

ریو (Reo): یک مدل هماهنگی مبتنی بر کانال برای ترکیب مولفه‌ها

سید محمدمهدی احمدپناه ۹۶۰۲۳۰۰۳

چکیده

در این گزارش، معرفی مختصری درباره زبان مدل‌سازی ریو (Reo) ارائه می‌شود. ریو پارادایمی مبتنی بر کانال برای ترکیب مولفه‌های نرم‌افزاری است. ریو یک زبان هماهنگی برون^۱ است که در آن، هماهنگ‌کننده‌های پیچیده، یا اصطلاحاً رابط‌ها^۲، از ترکیب واحدهای کوچک‌تر ساخته می‌شوند. ساده‌ترین رابط‌ها در ریو، مجموعه‌ای از کانال‌ها هستند که رفتار آن‌ها به طور مشخص توسط کاربران تعیین شده است. از دیدگاه دیگر، می‌توان ریو را به عنوان زبانی برای هماهنگی پردازش‌های همروند، یا به بیان دیگر، زبان ارتباط برای ایجاد ترکیبی از رابط‌ها نام برد که مولفه‌های گوناگون را در یک سیستم مبتنی بر مولفه هماهنگ می‌کند. تاکید ریو صرفاً بر روی رابط‌ها و ترکیب آن‌ها است و دخالتی در مولفه‌ها و موجودیت‌های دیگر سیستم ندارد. هر رابط در ریو، الگوی هماهنگی مشخصی را برای یک موجودیت یا مولفه تعیین می‌کند که باید عملیات ورودی/خروجی از طریق آن رابط انجام شود، در حالی که نیازی به دانستن جزئیات موجودیت‌ها ندارد.

ترکیب کانال‌ها در ریو، مکانیزم قدرت‌مندی برای ایجاد رابط‌ها است. قابلیت بیانگری ریو نسبت به سایر روش‌های مدل‌سازی مانند پتری‌نت، جریان داده، زبان‌های همگام‌سازی و زبان‌های پردازش جریان بیشتر است. از طرفی نشان داده شده است که الگوهای هماهنگی که توسط عبارات منظم قابل تعریف هستند، در زبان ریو و به کمک ترکیب مجموعه کوچکی از انواع کانال‌های اولیه قابل مدل‌سازی هستند.

مقدمه

طراحی پیمانه‌ای^۳ و ساخت نرم‌افزار از طریق ساخت مولفه‌های کوچک‌تر و برقراری ارتباط بین آن‌ها یکی از اساسی‌ترین روش‌های مهندسی نرم‌افزار است. این در حالیست که انتظار می‌رود مولفه‌های نرم‌افزاری مستقل از یکدیگر بوده و محیط کاربری آن‌ها تعیین‌کننده نحوه ارتباط بین هر یک از آن‌ها باشد. بنابراین، نحوه ارتباط

¹ exogenous coordination model

² connector

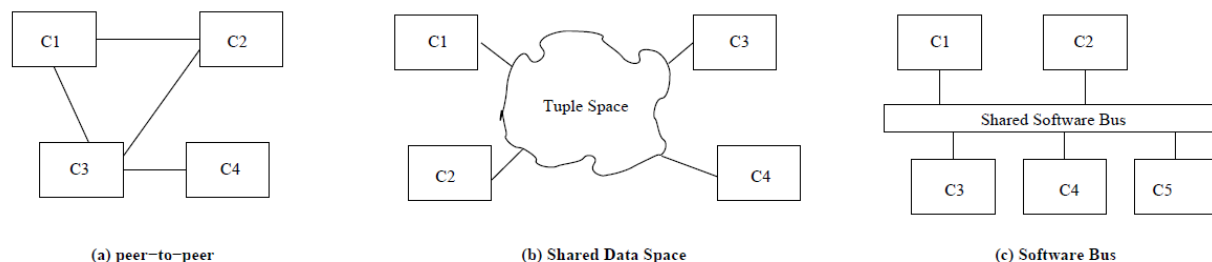
³ modular

بین مولفه‌های مختلف یک سیستم، پس از طراحی هر یک از مولفه‌ها، چالش دیگر این حوزه به شمار می‌رود. هر چقدر یک سیستم دارای پیچیدگی بیشتری باشد، چگونگی ارتباط بین مولفه‌های مختلف آن نیز پیچیده‌تر خواهد بود. در شرایطی که اغلب چگونگی ارتباط بین مولفه‌ها توسط کدهای دیگری توسعه داده می‌شود، این کدها مختص یک بستر و محیط پیاده‌سازی خاص خواهد بود و نمی‌توان از آن، برای سیستم دیگری استفاده کرد. به این ترتیب، توسعه و نگهداری سیستم‌های پیچیده، که دارای کدهای زیادی برای بیان ارتباط بیان مولفه‌ها هستند، کاری طاقت‌فرسا و هزینه‌بر می‌شود.

رویکرد پیشنهادشده در این زمینه، استفاده از کانال‌ها به عنوان پایه‌های ساختن رابطه‌ها برای بیان ارتباط بین مولفه‌های مختلف یک سیستم است. ریو از تعدادی عملیات اولیه برای ترکیب کانال‌های مختلف جهت ایجاد رابطه‌های پیچیده استفاده می‌کند. منظور از یک کانال، وسیله‌ای نقطه‌به‌نقطه برای ارتباط است که دارای دو انتهای مجزا است و می‌توان آن را به طور یکتا از دیگر کانال‌ها تمیز داد.

مدل‌های هماهنگی مبتنی بر کانال کامل هستند؛ به این معنا که می‌توانند به سادگی پایه‌های روش‌های مدل‌سازی دیگر مانند تبادل پیام، فضای اشتراکی و فراخوانی رویه از راه دور را مدل کنند. علاوه بر این، مدل مبتنی بر کانال دارای مزایای زیر است:

- کارایی: مشابه فراخوانی رویه از راه دور و تبادل پیام، مدل‌های مبتنی بر کانال از ارتباط نقطه‌به‌نقطه پشتیبانی می‌کنند. اما برخلاف مدل‌های فضای داده اشتراکی، مقصد ارتباط‌ها همیشه یکتا است و در سیستم تعیین شده است که این باعث می‌شود در سیستم‌های توزیع‌شده، بتوان پیاده‌سازی‌های کاراتری از مدل‌های نقطه‌به‌نقطه داشت.
- امنیت: در حالی که در مدل‌های داده اشتراکی، داده مربوط به همه ارتباطات توسط هر کس قابل مشاهده و استفاده است، در مدل‌های نقطه‌به‌نقطه، به جهت مشخص بودن دقیق مبدا و مقصد پیام، از افشای اطلاعات مربوط به کانال ارتباطی برای اشخاص دیگر جلوگیری می‌شود.
- بیانگری معماری: در مدل‌های اشتراکی و باس داده، معماری ثابتی در نظر گرفته می‌شود و امکان بهبود معماری وجود ندارد. در حالی که در مدل‌های ارتباطی نقطه‌به‌نقطه، می‌توان نحوه ارتباط بین مولفه‌های مختلف سیستم را به خوبی مشخص کرد و قدرت بیانگری برای تعیین معماری سیستم بسیار بیشتر است.
- بی‌نامی: منظور از ارتباط بی‌نام آن است که برای برقرار ارتباط، نیازی به شناخت طرفین نیست. در مدل‌های فراخوانی رویه‌ها از راه دور و تبادل پیام، باید طرفین یک ارتباط دقیقاً مشخص باشند تا بتوان از این مدل‌ها استفاده کرد. در مدل‌های مبتنی بر کانال، ارتباط بی‌نام قابل مدل‌سازی است.

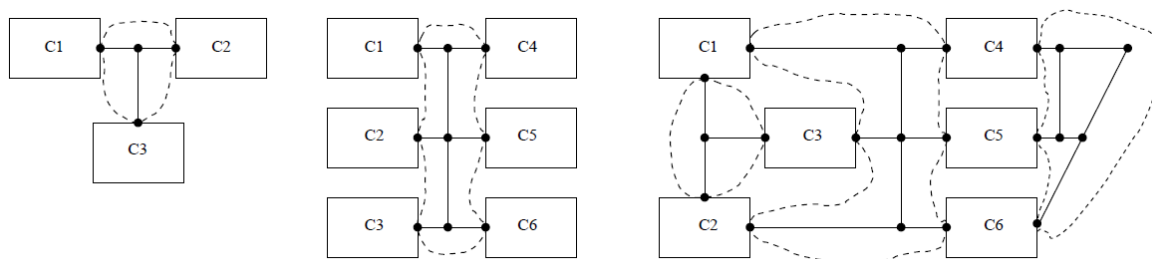


شکل ۱- قدرت بیانگری معماری مدل‌های مختلف

مفهوم اساسی در ریو، مفهوم «هماهنگی از بیرون» است که در آن فعالیت‌های مولفه‌ها و موجودیت‌های مختلف مستقل از جزئیات و درون هر مولفه هماهنگ می‌شود. ریو مبتنی بر حساب کانال‌ها است که در آن رابط‌های پیچیده از ترکیب رابط‌های ساده‌تر ساخته می‌شوند که ساده‌ترین آن‌ها، مجموعه دلخواهی از کانال‌ها است که رفتار آن‌ها به خوبی تعریف شده باشد. بنابراین می‌توان از ریو به عنوان کدی برای چسباندن مولفه‌های مختلف به یکدیگر استفاده کرد.

موجودیت‌ها در ریو می‌توانند بخش‌هایی از کدهای ترتیبی، اشیای فعال یا منفعل، ریشه‌ها، پردازنده‌ها، عامل‌ها یا مولفه‌های نرم‌افزاری باشد. در ریو به این موجودیت‌ها، به طور کلی، اصطلاحاً نمونه‌های مولفه گفته می‌شود. از منظر ریو، یک سیستم از تعدادی نمونه مولفه در حال اجرا در یک یا چند موقعیت تشکیل شده است که فعالیت‌های آن‌ها از طریق رابط‌هایی هماهنگ می‌شود. یک نمونه مولفه، مجموعه ناتمامی از موجودیت‌های فعال است که تنها راه ارتباطی آن‌ها با دیگر موجودیت‌ها، عملیات ورودی/خروجی است که در مجموعه‌ای از پایانه‌های کانال مرتبط با آن مولفه انجام می‌شود. با توجه به این‌که ریو تنها به ارتباط بین مولفه‌ها تاکید دارد، پس هر گونه جزئیات مربوط به داخل نمونه مولفه‌ها خارج از مدل‌سازی در نظر گرفته می‌شود.

در ادامه، بخش‌هایی از نحو^۴ و معناشناخت^۵ این زبان به همراه مثال‌هایی آمده است.



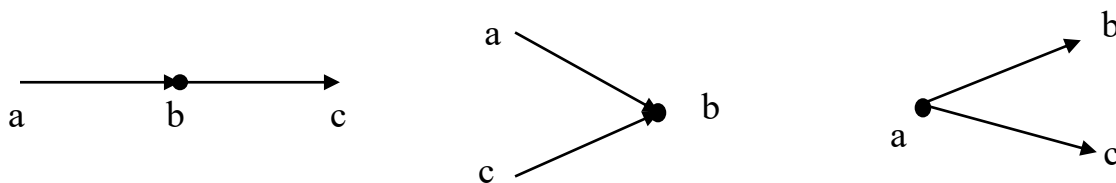
شکل ۲- نمایی از مولفه‌ها و رابط‌ها

^۴ syntax^۵ semantics

نحو مدل

در این زبان مدل‌سازی، تنها روش ارتباط بین دو نمونه مولفه، یک کانال است. یک کانال، یک رابط اتمیک در ریو به شمار می‌رود که با ترکیب و در کنار هم قرارگرفتن آن‌ها، رابط‌های پیچیده‌تری ساخته می‌شود. هر کانال به طور یکتا قابل تشخیص است. هر کانال دارای دو پایانه جهت‌دار است که وظیفه ورودی و خروجی داده در آن کانال برعهده آن‌ها است. دو نوع پایانه کانال مبدأ^۶ و مقصد^۷ وجود دارد. یک پایانه کانال از نوع مبدأ، داده را به درون کانال می‌پذیرد و پایانه از نوع مقصد، داده را به خارج از آن کانال توزیع می‌کند. کانال‌های مورد استفاده در ریو منحصراً برای انتقال داده از طریق عملیات ورودی/خروجی انجام‌شده در پایانه‌ها به کار گرفته می‌شوند.

هر رابط مجموعه‌ای از پایانه‌های کانال و کانال‌های متصل به آن‌ها است که در قالب یک گراف شامل گره‌ها و یال‌ها قابل بیان است. به هر یک از گره‌ها می‌تواند صفر یا بیشتر پایانه کانال منتسب شود. هر پایانه کانال فقط به یک گره مربوط می‌شود. همچنین هر یال در این گراف، بیانگر یک کانال بین دو گره است. لازم به ذکر است که ممکن است یک گره در رابط‌های ساخته‌شده در ریو، به صورت ترکیبی (هم از نوع مبدأ و هم از نوع مقصد) باشد.



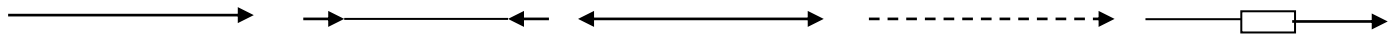
شکل ۳ - انواع گره (به ترتیب از راست به چپ): مبدأ، مقصد، ترکیبی

برای ارتباط بین گره‌های مختلف، از انواع مختلفی از کانال‌ها می‌توان استفاده کرد که توسط کاربر مشخص می‌شود. به طور کلی می‌توان انواع کانال‌ها را به دو دسته همگام و غیرهمگام دسته‌بندی کرد. کانالی را همگام می‌نامیم اگر موفقیت انجام جفت عملیات دو طرف کانال منوط به آمادگی همزمان دو پایانه باشد. در غیر این صورت، کانال ناهمگام است. به طور مثال، اگر یک پایانه آماده نوشتن و پایانه دیگر آماده برداشتن از کانال باشد، می‌توان کانال همگام انتقال داده را انجام دهد. همچنین، کانالی را با اتلاف می‌نامیم اگر تحویل داده به پایانه دیگر کانال لزوماً انجام نشود یا فقط بخشی از داده منتقل شود. بنابراین، تضمینی برای انتقال کامل داده در این نوع کانال وجود ندارد.

^۶ source

^۷ sink

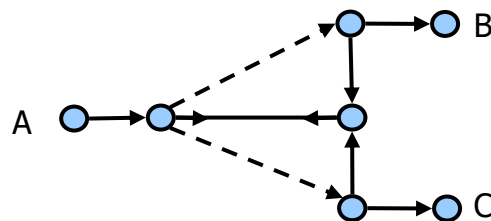
طبق دسته‌بندی بالا، همان‌طور که در شکل ۴ قابل مشاهده است، بخشی از انواع کانال‌ها عبارتند از: همگام، همگام زه‌کشی، همگام فواره‌ای، همگام با ائتلاف و ناهمگام.



شکل ۴-انواعی از کانال‌ها (به ترتیب از چپ به راست): همگام، همگام زه‌کشی، همگام فواره‌ای، همگام با ائتلاف و ناهمگام

در کانال همگام، یک پایانه آماده نوشتن در کانال و پایانه دیگر آماده برداشتن از کانال به طور همزمان است. در کانال همگام زه‌کشی، هر دو طرف آماده نوشتن روی کانال هستند و در کانال همگام فواره‌ای، هر دو پایانه آماده برداشتن داده از کانال هستند. در کانال‌های ناهمگام، بافری در کانال در نظر گرفته می‌شود که می‌تواند مقدار حافظه آن متفاوت باشد. عملکرد بافر موجود در کانال مشابه صف است.

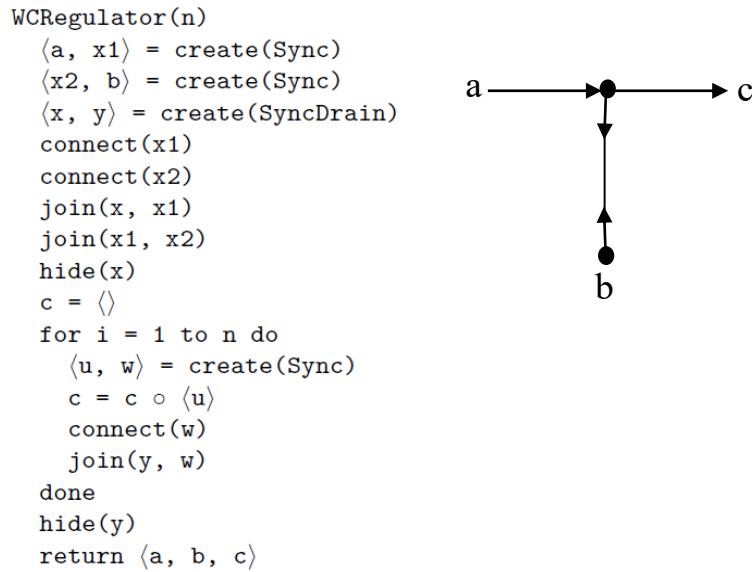
به این ترتیب و با معرفی انواع گره‌ها و کانال‌ها، می‌توان رابط‌های مختلفی را ساخت و از ترکیب آن‌ها به رابط‌های پیچیده‌تری دست پیدا کرد. به طور مثال، برای ایجاد یک رابط که دارای عملکرد یای انحصاری (XOR) است، می‌توان رابط زیر را پیشنهاد کرد.



شکل ۵- رابط پیشنهادی برای مدل‌سازی عملکرد یای انحصاری

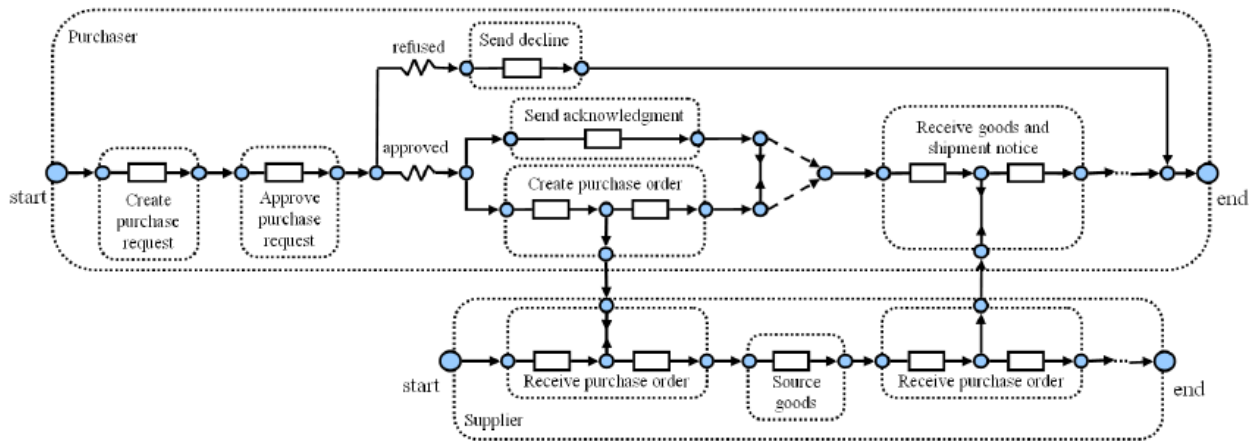
همان‌طور که تا اینجا مطرح شد، نحو این زبان به صورت گرافیکی و مشابه مدارات منطقی و الکتریکی است. اما هر یک از این مدارات را می‌توان به کد تبدیل کرد. به بیان دیگر، نحو دیگر این زبان شامل گزاره‌هایی مانند `hide` و `split`, `join`, `write`, `take`, `read`, `wait`, `disconnect`, `connect`, `move`, `forget`, `create` است که می‌توان با استفاده از این گزاره‌ها نیز یک مدار ریو را مدل کرد.

به عنوان مثال، مدار زیر بیانگر تنظیم‌کننده جریان همگام برای نوشتن است. به این معنا که زمانی داده نوشته شده توسط `a` به `c` منتقل می‌شود که `b` این اجازه را داده باشد. در شکل ۶، مدار ریوی مربوط به این عملکرد و همچنین کد متناظر با آن آمده است.



شکل ۶ - مدار ریوی تنظیم‌کننده جریان همگام برای نوشتن و کد متناظر با آن

بنا بر انواع کانال‌ها و گره‌های تعریف‌شده در ریو توسط کاربر، می‌توان با ترکیب آن‌ها سیستم‌های پیچیده‌تری را نیز مدل کرد. به طور نمونه از یک سیستم در دنیای واقعی، شکل ۷ مدل‌سازی یک سیستم ثبت سفارش خرید است که از ترکیب رابط‌های مختلفی تشکیل شده است.



شکل ۷ - مدل‌سازی یک سیستم ثبت سفارش خرید در ریو

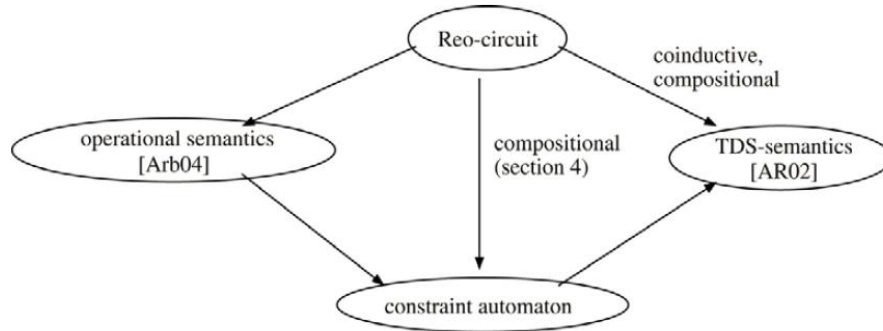
معناشناخت مدل

تاکنون به کمک روش‌های مختلفی برای زبان ریو معناشناخت‌هایی ارائه شده است. معناشناخت‌های صوری ارائه‌شده برای این زبان در قالب معناشناخت هم‌جبری^۸، مبتنی بر جریان‌های داده زمان‌دار^۹، خودکاره

^۸ coalgebraic semantics

^۹ timed-data streams

قیددار^{۱۰}، معناشناخت عملیاتی (پیاپی‌سازی شده در Maude) و منطق خطی شهودگرایانه هستند. در اینجا به طور مختصر درباره معناشناخت مبتنی بر خودکاره قیددار و جریان‌های داده زمان‌دار اشاره می‌شود. ارتباط بین این معناشناخت‌ها در شکل ۸ قابل مشاهده است.



شکل ۸ - ارتباط بین معناشناخت‌های مختلف برای زبان ریو

یک خودکاره قیددار را می‌توان به صورت چهارتایی $A = (Q, N, \rightarrow, Q_0)$ تعریف کرد که در آن:

- Q مجموعه متناهی از حالت‌ها،
- N مجموعه متناهی از نام‌ها،
- رابطه \rightarrow ، تابع گذار و زیرمجموعه‌ای از $Q \times 2^N \times DC \times Q$ ، و
- $Q_0 \subseteq Q$ مجموعه حالات آغازین است.

در واقع، هر حالت بیانگر زوج قید همگام‌سازی و قید داده است و یک گذار به صورت زیر تعریف می‌شود:

$$q \xrightarrow{N, g} p$$

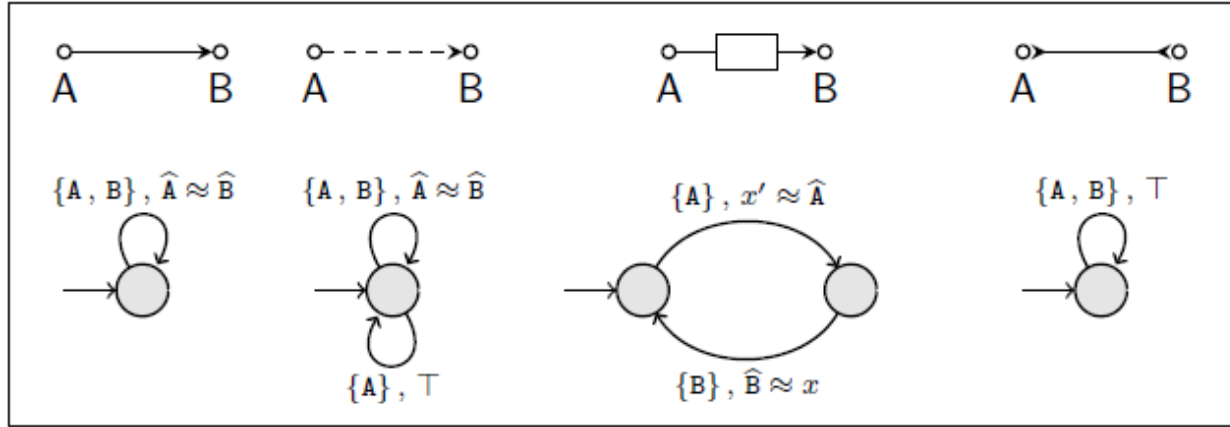
$$g \in DC(N, Data)$$

$$data(A) = d_A : A \in N$$

بنا بر تعریف انجام‌شده، می‌توان هر یک از کانال‌های ریو را در قالب یک خودکاره قیددار بیان کرد (شکل ۹). به این ترتیب، معنای یک مدار ریویی را می‌توان به کمک ضرب^{۱۱} خودکاره‌های تشکیل‌دهنده آن توصیف کرد. در شکل ۱۰، نحوه ضرب دو خودکاره قیددار به طور صوری بیان شده است.

¹⁰ constraint automata

¹¹ production



شکل ۱۰ - بخشی از معناساخت کانال‌ها در قالب خودکاره‌های قیددار

$$\mathcal{A}_1 = (\mathcal{Q}_1, \mathcal{N}_1, \longrightarrow_1, \mathcal{Q}_{0,1})$$

$$\mathcal{A}_2 = (\mathcal{Q}_2, \mathcal{N}_2, \longrightarrow_2, \mathcal{Q}_{0,2})$$

$$\mathcal{A}_1 \bowtie \mathcal{A}_2 = (\mathcal{Q}_1 \times \mathcal{Q}_2, \mathcal{N}_1 \cup \mathcal{N}_2, \longrightarrow, \mathcal{Q}_{0,1} \times \mathcal{Q}_{0,2})$$

$$\frac{q_1 \xrightarrow{N_1, g_1}_1 p_1, \quad q_2 \xrightarrow{N_2, g_2}_2 p_2, \quad N_1 \cap \mathcal{N}_2 = N_2 \cap \mathcal{N}_1}{\langle q_1, q_2 \rangle \xrightarrow{N_1 \cup N_2, g_1 \wedge g_2} \langle p_1, p_2 \rangle}$$

$$\frac{q_1 \xrightarrow{N, g}_1 p_1, \quad N \cap \mathcal{N}_2 = \emptyset}{\langle q_1, q_2 \rangle \xrightarrow{N, g} \langle p_1, q_2 \rangle}$$

شکل ۹ - ضرب خودکاره‌های قیددار

همچنین می‌توان به کمک مفهوم جریان داده‌های زمان‌دار نیز معناساخت این مدل را ارائه کرد. منظور از این معناساخت آن است که هر حالت در قالب یک زوج داده و زمان بیان می‌شود و هر کانال ریو، بیانگر شروطی روی حالت‌های پشت سر یکدیگر است.

$$TDS = \{ \langle \alpha, a \rangle \in Data^\omega \times \mathbb{R}_+^\omega \mid \forall n \geq 0 : a_n < a_{n+1} \text{ and } \lim_{n \rightarrow \infty} a_n = \infty \}$$

به طور نمونه، در ادامه معنای کانال‌های همگام، همگام زه‌کشی و ناهمگام را در قالب جریان داده‌های زمان‌دار توصیف شده است.

$$\langle \alpha, a \rangle \text{ Sync } \langle \beta, b \rangle \equiv \alpha = \beta \wedge a = b$$

$$\langle \alpha, a \rangle \text{ SyncDrain } \langle \beta, b \rangle \equiv a = b$$

$$\langle \alpha, a \rangle \text{ FIFO1 } \langle \beta, b \rangle \equiv \alpha = \beta \wedge a < b < a'$$

به این ترتیب می‌توان نحوه عملکرد یک مدار ریو را به کمک این نوع معناشناخت نیز تعبیر کرد. برای مثال، مدار تکرارکننده به شرح زیر قابل توصیف است:

$$R(\langle \alpha, a \rangle; \langle \beta, b \rangle, \langle \gamma, c \rangle) \equiv \alpha = \beta = \gamma \wedge a = b = c$$



ابزار

یکی از نقاط قوت اصلی این زبان مدل‌سازی، وجود ابزارهای کاربردی برای آن است که در اختیار کاربران قرار می‌گیرد. افزونه‌ای برای Eclipse ارائه شده است که می‌توان به صورت گرافیکی، رابط‌ها و مدارهای ریو را طراحی کرد و با پویانمایی عملکرد آن را شبیه‌سازی کرد. همچنین می‌توان به کمک این ابزار، از روی مدارهای گرافیکی، کد را به طور خودکار تولید و اجرا کرد. ضمناً به کمک واری‌گری‌های مدل Vereofy و mCRL2 رابط‌های طراحی‌شده را واری‌گری کرد.

نتیجه‌گیری

ریو به عنوان یک زبان برای مدل‌سازی پروتکل‌ها و ارتباط بین مولفه‌های مختلف یک سیستم معرفی شده است. مزیت اصلی این زبان، قابلیت ترکیب‌پذیری آن است. به این معنا که تعامل و ارتباط بین مولفه‌ها به عنوان مفهوم اصلی در نظر گرفته می‌شود و بر اساس تعریف کانال‌های مختلف و ترکیب آن‌ها با یکدیگر، رابط‌های مولفه‌ها ساخته می‌شود. نحو این زبان به طور گرافیکی و به کمک نمادهای مشابه مدارهای منطقی و الکتریکی است و رابط‌ها به شکل گراف‌هایی مشخص می‌شود. معناشناخت این زبان در قالب‌های گوناگونی مانند خودکارهای قیددار و جریان‌های داده زمان‌دار مطرح شده است. این زبان دارای ابزار قدرتمند و کاربردی‌ای است که می‌توان با استفاده از آن، مدارهای ریو را رسم، شبیه‌سازی و درستی‌یابی کرد.

مراجع

[1] F. Arbab “Reo: A Channel-Based Coordination Model for Component Composition,” *Mathematical Structures in Computer Science*, vol 14(03), pp. 329-366, 2004.

[2] C. Baier, M. Sirjani, F. Arbab, and J. Rutten “Modeling Component Connectors in Reo by Constraint Automata,” Science of Computer Programming, vol 61(2), pp. 75-113, 2006.

[3] N. Kokash, and F. Arbab, “Formal Behavior Modeling and Compliance Analysis for Service-Oriented Systems,” Formal Methods for Components and Objects, pp. 21-41, 2008.

[4] M. R. Mousavi, M. Sirjani, and F. Arbab, “Formal Semantics and Analysis of Component Connectors in Reo,” Electronic Notes in Theoretical Computer Science, vol. 154, pp. 83–99, 2006.

[5] Reo homepage, <http://reo.project.cwi.nl/reo/>

Animations from <http://reo.project.cwi.nl/webreo/>