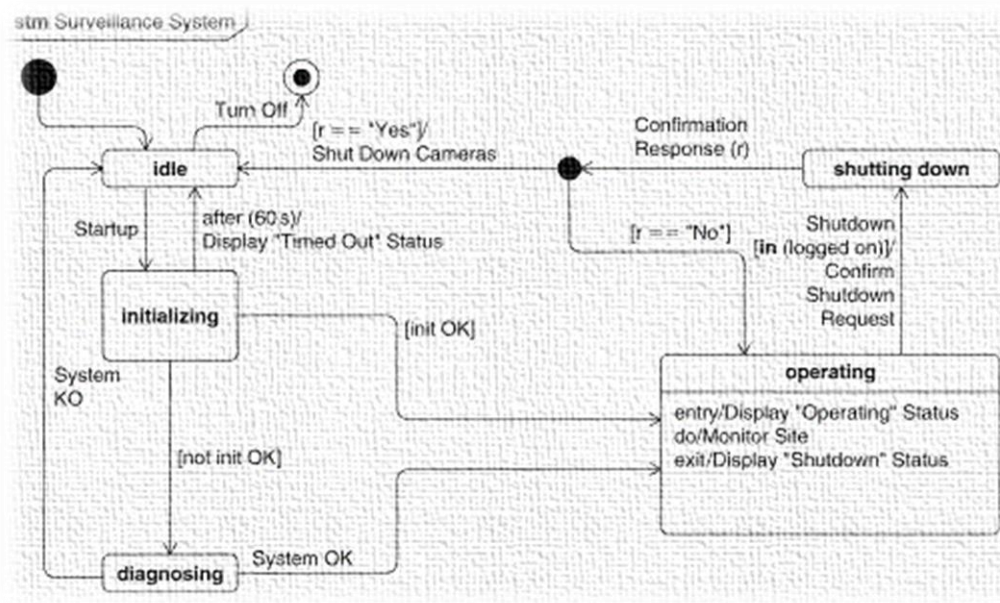


نمودار ماشین وضعیت یا State Machine diagram در UML



تهیه و تنظیم: پیمان مالکی



فهرست مطالب

3	مقدمه
5	چه زمانی و به چه دلایلی تهیه این نمودار اهمیت خواهد داشت؟
6	Behavioral State Machine
7	Vertex
7	Behavioral State
7	Simple State
8	Composite State
10	Submachine State
10	Region
10	Pseudostate
11	Initial Pseudostate
11	Terminate Pseudostate
11	Entry Point
12	Exit Point
12	Choice
13	Fork
13	Join
14	Junction
14	Shallow History Pseudostate
14	Deep History Pseudostate
14	Final State
15	Behavioral Transition
17	protocol state machine
18	Protocol State Machine
18	Protocol State
19	Protocol Transition
21	خلاصه و چکیده
24	لغت نامه تصویری
24	Simple State with name:
24	Simple State with name and activities:
24	Vertex:

24	Transition:
24	Composite State:
25	Composite State with hidden decomposition:
25	Composite Notation:
25	Region:
25	Initial Pseudostate:
25	Terminate Pseudostate
26	Final State:
26	Trigger:
26	Guard with pre-condition and post-condition:
26	Entry Point
26	Exit Point
27	Choice:
28	Join:
28	Fork:
28	shallow history pseudostate:
29	deep history pseudostate:

نمودار ماشین وضعیت^۱ یک نمودار رفتاری است که رفتار گسسته و محدود شده، بخشی از سیستم طراحی شده را از طریق انتقال^۲ وضعیت‌های گذرا نشان می‌دهد. نمودارهای ماشین وضعیت نیز می‌توانند برای بیان قرارداد استفاده از بخشی از یک سیستم استفاده شوند. دو نوع ماشین وضعیت، تعریف شده در UML 2.4 عبارتند از:

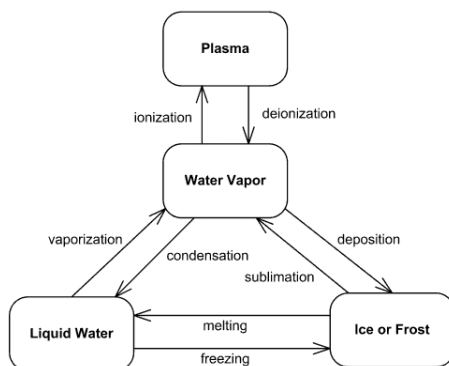
- behavioral state machine یا ماشین وضعیت رفتاری
- protocol state machine یا ماشین وضعیت پروتکل

نودها و edge های زیر معمولاً در نمودار ماشین وضعیت ترسیم می‌شوند:

- behavioral state یا وضعیت رفتاری
- behavioral transition یا انتقال رفتاری
- protocol state یا وضعیت قرارداد
- protocol transition یا انتقال قرارداد
- different pseudostates یا شبه وضعیت‌های مختلف

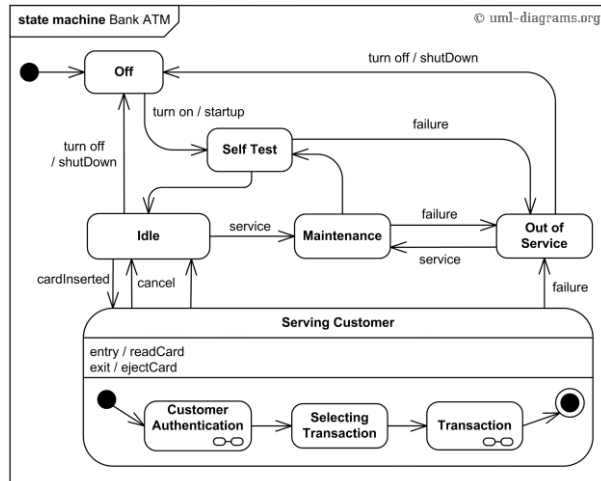
می‌توانید نمونه‌هایی از نمودارهای ماشین وضعیت را در اینجا ببینید:

- نمودار فاز آب به عنوان ماشین وضعیت

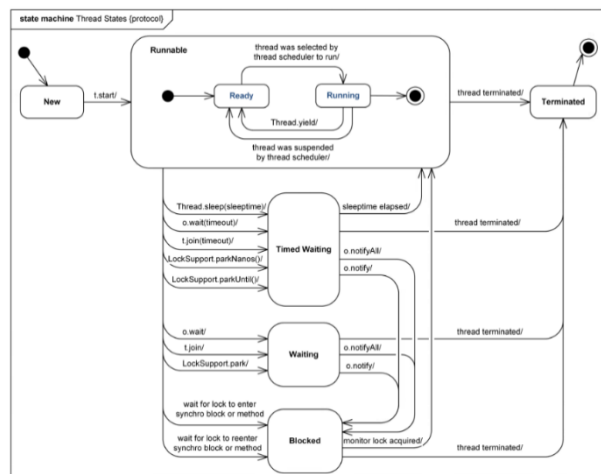


- ماشین وضعیت خودپرداز بانک

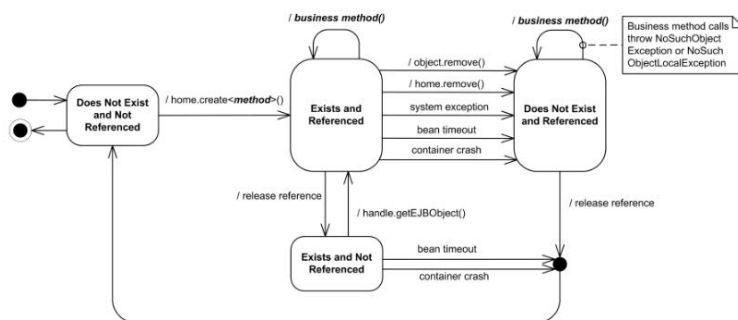
¹ State Machine diagram
² transition



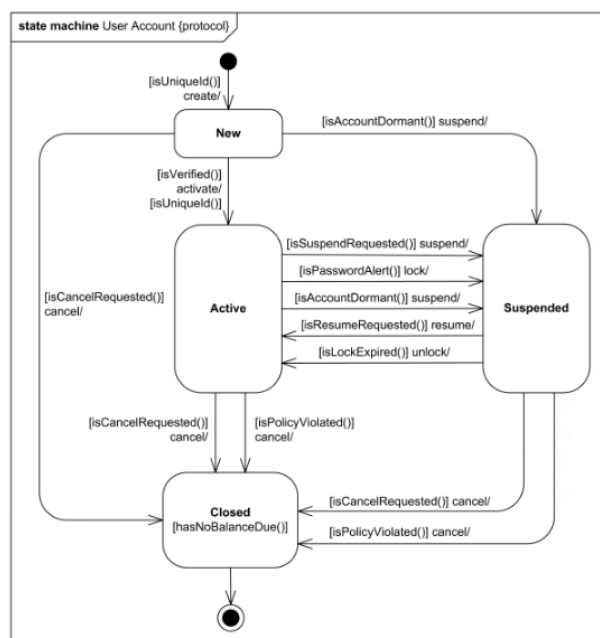
- وضعیت‌های Thread جاوا و چرخه حیات



- Session - چرخه حیات یک شیء Session - Java EJB



- نمودار ماشین وضعیت حساب کاربری



چه زمانی و به چه دلایلی تهیه این نمودار اهمیت خواهد داشت؟

نمودار ماشین وضعیت معمولاً برای توصیف رفتاری که باعث تغییر حالت یک شیء می‌شود، استفاده می‌شود. یک شیء بسته به حالتی که در آن قرار دارد به یک رویداد متفاوت پاسخ می‌دهد و این پاسخ باعث تغییر حالت آن می‌شود. نمودارهای ماشین حالت معمولاً با محوریت اشیاء طراحی می‌شوند اما می‌توانند مبتنی بر هر عنصری که رفتاری با موجودیت‌های دیگر دارد مانند: بازیگران، Use Case ها، متودها، سیستم‌های فرعی و غیره دارد نیز باشد و معمولاً همراه با نمودارهای تعامل (معمولاً نمودارهای توالی Sequence Diagram) استفاده می‌شوند.

نمودارهای وضعیت برای ارائه یک توصیف انتزاعی از رفتار یک سیستم استفاده می‌شود. این رفتار بر اساس مجموعه‌ای از رویدادها که می‌تواند یک یا چند وضعیت را بوجود آورند تهیه می‌شود.

نمودار وضعیت رفتار کلاس‌ها (Class) را در پاسخ به محرک‌های خارجی نشان می‌دهد. همانطور که گفتیم به طور خاص یک نمودار وضعیت رفتار یک شیء واحد را در پاسخ به مجموعه‌ای از رویدادها در یک سیستم توصیف می‌کند. نمودار وضعیت، یک مدل رفتاری متشکل از وضعیت‌ها، انتقال از وضعیتی به وضعیت دیگر و اقدامات یا Action ها را نشان می‌دهد.

Behavioral State Machine تخصصی رفتار است و برای تعیین رفتار مجرد بخشی از سیستم طراحی شده از طریق انتقال³ حالات گذرا استفاده می‌شود. عنوان State Machine فرمالیته که در این مورد استفاده می‌شود، یک نوع مبتنی بر شی از نمودارهای حالت Harel است. رفتارها به شکل پیمایش تصویری نودهای State مرتبط شده با انتقال، مدل‌سازی می‌شود. انتقال با ارائه مجموعه‌ای از رویدادها، تحریک می‌شود. در طول پیمایش، ماشین وضعیت، می‌تواند برخی از Activity ها را اجرا کند.

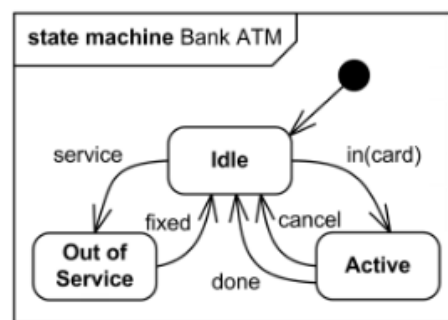
Behavioral State Machine می‌تواند متعلق به طبقه‌بندی کننده⁴ رفتاری باشد که محتوای آن نامیده می‌شود. محتوای تعریف می‌کند که کدام سیگنال و تحریک کننده برای این ماشین وضعیت، تعریف شده است و کدام مشخصات و عملیات در Activity های ماشین وضعیت موجود است. تحریک کننده‌های سیگنال و فراخوانی برای ماشین وضعیت با توجه به دریافت‌ها و عملیات این طبقه‌بندی کننده تعریف می‌شوند. ماشین وضعیت ممکن است یک مشخصه رفتاری مرتبط (specification) و متودی برای مشخصه رفتاری مورد نظر باشد. در این حالت Behavioral State Machine این مشخصه رفتاری را مشخص می‌کند. پارامترهای ماشین وضعیت با پارامترهای مشخصه رفتاری مطابقت دارند و ابزاری برای دسترسی به پارامترهای مشخصه رفتاری در ماشین وضعیت فراهم می‌کند.

مخزن رویداد⁵ برای ماشین وضعیت، مخزن رویداد نمونه با توجه به طبقه‌بندی کننده محتوای رفتاری، یا طبقه‌بندی کننده‌ای است که دارای مشخصه رفتاری است که ماشین وضعیت یک متود برای آن است.

طبقه‌بندی کننده متود یک مشخصه رفتاری از محتوای ماشین وضعیت باید طبقه‌بندی کننده‌ای باشد که دارای مشخصه رفتاری مورد نظر است. یک ماشین وضعیت بدون طبقه‌بندی کننده محتوای ممکن است از تحریک کننده‌هایی استفاده کند که مستقل از دریافت‌ها یا عملیات طبقه‌بندی کننده هستند، یعنی یا باید محرک‌های سیگنال باشد یا محرک‌های فراخوانی بر اساس پارامترهای الگوی عملیاتی (parameterized) ماشین وضعیت باشد.

ارتباط بین ماشین وضعیت و طبقه‌بندی کننده محتوی یا مشخصه رفتاری آن نماد خاصی ندارد.

ماشین وضعیت می‌تواند در یک قاب با برچسب State Machine یا به صورت اختصاری stm، ارائه شود. ناحیه محتوای فریم معمولاً خود ماشین وضعیت است اما به طور کلی می‌تواند شامل انواع دیگری از نمودارهای UML باشد.



Behavioral State Machine سطح بالا برای دستگاه خودپرداز بانک

transition³
 classifier⁴
 event pool⁵

protocol state machine زیر کلاسی از Behavioral State Machine می‌باشد.

Vertex

Vertex عنصری انتزاعی از یک نود در گراف state machine است. به طور کلی، می‌تواند منبع یا مقصد هر تعداد انتقال باشد.

زیر کلاس‌های Vertex عبارتند از:

- state یا وضعیت
- pseudostate یا شبه وضعیت

State یک Vertex یا راس است که موقعیتی را مدل می‌کند که در طی آن، شرایط ثابت^۶ (معمولاً ضمنی) برقرار است.

Behavioral State

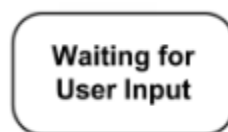
State در Behavioral State Machine ها، موقعیتی را مدل‌سازی می‌کند که در طی آن، برخی شرایط ثابت (معمولاً ضمنی) برقرار است. شرایط یا وضعیت ثابت مورد نظر ممکن است یک وضعیت ثابت مانند یک شیء در انتظار وقوع یک رویداد خارجی را نشان دهد. با این حال، می‌تواند شرایط پویا^۷ مانند فرآیند انجام برخی رفتارها را نیز مدل‌سازی کند (به عنوان مثال، عنصری در مدل مورد بررسی، زمانی که رفتار شروع می‌شود، وارد State می‌شود و به محض تکمیل رفتار آن را ترک می‌کند).

State های ارثی با خطوط خط چین‌دار یا خطوط خاکستری ترسیم می‌شوند. UML انواع State های زیر را تعریف می‌کند:

- simple state یا وضعیت ساده
- composite state یا حالت مرکب
- submachine state یا زیرماشین وضعیت

Simple State

یک Simple State، حالتی است که زیر وضعیت^۸ ندارد. آن هیچ محدوده‌ای ندارد و هیچ زیر وضعیتی ندارد. Simple State به صورت یک مستطیل با گوشه‌های گرد و نام وضعیت در داخل مستطیل نشان داده می‌شود.



Simple State به نام Waiting for Customer Input

به صورت اختیاری، State ممکن است دارای نام State در داخل یک برگه^۹ نام پیوست شده باشد. برگه نام، یک مستطیل است که معمولاً در قسمت بیرونی بالای یک State قرار دارد.

Simple State ممکن است محفظه‌هایی^{۱۰} داشته باشد. این محفظه‌های State عبارتند از:

^۶ invariant

^۷ dynamic

^۸ substate

^۹ Tab

^{۱۰} compartments

- name compartment یا محفظه نام
- internal activities compartment محفظه Activity های داخلی
- internal transitions compartment محفظه انتقال داخلی

Name Compartment نام (optional) State را به عنوان یک رشته کاراکتری نگه می‌دارد. State های بدون نام، State های anonymous نامیده می‌شوند و همگی State های متمایز^{۱۱} (different) در نظر گرفته می‌شوند. در صورت استفاده از برگه نام نباید از Name Compartment ها استفاده شود و بالعکس. توصیه می‌شود از State با نام یکسان چندین بار در یک نمودار استفاده نکنید.

بخش Activity های داخلی، فهرستی از Action های داخلی یا وضعیت (do) Activity ها (behaviors) را در خود نگه می‌دارد که در حالی که عنصر در State مورد نظر است، انجام می‌شود. برچسب Activity شرایطی را مشخص می‌کند که تحت آن رفتار مشخص شده توسط expression مربوط به Activity، فراخوانی می‌شود. expression رفتار ممکن است از هر صفت و خاتمه‌دهنده‌ای استفاده کند که در محدوده نهاد موجودیت است. برای آیتم‌های لیست که expression آنها empty است، جداکننده اسلش، اختیاری است.

چندین برچسب برای منظوره‌های خاص، رزرو شده‌اند و نمی‌توان از آنها به عنوان نام رویداد استفاده کرد. موارد زیر برچسب‌های Activity رزرو شده هستند:

- entry یا ورود (رفتار انجام شده هنگام ورود به State)
- do یا انجام (رفتار مداوم، تا زمانی که عنصر در State است انجام می‌شود)
- exit یا خروج (رفتار انجام شده در هنگام خروج از State)



Simple state به نام Waiting for Customer Input با محفظه نام و محفظه Activity های داخلی

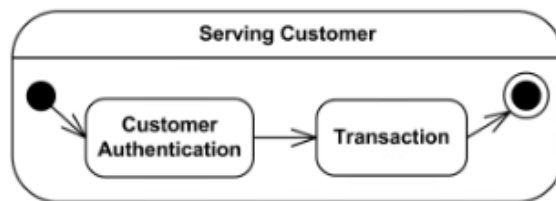
internal transitions compartment یا محفظه انتقال داخلی، حاوی لیستی از انتقال‌های داخلی است که در آن هر مورد دارای شکلی است که برای تحریک کننده، توضیح داده شده است. اگر شرایط عبارت محدود کننده، متفاوت باشد، نام هر رویداد ممکن است بیش از یک بار در هر State ظاهر شود. پارامترهای رویداد و شرایط عبارت محدود کننده، اختیاری هستند. اگر رویداد دارای پارامترهایی باشد، می‌توان آنها را در عبارت از طریق متغیر رویداد^{۱۲} فعلی استفاده کرد.

Composite State

به طور کلی، Composite State به وضعیتی گفته می‌شود که دارای زیر وضعیت‌ها (nested states) یا وضعیت‌های درونی باشد. زیر وضعیت‌ها می‌توانند متوالی (disjoint) یا همزمان (orthogonal) باشند. UML 2.4، Composite State را وضعیتی تعریف می‌کند که شامل یک یا چند ناحیه است. (توجه داشته باشید، آن ناحیه به عنوان یک بخش متعادل از یک Composite State یا یک State Machine تعریف می‌شود.) یک State مجاز نیست هم ناحیه و هم یک ماشین فرعی داشته باشد. Composite State ساده، فقط شامل یک ناحیه است.

¹¹ distinct

¹² event variable



Composite State ساده Serving Customer دو زیر وضعیت دارد

Orthogonal composite state بیش از یک ناحیه دارد. هر ناحیه دارای مجموعه‌ای از وارونه‌های متمایز متقابل^{۱۳} و مجموعه‌ای از انتقال است. یک وضعیت معین ممکن است فقط به یکی از این دو روش تجزیه شود.

هر وضعیتی که در یک ناحیه از یک **Composite State** محصور شود، زیر وضعیت آن **Composite State** نامیده می‌شود. هنگامی که در هیچ وضعیت دیگری وجود نداشته باشد، آن را یک زیر وضعیت مستقیم^{۱۴} می‌نامند. در غیر این صورت، به عنوان یک زیر وضعیت غیر مستقیم^{۱۵} شناخته می‌شود.

هر ناحیه از یک **Composite State** ممکن است یک شبه وضعیت اولیه^{۱۶} و یک وضعیت نهایی^{۱۷} داشته باشد. انتقال به وضعیت محصور، نشان دهنده، گذار به شبه وضعیت اولیه در هر ناحیه است. یک شیء تازه ایجاد شده بالاترین انتقال‌های پیش‌فرض خود را می‌گیرد که از بالاترین شبه وضعیت اولیه هر ناحیه نشأت می‌گیرد.

Composite State ممکن است دارای نام وضعیت در داخل یک برگه نام پیوست شده باشد. برگه نام یک مستطیل است که معمولاً در قسمت بیرونی بالای یک وضعیت قرار دارد. **Composite State** ممکن است دارای محفظه‌هایی باشد. محفظه‌های وضعیت عبارتند از:

- **name compartment** یا محفظه نام
- **internal activities compartment** یا محفظه Activity های داخلی
- **internal transitions compartment** یا محفظه انتقال داخلی
- **decomposition compartment** یا محفظه تجزیه

سه محفظه اول مانند **simple state** است. **decomposition compartment**، ساختار ترکیب وضعیت را به عنوان یک نمودار درونی با ناحیه‌ها، وضعیت‌ها و انتقال‌ها نشان می‌دهد. برای راحتی و ظاهر، محفظه‌های متنی ممکن است به صورت افقی در ناحیه گرافیکی کوچک شوند.

در برخی موارد، پنهان کردن تجزیه یک **Composite State** راحت است. به عنوان مثال، ممکن است تعداد زیادی وضعیت درونی در یک **Composite State** وجود داشته باشد و ممکن است به سادگی در فضای گرافیکی موجود برای نمودار جا نشوند. در آن صورت، **Composite State** ممکن است با یک نمای **simple state** با یک نماد "composite" خاص، معمولاً در گوشه سمت راست پایین نمایش داده شود. این نماد، متشکل از دو وضعیت افقی کوچک قرار گرفته در کنار هم و متصل است. این نماد یک نشانه بصری اختیاری است که نشان می‌دهد وضعیت دارای آن نماد دارای تجزیه است که در این نمودار خاص نشان داده نشده است. در عوض، محتویات **Composite State** در یک نمودار

¹³ mutually exclusive disjoint subvertices

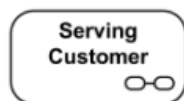
¹⁴ direct substate

¹⁵ indirect substate

¹⁶ initial pseudostate

¹⁷ final state

جداگانه نشان داده شده است. این حالت پنهان سازی، موضوعی است که باعث سهولت گرافیکی می شود و از نظر محدودیت دسترسی هیچ اهمیت معنایی ندارد.



Composite State با نام Serving Customer با نماد تجزیه پنهان

یک Composite State ممکن است یک یا چند نقطه ورود و خروج در مرز بیرونی خود یا در مجاورت آن مرز (داخل یا خارج) داشته باشد.

Submachine State

Submachine State درج مشخصات یک وضعیت به شکل زیرماشین در ماشین وضعیت^{۱۸} را مشخص می کند. ماشین وضعیتی که شامل وضعیتی به شکل زیرماشین است، ماشین وضعیت حاوی^{۱۹} نامیده می شود. همان ماشین وضعیت مورد نظر ممکن است بیش از یک بار در جایگاه یک ماشین وضعیت حاوی یک زیرماشین باشد.

وضعیت زیرماشین از نظر معنایی معادل Composite State است. نواحی Submachine State، نواحی Composite State هستند. ورود، خروج Action های رفتار و انتقال داخلی به عنوان بخشی از وضعیت، تعریف می شوند. Submachine state یک مکانیسم تجزیه است که امکان فاکتورگیری رفتارهای مشترک و استفاده مجدد از آنها را فراهم می کند.

Name Compartment (اختیاری است) نام وضعیت را به عنوان یک رشته نگه می دارد. نامی که به state machine اشاره دارد به صورت یک رشته در ادامه " " بعد از نام وضعیت نشان داده می شود.

Region

یک Region در UML 2.4 به عنوان یک بخش متعادل از یک Composite State یا یک state machine تعریف شده است. منطقه یا همان Region شامل وضعیت ها و انتقال است.

یک state machine یا Composite State با مناطقی به شکل کاشی کاری در نمودار state/state machine با استفاده از خطوط خط چین دار برای تقسیم آن به مناطق مشخص، نشان داده می شود. هر منطقه ممکن است یک نام اختیاری داشته باشد و شامل وضعیت متمایز درونی و انتقال بین آنها باشد. محفظه های متن کل وضعیت با یک خط توپر از نواحی متعادل جدا می شوند.

یک Composite State یا state machine فقط با یک منطقه با نشان دادن یک نمودار وضعیت درونی در منطقه نمودار نشان داده می شود. برای نشان دادن اینکه یک منطقه ارثی گسترش یافته است، کلمه کلیدی «extended» با نام منطقه همراه خواهد شد.

Pseudostate

شبه وضعیت یا Pseudostate، یک vertex انتزاعی است که انواع مختلفی از رؤس گذرا را در نمودار state machine در بر می گیرد.

Pseudostate ها معمولاً برای اتصال چندین انتقال به مسیرهای انتقال وضعیت پیچیده تر استفاده می شوند. به عنوان مثال، با ترکیب یک انتقال که وارد یک Pseudostate چندشاخه می شود با مجموعه ای از انتقال ها که از Pseudostate چندشاخه، خارج می شوند، یک انتقال ترکیبی به دست می آوریم که به مجموعه ای از orthogonal target state ها منجر می شود.

¹⁸ submachine state machine

¹⁹ containing state machine

Pseudostate ها عبارتند از:

- initial pseudostate یا شبه وضعیت اولیه
- terminate pseudostate یا خاتمه شبه وضعیت
- entry point یا نقطه ورود
- exit point یا نقطه خروج
- choice یا انتخاب
- join یا پیوستن
- fork یا چند شاخه
- junction یا اتصال
- shallow history pseudostate شبه وضعیت تاریخچه سطحی
- deep history pseudostate شبه وضعیت تاریخچه عمیق

Initial Pseudostate

یک Initial Pseudostate یک vertex یا راس پیش فرض را نشان می دهد که منبعی برای یک انتقال واحد به وضعیت پیش فرض یک Composite State است. در یک منطقه، حداکثر یک راس اولیه می تواند وجود داشته باشد. انتقال خروجی از راس اولیه، یک رفتار دارد، اما یک تحریک کننده یا محدود کننده ندارد. یک Initial Pseudostate به صورت یک دایره کوچک توپر نشان داده می شود.

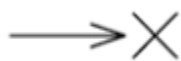


Initial Pseudostate به وضعیتی به نام Waiting for User Input انتقال می یابد

در ناحیه ای از ماشین وضعیت رفتار طبقه بندی کننده، انتقال از یک Initial Pseudostate ممکن است با رویداد تحریک کننده ای که شیء را ایجاد می کند برچسب گذاری شود. در غیر این صورت، باید بدون برچسب باشد. اگر بدون برچسب باشد، هر انتقال از وضعیت محصور را نشان می دهد.

Terminate Pseudostate

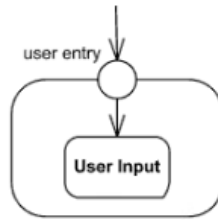
Terminate Pseudostate نشان می دهد که اجرای این state machine با استفاده از شیء محتوایی آن، خاتمه یافته است. state machine از هیچ وضعیتی خارج نمی شود و هیچ Action خروجی دیگری به جز Action های مرتبط با انتقال، منجر به اجرای Terminate Pseudostate نمی شوند. وارد کردن یک Terminate Pseudostate معادل فراخوانی یک DestroyObjectAction است. یک Terminate Pseudostate به صورت خطوط مورب متقاطع نشان داده می شود.



انتقال به Terminate Pseudostate

Entry Point

شبه وضعیت Entry Point، نقطه ورود یک state machine یا Composite State است. در هر ناحیه از state machine یا Composite State، حداکثر یک انتقال منفرد به یک vertex در همان ناحیه دارد. یک Entry Point به صورت یک دایره کوچک در مرز نمودار state machine یا Composite State با نام مرتبط با آن نشان داده می شود.

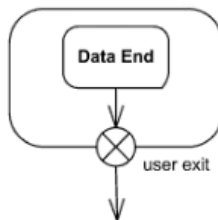


Entry Point به نام user entry

به صورت اختیاری ممکن است هم در نمودار state machine و هم خارج از مرز نمودار state machine یا Composite State قرار گیرد.

Exit Point

شبه وضعیت Exit Point، نقطه خروجی از یک state machine یا Composite State است. قراردادن یک Exit Point در هر ناحیه از state machine یا Composite State که توسط یک وضعیت زیرماشین ارجاع داده شده است، به معنای خروج از این Composite State یا وضعیت زیرماشین و شروع انتقالی است که این Exit Point را به عنوان منبع در state machine احاطه کننده زیرماشین یا Composite State دارد. یک نقطه خروج به صورت دایره کوچکی با یک ضربدر، در مرز نمودار state machine یا Composite State با نام مرتبط با آن نشان داده می‌شود.

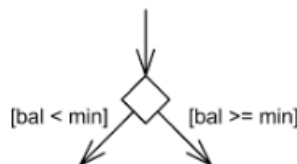


Exit Point به نام user exit

به صورت اختیاری ممکن است هم در نمودار state machine یا Composite State و هم خارج از مرز نمودار state machine یا Composite State قرار گیرد. به صورت جایگزین، نماد «براکت» را می‌توان برای نمادهای انتقال گرا، نیز استفاده کرد.

Choice

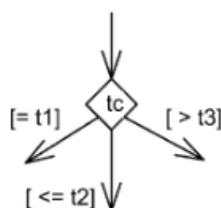
شبه وضعیت Choice یک شاخه مشروط پویا را تحقق می‌بخشد. محدود کننده‌های تحریک کننده‌های انتقال خروجی، خود را ارزیابی می‌کنند تا تنها یک انتقال خروجی را انتخاب کنند. تصمیم‌گیری در مورد مسیری که باید در پیش گرفت، ممکن است تابعی از نتایج Action های انجام شده قبلی در همان مرحله اجرا تا تکمیل باشد. انتخاب‌های پویا باید از نقاط انشعاب اتصال ایستا متمایز شوند. یک شبه وضعیت Choice به عنوان یک نماد الماس شکل نشان داده می‌شود.



انتخاب انتقال خروجی بر اساس شرایط

اگر بیش از یک محدودکننده به درستی ارزیابی شود، یک مورد، دلخواه انتخاب می‌شود. اگر هیچ یک از محدودکننده‌ها، درست ارزیابی نشود، مدل معیوب^{۲۰} در نظر گرفته می‌شود. برای جلوگیری از این امر، در صورت لزوم، یک انتقال خروجی را با محدودکننده از پیش تعریف شده «else» تعریف کنید.

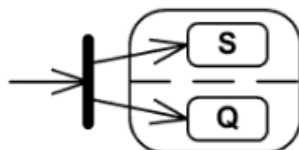
اگر همه محدودکننده‌های مرتبط با محرک‌های انتقال خروجی از شبه وضعیت Choice، عبارات باینری هستند که یک عملوند چپ مشترک دارند، می‌توان از نماد ساده‌سازی شده، استفاده کرد. عملوند سمت چپ در داخل نماد الماس گونه قرار می‌گیرد و بقیه عبارات محدودکننده روی انتقال‌های خروجی قرار می‌گیرد.



Choice بر اساس محدودکننده‌های اعمال شده بر روی مقدار داخل نماد الماس گونه

Fork

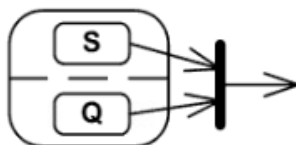
رئوس شبه وضعیت Fork برای تقسیم یک انتقال ورودی به دو یا چند انتقال که به رئوس هدف متعامد ختم می‌شوند (یعنی رئوس در مناطق مختلف یک Composite State) عمل می‌کنند. بخش‌هایی که از راس Fork خارج می‌شوند نباید محدودکننده یا تحریک کننده داشته باشند. نماد Fork یک میله پهن کوتاه است. میله مورد نظر ممکن است یک یا چند فلش به وضعیت‌ها داشته باشد. یک رشته انتقال ممکن است در نزدیکی میله مورد نظر، نشان داده شود.



Fork انتقال را به دو انتقال تقسیم می‌کند

Join

شبه وضعیت Join، چندین انتقال را که از رئوس منبع در مناطق مختلف سرچشمه می‌گیرند، ادغام می‌کند. انتقال‌هایی که وارد یک راس Join می‌شوند نمی‌توانند محدود کننده یا تحریک کننده داشته باشند. نماد Join یک میله پهن کوتاه است. میله ممکن است یک یا چند فلش از وضعیت‌های مبدا داشته باشد. یک رشته انتقال ممکن است در نزدیکی میله، نشان داده شود.



Join انتقال‌ها را به یک انتقال منفرد می‌پیوندد

Junction

رئوس شبه وضعیت Junction، رئوسی هستند که برای زنجیره‌ای کردن چندین انتقال استفاده می‌شوند. آنها برای ساخت مسیرهای انتقال ترکیبی بین وضعیت‌ها استفاده می‌شوند. به عنوان مثال، یک Junction می‌تواند برای همگرا کردن چندین انتقال ورودی به یک انتقال خروجی منفرد که یک مسیر انتقال مشترک را نشان می‌دهد استفاده شود (این به عنوان merge یا ادغام شناخته می‌شود).

برعکس، می‌توان از آنها برای تقسیم یک انتقال ورودی به بخش‌های انتقال خروجی متعدد با شرایط محدود کننده مختلف استفاده کرد. این یک شاخه شرطی ایستا را محقق می‌کند. در مورد دوم، انتقال‌های خروجی که شرایط محدود کننده آنها false ارزیابی می‌شود، غیرفعال می‌شوند.

یک محدود کننده از پیش تعریف شده به شکل "else" ممکن است حداکثر برای یک انتقال خروجی تعریف شود. این انتقال در صورتی فعال می‌شود که تمام محدود کننده‌هایی که سایر انتقال‌ها را برچسب‌گذاری می‌کنند، false باشند. یک Junction با یک دایره سیاه کوچک نشان داده می‌شود.

انتقال‌های چندگانه بدون محرک و بدون اثر که از مجموعه‌ای از وضعیت‌ها سرچشمه می‌گیرند و یک راس Junction را با یک انتقال خروجی هدف قرار می‌دهند، ممکن است به عنوان نماد وضعیت با فهرستی از نام‌های وضعیت و نماد انتقال خروجی مربوط به انتقال خروجی از محل Junction معرفی شوند.

مورد خاصی از انتقال از محل Junction که سابقه‌ای به عنوان هدف دارد، ممکن است به صورت اختیاری به عنوان هدف که نماد وضعیت فهرست حالت است، ارائه شود.

Shallow History Pseudostate

Shallow History Pseudostate، آخرین زیر وضعیت فعال از وضعیت حاوی آن را نشان می‌دهد (اما نه زیر وضعیت‌های آن زیر وضعیت). یک وضعیت مرکب یا composite state حداکثر می‌تواند یک Shallow History Pseudostate داشته باشد. انتقالی که به Shallow History Pseudostate می‌آید، معادل انتقالی است که به آخرین زیر وضعیت فعال یک وضعیت می‌رسد. حداکثر یک انتقال ممکن است از رأس تاریخچه به Shallow History Pseudostate پیش‌فرض سرچشمه بگیرد. این انتقال در صورتی انجام می‌شود که composite state قبلاً هرگز فعال نبوده باشد. Action ورود به وضعیت نشان داده شده توسط Shallow History Pseudostate انجام می‌شود. Shallow History Pseudostate با یک دایره کوچک که داخل آن "H" وجود دارد، نشان داده می‌شود. این برای منطقه وضعیت که مستقیماً آن را محصور می‌کند، اعمال می‌شود.

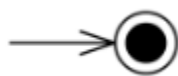
Deep History Pseudostate

شبه وضعیت Deep History Pseudostate، آخرین پیکربندی فعال composite state را نشان می‌دهد که مستقیماً حاوی این شبه وضعیت است (به عنوان مثال، پیکربندی وضعیتی که در آخرین خروج از composite state، فعال بود). یک composite state حداکثر می‌تواند یک راس Deep History Pseudostate داشته باشد. حداکثر یک انتقال ممکن است از کانکتور تاریخچه به Deep History Pseudostate پیش‌فرض، نشأت بگیرد. این انتقال در صورتی انجام می‌شود که composite state قبلاً هرگز فعال نبوده باشد. Action های ورود به وضعیت‌هایی که در مسیر مستقیم ضمنی از Deep History Pseudostate به درونی‌ترین وضعیت(های) که توسط یک Deep History Pseudostate نشان داده می‌شود، انجام می‌شود. Action ورود فقط یک بار برای هر وضعیت در پیکربندی وضعیت فعال در حال بازبینی انجام می‌شود. Deep History Pseudostate با یک دایره کوچک حاوی "H*" نشان داده می‌شود. این برای منطقه وضعیتی که مستقیماً آن را محصور می‌کند، اعمال می‌شود.

Final State

Final State نوع خاصی از وضعیت است که نشان می‌دهد منطقه محصور، کامل شده است. اگر ناحیه محصور مستقیماً در یک state machine قرار داشته باشد و سایر مناطق در state machine نیز تکمیل شوند، به این معنی است که کل state machine تکمیل شده

است. توجه داشته باشید که بنا به دلایلی، UML 2.4، Final State را به عنوان یک زیر کلاس از State تعریف می‌کند اما نه به عنوان pseudostate. Initial state یک pseudostate است.) Final State به صورت دایره‌ای نشان داده می‌شود که یک دایره کوچک پر را احاطه کرده است.



انتقال به Final State

Behavioral Transition

Transition یک رابطه مستقیم بین یک راس منبع و یک راس هدف است. ممکن است بخشی از یک انتقال ترکیبی باشد که **state machine** را از یک پیکربندی وضعیت به وضعیت دیگر می‌برد و نشان دهنده پاسخ کامل **state machine** به وقوع یک رویداد از یک نوع خاص است. نماد پیش‌فرض برای یک Behavioral Transition با BNF زیر (نسخه کمی تغییر یافته و ثابت BNF از مشخصات UML 2.4) توضیح داده شده است:

```
transition ::= [ triggers ] [ guard ] [ '/' behavior-expression ]
```

```
triggers ::= trigger [ ',' trigger ]*
```

```
guard ::= '[' constraint ']'
```

فهرست اختیاری تحریک کننده‌ها یا **triggers**، رویدادهایی را مشخص می‌کند که ممکن است باعث انتقال وضعیت شوند. یک رویداد اگر با رویداد مرتبط با تحریک کننده، مطابقت داشته باشد، یک راه‌انداز را اقناع می‌کند. از آنجایی که ممکن است بیش از یک انتقال توسط یک رویداد فعال شود، شرط لازم اما کافی برای تحریک کننده یک انتقال نیست.

قید محدود کننده²¹ یک عبارت بولی است که بر حسب پارامترهای رویداد تحریک کننده و صفات و پیوندهای شیء، نوشته شده است. قید محدود کننده، همچنین ممکن است شامل آزمایش‌های وضعیت‌های متعامد **state machine** فعلی، یا وضعیت‌های مشخص شده برخی از شیء‌های قابل دسترس (مثلاً "در حالت فعال") باشد.

در یک انتقال ساده با یک محدود کننده، محدود کننده قبل از شروع انتقال، ارزیابی می‌شود. در انتقال‌های مرکب که شامل چندین محدود کننده می‌شود، همه محدود کننده‌ها قبل از شروع انتقال، ارزیابی می‌شوند، مگر اینکه نقاط **choice** در امتداد یک یا چند مسیر وجود داشته باشد. ترتیب ارزیابی محدود کننده‌ها، تعریف نشده است. محدود کننده‌ها نباید شامل عباراتی باشند که باعث عوارض جانبی شود.

عبارت رفتار²² اگر انتقال با **transition** فعال شود و در آن زمان، اجرا می‌شود. ممکن است بر حسب **operation** ها، **attribute** ها و **link** های محتوای شیء و پارامترهای رویداد تحریک کننده یا هر ویژگی دیگری که در محدوده آن قابل مشاهده است نوشته شود. عبارت رفتار ممکن است یک توالی **Action** باشد. مثالی از انتقال با قید محدود کننده و رشته انتقال یا **transition**:

```
left-mouse-down(coordinates) [coordinates in active_window] / link:=select-link(coordinates);link.follow()
```

انتقال یا **transition** بالا به صورت زیر تفسیر می‌شود:

²¹ guard-constraint

²² behavior-expression

هنگامی که رویداد کلیک بر روی کلید سمت چپ ماوس رخ می‌دهد (تحریک کننده) و مختصات کلیک در پنجره active_window (محدود کننده) قرار دارد، پیوند، انتخاب شده و دنبال می‌شود (عبارت رفتار)، و انتقال، فعال می‌شود.

تحریک کننده‌ها و تأثیرات بعدی یک انتقال ممکن است به صورت متنی مطابق نگارش بالا یا با استفاده از نمادهای گرافیکی روی یک انتقال یا transition علامت‌گذاری شوند.

انتقال‌هایی که از composite state ها منشا می‌گیرند، انتقال‌های سطح بالا^{۲۳} یا انتقال گروه^{۲۴} نامیده می‌شوند. اگر فعال شوند، منجر به خروج همه زیر وضعیت‌های composite state می‌شوند که Activity های خروجی خود را از درون‌ترین وضعیت‌ها در پیکربندی وضعیت فعال اجرا می‌کنند.

انتقال مرکب نشان‌دهنده مسیری است که از لحاظ معنایی کامل است و از یک یا چند انتقال، ساخته شده است، که از مجموعه‌ای از وضعیت‌ها (در مقابل شبه وضعیت) سرچشمه می‌گیرد و مجموعه‌ای از وضعیت‌ها را هدف قرار می‌دهد.

یک انتقال داخلی بدون خروج یا ورود مجدد به وضعیتی که در آن تعریف شده است اجرا می‌شود. این درست است حتی اگر state machine در یک وضعیت درونی باشد.

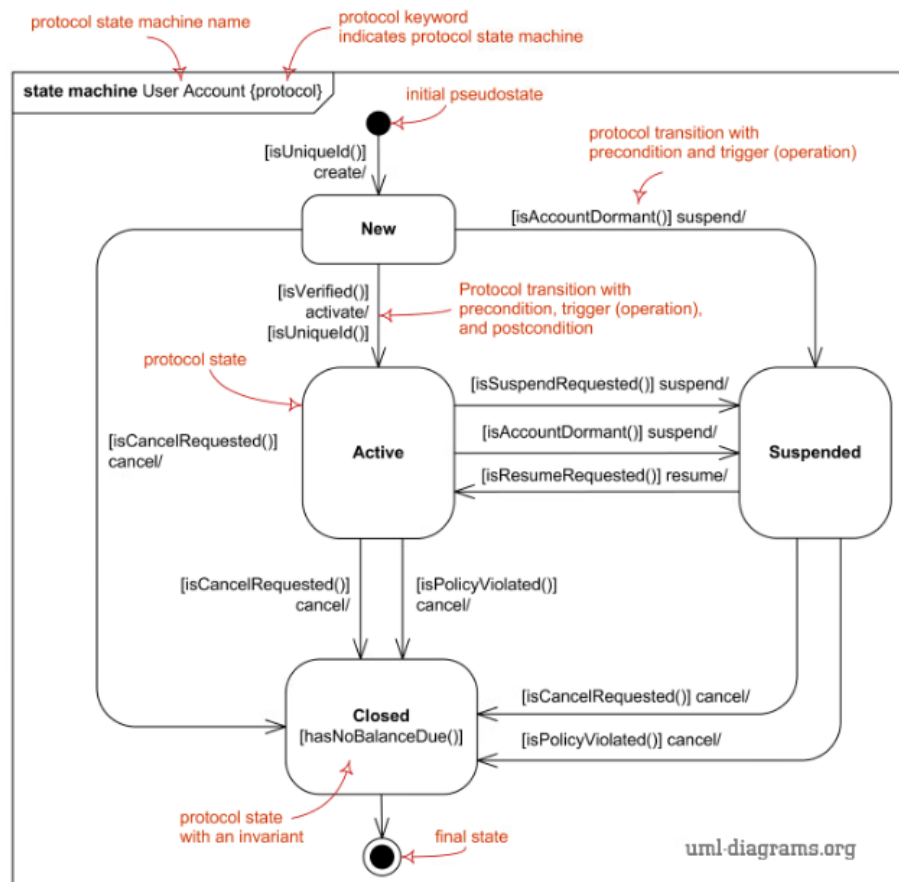
انتقال اتمام، انتقالی است که از یک وضعیت یا یک نقطه خروج سرچشمه می‌گیرد، اما دارای یک تحریک کننده صریح نیست، اگرچه ممکن است یک محدود کننده تعریف شده باشد. یک انتقال اتمام به طور ضمنی توسط یک رویداد اتمام، ایجاد می‌شود.

high-level transitions²³
group transitions²⁴

نمودارهای protocol state machine در UML برای بیان یک پروتکل یا چرخه حیات برخی طبقه‌بندی‌کننده‌ها استفاده می‌شود. نشان می‌دهد که کدام عملیات طبقه‌بندی‌کننده را می‌توان در هر وضعیت طبقه‌بندی‌کننده، فراخوانی کرد، تحت چه شرایطی خاص، و برخی از شرایط postconditions اختیاری را پس از انتقال طبقه‌بندی‌کننده به وضعیت هدف، برآورده می‌کند.

از آنجایی که این نمودارها، چرخه حیات را نشان می‌دهند، برای نشان دادن وضعیت‌های پایدار مختلف کلاسی از اشیاء که می‌توانند برای مدتی وجود داشته باشند، مفید هستند و توضیح می‌دهند که چگونه اشیاء می‌توانند در طول زمان، وضعیت خود را تغییر دهند. برای مثال، می‌توانیم نشان دهیم که چگونه حساب کاربری می‌تواند ایجاد، فعال، تعلیق و لغو شود.

عناصر اصلی نمودار protocol state machine عبارتند از وضعیت پروتکل^{۲۵}، انتقال پروتکل^{۲۶} و شبه وضعیت‌های^{۲۷} مختلف، همانطور که در نمودار کلی زیر نشان داده شده است.



نمای کلی نمودار protocol state machine

²⁵ protocol state
²⁶ protocol transition
²⁷ pseudostate

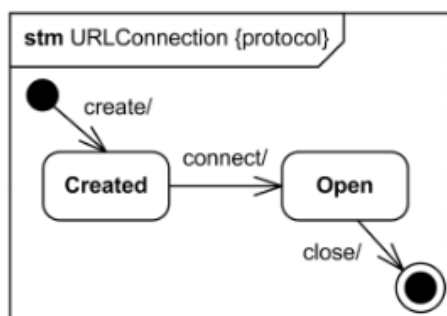
protocol state machine diagram تقریباً مترادف protocol state machine است زیرا یک نمودار معمولاً یک protocol machine را نشان می‌دهد.

Protocol State Machine

Protocol State Machine، نوع ویژه‌ای از Behavioral State Machine است و برای بیان پروتکل یا چرخه حیات یک طبقه‌بندی کننده استفاده می‌شود. مشخص می‌کند که کدام عملیات طبقه‌بندی کننده را می‌توان در چه وضعیتی و تحت چه شرایطی، فراخوانی کرد، بنابراین توالی‌های فراخوانی مجاز را روی عملیات طبقه‌بندی کننده، مشخص می‌کند. Protocol State Machine ها، انتقال‌های قانونی را بیان می‌کنند که یک طبقه‌بندی کننده، می‌تواند تحریک کند.

Protocol State Machine همیشه در محتوای یک طبقه بندی کننده، تعریف می‌شود. یک طبقه بندی کننده ممکن است چندین Protocol State Machine داشته باشد. تمام انتقال‌های یک Protocol State Machine باید انتقال پروتکل^{۲۸} باشد.

نماد Protocol State Machine مشابه Behavioral State Machine ها است. کلمه کلیدی {protocol} نزدیک به نام State Machine قرار می‌گیرد تا نمودارهای Protocol State Machine را متمایز کند.

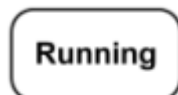


URLConnection برای Protocol State Machine

Protocol State

وضعیت‌های یک Protocol State Machine (وضعیت‌های پروتکل) یک نمای خارجی از کلاس را ارائه می‌کند که در معرض بهره‌برداران آن قرار می‌گیرد. بسته به محتوای، وضعیت‌های پروتکل می‌توانند با وضعیت‌های داخلی نمونه‌ها که توسط Behavioral State Machine ها بیان می‌شوند مطابقت داشته باشند، یا می‌توانند متفاوت باشند.

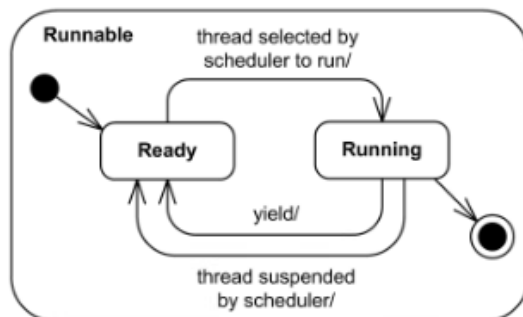
وضعیت‌های Protocol State Machine در معرض کاربران طبقه‌بندی کننده‌های محتوای آنها قرار می‌گیرد. یک وضعیت پروتکل نشان‌دهنده وضعیت پایدار طبقه‌بندی کننده محتوای آن است: وقتی نمونه‌ای از طبقه‌بندی کننده هیچ عملیاتی را پردازش نمی‌کند، کاربران این نمونه همیشه می‌توانند پیکربندی وضعیت آن را بدانند.



نمونه وضعیت پروتکل در Running

وضعیت‌های یک Protocol State Machine نمی‌تواند ورود، خروج یا انجام Action های Activity باشند. Protocol State Machine همچنین نمی‌تواند shallow history pseudostates یا deep history pseudostates داشته باشند.

Protocol State Machine ها می‌توانند وضعیت‌های زیر ماشین^{۲۹}، composite state و مناطق همزمان داشته باشند.



نمونه وضعیت پروتکل ترکیبی به نام Runnable

به عنوان مثال، مناطق همزمان، بیان پروتکل را ممکن می‌سازد که در آن یک نمونه می‌تواند چندین وضعیت فعال به طور همزمان داشته باشد. زیر ماشین‌ها و انتقال‌های ترکیبی مانند Behavioral State Machine برای فاکتورسازی Protocol State Machine پیچیده استفاده می‌شوند.

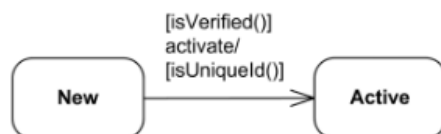
Protocol Transition

Protocol Transition، نوع ویژه‌ای انتقال (رفتاری) است که برای Protocol State Machine ها استفاده می‌شود که یک انتقال قانونی را برای یک عملیات مشخص می‌کند. Protocol Transition دارای ویژگی‌های زیر است:

- یک pre-condition یا پیش شرط (محدود کننده)
- trigger یا تحریک کننده
- یک post-condition یا پس شرط.

Protocol Transition معمولاً با عملیاتی مرتبط است که به طبقه‌بندی‌کننده محتوای Protocol State Machine تعلق دارد. Protocol Transition مشخص می‌کند که عملیات مرتبط (ارجاع شده) را می‌توان برای نمونه در وضعیت تحت شرایط اولیه (محدود کننده) فراخوانی کرد و در پایان انتقال، با برآورده شدن شرایط نهایی به حالت مقصد رسید.

انتقال مرکب همچنین می‌تواند برای Protocol State Machine ها استفاده شود. Protocol Transition به شکل یک پیکان، انتقال از راس مبدا به راس هدف، با محتوای اختیاری که انتقال را توصیف می‌کند، ارائه می‌شود.



Protocol Transition از وضعیت New به وضعیت Active با pre-condition (محدود کننده)، trigger و یک post-condition

نماد متنی برای Protocol Transition با قوانین نگارش زیر توضیح داده می‌شود (توجه داشته باشید، هیچ قانون نگارشی برای Protocol Transition در مشخصات UML 2.5 وجود ندارد، بنابراین من آنها را ایجاد کردم):

protocol-transition ::= [pre-condition] trigger '/' [post-condition]

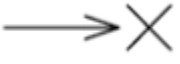
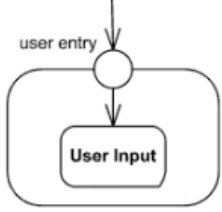
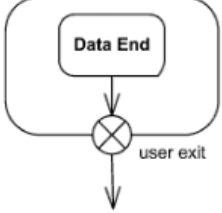
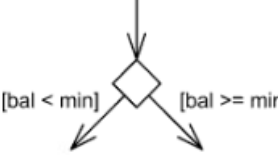
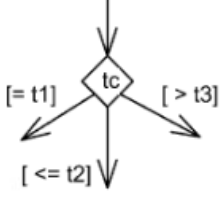
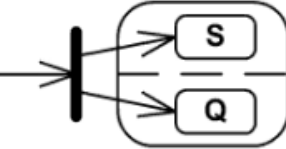
pre-condition ::= '[' constraint ']

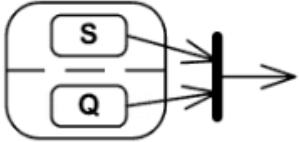
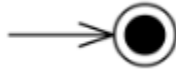
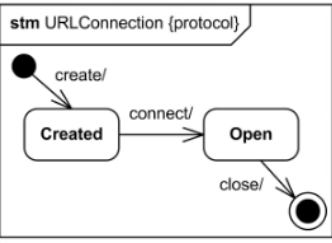
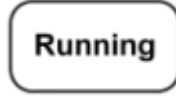
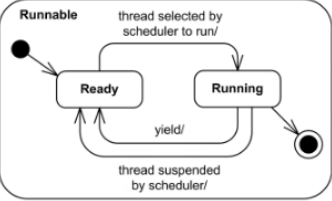
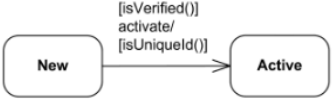
post-condition ::= '[' constraint ']

توجه داشته باشید که اگرچه مشخصات UML 2.5 می‌گوید که "نمادهای معمول StateMachine اعمال می‌شود"، و این نظر jocose است زیرا نگارش Protocol Transition (نشان داده شده در بالا) کاملاً با نگارش behavioral transition متفاوت است.

محدود کننده behavioral transition ها برای Protocol Transition ها، پیش شرط یا pre-condition نامیده می‌شود و قبل از تحریک کننده یا trigger قرار می‌گیرد. پس شرط یا Post-condition به Protocol Transition اضافه شد و بعد از تحریک کننده یا trigger ظاهر می‌شود. Protocol Transition ها هیچ عبارت رفتاری ندارند. مشخص نیست که آیا تحریک کننده یا trigger برای Protocol Transition مانند behavioral transition اختیاری است یا نه. هم دستور Protocol Transition و هم مثال‌ها، «/» را بعد از تحریک کننده یا trigger نشان می‌دهند، که با behavioral transition، که «/» نشانه‌ای است که قبل از بیان رفتار قرار می‌گیرد، متفاوت است. همه چیز دیگر "نمادهای معمول StateMachine" است.

شرح	Notation
Behavioral State Machine	
Behavioral State Machine مخصوص رفتار است و برای تعیین رفتار گسسته بخشی از سیستم طراحی شده از طریق انتقال وضعیت محدود استفاده می‌شود. State Machine فرمالیته که در این مورد استفاده می‌شود، یک نوع مبتنی بر شیء از نمودارهای حالت Harel است. State Machine می‌تواند در قابی با برچسب State Machine یا به صورت اختصاری، stm، ارائه شود.	<div><div>state machine Bank ATM</div><p>Behavioral State Machine سطح بالا برای دستگاه خودپرداز بانک</p></div>
Simple State	
Simple State وضعیتی است که زیرشاخه ندارد. Simple State به صورت یک مستطیل با گوشه‌های گرد و نام وضعیت در داخل مستطیل نشان داده می‌شود.	<div><div>Waiting for User Input</div><p>Waiting for نام Simple State Customer Input</p></div>
بخش Activity های داخلی فهرستی از Action های داخلی یا وضعیت (انجام) Activity های (رفتارها) را در خود نگه می‌دارد که در زمانی که عنصر در داخل وضعیت یا State است، انجام می‌شود. برچسب های رزرو شده برای اهداف Activity ها عبارتند از:	<div><div>Waiting for User Input</div><div>entry/ welcome exit/ thanks</div><p>Waiting for نام Simple state Customer Input با محفظه نام و محفظه Activity های داخلی</p></div>
<ul style="list-style-type: none">entry (رفتار انجام شده هنگام ورود به وضعیت یا State)do (رفتاری، که تا زمان حضور عنصر در وضعیت یا State، انجام می‌شود)exit (رفتار انجام شده در هنگام خروج از وضعیت یا State)	
Composite State	
Composite State وضعیتی است که دارای زیر وضعیت‌ها (وضعیت‌های درونی) است. UML 2.4 وضعیت Composite State را وضعیتی تعریف می‌کند که شامل یک یا چند ناحیه یا region است. یک وضعیت، مجاز به داشتن هر دو موضوع ناحیه یا region و زیر ماشین یا submachine نیست. Composite State ساده فقط شامل یک ناحیه یا region است.	<div><div>Serving Customer</div><p>Serving Composite State ساده Customer دو زیر وضعیت دارد</p></div>
Composite State ممکن است با تجزیه پنهان یا همان hidden decomposition نشان داده شود. تجزیه پنهان را با یک گرافیک شبیه به وضعیت ساده با یک نماد "composite" خاص، معمولاً در گوشه سمت راست پایین نشان داده می‌شود. این نماد از دو وضعیت یا State افقی متصل تشکیل شده است و یک نشانه بصری اختیاری است که نشان می‌دهد، وضعیت مورد نظر دارای تجزیه‌ای است که در این نمودار خاص نشان داده نشده است.	<div><div>Serving Customer</div><p>Serving Composite State با نام Customer با نماد تجزیه پنهان</p></div>
Initial Pseudostate	
یک شبه وضعیت اولیه یا Initial Pseudostate یک راس یا vertex پیش‌فرض را نشان می‌دهد که منبع یک انتقال یا transition منحصر به فرد به وضعیت پیش‌فرض یک Composite State است. در یک منطقه یا region حداکثر یک راس اولیه می‌تواند وجود داشته باشد. انتقال خروجی از راس اولیه ممکن است یک رفتار داشته باشد، اما نه یک تحریک کننده یا trigger یا محدود کننده یا guard. یک شبه وضعیت اولیه یا Initial Pseudostate به صورت یک دایره کوچک توپر نشان داده می‌شود.	<div><p>Initial Pseudostate به وضعیتی به نام Waiting for User Input انتقال می‌یابد</p></div>
Terminate Pseudostate	

<p>شبه وضعیت خاتمه یا Terminate Pseudostate نشان می‌دهد که اجرای این state machine با استفاده از شیء موجود، خاتمه یافته است. state machine دیگری دارای انتقال خروجی از هیچ وضعیتی نمی‌شود و هیچ Action خروجی دیگری به جز Action های مرتبط با انتقال منجر به شبه وضعیت خاتمه یا Terminate Pseudostate. انجام نمی‌دهد. ورود به یک شبه وضعیت خاتمه یا Terminate Pseudostate معادل فراخوانی یک DestroyObjectAction است. یک شبه وضعیت خاتمه یا Terminate Pseudostate به صورت خطوط مورب متقاطع، نشان داده می‌شود.</p>	 <p>انتقال به Terminate Pseudostate</p>
Entry Point	
<p>شبه وضعیت نقطه ورود یا Entry Point، نقطه ورود یک state machine یا Composite State است. در هر ناحیه یا region از state machine یا Composite State، حداکثر یک انتقال یا transition منحصر به فرد به یک راس یا vertex در همان ناحیه یا region دارد. یک نقطه ورودی یا Entry Point به صورت یک دایره کوچک در مرز نمودار state machine یا Composite State با نام مرتبط با آن نشان داده می‌شود.</p>	 <p>Entry Point به نام user entry</p>
Exit Point	
<p>شبه وضعیت یا pseudostate نقطه خروج، نقطه خروجی یک ماشین وضعیت یا state machine یا وضعیت مرکب یا composite state است. وارد کردن یک نقطه خروجی در هر ناحیه از وضعیت مرکب یا composite state یا ماشین وضعیت یا state machine که توسط یک وضعیت زیرماشین یا submachine، ارجاع داده شده است، به معنای خروج از این وضعیت مرکب یا composite state یا وضعیت زیرماشین یا submachine و شروع انتقالی است که این نقطه خروج به عنوان منبع در ماشین وضعیت یا state machine است که زیرماشین یا submachine یا وضعیت مرکب یا composite state را در بر می‌گیرد. وضعیت مرکب یا composite state یک نقطه خروج به صورت دایره کوچکی با یک ضربدر در مرز نمودار ماشین وضعیت یا state machine یا وضعیت مرکب یا composite state با نام مرتبط با آن نشان داده می‌شود.</p>	 <p>Exit Point به نام user exit</p>
Choice	
<p>شبه وضعیت Choice یا انتخاب یک شاخه‌بندی شرطی پویا را تحقق می‌بخشد. محدودکننده‌ها یا guard ها، تحریک‌کننده‌های یا trigger های انتقال خروجی خود را ارزیابی می‌کند تا تنها یک انتقال خروجی را انتخاب کند. یک شبه وضعیت Choice یا انتخابی به عنوان نمادی به شکل الماس نشان داده می‌شود.</p>	 <p>Choice یا انتخاب انتقال خروجی بر اساس شرایط</p>
<p>اگر ارزیابی بیش از یک محدودکننده یا guard مقدار true داشته باشد، یک مورد به دلخواه انتخاب می‌شود. اگر ارزیابی هیچ یک از محدودکننده‌ها یا guard ها مقدار true نداشته باشد، مدل، نامناسب در نظر گرفته می‌شود. برای جلوگیری از این امر، در صورت لزوم، یک انتقال خروجی را با محدودکننده یا guard از پیش تعریف شده «else» تعریف کنید. اگر همه محدودکننده‌ها یا guard های مرتبط با محرک‌ها یا trigger های انتقال خروجی از شبه وضعیت انتخابی یا choice، عبارات باینری هستند که یک عملوند چپ مشترک دارند، می‌توان از نماد ساده‌سازی شده استفاده کرد. در این حالت، عملوند سمت چپ در داخل نماد الماس شکل، قرار می‌گیرد و بقیه عبارات محدودکننده یا guard روی انتقال‌های خروجی قرار می‌گیرد.</p>	 <p>Choice یا انتخاب بر اساس محدودکننده‌های یا guard های اعمال شده بر روی مقدار یا عبارت داخل نماد الماس گونه</p>
Fork	
<p>رئوس شبه وضعیت چندشاخه‌کننده یا Fork برای تقسیم یک انتقال ورودی به دو یا چند انتقال که به رئوس یا Vertices هدف متعادل ختم می‌شوند (یعنی رئوسی در مناطق مختلف یک وضعیت ترکیبی یا composite state) عمل می‌کنند. بخش‌هایی که از رئوس چندشاخه‌کننده یا Fork خارج می‌شوند نباید محدودکننده یا guard یا تحریک‌کننده یا trigger داشته باشند. نماد چندشاخه‌کننده یا Fork یک میله پهن کوتاه است. این میله ممکن است یک یا چند فلش خروجی از خود به وضعیت‌ها داشته باشد. یک رشته متنی انتقال، ممکن است در نزدیکی میله نشان داده شود.</p>	 <p>چندشاخه‌کننده یا Fork انتقال را به دو انتقال تقسیم می‌کند</p>

<p>شبه وضعیت پیونددهنده یا Join چندین انتقال را که از رئوس منبع در مناطق مختلف سرچشمه می‌گیرند، ادغام می‌کند. انتقال‌هایی که وارد یک راس پیونددهنده یا Join می‌شوند، نمی‌توانند محدودکننده یا guard یا تحریک کننده یا trigger داشته باشند. نماد پیونددهنده یا Join یک میله پهن کوتاه است. این میله ممکن است یک یا چند فلش از وضعیت‌های مبدا به خود داشته باشد. یک رشته انتقال، ممکن است در نزدیکی این میله، نشان داده شود.</p>	<p>Join</p>  <p>پیونددهنده یا Join انتقال‌ها را به یک انتقال منفرد می‌پیونداند</p>
<p>وضعیت نهایی یا Final State نوع خاصی از وضعیت است که نشان می‌دهد منطقه دربرگیرنده این وضعیت (Final State) کامل شده است. وضعیت نهایی یا Final State به صورت دایره‌ای نشان داده می‌شود که یک دایره کوچک توپر را احاطه کرده است.</p>	<p>Final State</p>  <p>انتقال به یک وضعیت نهایی یا Final State</p>
<p>ماشین وضعیت پروتکل یا Protocol State Machine نوعی ماشین وضعیت رفتاری یا behavioral state machine تخصصی است و برای بیان پروتکل‌های استفاده‌شده و چرخه‌های طبقه‌بندی کننده یا lifecycle of classifier، استفاده می‌شود. مشخص می‌کند که کدام عملیات طبقه‌بندی کننده را می‌توان در چه وضعیتی و تحت چه شرایطی فراخوانی کرد، بنابراین توالی‌های فراخوانی مجاز بر اساس طبقه‌بندی کننده‌ها را مشخص می‌کند. ماشین وضعیت پروتکل یا Protocol State Machine همیشه در زمینه یک طبقه بندی کننده تعریف می‌شود. نماد ماشین وضعیت پروتکل یا Protocol State Machine مشابه ماشین وضعیت رفتاری یا behavioral state machine است. کلمه کلیدی {protocol} نزدیک به نام ماشین وضعیت یا state machine قرار می‌گیرد تا نمودارهای ماشین وضعیت پروتکل یا Protocol State Machine را متمایز کند.</p>	<p>Protocol State Machine</p>  <p>Protocol State Machine برای کلاس URLConnection</p>
<p>وضعیت‌ها یا State های یک ماشین وضعیت پروتکل یا Protocol State Machine نمی‌تواند ورود یا entry، خروج یا exit یا اجرای Action های یک Activity را داشته باشد. ماشین وضعیت پروتکل یا Protocol State Machine نیز نمی‌تواند shallow history pseudostates یا deep history pseudostates داشته باشند.</p>	<p>Simple Protocol State</p>  <p>نمونه‌ای از وضعیت پروتکل یا Protocol Running به نام State</p>
<p>ماشین وضعیت پروتکل یا Protocol state machine ها می‌توانند وضعیت‌های زیرماشین یا submachine state، وضعیت‌های ترکیبی یا composite state و مناطق همزمان یا concurrent region، داشته باشند.</p>	<p>Composite protocol state</p>  <p>نمونه وضعیت پروتکل ترکیبی یا Composite protocol state به نام Runnable</p>
<p>انتقال پروتکل یا Protocol transition می‌تواند یک پیش شرط یا pre-condition (محدود کننده)، تحریک کننده یا trigger و یک پس شرط یا post-condition از آن داشته باشد. انتقال پروتکل یا Protocol transition به شکل یک پیکان انتقال از راس یا vertex مبدا به راس یا vertex هدف، با متنی اختیاری که انتقال را توصیف می‌کند، ارائه می‌شود.</p>	<p>Protocol transition</p>  <p>Protocol Transition از وضعیت New به وضعیت Active یا pre-condition (محدود کننده)، trigger و یک post-condition</p>

Simple State with name:

وضعیت نمونه به همراه نام وضعیت



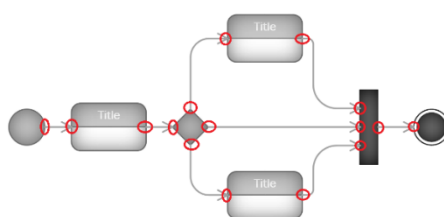
Simple State with name and activities:

وضعیت نمونه به همراه نام وضعیت و جزئیات رفتارهای یا فعالیت‌های درون آن



Vertex:

رأس



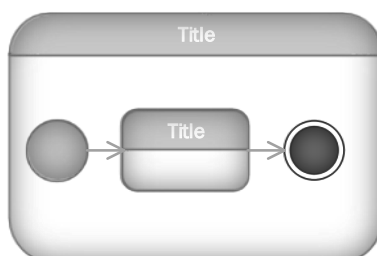
Transition:

انتقال از محل آغاز به نام منبع به محل نهایی به نام مقصد



Composite State:

وضعیت ترکیبی یا مرکب یا پیچیده با جزئیات داخلی آن



Composite State with hidden decomposition:

وضعیت ترکیبی یا مرکب یا پیچیده بدون جزئیات داخلی یا پوشیده شده



Composite Notation:

نماد معرفی ترکیبی یا مرکب یا پیچیده بودن یک وضعیت



Region:

ناحیه یا منطقه، که یک نماد طبقه‌بندی کننده بر اساس یک موضوع یا شاخص است



Initial Pseudostate:

شبه وضعیت آغازین



Terminate Pseudostate:

شبه وضعیت جمع‌آوری ماشین وضعیت و اشیاء درون آن



Final State:

وضعیت پایانی یک وضعیت یا ماشین وضعیت بدون جمع‌آوری ماشین وضعیت یا اشیاء درون آن و یا زیر وضعیت‌های دیگر



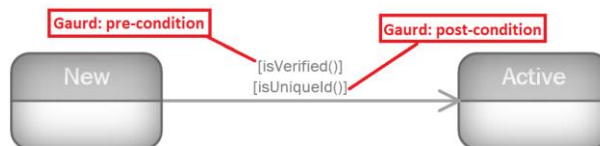
Trigger:

تحریک کننده که عامل یا رفتار اصلی انتقال است و در حین انتقال صورت می‌گیرد



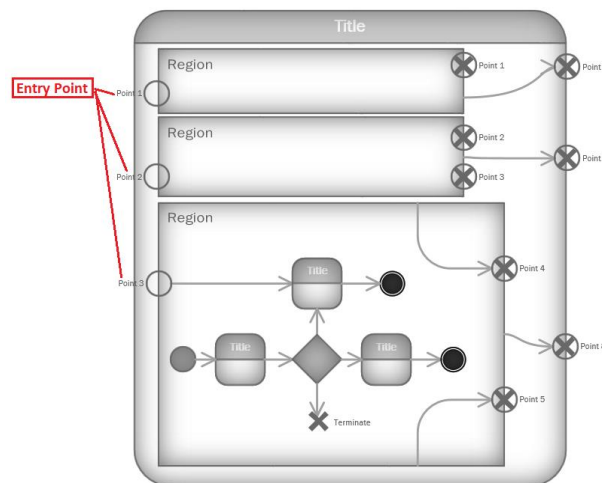
Guard with pre-condition and post-condition:

محدود کننده انتقال با پیش شرط (قبل از ورود به انتقال) و پس شرط (قبل از خروج از انتقال)



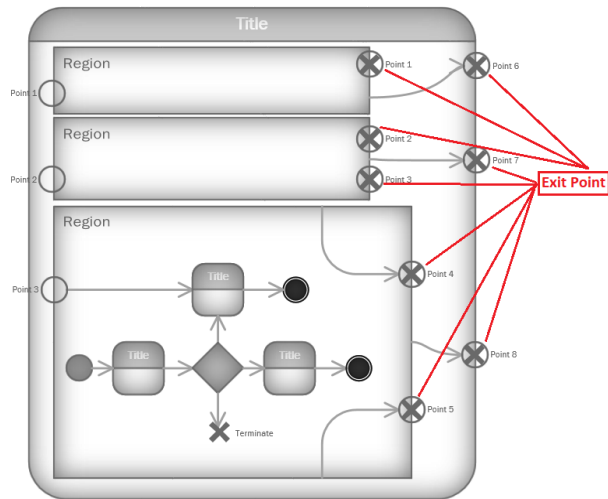
Entry Point

نقطه ورود به یک ناحیه، ماشین وضعیت یا وضعیت ترکیبی



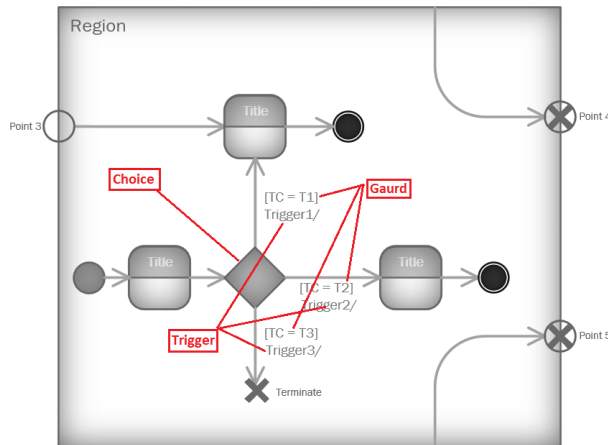
Exit Point

نقطه خروج از یک ناحیه، ماشین وضعیت یا وضعیت ترکیبی

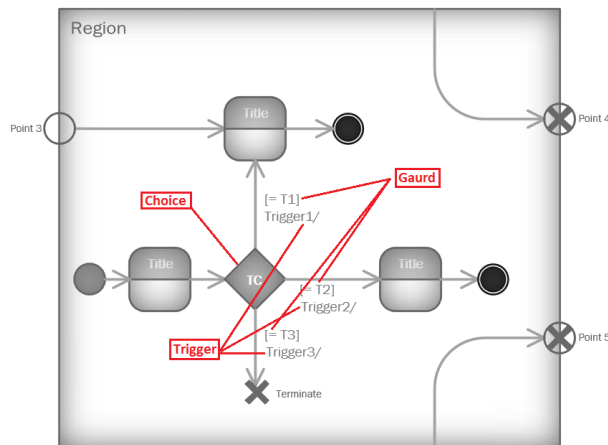


Choice:

شبه وضعیت انتخاب

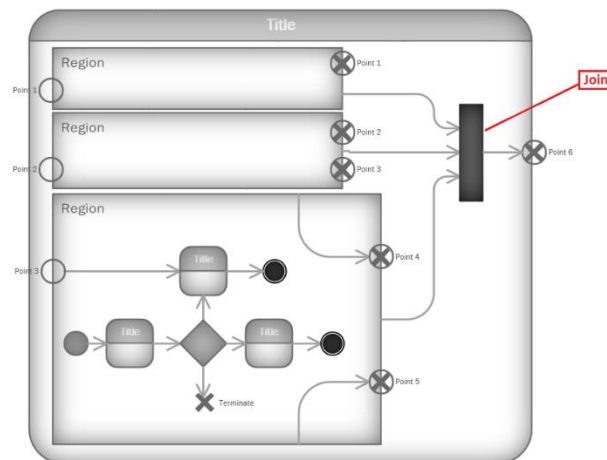


OR



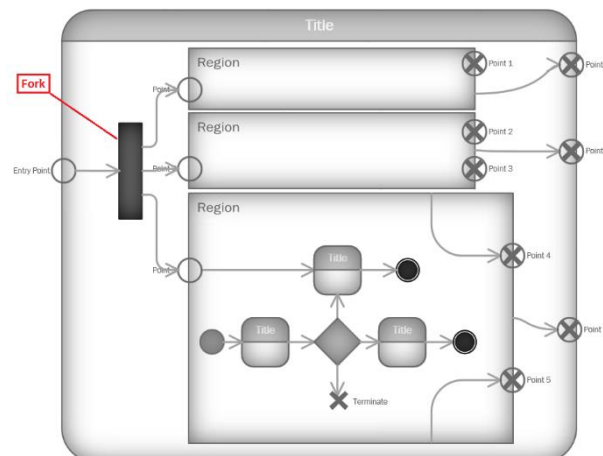
Join:

شبه وضعیت پیوند دهنده



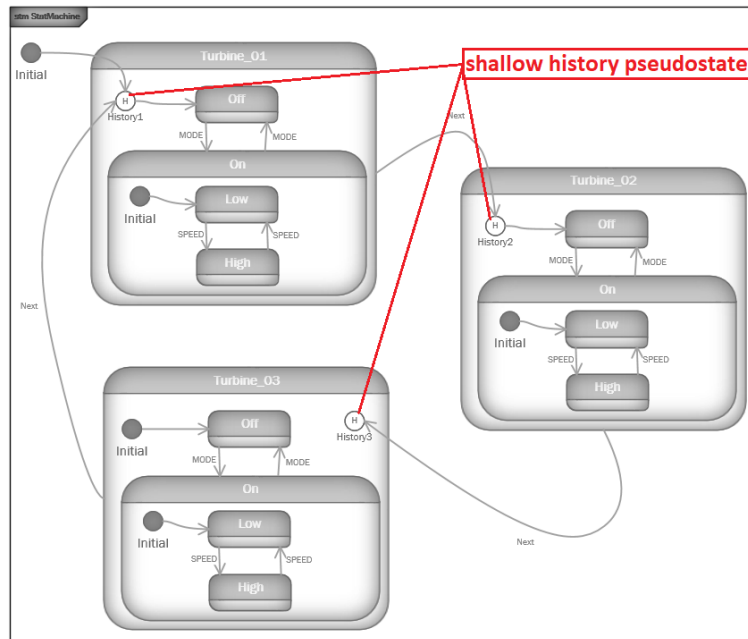
Fork:

شبه وضعیت چندشاخه کننده



shallow history pseudostate:

شبه وضعیتی که در هر لحظه، آخرین زیروضعیت درون یک وضعیت ترکیبی یا ماشین وضعیت را فقط در سطحی که درون آن قرار دارد، در خود حفظ می‌کند و می‌توان به آن مراجعه نمود و در هنگام مراجعه به آن، وضعیت از آنجایی که از محیط وضعیت ترکیبی و یا ماشین وضعیت خارج شده بود و یا وقفه افتاده بود، دنبال می‌شود. اگر وضعیتی وجود نداشته باشد، باید به واسطه یک خط انتقال، این شبه وضعیت shallow history به یک وضعیت، ناحیه یا شبه وضعیت اشاره داشته باشد.



deep history pseudostate:

شبه وضعیتی که در هر لحظه، آخرین زیروضعیت/زیروضعیت‌های سلسله مراتبی درون سلسله مراتبی از وضعیت‌های ترکیبی یا ماشین‌های وضعیت تودرتو برای تمام سطوحی که به صورت سلسله مراتب وجود دارد، حفظ می‌کند و می‌توان به آنها مراجعه نمود و در هنگام مراجعه به آنها، وضعیت از آنجایی که از درونی‌ترین زیروضعیت فعال در محیط وضعیت ترکیبی و یا ماشین وضعیت فعال، خارج شده بود و یا وقفه افتاده بود، دنبال می‌شود. اگر وضعیتی وجود نداشته باشد، باید به واسطه یک خط انتقال، این شبه وضعیت **deep history** به یک وضعیت، ناحیه یا شبه وضعیت اشاره داشته باشد.

