می‌تواند نشان دهنده‌ی شروط لازم برای اجرای یک فعالیت باشد و یا این‌که نشان‌دهنده‌ی حالت ایجاد شده بعد از اجرای یک فعالیت باشد. یک And-split گذاری را نشان می‌دهد که بیشتر از یک خروجی به مکان‌ها دارد و And-join به گذاری گفته می‌شود که بیشتر از یک ورودی از مکان‌ها دارد. OR-split به مکانی گفته می‌شود که بیشتر از یک کمان خروجی دارد و OR-join به کمانی گفته می‌شود که بیشتر از یک کمان ورودی دارد.

به شبکه پتری که به مدل کردن جریان در یک جریان کار بپردازد شبکه جریان کار[۳] نامیده می‌شود.

تعریف ۴-۲- شبکه‌های جریان‌کار: فرض کنید N=(P,T,F) یک P/T باشد و $\bar{t}$ یک متغیر جدید باشد که در PUT وجود ندارد. N یک شبکه جریان کار نامیده می‌شود اگر:

۱- مورد ایجاد: P شامل یک مکان ورودی i شود که در آن: •i=ø

۲- مورد اتمام: P شامل یک مکان خروجی O شود که در آن: O•=ø

[۱] Building Block
[۲] Condition
[۳] Workflow net (WF-net)

۳- نوع ارتباط: $\bar{N}$= (P, T ∪ {$\bar{t}$},F∪{(o, $\bar{t}$),($\bar{t}$, i)}) یک ارتباط قوی باشد.

در شکل (۴-۱)، یک WF-net را مشاهده می‌شود. باید دقت شود که این شبکه، یک شبکه‌ی متصل قوی نیست اما شبکه‌ی $\bar{N}$= (P, T ∪ {$\bar{t}$},F∪{(o, $\bar{t}$),($\bar{t}$, i)}) یک شبکه‌ی متصل قوی می‌باشد.

تعریف ۴-۳- شبکه‌ی جریان کار خوش ساخت[۱]: فرض کنید N=(P,T,F) یک WF-net باشد. این شبکه خوش ساخت است اگر و فقط اگر: ۱- امن باشد. ۲- زنده باشد. ۳- هر حالت قابل دسترسی که شامل مکان خروجی است، نباید شامل مکانی دیگر شود. ۴- بتوان از هر حالت قابل دسترس، به حالتی که شامل مکان خروجی است، با یک دنباله اجرا انتقال یافت.

۴-۵- مساله اکتشاف[۲]

بطور کلی می‌توان گفت هدف از فرایندکاوی، یافتن یک مدل جریان‌کاری(مانند WF-net) بر اساس ثبت-رخدادها می‌باشد. باید دقت شود که ترتیب رخدادها در یک دنباله از اهمیت ویژه‌ای بر خوردار است، در صورتی که ترتیب دنباله ها در یک ثبت‌رخداد اهمیتی ندارد. تعریف دقیق ثبت‌رخداد به صورت زیر می‌باشد.

تعریف ۴-۴- فرض کنید T مجموعه‌ای از رخدادها باشد.$\sigma \in T^*$ یک دنباله جریان‌کاری است و $W \in p(T^*)$ یک ثبت‌رخداد می‌باشد. (p یک مجموعه توانی از T^* می‌باشد).

در جدول (۴-۱)، case۱، یک دنباله جریان‌کاری است که ABCD می‌باشد و ثبت‌رخداد جدول (۴-۱) عبارت است از:

{ABCD, ACBD, AED}

[۱] Sound Wf-net
[۲] rediscover

در فرایند کاوی (در برخی از الگوریتم‌ها) تعداد تکرار یک دنباله اهمیتی ندارد. بطور مثال در جدول (۴-۱)، دنباله جریان‌کاری ABCD دوبار اتفاق افتاده است(در case ۱ و case ۳)، دنباله جریان‌کار ACBD، دوبار اتفاق افتاده است(در case ۲ و case ۴) و دنباله جریان‌کار AED تنها یکبار اتفاق افتاده است (case ۵). تکرار در ثبت-رخدادها تاثیری ندارد. بنابراین می‌توان نتیجه گرفت که ثبت‌رخداد جدول (۴-۱)، بصورت {ABCD, ACBD, AED} می‌باشد.

برای یافتن یک مدل جریان‌کار بر اساس ثبت‌رخداد، ابتدا باید به ارتباطات منطقی بین رخدادها پرداخته شود. بطور مثال، هرگاه یک رخداد همواره بعد از یک رخداد دیگر اتفاق افتد. احتمالا یک ارتباط منطقی بین این دو رخداد وجود دارد. برای آنالیز کردن این ارتباطات، تعریف زیر در نظر گرفته می‌شود.

تعریف ۴-۵- ارتباط ترتیبی بین رخدادها: فرض کنید که W یک ثبت‌رخداد بر روی T باشد ($W \in p(T^*)$) و همچنین $a, b \in T$ آن‌گاه:

- $a >_w b$ اگر و فقط اگر یک دنباله جریان کار $\sigma = t_1,t_2,t_3,...,t_{n-1}$، $\sigma \in W$، یک $i \in \{1,2,3..,n-1\}$ موجود باشد که در آن $t_i=a$ و $t_{i+1}=b$.
- $a \rightarrow_w b$ اگر و فقط اگر $a >_w b$ و $b \tilde{>}_w a$.
- $a \#_w b$ اگر و فقط اگر $a \tilde{>}_w b$ و $b \tilde{>}_w a$.
- $a \|_W b$ اگر و فقط اگر $a >_w b$ و $b >_w a$.

بطور مثال ثبت‌رخداد W={ABCD, ACBD, AED} را در نظر بگیرید (ثبت‌رخداد جدول (۴-۱)). ارتباط $>_w$ بصورت $A >_w B$، $A >_w C$، $A >_w E$، $B >_w C$، $B >_w D$، $C >_w B$، $C >_w D$ و $E >_w D$ است و همچنین ارتباط $\rightarrow_w$ که از ارتباط $>_w$ نتیجه می‌شود و ارتباط منطقی[۱] نام دارد، شامل $A \rightarrow_w B$، $A \rightarrow_w C$، $A \rightarrow_w E$،

[۱] Causal relation

$B \rightarrow_w D$، $C \rightarrow_w D$ و همچنین $E \rightarrow_w D$ می‌باشد. باید دقت شود که $B \sim \rightarrow_w C$ چون $C >_w B$. ارتباط $\|_w$ برای رخدادهای موازی در نظر گرفته شده است. در ثبت‌رخداد W، رخدادهای B و C موازی هستند، بنابراین $B \|_W C$ و همچنین $C \|_W B$. اگر دو رخداد با هر ترتیبی بتوانند پشت سر هم قرار گیرند، آن دو رخداد موازی در نظر گرفته می‌شود. ارتباط $\#_w$ دو گذری را نشان می‌دهد که هیچگاه پشت سر هم، بصورت مستقیم، اجرا نشده‌اند. به عبارت دیگر، هیچ ارتباط مستقیم یا موازی بین آن رخدادها نیست.

برای ساده‌سازی استفاده از ثبت‌رخدادها تعریف زیر را در نظر می‌گیریم.

تعریف ۴-۶- عضویت ($\in$)، اولین (first)، آخرین (last) : فرض کنید A یک مجموعه از فعالیت‌ها باشد و همچنین دنباله‌ی $\sigma = a_1a_2a_3\ldots a_n \in A^*$ یک دنباله بر روی مجموعه A به طول n باشد. عضویت ($\in$)، اولین (first)، آخرین (last) بصورت زیر تعریف می‌شوند.

۱- $a \in \sigma$ اگر و تنها اگر $a \in \{a_1,a_2,a_3,\ldots,a_n\}$،

۲- $first(\sigma)=a_1$ اگر $n>1$ و

۳- $last(\sigma)=a_n$ اگر $n>1$

برای اینکه یک الگوریتم اکتشاف جریان‌کار بدرستی کار کند، باید ثبت رخداد کامل باشد. بعبارتی دیگر در یک فرایند پیچیده، حتی تعداد زیادی از دنباله جریان‌کار نمی‌تواند، رفتار یک فرایند را نشان دهد. بطور کلی کامل بودن یک ثبت رخداد بصورت زیر تعریف می‌شود.

تعریف ۴-۷- ثبت رخداد کامل: فرض کنید که $N=(P,T,F)$ یک WF-net خوش ساخت باشد. W یک ثبت رخداد از N است اگر و فقط اگر $W \in p(T^*)$ و هر دنباله $\sigma \in W$ یک دنباله اجرا از N است که در حالت i شروع و در حالت o پایان می‌یابد. W یک ثبت رخداد کامل از N است اگر وفقط اگر برای هر ثبت رخداد W'

از $N : >_{w'}$ زیر مجموعه‌ای از $>_w$ باشد و به ازای هر $t \in T$ یک دنباله $\sigma \in W$ موجود باشد که در آن داشته باشیم، $t \in \sigma$.

یک ثبت‌رخداد از یک WF-net خوش ساخت تنها شامل رفتارهایی می‌شود که توسط فرایند مربوط قابل نمایش باشد. یک ثبت رخداد کامل است اگر به ازای هر فعالیت‌هایی که در شبکه بعد از فعالیت دیگری قرار می‌گیرد، در ثبت‌رخداد نیز دنباله جریان‌کاری نیز موجود باشد که این دو رخداد پشت سر هم قرار گیرند. باید به این نکته دقت داشت که گذری که مکان ورودی i را به مکان خروجی o در یک WF-net متصل می‌کند، برای $>_w$ پنهان است. بنابراین با بیان ویژگی زیر، این نکته را در نظر می‌گیریم.

ویژگی ۴-۱- فرض کنید که شبکه $N=(P,T,F)$ یک WF-net خوش ساخت باشد. اگر W یک ثبت‌رخداد کامل از N باشد آنگاه:

$$\{ t \in T \mid \exists_{t' \in T} t >_w t' \vee t' >_w t \} = \{ t \in T \mid t \notin i\bullet \cap \bullet o \}.$$

اثبات در [۴] قابل مشاهده است.

تعریف کامل بودن ثبت‌رخداد (تعریف ۴-۷)، شاید اختیاری بنظر برسد، اما اینگونه نیست. همچنین نمی‌توان انتظار داشت تمام سری اجراهای ممکن در ثبت‌رخداد قابل نمایش باشد. اولا امکان دارد بی نهایت سری ممکن داشته باشیم (بطور مثال وجود یک حلقه در شبکه). دوما امکان دارد فرایندهای موازی شامل تعداد نمایی از حالت‌ها باشد. بنابراین امکان دارد تعداد سری اجراهای ممکن، بسیار زیاد شود.

تعریف ۴-۸- توانایی بازسازی[۱]: فرض کنید که شبکه $N=(P,T,F)$ یک WF-net خوش ساخت باشد و α یک الگوریتم اکتشاف[۱] باشد که با استفاده از یک ثبت‌رخداد از N، یک WF-net خوش ساخت می‌سازد. اگر برای هر ثبت‌رخداد کامل W از N، الگوریتم اکتشاف بتواند N را بسازد، α می‌تواند N را بازسازی کند.

[۱] Rediscover

باید دقت داشت که هیچ الگوریتم اکتشاف نمی‌تواند نام مکان‌ها را پیدا کند. بطور مثال α می‌تواند N را بازسازی کند، اگر و تنها اگر $\alpha(W) = N$، تغییر نام مکان‌ها را انجام دهد. هیچ الگوریتم اکتشافی وجود ندارد که بتواند تمام WF-net های خوش ساخت را بسازد. بطور مثال اگر در شکل (۴-۱)، مکان P بین گذارهای A و D زیاد شود، هیچ الگوریتم اکتشاف نمی‌تواند P را شناسایی کند. در صورتی که این مکان در شبکه وجود دارد. مکان‌هایی که رفتار سیستم را تغییر نمی‌دهند، مکان‌های ضمنی[۲] گفته می‌شوند. بطور کلی به مکان‌های ضمنی نمی‌توانند در ثبت‌رخدادها ظاهر شوند.

۴-۶- اکتشاف جریان‌کار

قبل از اینکه الگوریتم اکتشاف شبکه جریان‌کار معرفی شود، ابتدا به معرفی ارتباط منطقی بین رخدادها در ثبت رخداد پرداخته می‌شود. ابتدا در یک قضیه نشان داده خواهد شد که ارتباط منطقی $\rightarrow_W$ نشان‌دهنده حضور مکان است. سپس دسته شبکه‌هایی که قابل بازسازی هستند، توضیح داده خواهد شد.

اگر بین دو گذر در یک ثبت رخداد ارتباط منطقی وجود داشته باشد، بین آن دو گذر یک مکان وجود خواهد داشت. که این دو گذار را به یکدیگر متصل می‌کند.

قضیه ۴-۱- فرض کنید که شبکه $N=(P,T,F)$ یک WF-net خوش ساخت باشد و W یک ثبت رخداد کامل از N باشد. برای هر $a, b \in T$ که $a \rightarrow_W b$ نشان دهنده‌ی $a\bullet \cap \bullet b \neq \emptyset$ است.

اثبات در [۴] است.

برای مثال شبکه زیر را در نظر بگیرید:

---

[۱] Mining algorithm
[۲] implicit

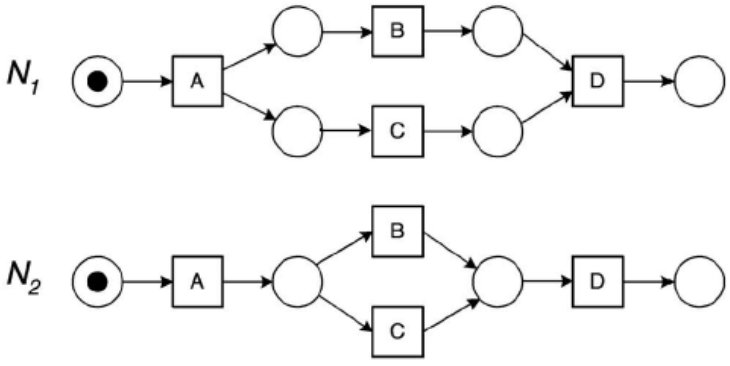
N_1 =({i, p_1, p_2, p_3, p_4, o}, {A, B,C,D}, {(i , A), (A , p_1), (A , p_2), (p_1 , B), (B , p_3), (p_2, C), (C , p_4), (p_3 , D), (p_4 , D) , (D , o)})

این شبکه در شکل (۴-۲) (N_1) قابل مشاهده است. در این شبکه دو فعالیت B و C با هم موازی هستند (همزمان اجرا می‌شود). فرض کنید ثبت‌رخداد کامل W_1={ABCD, ACBD} روی N_1 تعریف شده است.ارتباط منطقی A $\rightarrow_{w1}$ B نشان‌دهنده وجود یک مکان بین A و B است، که این مکان مربوط به p_1 از شبکه N_1 است.

در مثالی دیگر شبکه زیر را در نظر بگیرید:

N_2 =({i, p_1, p_2, o}, {A, B,C,D}, {(i , A), (A , p_1), (p_1 , B), (B , p_2), (p_1, C), (C , p_2), (p_2 , D), (D , o)})

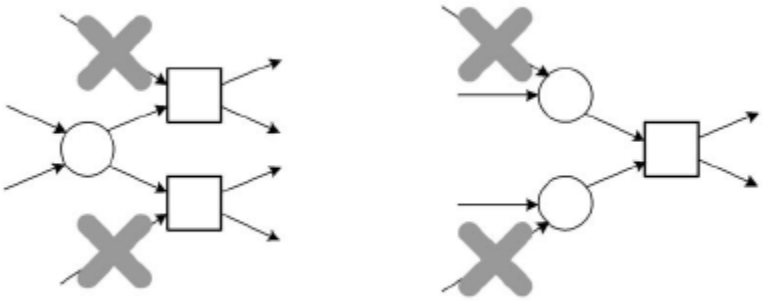
در این WF-net، یک انتخاب بین B و C وجود دارد. می‌توان این شبکه درشکل (۴-۲) (N_2) مشاهده کرد. فرض کنید ثبت‌رخداد کامل W_1={ACD, ABD} روی N_2 تعریف شده است. A $\rightarrow_{w2}$ B نشان‌دهنده وجود یک مکان بین A و B است و همچنین A $\rightarrow_{w1}$ C نشان‌دهنده وجود یک مکان بین A و C است. هر دو مکان-ها مربوط به p_1 از شبکه N_2 است. در مثال اول، دو ارتباط منطقی A $\rightarrow_{w1}$ B و A $\rightarrow_{w1}$ C مربوط به دو مکان مختلف می‌باشند، در صورتی که در صورتی که در مثال دوم دو ارتباط منطقی A $\rightarrow_{w2}$ B و A $\rightarrow_{w2}$ C مربوط به یک مکان می‌باشد.



شکل (۴-۲) دو مثال از WF-net

۴-۶-۱- مکان‌های متصل و ارتباط منطقی

در بازسازی یک WF-net، همه‌ی مکان‌ها قابل شناسایی نیستند. بطور مثال، یک مکـان امکـان دارد ضـمنی باشد. همانطور گفته شد به مکانی ضمنی گفته می‌شود که تاثیری در رفتار یک سیستم نداشته باشد. این مکان-ها را نمی‌توان شناسایی کرد. بنابراین در مساله‌ی بازسازی، WF-netها بدون مکان ضمنی در نظـر گرفتـه مـی-شوند. در شکل (۴-۱)، هیچ مکان ضمنی وجود ندارد. البته همانطور که بیان شد، اگر مکان P بین گذارهای A و D زیاد شود، این مکان ضمنی است و این مکان در هیچ ثبت رخدادی ظاهر نمی‌شود.



شکل (۴-۳) دو ساختاری که در SWF-net نمی‌باشد

در بازسازی یک WF-net، مساله ای که از اهمیت ویژه‌ای برخوردار است ساختاریست کـه در WF-net بکـار رفته است. این ساختار باید بتواند بطور واضح نشان‌دهنده‌ی رفتارش باشد. بنابراینWF-net باید فاقـد بعضـی از ساختارها، مانند آنچه که در شکل (۴-۳) نشان داده شده است، باشد. قسمت چپ شـکل، نشـان مـی‌دهـد کـه انتخاب و همزمانی نباید اتفاق افتد. اگر دو گذر یک مکان ورودی داشته باشند، باید برای بدست آوردن نشانه بـا یکدیگر رقابت کنند. بنابراین نیازی به همزمانی ندارند. قسمت راسـت شـکل (۴-۳) نشـان مـی‌دهـد کـه وقتـی همزمانی انخاب می‌شود، همه‌ی گذرهای ماقبل آن باید اجرا شوند. بعبارتی دیگر، هیچگاه همزمانی نباید بعـد از

OR-join اتفاق افتد. WF-net ای که چنین ویژگی‌هایی داشته باشد شبکه جریان کار ساخت یافته[1] نامیده می-شود.

تعریف ۴-۹- (SWF-net). فرض کنید که N=(P,T,F) یک WF-net باشد. این شبکه یک SWF-net (شبکه جریان کار ساخت یافته) است، اگر و تنها اگر:

۱- برای همه‌ی $p \in P$ و $t \in T$ که $(p,t) \in F$ داشته باشیم: $|p\bullet| > ۱$ نشان دهنده‌ی $|\bullet t| = ۱$ باشد.

۲- برای همه‌ی $p \in P$ و $t \in T$ که $(p,t) \in F$ داشته باشیم: $|\bullet t| > ۱$ نشان دهنده‌ی $|\bullet p| = ۱$ باشد.

۳- هیچ مکان ضمنی وجود نداشته باشد.

در نگاه اول شاید تعریف ۴-۹ بسیار محدود کننده بنظر برسد. اما در واقع این‌گونه نیست. SWF-net تمام ساختارهای مهم و اساسی را در WF-net همچون توالی، شرایط موازی بودن، حلقه و ساختار های بلوکی اساسی (And-join, OR-join, And-split, OR-split) را شامل می‌شود. بطور کلی اگر شرایط تعریف ۴-۹ رعایت نشود، ارتباط $>_w$ اطلاعات کافی را برای اکتشاف همه‌ی فرایندها ندارد.

SWF-net همچنین ویژگی‌های جالب دیگری نیز دارند.

ویژگی ۴-۲- فرض کنید که N=(P,T,F) یک SWF-net باشد. برای هر $a, b \in T$ و $p_۱, p_۲ \in P$ داریم: اگر $p_۱ \in a\bullet \cap \bullet b$ و $p_۲ \in a\bullet \cap \bullet b$ آنگاه $p_۱ = p_۲$.

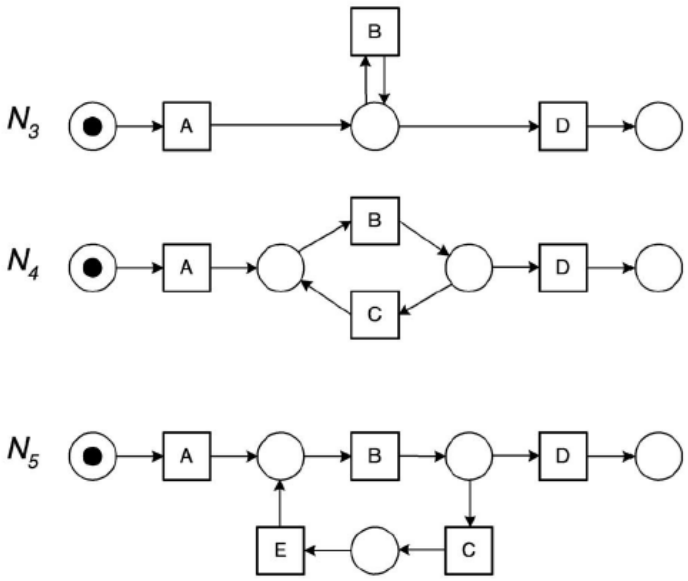
این ویژگی مستقیما از تعریف ۴-۹ بدست می‌آید و نشان می‌دهد هیچ دو گذری نباید توسط چند مکان به یکدیگر متصل باشند. این ویژگی نشان می‌دهد ساختار SWF-net رفتارش را بدرستی نشان می‌دهد.

[1] Structured workflow net

قضیه ۴-۲- فرض کنید N=(P,T,F) یک SWF-net خوش ساخت باشد و W یک ثبت‌رخداد کامل برای N باشد. برای هر $a, b \in T$ که $a\bullet \cap \bullet b \neq \emptyset$ آنگاه $a>_w b$.

اثبات این قضیه در [۵۲] می باشد.

باید به ای مورد دقت داشت که $a\bullet \cap \bullet b \neq \emptyset$ نمی‌تواند دلالت بر $a \rightarrow_w b$ داشته باشد. بطور مثال در شبکه-های نشان داده شده در شکل (۴-۲)، دو رخداد به هم متصلند اگر و تنها اگر با هم ارتباط منطقی داشته باشند. اما این موضوع در مورد شبکه‌های N_3 و N_4 در شکل (۴-۴) صدق نمی‌کند. اما در شبکه N_3، ارتباطات $A \rightarrow_{w3} B$، $A \rightarrow_{w3} D$، $B \rightarrow_{w3} D$ وجود دارد، در صورتی که ارتباط $B \rightarrow_{w3} B$ بر قرار نیست. با وجود این که مکانی وجود دارد که B را به B متصل کند. در شبکه‌ی N_4 اگرچه مکانی وجود دارد که B را به C متصل می‌کند و یا مکانی هست که C را به B متصل می‌کند، اما $B\sim \rightarrow_{w3} C$ و $C\sim \rightarrow_{w3} B$. این مثال نشان می‌دهد حلقه‌هایی بطول ۱ (شبکه N_3) و بطول ۲ (شبکه N_4) نمی‌توانندزیاد مفید واقع شوند.



شکل (۴-۴) سه مثال از WF-net

قضیه ۴-۳- فرض کنید N=(P,T,F) یک SWF-net خوش ساخت باشد و W یک ثبت‌رخداد کامل برای N باشد. برای هر $a, b \in T$ که $a\bullet \cap \bullet b \neq \emptyset$ و $b\bullet \cap \bullet a = \emptyset$ آنگاه $a \rightarrow_w b$.

اثبات این قضیه در [۴] می‌باشد.

بنابراین شبکه باید فاقد حلقه‌هایی بطول یک یا دو باشد. با توجه به آنچه گفته شد ویژگی زیر را می‌توان بدست آورد.

ویژگی ۴-۳- فرض کنید N=(P,T,F) یک SWF-net خوش ساخت باشد و W یک ثبت رخداد کامل برای N باشد. برای هر $a, b \in T : a\bullet \cap \bullet b \neq \emptyset$ اگر و تنها اگر $a \rightarrow_w b$.

در قضیه ۴-۳ ارتباط بین مکان‌های متصل و ارتباط منطقی نشان داده شد. اما ارتباط منطقی تنها یکی از چهار ارتباط ترتیبی بیان شده در تعریف ۴-۵ است. در قضیه بعدی ارتباط بین اشتراک‌گذاری مکان‌های ورودی/ خروجی و $\#_w$ بیان خواهد شد.

قضیه ۴-۴- فرض کنید N=(P,T,F) یک SWF-net خوش ساخت باشد که در آن برای هر $a, b \in T$ داشته باشیم: $a\bullet \cap \bullet b = \emptyset$ یا $b\bullet \cap \bullet a = \emptyset$ و W یک ثبت رخداد کامل برای N باشد. آنگاه:

۱- اگر $a, b \in T$ و $a\bullet \cap b \bullet \neq \emptyset$ آنگاه $a \#_w b$

۲- اگر $a, b \in T$ و $a \bullet \cap \bullet b \neq \emptyset$ آنگاه $a \#_w b$

۳- اگر $a, b, t \in T$ و $a \rightarrow_w t$ و $b \rightarrow_w t$ و $a \#_w b$ آنگاه $a\bullet \cap b\bullet \cap \bullet t \neq \emptyset$

۴- اگر $a, b, t \in T$ و $t \rightarrow_w a$ و $t \rightarrow_w b$ و $a \#_w b$ آنگاه $t\bullet \cap \bullet a \cap \bullet b \neq \emptyset$

همچنین در یک SWF-net خوش ساخت که در آن حلقه های کوتاه (به اندازه‌ی یـک و دو) نیسـت، $a \Vert_W b$ نشان‌دهنده‌ی $a \bullet \cap b \bullet = \bullet a \cap \bullet b = \emptyset$ است. علاوه بر این $a \rightarrow_w t$ و $b \rightarrow_w t$ و $t \bullet \cap \bullet b \cap \bullet a = \emptyset$ نشان‌دهنده‌ی $a \Vert_W b$ است و همچنین $t \rightarrow_w a$ و $t \rightarrow_w b$ و $t \bullet \cap \bullet b \cap \bullet a = \emptyset$ نشان‌دهنده‌ی $a \Vert_W b$ است.

۴-۶-۲- الگوریتم اکتشاف آلفا[۱]

با توجه به آنچه در قسمت‌های پیش بیان شد، در این قسمت به معرفی یک الگوریتم اکتشاف پرداخته مـی-شود. این الگوریتم بر این واقعیت بنا شده است که در همه‌ی WF-net ها دو رخداد به هم متصل هستند اگـر و تنها اگر این ارتباط در ثبت‌رخدادها قابل شناسایی باشد.

تعریف ۴-۱۰- الگوریتم اکتشاف آلفا: فرض کنید که W یک ثبت‌رخداد تعریـف شـده بـر روی T باشـد. $\alpha(W)$ بصورت زیر تعریف می‌شود:

۱- $T_w = \{t \in T \mid \exists_{\sigma \in W} t \in \sigma\}$

۲- $T_I = \{t \in T \mid \exists_{\sigma \in W} t = first(\sigma)\}$

۳- $T_O = \{t \in T \mid \exists_{\sigma \in W} t = last(\sigma)\}$

۴- $X_W = \{(A,B) \mid A \subseteq T_w \wedge B \subseteq T_w \wedge \forall_{a \in A} \forall_{b \in B} a \rightarrow b \wedge \forall_{a_1, a_2 \in A} a_1 \#_w a_2 \wedge \forall_{b_1, b_2 \in A} b_1 \#_w b_2\}$

۵- $Y_W = \{(A,B) \in X_W \mid \forall_{(A', B') \in X_W} A \subseteq A' \wedge B \subseteq B' \Rightarrow (A,B) = (A',B')\}$

۶- $P_W = \{p(A,B) \mid (A,B) \in Y_W\} \cup \{i_w, i_o\}$

۷- $F_W = \{(a, p(A,B)) \mid (A,B) \in Y_W \wedge a \in A\} \cup \{(p(A,B), b) \mid (A,B) \in Y_W \wedge b \in B\}$

$\cup \{(i_w, t) \mid t \in T_I\} \cup \{(t, o_w) \mid t \in T_o\}$

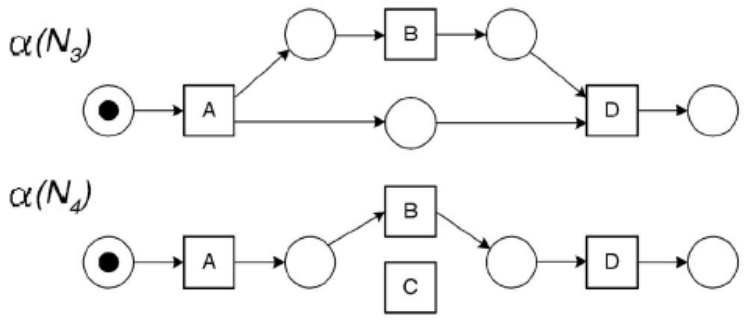
۸- $\alpha(W) = (P_W, T_W, F_W)$

[۱] mining algorithm α

این الگوریتم اکتشاف، شبکه‌ی (P_W,T_W,F_W) را می‌سازد. مجموعه گذار T_W براحتی از روی ثبت‌رخداد بدست می‌آید. در واقع اگر دنباله رخدادی بطول یک نداشته باشیم، T_W می‌تواند از روی $>_W$ بدست آید. با توجه به اینکه می‌توان گذارهای اولیه T_I و گذارهای انتهایی T_O را بدست آورد، براحتی می‌توان اتصال بین این گذارها و i_W و o_W را بازسازی کرد. علاوه بر مکان منبع[۱] i_W و مکان حفره[۲] o_W مکان‌های $p(a,b)$ نیز در نظر گرفته می‌شوند. یک مکان بین a و b زیاد می‌شود اگر و تنها اگر $a \rightarrow_W b$. گاهی اوقات این مکان‌ها برای OR-join و یا OR-split با یکدیگر ادغام می‌شوند. به همین منظور ارتباط بین X_W و Y_W ساخته شده است. $(A, B) \in X_W$ اگر یک ارتباط منطقی بین هر عضو A و هر عضو B وجود داشته باشد و هر یک از اعضای A یا B پشت سر هم اتفاق نیفتند. باید دقت کرد که اگر $a \rightarrow_W b$، $b \rightarrow_W a$ یا $a \parallel_W b$ آنگاه a و b نمی‌توانند در A (یا B) باشند. در ادامه این فصل مثالی از این الگوریتم نشان داده خواهد شد.

حال با توجه به آنچه در تعریف ۴-۱۰ بیان شد، آیا می‌توان WF-net ها را با استفاده از $\alpha(W)$ بازسازی کرد؟ SWF-netهای نشان داده شده در شکل ۴-۲ و ۴-۴ را در نظر بگیرید. اگر α برای یک ثبت‌رخداد کامل از N_1 بکار برده شود، شبکه بدست آمده همان N_1 است که نام مکان‌های آن تغییر کرده است. همچنین اگر α برای یک ثبت‌رخداد کامل از N_2 بکار برده شود، شبکه بدست آمده همان N_2 است که نام مکان‌های آن تغییر کرده است. همانطور که در شکل ۴-۵ نشان داده شده، α نمی‌تواند N_3 و N_4 را بازسازی کند. با حذف اتصالات بین B و A و حذف اتصالات بین B و D، $\alpha(W_3)$ شبیه N_3 می‌شود و همچنین $\alpha(W_4)$ شبیه N_4 است که تنها کمان‌های ورودی و خروجی C حذف شده‌اند. در بازسازی هر یک از این دو شبکه، دو کمان از بین رفته‌اند. شبکه‌های N۳ و N۴ نشان می‌دهد که الگوریتم اکتشاف نمی‌تواند حلقه‌های کوتاه را شناسایی کند. البته حلقه‌هایی بطول سه یا بیشتر نمی‌تواند مشکلی ایجاد کند. برای مثال، اگر α برای یک ثبت‌رخداد کامل از N_5 بکار برده شود، شبکه بدست آمده همان N_5 است که نام مکان‌های آن تغییر کرده است (شکل (۴-۴)).

[۱] Source place
[۲] Sink place



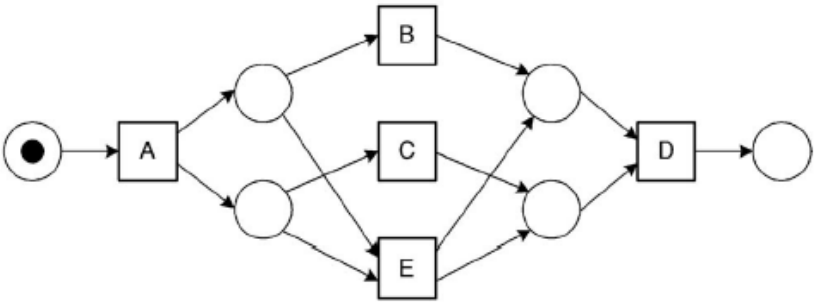
شکل (۴- ۵) شبکه های بدست آمده توسط الگوریتم آلفا

قضیه ۴-۵- فرض کنید N=(P,T,F) یک SWF-net خوش ساخت باشد و W یک ثبت‌رخداد کامل برای N باشد. اگر برای هر $a, b \in T$: $a\bullet \cap \bullet b = \emptyset$ یا $b\bullet \cap \bullet a = \emptyset$ آنگاه $\alpha(W) = N$ (نام مکان‌ها تغییر می‌کند).

اثبات در [۴] است.

این قضیه نشان می‌دهد که الگوریتم آلفا تنها می‌تواند SWF-netهایی را بازسازی کند که حلقه‌ی کوتاه نداشته باشند.

شبکه‌های N_1, N_2 و N_5 شرایط بیان شده در قضیه ۴-۵ را دارند. بنابراین قابل بازسازی هستند و الگوریتم آلفا می‌تواند این شبکه‌ها را بازسازی کند. شبکه‌ی بیان شده در شکل ۴-۱، یک SWF-net است که حلقه کوتاه ندارد. اگر AND-join و AND-split در ثبت‌رخداد قابل مشاهده باشد، الگوریتم آلفا می‌تواند آن را بازسازی کند. اما همانطور که در جدول(۴-۱) نشان داده شده است این دو گذار هیچگاه در سیستم اتفاق نمی‌افتد. بنابراین باید دو گذر AND-join و AND-split باید بصورت پنهان در نظر گرفته شوند. در عین حال، وقتی الگوریتم آلفا بر روی W = {ABCD, ACBD, AED} اجرا می‌شود، شبکه‌ی نشان داده شده در شکل (۴-۶) بدست می‌آید.



شکل (۴-۶) مدل فرایندی مربوط به جدول (۴-۱)

برای توضیح بیشتر الگوریتم آلفا، نتایج بدست آمده در هر قدم از این الگوریتم را بر روی ثبت‌رخداد جدول (۴-۱) ، $W = \{ABCD, ACBD, AED\}$، نشان می‌دهیم:

۱- $T_W=\{A, B, C, D, E\}$

۲- $T_I=\{A\}$

۳- $T_O=\{D\}$

۴- $X_W=\{(\{A\},\{B\}), (\{A\},\{C\}), (\{A\},\{E\}), (\{B\},\{D\}), (\{C\},\{D\}), (\{E\},\{D\}),$
$(\{A\},\{B,E\}), (\{A\},\{C,E\}), (\{B,E\},\{D\}), (\{C,E\},\{D\})\}$

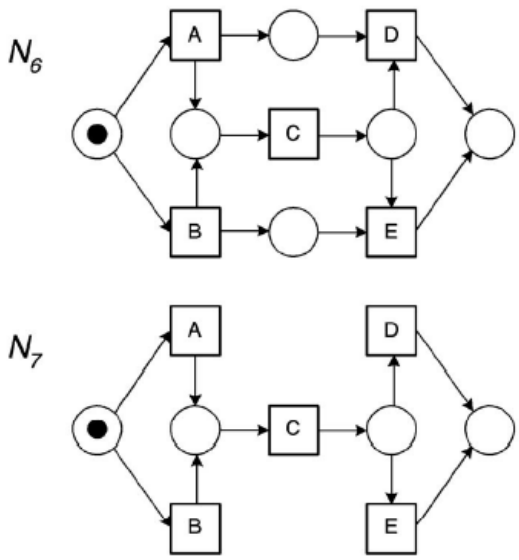
۵- $Y_w=\{(\{A\},\{B,E\}), (\{A\},\{C,E\}), (\{B,E\},\{D\}), (\{C,E\},\{D\})\}$

۶- $P_w=\{i_w, o_w, p(\{A\},\{B,E\}), p(\{A\},\{C,E\}), p(\{B,E\},\{D\}), p(\{C,E\},\{D\})\}$

۷- $F_w = \{(i_w,A), (A, p(\{A\},\{B,E\})), (p(\{A\},\{B,E\}),B), \ldots, (D, o_w)\}$

۸- $\alpha(W) = (P_w, T_w, F_w)$

اگرچه شبکه بدست آمده (شکل(۴-۶)) یک SWF-net نیست، اما یک WF-net خوش ساخت است که رفتاری شبیه رفتار شبکه شکل (۴-۱)را دارد. با این که شبکه بدست آمده SWF-net نیست، اما باز هم قابل بازسازی است. این مثال نشان می‌دهد که الگوریتم آلفا بر روی SWF-net محدود نمی‌شود. البته نمی‌توان تضمین کرد که همه‌ی WF-netهای خوش ساخت قابل بازسازی باشد.

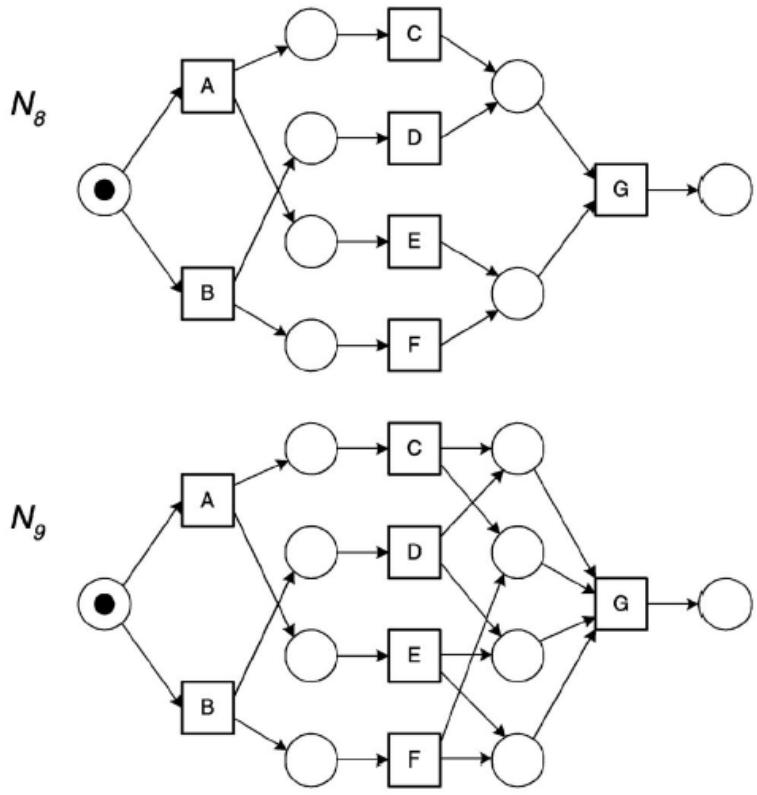


شکل (۴-۷) شبکه‌ی $N_۶$ نمی‌تواند توسط الگوریتم آلفا بازسازی شود

۴-۶-۳- محدودیت‌های الگوریتم الفا

همانطور که در قضیه‌ی ۴-۵ بیان شد، الگوریتم آلفا می‌تواند دسته‌ی بزرگی از فرایندها را بازسازی کند. برای توضیح بیشتر محدودیت‌های ایجاد شده توسط تعریف ۴-۹، شکل (۴-۷) و شکل (۴-۸) را درنظر بگیرید. شبکه $N_۶$، یک WF-net خوش ساخت است اما یک SWF-net نیست، چون شرط اول رعایت نشده است. اگر الگوریتم آلفا بر روی یک ثبت‌رخداد کامل از $N_۶$ اجرا شود ($W_۶$)، شبکه $N_۷$ بدست می‌آید. پس بنابراین می‌توان نتیجه گرفت که $N_۶$ نمی‌تواند توسط الگوریتم آلفا بازسازی شود. اگرچه $N_۷$ یک SWF-net خوش ساخت است اما رفتاری متفاوت از $N_۶$ دارد. بطور مثال دنباله جریان‌کار ACE در $N_۷$ امکان‌پذیر است اما در $N_۶$ امکان‌پذیر نیست.

شکل (۴-۸)، دومین شرط تعریف ۴-۹ را مورد بررسی قرار می‌دهد. همانطور که در شکل (۴-۸) مشاهده می‌کنید $N_۸$، دومین شرط تعریف ۴-۹ را رعایت نمی‌کند. اگر الگوریتم آلفا بر روی یک ثبت‌رخداد کامل از $N_۸$ اجرا شود ($W_۸$)، شبکه $N_۹$ بدست می‌آید. پس بنابراین می‌توان نتیجه گرفت که $N_۸$ نمی‌تواند توسط الگوریتم آلفا بازسازی شود.

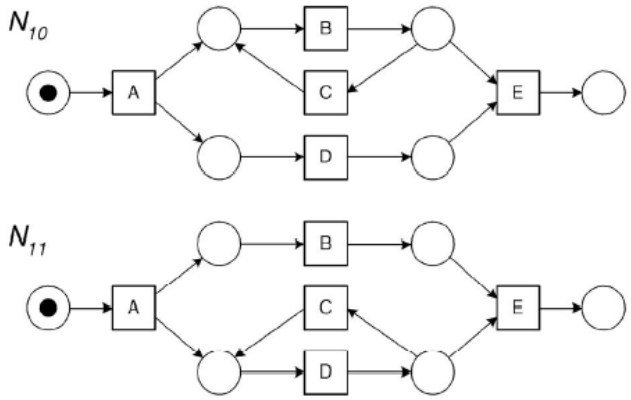


شکل (۴-۸) شبکه‌ی N_8 نمی‌تواند توسط الگوریتم آلفا بازسازی شود

اگرچه شرایط گفته شده در تعریف ۴-۹ برای بازسازی یک شبکه از ثبت‌رخداد، ضروری بنظر می‌رسد، اما الگوریتم آلفا فقط بر روی SWF-netها محدود نمی‌شود. حتی گاهی اوقات، اگر الگوریتم آلفا نتواند شبکه را بطور دقیق بازسازی کند، اما شبکه‌ای را که بعنوان خروجی می‌دهد، از نظر رفتار شبیه شبکه‌ی اصلی است (مانند N_8).

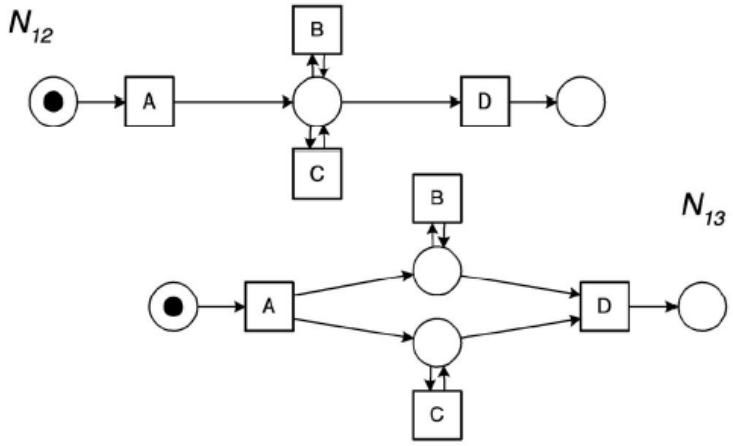
شرایط دیگری که توسط قضیه‌ی ۴-۵ بیان شده، نبودن حلقه‌های کوتاه است. اگرچه می‌توان آلفا الگوریتم را طوری تغییر داد که بتواند این شبکه‌ها را بازسازی کند، اما این کار چندان ساده نیست. برای توضیح بیشتر به شکل (۴-۹) و (۴-۱۰) دقت کنید. در شکل (۴-۹) دو WF-net نشان داده شده است. اگر چه این دو شبکه رفتاری کاملا متفاوت دارند، اما ثبت‌رخداد کامل آن‌ها با توجه به ارتباط $\succ$، یکسان است. همچنین این دو شبکه، با توجه به گذار E، SWF-net نیستند. چون این شبکه‌ها دارای حلقه‌ی کوتاه هستند، الگوریتم آلفا نمی‌تواند

آن‌ها را بدرستی بازسازی کند. درهر ثبت‌رخدادهای N_{10} و N_{11} ارتباطات $B \parallel C, B > C, C > B, C > D, D >$ $C \parallel D$ و $B \parallel D$ برقرار است. این ارتباطات نشان می‌دهد که سه گذر B، C، D با یکدیگر موازی هستند. در عین حال در هر ثبت‌رخداد کاملی، هیچگاه گذر C بعد از A اتفاق نمی‌افتد. با توجه به اینکه ثبت‌رخدادهای کامل این دو شبکه نسبت به < شبیه به هم هستند، هیچ الگوریتم اکتشافی نمی‌تواند بین آن‌ها تمایز قائل شود.



شکل (۴-۹) دو WF-net که دارای رفتاری متفاوت و ثبت‌رخداد کاملی یکسان هستند

شکل (۴-۱۰) مثال دیگری را در همین زمینه نشان می‌دهد. شبکه‌های N_{12} و N_{13} هر دو SWF-net هستند و هر دو دارای رفتاری یکسان هستند اما هیچ الگوریتمی نیست که بین آن‌ها تمایز قائل شود. این مساله نیز مانند شکل (۴-۸) مساله مکان ضمنی را مورد بررسی قرار می‌دهد.



شکل (۴-۱۰) دو SWF-net که دارای رفتار یکسانی هستند اما هیچ الگوریتمی نمی‌تواند بین آن‌ها تمایز قائل شود.

علاوه بر شرایطی که در قضیه ۴-۵ و تعریف ۴-۹ بیان شد، شرایط دیگری همچون رخدادهای پنهان باید در نظر گرفته شود. علاوه براین طبق تعریفی که از شبکه پتری شد، هر گذر برچسب منحصربفردی دارد [۵۹] و همچنین طبق تعریفی که از WF-net شد هر فعالیت تنها یکبار در شبکه حضور پیدا می‌کند. اگر این شرط (هر فعالیت تنها یکبار در شبکه حضور پیدا می‌کند) را در نظر نگیریم، یک فعالیت t چندین بار می‌تواند در شبکه حضور پیدا کند. این مساله رخدادهای تکراری[۱] نام دارد. وجود چنین رخدادهای تکراری نیز باعث می‌شود الگوریتم آلفا نتواند بدرستی عمل کند.

۴-۷- مروری بر الگوریتم‌های اکتشاف

در این قسمت دسته‌های مختلف الگوریتم اکتشاف بیان خواهد شد. در هر دسته فرضیات بر روی ثبت رخداد همچون کامل بودن یک ثبت‌رخداد، بیان خواهد شد.

[۱] Duplicate task

۴-۷-۱- الگوریتم‌های مبتنی بر انتزاع[۱]

در این قسمت الگوریتم‌های مبتنی بر انتزاع بیان خواهد شد. در این الگوریتم‌ها، شبکه بر اساس یک دید انتزاعی از ثبت‌رخداد ساخته می‌شود. اکثر این الگوریتم‌ها از الگوریتم آلفا مشتق شده‌اند، بنابراین الگوریتم‌های سری‌آلفا نامیده می‌شوند. فرایند اکتشاف این الگوریتم‌ها بسیار شبیه هم هستند.

همه‌ی الگوریتم‌های سری آلفا یک‌سری فرضیات بر روی ثبت رخداد در نظر می‌گیرند. بطور کلی سه فرضیه-ی مهمی که در نظر گرفته می‌شود عبارتند از: ۱- هیچ داده‌ی نویزی در ثبت‌رخداد وجود ندارد. به عبارتی دیگر هیچ رخدادی نادیده گرفته نمی‌شود و یا رخدادهای اضافی در ثبت‌رخداد وجود ندارد. ۲- ثبت‌رخداد باید کامل باشد. ۳- مدل فرایندی ایجاد شده باید بتواند بصورت شبکه پتری بیان شود و بعضی از ساختارهای ویژه (مانند حلقه‌های کوتاه، رخدادهای پنهان و ...) را نداشته باشد.

الگوریتم آلفا اولین الگوریتم اکتشاف از این دسته می‌باشد و بقیه‌ی الگوریتم‌های این دسته از این الگوریتم استنتاج می‌شوند. همانطور که بیان شد این الگوریتم SWF-netهایی را بازیابی می‌کند که حلقه‌های کوتاه ندارند و همچنین ثبت‌رخدادها باید کامل باشند.

الگوریتم آلفا پلاس [۲] اولین الگوریتم استنتاج شده از الگوریتم آلفا است. این الگوریتم می‌تواند از حلقه‌های کوتاه در SWF-netهای خوش ساخت پشتیبانی کند [۶]. بنابراین با وجود این الگوریتم شرط نبود حلقه‌های کوتاه از بین می‌رود. همچنین در این الگوریتم علاوه بر اینکه ثبت‌رخداد باید کامل باشد، باید نشان دهنده‌ی حلقه‌های کوتاه نیز باشد(بطور مثال اگر دو گذار $t_۱$ و $t_۲$ در یک حلقه‌ی کوتاه بطول دو حضور داشته باشند آنگاه زیر ثبت‌های "$t_۲, t_۱, t_۲$" و "$t_۱, t_۲, t_۱$" باید حداقل یکبار در ثبت‌رخداد ظاهر شوند).

---

[۱] Abstraction-Based algorithm
[۲] α^+-Algorithm

الگوریتم آلفا - تیسینگهوا[۱] بر روی ثبت‌رخدادی تمرکز می‌کند که دارای رخدادهای غیر یکپارچه [۶۰] است. هر اجرای رخداد t به دو رخداد شروع رخداد و تکمیل رخداد شکسته می‌شود.

الگوریتم آلفا پلاس پلاس[۲] ، الگوریتمی است که ساختار انتخاب غیر آزاد[۳] را تشخیص می‌دهد [۶۱]. الگوریتم آلفا شارپ[۴]، الگوریتمی است که شبکه‌هایی که در گروه SWF-net نیستند را نیز شناسایی می‌کند [۶۲] . این الگوریتم از الگوریتم آلفا پلاس استنتاج می‌شود.

تاکنون مروری بر الگوریتم‌های مبتنی بر انتزاع شد. در عین حال هیچ الگوریتم مبتنی بر انتزاعی نیست که بتواند همه ویژگی های ساختاری SWF-net را پوشش دهد.

۴-۷-۲- الگوریتم‌های مبتنی بر هیوریستیک[۵]

اگرچه الگوریتم‌های بیان شده در قسمت قبل، می‌تواند بسیاری از ساختارهای مدل فرایندی را بازسازی کند، اما بعضی از فاکتورهایی که بطور طبیعی در یک ثبت‌رخداد وجود دارد، نادیده گرفته می‌شوند. یکی از این فاکتورهای مهم وجود نویز است. نویز به دو دلیل می‌تواند ایجاد شود: دنباله رخدادها بصورت اشتباه ثبت شوند یا دنباله رخدادها حالت خاصی از سیستم را نشان دهد. بطور کلی می‌توان گفت که نویز رفتاری است که به ندرت در سیستم اتفاق می‌افتد.

الگوریتم‌های مبتنی بر انتزاع، هیچکدام تعداد تکرار یک ترتیب از ارتباطات را در نظر نمی‌گیرد. در این قسمت الگوریتم‌هایی بیان خواهد شد که تعداد تکرارهای رخدادها را در بازسازی یک مدل فرایندی در نظر می‌گیرد. این الگوریتم‌ها با الگوریتم‌های اکتشاف هیوریستیک مشهورند [۶۳,۶۴].

---

۱ Tsinghua-α-algorithm
۲ α^{++}-algorithm
۳ Non-free-choice
۴ $\alpha^{\#}$-algorithm
۵ Heuristic-Based algorithm

الگوریتم‌های اکتشاف هیورستیک می‌توانند مدل فرایندی را بازسازی کنند که ثبت‌رخداد آن دارای نویز است و رفتار واقعی سیستم را در ثبت‌رخداد شناسایی کند. همچنین این دسته از الگوریتم، تمام ساختارهای عمومی یک مدل فرایندی (همچون حلقه‌ها، رخدادهای پنهان، رخدادهای موازی و...) را بجز رخدادهای تکراری پشتیبانی می‌کند. الگوریتم‌های هیوریستیک دارای دو گام اساسی هستند. در اولین گام گراف وابستگی ساخته می‌شود. این گراف وابستگی شامل وابستگی منطقی بین رخدادهاست. بر عکس الگوریتم‌های مبتنی بر انتزاع، این دسته از الگوریتم‌ها، از تعداد تکرار ارتباط بین رخدادها برای وزن‌دهی یال‌های گراف وابستگی استفاده می‌کنند و سپس با استفاده از این گراف وابستگی رخدادهای مرتبط با یک رخداد خاص را شناسایی می‌کند. در دومین گام، ارتباط معنایی نقاط Split و join در گراف وابستگی شناسایی می‌شوند. به عبارت دیگر با توجه به تعداد تکرار وابستگی‌ها، Split یا join بودن مشخص می‌شود.

۴-۷-۳- الگوریتم‌های مبتنی بر جستجو[۱]

اگرچه استفاده از الگوریتم‌های هیوریستیک می‌تواند مشکل داده‌های نویزی را برطرف کند، اما این دسته از الگوریتم‌ها بر اساس اطلاعات محلی ثبت‌رخداد هستند. به عبارت دیگر، ارتباط بین رخدادها بر اساس پشت سر هم قرار گرفتن در یک دنباله جریان کار است. علاوه بر این الگوریتم‌هایی که تابحال بیان شده، هیچکدام نمی‌توانند همزمان مشکل نویز و پوشش همه ساختارهای عمومی را برطرف کنند. بنابراین الگوریتم‌های فرایندکاوی برای جستجوی کلی[۲] از الگوریتم‌هایی مانند الگوریتم ژنتیک استفاده می‌کنند. این الگوریتم‌ها به الگوریتم‌های اکتشاف ژنتیک (GA) و الگوریتم‌های اکتشاف ژنتیک تکراری[۳] (DGA) مشهورند [۶۵،۶۶]. الگوریتم DGA الگوریتمی است که از روی الگوریتم GA استنتاج می‌شود. این الگوریتم می‌تواند مدل فرایندی که شامل رخداد تکراری هست، را نیز بازسازی کند. در ادامه ابتدا به مفاهیم اساسی الگوریتم ژنتیک پرداخته می‌شود.

[۱] Search-Based algorithm
[۲] Global search
[۳] Duplicates genetic algorithm miner

الگوریتم ژنتیک الگوریتمی است که معمولا برای جستجو در دامنه‌های بزرگ استفاده می‌شود (فضای جستجو زیاد باشد). این الگوریتم با یک جمعیت [۱] از کروموزم‌ها آغاز می‌شود. برای ارزیابی هر کروموزم از یک تابع شایستگی[۲] استفاده می‌شود. در مساله فرایند کاوی هر کروموزم یک مدل فرایندی ممکن است و تابع شایستگی آن، تابعی است که میزان تولید درست ثبت‌رخداد را از مدل فرایندی مشخص می‌کند. در هر مرحله از الگوریتم، جمعیت کروموزم‌ها، توسط عملگرهای ژنتیک تغییر پیدا می‌کند. ابتدا توسط عملگر انتخاب[۳] کروموزم‌هایی انتخاب می‌شوند سپس توسط عملگرهای ترکیب[۴] (ترکیب چند قسمت از کروموزم‌ها) و جهش[۵] (تغییر تصادفی بر روی کروموزم‌ها) کروموزم‌های جمعیت جدید ایجاد می‌شود. نحوه تعریف کروموزم‌ها، تابع شایستگی و عملگرهای ژنتیک سه اصل اساسی در الگوریتم ژنتیک است. در ادامه نحوه‌ی کار GA توضیح داده خواهد شد.

کروموزم‌ها در GA از همه‌ی ساختارها بجز رخداد تکراری پشتیبانی می‌کند. همچنین تابع شایستگی تضمین می‌کند که کروموزم‌هایی که دارای شایستگی بالا هستند، می‌توانند بدرستی اکثر دنباله رخدادها را در ثبت‌رخداد ایجاد کند. همانطور که بیان شد تابع شایستگی باید طوری انتخاب شود که بهترین کروموزم را انتخاب کند. در این مساله، بهترین کروموزم (مدل فرایندی)، کروموزمی است که بتواند اکثر دنباله رخدادها را در ثبت‌رخداد دنبال[۶] کند. عملگرهای ژنتیک بر روی مجموعه‌های ورودی و خروجی رخدادها عمل می‌کند. مجموعه ورودی یک رخداد شامل همه مکان‌های ورودی آن رخداد و همه‌ی رخدادهای ورودی این مکان‌های ورودی است. همچنین مجموعه خروجی یک رخداد شامل همه‌ی مکان‌های خروجی آن رخداد و رخدادهای خروجی آن مکان‌هاست.

---

[۱] population
[۲] fitness
[۳] select
[۴] crossover
[۵] mutation
[۶] parse

این الگوریتم، مدل فرایندی را ایجاد می‌کند که بتواند بیشترین رفتارهای تکراری در ثبت‌رخداد را تولید کند. همچنین پیچیدگی زمانی اجرای الگوریتم نسبت به تعداد رخدادها و تعداد دنباله رخـدادهـای واحـد در ثبـت-رخداد، نمایی است.

۴-۷-۴- الگوریتم‌های اکتشاف حالت

تابحال تمام الگوریتم‌های بیان شده، بر اساس ترتیب رخدادها در ثبت‌رخداد بود و با در نظر گـرفتن ترتیـب رخدادها یک شبکه پتری ایجاد می‌کرد. الگوریتم‌های اکتشاف حالت دیدگاه متفاوتی نسبت بـه مسـاله اکتشـاف دارند. این دسته از الگوریتم‌ها ابتدا هر دنباله رخداد از ثبت‌رخداد را تبدیل به دنباله‌ای از حالت‌هـا و گـذارهـای بین آن‌ها می‌کند. سپس همه‌ی دنباله‌ها با هم ترکیب شده و تشکیل یک سیستم گذار[۱] را می‌دهند و سپس این سیستم گذار تبدیل به یک شبکه پتری می‌شود.

در [۶۷] بطور کامل توضیح داده شده است که چگونه یک ثبت‌رخداد تبدیل به دنبالـه‌ای مبتنـی بـر حالـت می‌شود. می‌توان پسوند هر رخداد را در دنباله رخداد، به عنوان حالت قبل از اجرای رخداد در نظر گرفت. البتـه تکنیک‌های پیشرفته‌ی زیادی وجود دارد که می‌تواند حتی رفتارهای بعد از رخداد را بعنوان حالتی از سیستم در نظر می‌گیرد.

برای ساخت یک شبکه پتری از روی یک سیستم گذار از تئوری ناحیه مبتنی بر حالـت[۲] اسـتفاده مـی‌شـود. بعبارتی دیگر این تئوری شبکه پتری را از روی مدل حالت استنتاج می‌کند[۷۱-۶۸]. البته ایـن تئـوری بـر روی مجموعه محدودی از سیستم‌های گذار، می‌تواند به کار برده شود.

[۱] Transition system
[۲] State-based region theory

۴-۸- نتیجه‌گیری

در این فصل ابتدا به مفهوم فرایندکاوی پرداخته شد و اهمیت آن مدل کردن سیستم بیان شد. مطمئنا اگر نتوان رفتار یک سیستم را بدرستی مدل کرد، نتایجی که از آنالیز آن مدل بدست خواهد آمد دقیق نخواهد بود. سپس ثبت‌رخداد بعنوان یکی از مفاهیم مهم در فرایند کاوی بیان شد. ثبت‌رخداد، معمولا به عنوان نقطه شروع در زمینه‌ی تحقیقاتی فرایند کاوی به‌کار می‌رود. ثبت‌رخداد بعنوان منبع اصلی داده در فرایند کاوی به‌کار می‌رود. سپس با یک مثال (جدول (۴-۱)) اصول فرایند کاوی بیان شد. به‌طور کلی می‌توان گفت که ایده‌ی اصلی فرایندکاوی، ساخت یک مدل فرایندی از ثبت‌ای از رخدادها است. فرایندکاوی حداقل به دو دلیل می‌تواند مفید باشد. اول اینکه، فرایندکاوی می‌تواند بعنوان ابزاری برای شناسایی و ردیابی کارهای کاربران و فعالیت‌های یک سیستم بکار رود. دوم این‌که فرایندکاوی می‌تواند بعنوان ابزاری برای آنالیز دلتا به کار رود. سپس مفهوم شبکه جریان کار بیان شد. به شبکه پتری که به مدل کردن جریان در یک جریان کار بپردازد شبکه جریان کار (WF-net) نامیده می‌شود.

در ادامه مساله اکتشاف بیان شد. در این قسمت به بررسی دسته‌ای از الگوریتم‌های اکتشاف پرداخته شد که دیدگاه اصلی این دسته از الگوریتم‌ها این است که برای یافتن یک مدل جریان‌کار بر اساس ثبت‌رخداد، ابتدا باید به ارتباط منطقی بین رخدادها پرداخته شود. در ادامه برای توصیف یکی از مهمترین الگوریتم‌های فرایند کاوی (الگوریتم آلفا) به تشریح فرضیات و تعاریف لازم پرداخته شد. تعاریفی همچون WF-net، SWF-net، توانایی بازسازی، مکان‌های متصل و ارتباطات منطقی که برای توصیف الگوریتم آلفا ضروری است. در ادامه به توصیف الگوریتم آلفا همراه با یک مثال پرداخته شد و محدودیت‌های بکار رفته بر روی الگوریتم آلفا، همچون نبودن حلقه‌های کوتاه و رخدادهای تکراری، بیان شد.

در ادامه مروری بر همه الگوریتم‌های اکتشاف انجام شد. بطور کلی الگوریتم اکتشاف به چهار دسته تقسیم می‌شوند. اولین دسته، الگوریتم‌های مبتنی بر انتزاع است. در این الگوریتم‌ها، شبکه بر اساس یک دید انتزاعی از ثبت‌رخداد ساخته می‌شود. اکثر این الگوریتم‌ها از الگوریتم آلفا مشتق شده‌اند. دومین دسته الگوریتم‌های مبتنی بر هیوریستیک است. این دسته از الگوریتم‌ها، مساله نویز و همچنین تعداد تکرارهای رخدادها را در بازسازی یک مدل فرایندی در نظر می‌گیرد. سومین دسته از الگوریتم‌ها، الگوریتم‌های مبتنی بر جستجوست. این الگوریتم‌ها که به الگوریتم‌های اکتشاف ژنتیک (GA) و الگوریتم‌های اکتشاف ژنتیک تکراری[۱] (DGA) مشهورند بر اساس الگوریتم ژنتیک بنا شده‌اند. دسته چهارم، الگوریتم‌های اکتشاف حالت است. الگوریتم‌های اکتشاف حالت دیدگاه متفاوتی نسبت به مساله اکتشاف دارند. این دسته از الگوریتم‌ها ابتدا هر دنباله رخداد از ثبت‌رخداد را تبدیل به دنباله‌ای از حالت‌ها و گذارهای بین آن‌ها می‌کند. سپس همه‌ی دنباله‌ها با هم ترکیب شده و تشکیل یک سیستم گذار را می‌دهند و سپس این سیستم گذار تبدیل به یک شبکه پتری می‌شود.

در این فصل انواع الگوریتم‌های فرایند کاوی معرفی شد. اما مساله‌ای که از اهمیت ویژه‌ای برخوردار است، کامل بودن ثبت‌رخداد است. آیا تنها کامل بودن یک ثبت‌رخداد برای تشخیص مدل فرایندی کافی است؟ در فصل آینده این موضوع را مورد بررسی قرار می‌دهیم و یک الگوریتم پیشنهادی در مورد تشخیص یک ثبت-رخداد بهینه بیان می‌کنیم.

[۱] Duplicates genetic algorithm miner

فصل پنجم

تعیین تعداد دنباله رخدادها در یک بت رخداد بهینه

۵-۱- مقدمه

همانطور که در فصل پیش بیان شد هدف از فرایندکاوی، مهیا کردن ابزار و تکنیک‌هایی است، که بتوان فرایندها، داده‌ها، کنترل و ساختار اجتماعی یک سیستم را از دنباله‌ای از رخدادها استخراج نمود.

اگرچه اکثر مسایل در زمینه‌ی کاربرد فرایند کاوی بطور رضایت‌بخشی حل شده است، اما همچنان چالش-هایی در جزییات این زمینه وجود دارد [۵۹]. یکی از این چالش‌ها که کمتر به آن توجه شده، مساله کامل بودن ثبت‌رخداد است.

بطور کلی می‌توان گفت کیفیت ساخت یک مدل فرایندی از روی ثبت‌رخداد به دو موضوع مهم بستگی دارد. اول به الگوریتم اکتشاف، دوم به مقدار اطلاعات نمایش داده شده در ثبت‌رخداد. الگوریتم اکتشاف به چند دست تقسیم می‌شود که در فصل پیش آن‌ها را مورد بررسی قرار دادیم. تعریف کامل بودن ثبت‌رخداد برای هر الگوریتم متفاوت است.

اما پرسشی که در اینجا مطرح می‌شود این است که، چطور می‌توان کامل بودن یک ثبت‌رخداد را برای یک وضعیت خاص تعیین کرد؟ و یا اینکه آیا می توان یک ثبت رخداد بهینه برای وضعیت خاص تعیین کرد؟ در این فصل به این موضوع پرداخته می‌شود و با بیان یک الگوریتم پیشنهادی وضعیت بهینه بودن ثبت‌رخداد را برای یک وضعیت خاص مورد بررسی قرار می‌گیرد.

۵-۲- مساله کامل بودن ثبت‌رخداد

اطلاعاتی که در یک ثبت‌رخداد نشان داده می‌شود، یکی از مسائلی است که در تشخیص یک مدل فرایندی از اهمیت ویژه‌ای برخوردار است. همان‌طور که در زمینه‌هایی همچون داده کاوی[۱] و یادگیری ماشین[۲] نمی توان

[۱] Data mining
[۲] Machine learning

همه‌ی داده‌های ممکن را در قسمت آموزش بکار برد، در فرایندکاوی نیز نمی‌توان همه‌ی اطلاعات ممکن را در ثبت‌رخداد بکار برد. بنابراین همه‌ی الگوریتم‌های اکتشاف، یکسری فرض‌ها و محدودیت‌هایی برای کامل بودن ثبت‌رخداد در نظر می‌گیرند. در بعضی از این الگوریتم‌ها، تنها اطلاعاتی در مورد ترتیب رخدادها مورد استفاده قرار می‌گیرد و در بعضی دیگر از الگوریتم‌ها تعداد تکرار رخدادها، نیز دارای اهمیت می‌باشد. در ادامه، مساله کامل بودن ثبت‌رخداد را برای الگوریتم‌های مطرح شده در فصل پیش، مورد بررسی قرار می‌دهیم.

قویترین فرض در مورد کامل بودن، کامل کلی[۱] (GC) نامیده می‌شود. وقتی الگوریتمی یک ثبت‌رخداد را کامل کلی در نظر می‌گیرد، فرض می‌کند که ثبت‌رخداد همه‌ی رفتارهای ممکن فرایند را نشان می‌دهد. در عین حال، اغلب نمی‌توان همه اطلاعات را در یک ثبت‌رخداد نشان داد. بطور مثال اگر یک سیستم شامل حلقه باشد. رخدادهای داخل حلقه می‌توانند بینهایت بار انجام شوند و یا همچنین در سیستم‌هایی که دارای فرایندهای موازی هستند، تعداد دنباله رخدادها خیلی زیاد می‌شود. ثبت‌رخدادی که کامل کلی باشد، معمولا بصورت تئوری مورد بررسی قرار می‌گیرد. در عمل، در سیستم‌های کاربردی، احتمال اینکه ثبت‌رخدادی کامل کلی باشد، صفر در نظر گرفته می‌شود.

اکثر الگوریتم‌های اکتشاف فرضیات و توصیفات متفاوتی را در مورد کامل بودن ثبت‌رخداد در نظر می‌گیرند. توصیفی که اغلب مورد استفاده قرار می‌گیرد "کامل بودن مستقیم توالی"[۲](DS) نام دارد. یک ثبت‌رخداد در صورتی DS است که: " اگر دو گذار بتوانند بصورت مستقیم پشت سر هم اجرا شوند، آنگاه این اتفاق باید حداقل یکبار در ثبت رخداد نشان داده شود. ". این تعریف می‌تواند با در نظر گرفتن حلقه‌هایی بطول دو، گسترش داده شود. این DS تعمیم یافته شده +DS نام دارد. یک ثبت‌رخداد در صورتی +DS است که : " اگر یک گذار دوبار توالی، در حالی که یک گذار بین آن‌ها اتفاق افتد، اجرا شود، آنگاه این اتفاق باید حداقل یکبار در ثبت‌رخداد نشان داده شود. ". و اگر طول حلقه را بیشتر از دو در نظر بگیریم توصیف، ++DS نام دارد.

[۱] Global completeness
[۲] Completeness of direct succession

کمترین محدودیت مربوط به توصیف کامل بودنی است که مربوط به وابستگی منطقی است. این توصیف وابستگی منطقی[۱] (CD) نام دارد. یک ثبت‌رخداد در صورتی CD است که: " اگر دو گذار بایکدیگر ارتباط منطقی داشته باشند آنگاه آن‌ها حداقل یکبار بطور مستقیم متوالی در ثبت‌رخداد ظاهر می‌شوند."

سرانجام، الگوریتم‌هایی که از دیدگاه هیوریستیک استفاده می‌کنند، از ثبت‌رخدادهایی استفاده می‌کنند که توصیف کامل بودن آن‌ها، مربوط به تکرارهای یک توالی است. این توصیف کامل بودن به دو دسته تقسیم می‌شود. اول اینکه، اگر یک وابستگی منطقی بین دو گذار وجود دارد، آنگاه ثبت‌رخداد باید نشان‌دهنده‌ی تعداد تکرارهای این توالی باشد. به این توصیف کامل بودن رخداد مفید[۲] (ES) می‌گویند. دیگر توصیف کامل دنباله مفید[۳] (TS) نام دارد که مربوط به تکرار یک دنباله جریان کار در یک ثبت رخداد می‌باشد. در جدول (۵-۱) بطور خلاصه توصیف کامل ثبت‌رخداد مربوط به الگوریتم‌های اکتشاف بیان شده در فصل پیش را مشاهده می‌کنید همانطور که بیان شد، الگوریتم‌های اکتشاف حالت دیدگاه متفاوتی نسبت به مساله اکتشاف دارند. این دسته از الگوریتم‌ها ابتدا هر دنباله رخداد از ثبت‌رخداد را تبدیل به دنباله‌ای از حالت‌ها و گذارهای بین آن‌ها می‌کند. سپس همه‌ی دنباله‌ها با هم ترکیب شده و تشکیل یک سیستم گذار را می‌دهند و سپس این سیستم گذار تبدیل به یک شبکه پتری می‌شود.

[۱] Causally dependency
[۲] Event significance
[۳] Trace significance

جدول(۵-۱) توصیف کامل بودن الگوریتم‌های بین شده در فصل ۴

نام الگوریتم	توصیف کامل بودن
A	DS
α +	DS+
tsinghua-α	CD
α ++	DS++
α #	DS+
Heuristic Miner	ES
State Discovery	None
Genetic Algorithm	TS
Dupl.GA	TS

۵-۳- فرضیات مساله

در این قسمت فرضیات مساله بهینگی ثبت‌رخداد مورد بررسی قرار می‌گیرد. در ابتدا بهینگی ثبت‌رخداد را تعریف می‌کنیم.

تعریف ۵-۱- ثبت‌رخداد بهینه: ثبت‌رخدادی بهینه گفته می‌شود که با کمترین تعداد دنباله رخداد بتواند یک مدل فرایندی را بازسازی کند.

بعبارتی دیگر قصد داریم ببینیم که برای بازسازی یک مدل فرایندی خاص، ثبت‌رخداد کامل حداقل باید شامل چند دنباله رخداد باشد. بنابراین با در نظر گرفتن یک مدل فرایندی خاص، تعداد دنباله رخدادها در ثبت-رخداد بهینه را بدست می‌آوریم. برای اینکار ابتدا باید مدل فرایندی مورد نظر را تعریف کنیم. اکثر سیستم‌های جریان‌کار از ساختار بلوکی تشکیل شده‌اند. یک شبکه پتری می‌تواند بعنوان ابزاری برای مسیریابی دنباله‌ها به کار رود. به شبکه پتری که به مدل کردن جریان در یک جریان‌کار بپردازد، شبکه جریان کار(WF-net) نامیده

می‌شود. (تعریف دقیق WF-net در فصل پیش بیان شد). در این تحقیق از SWF-net خوش ساخت بعنوان مدل فرایندی خاص مورد استفاده قرار می‌گیرد (به دلیل جامعیت این شبکه، اکثر الگوریتم‌های اکتشاف بر روی SWF-netها کار می‌کنند). با توجه به اینکه مبنای این کار، الگوریتم آلفا می‌باشد، فرض می‌کنیم شبکه مورد نظر فاقد مکان ضمنی، رخداد تکراری و حلقه کوتاه باشد. همچنین برای جلوگیری از ایجاد حلقه بینهایت، شبکه را به SWF-net خوش ساخت فاقد حلقه محدود می‌کنیم.

همانطور که در قسمت قبل بیان شد، توصیفات مختلفی برای یک ثبت‌رخداد کامل وجود دارد و توصیفی که اغلب مورد استفاده قرار می‌گیرد، توصیف DS است. با توجه به اینکه، در ابتدا مدل فرایندی در دسترس است، ثبت‌رخدادی که مورد بررسی قرار می‌گیرد فاقد هر گونه نویز است. بنابراین می‌توان توصیف DS را برای ثبت-رخداد کامل مطرح شده در این مساله در نظر گرفت.

۵-۴- تولید ثبت‌رخدادهای ممکن از یک مدل فرایندی

هدف نهایی فرایندکاوی، ساخت یک مدل فرایندی از روی ثبت‌رخداد می‌باشد. همانطور که در قسمت قبل بیان شد، اطلاعات ثبت‌رخداد، نقش مهمی در ساختن مدل فرایندی دارد. در این قسمت قصد داریم مهندسی معکوس فرایند کاوی را انجام دهیم. بعبارتی دیگر می‌خواهیم همه دنباله رخدادهای ممکن را از یک مدل فرایندی تولید کنیم. این عمل در ارزیابی ثبت‌رخداد، از اهمیت ویژه‌ای برخوردار است.

همانطور که بیان شد، در این مساله، SWF-net بعنوان مدل فرایندی خاص مورد استفاده قرار می‌گیرد. برای تولید همه‌ی دنباله رخدادهای یک SWF-net از نظریه زبان‌های صوری[۱] استفاده می‌کنیم.

[۱] Formal language theory

۵-۴-۱- زبان صوری

زبان صوری در علم کامپیوتر، کاربرد زیادی دارد. در واقع، اساس همه‌ی زبان‌های برنامه نویسی، زبان صوری است.

فرض کنید که Σ یک الفبا[۱] باشد. الفبا یک مجموعه متناهی و ناتهی از سمبل‌ها[۲] است. رشته[۳] (کلمه[۴]) یک ثبت متناهی از الفبا است. رشته‌ی خالی توسط λ نشان داده می‌شود. مجموعه‌ای از همه‌ی رشته‌هایی که بر روی الفبای Σ تعریف شده‌اند، را با Σ^* نشان می‌دهند. همچنین مجموعه‌ی غیر تهی از رشته‌ها که بر روی Σ تعریف شده‌اند، را با نشان Σ^+ می‌دهند. بعبارتی دیگر می‌توان نشان داد که: $\Sigma^+ = \Sigma^* - \{\lambda\}$. یک زیر مجموعه از Σ^* زبان[۵] نامیده می‌شود [۷۲].

تعریف ۵-۲- زبان صوری: به یک مجموعه از رشته سمبل‌ها زبان صوری گفته می‌شود.

به مجموعه قوانینی که بر روی رشته‌ها در زبان صوری تعریف می‌شود، گرامر صوری می‌گویند. این قوانین نحوه‌ی ایجاد رشته‌ها را از الفبا توضیح می‌دهد. یک زبان صوری اغلب توسط گرامر صوری[۶] همچون گرامر منظم[۷]، گرامر مستقل از متن[۸] و گرامر نامحدود[۹]، تعریف می‌شود.

تعریف ۵-۳- گرامر نامحدود: گرامر نامحدود به چندتایی (V, Σ, R, S) گفته می‌شود که در آن:

۱- V مجموعه متناهی از سمبل‌هایی غیر از الفباست.

۲- Σ مجموعه متناهی از الفباست.

---

[۱] Alphabet
[۲] symbols
[۳] string
[۴] word
[۵] language
[۶] Formal grammar
[۷] Regular grammar
[۸] Context-free grammar
[۹] Unrestricted grammar

۳- R به مجموعه متناهی از قوانینی گفته می‌شود که به صورت $A \rightarrow W$ تعریف می شوند که در آن $A \in (\Sigma \cup V)^+ , W \in (\Sigma \cup V)^*$

۴- S سمبل شروع است $(S \in V)$.

با توجه به اینکه تهی عضوی از $(\Sigma \cup V)^*$ است، قانونی بصورت تعریف $A \rightarrow \lambda$ می‌کنیم.

تعریف ۵-۴- گرامر یک زبان نامحدود : فرض کنید G یک گرامر نامحدود باشد. زبان این گرامر، که با L(G) نشان داده می‌شود، بصورت زیر تعریف می‌شود:

$$L(G) = \{W \in \Sigma^* \mid S \overset{*}{\Rightarrow} W\} \qquad (۱)$$

بعبارتی دیگر، تمام رشته‌های تولید شده از گرامر، زبان گرامر است.

۵-۴-۲- زبان شبکه پتری

در سال‌های اخیر، تئوری زبان صوری بطور گسترده‌ای در زمینه‌ی شبکه پتری بکار رفته است و اکثر محققین از شبکه پتری بعنوان تولید کننده‌ی یک زبان، استفاده می‌کنند. برای اینکار گذارها بعنوان سمبل‌ها استفاده می‌شوند و دنباله رخدادهای شبکه بعنوان رشته در نظر گرفته می‌شوند. بنابراین می‌توان ثبت‌رخداد تولید شده از این شبکه را بعنوان زبان در نظر گرفت. توصیف کامل زبان شبکه پتری در [۷۳,۷۴] قابل مشاهده است.

حال می‌خواهیم با استفاده از گرامر نامحدود، همه‌ی دنباله رخدادها را از SWF-net، تولید نماییم. برای این کار، گرامر نامحدود SWF-net مورد نظر را بدست می‌آوریم. زبان بدست آمده از این گرامر، همه‌ی دنباله رخداد-های ممکن از این شبکه است. بطور دقیق‌تر فرض کنید $PN = (P,T, F)$ یک SWF-net باشد. چندگانه‌ی $Q = (V, \Sigma, R, S)$ می‌تواند SWF-net را مانند شکل (۵-۱)، به گرامر نامحدود تبدیل کند.

1. Each place is a non-terminal symbols. V= {place in P/T-net}

2. Σ= {transitions in P/T-net}

3. R= {N→aM | ∀ a ϵ Σ, N ϵ {a sequence of •a}, M ϵ {a sequence of a•}} ∪ {MN→MN | ∀ M,N ϵ V} ∪ {Na →aN | ∀a ϵ Σ, ∀N ϵ V } ∪ {N→ʎ | N is end place in petri net } ∪ {S →M| M is first place }

4. S is first symbol

شکل(۵-۱) گرامر مربوط به یک SWF-Net

لم ۵-۱- زبان تولید شده از گرامر Q، همه‌ی دنباله رخدادهای SWF-net مربوط است.

اثبات: برای اثبات این مساله از استقرا استفاده می‌کنیم. در ابتدا فرض کنید .M نشانه‌گذاری ابتدایی یک SWF-net باشد و M_n نشانه‌گذای نهایی در SWF-net باشد و $T= t.t_1 \dots t_n$ رشته تولید شده توسط این گرامر باشد. با توجه به اینکه سمبل‌ها همان گذارهای ما هستند، رشته‌ی T، دنباله رخداد از SWF-net است اگر:

$$M. \xrightarrow{t_0} M_1 \xrightarrow{t_1} \dots \xrightarrow{tn} M_n \qquad (۲)$$

اثبات در [۷۵] قابل مشاهده است.

حال با استفاده از استقرا اثبات می‌کنیم که اگر رشته T توسط گرامر نامحدود مربوط تولید شود آنگاه رابطه (۲) برقرار است.

۱. فرض کنید n=۰، بعبارتی دیگر شبکه فاقد هر گونه گذاری (فعالیت) باشد و ثبت تولید شده برابر تهی است آنگاه $M. = M_n$

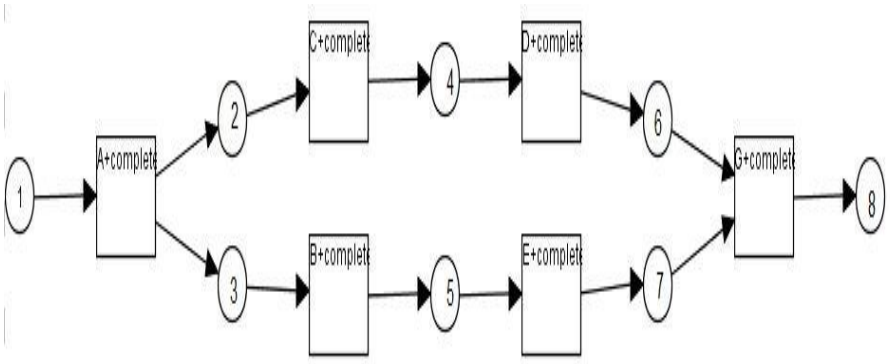
۲. فرض کنید که برای N=k، آنگاه $M. \xrightarrow{t_0} M_1 \xrightarrow{t_1} \dots \xrightarrow{tk} M_k$

۳. حال باید ثابت کنیم برای N=k+۱ داریم $M. \xrightarrow{t_0} M_1 \xrightarrow{t_1} \dots \xrightarrow{tk+1} M_{k+1}$

با توجه به آنچه که فرض شد، کافیست نشان دهیم که بعد از اجرای گذار t_{k+1} داریم: $M_k \xrightarrow{tk+1} M_{k+1}$.

قبل از اجرای گذار t_{k+1}، SWF-net در حالت M_k قرار دارد و در گرامر، برای گـذار t_{k+1} یـک قـانون بصـورت $N \rightarrow t_{k+1}M$ وجود دارد. این قانون نشانه‌ها را از مکان‌های t_{k+1} ● به مکان‌های ● t_{k+1} انتقال می‌دهد (ایـن کـار پس از اجرا t_{k+1} اتفاق می‌افتد). بنابراین حالت M_{k+1} بدست می‌آید. اما هیچ ترتیبی بین مکان‌ها در حالـت M_{k+1} وجود ندارد. از قانون $NM \rightarrow MN$ (M و N،دنباله‌ای از مکان‌ها در شبکه هستند) برای نشـان دادن عـدم وجـود ترتیب بین مکان‌ها استفاده می‌کنیم. حال برای اجرای قانون $N \rightarrow t_{k+1}M$، همه‌ی مکان‌ها باید در کنـار یکـدیگر قرار داشته باشند (چون M و N، دنباله‌ای از مکان‌ها در شبکه هستند). برای نشان دادن این موضـوع از قـانون $Na \rightarrow aN$ $(\forall a \in \Sigma,\ \forall N \in V)$ استفاده می‌کنیم. بنابراین می‌توان نتیجه گرفت که گذار t_{k+1} باعث می شود کـه شبکه از حالت M_k به حالت M_{k+1} برود. بعبارتی دیگر: $M_k \xrightarrow{tk+1} M_{k+1}$.

گرامر با قانون $S \rightarrow M$ شروع می‌شود (M مکان شروع شبکه است) و هنگام شـروع همـواره در حالـت .M قرار دارد و با قانون $N \rightarrow \lambda$ (مکان انتهایی در شبکه SWF-net است) پایان می‌یابد و همواره در حالت M_n پایـان می‌یابد. بنابراین این گرامر همه‌ی رشته‌هایی را تولید می‌کند که حالت .M یک شبکه را به حالت M_n می‌برد. □.



شکل (۵-۲) یک مثال از SWF-net

برای توضیح بیشتر، به SWF-net شکل (۵-۲) دقت کنید. گرامر نامحدود این شبکه در شـکل (۵-۳) نشـان داده شده است.

1. Each place is a non-terminal symbols.
   V= {1, 2, 3, 4, 5, 6, 7, 8}
2. Σ= {A, B, C, D, E, G}
3. R= {S→1, 1→A2, 1→A3, …}
4. S is first non-terminal symbols.

شکل (۵-۳) گرامر مربوط به شبکه شکل (۵-۲)

قوانین R بطور کامل در شکل (۵-۴) نشان داده شده است.

R:

S→1
1→A23
2→C4, 3→B5
4→D6, 5→E7
67→G8
67→76, 87→78
56→65, 54→54
43→34, 23→32
21→12
1A→A1, 1B→B1, 1C→C1, 1D→D1, 1E→E1, 1G→G1
2A→A2, 2B→B2, 2C→C2, 2D→D2, 2E→E2, 2G→G2
3A→A3, 3B→B3, 3C→C3, 3D→D3, 3E→E3, 3G→G3
4A→A4, 4B→B4, 4C→C4, 4D→D4, 4E→E4, 4G→G4
5A→A5, 5B→B5, 5C→C5, 5D→D5, 5E→E5, 5G→G5
6A→A6, 6B→B6, 6C→C6, 6D→D6, 6E→E6, 6G→G6
7A→A7, 7B→B7, 7C→C7, 7D→D7, 7E→E7, 7G→G7
8A→A8, 8B→B8, 8C→C8, 8D→D8, 8E→E8, 8G→G8
8→ʎ

شکل(۵-۴) قوانین مربوط به گرامر شکل (۵-۳)

زبانی که توسط این گرامر تولید می‌شود به صورت زیر است:

L (G) = {ABCDEG, ABCEDG, ACBEDG, ACBDEG, ACDBEG, ABECDG} (۳)

بنابراین این شبکه می‌تواند شش دنباله رخداد را ایجاد کند که این دنباله رخدادها تشکیل یک ثبت‌رخداد کامل را می‌دهند. باید دقت داشت که شرایط اولیه نقش مهمی در این الگوریتم دارد. بطور مثال وجود حلقه باعث می‌شود که این گرامر دارای زبان نامتناهی شود. یا بعبارت دیگر الگوریتم وارد یک حلقه بینهایت می‌شود. که برای حل این مساله نیز می‌توان راه کارهایی را در نظر گرفت.

۵-۵- شناسایی رفتارهای اساسی مدل فرایندی

در این قسمت قصد داریم رفتارهای اساسی یک مدل فرایندی را شناسایی کنیم. ثبت‌رخدادی که این مدل فرایندی را می‌سازد، حتما باید این رفتارها را پوشش دهد. بنابراین می‌توان با شناسایی رفتار اساسی یک مدل فرایندی، دنباله رخدادهای با ارزش (دنباله رخدادهایی که نشاندهنده‌ی بیشترین رفتار اساسی در یک مدل فرایندی است) را شناسایی کرد.

۵-۵-۱- رفتارهای اساسی WF-net

همانطور که بیان شد، در این مساله، WF-net بعنوان مدل فرایندی خاص مورد استفاده قرار می‌گیرد. اکثر سیستم‌های جریان‌کار، از ساختار بلوکی استانداردی همچون And-join، And-split، OR-split و OR-join تشکیل شده‌اند. در فصل سوم این ساختار بلوکی بطور کامل توضیح داده شده است. در ادامه به ساختارهایی که نشان‌دهنده‌ی رفتار اصلی مدل فرایندی است می‌پردازیم:

- ساختار توالی: ساختار توالی یکی از مهمترین ساختارهای یک جریان‌کار است. ساختار توالی به قسمتی از مدل فرایندی گفته می‌شود که دو گذار توسط یک مکان (که در بین آن‌ها قرار دارد) به یکدیگر متصل هستند. ساختار توالی مبنای دیگر ساختار بلوکی در مدل‌فرایندی می‌باشد. بنابراین می‌توان این ساختار را بعنوان یکی از اصلی‌ترین رفتارهای یک مدل‌فرایندی دانست.

-ساختار بلوکی همزمانی: دو گذار که توسط این ساختارهای بلوکی به یکدگر متصل شده‌اند، می‌توانند بصورت همزمان و موازی اجرا شوند. بطور مثال، فرض کنید که دو گذار A و B توسط ساختار بلوکی And-join به یکدیگر متصل شده‌اند. این دو گذار می‌توانند همزمان با هم اجرا شوند، اما امکان دارد یکی زودتر از دیگری پایان یابد. بنابراین دو زیر دنباله AB و BA می‌تواند نشاندهنده‌ی رفتار این ساختار باشد. این ساختارهای بلوکی نیز بعنوان یکی از رفتار اساسی و مهم در مدل فرایندی محسوب می‌شود.

-ساختار بلوکی انتخاب: این گذار موقعیت انتخاب را نشان می‌دهد. وقتی که دو گذار A و B توسط ساختار بلوکی OR-Split به یکدیگر متصل شده‌اند، بین این دو گذار همواره یک انتخاب صورت می‌گیرد. اگر شرایط موجود در SWF-net در نظر گرفته شود، در یک دنباله رخداد هرگز دو رخداد A و B همزمان اتفاق نمی‌افتد.

ساختارهای بلوکی همزمانی و ساختارهای بلوکی انتخاب، هر دو از رفتارهای اساسی یک مدل فرایندی محسوب می‌شود. ساختار بلوکی همزمانی تاثیر بسزایی در ثبت‌رخداد دارد. اما برخلاف ساختار بلوکی همزمانی، ساختار بلوکی انتخاب، تاثیری زیادی بر ثبت‌رخداد ندارد. تنها شرطی که این ساختار بلوکی بر روی SWF-net ایجاد می‌کند، نبود همزمان رخدادهای مربوط در یک دنباله رخداد است. بنابراین می‌توان رفتار ساختار بلوکی انتخاب را با توجه به ساختار توالی آن، در ثبت‌رخداد در نظر گرفت.

۵-۵-۲- شناسایی رفتارهای اساسی مدل فرایندی

با توجه آنچه در قسمت پیش بیان شد، حال رفتار یک SWF-net را شناسایی می‌کنیم. این رفتار را باید به گونه‌ای نشان داد که بتوان آن را در ثبت‌رخداد مورد بررسی قرار داد. برای این کار، رفتار یک SWF-net را به صورت مجموعه‌ای از توالی‌ها نشان می‌دهیم (یک دنباله رخداد، بصورت ترتیبی از رخدادها بیان می‌شود). در ادامه مجموعه رفتار یک SWF-net را که به صورت مجموعه‌ای از توالی‌ها است، را تعریف می‌کنیم.

تعریف ۵-۵- مجموعه توالی رفتار: فرض کنید N یک SWF-net باشد. UF مجموعه‌ای از توالی هاست که نشان‌دهنده‌ی رفتار N است اگر:

۱- اگر در N، دو گذار x و y تشکیل یک ساختار توالی دهند (اگر x بعنوان ورودی مکان و y بعنوان خروجی مکان در نظر گرفته شوند)، آنگاه $xy \in UF$.

۲- اگر در N، دو گذار x و y در ساختار بلوکی همزمانی باشند آنگاه $xy \in UF$ و $yx \in UF$.

در تعریف ۵-۵ ساختار بلوکی انتخاب در نظر گرفته نشده است. همانطور که بیان شد تنها شرطی که ساختار بلوکی بر روی ثبت‌رخداد دارد، نبود همزمان رخدادهای مربوط در یک دنباله رخداد است (چون از بین رخدادها همواره یکی انتخاب می‌شود). اما این شرط بطور طبیعی در دنباله رخدادها رعایت می‌شود. در صورت رعایت نشدن این شرط، دنباله رخداد مربوطه نویز محسوب می‌شود. اما همه‌ی دنباله رخدادها را از یک مدل فرایندی خاص تولید شده‌اند و طبق آنچه در قسمت فرضیات بیان شد، در ثبت‌رخداد مربوطه هیچ نویزی وجود ندارد.

۵-۵-۳- تعمیم رفتارهای اساسی شناخته شده در یک مدل فرایندی

در قسمت قبل توانستیم رفتارهای اساسی یک SWF-net را بدست آوریم. اما همانطور که در فصل پیش بیان شد، الگوریتم آلفا حتی در مورد بعضی از WF-net ها نیز درست عمل می‌کند. بنابراین در این قسمت قصد داریم مدل فرایندی را به WF-net تعمیم دهیم نحوه‌ی بدست آمدن مجموعه توالی رفتارها در شکل (۵-۵) نشان داده شده است.

Let N be a WF-net then:

۱. ST= {(A, B)| for each place p ϵ N (except first and end place), B= p● and A=●p}

۲. U۱= {{xy |∀ (A, B) ϵ ST, (x, y) ϵ A×B}

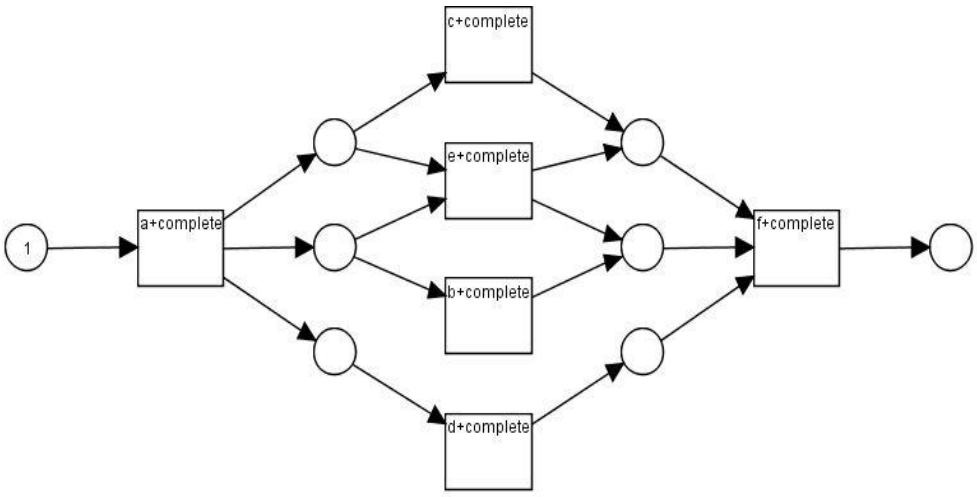
۲. U۲= { xy, yx |∀(A, B) ϵ ST, ∀(C, D) ϵ ST, D∩B ≠ φ, (x,y) ϵ A×C , x ≠y , ∄A x,y ϵ A }

۳. U۳= { xy, yx |∀(A, B) ϵ ST,∀(C, D) ϵ ST, A∩C ≠ φ, (x,y) ϵ B×D, x ≠ y , ∄D x,y ϵ D}

٤. UF= U۱ ∪ U۲ ∪ U۳

شکل (۵-۵) تعمیم مجموعه توالی رفتار یک WF-net

در شکل (۵-۵)، UF مجموعه توالی رفتار یک WF-net را نشان می‌دهد. البته باید دقت داشت که این مجموعه رفتارهای اساسی یک WF-net را نشان می‌دهد و هیچ تضمینی نمی دهد که این مدل فرایندی قابل بازسازی باشد. امکان دارد که الگوریتمی بتواند آن را بازسازی کند یا خیر. این قسمت فقط نشان می‌دهد که اگر این مدل فرایندی قابل بازسازی باشد آنگاه مجموعه UF، رفتارهای اساسی آن مدل فرایندی را نشان می‌دهد.



شکل (۵-۶) مثالی از یک WF-net

زوج مرتب‌های مجموعه ST، گذارهای ورودی و خروجی یک مکان را نشان می‌دهد. بنحوی که مولفه‌ی اول این زوج مرتب گذارهای ورودی مکان و مولفه‌ی دوم گذارهای خروجی مکان را نشان می‌دهد.

مجموعه‌ی U۱، رفتار ساختار توالی را در شبکه نشان می‌دهد. برای این کار باید ضرب کارتزین بین مجموعه ورودی‌های یک مکان و مجموعه خروجی آن مکان را بدست آورد. البته باید دقت کرد که مکان ابتـدایی ورودی ندارد و همچنین مکان انتهایی خروجی ندارد بنابراین این دو مکان را در نظر نمی‌گیریم.

مجموعه U۲ و U۳ نحوه ساختار بلوکی همزمانی را نشان می‌دهد. برای توضیح بیشتر به شکل (۵-۶)، دقـت کنید. این شبکه بدلیل نقض شرط دوم تعریف ۴-۹، SWF-net نیست. اما الگوریتم آلفا آن را بدرستی بازسازی می‌کند.

در این شبکه گذار f و a یک ساختار بلوکی همزمانی را نشان می‌دهد. بنابراین چهار گذر b, c, d, e باید بطور همزمان اجرا شوند. طبق ویژگی ۴-۲، هیچ دو گذری نباید توسط چند مکان به یکدیگر متصل باشند. اما در این جا گذار e توسط دو مکان به f متصل شده است. همچنین گذار e نمی‌تواند همزمان با b (و یا c) اتفـاق افتـد. چون مکان خروجی این گذارها یکی است و اگر این دو گذار همزمان اجرا شوند این مکان شامل دو نشانه مـی-شود، که این شرایط امن بودن مدل فرایندی را نقض می‌کند. مجموعه U۲ نیز همین موضوع را نشان می‌دهنـد. اگر مجموعه‌های ورودی دو مکان با یکدیگر اشتراک داشته باشند، گذارهای مشترک نمی‌توانند با دیگر گذارهای مجموعه‌های ورودی دو مکان، همزمان باشند.

همچنین با توجه به ساختار بلوکی همزمانی که a ایجاد مـی‌کنـد، مجموعـه مکـان‌هـای ورودی گـذار e بـا مجموعه مکان‌های ورودی b (و یا c) دارای اشتراک هستند. این مکان‌های مشترک حداکثر شامل یـک نشـانه هستند. پس بنابراین این دو گذار هیچ‌گاه همزمان اتفاق نمی‌افتند. مجموعه U۳ نیز همین موضوع را نشان می-دهند. اگر مجموعه‌های خروجی دو مکان با یکدیگر اشتراک داشته باشند، گذارهای مشترک نمی‌توانند با دیگـر گذارهای مجموعه‌های خروجی دو مکان، همزمان باشند. اما دیگر گذارها می‌توانند بطور همزمان اجرا شوند.

بنابراین می‌توان نتیجه گرفت که سه گذار d, b, c بطور همزمان اجرا می‌شوند اما گذار e فقط با گذار d همزمان است. مجموعه‌های U۱ و U۲ و U۳ با توجه به آنچه گفته شد رفتارهای اساسی را در یک WF-net نشان می‌دهد. پس بنابراین می‌توان با اجتماع این سه مجموعه به مجموعه‌ی نهایی UF دست یافت.

برای توضیح بیشتر رفتارهای اساسی مدل فرایندی شکل (۵-۲) را بصورت مجموعه‌ای از توالی‌ها بیان می‌کنیم. با توجه به شکل (۵-۲)، مجموعه ST از (۴) بدست می‌آید.

ST={({D},{G}),({E},{G}),({C},{D}),({B},{E}),({A},{B}),({A},{C})} (٤)

و همچنین مجموعه‌ی U۱ از (۵) بدست می‌آید.

U۱ = {DG, EG, CD, BE, AB, AC} (۵)

و همچنین مجموعه U۲ و U۳ از (۶) و (۷) بدست می‌آید.

U۲ = {ED, DE} (۶)

U۳ = {BC, CB} (۷)

بنابرین مجموعه UF برابراست با:

UF = {DG, EG, CD, BE, AB, AC, ED, DE, BC, CB} (۸)

مجموعه UF مجموعه توالی رفتار مدل فرایندی شکل (۵-۲) است.

۵-۶- الگوریتم پیشنهادی برای تعیین تعداد دنباله رخدادها در ثبت رخداد بهینه

در قسمت قبل توانستیم رفتارهای اساسی یک WF-net بصورت مجموعه‌ای از توالی‌ها را بدست آوریم. همچنین در قسمت ۴ توانستیم همه‌ی دنباله رخدادهای یک مدل فرایندی را بدست آوریم. حال در این قسمت قصد داریم با توجه به مجموعه رفتار بدست آمده، ثبت‌رخدادی را پیدا کنیم که با کمترین تعداد دنباله رخداد بتواند این رفتارهای اساسی را پوشش دهد. برای این کار باید در همه دنباله رخدادهای یک مدل فرایندی را در نظر بگیریم و دنباله رخدادهایی را انتخاب کنیم که بیشترین رفتار مدل فرایندی را نشان می‌دهد. باید دقت

داشت که ثبت‌رخداد بهینه باید بتواند همه‌ی رفتارهای اساسی (که در مجموعه توالی نشان داده شده) مدل فرایندی را پوشش دهد. برای این کار از الگوریتم بیان شده در شکل (۵-۷) استفاده می‌کنیم.

```
1. Oplog={}
2. While UF is not NULL
        3. for each log ϵ L(G) - Oplog
              S= {δ | δ ϵ UF and δ ϵ log}
              Assign number of elements of S to Numlog
      End for
        4. Descent sort loges based on Numlog
        5. Log1=Select first log.
        6. U1= {δ | δ ϵ UF and δ ϵ log1}
        7. UF=UF-U1
        8. added log1 to Oplog
End while
```

شکل(۵-۷) الگوریتم پیشنهادی برای تعیین تعداد دنباله رخدادها در ثبت‌رخداد بهینه

این الگوریتم حریصانه مجموعه UF (که نشان دهنده‌ی رفتار مدل فرایندی است) و مجموعه کامل ثبت-رخداد تولید شده از مدل فرایندی (L(G)) را بعنوان ورودی دریافت می‌کند و مجموعه Oplog را بعنوان خروجی بر می‌گرداند. مجموعه‌ی Oplog، ثبت‌رخداد بهینه را نشان می‌دهد.

در این الگوریتم ابتدا مجموعه ثبت‌رخداد بهینه (Oplog)، تهی در نظر گرفته می‌شود. سپس به هر دنباله-رخداد تولید شده عددی (Numlog) نسبت می‌دهد. این عدد نشان‌دهنده‌ی تعداد رفتاری است که این دنباله رخداد می‌تواند پوشش دهد. سپس دنباله رخدادها بر اساس Numlog مرتب می‌شوند. دنباله رخدادی که بیشترین Numlog را داشته باشد، انتخاب می‌شود و در ثبت‌رخداد بهینه قرار می‌گیرد و رفتارهای پوشش داده

شده از UF حذف می‌شوند. این کار ادامه پیدا می‌کند تا وقتی که مجموعه UF برابر تهی شود. به عبارت دیگر تا وقتی که همه‌ی رفتارهای مدل فرایندی پوشش داده شود. در انتها مجموعه‌ی Oplog، ثبت‌رخداد بهینه را نشان می‌دهد.

همان‌طور که بیان شد خروجی این الگوریتم مجموعه‌ی Oplog است که ثبت‌رخداد بهینه را نشان می‌دهد. اما باید دقت داشت که این مجموعه خروجی منحصربفرد نیست. چون امکان دارد دو دنباله رخداد به یک اندازه رفتار سیستم را پوشش دهند و باید بین این دو دنباله رخداد یکی انتخاب شود. بنابراین ما فقط تعداد اعضای Oplog را بعنوان تعداد دنباله رخدادهای موجود در ثبت‌رخداد بهینه در نظر می‌گیریم.

برای توضیح بیشتر الگوریتم را قدم بقدم اجرا می‌کنیم. فرض کنید می‌خواهیم ثبت‌رخداد بهینه را برای مدل فرایندی شکل (۵-۲) بدست آوریم. در قسمت‌های پیش توانستیم مجموعه همه‌ی دنباله رخدادها (L(G)) و مجموعه توالی رفتار (UF) را بدست آوریم. شکل (۵-۸) اجرای قدم بقدم این الگوریتم را نشان می‌دهد.

First:

UF= {DG, EG, CD, BE, AB, AC, DE, ED, CB, BC}

L (G) (total event trace) = {ABCDEG, ABCEDG, ACBEDG, ACBDEG, ACDBEG, ABECDG}

Step 1:

Loges	Numlog
ABCDEG	5
ACBEDG	5
ABECDG	4
ACBDEG	4
ACDBEG	4
ABCEDG	4

Step 2:

OPlog={ ABCDEG } ,UF={DG,BE,AC,ED,CB}

Loges	Numlog
ACBEDG	5
ABECDG	2
ACBDEG	2
ACDBEG	2
ABCEDG	2

Final:

OPlog = {ABCDEG, ACBEDG}, UF = {}

OPlog = {ABCDEG, ACBEDG} final result

شکل(۵-۸) مثالی از الگوریتم پیشنهادی برای تعیین تعداد دنباله رخدادها در ثبت‌رخداد بهینه

همانطور که در شکل (۵-۸) مشاهده می‌کنید، ثبت‌رخداد کامل شامل شش دنباله رخداد است در صورتی که ثبت‌رخداد بهینه شامل دو دنباله رخداد است.

۵-۶-۱- تحلیل الگوریتم پیشنهادی

الگوریتم پیشنهادی، یک نوع الگوریتم حریصانه است که همواره دنباله رخدادی را انتخاب می‌کند کـه دارای بیشترین ویژگی است. اما مساله‌ای مهم این است که آیا این الگوریتم همواره تعداد اعضای ثبت رخداد بهینـه را درست بدست می‌آورد. برای توضیح بیشتر به مثال زیر دقت کنید.

فرض کنید در یک مدل فرایندی مجموعه رفتار بهینه و مجموعه دنباله‌رخدادهای کامل بصورت زیر است.

UF = {AB, BC, CD, BA, DE, CE}

L (G) = {ABCDE, ABCED, BACDE, EBADC}

وقتی الگوریتم پیشنهادی برای این مثال اجرا می‌شود، مجموعه‌ی {ABCDE, ABCED, BACDE} بعنوان ثبت‌رخداد بهینه بدست می‌آورد، که دارای سه دنباله رخداد است. امـا در حـالی کـه مـی‌تـوان بـا ثبـت‌رخـداد { ABCED, BACDE } همه‌ی رفتار UF را پوشش داد، که دارای دو دنباله رخـداد اسـت. بنـابراین مـی‌تـوان نتیجه گرفت که این الگوریتم همواره حل بهینه را تضمین نمی‌کند.

در این قسمت قصد درایم یک تخمینی از حل بهینه را بدست آوریم. ابتدا مسـاله را تبـدیل بـه یـک مسـاله تصمیم گیری می‌کنیم یا به عبارت دیگر "آیا می توان حداکثر با K تا دنباله رخداد، همه اعضای UF را پوشـش داد؟"

قضیه۵-۱- یک مساله C، NP کامل است اگر:

۱. در NP باشد.

۲. به ازای یک مساله NP کامل دیگر مثل B داشته باشیم: $B \propto C$

به عبارت دیگر مساله B به C کاهش پذیر باشد.

لم ۵-۲- مساله تعیین تعداد دنباله رویداد در یک دنباله رویداد بهینه یک مساله‌ی NP کامل است.

اثبات:

ابتدا نشان می‌دهیم مساله ما جزء مسایل NP است. برای این کار باید یک الگوریتم تصدیق کننده تعریف کنیم که با دریافت مجموعه‌ای از دنباله رخدادها که تایید کند که اعضای این مجموعه از k بیشتر نیست و زیر دنباله های این دنباله رخدادها مجموعه UF را پوشش می دهد. (که چنین الگوریتمی به را حتی قابل تعریف است). پس بنابراین این مساله جزء مسایل NP است.

حال مساله پوشش راس ها را به عنوان یک مساله NP کامل را در نظر می‌گیریم. بنابراین باید نشان دهیم:

Vetex cover∞ problem

مساله پوشش راس ها این گونه بیان می‌شود: فرض کنید گراف G=(V,E) داریم که قصد داریم مجموعه راس‌هایی با مینمم اعضا را بدست آوریم که همه یال‌ها حداقل با یکی از این رئوس در ارتباط باشد. به عبارت دیگر، " ایا می توان مجموعه ای حداکثر با j عضو از راس‌ها بدست آورد که همه یال‌ها حداقل با یکی از این رئوس در ارتباط باشد."

حال قصد داریم با یک الگوریتم تبدیل مساله پوشش راس ها را به مساله بیان شده کاهش دهیم.

برای این کار UF=E در نظر می گیریم و راس‌های G را از ۱ تا n نامگذاری می‌کنیم و مجموعه S_i را مجموعه ای از یال های مجاور راس i در نظر می‌گیریم. دقت کنید که برای همه i ها داریم: S_i زیر مجموعه ای از UF است. و در نهایت K=j قرار می دهیم.

برای اثبات جوابمان باید نشان دهیم که جواب یک نمونه از مساله پوشش راس ها بله است اگـر و فقـط اگـر نمونه مساله ما نیز جواب بله داشته باشد.

فرض کنیم مجموعه پوشش راس ها حداکثر j باشد. طبق الگوریتم تبدیل ما، مجموعه S مطابق بـا مجموعـه دنباله رخداد ها می‌باشد و با توجه به این که K=j مجموعه دنباله رخدادهای ما حداکثر K تا دنباله رخداد دارد. و همچنین ما بیان داشتیم که مجموعه دنباله رخداد ها همه اعضای UF را پوشش می‌دهد. حال با توجه به این-که UF=E ، فرض کنید یالی مانند e در E موجود می باشد. چون S تمام راس های مجاور یال ها را پوشش می دهد، حداکثر یک راس در S وجود دارد که با e مجاور است. بنابراین با توجه به این کـه مجموعـه S مطـابق بـا مجموعه دنباله رخدادها می‌باشد، می توان نتیجه گرفت که حداقل یک دنباله رخداد وجود دارد که شامل صفت مربوطه از UF باشد. (شامل این یال از E)

حال فرض کنید که یک نمونه از مساله خودمان با مجموعه دنباله رخداد (C) با اندازه K داریم. با توجـه بـه این‌که هر دنباله رخداد در C مرتبط به یک راس در گراف G است، S را مجموعه ای از این راس ها فرض کنید. همچنین $|S|=|C|$ و S حداکثر شامل j راس می‌باشـد. در ادامـه کـار یـک یـال (e) را در G در نظـر بگیریـد. همچنین طبق آنچه گفته شده هر یال (مانند e) مطابق با یک صفت در UF است، پس بنابراین در مجموعـه C باید حداقل یک دنباله رخداد باشد که یال e مر بوطه را پوشش دهد. بنابراین مجموعه S باید حداقل دارای یـک راس مجاور با e باشد.

بنابراین می توان نتیجه گرفت که مساله‌ی بهینگی بیان شده NP کامل و از نوع NP سخت است. الگوریتمی که بیان شد تضمینی برای ارائه‌ی حل بهینه ندارد اما حل‌هایی می‌دهد که به حل‌های بهینه نزدیکند. غالبا می-توانیم حدی بدست آوریم که تضمینی برای میزان نزدیکی حل به حالت بهینه باشد.

برای حل این مساله راه‌حل‌های زیادی را می‌توان در نظر گرفت که یکی از آن‌ها تقریب راه حل بهینه است. فرض کنید تعداد دنباله رخدادهای بهینه برابر k=OPT باشد و همچنین E_t تعداد رفتارهایی از UF است که تا مرحله t هنوز پوشش داده نشده است. بنابراین $E_0 = E$ و همچنین برای هر E_t تعداد دنباله‌رخدادهای مورد نیاز بیشتر از k نیست. الگوریتم پیشنهادی در مرحله t+1 ، دنباله‌رخدادی را انتخاب می‌کند که بیشترین رفتار را از E_t پوشش دهد. تعداد رفتارهای پوششی برای دنباله رخداد انتخاب شده، باید حداقل برابر $|Et|/k$ باشد. در غیر این صورت هیچ راهی وجود ندارد که رفتارهای E_t را بتوان در k دنباله رخداد پوشش داد. بنابر این می توان نتیجه گرفت که:

$$|Et+1| \leq |Et| - |Et|/k$$

و پس از حل آن داریم:

$$|Et| \leq n\,(1-1/k)^t$$

الگوریتم زمانی پایان می‌یابد که $|Et| < 1$ بنابراین برای حل t داریم:

$$(1-\frac{1}{k})^t \geq \frac{1}{n}$$

$$\Rightarrow \quad n \geq (\frac{k}{k-1})^t$$

$$\Rightarrow \quad \ln n \geq t \ln(1+\frac{1}{k-1}) \approx \frac{t}{k}$$

$$\Rightarrow \quad t \leq k \ln n = OPT \ln n$$

می توان با آنالیز بیشتر به تخمین بهتری دست یافت. فرض کنید $|Et| < K$ بنابراین:

$$n\left(1-\frac{1}{k}\right)^t = k$$

$$\Rightarrow \quad n\frac{1}{e^{t/k}} = k$$

$$\Rightarrow \quad e^{t/k} = \frac{n}{k}$$

$$\Rightarrow \quad t = k \ln\frac{n}{k}$$

بنابراین بعد از $k\ log_e\ n/k$ مرحله تنها k رفتار باقی می‌ماند. هر دنباله‌رخداد در هر مرحله حداقل یک رفتار را مورد پوشش قرار می‌دهد بنابراین:

$$ALG \leq\ OPT\ (\ln\frac{n}{opt} + ۱) \qquad (۹)$$

(n تعداد رفتار است)

که در (۹)، ALG تخمین تعداد دنباله رخدادها در ثبت‌رخداد بهینه است. البته این بدان معنی نیست که ALG همواره $\ln\frac{n}{opt} + ۱$ برابر OPT است. به ازای بسیاری از دنباله‌ها ممکن است نزدیک یا حتی برابر OPT باشد. این مساله در واقع نمایشی دیگر از مساله‌ی پوشش[۱] مجموعه‌هاست و دقیقا حل‌هایی که برای پوشش مجموعه‌ها بکار می‌رود، در مساله‌ی یافتن ثبت‌رخداد بهینه صدق می‌کند.

البته برای این مساله راه حل‌های دیگری وجود دارد که در بعضی از آن‌ها می‌توان حل بهینه را بصورت دقیق بدست آورد. بطور مثال الگوریتم‌های شاخه وحد، حل بهینه را بصورت دقیق محاسبه می‌کند اما پیچیدگی زمانی آن بصورت نمایی است . الگوریتم‌های شاخه و حد از درخت فضای حالت برای حل مساله استفاده می‌کند. الگوریتم شاخه و حد در هر گره عددی را محاسبه می‌کند تا تعیین شود که آیا آن گره امیدبخش هست یا خیر. شاید راه‌حل‌هایی که دارای پیچیدگی زمانی نمایی هستند در آزمایشیات جواب دقیق‌تری به ما بدهند اما اگر یک سیستم دارای ثبت‌رخدادی بزرگی باشد، این دسته از الگوریتم‌ها کارایی چندانی ندارند. بطور مثال فرض کنید که در یک سیستم n کار بطور موازی انجام می‌شوند آنگاه تعداد اعضای ثبت‌رخداد کامل این سیستم !n است و اگر بخواهیم از شیوه‌ی شاخه و حد استفاده کنیم. مطمئنا زمان زیادی باید صرف شود.

[۱] Set cover

۵-۷- نتیجه‌گیری

در این فصل یکی از چالش‌های مهم فرایندکاوی مورد بررسی قرار گرفت. همواره ثبت‌رخدادی که برای اکتشاف استفاده می‌شود باید شامل اطلاعات مهمی از سیستم باشد. وجود نویز مطمئنا مشکلاتی را برای الگوریتم اکتشاف ایجاد می‌کند. البته الگوریتم‌های اکتشافی همچون دسته الگوریتم‌های هیورستیک و دسته الگوریتم‌های ژنتیک تا حدودی می‌توانند مشکل نویز را برطرف کنند. در این فصل دو پرسش مهم را تحلیل کردیم : " چطور می‌توان کامل بودن یک ثبت‌رخداد را برای یک وضعیت خاص تعیین کرد؟ و یا اینکه آیا می توان یک ثبت‌رخداد بهینه برای وضعیت خاص تعیین کرد؟ ".

در این فصل ابتدا ثبت رخداد کامل را تشریح کردیم و انواع ثبت‌های رخداد مورد نیاز را برای الگوریتم‌های مختلف اکتشاف توصیف کردیم. سپس فرضیات لازم را برای حل مساله ثبت رخداد بهینه انجام دادیم. با توجه به این‌که ما مبنای کار خود را الگوریتم الفا قرار دادیم، اکثر فرضیات در نظر گرفته شده، فرضیاتی است که برای الگوریتم آلفا در نظر گرفته می‌شود. در قسمت بعد با انجام یک مهندسی معکوس، یک ثبت رخداد کامل از مدل فرایندی را بدست آوردیم. این ثبت رخداد کامل شامل همه‌ی دنباله‌رخدادهای ممکن برای مدل فرایندی بود. در این قسمت برای تولید ثبت‌رخداد کامل از نظریه‌ی زبان صوری استفاده کردیم. سپس در قسمت بعدی رفتار-های اساسی یک مدل فرایندی خاص را مورد بررسی قرار دادیم. رفتارهایی که در انتخاب دنباله‌رخدادها از اهمیت ویژه‌ای برخوردار بود. این رفتارهای اساسی را بصورت مجموعه‌ای از رخدادهای متوالی نشان دادیم و در انتها نیز با بیان یک الگوریتم حریصانه تخمینی از تعداد دنباله‌رخدادها را در یک ثبت رخداد بهینه بدست آوردیم. مساله تعیین ثبت رخداد بهینه با استفاده از مجموعه رفتارهای اساسی، یک مساله‌ی NP سخت است و الگوریتم حریصانه تخمینی از حل بهینه را می‌دهد. هر چند الگوریتم‌های دیگری نیز وجود دارد که حل بهینه را بطور دقیق محاسبه می‌کنند. اما پیچیدگی زمانی آن‌ها نمایی است.

فصل ششم

پیاده‌سازی و ارزیابی نتایج

۶-۱-مقدمه

در فصل‌های چهار به معرفی فرایندکاوی پرداختیم و در فصل ۵ توانستیم تعداد دنباله‌رخدادها در ثبت‌رخداد بهینه را برای بازسازی یک مدل فرایندی خاص بدست آوریم. مساله‌ی تعیین تعداد دنباله رخدادهای بهینه از دسته مسائل NP سخت است. بنابراین نمی‌توان برای آن راه حل چند جمله‌ای یافت. بنابراین ما توانستیم در فصل پیش تخمینی از جواب بهینه را بدست آوریم. در این فصل قصد داریم با استفاده از نرم‌افزارهای فرایند-کاوی به پیاده‌سازی چند مثال بپردازیم و تعداد دنباله‌رخدادهای بهینه را بدست آوریم.

در این فصل ابتدا نرم افزار Prom را تشریح می‌کنیم و در ادامه به بیان چند مثال می پردازیم.

۶-۲- نرم‌افزار Prom

با گسترش و پیشرفت علم فرایندکاوی، نیاز به نرم‌افزاری که بتواند این زمینه‌ی تحقیقاتی را پشتیبانی کند روز به روز بیشتر شد [۷۶]. در سال‌های ابتدایی، محققین برای نمایش و پیاده‌سازی الگوریتم و متدهای خود از نرم‌افزارهای مختلفی استفاده می‌کردند. این نرم‌افزارها اکثرا برای استفاده صنعتی مناسب نبودند و نمی‌توانستند دنباله‌رخدادهایی که دارای دنباله‌رخدادهای فراوانی هستند، را پشتیبانی کنند. این نرم‌افزارها بیشتر برای تست الگوریتم‌ها بکار می‌رفتند.

در سال ۲۰۰۳، تلاش‌های زیادی برای ایجاد نرم‌افزاری جامع برای فرایندکاوی انجام شد. سرانجام نرم‌افزار Prom ایجاد شد. این نرم‌افزار دارای ویژگی‌های مهم زیر است:

۱- این نرم‌افزار ، یک نرم‌افزار دارای معماری آزاد محسوب می‌شود ومی‌تواند برای الگوریتم‌های جدید گسترش یابد.

۲- این نرم‌افزار می‌تواند بطور گسترده برای کاربردهای صنعتی بکار رود.

۳- استفاده از این نرم‌افزار بسیار آسان می‌باشد.

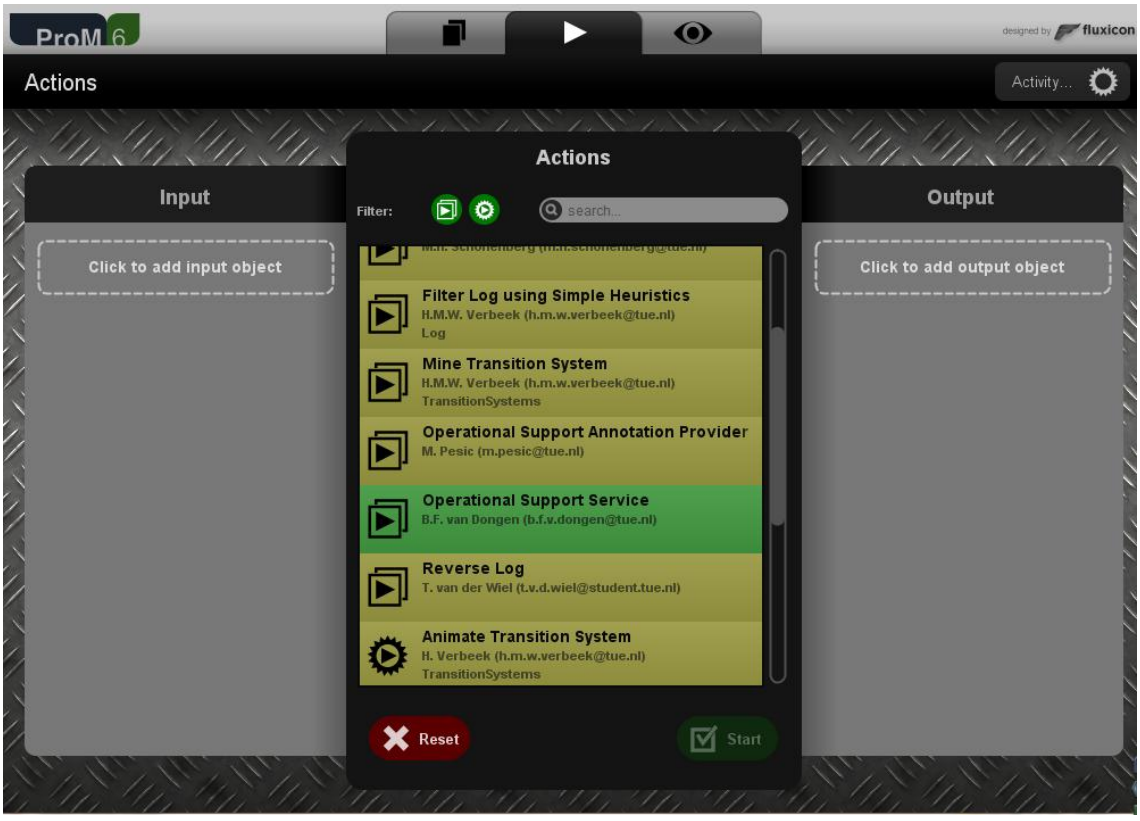
در سال‌های اخیر Prom بصورت چشمگیری گسترش یافته است. اکنون این نرم‌افزار می‌تواند مدل‌های فرایندی گوناگونی را همچون شبکه پتری و شبکه هیوریستیک را پشتیبانی کند. علاوه بر این این نرم‌افزار می‌تواند از دنباله‌رخدادهای بزرگ (در اندازه‌های صنعتی) پشتیبانی کند. امروزه این نرم‌افزار بطور گسترده در صنعت استفاده می‌شود.

اگرچه این نرم‌افزار اکثر الگوریتم‌ها را پوشش می‌دهد و نرم‌افزاری خوب برای فرایندکاوی محسوب می‌شود، اما دارای محدویت‌هایی نیز می‌باشد. از جمله اینکه این نرم‌افزار با ثبت‌رخدادها کار می‌کند. اما گاهی اوقات در صنعت، سیستم دارای اطلاعاتی است که نمی‌توان آن را در ثبت‌رخداد نشان داد.

در ادامه با تشریح نحوه محیط این برنامه به بیان چند مثال با این نرم‌افزار می پردازیم. همان‌طور که بیان شد این نرم‌افزار دارای کاربردهی فراوانی است. در این فصل فقط به توصیف آن قسمتی از نرم‌افزار می‌پردازیم که مربوط به پیاده‌سازی این پایان‌نامه است.

شکل (۶-۱) محیط کار نرم‌افزار Prom

در شکل (۶-۱)، محیط کار این نرم‌افزار را مشاهده می‌کنید. این نرم‌افزار یک ثبت‌رخداد را بعنوان ورودی دریافت می‌کند. این ثبت‌رخدادها بطور جداگانه‌ای در یک فایل با پسوند xes. ذخیره می‌شوند. در نرم‌افزار Prom، عملگرها وفیلترهای مختلفی بر روی ثبت‌رخداد اجرا می‌شود. این عملگرها در قسمت فیلترهای نرم افزار لیست شده است (شکل(۶-۲)). وقتی که ثبت‌رخداد را بعنوان ورودی دریافت کردیم، می‌توان با مراجعه به قسمت فیلترها، عملگر مورد نظر را بر روی ثبت‌رخداد انجام داد. یکی از این عملگرها، الگوریتم آلفا می‌باشد. این عملگر، الگوریتم آلفا را بر روی ثبت‌رخداد ورودی اجرا می‌کند. اگر ثبت‌رخداد ورودی شرایط لازم را برای آلفا الگوریتم نداشته باشد، همچنان این الگوریتم بر روی دنباله‌رخداد اجرا می‌شود، اما خروجی اشتباه است. بعبارتی دیگر مدل فرایندی بدرستی بازسازی نمی‌شود.



شکل (۶-۲) شکل لیست فلیترها و عمگرهای Prom

۶-۳- بازسازی مدل فرایندی با استفاده از ثبت‌رخداد بهینه

در این قسمت به توصیف چند مدل فرایندی می پردازیم و ثبت‌رخداد بهینه را از آن ایجاد می‌کنیم و سپس با استفاده از نرم‌افزار Prom، به بازسازی آن می‌پردازیم.

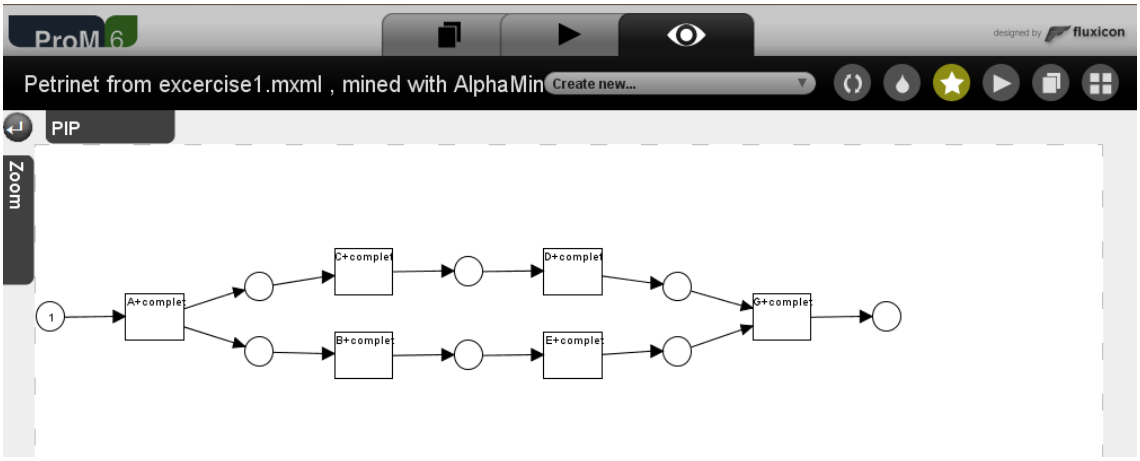
۶-۳-۱- مثال اول

در اولین توصیف مدل فرایندی، WF-net شکل(۵-۲) را در نظر بگیرید. در فصل پیش توانستیم مجموعه همه‌ی ثبت‌رخدادها، مجموعه‌رفتارهای اساسی و ثبت‌رخداد بهینه را بصورت زیر بدست آوریم.

L (G) = {ABCDEG, ABCEDG, ACBEDG, ACBDEG, ACDBEG, ABECDG}
UF = {DG, EG, CD, BE, AB, AC, DE, ED, CB, BC}

OPlog= {ABCDEG, ACBEDG}

حال با استفاده از Oplog به بازسازی مدل‌فرایندی می‌پردازیم. شکل (۶-۳) نتیجه بدست آمده از نرم‌افزار Prom را نشان می‌دهد.

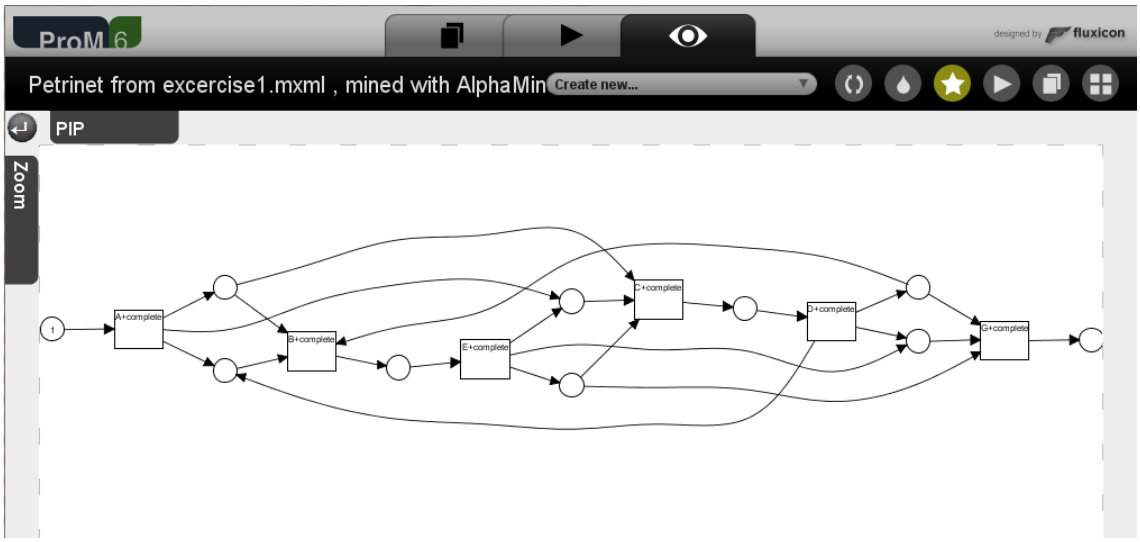


شکل (۶-۳) بازسازی مدل فرایندی شکل (۵-۲) با ثبت‌رخداد بهینه

حال برای اینکه کارایی ثبت‌رخداد بهینه را نشان دهیم، سعی بر بازسازی مدل فرایندی بـا اسـتفاده از ثبـت رخداد زیر می‌کنیم.

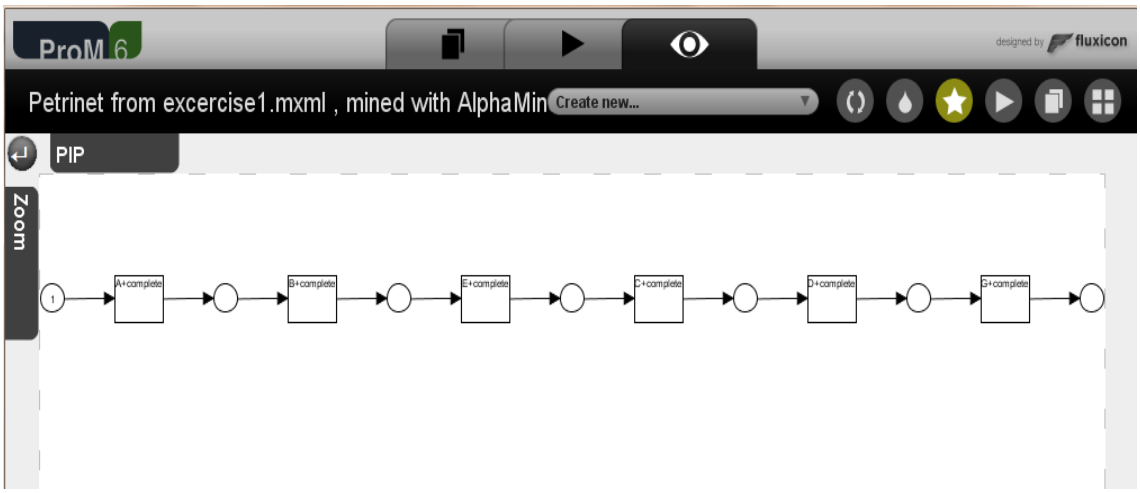
eventlog۱= { ACDBEG, ABECDG }

شکل (۶-۴) نتیجه بدست آمده از نرم‌افزار Prom را نشان می‌دهد.



شکل (۶-۴) ساخت مدل فرایندی توسط ثبت‌رخداد eventlog۱= { ACDBEG, ABECDG }

شکل (۶-۴) نشان می‌دهد که مجموعه eventlog۱ رفتارهای اساسی مـدل فراینـدی شـکل (۵-۲) را نشـان نمی‌دهد. البته هدف ما از الگوریتم پیشنهادی، تخمین تعداد اعضای ثبت‌رخداد بهینه است که در اینجـا تعـداد اعضای ثبت‌رخداد بهینه ۲ تخمین زده می‌شود. البته اگر با استفاده از الگوریتم‌های شـاخه و حـد ثبـت رخـداد بهینه محاسبه شود. باز هم به همین جواب می‌رسیم. البته در اینجا بدیهی است که اگر از ۱ دنباله‌رخداد (کمتر از ۲) برای بازسازی استفاده کنیم، بازسازی درست انجام نمی‌شود. چون ۱ دنباله رخداد فقط توالی بین رخداد-ها را نشان می‌دهد. بطور مثال اگر از ثبت رخداد ABECDG استفاده‌کنیم، شکل (۶-۵) بدست می‌آید کـه بـاز هم مدل فرایندی را درست بازسازی نمی‌کند.



شکل (۶-۵) مدل فرایندی که ثبت رخداد ABECDG می‌سازد

۶-۳-۲- مثال دوم

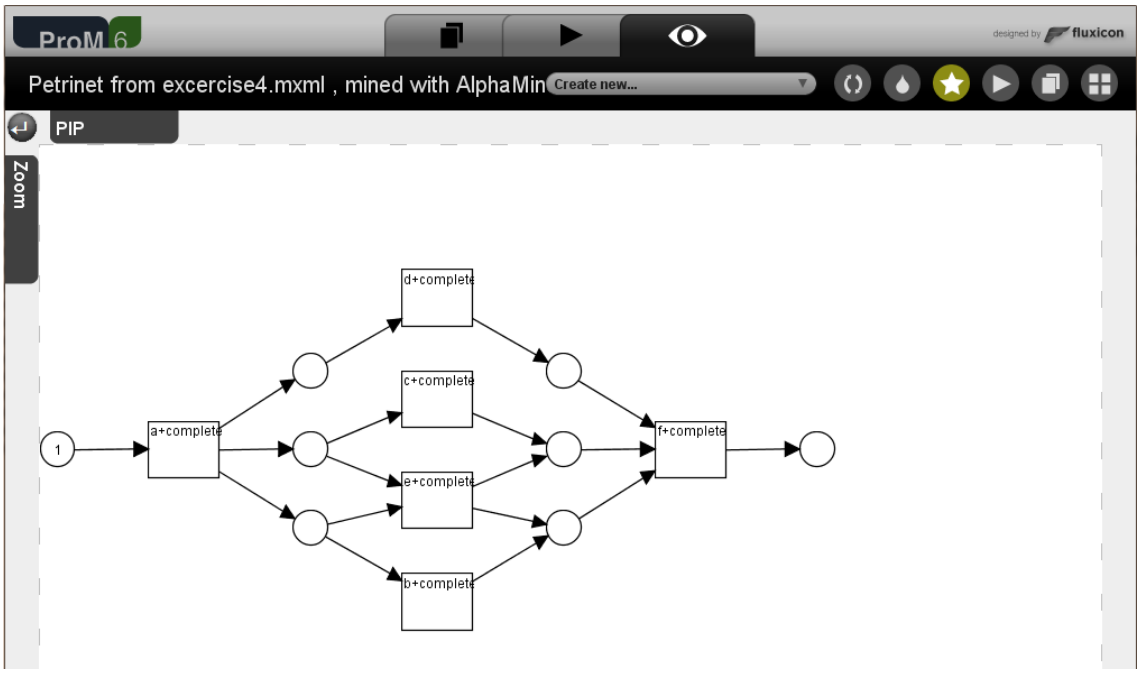
برای دومین توصیف مدل فرایندی، WF-net شکل (۶-۵) را در نظر بگیرید. با توجه به آنچه در فصل پیش بیان شد، مجموعه همه‌ی دنباله رخدادها، مجموعه‌رفتارهای اساسی و ثبت‌رخداد بهینه‌ی این مدل فرایندی بصورت زیر بدست می‌آید.

L(G) = {abcdf, acbdf, abdcf, acdbf, adef, aedf, adcbf, adbcf}

UF ={ac, ab, ad, ae, cf, bf, df, ef, cb, bc, cd, dc, ed, de, bd, db}

OPlog= {abcdf, adef, aedf, acbdf, abdcf, acdbf}

حال با استفاده از Oplog به بازسازی مدل‌فرایندی می‌پردازیم. شکل (۶-۶) نتیجه بدست آمده از نرم‌افزار Prom را نشان می‌دهد.



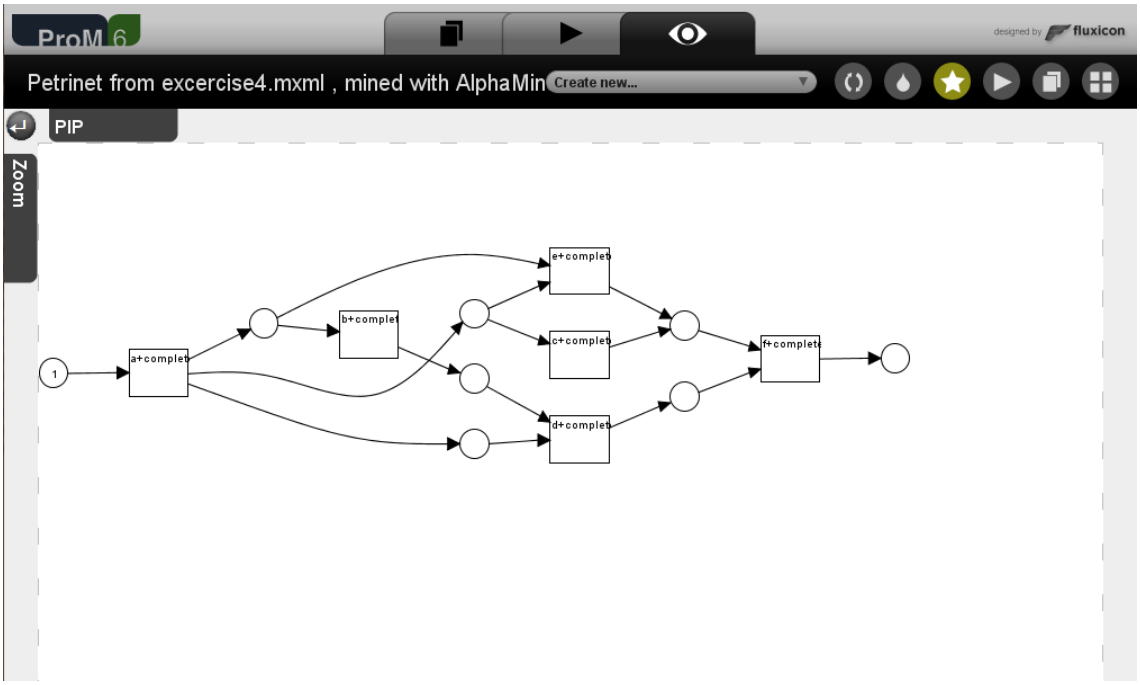
شکل(۶-۶) بازسازی مدل فرایندی شکل (۵-۶) با ثبت‌رخداد بهینه

همان‌طور که در شکل(۶-۶) مشاهده می‌کنید مدل فرایندی بدرستی بازسازی شد. در اینجا نیز باید توضیح داده‌شود که تعداد اعضای مجموعه Oplog تخمینی از تعداد اعضای ثبت رخدادبهینه است. اما اگر از الگوریتم-های شاخه و حد استفاده کنیم باز هم به همین جواب می‌رسیم. بنابراین می‌توان نتیجه گرفت که Oplog دقیقا ثبت‌رخداد بهینه را نشان می‌دهد. حال برای اینکه کارایی ثبت رخداد بهینه را نشان دهیم، سعی بر بازسازی مدل فرایندی با استفاده از ثبت رخداد زیر می‌کنیم.

eventlog۲= { abcdf, adef, aedf, acbdf, abdcf }

این ثبت‌رخداد دارای ۵ عضو می‌باشد. در حالی که الگوریتم پیشنهادی نشان می‌دهد که ثبت‌ی بهینه باید شامل حداقل ۶ عضو باشد.

شکل (۶-۷) نتیجه بدست آمده از نرم‌افزار Prom را نشان می‌دهد. با توجه به شکل مجموعه eventlog۲ رفتارهای اساسی مدل فرایندی شکل (۵-۶) را نشان نمی‌دهد.



شکل (۶-۷) ساخت مدل فرایندی توسط ثبت‌رخداد eventlog۲= { abcdf, adef, aedf, acbdf, abdcf }

همانطور که در شکل (۶-۷) مشخص است، اگر ما ثبت‌رخداد خود را با ۵ دنباله‌رخداد در نظر بگیریم، مدل فرایندی بدرستی بازسازی نمی‌شود.

۶-۴- نتیجه‌گیری

در این فصل به بازسازی چند مدل فرایندی پرداختیم و اهمیت ثبت‌رخداد را در بازسازی یک مدل فرایندی نشان دادیم. ثبت‌رخدادی که کامل نباشد نمی‌تواند یک مدل فرایندی را بدرستی بازسازی کند. ما توانستیم با بدست آوردن رفتار اساسی یک مدل فرایندی به یک ثبت‌رخداد بهینه دست پیدا کنیم.

همچنین در این فصل نشان دادیم که اگر تعداد اعضای ثبت‌رخداد کمتـر از آنچـه کـه مـا در فصـل پـیش، تخمین زده بودیم باشد، مدل فرایندی بدرستی بازسازی نمی‌شود. علاوه بر این نشان دادیم که علاوه بـر تعـداد اعضای ثبت‌رخداد بهینه، خود ثبت‌رخداد بهینه درست بدست آمده است.

فصل هفتم

نتیجه گیری و کارهای آتی

۷-۱- نتیجه‌گیری

در این پایان‌نامه، شیوه‌ای برای تعیین تعداد دنباله‌رخدادهای در یک ثبت‌رخداد بهینه برای یک مدل-فرایندی خاص (شبکه‌های جریان کار) بیان شد. در سال‌های اخیر، تحقیقات و بررسی‌های زیادی بر روی کامل بودن یک ثبت رخداد انجام شده است و توصیفات مختلفی برای کامل بودن یک ثبت‌رخداد در نظر گرفته شده است. اما تعیین ثبت‌رخدادی که بهینه باشد، کمتر به آن توجه شده است.

در ابتدا برای حل مساله یکسری فرضیات در نظر گرفته شد. شبکه‌های جریان کار (Wf-net) بعنوان یک مدل‌فرایندی مورد بررسی قرار گرفت. یکی از فرضیات مهم در اختیار داشتن ثبت‌رخداد کامل از مدل‌فرایندی مورد نظر بود. برای این‌کار، با انجام مهندسی معکوس بر روی یک WF-net مجموعه‌ی همه دنباله‌رخدادهای ممکن (که یک ثبت‌رخداد کامل محسوب می‌شود) را بدست آوردیم. برای اینکار از زبان‌های صوری استفاده شد و با بیان یک گرامر نامحدود، مجموعه‌ی همه دنباله‌رخدادهای ممکن از WF-net استخراج شد.

در ادامه به بررسی رفتارهای اساسی یک WF-net پرداخته شد و این رفتارها بصورتی نمایش داده شد (مجموعه‌ای از جفت رخدادهای متوالی) تا بتواند در ثبت‌رخداد کامل مورد بررسی قرار بگیرد. سپس با بیان یک الگوریتم حریصانه پیشنهادی به حل بهینگی پرداختیم. هر چند این الگوریتم مساله بهینگی را دقیق حل نمی-کند و یک تخمینی از حل بهینگی ارائه می‌دهد اما دارای پیچیدگی زمانی چندجمله‌ای است و در سیستم‌هایی که دارای ثبت‌رخداد های بزرگی هستند بهتر عمل می‌کند. در عین حال برای بدست آوردن حل بهینه دقیق می‌توان از راهبرد شاخه و حد نیز استفاده کرد. اما این راهبرد برای این مساله دارای پیچیدگی زمانی نمایی است و در سیستم‌هایی که دارای ثبت‌رخدادهای بزرگ هستند، خوب عمل نمی‌کند و وقت زیادی را از سیستم می‌گیرند.

در این پایان‌نامه مبنای کار خود را الگوریتم آلفا قرار دادیم و نتایج بدست آمده را با استفاده از این الگوریتم و نرم‌افزار Prom نشان دادیم.

بطور کلی می‌توان دستاوردهای این پایان‌نامه را بطور زیر خلاصه کرد:

۱- تعیین یک ثبت‌رخداد کامل برای یک مدل فرایندی: تعیین یک ثبت‌رخداد کامل همواره از مسایلی بوده است که با چالش‌های زیادی روبرو بوده است. در این پایان نامه با استفاده از زبان‌های صوری و در نظر گرفتن یک مدل فرایندی خاص، همه‌ی دنباله‌رخدادهایی ممکن را بعنوان یک ثبت‌رخداد کامل در نظر گرفته شد.

۲- تعیین رفتارهای اساسی یک مدل فرایندی: همواره مدل‌های فرایندی از ساختارهای بلوکی خاصی تشکیل شده‌اند. در این پایان‌نامه با در نظر گرفتن این ساختارهای بلوکی، رفتارهای اساسی یک مدل فرایندی شناسایی شد و این رفتارهای اساسی بصورت مجموعه‌ای از جفت رخدادهای متوالی نشان داده شد.

۳- تعیین تعداد دنباله رخدادها در ثبت‌رخداد بهینه: همواره با یک ثبت‌رخداد کامل می‌توان یک مدل فرایندی را بازسازی کرد. در این پایان‌نامه با استفاده از ثبت‌رخداد کامل بدست آمده و رفتارهای اساسی، به بهینه‌سازی ثبت‌رخداد پرداخته می‌شود. البته الگوریتم پیشنهادی تخمینی از تعداد دنباله رخدادها دری ثبت-رخداد بهینه را بدست می‌آورد و الگوریتم‌هایی که بتواند جواب دقیق را بدست آورد، را نیز توضیح می‌دهد.

۷-۲- پیشنهادها

همواره یک سیستم دارای ویژگی‌های زیادی است که مدل‌سازی همه‌ی ویژگی‌های آن امکان پذیر نیست و همواره الگوریتم‌های فرایندکاوی، فرضیات زیادی را بر روی مساله در نظر می‌گیرند. این پایان‌نامه نیز برای حل مساله، فرضیات زیادی را در نظر گرفته است. بطور مثال نبود حلقه، کارهای تکراری و مکان‌های ضمنی و یا همچنین امن بودن مدل فرایندی و ...، از فرضیاتی است که برای حل مساله در نظر گرفته شده است. این فرضیه‌ها محدودیت‌هایی بر روی سیستم ایجاد می‌کند. در آینده می‌توان با کم کردن این محدودیت‌ها می‌توان

به کارایی این الگوریتم‌ها افزود. کم کردن این محدودیت‌ها بر روی الگوریتم‌های پایه و اساسی همچون الگوریتم آلفا انجام شده است و الگوریتم‌های جامع‌تری جایگزین آن‌ها شده‌اند.

عدم وجود حلقه از محدودیت‌هایی است که برای حل مساله در این پایان‌نامه در نظر گرفته شده است. برای حل این محدودیت می‌توان تغییراتی در مدل فرایندی بوجود آورد و یا اینکه الگوریتم تولید ثبت رخداد کامل را بطور پایان‌پذیر تعریف کرد.

در این پایان‌نامه شناسایی رفتارهای اساسی بر روی WF-netها انجام شده است. WF-netها نوعی از مدل فرایندی هستند که دارای محدودیت‌هایی هستند. در آینده می‌توان شناسایی رفتارهای اساسی یک مدل فرایندی را تعمیم داد و دسته‌ی بزرگتری از مدل فرایندی را در نظر بگیریم.

همچنین الگوریتمی که برای تعیین تعداد دنباله رخدادها در ثبت‌رخداد بهینه بکار رفت. مساله را بطور تخمینی حل کرد. تعیین الگوریتمی که بتواند بطور دقیق این مساله را در زمان چند جمله‌ای حل کند، از دیگر چالش‌هایی است که در آینده می‌تواند مورد بررسی قرار گیرد.

مراجع:

[١] W. M. P. van der Aalst. The application of Petri nets to workfow management. The Journal of Circuits, Systems and Computers, ٨(١):٢١-٦٦, ١٩٩٨.

[٢] D. Georgakopoulos, M. Hornick, and A. Sheth. An overview of workfloow management: from process modeling to workflow automation infrastructure. Distributed and Parallel Databases, ٣(٢):١١٩-١٥٣, ١٩٩٥.

[٣] Alexander Schill, Christian Mittasch. CodAlf: A decentralized workflow management system on top of OSF DCE and DC plus plus. Proc Of Int'l Symposium on Autonomous Decentralized Systems'٩٧, ١٩٩٧:٢٠٥-٢١٢.

[٤] Lectures on Petri Nets I: Basic Models, Lecture Notes in Computer Science, vol. ١٤٩١, W. Reisig and G. Rozenberg, eds., Berlin: Springer-Verlag, ١٩٩٨.

[٥] van der Aalst, W.M.P., Weijters, A.J.M.M. (eds.): Process Mining, Special Issue of Computers in Industry, vol. ٥٣(٣). Elsevier Science Publishers, Amsterdam (٢٠٠٤).

[٦] de Medeiros, A.K.A., van Dongen, B.F., van der Aalst, W.M.P., Weijters, A.J.M.M.: Process Mining for Ubiquitous Mobile Systems: An Overview and a Concrete Algorithm. In: Baresi, L., Dustdar, S., Gall, H.C., Matera, M. (eds.) UMICS ٢٠٠٤. LNCS, vol. ٣٢٧٢, pp. ١٥١–١٦٥. Springer, Heidelberg (٢٠٠٤)

[٧] van der Aalst, W.M.P., Weijters, A.J.M.M., Maruster: Workflow Mining: Discovering Process Models from Event Logs. IEEE Transactions on Knowledge and Data Engineering ١٦(٩), ١١٢٨–١١٤٢ (٢٠٠٤

[٨] van Dongen, B.F., Alves de Medeiros, A.K., Wen, L.: Process mining: Overview and outlook of petri net discovery algorithms. T. Petri Nets and Other Models of Concurrency ٥٤٦٠, ٢٢٥–٢٤٢ (٢٠٠٩)

[٩] http://www.aiim.org/wfmc.

[١٠] David Hollingsworth, WfMC. The workflow reference model. ١٩٩٤.

[١١] C. Plesums: Introduction to Workflow Management, Workflow Management Handbook, ٢٠٠٣.

[١٢] James G. Kobielus, Workflow Strategies,book, page: ٣٩, ١٩٩٧.

[١٣] James G. Kobielus, Workflow Strategies, book, page: ٢٠-٢١, ١٩٩٧.

[١٤] An Introduction toWorkflow Management SystemsCenter for Technology in Government University at Albany / SUNY ١٩٩٧ Center for Technology in GovernmentThe Center grants permission to reprint this document provided that it is printed in its entirety.

[١٥] James G. Kobielus, Workflow Strategies, book, Chapter ١ and chapter ٢ , page ١٣ and page ٤٤.

[١٦] Jiacun Wang ; Tepfenhart, W. ; Rosea, D. ,"EPC Workflow Model to WIFA Model Conversion". In: ٢٠٠٦ IEEE International Conference on Systems, Man, and Cybernetics, Taipei, Taiwan, pp. ٢٧٥٨-٢٧٦٣.

[١٧] Ferdian , A Comparison of Event-driven ProcessChains and UML Activity Diagram for Denoting Business Processes, chapter ٣ Project Work submitted by Information and Communication Systems, ٢٠٠١.

[١٨] B. Curtis, M. Kellner, and J. Over, "Process modeling," Communications of the ACM, vol. Vol.٣٥ No. ٩, pp. ٧٥-٩٠, ١٩٩٢.

[١٩] Carl Adam Petri and Wolfgang Reisig (٢٠٠٨) Petri net. Scholarpedia, ٣(٤):٦٤٧٧.

[٢٠] W. M. Wonham, "A control theory for discrete event systems", Advanced Computing Concepts and Techniques in Control Engineering, M. J. Denham and A. J. Laub, Eds. New York: Springer Verlag, pp. ١٢٩-١٦٩, ١٩٨٩.

[٢١] P. J. Ramadge and W. M. Wonham, "The control of discrete event systems", Proceedings IEEE, vol. ٧٧, no. ١, pp. ٨١-٩٨, ١٩٨٩.

[٢٢] Ali A. Pouyan, "A coputer-based Petri net approach to modelling AMSs", Ph.D. thesis, Swinburne University of Technology, Melbourne, Australia, ١٩٩٧.

[٢٣] Ali A. Pouyan and E. Shayan, "Modelling, analysis and simulation of flexible manufacturing systems", Proceedings The Third International Conference on Modelling and Simulation, Melbourne, Australia, pp. ٥٠٥-٥١١, ١٩٩٧.

[٢٤] Y. Yaw, "Analysis and synthesis of distributed systems and protocols", Ph.D. dissertation, Dept. of Electrical Engineering and Computer Science, University ofbCalifornia, Berkeley, ١٩٨٧.

[٢٥] Y. Li and C. M. Woodside, "Complete decomposition of stochastic Petri nets representing generalised service networks", IEEE Transactions on Computers, vol.٤٤, no. ٤, pp. ٥٧٧-٥٩٢،١٩٩٥.

[٢٦] M. Li and N. D. Georganas, "Exact parametric analysis of stochastic Petri nets", IEEE Transactions on Computers, vol. ٤١, no. ٩, pp. ١١٧٦-١١٨٠،١٩٩٢.

[٢٧] Jensen, K.: An Introduction to the Theoretical Aspects of Coloured Petri Nets. In: de Bakker, J.W., de Roever, W.P., Rosenberg, G. (eds.): A Decade of Concurrency. LNCS ٨٠٣. Berlin, Heidelberg, New York: Springer-Verlag, ١٩٩٤,pp. ٢٣٠-٢٧٢

[٢٨] M. Notomi and T. Murata, "Hierarchical reachability graph of bounded Petri nets for concurrent software analysis", IEEE Transactions on Software Engineering,vol. ٢٠, no. ٥, ١٩٩٤.

[٢٩] W. J. Yeh and M. Young, "Compositional reachability analysis using process algebra", Proceedings ٤th ACM symposium on Software Testing, Analysis, and Verification, pp. ٤٩-٥٩, ١٩٩١.

[٣٠] R. Devillers, "The semantics of capacities in P/T nets", Advances in Petri nets, Lecture Notes in Computer Science, G. Rozenberg, Ed. Berlin: Springer Verlag, pp. ١٢٨-١٥٠, ١٩٨٩.

[٣١] M. D. Jeng and F. DiCesare, "Synthesis using resource control nets for modelling shared resource systems", IEEE Transaction on Robotics and Automation, vol. ١١,no. ٣, pp. ٣١٧-٣٢٧, ١٩٩٥.

[٣٢] A. Giua and F. DiCesare, "Decidability and closure properties of weak Petri net languages in supervisory control", IEEE Transactions on Automatic Control, vol. ٤٠, no. ٥, pp. ٩٠٦-٩١٠, ١٩٩٥.

[٣٣] R. Kumar and L. E. Holloway, "Supervisory control of deterministic Petri nets with regular specification languages", IEEE Transactions on Automation and Control, vol. ٤١, no. ٢, pp. ٢٤٥-٢٤٩, ١٩٩٦.

[٣٤] Liu Fengduan,The Fundamental of Behavioral Sciences,Fudan University Press,١٩٩١,pp١-٢.

[٣٥] Wil van der Aalst and Kees van Hee, "Workflow Management: Models, Methods and Systems", ISBN ٠٢٦٢٠٢٢٨٩١, MIT Press, December, ٢٠٠٠.

[٣٦] Kamel Barkaoui, Rahma Ben Ayed, and Zohra sbai, "Workflow Soundness Verification based on Structure Theory of Petri nets", International Journal of Computing & Information Sciences, Vol. ٥, No.١, April, ٢٠٠٧.

[٣٧] Laoutaris N, Stavrakakis I (٢٠٠٢), "Intra stream synchronization for continuous media streams: a survey of playout schedulers", IEEE Netw Mag ١٦(٣):٣٠–٤٠.

[٣٨] Murat UZAM, Anthony H. JONES, A New Petri-Net-Based Synthesis Technique for Supervisory Control of Discrete Event Systems, Turk J ElecEngin. ١٠ (٢٠٠٢) ٨٥-١٠٩.

[٣٩] Kurt Jensen, "Telephones System", Aarhus University, Denmark (kjensen@daimi.au.dk).

[٤٠] K.M. van Hee. Information System Engineering: a Formal Approach. Cambridge University Press, ١٩٩٤.

[٤١] P. Freedman, "Time, Petri nets and robotics", IEEE Transactions on Robotics and Automation, vol. ٧, no. ٤, pp. ٤١٧-٤٣٣, ١٩٩١.

[٤٢] J. P. Tsai, S. J. Yang, and Y. H. Chang, "Timing constraint Petri nets and their application to schedulability analysis of real time system specifications", IEEE Transactions on Software Engineering", vol. ٢١, n٠. ١, pp. ٣٢-٤٩, ١٩٩٥.

[٤٣] M. MENASCHE, B. BERTHOMIEU, "Time Petri nets for analyzing and verifying time dependent protocols", Protocol Specification, Testing and Verification III, ١٦١–١٧٢،١٩٨٣.

[٤٤] B. BERTHOMIEU, M. DIAZ, "Modeling and verification of time dependent systems using time Petri nets", IEEE Transactions on Software Engineering, vol. ١٧(٣),٢٥٩–٢٧٣, March ١٩٩١.

[٤٥] S. Ramaswamy and K. P. Valavanis, "Hierarchical Time-Extended Petri Nets (HEPN's) based error identification and recovery for multilevel systems", IEEE transactions on Computers, vol. ٢٠, no. ١, pp. ١٦٤-١٧٥, ١٩٩٦.

[٤٦] Ali A. Pouyan and E. Shayan, "Synthesis and analysis of a flexible manufacturing cell using Petri nets", Proceedings The ٧th International Conference on Manufacturing Engineering, Cairns, Australia, pp. ٦١١-٦٢٠, ١٩٩٧.

[٤٧] Ali A. Pouyan and E. Shayan, "A Petri net-based approach to logic controller design", Proceedings Advanced Engineering & Design, Automation Technology, Jakarta, Indonesia, pp. ٤٥-٥٢, ١٩٩٦.

[٤٨] A. Pouyan and E. Shayan, "State of the art in application of Petri nets in automated manufacturing systems", Proceedings The ٦th International Conference on Manufacturing Engineering, Melbourne, Australia, vol. ١, pp. ١٢٩-١٣٨, ١٩٩٥.

[٤٩] M. M. Hanna, A. Buck, and R. Smith, "Fuzzy Petri nets with neural networks to model products quality from a CNC-milling machine center", IEEE Transactions on System, Man, and Cybernetics, vol. ٢٦, no. ٥, pp. ٦٣٨-٦٤٥, ١٩٩٦.

[٥٠] G. Chiola, M. A. Marsan, G. Balbo, and G. Conte," Generalised stochastic Petri nets a definition at the net level and its applications ", IEEE Transactions on Software Engineering", vol. ١٩, no. ٢, pp. ٨٩-١٠٧, ١٩٩٣.

[٥١] R. Agrawal, D. Gunopulos, and F. Leymann, "Mining Process Models from Workflow Logs," Proc. Sixth Int'l Conf. Extending Database Technology, pp. ٤٦٩-٤٨٣, ١٩٩٨.

[٥٢] W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster, "Workflow Mining: Which Processes can be Rediscovered?"BETA Working Paper Series, WP ٧٤, Eindhoven Univ. of Technology, Eindhoven, ٢٠٠٢.

[٥٣] L. Maruster, A.J.M.M. Weijters, W.M.P. van der Aalst, and A. van den Bosch, "Process Mining: Discovering Direct Successors in Process Logs," Proc. Fifth Int'l Conf. Discovery Science (DiscoveryScience ٢٠٠٢), pp. ٣٦٤-٣٧٣, ٢٠٠٢.

[٥٤] van der Aalst W.M.P. [et al.] Business Process Mining: An Industrial Application [Journal] // Information Systems ٣٢(٥). - ٢٠٠٧. - pp. ٧١٣-٧٣٢.

[٥٥] W.M.P. van der Aalst and K.M. van Hee, Workflow Management: Models, Methods, and Systems. Cambridge, Mass.: MIT Press, ٢٠٠٢.

[٥٦] Workflow Handbook ٢٠٠١, Workflow Management Coalition, L. Fischer, ed. Lighthouse Point, Fla.: Future Strategies, ٢٠٠١.

[٥٧] S. Jablonski and C. Bussler, Workflow Management: Modeling Concepts, Architecture, and Implementation. London: Int'l Thomson Computer Press, ١٩٩٦.

[٥٨] F. Leymann and D. Roller, Production Workflow: Concepts and Techniques. Upper Saddle River, New Jersey, Prentice-Hall PTR, ١٩٩٩.

[٥٩] Lectures on Petri Nets I: Basic Models, Lecture Notes in Computer Science, vol. ١٤٩١, W. Reisig and G. Rozenberg, eds., Berlin: Springer-Verlag, ١٩٩٨.

[٦٠] Wen, L., Wang, J., van der Aalst, W.M.P., Wang, Z., Sun, J.: A Novel Approach for Process Mining Based on Event Types. BETA Working Paper Series, WP ١١٨, Eindhoven University of Technology, Eindhoven (٢٠٠٤): An Overview and a Concrete Algorithm. In: Baresi, L., Dustdar, S., Gall, H.C., Matera, M. (eds.)

[٦١] Wen, L., van der Aalst, W.M.P., Wang, J., Sun, J.: Mining process models with non-free-choice constructs. Data Mining and Knowledge Discovery ١٥(٢), ١٤٥–١٨٠ (٢٠٠٧).

[٦٢] Wen, L.,Wang, J., Sun, J.: Mining invisible tasks from event logs. In: Dong, G., Lin X., Wang, W., Yang, Y., Yu, J.X. (eds.) APWeb/WAIM ٢٠٠٧. LNCS, vol. ٤٥٠٥, pp. ٣٥٨–٣٦٥. Springer, Heidelberg (٢٠٠٧)

[٦٣] Weijters, A.J.M.M., van der Aalst, W.M.P.: Rediscovering Workflow Models from Event-Based Data using Little Thumb. Integrated Computer-Aided Engineering ١٠(٢), ١٥١–١٦٢ (٢٠٠٣).

[٦٤] Weijters, A.J.M.M., van der Aalst, W.M.P., Alves de Medeiros, A.K.: Process Mining with HeuristicsMiner Algorithm. BETA Working Paper Series, WP ١٦٦, Eindhoven University of Technology, Eindhoven (٢٠٠٦)

[٦٥] de Medeiros, A.K.A.: Genetic Process Mining. PhD thesis, Eindhoven University of Technology, Eindhoven, The Netherlands (٢٠٠٦).
[٦٦] de Medeiros, A.K.A., Weijters, A.J.M.M., van der Aalst, W.M.P.: Genetic Process Mining: an Experimental Evaluation. DataMining and Knowledge Discovery ١٤(٢), ٢٤٥–٣٠٤ (٢٠٠٧).
[٦٧] Kindler, E., Rubin, V., Sch¨afer, W.: Process Mining and Petri Net Synthesis. In: Eder, J., Dustdar, S. (eds.) BPM Workshops ٢٠٠٦. LNCS, vol. ٤١٠٣, pp. ١٠٥–١١٦. Springer, Heidelberg (٢٠٠٦).
[٦٨] Badouel, E., Bernardinello, L., Darondeau, P.: The Synthesis Problem for Elementary Net Systems is NP-complete. Theoretical Computer Science ١٨٦(١-٢), ١٠٧–١٣٤ (١٩٩٧).
[٦٩] Badouel, E., Darondeau, P.: Theory of regions. In: Reisig, W., Rozenberg, G. (eds.) APN ١٩٩٨. LNCS, vol. ١٤٩١, pp. ٥٢٩–٥٨٦. Springer, Heidelberg (١٩٩٨).
[٧٠] Cortadella, J., Kishinevsky, M., Lavagno, L., Yakovlev, A.: Synthesizing Petri Nets from State-Based Models. In: Proceedings of the ١٩٩٥ IEEE/ACM International Conference on Computer-Aided Design (ICCAD ١٩٩٥), pp. ١٦٤–١٧١. IEEE Computer Society, Los Alamitos (١٩٩٥).
[٧١] Cortadella, J., Kishinevsky, M., Lavagno, L., Yakovlev, A.: Deriving Petri Nets from Finite Transition Systems. IEEE Transactions on Computers ٤٧(٨), ٨٥٩–٨٨٢ (١٩٩٨).
[٧٢] Sherzod Turaev, “Petri Net Controlled Grammars “, PhD Dissertation, ٢٠١٠, chapter ٤, page ٥٠.
[٧٣] J.L. Peterson. Petri net theory and modeling of systems. Prentice-Hall, Englewood Cliffs, NJ, ١٩٨١.
[٧٤] M. Jantzen. Language theory of Petri nets. In Advances in Petri Nets, volume ٢٥٤-I of LNCS, pages ٣٩٧–٤١٢. Springer, ١٩٨٧.
[٧٥] Javir esparza, “ Free choice petri net ” book, chapter ٢, page ١٩-٢٢, ١٩٩٥.
[٧٦] www.processmining.org.

Abstract

Business process management includes simulating, analyzing and planning a business system. Today, information technology is inclined to workflow management. A workflow management system is a computer system that manages and defines a series of tasks within an organization to produce a final outcome or outcomes. In last decade, workflow technology was used in complex information system. For simulating workflow system, process model is used. Constructing a process model from event logs is one of important field in business process management.

This thesis survey useful information in event log for constructing a process model. In other words, the goal of this thesis is to investigate minimum number of logs required to construct a process model. Petri net is one of the important and convenient models in this area and this thesis used this model to investigate novel solution.

There are many algorithms to construct process model from event log, by attention to workflow model. Completeness event log is very important for construct process model. Whatever knowledge is extracted from event log be more complete, process model will be constructed more accurate and management if system will be better. But extra information in event logs made reduction efficiency in discover algorithm. So consider an optimum event logs made discover algorithm more efficiency. In this thesis, we consider α-algorithm as base algorithm and proposed algorithm get optimum event logs based α-algorithm

Keywords: petri net, process mining, event log