قابلیت‌های نرم‌افزار *SharifSAN* در ارزیابی کارایی سیستم‌های رایانه‌یی و مشخصات طراحی و پیاده‌سازی آن

محمد عبداللهی ازگمی (دانشجوی دکتری)
علی موقر رحیم‌آبادی (دانشیار)
دانشکده‌ی مهندسی کامپیوتر، دانشگاه صنعتی شریف

شبکه‌های فعالیت تصادفی[1]، یکی از بسط‌های شبکه‌های پتری[2] است که برای مدل‌سازی و تحلیل جنبه‌های کارکردی[3] و عملیاتی سیستم‌ها قدرت و انعطاف بالایی فراهم می‌کند.[1] تا کنون چندین ابزار مدل‌سازی بر اساس تعریف قدیمی[1و2] شبکه‌های فعالیت تصادفی طراحی شده است، اما اخیراً تعریف جدیدی از این مدل ارائه شده[3] که برخی از مشکلات تعریف قدیمی را برطرف ساخته است. این تعریف، مبنای طراحی یک نرم‌افزار مدل‌سازی و تحلیل با نام *SharifSAN* است که به‌منظور فراهم‌سازی امکانات یکپارچه برای درستی‌یابی[4] و ارزیابی کارایی، قابلیت اتکاء و قابلیت کارایی سیستم‌های رایانه‌یی و ارتباطی طراحی شده است. *SharifSAN* نخستین ابزار مدل‌سازی است که براساس تعریف جدید شبکه‌های فعالیت تصادفی ایجاد شده و امکانات کاملی برای ساخت مدل‌ها از طریق واسط کاربر گرافیکی، درستی‌یابی سیستم‌ها با استفاده از منطق زمانی، ارزیابی کارایی با روش‌های تحلیل نمودار دسترسی[5] و شبیه‌سازی گسسته ـ رخداد فراهم می‌کند. در مقاله‌ی منتشره‌ی قبلی[4] به تشریح قابلیت‌های تحلیل جنبه‌های کارکردی سیستم‌ها و امکانات درستی‌یابی این ابزار پرداختیم. در این نوشتار نیز ضمن تشریح قابلیت‌های آن برای ارزیابی کارایی، به معرفی مشخصات، ویژگی‌ها، طراحی و پیاده‌سازی و در نهایت کاربردهای آن می‌پردازیم. تا کنون از این ابزار در چندین پروژه‌ی پژوهشی در زمینه‌ی شبکه‌های سریع استفاده شده است، که در اینجا به آنها اشاره خواهد شد.

مقدمه

یکی از شاخه‌های مهم رشته‌ی کامپیوتر، مدل‌سازی و تحلیل سیستم‌هاست که هدف آن ارائه‌ی سیستم به نحوی است که قابلیت تحلیل خودکار را داشته باشد و درستی‌یابی سیستم و مطابقت آن با نیازهای کارکردی یا اندازه‌گیری معیارهایی درباره‌ی جنبه‌های عملیاتی سیستم امکان‌پذیر شود. پیشرفت گسترده‌ی فناوری باعث ایجاد سیستم‌های نوین و پیچیده‌یی شده که‌به‌ویژه لازم است نسبت به عملکرد سیستم‌های حساس و بحرانی که خرابی یا عدم‌کارایی آنها زیان‌های غیر قابل جبرانی برای استفاده کنندگان به همراه دارد، مطمئن بود. این اطمینان باید قبل از کاربرد واقعی سیستم‌ها حاصل شده باشد. علم مدل‌سازی به پژوهشگران و طراحان امکان می‌دهد که با ساخت مدلی از سیستم به تحلیل خودکار نیازهای کارکردی و عملیاتی آنها بپردازند. از این‌رو، به‌عنوان یکی از گام‌های طراحی سیستم‌های پیچیده‌ی امروزی مورد استفاده قرار می‌گیرد.

برای مدل‌سازی سیستم‌ها روش‌های صوری مختلفی پیشنهاد شده‌اند. سادگی و قابلیت تحلیل شبکه‌های پتری (PN) ـ که از قابلیت نمایش گرافیکی برخوردار است ـ باعث شده است که کاربردهای گسترده‌یی در مدل‌سازی و تحلیل سیستم‌ها داشته باشد. به‌منظور افزایش قدرت، کاربردپذیری و قابلیت انعطاف مدل‌سازی، بسط‌های مختلفی برای شبکه‌های پتری پیشنهاد شده، که شبکه‌های فعالیت تصادفی (SAN)، یکی از آنهاست[1و2] و از نظر قدرت و انعطاف، برتر از بسط‌هایی نظیر شبکه‌های پتری تصادفی تعمیم یافته[6] (GSPN) است.[5]

استفاده از روش‌های صوری و درستی‌یابی جنبه‌های کارکردی سیستم‌ها، اطمینان نسبت به درستی مشخصات و طرح سیستم‌ها را افزایش می‌دهد ونگرشی برای اجتناب از خطاست که اخیراً در طراحی سیستم‌های قابل اطمینان به شدت مورد توجه قرار گرفته است. برای درستی‌یابی خودکار، نیاز به ساخت مدلی از سیستم، و نیز روشی مبتنی بر ریاضیات برای بیان خصوصیت‌ها یا جنبه‌های کارکردی آن است. منطق زمانی[7] یکی از این روش‌هاست.[6] همچنین روش‌های ارزیابی برای تحلیل جنبه‌های عملیاتی و وابسته به زمان، نظیر اطمینان‌پذیری و قابلیت کارایی، مورد استفاده قرار می‌گیرند. اگرچه رده‌های مشخصی از مدل‌های مبتنی بر شبکه‌های پتری، با روش‌های تحلیل عددی[8] ارزیابی می‌شوند، در حالت کلی

و برای ارزیابی همه‌ی مدل‌های ایجاد شده با شبکه‌های پتری و بسط‌های آن، شبیه‌سازی گسسته ـ رخداد[۹][۷] قابل استفاده است.

برای مدل‌سازی با شبکه‌های پتری و بسط‌های آن، نیاز به یک ابزار نرم‌افزاری است، تا به مدل‌ساز امکان ساخت مدل را داده و قابلیت‌های تحلیل خودکار جنبه‌های کارکردی و ارزیابی معیارهای عملیاتی مدل‌ها را فراهم سازد. براساس تعریف قدیمی مدل شبکه‌های فعالیت تصادفی[۱و۲]، تاکنون ابزارهایی نظیر METASAN[۸]، UltraSAN [۹و۱۰] و Mobius [۱۱] ایجاد شده‌اند. طبق تعریف قدیمی شبکه‌های فعالیت تصادفی، خوش‌رفتار بودن[۱۰] مدل موردنیاز بود. در حالت کلی، بررسی خوش‌رفتار بودن مدل که باید توسط ابزار مدل‌سازی انجام شود، غیرقطعی، و از نظر محاسباتی پیچیده است. نرم‌افزار *SharifSAN* که در این نوشتار به تشریح قابلیت‌های آن می‌پردازیم، براساس تعریفی جدید از شبکه‌های فعالیت تصادفی است، که عاری از مشکل ذکر شده است. همچنین تعریف جدید، مفاهیم اولیه‌ی کمتری دارد و از نظر نمایش گرافیکی نیز ساده‌تر است. این دو تعریف تفاوت‌های دیگری نیز دارند.[۳و۱۲]

در نوشتار منتشره‌ی قبلی به تشریح قابلیت‌های تحلیل جنبه‌های کارکردی سیستم‌ها و امکانات درستی‌یابی *SharifSAN* پرداختیم. اکنون در این نوشتار به تشریح قابلیت‌های ارزیابی کارایی آن می‌پردازیم. تاکنون از این ابزار در چندین پروژه‌ی پژوهشی در زمینه‌ی شبکه‌های سریع[۱۱] استفاده شده است که به برخی از آنها اشاره خواهیم کرد.

ارزیابی کارایی با شبکه‌های فعالیت تصادفی

ارزیابی کارایی به‌منظور محاسبه‌ی معیارهای مربوط به عملکرد سیستم انجام می‌شود. یکی از کاربردهای اصلی شبکه‌های پتری، در زمینه‌ی ارزیابی کارایی سیستم‌هاست. معیارهای کارایی که در ادامه مطرح می‌شوند، معیارهایی هستند که مقادیر آنها در حالت پایدار[۱۲] سیستم مورد نظر است. تحلیل حالت پایدار در مورد رده‌هایی از شبکه‌های پتری و بسط‌های آن که قابل تبدیل به زنجیره‌های مارکوف[۱۳] هستند، با حل عددی زنجیره‌ی مارکوف ادغام شده، انجام می‌شود. در غیر این‌صورت، شبیه‌سازی گسسته ـ رخداد در حالت پایدار برای محاسبه‌ی معیارهای کارایی قابل استفاده است.

معیارهای کارایی

در یک دسته‌بندی ارائه شده برای معیارهای کارایی[۱۳]، از نظر ما فقط معیارهای زیر که ارتباط مستقیم با عملکرد سیستم دارند، معیارهای کارایی‌اند:

- **سرعت پاسخ‌دهی**

این معیار، سرعت اجرای عملیات در سیستم را نشان می‌دهد. زمان انتظار، زمان پردازش، و طول صف نمونه‌هایی از این معیارها هستند. در حالت‌های مختلف، ممکن است مقادیر میانگین، حداکثر یا حداقل این معیارهای مورد نظر باشد.

- **میزان به کارگیری**[۱۴]

این معیار نشانگر نحوه‌ی به‌کارگیری اجزاء مختلف سیستم است. توان عملیاتی[۱۵] و ضریب بهره‌برداری[۱۶] از جمله‌ی این معیارها هستند. این دسته معیارها با معیارهای سرعت پاسخ‌دهی در تقابل‌اند. زیرا معمولاً زمان پاسخ‌دهی سیستمی که سعی در به کارگیری بیشتر منابع خود داشته باشد طولانی‌تر است.

معیارهای دیگری چون مأموریت‌پذیری[۱۷]، اطمینان پذیری و سودمندی[۱۸] نیز[۱۳] در این دسته‌بندی گنجانده شده‌اند، اما از نظر ما دو مورد نخست معیارهای اطمینان‌پذیری بوده و معیارهای سودمندی ـ نظیر سهولت استفاده و قابلیت فهم ـ اصولاً معیارهای کیفی‌اند و قابل ارزیابی (سنجش) نیستند.

تحلیل عددی حالت پایدار

برای تحلیل مدل‌های شبکه‌های پتری تصادفی و نیز شبکه‌های فعالیت تصادفی که توزیع همه‌ی فعالیت‌های زمانی آن نمایی باشد، می‌توان با به‌دست آوردن زنجیره‌ی مارکوف ادغام شده، از روش‌های حل زنجیره‌های مارکوف زمان پیوسته استفاده کرد. این امر در صورتی امکان‌پذیر است که نمودار دسترسی شبکه، متناهی و اِرگودیک[۱۹] (همه‌سویی) باشد، که برای مقاصد ارزیابی کارایی اغلب این شرط برقرار است. بنابراین حالت‌های پایدار نمودار دسترسی، که در مورد GSPN و SAN، شامل مجموعه‌ی علامت‌گذاری‌هایی[۲۰] است که در آنها انتقال فوری یا فعالیت آنی توانا نباشد، به دست می‌آید و تبدیل به ماتریس نرخ ـ گذر[۲۱] می‌شود. آنگاه همانند تحلیل مدل‌های مارکوف، معادلات توازن[۲۲] برای به دست آوردن احتمالات حالت پایدار حل می‌شوند:

$$\Pi Q = 0$$

$$\Pi e = 1$$

که در آن Π بردار احتمالات پایدار، Q ماتریس نرخ ـ گذر و e یک بردار ستونی یکّه است.

همه‌ی معیارهای کارایی ارائه شده‌ی فوق، با استفاده از این احتمالات قابل محاسبه‌اند. به‌دلیل تُنُکی ماتریس نرخ ـ گذر (Q)،

روش‌های تکراری نظیر روش گاوس ـ سیدل و روش حذف گاوسی برای حل این دستگاه معادلات قابل استفاده‌اند.[۱۴] اما این روش‌ها در مورد مدل‌های بزرگ، از نظر سرعت با مشکل مواجه می‌شوند. از طرفی، ماتریس Q در مورد مدل‌های بزرگ، قابل نگهداری در حافظه‌ی اصلی نیست و نگه‌داری و پردازش آن مشکلات زیادی را ایجاد می‌کند. برای مقابله با این مشکلات، راه حل‌های متعددی ارائه شده است که بدون بهره‌گیری از آنها حل زنجیره‌های مارکوف حاصله از مدل‌های GSPN و SAN بسیار بزرگ، میسر نخواهد بود.

شبیه‌سازی حالت پایدار

اگر مدل شبکه‌های فعالیت تصادفی، به‌دلیل داشتن فعالیت‌های غیرنمایی یا نامتناهی بودن فضای حالت[۲۳] با روش تحلیل حالت پایدار قابل حل نباشد، مجبور به استفاده از روش شبیه‌سازی رایانه‌یی خواهیم بود. برای این منظور، اغلب از روش شبیه‌سازی گسسته ـ رخداد استفاده می‌شود، که روش‌های مختلفی نظیر تکرارهای مستقل[۲۴]، روش تک مسیر[۲۵] و روش باز تولید[۲۶] از نظر سازمان‌دهی اجراهای شبیه‌سازی وجود دارد.[۷] اما برای شبیه‌سازی گسسته ـ رخداد شبکه‌های فعالیت تصادفی، نخست باید مدل را برای مدتی، تا رسیدن به حالت پایدار، بدون جمع‌آوری داده‌های آماری اجرا کرد. آنگاه با توجه به بازه‌ی اطمینان[۲۷] مورد نظر، برای مدت طولانی‌تری مدل را اجرا، و داده‌های آماری مورد نظر را جمع‌آوری کرد. هر اجرای شبیه‌سازی شده‌ی شبکه‌های فعالیت تصادفی، شامل عملیات زیر است:

۱. تعیین فعالیت‌هایی که در علامت‌گذاری جاری توانا هستند.
۲. فعال‌سازی مجدد فعالیت‌هایی که در علامت‌گذاری جاری، گزاره‌ی فعال‌سازی مجدد آنها ارزش درست دارد.
۳. تولید زمان اجرای فعالیت برای فعالیت‌های زمانی توانا یا فعال شده.
۴. تأثیر دادن نرخ توناسازی در باقی‌مانده‌ی زمان اجرای فعالیت‌هایی که این نرخ برای آنها تعریف شده است.
۵. اگر تعدادی فعالیت آنی توانا است، بر اساس یک روش احتمالی[۳] از میان آنها یکی انتخاب و شلیک می‌شود.
۶. در غیر این‌صورت، با توجه به باقی‌مانده‌ی زمان اجرای فعالیت‌های زمانی، رخداد قریب‌الوقوع انتخاب و فعالیت مربوطه شلیک می‌شود و ساعت شبیه‌سازی با توجه به آن بهنگام می‌شود.
۷. آمارهای مقطعی محاسبه می‌شوند.
۸. اگر زمان شبیه‌سازی به زمان مورد نظر نرسیده به مرحله‌ی ۱ برگشت می‌شود.

مقایسه‌ی تحلیل عددی و شبیه‌سازی

چنان‌که گفتیم، برای حل مدل‌های شبکه‌های فعالیت تصادفی با استفاده ازنرم‌افزار *SharifSAN*، تحلیل عددی و شبیه‌سازی قابل استفاده است. انتخاب یکی از این دو روش، باید با توجه به مزایا و معایب کلی که در اینجا به مهم‌ترین موارد اشاره می‌شود، انجام شود.

مزایای تحلیل عددی

- تحلیل عددی به‌مراتب سریع‌تر از شبیه‌سازی است.
- تحلیل عددی با محاسبات دقیق همراه است و برخلاف شبیه‌سازی، نتایج به‌دست آمده مستقل از دنباله‌ی اعداد شبه تصادفی تولید شده است.
- برای متغیرهای کارایی که از نوع لحظه‌ی زمان هستند، علاوه بر میانگین و واریانس و بدون صرف هزینه‌ی اضافی، توزیع‌ها نیز به‌دست می‌آیند.
- دقت تحلیل عددی برای اغلب تحلیل‌گرهای عددی می‌تواند بدون افزایش قابل توجه زمان محاسبه افزایش یابد.

معایب تحلیل عددی

- تحلیل‌گرهای عددی برای همه‌ی مدل‌های شبکه‌های فعالیت تصادفی در دسترس نبوده و تنها برای رده‌های خاصی از مدل‌ها (از جمله مدل‌های مارکوفی که در آنها همه‌ی فعالیت‌ها زمان اجرای نمایی دارند) وجود دارند.
- فضای حالت مدل باید متناهی باشد. نگه‌داری و پردازش فضاهای حالت بسیار بزرگ مشکلات زیادی را ایجاد می‌کند.
- ایجاد مدل‌هایی که هم دارای نتایج کارایی مورد نظر و هم دارای فضای حالت مجاز برای تحلیل‌گر عددی باشند، کار مشکلی است و باید وقت زیادی برای یافتن مدلی که فضای حالت قابل قبولی دارد صرف کرد.
- تحلیل عددی در صورت مواجهه با مدل‌های پیچیده زمان‌گیر خواهد بود.

مزایای شبیه‌سازی

- شبیه‌سازی برای همه‌ی مدل‌های شبکه‌های فعالیت تصادفی قابل به کارگیری است و فقط به برخی از انواع فرایندهای تصادفی محدود نیست.
- شبیه‌سازی نیاز به تولید فضای حالت و در نتیجه متناهی‌بودن فضای حالت ندارد. بنابراین برای حل مدل‌های با جزئیات بیشتر قابل استفاده است.

معایب شبیه‌سازی

- شبیه‌سازی، درصدی از معیار کارایی را فراهم می‌کند. نتایج شبیه‌سازی در بازه‌ی اطمینان مشخص شده به‌وسیله‌ی کاربر معتبرند، همیشه امکان وجود نتیجه‌ی درست در داخل بازه‌ی اطمینان داده شده نیست.
- افزایش دقت شبیه‌سازی و باریک‌تر کردن بازه‌ی اطمینان، به‌طور فزاینده‌یی زمان شبیه‌سازی را بالا خواهد برد.
- توزیع‌های کامل متغیرهای کارایی (مثلاً توزیع طول صف در یک سیستم) را نمی‌توان با شبیه‌سازی به دست آورد.
- رخداد نادر[28] ممکن است اتفاق بیفتد. اگر شبیه‌سازی برای تخمین یک احتمال کوچک، نظیر احتمال خرابی در یک سیستم با اطمینان بالا[29] استفاده شود زمان شبیه‌سازی برای مشاهده‌ی این نوع رخداد بسیار طولانی خواهد بود.
- حتی اگر رخداد نادر اتفاق نیفتد، ممکن است زمان‌های شبیه‌سازی مدل‌های پیچیده‌بسیار طولانی باشد. از این رو، نمی‌توان شبیه‌سازی را اِکسیری برای حل انواع مسائل مدل‌سازی تلقی کرد.

مشخصات کلی نرم‌افزار

ویژگی‌های اصلی نرم‌افزار *SharifSAN* را می‌توان چنین برشمرد:

- مبتنی بودن بر تعریف جدید شبکه‌های فعالیت تصادفی [3]؛
- دارابودن واسط کاربر گرافیکی برای ویرایش مدل‌ها؛
- قابلیت پویاسازی[30] مدل‌ها (نمایش مرحله به مرحله‌ی اجرای مدل، شلیک کردن فعالیت‌ها و جریان علامت‌ها)؛
- استفاده از روش تفسیر برای بررسی نحو و معنی مدل‌های SAN و انجام خودکار این بررسی‌ها در زمان ویرایش مدل‌ها؛
- بیان خصوصیت‌های درستی سیستم‌ها با استفاده از فرمول‌های منطق زمانی؛
- تولید فضای حالت و نمودار دسترسی؛
- درستی‌یابی فرمول‌های منطق زمانی در فضای حالت با الگوریتم‌های بررسی مدل[31]؛
- تحلیل حالت پایدار با حل عددی معادلات توازن زنجیره‌های مارکوف ادغام شده و محاسبه‌ی احتمالات حالت پایدار؛
- شبیه‌سازی گسسته ـ رخداد مدل‌های SAN در حالت پایدار؛
- محاسبه‌ی معیارهای کارایی و سایر آمارهای مورد نیاز درباره‌ی رفتار مدل با توجه به درخواست‌های کاربر.

نکته‌ی دیگری که باید به آن اشاره کرد، ویژگی مهمی است که در طرح این ابزار وجود دارد و آن امکانات یکپارچه برای درستی‌یابی با استفاده از منطق زمانی و ارزیابی کارایی، اطمینان‌پذیری و قابلیت کارایی است. این بدان معناست که *SharifSAN*، علاوه بر اعتبارسنجی جنبه‌های عملیاتی، برای اعتبارسنجی جنبه‌های کارکردی سیستم‌ها نیز قابل استفاده است و می‌توان مدل‌ها را ابتدا از نظر جنبه‌های کارکردی و درستی (نظیر عدم وجود بن‌بست) مورد بررسی قرار داد و آنگاه به جنبه‌های عملیاتی آنها (نظیر کارایی) پرداخت. این قابلیت، به‌ویژه در کاربردهایی نظیر مدل‌سازی و تحلیل سیستم‌های ارتباطی و شبکه‌های رایانه‌یی که تحلیل پروتکل‌ها در آنها بسیار اهمیت داشته و جزء مهمی از آنهاست، بسیار سودمند خواهد بود. تنها معدودی از ابزارهای مبتنی بر شبکه‌های پتری دارای این قابلیت هستند[15] ولی هیچ‌یک از ابزارهای مبتنی بر شبکه‌های فعالیت تصادفی، این ویژگی را ندارند.

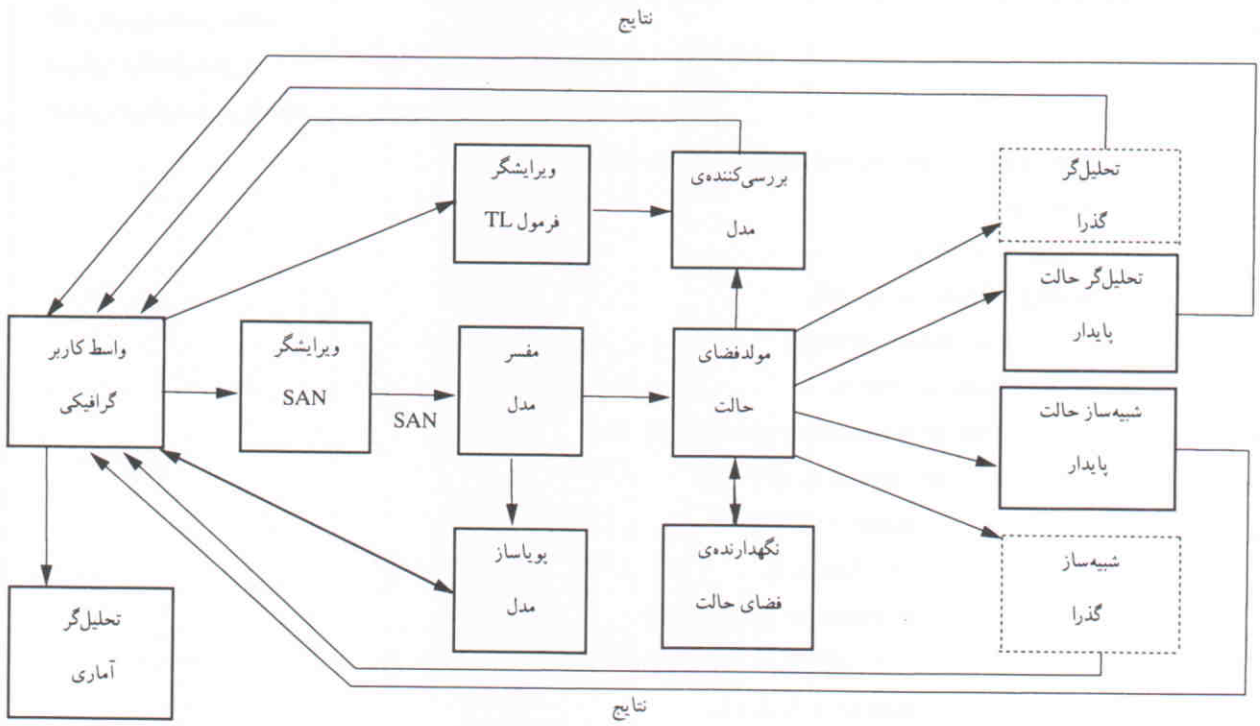
معماری نرم‌افزار *SharifSAN*

اجزاء این معماری و ارتباطات آنها که در شکل ۱ نمایش داده شده است عبارت‌اند از:

۱. واسط کاربر گرافیکی: پیمانه‌ی واسط کاربری نرم‌افزار است که امکانات نرم‌افزاری را از این طریق در اختیار کاربر قرار می‌دهد.

۲. ویرایشگر SAN: یکی از اجزاء واسط کاربری است که امکان ویرایش مدل‌های SAN و ذخیره و بازیابی آنها را فراهم می‌کند. ترسیم مدل به‌صورت هوشمند و با بررسی قواعد SAN به‌صورت تعاملی انجام می‌شود. این ویرایشگر امکان ایجاد مدل‌های بزرگ و حرکت دادن، حذف یا اضافه کردن اجزاء آنها را فراهم می‌سازد.

۳. مفسّر مدل: یکی از تفاوت‌های نرم‌افزار *SharifSAN* با سایر ابزارها، استفاده از روش تفسیر برای ترجمه‌ی توابع و گزاره‌های فعال‌سازی دروازه‌ها و فعالیت‌هاست. از آنجا که این توابع ممکن است عمومی و همانند توابع زبان‌های برنامه‌سازی باشند، باید برای ایجاد نمونه‌ی اجرایی مدل، ترجمه شوند. روش مورد استفاده در METASAN و UltraSAN، تبدیل تعریف مدل به فایل‌های زبان C، و استفاده از ابزارهای Lex&Yacc و مترجم زبان C برای ایجاد نمونه‌ی اجرایی مدل است. اما در *SharifSAN*، از روش تفسیر استفاده می‌شود و به‌همین منظور یک مفسر در داخل نرم‌افزار پیاده‌سازی شده است. دیگر امکان پیش‌بینی شده عبارت است از استفاده از کتابخانه‌ی پیوند پویا[32] (DLL) که از امکانات محیط ویندوز است. برای این منظور ممکن است مدل‌ساز تعدادی تابع از قبل پیش‌بینی شده را متناسب با نیاز مدل‌سازی خود پیاده‌سازی کند و پس از ترجمه وتبدیل به DLL تحویل *SharifSAN* دهد. این روش باعث ایجاد یک نرم‌افزار یکپارچه شده است که امکاناتی نظیر پویاسازی مدل‌ها، ترسیم هوشمند و بررسی تعاملی قواعد SAN براساس آن میسر شده‌است.



شکل ۱. معماری نرم‌افزار *SharifSAN*.

توضیح: مؤلفه‌هایی از نرم‌افزار که با خط چین مشخص شده‌اند، مربوط به ارزیابی قابلیت اتکاء و قابلیت کارایی هستند که در این نوشتار به آنها پرداخته نمی‌شود.

۴. پویاساز مدل: این پیمانه امکان پویاسازی مدل، یعنی اجرای مرحله به مرحله‌ی آن را میسر می‌کند. در حقیقت این امکان، نوعی شبیه‌سازی بصری [۳۳] است که نتایج شبیه‌سازی مدل بر روی واسط کاربری نمایش داده می‌شود، و برای مشاهده‌ی اتفاقات در مدل قابل استفاده است.

۵. مولد فضای حالت: این پیمانه وظیفه‌ی تولید نمودار فضای حالت شبکه را مطابق الگوریتم شکل ۲ دارد.

۶. محرک فضای حالت: امکانات مدیریت و پردازش نمودار فضاهای حالت را فراهم می‌کند. در حال حاضر از روشی که توانایی نگه‌داری نمودارهایی با حدود یک میلیون گره را دارد، استفاده شده است.

۷. کنترل کننده‌ی مدل: وظیفه‌ی اصلی درستی‌یابی را بر عهده داشته و با توجه به الگوریتم‌های درستی‌یابی، فرمول‌های منطق زمانی تبدیل شده به درخت عبارت را با ساخت نمودار دسترسی برچسب‌دار [۳۴] بررسی کرده و درستی یا نادرستی آنها را به کاربر اطلاع می‌دهد.[۴]

۸. تحلیل‌گر حالت پایدار: امکان تحلیل مدل‌های دارای فعالیت‌های نمایی و ارگودیک را میسر می‌سازد. برای این منظور از روش‌های حل عددی تکراری گاوس - سیدل و روش حذف گاوسی استفاده می‌شود. پس از حل، احتمالات حالت پایدار و معیارهای کارایی محاسبه می‌شوند.

۹. شبیه‌ساز حالت پایدار: امکان شبیه‌سازی گسسته ـ رخداد را برای مدل‌های دارای فعالیت‌های عمومی و محاسبه‌ی معیارهای کارایی و داده‌های آماری تعریف شده به‌وسیله‌ی کاربر را فراهم می‌کند.

۱۰. تحلیل‌گر گذرا: امکان تحلیل مدل‌های غیرارگودیک و محاسبه‌ی معیارهای اطمینان‌پذیری و قابلیت کارایی را میسر خواهد کرد.

۱۱. شبیه‌ساز گذرا: امکان شبیه‌سازی گسسته ـ رخداد مدل‌های غیرارگودیک، و محاسبه‌ی معیارهای اطمینان‌پذیری و قابلیت کارایی را میسر خواهد کرد.

۱۲. تحلیل‌گر آماری: امکانات تحلیل‌های آماری نتایج به‌دست آمده، و تبدیل آنها به نمودارهای آماری و ذخیره‌سازی در قالب فایل‌های HTML را فراهم می‌کند.

ویژگی‌های طراحی و پیاده‌سازی

نرم‌افزار *SharifSAN* در محیط ویندوز پیاده‌سازی شده است و امکانات کاملی برای ویرایش مدل‌های SAN، پویاسازی مدل‌ها،

الگوریتم تولید فضای حالت:

ورودی: شبکه‌های فعال

خروجی: نمودار دسترسی (RG)

```
R ← {S0}  // (R, A) defines RG and S0 is the initial state
A ← ϕ
TState ← { S0 }
WHILE TState ≠ ϕ DO
        S ← Select (TState)
        Stable ← TRUE
        FOR all instantanous activities (i) Do
              IF enabled (i, s) THEN
                Stable ← FALSE
                j ← Fire (i, s)
                IF NewState (j, R) THEN
                      TState ← TState ∪ { j }
                NewArc ( A, S, j )
              END IF
      IF Stable THEN
         FOR all timed activities (i) DO
              IF enabled (i, s) THEN
               j ← Fire (i, s)
               IF NewState (j, R) THEN
               TState ← TState ∪ { j }
               NewArc ( A, s, j )
              END IF
END WHILE
```

شکل ۲. الگوریتم تولید فضای حالت.

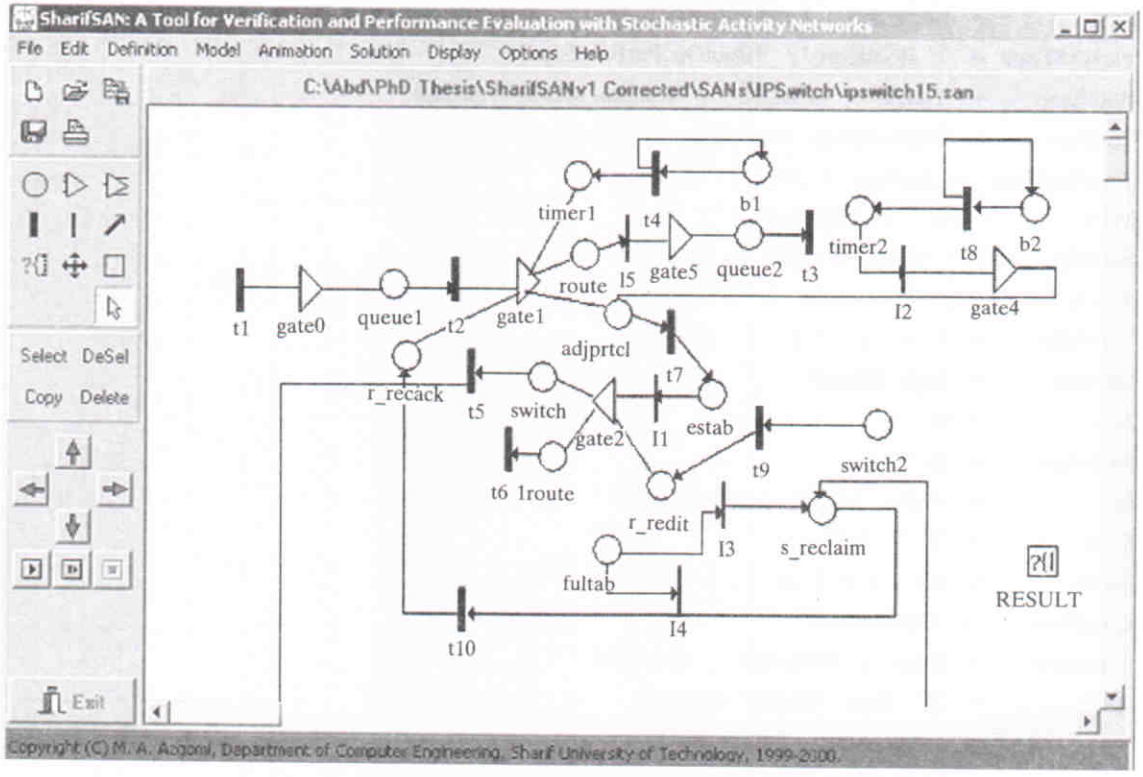
درستی‌یابی مدل‌ها با منطق زمانی و ارزیابی کارایی (و در آینده ارزیابی اطمینان‌پذیری و قابلیت کارایی)، با روش‌های تحلیل عددی و شبیه‌سازی فراهم می‌کند. در ادامه به مهم‌ترین ویژگی‌های طراحی و پیاده‌سازی این ابزار می‌پردازیم.

واسط کاربری

واسط کاربری این نرم‌افزار به‌نحوی طراحی شده است که از طریق ویرایشگر مدل امکانات فراوانی برای طراحی مدل‌های شبکه‌های فعالیت به‌صورت بصری در اختیار مدل‌ساز قرار می‌دهد. مدل‌ساز می‌تواند به‌سهولت اجزاء تشکیل‌دهنده‌ی مدل را انتخاب کرده، در کنار هم قرار داده و به‌هم متصل کند. اجزاء مدل شامل مکان‌ها دروازه‌های ورودی[۳۵]، دروازه‌های خروجی[۳۶]، فعالیت‌های زمانی و فعالیت‌های آنی، به‌صورت اشیاء مجزا در صفحه‌ی طراحی مدل قابل کنترل هستند. به‌طوری که پس از ترسیم یک مدل، امکان تغییر خصوصیات آنها، حرکت دادن و حذف و اضافه‌ی آنها و نیز ذخیره‌سازی و بازیابی مدل‌ها وجود دارد. در شکل ۳، نمایی از صفحه‌ی اصلی نرم‌افزار نمایش داده شده است.

برخی از مهم‌ترین ویژگی‌های واسط کاربری نرم‌افزار عبارت‌اند از:

- **تعریف شبکه‌ی فعالیت به‌صورت بصری:** قابلیت‌های کاملی در این زمینه فراهم شده است. مفاهیم اولیه‌ی شبکه‌های فعالیت



شکل ۳. نمایی از واسط کاربری *SharifSAN*.

تصادفی، به‌صورت شمایل‌هایی در صفحه‌ی ویرایشگر SAN وجود دارند که به‌راحتی قابل انتخاب هستند. متناسب با هر جزء خصوصیت‌هایی از کاربر درخواست می‌شود و مقداردهی اولیه‌ی مدل با علامت‌گذاری مکان‌ها انجام می‌شود.

- **نمایش نمودار دسترسی:** پس از تولید فضای حالت، ساختار نمودار دسترسی برای مدل‌های کوچک، قابل نمایش بر روی صفحه است.
- **امکان حرکت دادن اجزاء شبکه یا کل شبکه:** کاربر امکان حرکت دادن هر کدام از اجزاء یا کل مدل را بر روی صفحه دارد. این قابلیت به اصلاح مدل و گسترش آن کمک می‌کند.
- **رسم هوشمند خطوط اتصال گره‌های شبکه:** اتصال اجزاء مدل شبکه‌های فعالیت تصادفی دارای قواعد خاصی است که در هنگام رسم خطوط اتصال گره‌های شبکه، به‌صورت خودکار کنترل می‌شوند.
- **ذخیره و بازیابی مدل‌های تعریف‌شده:** امکان ذخیره‌سازی مدل‌های تعریف شده در قالب فایل متنی و بازیابی و اصلاح آنها وجود دارد.
- **ذخیره‌سازی تصویر شبکه در قالب فایل گرافیکی:** یکی دیگر از قابلیت‌های فراهم شده، تولید خروجی گرافیکی مجزا با قالب‌های استاندارد از مدل ساخته شده است.

مفسر شبکه‌های فعالیت تصادفی

در نرم‌افزار *SharifSAN*، بر خلاف سایر نرم‌افزارهای ایجاد شده براساس SAN، از روش تفسیر برای ترجمه‌ی مدل استفاده می‌شود. در سایر نرم‌افزارها مدل‌های ایجاد شده با ویرایشگر SAN، به تعاریفی به زبان C تبدیل شده و با بهره‌گیری از Lex & YACC اتصال فایل‌های دیگر، تبدیل به مدل اجرایی می‌شود. هدف اصلی از استفاده از روش تفسیر ایجاد یک نرم‌افزار یکپارچه است. بنابراین مفسری به‌صورت داخلی ایجاد شده است که مطابق روش، نزولی - بازگشت‌کننده[۳۷] عمل می‌کند و تغییرات اعمال شده در SAN، گزاره‌ها، توابع و متغیرهای تعریف شده به‌وسیله‌ی کاربر را تفسیر می‌کند. این مفسر عهده‌دار وظیفه‌ی بررسی قواعد SAN نیز هست و در صورت وجود خطاهای نحوی و معنایی، به کاربر پیغام‌های خطای مناسب می‌دهد. برای تفسیر گزاره‌ها و توابع قابل تعریف در مدل، از دستور زبان شکل ۴ استفاده می‌شود که با نمادگذاری BNF ارائه شده است.

```
Expr          → BoolExpr | ParBoolExpr | 'NOT' ParBoolExpr
ParBoolExpr   → '(' BoolExpr ')' {BoolOp ParBoolExpr}*
BoolExpr      → 'TRUE' | 'FALSE' | TokExpr RelOp TokExpr
TokExpr       → SimpTokExpr {IArithOp SimpTokExpr}*
SimpTokExpr   → IntNum | IVar | Mark
Mark          → 'X' '[' NodeLab ']'
BoolOp        → [ 'AND' | 'OR' | 'XOR' ]
RelOp         → [ '<' | '<=' | '>' | '>=' | '=' | '<>' ]
IArithOp      → [ '+' | '-' ]
IntNum        → Digit {Digit}*
IVar          → Id
NodeLab       → Id
Id            → Alpha {Alpha | Digit}*
Digit         → '0' | .. | '9'
Alpha         → 'a'| .. | 'z' | 'A' | .. | 'Z'
GateFunc      → {Statement}*
Statement     → IfStat | WhileStat | SimpStat
IfStat        → 'IF' Expr 'THEN' <EOL>
                  SimpStats
                { ' ELSEIF' Expr 'THEN' <EOL>
                   SimpStats}*
                { 'ELSE' <EOL>
                   SimpStats}
                'ENDIF' <EOL>
WhileStat     → 'WHILE' Expr 'Do'
                    SimpStats
                'ENDWHILE'
SimpStats     → {SimpStat <EOL>
                  SimpStat <EOL>
                  . . .
                  SimpStat <EOL>
SimpStat      → AssignStat | UDFuncCall
AssignStat    → 'X' '[' NodeLab ']' ' := ' TokExpr
                | Id ' := ' TokExpr
                | Id ' := ' DExpr
UDFuncCall    → ' CALL ' Id '(' {DExpr}* ')'
DExpr         → SimpDExpr {DArithOp SimpDExpr}*
SimpDexpr     → DNum | DVar | DVar | Mark | '(' DExpr ')'
IArithOp      → [ '+' | '-' | '*' | '/' ]
DVar          → Id
DNum          → Digit {Digit}* '.' Digit {Digit}*
<EOL>         → End_Of_Line
```

شکل ۴. دستور زبان گزاره‌ها و توابع SAN با نمادگذاری BNF.

مولد فضای حالت

پس از آنکه سیستم را با استفاده از واسط کاربر گرافیکی *SharifSAN* مدل‌سازی کرده‌ایم، فضای حالت سیستم تولید می‌شود. این فضای حالت یا نمودار دسترسی، برای انجام درستی‌یابی به کمک منطق زمانی و ارزیابی کارایی با روش حل عددی مورد استفاده قرار می‌گیرد. روشی که هم اکنون برای نگه‌داری نمودار دسترسی مورد استفاده قرار می‌گیرد، دارای محدودیت‌هایی است، از جمله عدم توانایی نگه‌داری فضای حالت بسیار بزرگ. از این‌رو، در آینده باید روش‌های بهتری برای مدیریت نمودار دسترسی طراحی و پیاده‌سازی شود، که در معماری نرم‌افزار نیز در قالب پیمانه‌ی «مولد فضای حالت» در نظر گرفته شده است. در شکل ۲، الگوریتم تولید فضای حالت ارائه شده است.

سیستم درستی‌یابی

پس از تولید فضای حالت، فرمول‌های منطق زمانی به‌منظور درستی‌یابی، از کاربر دریافت می‌شود. مشخصات سیستم درستی‌یابی، شامل نحوه‌ی تبدیل شبکه‌های فعالیت به سیستم گذر معادل، دستور زبان فرمول‌های منطق زمانی، مفسر فرمول‌ها، الگوریتم‌های وارسی مدل و نحوه‌ی درستی‌یابی فرمول‌های منطق زمانی در فضای حالت قبلاً منتشر شده‌اند.[۴]

حل‌کننده‌ی عددی حالت پایدار

در صورتی که فضای حالت مدل متناهی باشد و همه‌ی فعالیت‌های زمانی دارای توزیع نمایی باشند، می‌توان سیستم را به‌صورت تحلیلی حل کرد. برای این منظور تحلیل‌گر عددی حالت پایدار، این اعمال را انجام می‌دهد:

۱. **به‌دست آوردن نمودار دسترسی.** فضای حالت سیستم را می‌توان با یک نمودار نمایش داد. بنابراین در الگوریتم شکل ۴ که به عنوان مولد فضای حالت سیستم استفاده می‌شود، تمهیدات لازم برای ساخت این نمودار اندیشیده شده است. در این نمودار حالت‌های پایدار و نیز حالت ناپایدار (که یک فعالیت آنی در آن تواناست)، وجود دارند. اگر تنها حالت‌های پایدار این نمودار را در نظر بگیریم، نمودار دسترسی با یک زنجیره‌ی مارکوف هم‌شکل خواهد بود. لذا با به‌دست آوردن زنجیره‌ی مارکوف ادغام شده، می‌توانیم با استفاده از روش‌های حل عددی زنجیره‌های مارکوف، SAN را تحلیل کنیم.

۲. **بررسی ارگودیک بودن نمودار.** شرط وجود احتمالات پایدار، ارگودیک بودن زنجیره‌ی مارکوف است. یعنی در نمودار دسترسی باید همه‌ی گره‌ها به گره ریشه که همان حالت اولیه‌ی سیستم است، بازگشت کنند. در اغلب مدل‌های کارایی، این وضعیت برقرار است، اما در اطمینان‌پذیری مدل‌های حاصله ارگودیک نیستند. برای بررسی این خاصیت، از یک الگوریتم جستجوی عمق اول[۳۸] برای بررسی اتصال کامل نمودار استفاده می‌کنیم.

۳. **به‌دست آوردن ماتریس نرخ ـ گذر.** با پردازش نمودار دسترسی، می‌توان یک ماتریس مربعی (Q) که ابعاد آن برابر است با تعداد حالت‌های پایدار فضای حالت، به‌دست آورد. هر عنصر q_{ij} ماتریس نرخ ـ گذر، نشان دهنده‌ی نرخ گذر از حالت پایدار i به حالت پایدار j است. اگر حالت پایدار i از طریق تعدادی حالت ناپایدار به حالت j منتهی شود، نرخ q_{ij} با احتساب احتمالات مربوط به حالت‌های ناپایدار محاسبه می‌شود.

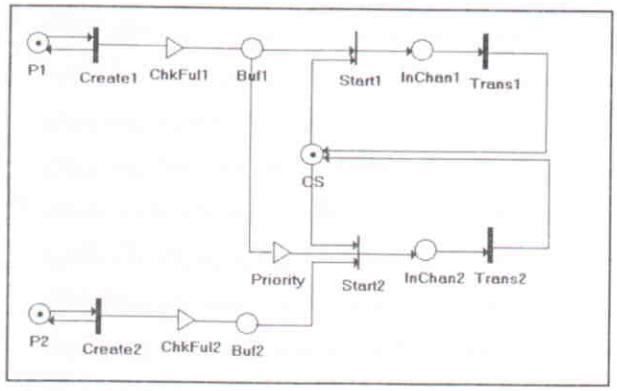
۴. **محاسبه‌ی احتمالات پایدار.** برای این منظور باید معادلات توازن زنجیره‌ی مارکوف حل شود. با استفاده از ماتریس نرخ ـ گذر (Q)، دستگاه معادلات توازن به‌دست آمده و از یکی از روش‌های تکراری نظیر گاوس ـ سیدل، و حذف گاوسی برای حل آن استفاده می‌شود.

۵. **محاسبه‌ی معیارها.** برای این منظور، تحلیل‌گر حالت پایدار، احتمالات پایدار مربوط به هر کدام از حالت‌ها و متوسط تعداد علائم در هر کدام از مکان‌ها را از طریق نمودار فضای حالت محاسبه می‌کند.

شبیه‌ساز حالت پایدار

اگر نمودار دسترسی شبکه، نامتناهی یا غیر ارگودیک باشد یا تابع زمان اجرای یکی از فعالیت‌ها غیرنمایی باشد، مجبور به استفاده از شبیه‌سازی برای تحلیل مدل هستیم. روش مورد استفاده برای شبیه‌سازی مدل‌های SAN در نرم‌افزار *SharifSAN*، شبیه‌سازی گسسته ـ رخداد است که الگوریتم آن در بالا تشریح شد. شبیه‌سازی به‌صورت بصری نیز قابل انجام است، که در واقع پویاسازی مدل است. به این ترتیب که شبکه به‌صورت تصادفی و براساس توزیع زمان‌های اجرای فعالیت‌های زمانی و احتمالات فعالیت‌های آنی اجرا می‌شود و نتایج اجرا که تغییر علائم مکان‌های شبکه است، به‌صورت بصری نمایش داده می‌شود.

اما منظور از شبیه‌سازی مدل، به‌دست آوردن معیارهای کارایی و آمارهایی درباره‌ی رفتار مدل است. روش‌های تک‌مسیر و تکرار مستقل و با تعیین زمان مورد نظر برای قبل از حالت پایداری و حالت پایداری قابل استفاده‌اند. در حالت پایداری، داده‌های مورد نیاز برای محاسبه‌ی معیارهای کارایی و سایر داده‌های آماری که از طریق گره‌های پرس و جو[۳۹] به‌وسیله‌ی کاربر قابل تعریف‌اند، جمع‌آوری می‌شوند. در حین پیشرفت شبیه‌سازی و نیز پس از پایان آن، این



Enabling Predicates:
Priority: X[Buf1]=0
Functions:
ChkFul1:if X[Buf1]<10 thenX[Buf1]:=X[Buf1]+1
ChkFul2:if X[Buf2]<10 thenX[Buf2]:=X[Buf2]+1

شکل ۵. سیستمی با دو تولیدکننده و دو مصرف‌کننده.

معیارها و آمارها محاسبه شده و به کاربر نمایش داده می‌شوند. برای تولید اعداد تصادفی از یک روش همنهشتی خطی ضربی[۴۰]، با دوره‌ی تناوب طولانی (10^{31}) و برای تولید توزیع‌های نمایی، یکنواخت و نرمال از روش‌های مناسب استفاده شده است.

کاربردهای نرم‌افزار

هدف از طراحی نرم‌افزار، کاربرد آن در زمینه‌های مختلف مدل‌سازی و تحلیل، شامل جنبه‌های کارکردی و نیز عملیاتی بوده است. کاربردهای این نرم‌افزار در زمینه‌ی تحلیل جنبه‌های کارکردی و انجام درستی‌یابی سیستم‌های همروند با استفاده از منطق زمانی تشریح شده و مثال‌هایی در این زمینه ارائه شده است.[۴] اما تحلیل جنبه‌های عملیاتی برای مقاصد ارزیابی کارایی سیستم‌های رایانه‌یی و ارتباطی از اهداف اصلی طراحی این نرم‌افزار، بوده است. همچنین، آنگونه که در طرح نرم‌افزار پیش‌بینی شده است، پس از تکمیل قابلیت‌های تحلیل حالت گذرا[۴۱]، در زمینه‌ی ارزیابی اطمینان‌پذیری و قابلیت کارایی نیز دارای کاربرد خواهد بود.

در ادامه، مثال‌هایی از کاربرد *SharifSAN* در زمینه‌های مختلف ارائه می‌شود. مثال نخست، توانایی وارسی سیستم‌های همروند را با استفاده از فرمول‌های منطق زمانی نشان می‌دهد. مثال دوم، در زمینه‌ی ارزیابی کارایی با روش‌های حل عدد و شبیه‌سازی است. مثال آخر، کاربرد نرم‌افزار را در زمینه‌ی مدل‌سازی و تحلیل لوازم راه‌گزینی[۴۲] شبکه‌های سریع نشان می‌دهد.

مثالی از وارسی سیستم‌های همروند

سیستمی با دو تولید کننده و دو مصرف کننده، دو بافر محدود و یک کانال اشتراکی برای انتقال پیغام از تولید کننده به مصرف کننده را در نظر بگیرید. این کانال یک منبع است که باید به‌صورت دوبه‌دو ناسازگار[۴۳] در اختیار تولیدکنندگان قرار بگیرد. تولیدکننده‌ی nام پیغام‌ها را برای مصرف‌کننده‌ی nام می‌سازد. پیغام‌ها به‌منظور ارسال در بافر nام، که دارای ۱۰ مکان است، قرار می‌گیرند. فرض کنید که اولویت انتقال پیغام‌ها و در دست گرفتن کانال با تولیدکننده‌ی اول است و تا وقتی که در بافر اول پیغامی برای ارسال باشد اولویت ارسال با آن خواهد بود و اگر تولیدکننده‌ی دوم در حال ارسال باشد، تولیدکننده‌ی اول نمی‌تواند کار او را قطع کند (انحصاری بودن).

این سیستم با *SharifSAN* مدل و آنگاه وارسی شده است. شکل ۵ نمایی از این مدل است. حالت اولیه‌ی مدل در شکل مشخص شده است و گزاره‌های تواناسازی و توابع دروازه‌های خروجی ChkFul 1 و ChkFul 2 هم در انتهای شکل آمده‌اند. بر اساس گزاره‌ی تواناسازی، دروازه‌ی ورودی در صورتی باز است که در بافر ۱ علامتی موجود نباشد. اما توابع دروازه‌های خروجی، کنترل می‌کنند که بیش از ظرفیت بافرها (۱۰ مکان)، علامت در آنها قرار نگیرد.

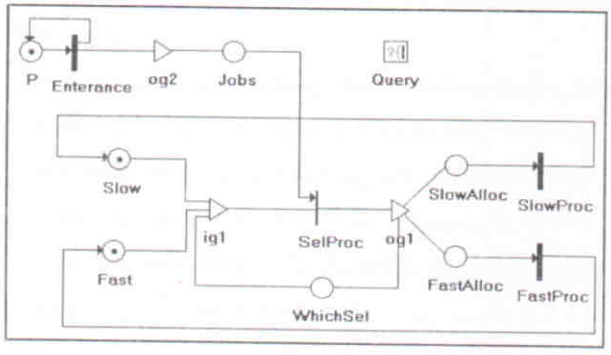
وارسی مدل: دو سوال در مورد مدل فوق مطرح و بررسی می‌شوند:

■ **سؤال ۱:** آیا دوبه‌دو ناسازگاری در استفاده از کانال رعایت می‌شود؟

این سؤال را با استفاده از فرمول‌های منطق زمانی انشعابی، چنین می‌توان بیان کرد:

NOT (EF (AND (X [InChan1] = 1, X [InChan2] = 1)))

نتیجه‌ی وارسی فرمول فوق مثبت بوده که معنی آن این است که در مدل فوق دوبه‌دو ناسازگاری رعایت می‌شود.



ig1:
Predicate:
(X[Fast]<>0)OR(X[Slow]<>0)
Function
IF X[Fast]<>0 THEN
X[Fast]:=0
X[WhichSel]:=0
ELSE
X[Slow]:=0
X[WhichSel]:=1
ENDIF

og1:
Function
IF X[WhichSel]=0 THEN
X[FastAlloc]:=1
ELSE
X[SlowAlloc]:=1
ENDIF

og2:
Function
Arrivals:=Arrivals+1
IF X[Jobs]<3 THEN
X[Jobs]:=X[Jobs]+1
ELSE
Rejected:=Rejected+1
ENDIF

شکل ۶. مثالی از یک سیستم رایانه‌یی با دو سرویس‌دهنده.

جدول ۱. نتایج تحلیل عددی و شبیه‌سازی مدل شکل ۶

معیار	نتایج شبیه‌سازی	نتایج حل عددی
میانگین تعداد علامت‌ها در مکان Jobs	۰/۱۸۸۹۱	۰/۱۸۸۷۶
میانگین تعداد علامت‌ها در مکان SlowAlloc	۰/۴۱۰۱۸	۰/۴۰۹۶۱
میانگین تعداد علامت‌ها در مکان FastAlloc	۰/۴۸۵۱۲	۰/۴۸۴۰۷
میانگین تعداد علامت‌ها در مکان whichSel	۰/۳۱۱۶۱	۰/۳۱۱۶۹
میانگین تعداد علامت‌ها در مکان Fast	۰/۵۱۶۳۳	۰/۵۱۵۹۳
میانگین تعداد علامت‌ها در مکان Slow	۰/۵۸۹۸۱	۰/۵۹۰۳۹
احتمال وجود ۰ نشانه در مکان Jobs	۰/۸۷۶۷۹	۰/۸۷۷۰۷
احتمال وجود ۱ نشانه در مکان Jobs	۰/۰۷۳۳۳	۰/۰۷۲۹۷
احتمال وجود ۲ نشانه در مکان Jobs	۰/۰۳۳۹۶	۰/۰۳۴۰۵
احتمال وجود ۳ نشانه در مکان Jobs	۰/۰۱۵۹۰	۰/۰۱۵۸۹
احتمال وجود ۰ نشانه در مکان SlowAlloc	۰/۵۸۹۸۱	۰/۵۹۰۳۹
احتمال وجود ۱ نشانه در مکان SlowAlloc	۰/۴۱۰۱۸	۰/۴۰۹۶۰
احتمال وجود ۰ نشانه در مکان FastAlloc	۰/۵۱۴۸۷	۰/۵۱۵۹۲
احتمال وجود ۱ نشانه در مکان FastAlloc	۰/۴۸۵۱۲	۰/۴۸۴۰۷
احتمال وجود ۰ نشانه در مکان WhichSel	۰/۶۸۸۳۸	۰/۶۸۸۳۰
احتمال وجود ۱ نشانه در مکان WhichSel	۰/۳۱۱۶۱	۰/۳۱۱۶۹
احتمال وجود ۰ نشانه در مکان Fast	۰/۴۸۵۱۲	۰/۴۸۴۰۷
احتمال وجود ۱ نشانه در مکان Fast	۰/۵۱۴۸۷	۰/۵۱۵۹۲
احتمال وجود ۰ نشانه در مکان Slow	۰/۴۱۰۱۸	۰/۴۰۹۶۰
احتمال وجود ۱ نشانه در مکان Fast	۰/۵۸۹۸۱	۰/۵۹۰۳۹
تعداد کل کارهای ورودی به سیستم	۷۰۱۲۲۱	—
تعداد کل کارهای رد شده به دلیل پر بودن ظرفیت	۱۰۹۸۸	—
احتمال رد شدن کارهای ورودی	۰/۰۱۵۶۶	

■ **سؤال ۲:** آیا امکان قحطی[۴۴] برای تولیدکننده دوم وجود دارد؟

این سؤال را نیز می‌توان با فرمول‌های منطق زمانی به‌صورت زیر بیان کرد:

NOT (AG (AF (X [InChan2] = 1)))

نتیجه‌ی وارسی فرمول این سؤال هم مثبت است. این مسأله با پویاسازی مدل نیز به‌وضوح مشخص می‌شود و پس از مدتی، به‌دلیل آنکه در بافر ۱ تعدادی علامت وجود دارد، به‌خاطر رعایت اولویت، تولیدکننده‌ی دوم دیگر فرصت ارسال پیغام را پیدا نمی‌کند.

مثالی از ارزیابی کارایی سیستم‌های رایانه‌یی

برای نشان دادن قابلیت‌های ارزیابی کارایی نرم‌افزار، مثالی از یک سیستم رایانه‌یی راکه در شکل ۶ نشان داده شده است، در نظر بگیرید. این شکل نشانگر سیستمی با دو سرویس‌دهنده است، که سرعت یکی دو برابر دیگری است. کارهایی برای دریافت سرویس به این سیستم وارد می‌شوند و پس از آن، یکی از دو سرویس‌دهنده را انتخاب می‌کنند که اولویت با سرویس‌دهنده‌ی سریع‌تر است. اگر هر دو سرویس‌دهنده مشغول باشند، کارها در یک صف با ظرفیت محدود منتظر می‌شوند. کارهایی که سیستم را پر ببینند، بلافاصله سیستم را ترک می‌کنند. زمان سرویس برای کارهای مختلف متفاوت است.

سیستم فوق، به‌دلیل نمایی بودن همه‌ی فعالیت‌های زمانی و ارگودیک بودن، هم با روش تحلیل عددی و هم با روش شبیه‌سازی تحلیل شده است.

همان‌گونه که مشاهده‌می‌کنید نتایج تحلیل عددی و شبیه‌سازی در جدول ۱ با دقت خوبی با هم مطابقت دارند.

نمونه‌هایی از کاربرد در شبکه‌های سریع

قابلیت به کارگیری *SharifSAN* در سیستم‌های ارتباطی، به‌ویژه در شبکه‌های سریع، از ویژگی‌های طراحی و پیاده‌سازی آن است. بر این اساس تاکنون این نرم‌افزار در پروژه‌هایی در زمینه‌ی شبکه‌های سریع به‌طور عملی مورد استفاده قرار گرفته و نتایج سودمندی را نیز به‌همراه داشته است.

با توجه به نیازهای ایجاد شده در فرایند مدل‌سازی و تحلیل پروژه‌هایی که در ادامه توصیف خواهند شد، تغییراتی در نرم‌افزار داده شده و امکاناتی بدان اضافه شده است. ضمن آن که به‌کارگیری عملی نرم‌افزار، به اِشکال‌زدایی و اطمینان از صحت عملکرد آن نیز کمک کرده است. برای فراهم ساختن امکانات به‌کارگیری بیشتر و مؤثرتر این نرم‌افزار، بررسی دقیق‌تر نیازهای مدل‌سازی و تحلیل آن در زمینه‌ی مخابرات و شبکه‌های سریع در حال انجام است.

در ادامه به دو نمونه از کاربردهای نرم‌افزار، اشاره می‌شود:

- از نرم‌افزار *SharifSAN* برای مدل‌سازی ساختار راه‌گزین بر چسب[۴۵] و تحلیل کارایی مؤلفه‌های آن استفاده شده است.[۱۶] برای ارزیابی طرح، مدل شبکه‌های فعالیت تصادفی آن با استفاده از نرم‌افزار *SharifSAN* ایجاد شده است. چون مدل‌های حاصله ــ به‌دلیل تبعیت برخی از عناصر سرویس‌دهنده در سیستم از توزیع‌های قطعی یا یکنواخت ــ مارکوفی نیستند، فقط از روش شبیه‌سازی برای ارزیابی مدل‌ها استفاده شده است. معیارهای کارایی، شامل توزیع صف ورودی، توزیع صف خروجی، احتمال اتلاف سلول براثر محدودیت صف ورودی و تأخیر را محاسبه کرده و از آنها برای تنظیم پارامترهای طرح استفاده کرده‌اند.
- از این نرم‌افزار برای مدل‌سازی و تحلیل طرح ساختار راه‌گزین IP[۴۶] نیز استفاده شده است.[۱۷و۱۸] در این پروژه، ساختار نرم‌افزاری این نوع راه‌گزین با استفاده از *SharifSAN* مدل‌سازی شده‌است. مدل‌های حاصله با روش شبیه‌سازی تحلیل شده و معیارهای کارایی، شامل پارامترهای تأخیر، تعداد جریان سرویس داده شده در واحد زمان، درصد راه‌گزینی بسته‌ها، میزان اتلاف سلول و اندازه‌ی بافرهای ورودی و خروجی محاسبه شده است.

نتیجه‌گیری

در این نوشتار به تشریح ویژگی‌های ارزیابی کارایی یک ابزار مدل‌سازی و تحلیلی، مبتنی بر تعریف جدید مدل شبکه‌های فعالیت تصادفی، با نام *SharifSAN* پرداخته شده است.

این نرم‌افزار در حال حاضر امکانات یکپارچه‌یی را برای پویاسازی، درستی‌یابی با استفاده از منطق زمانی و ارزیابی کارایی، با روش‌های تحلیل عددی پایدار و شبیه‌سازی گسسته ـ رخداد فراهم می‌کند. نرم‌افزار دارای واسط کاربری گرافیکی بوده و از روش تفسیر برای ایجاد مدل‌های اجرایی استفاده می‌کند. اما تکمیل این نرم‌افزار مستلزم کارهای دیگری نیز هست. در وهله‌ی نخست باید بخش‌های تحلیل گذرا و ارزیابی اطمینان‌پذیری و قابلیت کارایی که در طرح آن وجود دارد، پیاده‌سازی شوند. همچنین از آنجا که هدف اصلی از طراحی این ابزار، امکان به‌کارگیری برای مدل‌سازی و تحلیل سیستم‌های پیچیده و بزرگ است، باید روش‌های مورد نیاز و متناسب با این نوع کاربردها در آن گنجانده شوند. در صورت ایجاد قابلیت و افزایش قابلیت‌های تحلیل گذرا و امکانات مدیریت فضاهای حالت بسیار بزرگ، و نیز تقویت و بهینه‌سازی الگوریتم‌های بخش درستی‌یابی نرم‌افزار، امکانات گسترده‌یی برای مدل‌سازی و تحلیل سیستم‌های رایانه‌یی و ارتباطی فراهم خواهد شد.

پانوشت

1. stochastic activity networks
2. Petri nets
3. functional aspects
4. verification
5. reachability graph
6. generalized stochastic Petri nets
7. temporal logic
8. numerical analysis
9. discrete-event simulation
10. well-behavedness
11. high-speed networks
12. steady state
13. markov chains
14. usage level
15. throughput
16. utilization factor
17. missionability
18. productivity
19. ergodic
20. markings
21. rate-transition matrix
22. balance equations
23. state space
24. independent replications
25. single-run method
26. regenerative method
27. confidence interval
28. rare event
29. highly dependable
30. animation

31. model checking
32. dynamic link library
33. visual simulation
34. labelled reachability graph
35. input gates
36. output gates
37. recursive - descent
38. depth-first search
39. query nodes
40. multiplicative linear congruentional method
41. transient analysis
42. switching
43. mutually exclusive
44. starvation
45. tag switch
46. IP switch

منابع

1. Movaghar, A. and Meyer, J.F. "Performability modeling with stochastic activity networks," Proceedings of the 1984 Real-Time System Symposium, Austin, TX, pp. 215-224, (Decem 1984).
2. Movaghar, A. *Performability Modeling with Stochastic Activity Networks,* Ph. D. Dissertation, The University of Michigan, (1985).
3. Movaghar, A. "Stochastic activity networks: A new definition and some properties," *Scientia Iranica,* **8** (4), Fall (2001).

۴. عبداللهی ازگمی، م. و موقر، ع. «ابزاری برای درستی‌یابی سیستم‌های همروند بر اساس شبکه‌های فعالیت»، مجموعه مقالات پنجمین کنفرانس بین‌المللی سالانه انجمن کامپیوتر ایران، دانشگاه شهید بهشتی، صص، ۳۰۵-۳۱۴، تهران، (اسفند ۱۳۷۸).

5. Marsan, M.A., Balbo, G. and Conte, G., *Performance Models of Multiprocessor Systems,* MIT Press (1986).
6. Manna, Z. and Pnuelli, A. *The Temporal Logic of Reactive and Concurrent Systems:* Specifications, Springer-Verlag, Berlin (1992).
7. Banks, J., carson II, J.S. and Nelson, B.L. *Discrete-Event System Simulation,* 2nd edition, Prentic-Hall (1996).
8. Sanders, W.H. and Meyer, J.F. "METASAN: A performability evaluation tool based on stochastic activity networks," *Proceedings ACM/IEEE-CS Fall Joint Computer Conference,* pp. 807-816 (Nov 1986).
9. CRHC "UltraSAN user's manual, version 3 "Center for Reliable and High-Performance Computing, Coordinated Science Laboratory, University of Illinois (1996).
10. Sanders, W.H. and et al, "UltraSAN version 3.0: architecture, features, and implementation", *Proccedings of the AIAA Computing in Aerospace 10th Conference,* San Antonio, TX, pp 327-338 (March 28-30, 1995).
11. Daly, D. and et al, "Mobius: An extensible frameworks for performance and dependability modeling, "*Proceedings of the 8th International Workshop on Petri Nets and Performance Models (PNPM'99),* Zaragoza, Spain (Sep 1999).
12. Movaghar, A. "Stochastic activity networks: A new definition", *Proceedings of the IASTED International Conference on Modelling and Simulation, Pittsburgh, PA,* PP 27-30 (May 1997).
13. Kant, K. *Introduction to Computer System Performance Evaluation,* McGraw-Hill (1992).
14. Rice, J.R. *Numerical Methods, Software and Analysis,* Academic Press (1993).
15. DAIMI "Petri nets tools database quick overview," URL: http:// WWW. dimi. au. dk /PetriNets /tools /form. html, Last update: (Feb 2000).

۱۶. عظیم‌دوست، ب. «طراحی ساختاری مدارهای واسطه جهت برقراری ارتباط IP بر روی شبکه ATM از طریق رویکرد Tag Switching»، پایان‌نامه کارشناسی ارشد، دانشگاه صنعتی شریف، (شهریور ۱۳۷۹).

۱۷. کهندانی، ف «طراحی ساختاری مدارهای واسطه جهت برقراری ارتباط IP بر روی شبکه ATM از طریق رویکرد IP Switching» پایان نامه کارشناسی ارشد، دانشگاه صنعتی شریف، (شهریور ۱۳۷۹).

۱۸. کهندانی، ف. و ولایی، ش. «طراحی ساختاری IP Switch و تحلیل آن با مدل SAN» مجموعه مقالات نهمین کنفرانس ایرانی مهندسی برق (ICEE ۱۰۱)، صص، ۷/۱ - ۷/۱۰، تهران، (خرداد ۱۳۸۰).