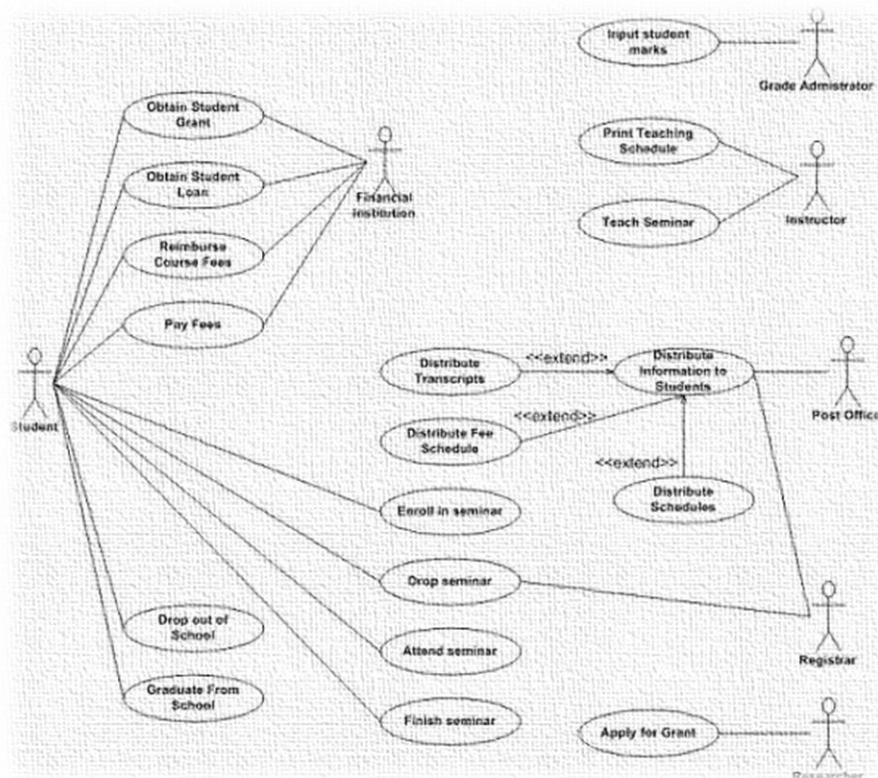


## نمودار موارد کاربرد یا Use Case diagram در UML



تهیه و تنظیم: پیمان مالکی



## فهرست مطالب

3	مقدمه
3	چه زمانی و به چه دلایلی تهیه این نمودار اهمیت خواهد داشت؟
5	Business Use Case Diagrams
6	System Use Case Diagrams
7	Use Case Subject
7	نماد
7	بازنگری ها
8	Business Model Subject
8	اشتباهات رایج
9	Software System Subject
10	Applicability of Use Cases
10	Ownership of Use Cases
11	UML Actor
12	Business Actor
13	ارتباط بین کنشگرها
14	UML Use Case
15	نامگذاری Use Case ها
15	Use Case نماد
16	Abstract Use Case
17	Business Use Case
18	Describing Use Case Behaviors
20	Relationships Between Use Cases
21	رابطه Generalization بین موارد کاربرد
21	رابطه Association بین موارد کاربرد
21	رابطه Association بین مورد کاربرد و کنشگر
22	تعداد ارتباط مجاز برای رابطه Association
24	UML Use Case Extend
24	Extension Point
26	UML Use Case Include
28	مقایسه ارتباطات بین Use Case ها

28	ارتباطات ازدست رفته Use Case ها
29	خلاصه و چکیده
35	لغتنامه تصویری
35	Subject:
35	Stereotype:
35	Actor:
36	Business Actor:
36	Actor Generalization:
36	Concrete Actor:
36	Abstract Actor:
37	Use Case:
38	Business Use Case:
38	Abstract Use Case:
38	Concrete Use Case
38	Use Case Generalization:
38	Use Case Inclusion:
39	Use Case Extension:
39	Extension Point:
39	Generalization Relationship:
40	Inclusion Relationship:
40	Extension Relationship:
41	Association Relationship:
41	Dependency Relationship:

نمودارهای مورد کاربرد<sup>۱</sup> معمولاً به نمودارهای رفتاری گفته می‌شود که برای توصیف مجموعه‌ای از اقدامات<sup>۲</sup> (مورد کاربرد) استفاده می‌شود که یک نمونه سیستم یا سیستم‌ها (یا به عبارتی موضوع/موضوع‌ها) باید یا می‌توانند با همکاری یک یا چند کاربر خارجی سیستم یا کنشگران<sup>۳</sup> انجام دهند. هر مورد کاربرد باید نتایج قابل مشاهده و ارزشمندی را برای کنشگران یا سایر ذینفعان سیستم ارائه دهد.

**توجه:** مشخصات UML 2.0 تا 2.4 نمودار مورد کاربرد را به عنوان یک توصیف کننده نمودار کلاس<sup>۴</sup> معرفی می‌کند و می‌دانیم که نمودار کلاس یک نمودار ساختاری است.

نمودارهای مورد کاربرد در واقع دو جنبه دارند. جنبه رفتاری که رفتار سیستم را توصیف می‌کند و جنبه دیگر، ساختار سیستم را توصیف می‌کند. این نمودارها به عنوان یک مورد خاص از نمودارهای کلاس هستند که در آنها طبقه‌بندی کننده‌ها به کنشگران یا موارد کاربرد مربوط به هر یک از آنها در انجمن‌ها، محدود می‌شوند

[UML 2.5 FTF - Beta 1] موارد کاربرد را از مدل‌سازی رفتار به مفاهیم تکمیلی UML منتقل کرده است. بنابراین، این یک معضل تاسف بار شده است که نمودارهای موارد کاربرد در UML، انواع مختلف پیدا کرده‌اند.

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

نمودار مورد کاربرد به جزئیات زیادی نمی‌پردازد؛ مثلاً انتظار نمی‌رود که ترتیب انجام مراحل را مدل‌سازی کند. توصیه می‌شود که از نمودارهای مورد کاربرد برای توصیفی‌تر کردن یک مورد کاربرد متنی<sup>۵</sup>، استفاده شود.

در UML، نمودارهای مورد کاربرد، رفتار یک سیستم را مدل‌سازی می‌کنند و به درک نیازهای سیستم کمک می‌کنند. نمودارهای مورد کاربرد، عملکردهای سطح بالا و حوزه یک سیستم را توصیف می‌کنند. این نمودارها همچنین تعاملات بین سیستم و کنشگران آن را مشخص می‌کنند. موارد کاربرد و کنشگران در نمودارهای مورد کاربرد توصیف می‌کنند که سیستم چه کاری انجام می‌دهد و کنشگران چگونه از آن استفاده می‌کنند، اما در خصوص نحوه عملکرد داخلی سیستم توضیحی نمی‌دهند.

نمودارهای مورد کاربرد، محتوی و الزامات کل سیستم یا بخش‌های مهم سیستم را نشان می‌دهند و تعریف می‌کنند. شما می‌توانید یک سیستم پیچیده را با یک نمودار مورد کاربرد، مدل‌سازی کنید، یا نمودارهای مورد کاربرد متعدد برای مدل‌سازی اجزای سیستم ایجاد کنید. شما معمولاً نمودارهای مورد کاربرد را در مراحل اولیه پروژه ایجاد می‌کنید و در طول فرآیند توسعه به آنها مراجعه می‌کنید.

نمودارهای مورد کاربرد در شرایط زیر مفید هستند:

- قبل از شروع یک پروژه، می‌توانید نمودارهای مورد کاربرد را برای مدل‌سازی یک کسب‌وکار ایجاد کنید تا همه شرکت‌کنندگان در پروژه درک درستی از پرسنل، مشتریان و فعالیت‌های کسب‌وکار مورد نظر، داشته باشند.
- هنگام استخراج نیازمندی‌ها، می‌توانید نمودارهای مورد کاربرد را ایجاد کنید تا نیازمندی‌های سیستم را به تصویر بکشید و آنچه را که سیستم باید انجام دهد به دیگران ارائه دهید.
- در طول مراحل تجزیه و تحلیل و طراحی، می‌توانید از موارد کاربرد و کنشگران نمودارهای مورد کاربرد خود برای شناسایی کلاس‌های مورد نیاز سیستم، استفاده کنید.

<sup>1</sup> Use Case

<sup>2</sup> Actions

<sup>3</sup> Actor

<sup>4</sup> Class Diagram

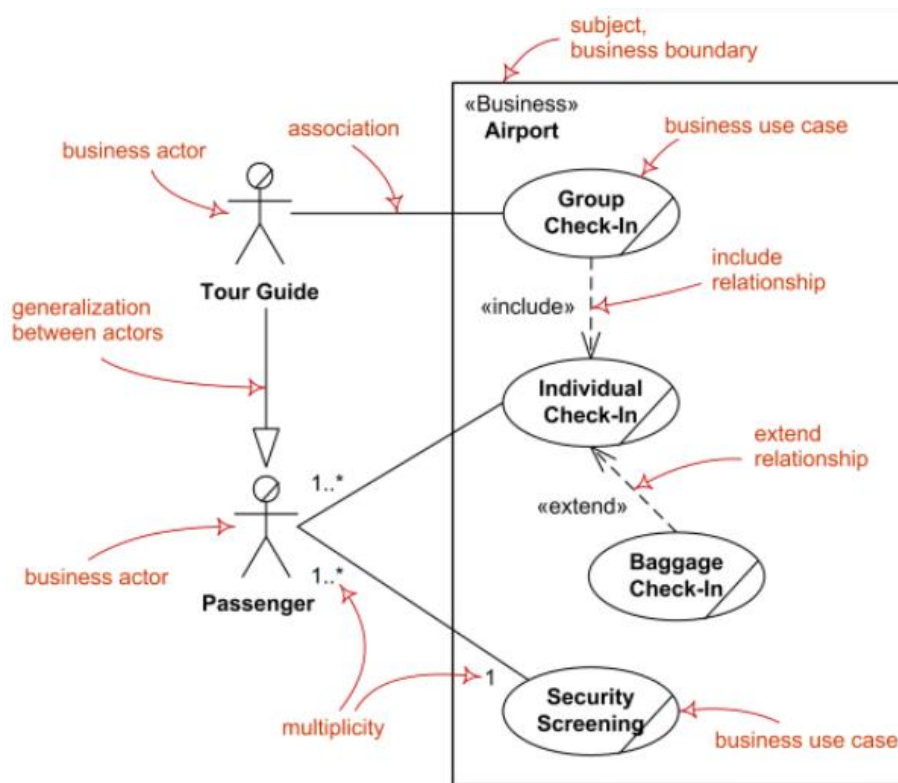
<sup>5</sup> Use Case Description

- در طول مرحله آزمایش، می‌توانید از نمودارهای مورد کاربرد برای شناسایی تست‌های سیستم، استفاده کنید.

در حالی که پشتیبانی از مدل سازی کسب و کار به عنوان یکی از اهداف UML اعلام شده است، مشخصات UML هیچ مجموعه نمادی را برای نیازهای کسب و کاری ارائه نمی دهد.

موارد کاربرد کسب و کار یا Business Use Case در RUP معرفی شدند تا عملکرد کسب و کاری<sup>۶</sup>، فرآیند یا فعالیت های<sup>۷</sup> موجود در کسب و کار مدل سازی شده را نشان دهند. یک کنشگر کسب و کاری، نقشی را نشان می دهد که توسط شخص یا سیستمی خارج از کسب و کار مدل شده، ایفا می کند و در تعامل با کسب و کار است. یک مورد کاربرد کسب و کار، باید نتیجه ای با ارزش و قابل مشاهده برای یک کنشگر کسب و کار ایجاد کند.

عناصر اصلی نمودار مورد کاربرد کسب و کار در تصویر زیر نشان داده شده است. مجدداً توجه داشته باشید که هم مورد کاربرد کسب و کار و هم کنشگر کسب و کار در استاندارد UML تعریف نشده اند، بنابراین شما یا باید از برخی ابزارهای UML برای پشتیبانی از آنها استفاده کنید یا نمادهای مدل سازی کسب و کار خود را ایجاد کنید.



عناصر اصلی نمودار مورد کاربرد کسب و کار شامل: کنشگر کسب و کار، مورد کاربرد کسب و کار،

مرزهای کسب و کار<sup>۸</sup>، ارتباطاتی از نوع شمولیت<sup>۹</sup> یا اشتراک و گسترش<sup>۱۰</sup> یا اجتماع.

<sup>6</sup> business function

<sup>7</sup> activity

<sup>8</sup> business boundary

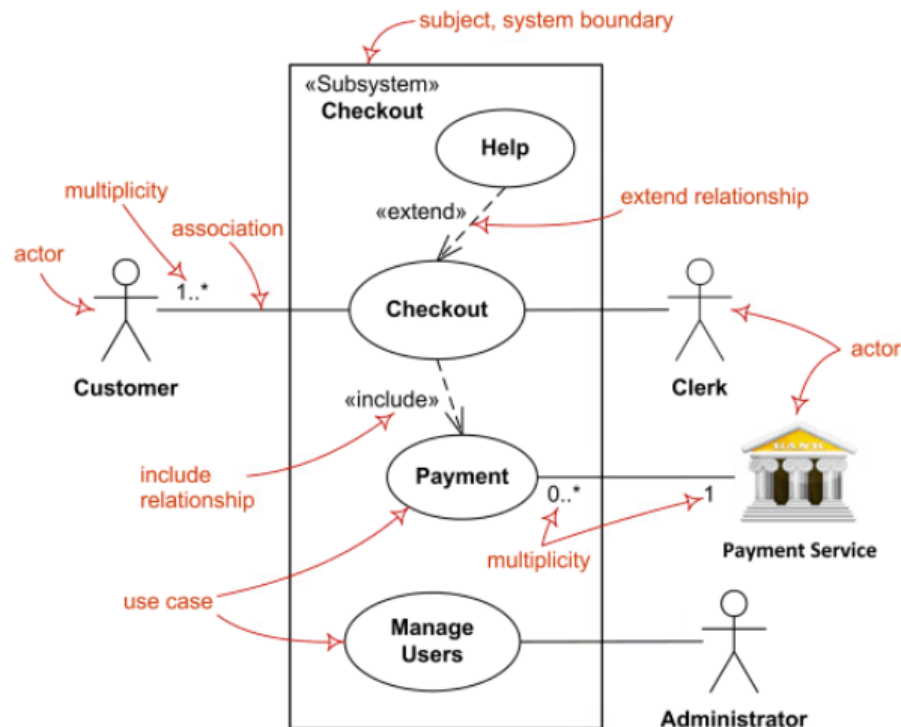
<sup>9</sup> include

<sup>10</sup> extend

از نمودارهای مورد کاربرد سیستم، برای تعیین موارد زیر استفاده می‌شود:

- الزامات<sup>11</sup> (خارجی)، کاربردهای مورد نیاز یک سیستم در حال طراحی یا تجزیه و تحلیل (موضوع) که آنچه سیستم قرار است انجام دهد را به تصویر می‌کشد.
- عملکرد ارائه شده توسط یک موضوع که سیستم می‌تواند انجام دهد.
- الزاماتی که موضوع مشخص شده در محیط خود ایجاد می‌کند تا با تعریف آنها مشخص کند محیط چگونه باید با سوژه تعامل داشته باشد تا بتواند خدمات خود را انجام دهد.

عناصر اصلی نمودار مورد کاربرد سیستم در استاندارد UML در تصویر زیر نشان داده شده است.



عناصر اصلی نمودار مورد کاربرد سیستم در استاندارد UML شامل: کنشگر، مورد کاربرد، موضوع،

ارتباطاتی از نوع شمولیت یا اشتراک و گسترش یا اجتماع.

<sup>11</sup> requirement

**Subject** یک مورد کاربرد، مرزهای یک کسب‌وکار، سیستم نرم‌افزاری، سیستم فیزیکی یا دستگاه، زیرسیستم، مؤلفه یا حتی کلاس منحصر به فرد را برای جمع‌بندی و تحلیل نیازمندی‌ها تعریف و نشان می‌دهد.

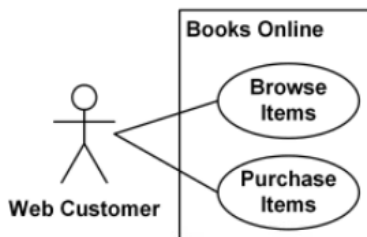
در اصطلاح UML، موضوع یا Subject، یک طبقه‌بندی کننده است که نقش "موضوع" را بازی می‌کند. به عبارت دیگر برای موضوع، کلاس ویژه جداگانه‌ای همانطور که برای کنشگر و مورد کاربرد ایجاد می‌شود، ایجاد نمی‌شود. از نظر UML 2.5 موضوع یک UseCase می‌تواند یک سیستم یا هر عنصر دیگری مانند یک جزء یا کلاس باشد که ممکن است رفتار داشته باشد. طبقه‌بندی کننده‌ای که ممکن است رفتارهای متعلق به خود داشته باشد، طبقه‌بندی کننده رفتاری نامیده می‌شود. با این حال، در نگارش انتزاعی UML 2.5، موضوع، به عنوان یک طبقه‌بندی اختیاری، گنجانده شده است.

موضوع یک طبقه‌بندی کننده (شامل زیرسیستم، جزء یا حتی کلاس) است که نشان‌دهنده یک کسب‌وکار، سیستم نرم‌افزاری، سیستم فیزیکی یا دستگاه تحت تحلیل، طراحی یا در نظر گرفته شده است که رفتار خاصی دارد و مجموعه‌ای از موارد کاربرد برای آن اعمال می‌شود.

موضوع، به موارد کاربرد قابل اجرا مربوط می‌شود و با موارد کاربرد متعلق به طبقه‌بندی کننده، متفاوت است. در UML 2.x تا 2.5، طبقه‌بندی کننده مورد کاربرد، طبقه‌بندی کننده‌ای بود که قابلیت مالکیت موارد کاربرد را داشت. در UML 2.5 هر طبقه‌بندی کننده ممکن است موارد کاربرد را داشته باشد.

#### نماد

موضوع (گاهی اوقات مرز سیستم نامیده می‌شود) توسط یک مستطیل که با یک نام موضوع، کلمات کلیدی مرتبط و کلیشه‌ها<sup>۱۲</sup> در گوشه بالا سمت چپ ارائه می‌شود. موارد کاربرد قابل استفاده برای موضوع در داخل مستطیل و کنشگران، خارج از مرز سیستم قرار دارند.



Books Online موضوعی است که با موارد کاربرد قابل اجرا و کنشگر Web Customer مرتبط است.

#### بازنگری‌ها

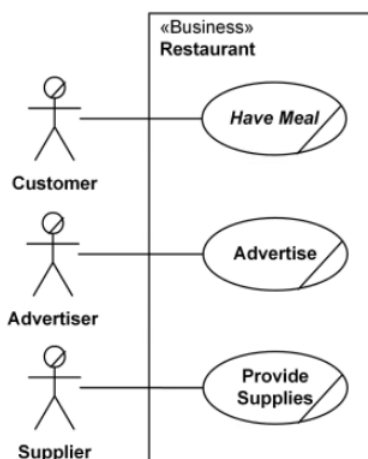
الزام قرار گرفتن نام موضوع در گوشه سمت چپ بالای مرز سیستم، فقط در UML 2.5 وجود دارد. نسخه‌های قبلی UML مجاز به استفاده از نام موضوع بودند و این نام در یکی از گوشه‌های سمت راست یا چپ بالا قرار می‌گرفت. شخصاً فکر می‌کنم استفاده از گوشه بالا سمت چپ بسیار محدود کننده است. وقتی اکثر کنشگران در سمت چپ هستند، نام موضوع در سمت راست بهتر به نظر می‌رسد. در برخی موارد من بدم نمی‌آید که نام موضوع حتی در وسط بالا باشد.



موارد کاربرد می‌تواند برای مدل‌سازی برخی از کسب‌وکارها برای تجزیه و تحلیل فرآیندهای کسب‌وکار، تشخیص مشکلات در حال تجربه، تعیین فرصت‌ها برای ارائه خدمات بهتر به مشتریان، استفاده شوند. موضوع موارد کاربرد، در این مورد، کسب‌وکار، شرکت، تقسیمات شرکتی، دپارتمان، تیم، می‌باشد. نمونه‌هایی از موضوعات کسب‌وکاری در زیر آورده شده است:

- دپارتمان فروشگاه
- فرودگاه
- رستوران

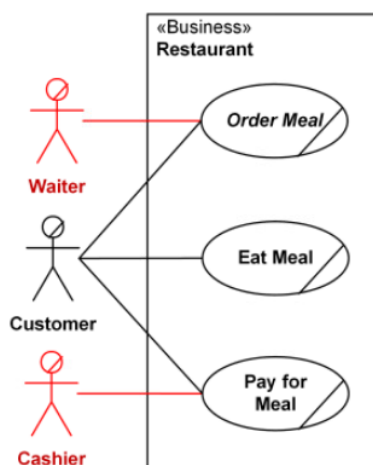
UML هیچ کلیشه استاندارد برای مدل‌سازی فرآیندهای کسب‌وکار ارائه نمی‌کند، اما در صورت نیاز برای وضوح، می‌توانیم از کلیشه‌های سفارشی مانند «Business» یا «Department» استفاده کنیم. مثال زیر «Business» Restaurant با کنشگران کسب‌وکار Customer، Advertiser و Supplier و موارد کاربرد مرتبط را نشان می‌دهد.



موضوع کسب‌وکار Restaurant با کنشگران کسب‌وکار و موارد کاربرد قابل اجرا

#### اشتباهات رایج

مثال زیر تصور غلط رایجی را برای «Business» Restaurant نشان می‌دهد که کنشگران کسب‌وکار به اشتباه شامل Cashier و Waiter هستند. اینها هم برای رستوران کار می‌کنند ولی بخشی از کسب‌وکار رستوران محسوب می‌شوند و اگر قرار بود که در این نمودار قرار گیرند باید در داخل محدوده موضوع قرار می‌گرفتند. آنها نباید به عنوان کنشگر نشان داده شوند زیرا کنشگران، کاربران خارجی کسب‌وکار مثل customers یا system هستند.



اشتباه: کسبوکار رستوران نباید گارسون و صندوقدار را به عنوان کنشگر داشته باشد

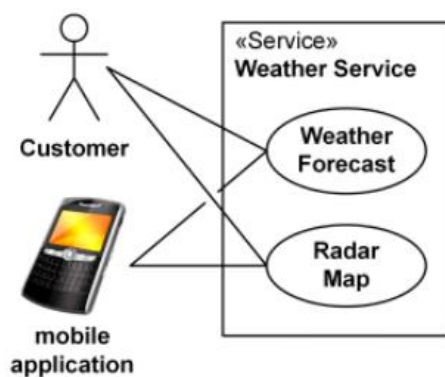
## Software System Subject

موارد کاربرد سیستم، سیستمی را توصیف می‌کند که برخی موارد یا فرآیندهای کاربرد/کاربردهای کسبوکار را خودکار می‌کند. موضوع در این مورد، نرم‌افزار و/یا سیستم سخت‌افزاری، زیرسیستم، جزء یا دستگاه است. نمونه‌هایی از سیستم‌ها:

- سایت اینترنتی
- سیستم پرداخت
- دستگاه بوجه خودکار (ATM)
- پایانه نقطه فروش یا همان ترمینال فروش (POS)

UML هیچ کلیشه استاندارد برای موضوع، ارائه نمی‌دهد اما نمونه‌های مشخصات UML 2.4 از کلیشه‌های اجزا<sup>۱۳</sup> استفاده می‌کنند:

- «Subsystem»
- «Process»
- «Service»
- «Component»



<sup>13</sup> component

موضوع Weather Service کلیشه‌ای به عنوان «Service» دارد

## Applicability of Use Cases

موارد کاربرد که به صورت بصری در داخل مرزهای سیستم قرار دارند، موارد کاربرد مناسب برای موضوع (subject) هستند (اما لزوماً موضوع نیستند).

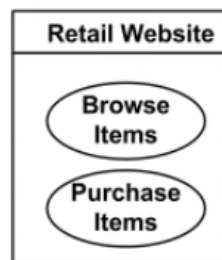


تفسیر این نمودار: موارد کاربرد Browse Items و Buy Items برای موضوع Retail Website قابل استفاده هستند

این امکان وجود دارد که برخی موارد کاربرد برای چندین موضوع قابل استفاده باشد.

## Ownership of Use Cases

در UML 2.4، موضوع می‌تواند شامل برخی یا همه موارد کاربرد قابل اجرا باشد. این مالکیت (تودرتو) یک مورد کاربرد با استفاده از نماد استاندارد برای طبقه‌بندی‌کننده تودرتو نشان داده می‌شود.



تفسیر این نمودار: موضوع Retail Website دارای موارد کاربرد Browse Items و Buy Items است

یک Actor یا کنشگر، طبقه‌بندی‌کننده رفتاری است که نقشی را مشخص می‌کند و این نقش توسط یک موجود خارجی که با موضوع تعامل دارد (مثلاً با مبادله سیگنال‌ها و داده‌ها) یا کاربر انسانی مرتبط با سیستم طراحی‌شده یا برخی سیستم‌ها یا سخت‌افزارهای دیگر که از خدمات مرتبط با موضوع استفاده می‌کنند، ارائه می‌شود.

اصطلاح نقش<sup>۱۴</sup> به‌طور غیررسمی بجای عبارت نوع، گروه یا گروه خاصی از کاربران که به خدمات خاصی از موضوع مدل‌سازی شده با موارد کاربرد مرتبط نیاز دارند، استفاده می‌شود. هنگامی که یک موجود خارجی با موضوع، تعامل دارد، نقش یک کنشگر خاص را ایفا می‌کند. آن موجودیت فیزیکی منفرد ممکن است چندین نقش مختلف را ارائه کند و یک نقش خاص ممکن است توسط یک یا چند نمونه مختلف ارائه شود.

از آنجایی که یک کنشگر، خارج از موضوع است، معمولاً در همان طبقه‌بندی‌کننده یا بسته‌ای<sup>۱۵</sup> تعریف می‌شود که موضوع را در بر می‌گیرد.

همه کنشگران باید مطابق با نقش فرضی خود، نامی داشته باشند. نمونه‌هایی از نام کنشگران (نقش کاربر<sup>۱۶</sup>) در زیر آورده شده است:

- مشتری (Customer)
- سرویس‌گیرنده وب (Web Client)
- دانشجو (Student)
- مسافر (Passenger)
- سیستم پرداخت (Payment System)

نماد استاندارد UML برای کنشگر، نماد "مرد چوبی"<sup>۱۷</sup> است که نام کنشگر در بالا یا پایین آن قرار دارد. نام کنشگران باید از دستورالعمل‌های حروف بزرگ و نقطه‌گذاری مانند کلاس‌ها، پیروی کند. نام کنشگران مفهومی باید به صورت مورب نشان داده شود.



Student

کنشگری به نام Student

نمادهای سفارشی که نوع کنشگر را نشان می‌دهند ممکن است برای نشان دادن یک منشگر خاص استفاده شوند، مانند استفاده از نماد(های) جداگانه برای کنشگران غیر انسانی.



Web Client

نماد سفارشی برای کنشگر Web Client

<sup>14</sup> role

<sup>15</sup> package

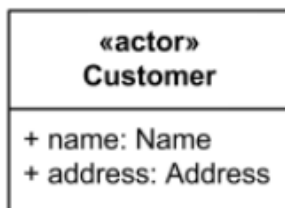
<sup>16</sup> user role

<sup>17</sup> Stick Man



نماد سفارشی برای کنشگر Bank

همچنین ممکن است یک کنشگر به شکل نماد مستطیل گونه مربوط به کلاس<sup>18</sup> با کلمه کلیدی استاندارد «actor» نشان داده شود که در صورت لزوم دارای نمادهای عمومی کلاس نیز خواهد بود.



کنشگر Customer در نمای کلاس

یک کنشگر فقط می‌تواند برای موارد (Use Case)، مؤلفه‌ها (Component)، و کلاس‌ها (Class)، انجمن‌های دودویی داشته باشد.

## Business Actor

یک کنشگر کسبوکار یا Business Actor (که در RUP برای حمایت از مدل‌سازی کسبوکار معرفی شد) نقشی را نشان می‌دهد که توسط شخص یا سیستمی خارج از کسبوکار مدل‌سازی شده و در تعامل با کسبوکار ارائه می‌شود. توجه داشته باشید که کنشگر کسبوکار در استاندارد UML تعریف نشده است. برخی از نمونه‌های معمولی از کنشگر کسبوکار عبارتند از:

- مشتری (Customer)
- تامین کننده (Supplier)
- حامی (Patron)
- مسافر (Passenger)
- قدرت (Authority)
- بانک (Bank)

هر کنشگر کسبوکار چیزی خارج از کسبوکار مدل شده را نشان می‌دهد و باید حداقل با یک مورد کاربرد کسبوکار (Business Use Case) درگیر باشد. کنشگر کسبوکار در RUP با نماد "مرد چوبی" با خطی که از سر آن عبور می‌کند نشان داده می‌شود.

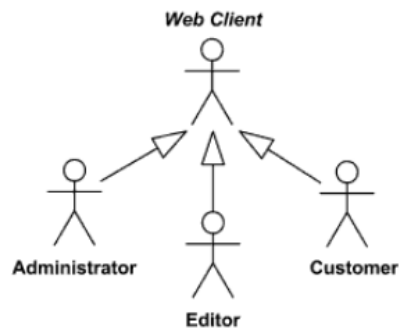


کنشگر کسبوکار به نام Passenger

<sup>18</sup> Class

### ارتباط بین کنشگرها

ما می‌توانیم کنشگران مفهومی یا واقعی را تعریف کرده و با استفاده از رابطه **generalization** آنها را خاص کنیم. رابطه **generalization** بین کنشگران به صورت یک خط مستقیم جهت‌دار با نوک پیکان بزرگ ارائه می‌شود (همانطور که برای رابطه **generalization** بین کلاس‌ها تعریف شده است).



کنشگر **Web Client** یک سوپر کلاس مفهومی برای **Administrator**، **Editor** و **Customer** است

موارد کاربرد یا Use Case ها اجازه می‌دهند تا نیازهای سیستم‌های در حال طراحی یا در مورد انتظار را شناسایی کنید، عملکرد ارائه شده توسط آن سیستم‌ها را توصیف کنید، و الزاماتی را که سیستم‌ها برای محیط خود ایجاد می‌کنند، تعیین کنید.

مشخصات UML، به عنوان مثال [UML 2.5 RTF - Beta 2] تعریفی کمی متفاوت از موارد کاربرد، ارائه می‌دهد. من تعریف زیر را از آن قطعات جمع‌آوری کردم.

مورد کاربرد نوعی طبقه‌بندی‌کننده رفتاری است که یک واحد [کامل] از عملکرد [مفید] را معرفی می‌کند. این واحد، توسط [یک یا چند] موضوع پوشش داده می‌شود که با همکاری یک یا چند کنشگر در ارتباط است، و [برای موارد کاربرد کامل] نتیجه قابل مشاهده‌ای را به همراه دارد که برای آن کنشگران [یا سایر ذینفعان] هر موضوع، ایجاد ارزش می‌کند.

مشخصات UML ایجاب می‌کند که "این عملکرد همیشه باید تکمیل شود تا Use Case مورد نظر کامل شود. به عبارت دیگر، اگر پس از اجرای آن، موضوع در حالتی باشد که در آن هیچ ورودی یا اقدام دیگری انتظار نمی‌رود و Use Case قابلیت شروع دوباره داشته باشد، یا در حالت خطا قرار گیرد، Use Case کامل تلقی می‌شود."

مشکل این الزام این است که اجتماع<sup>19</sup> موارد کاربرد را مطرح نمی‌کند و فقط اشتراک<sup>20</sup> موارد کاربرد را شامل می‌شود. اجتماع مورد کاربرد ممکن است لزوماً به خودی خود معنی نداشته باشد. مورد کاربرد شامل بخش مشترکی است که از یک مورد کاربرد جداگانه استخراج شده است. همچنین به نظر می‌رسد که الزام به ارائه یک نتیجه قابل مشاهده، غیر ممکن است.

همچنین این تردید وجود دارد که عملکرد مورد کاربرد، همیشه مفید باشد: "هر Use Case، واحدی از عملکرد مفید را مشخص می‌کند که موضوع مورد نظر در اختیار کاربران خود قرار می‌دهد."

"یک UseCase ممکن است برای هر موضوعات مختلفی اعمال شود." باوجود اینکه، توصیف کلاس Use Case در مشخصات UML اجازه می‌دهد تا Use Case هیچ موضوع مرتبطی نداشته باشد؛ منطقی است که انتظار داشته باشیم برای هر مورد کاربرد حداقل یک موضوع، داشته باشیم، در غیر این صورت تعریف مورد کاربرد معنایی نخواهد داشت.

در حالی که "مفاهیم کلیدی مشخص شده در این بخش عبارتند از کنشگران یا Actor ها، موارد کاربرد یا UseCase ها و موضوعات یا Subject ها" use case ها بعضی مواقع "سایر ذینفعان موضوع" نیز معرفی می‌شوند. وجود کنشگران به عنوان ذینفعان موضوع، کافی است و اگر این «ذینفعان»<sup>21</sup> (use case ها) چیز دیگری برای اضافه کردن دارند، باید به عنوان یک مفهوم جداگانه در مشخصات UML گنجانده شوند.

مشخصات UML تا زمانی که UML 2.5 منتشر نشده بود، مستلزم این بود که عملکرد مورد کاربرد توسط یک کنشگر آغاز شود. در UML 2.5 این محدودیت، حذف شد، به این معنی که ممکن است شرایطی وجود داشته باشد که عملکرد سیستم، توسط خود سیستم، شروع شود و در عین حال نتایج مفیدی را برای یک کنشگر، ارائه دهد. به عنوان مثال، سیستم می‌تواند به مشتری از ارسال سفارش او اطلاع دهد، پاکسازی و بایگانی اطلاعات کاربر را برنامه‌ریزی کند، اطلاعاتی را از سیستم دیگری درخواست کند و غیره.

همچنین، در UML 2.x تا UML 2.5، بیان شده که موارد کاربرد "بر اساس نیازهای کنشگران تعریف می‌شوند." که البته این جمله از UML 2.5 حذف شد زیرا برخی از کنشگران ممکن است به خودی خود نه نیازی داشته باشند و نه الزاماتی.

<sup>19</sup> Extending  
<sup>20</sup> Including  
<sup>21</sup> stakeholder

مشخصات UML هیچ دستورالعملی در مورد روش نامگذاری موارد کاربرد، ارائه نمی‌دهد. تنها شرط این است که هر مورد کاربرد باید یک نام داشته باشد. ما فقط باید از تعریف مورد کاربرد، پیروی کنیم تا برای واحد عملکردی که توسط یک سیستم انجام می‌شود، نامی بگذاریم که نتایج قابل مشاهده و مفیدی را برای یک کنشگر ارائه دهد. نمونه‌هایی از نامگذاری موارد کاربرد در زیر آورده شده است:

- خرید کردن (Make Purchase)
- ثبت سفارش (Place Order)
- به روز رسانی مبلغ آبونمان (Update Subscription)
- انتقال وجه (Transfer Funds)
- استخدام کارمند (Hire Employee)
- مدیریت حساب (Manage Account)

## نماد Use Case

Use case معمولاً به شکل بیضی که حاوی نام مورد کاربرد است، نشان داده می‌شود.



مورد کاربردی به نام User Registration

نام مورد کاربرد همچنین می‌تواند در زیر بیضی نیز قرار گیرد.



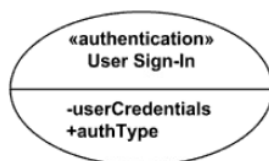
مورد کاربردی به نام User Registration

اگر یک موضوع (یا مرز سیستم) نمایش داده شود، بیضی مورد استفاده به صورت بصری در داخل این مستطیل یا مرز سیستم قرار می‌گیرد. توجه داشته باشید که این لزوماً به این معنی نیست که موضوع مورد نظر، دارای موارد کاربرد مشمول شده است، بلکه صرفاً به این معنی است که مورد کاربرد در آن موضوع نیز اعمال می‌شود.

موارد کاربرد هیچ کلمه کلیدی استاندارد یا کلیشه‌ای<sup>۲۲</sup> ندارند. موارد کاربرد را می‌توان با یک کلیشه سفارشی بالای نام نشان داد. این کلیشه سفارشی در قالب نوع یا طبقه‌بندی خاص، یک مشخصه use case را معرفی می‌کنند. فهرستی از مشخصات مورد کاربرد شامل عملیات<sup>۲۳</sup> و فیلدها<sup>۲۴</sup> را می‌توان در محدوده زیر نام مورد کاربرد در نماد بیضی شکل مورد کاربرد، نشان داد.

stereotype<sup>22</sup>  
operations<sup>23</sup>  
attributes<sup>24</sup>





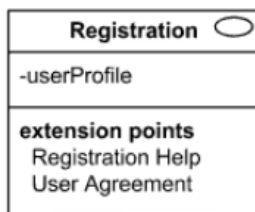
استفاده از کلیشه ای به عنوان «authentication» برای مورد کاربرد User Sign-In

موارد کاربرد دارای نقاط توسعه<sup>۲۵</sup>، ممکن است در محدوده مورد کاربرد تحت کلمه کلیدی extension points، فهرست شوند.



مورد کاربردی با نام Registration که دارای نقاط توسعه Registration Help و User Agreement است

یک مورد کاربرد همچنین می‌تواند با استفاده از نماد مستطیل استاندارد طبقه‌بندی کننده‌ها به همراه یک نماد بیضی در گوشه سمت راست بالای مستطیل و با محفظه‌های لیست جداگانه اختیاری برای مشخصات آن نشان داده شود.



مورد کاربردی به نام Registration با استفاده از نماد مستطیل استاندارد برای طبقه‌بندی کننده‌ها نشان داده شده است

## Abstract Use Case

تمام مشخصات UML 2.x و حتی UML 2.5 موارد کاربرد مفهومی یا Abstract Use Case را ذکر، تعریف یا توضیح نمی‌دهند. مشخصات UML 1.x اشاره کرده است که "نام یک مورد کاربرد مفهومی، ممکن است به صورت مورب نشان داده شود" اما در UML 2.0 این جمله بدون هیچ توضیحی از مشخصات UML حذف شد.

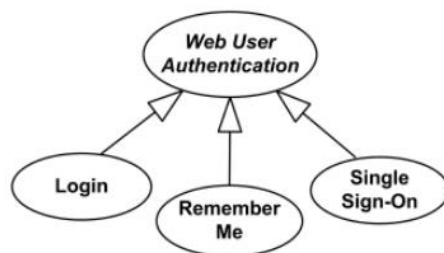
یکی از دلایل حذف این جمله می‌تواند این باشد که چون use case یک طبقه‌بندی کننده است و هر طبقه‌بندی کننده‌ای می‌تواند، مفهومی باشد (با نام به شکل مورب)، بدیهی است که باید برای موارد کاربرد نیز قابل اجرا باشد.

از طرف دیگر، چون جمله حذف شد و UML 2.5 اصلاً موارد کاربرد مفهومی را ذکر نمی‌کند و حتی یک نمونه از موارد کاربرد مفهومی را ارائه نمی‌دهد، می‌تواند به این معنی باشد که آنها انتظار دارند همه موارد کاربرد واقعی باشند نه مفهومی. البته بهتر بود که در این مورد، این وضعیت به صراحت در مشخصات UML توضیح داده می‌شد.

با فرض اینکه مورد کاربرد می‌تواند مفهومی باشد و تعریف مناسب برای طبقه‌بندی کننده اعمال شود، مورد کاربرد مفهومی، مورد کاربردی است که اعلان کامل و شفاف ندارد (ناقص است) و (معمولاً همانطور که مشخصات UML می‌گوید) قابل نمونه‌سازی نیست. یک مورد کاربرد، زمانی

<sup>25</sup> extension points

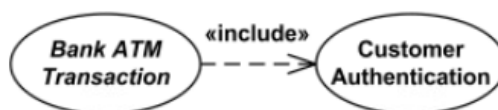
مفهومی در نظر گرفته می‌شود که توسط سایر موارد کاربرد، استفاده شود، به عنوان مثال، این نوع مورد کاربرد به هدف ایجاد یک رابطه generalization استفاده می‌شود. من امیدوارم، نام مورد کاربرد مفهومی که به صورت مورب نشان داده می‌شود، همچنان در UML 2.5 قابل استفاده باشد، همانطور که در UML 1.x مشخص شده است.



مورد کاربرد Web User Authentication یک مورد کاربرد مفهومی است

که توسط موارد کاربرد Login، Remember Me و Single Sign-On به صورت صریح و خاص تعریف شده است.

وقتی مشخصات UML 2.4 رابطه اشتراک بین موارد کاربرد را توضیح می‌دهند که "آنچه در یک مورد کاربرد پایه باقی می‌ماند معمولاً کامل نیست" اما به دلایلی از نامگذاری آن به عنوان یک مورد کاربرد مفهومی، اجتناب می‌کنند. به طور کلی، به این معنی است که موارد کاربرد اشتراکی همیشه مفهومی هستند.



اگر مورد کاربرد Bank ATM Transaction یک مورد کاربرد مفهومی باشد

به این معنی است که با موارد کاربرد Customer Authentication اشتراکاتی دارد

منابعی که اشتراکات موارد کاربرد را به شکل ارتباط بین موارد کاربرد مفهومی و موارد کاربرد واقعی تعریف می‌کنند، بسیار رایج است، اگرچه مشخصات UML از انجام آن اجتناب می‌کند. در حالی که ممکن است فرض کنیم که اشتراکات موارد کاربرد، همیشه مفهومی هستند، مورد کاربرد مشترک شده، احتمالاً می‌تواند مفهومی یا واقعی باشد. به طرز شگفت انگیزی، برخی منابع وجود دارد که من نمی‌توانم با آنها موافق باشم و توضیحی دقیقاً مخالف این حالت را ارائه می‌دهند. این منابع، شامل موارد کاربرد (پایه) "معمولاً واقعی" و موارد کاربرد مشترک شده ("افزودن") "معمولاً مفهومی" هستند.

خیلی پیچیده‌تر می‌شود وقتی که، منابع دیگر، موارد کاربرد مفهومی را به عنوان موارد کاربرد توصیفی<sup>۲۶</sup> در سطح مفهومی (موردهای کاربرد کسب‌وکاری، که گاهی اوقات موارد کاربرد واقعی نیز نامیده می‌شوند) را برخلاف موارد کاربرد سیستمی<sup>۲۷</sup>، تعریف می‌کنند.

## Business Use Case

اگرچه پشتیبانی از مدل‌سازی کسب‌وکار به عنوان هدف UML اعلام شده است، مشخصات UML هیچ مفهومی برای موارد کاربرد کسب‌وکار یا Business Use Case ارائه نمی‌دهد.

<sup>26</sup> describe use cases  
<sup>27</sup> system use cases

موارد کاربرد کسب‌وکار در RUP برای پشتیبانی از مدل‌سازی کسب‌وکار برای نشان دادن عملکرد، فرآیند یا فعالیت کسب‌وکار انجام شده در کسب‌وکار مدل‌شده، معرفی شدند. مورد کاربرد کسب‌وکار باید نتیجه‌ای با ارزش قابل مشاهده برای یک کنشگر کسب‌وکار ایجاد کند.

یک مورد کاربرد کسب‌وکار، تعریف می‌کند که وقتی مورد کاربرد توسط کنشگر کسب‌وکار درخواست می‌شود، چه اتفاقی در کسب‌وکار می‌افتد، گردش کار کامل یا فرآیند کسب‌وکار چگونه است و چگونه نتایج مورد نیاز کنشگر کسب‌وکار را تولید می‌کند. مورد کاربرد کسب‌وکار در RUP با استفاده از یک شکل بیضی و خطی که از آن عبور می‌کند مطابق شکل زیر نشان داده شده است.



مورد کاربرد کسب‌وکار به نام Individual Check-In

دو رویکرد برای نام‌گذاری موارد کاربرد کسب‌وکار وجود دارد. مورد کاربرد را می‌توان از دیدگاه کنشگر کسب‌وکار که در این حالت این نام، بیان‌کننده هدف یا نیاز کنشگر است نامگذاری کرد یا از نقطه نظر خود کسب‌وکار با نامگذاری فرآیندهای کسب‌وکار یا خدمات ارائه شده به کنشگران کسب‌وکار، نامگذاری کرد.



مورد کاربرد کسب‌وکار Apply for Job که نشان می‌دهد کنشگر Candidate شغل را تأیید می‌کند

مورد کاربرد کسب‌وکار Apply for Job هدف کنشگر کسب‌وکار Candidate را بیان می‌کند. نام جایگزین از نظر کسب‌وکار، Hire Staff می‌تواند باشد.



مورد کاربرد کسب‌وکار Serves Meal که کسب‌وکار ارائه غذا به کنشگر Customer را نشان می‌دهد

در این مورد، مورد کاربرد کسب‌وکار با توجه به فرآیند یا سرویس کسب‌وکاری Business Serves Meal to Customer نامگذاری می‌شود. نام جایگزین از نظر کنشگر می‌تواند Have Meal باشد.

با مقایسه نمونه‌هایی از نمودارهای مورد کاربرد کسب‌وکار برای رستوران می‌توانید تفاوت‌های دیگر بین این دو رویکرد را مشاهده کنید.

## Describing Use Case Behaviors

رفتارهای مورد کاربرد ممکن است در یک متن با زبان طبیعی (رفتار غیرشفاف)، که رویه رایج کنونی است، یا با استفاده از نمودارهای رفتاری UML برای رفتارهای خاص مانند موارد زیر توصیف شوند:

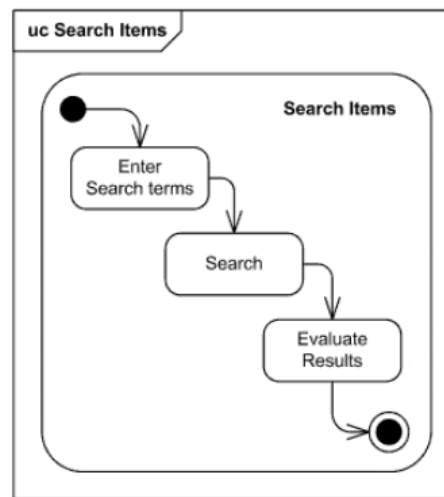
- activity
- state machine

## • interaction

رفتار یک مورد کاربرد همچنین ممکن است به طور غیرمستقیم از طریق یک **collaboration**، توصیف شود که از مورد کاربرد و کنشگران آن به عنوان طبقه‌بندی‌کننده‌هایی که قطعات آن را دسته‌بندی می‌کنند، استفاده می‌کند.

اینکه استفاده از کدام یک از این تکنیک‌ها مفید است، به ماهیت رفتار مورد کاربرد و همچنین به خواننده مورد نظر بستگی دارد. این توصیفات را می‌توان با هم نیز ترکیب کرد.

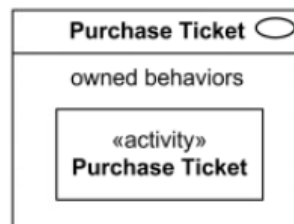
ابزارهای UML باید امکان پیوند دادن رفتارها را به مورد کاربرد توصیف شده فراهم کنند. Use Case را می‌توان در یک قاب با برچسب **use** یا **case** (شکل اختصاری) ارائه کرد. ناحیه محتوای قاب مورد نظر را می‌توان با انواع مختلف نمودارهای UML نشان داد که رفتار مورد کاربرد مورد نظر را توصیف می‌کند.



مورد کاربردی **Search Items** که به شکل قاب ارائه شده است

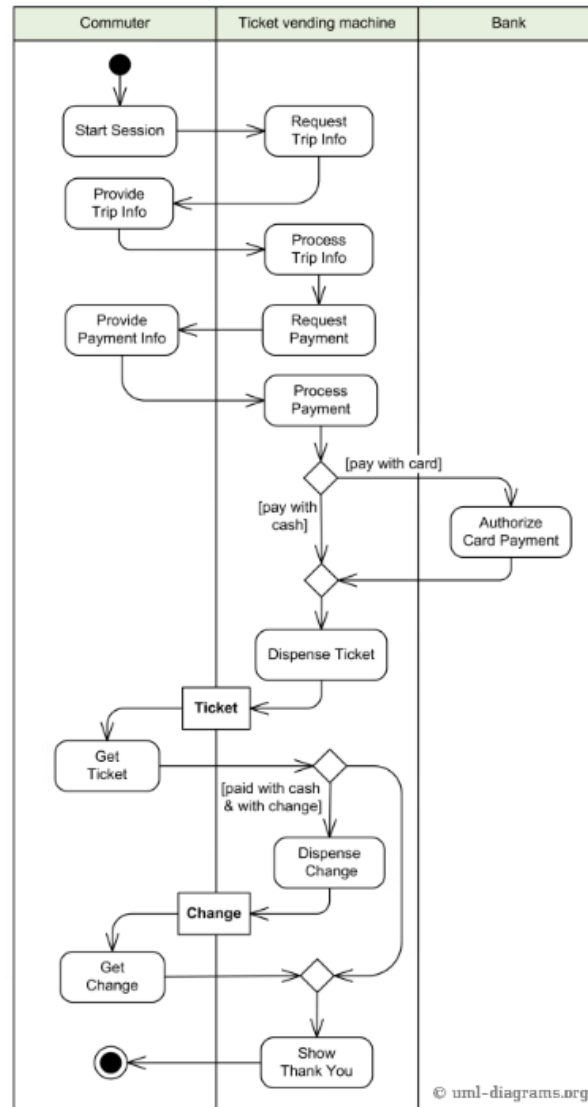
activity diagram مربوط به **Search Items** در آن گنجانده شده است

نمونه دیگری از چنین اتصال یک مورد کاربرد، رفتار نشان داده شده توسط فعالیت‌ها است که در زیر با استفاده از نماد UML 2.5 نشان داده شده است.



مورد کاربرد **Purchase Ticket** مالکیت رفتاری را نشان می‌دهد که توسط فعالیت **Purchase Ticket** مشخص شده است

مثال activity diagram مربوط به **Purchase Ticket** در زیر رفتار مورد کاربرد **Purchase Ticket** را شرح می‌دهد.



نمونه‌ای از رفتار مورد کاربرد Purchase Ticket که با استفاده از activity diagram شرح داده شده است.

اجرای یک مورد کاربرد یک رویداد رفتار بلادرنگ است. هر نمونه از طبقه‌بندی‌کننده‌ای که متوجه یک مورد کاربرد می‌شود، باید به روشی که توسط مورد کاربرد مورد نظر توصیف شده است، رفتار کند.

## Relationships Between Use Cases

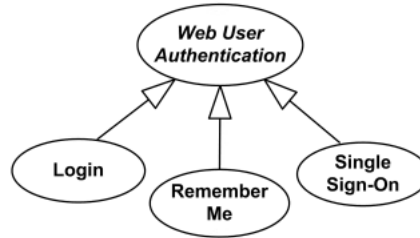
موارد کاربرد را می‌توان با استفاده از روابط زیر سازماندهی کرد:

- generalization
- association
- extend
- include

### رابطه Generalization بین موارد کاربرد

رابطه Generalization بین موارد کاربرد، شبیه به این رابطه بین کلاس‌ها است. مورد کاربرد فرعی، ویژگی‌ها و رفتار سوپر مورد کاربرد را به ارث می‌برد و ممکن است رفتار سوپر مورد کاربرد مورد نظر را بازنویسی<sup>۲۸</sup> کند.

رابطه Generalization به صورت یک خط مستقیم جهت‌دار با یک سر پیکان مثلث توخالی بزرگ نشان داده می‌شود، مانند Generalization بین طبقه‌بندی کننده‌ها، که از موارد کاربرد خاص به موارد کاربرد عمومی هدایت می‌شود.



مورد کاربرد Web User Authentication یک مورد کاربرد مفهومی است

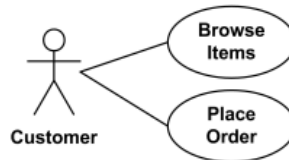
و توسط موارد کاربرد Login، Remember Me و Single Sign-On تعریف می‌شود.

### رابطه Association بین موارد کاربرد

موارد کاربرد فقط می‌توانند در رابطه Association باینری استفاده کنند. دو مورد کاربرد که موضوع یکسانی را مشخص می‌کنند را نمی‌توان با یک رابطه Association مرتبط کرد زیرا هر یک به طور جداگانه استفاده کامل از سیستم را توصیف می‌کند.

### رابطه Association بین مورد کاربرد و کنشگر

هر مورد کاربرد، یک عملکرد مفید منحصر به فرد را نشان می‌دهد که موضوعات در اختیار کنشگران قرار می‌دهند. ارتباط بین یک کنشگر و یک مورد کاربرد، نشان می‌دهد که کنشگر و مورد کاربرد به نحوی با یکدیگر تعامل یا ارتباط دارند. فقط ارتباط باینری بین کنشگران و موارد کاربرد مجاز است. یک کنشگر می‌تواند به یک یا چند مورد کاربرد مرتبط باشد.



کنشگر Customer با دو مورد کاربرد Browse Items و Place Order مرتبط است

یک مورد کاربرد ممکن است یک یا چند کنشگر مرتبط، داشته باشد.



مورد کاربرد Manage Account با کنشگران Customer و Bank مرتبط است

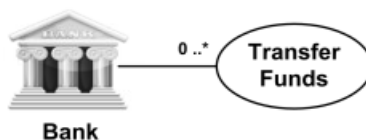
اگر چندین کنشگر مرتبط با یک مورد کاربرد وجود داشته باشد، ممکن است از نمودار مورد کاربرد، مشخص نباشد که کدام کنشگر شروع کننده مورد کاربرد است، به عنوان مثال، معلوم نیست که کلیشه "primary actor" به کدام می‌تواند تعلق داشته باشد. (در UML غیر استاندارد، کنشگران اصلی، کسانی هستند که از خدمات سیستم استفاده می‌کنند و کنشگران پشتیبان، کنشگرانی هستند که خدمات را به سیستم ارائه می‌دهند).

#### تعداد ارتباط مجاز برای رابطه Association

UML اجازه استفاده از چندگانگی<sup>29</sup> را در یک یا هر دو انتهای ارتباط بین کنشگران و مورد کاربرد می‌دهد.

#### چندگانگی در سمت مورد کاربرد

هنگامی که یک کنشگر، یک ارتباط چندگانه با یک مورد کاربرد دارد و این مورد کاربرد است که چند ارتباط با یک کنشگر دارد، به این معنی است که یک کنشگر منفرد می‌تواند در موارد کاربرد چندگانه از همان مورد کاربرد دخالت داشته باشد.



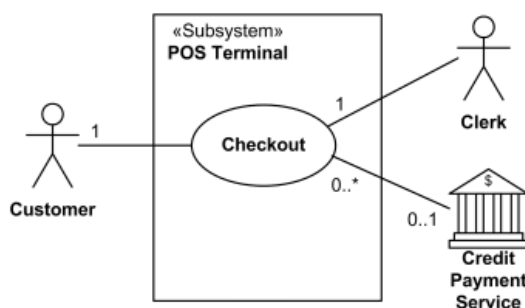
کنشگر Bank در چندین مورد کاربرد از Transfer Funds (تبادلات صندوق‌ها) شرکت دارد

مشخصات UML 2.x تا UML 2.5 هرگونه تفسیر معقولی از این نوع تعامل یک کنشگر معین با موارد کاربرد متعدد را نادیده می‌گیرد و عمداً آن را تعریف نشده نگه می‌دارد. به عنوان مثال، تعدد موارد کاربرد می‌تواند به این معنی باشد که یک کنشگر با موارد کاربرد چندگانه به شکل یا دلایل زیر تعامل داشته باشد:

- به صورت موازی (concurrently)
- در مقاطع مختلف زمانی (overlapping)
- متقابل انحصاری (random, sequentially و غیره)

#### چندگانگی در سمت کنشگر

تعداد کنشگر مورد نیاز ممکن است به صراحت با استفاده از عدد 1 یا بیشتر نشان داده شود. UML 2.5 همچنین اجازه می‌دهد تا کنشگر اختیاری<sup>30</sup> باشد. عدد 1..0 کنشگر به این معنی است که کنشگر ممکن است در هر یک از موارد کاربرد مرتبط با آنها شرکت کند (1) یا نکند (0).



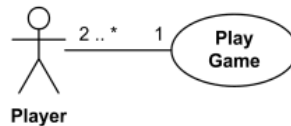
<sup>29</sup> multiplicity  
<sup>30</sup> optional

مورد کاربرد Checkout به 1 عدد کنشگر Customer نیاز دارد.

مورد کاربرد مورد نظر ممکن است به خدمات Credit Payment نیاز نداشته باشد (مثلاً اگر پرداخت به صورت نقدی باشد).

بنابراین ارتباط بین این مورد کاربرد با این خدمت در هر سمت می‌تواند عدد 0 را نیز داشته باشد.

هنگامی که یک مورد کاربرد با یک کنشگر یک ارتباط چندگانه دارد که در انتهای کنشگر بیشتر از یک است، به این معنی است که بیش از یک نمونه کنشگر در مورد کاربرد دخیل هستند.



دو یا چند کنشگر Player در مورد کاربرد Play Game دخیل هستند.

هر کنشگر در یک بازی Play شرکت می‌کند.

مشخصات UML 2.x تا UML 2.5 اجازه هرگونه تفسیر منطقی از این نوع تعامل چندین کنشگر با یک مورد کاربرد را می‌دهد و عمداً آن را تعریف نشده نگه می‌دارد.

به عنوان مثال، تعدد کنشگر می‌تواند به این معنی باشد که تعامل یک مورد کاربرد خاص با چندین نمونه کنشگر مجزا ممکن است به صورت زیر باشد:

- تعامل همزمان (concurrent)
- تعامل همپوشانی (overlapping)، در مقاطع مختلف زمانی
- تعامل متقابل منحصر به فرد (random, sequential) و غیره)

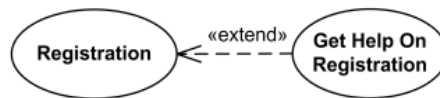


ارتباط Extend یک رابطه جهت‌دار است که مشخص می‌کند چگونه و چه زمانی رفتار تعریف‌شده در موارد کاربرد توسعه‌دهنده مکمل (اختیاری<sup>۳۱</sup>) را می‌توان در رفتار تعریف‌شده در مورد کاربرد توسعه‌یافته درج کرد.

مورد کاربرد توسعه‌یافته به خودی خود، مستقل از ارتباط Extend مورد کاربرد، معنادار است. مورد کاربرد توسعه یافته معمولاً رفتار اختیاری را تعریف می‌کند که لزوماً به خودی خود معنی ندارد. رابطه Extend متعلق به مورد کاربرد توسعه‌یافته است. یک مورد کاربرد توسعه‌دهنده می‌تواند بیش از یک مورد کاربرد را توسعه دهد، و یک مورد کاربرد توسعه‌دهنده ممکن است فقط یک مورد کاربرد توسعه‌یافته را توسعه دهد.

فضای توسعه در یک یا چند نقطه توسعه<sup>۳۲</sup> تعریف شده در مورد کاربرد توسعه‌یافته، صورت می‌گیرد.

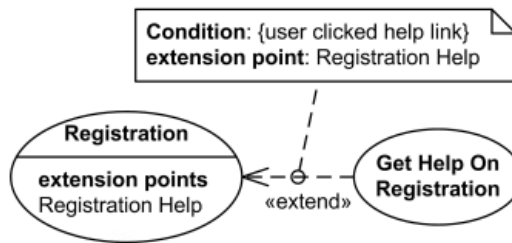
رابطه Extend به صورت یک خط چین با یک پیکان باز که از مورد کاربرد توسعه‌دهنده به مورد کاربرد توسعه‌یافته (پایه) هدایت می‌شود نشان داده می‌شود. فلش با کلمه کلیدی «extend» برچسب‌گذاری شده است.



مورد کاربرد Registration به تنهایی کامل و معنادار است.

می‌توان آن را با موارد کاربرد اختیاری Get Help On Registration توسعه داد.

اگر تمایل داشته باشید، می‌توانید شرایط رابطه Extend و همچنین ارجاع به نقاط توسعه در یک یادداشت پیوست شده به رابطه Extend مربوطه نشان دهید.



مورد کاربرد Registration به صورت مشروط توسط Get Help On Registration Use case در نقطه توسعه Registration Help توسعه داده می‌شود.

## Extension Point

نقطه توسعه یا Extension Point یک ویژگی یک مورد کاربرد است که نقطه‌ای را در رفتار مورد کاربرد معرفی می‌کند (ارجاع می‌دهد) که در آن نقطه، آن رفتار می‌تواند توسط یک مورد کاربرد دیگر (توسعه دهنده) بر اساس رابطه Extend مشخص شده است، توسعه یابد.

نقاط توسعه ممکن است در محلی در زیر نماد بیضی شکل مورد کاربرد، که با عنوان extension point نشان داده شده‌است، نوشته شوند. هر نقطه توسعه باید یک نام منحصر به فرد برای یک مورد کاربرد، داشته باشد. نقاط توسعه به صورت متنی مانند نگارش نشان داده شده در زیر می‌باشند:

<sup>31</sup> optional  
<sup>32</sup> extension point

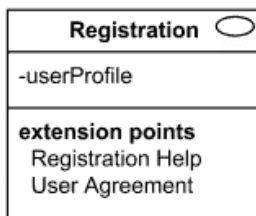
extension point ::= name [ : explanation ]

explanation که اختیاری است، شرحی است که معمولاً به عنوان متن غیررسمی ارائه می‌شود. این متن می‌تواند به اشکال دیگر باشد، مانند نام یک وضعیت (State) در یک ماشین وضعیت، یک فعالیت (Activity) در نمودار فعالیت، یا پیش‌شرط (precondition) یا پس‌شرط (precondition).



مورد کاربرد Registration با نقاط توسعه Registration Help و User Agreement

نقاط توسعه ممکن است در محدوده مستطیل مورد کاربرد با نماد بیضی، در زیر عنوان extension points نشان داده شوند.

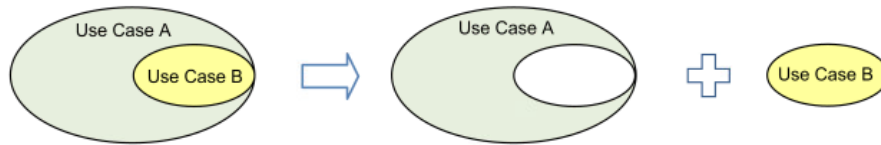


نقاط توسعه مورد کاربرد Registration که با استفاده از نماد مستطیل نشان داده شده است

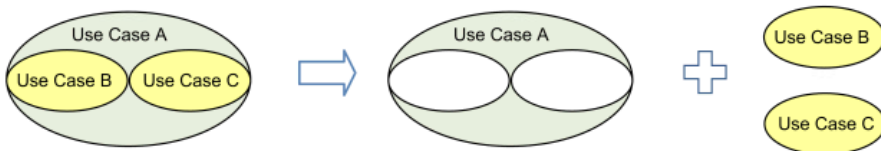
ارتباط Include یک رابطه مستقیم بین دو مورد کاربرد است که برای نشان دادن رفتار مورد کاربرد مشمول شده (اضافه) در رفتار مورد کاربرد مشمول کننده (پایه) درج شده است. رابطه Include می‌تواند موارد زیر استفاده شود:

- برای ساده کردن موارد کاربرد بزرگ با تقسیم آن به چندین مورد کاربرد
- برای استخراج بخش‌های مشترک از رفتارهای دو یا چند مورد کاربرد

یک مورد کاربرد بزرگ می‌تواند رفتارهایی داشته باشد که ممکن است به موارد کاربرد کوچک‌تر مجزا، جدا شوند تا با استفاده از رابطه Include در مورد کاربرد پایه قرار گیرند. هدف از این اقدام، مدولار کردن<sup>33</sup> رفتارها و مدیریت پذیرتر کردن آنهاست.

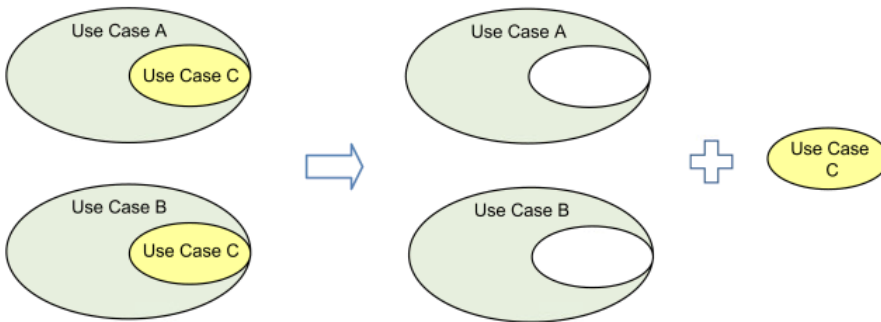


مورد کاربرد B از مورد کاربرد بزرگتر A به صورت یک مورد کاربرد جداگانه، استخراج می‌شود



موارد کاربرد B و C از مورد کاربرد بزرگتر A به صورت موارد کاربرد جداگانه، استخراج می‌شوند

هنگامی که دو یا چند مورد کاربرد، دارای رفتار مشترک هستند، این بخش مشترک می‌تواند در یک مورد کاربرد جداگانه استخراج شود تا به هر مورد کاربرد به واسطه رابطه Include اضافه شود.



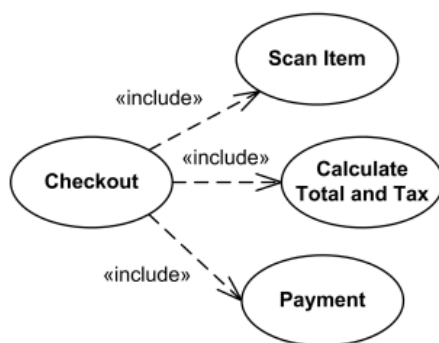
مورد کاربرد C از موارد کاربرد A و B استخراج می‌شود تا توسط هر دو مورد کاربرد، مجدد با استفاده از رابطه Include استفاده شود

اجرای مورد کاربرد مشمول شده، مشابه یک فراخوانی زیر روال یا دستور ماکرو در برنامه‌نویسی است. همه رفتارهای مورد کاربرد مشمول شده در یک مکان واحد در مورد کاربرد مشمول کننده قبل از ازسرگیری اجرای مورد کاربرد مشمول کننده، اجرا می‌شود.

توجه داشته باشید، در حالی که UML 2.x نقاط توسعه را برای رابطه Extend تعریف می‌کند، هیچ "نقاط دربرگیری"<sup>۳۴</sup> برای تعیین مکان یا شرایط شمولیت در یک رابطه Include وجود ندارد.

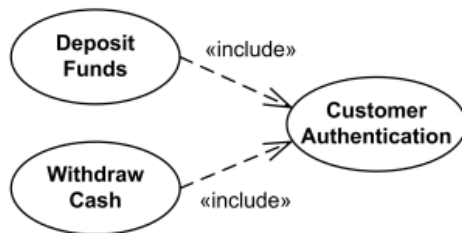
موارد کاربرد مشمول کننده بستگی به افزودن موارد کاربرد مشمول شده آن دارد، که آیا الزامی<sup>۳۵</sup> یا اختیاری<sup>۳۶</sup> است. به این معنی که موارد کاربرد مشمول کننده به خودی خود کامل نیست، و بنابراین منطقی است که موارد کاربرد مشمول کننده را موارد کاربردی مفهومی بنامیم. هیچ یک از مشخصات UML 2.x تا UML 2.5 حتی موارد کاربرد مفهومی را توضیح نداده‌اند. برخی از منابع UML، موارد کاربرد مفهومی را موارد کاربردی مشمول کننده، تعریف می‌کنند، در حالی که در واقع باید برعکس باشد: موارد کاربرد مشمول کننده زیرمجموعه موارد کاربرد انتزاعی باشند. به بحث در مورد تعریف موارد کاربرد مفهومی مراجعه کنید.

رابطه Include بین موارد کاربرد با یک فلش چین‌دار با سر پیکان باز از مورد کاربرد مشمول کننده (پایه) تا مورد کاربرد مشمول شده (بخش مشترک) نشان داده می‌شود. فلش با کلمه کلیدی «include» برچسب‌گذاری می‌شود.



مورد کاربرد Checkout شامل چندین مورد کاربرد است یا می‌توان گفت که این مورد کاربرد، مشمول کننده چندین مورد کاربرد است؛ موارد کاربرد مشمول شده Scan Item، Calculate Total and Tax، و Payment این موارد کاربرد هستند

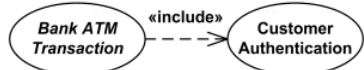
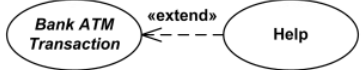

مورد کاربرد بزرگ و پیچیده Checkout دارای چندین مورد کاربرد استخراج شده است که هر مورد کوچکتر، یک واحد منطقی رفتار را توصیف می‌کند. توجه داشته باشید که مورد کاربرد مشمول کننده Checkout به خودی خود ناقص است و برای تکمیل به موارد کاربرد مشمول شده توصیف شده نیاز دارد.



موارد کاربرد Deposit Funds و Withdraw Cash در مورد کاربرد Customer Authentication قرار دارند

inclusion point<sup>34</sup>  
required<sup>35</sup>  
optional<sup>36</sup>

چندین نکته کلیدی از مشخصات UML 2.4 را در جدول زیر ترکیب کردم.

Include	Extend	Generalization
		
مورد کاربرد پایه ناقص است (مورد کاربرد مفهومی).	مورد کاربرد پایه به خودی خود کامل (واقعی) است که به طور مستقل تعریف شده است.	مورد کاربرد پایه می‌تواند مورد کاربرد مفهومی (ناقص) یا واقعی (کامل) باشد.
مورد کاربرد مشمول شده، اختیاری نیست و یک الزام است.	مورد کاربرد توسعه دهنده، اختیاری و تکمیلی است.	اگر مورد کاربرد پایه مفهومی باشد، مورد کاربرد تخصصی، اختیاری نیست و یک الزام است.
بدون مکان اشتراک بی واسطه، اما به واسطه برخی مکان‌ها، مشمول شده است.	حداقل یک مکان توسعه (extension point) بی واسطه دارد.	مکان مشخصی برای استفاده از تخصص وجود ندارد.
بدون شرط شمولیت بی واسطه.	می‌تواند شرط توسعه اختیاری داشته باشد.	بدون شرط بی واسطه برای استفاده در موضوع تخصصی کردن.

### ارتباطات از دست رفته Use Case ها

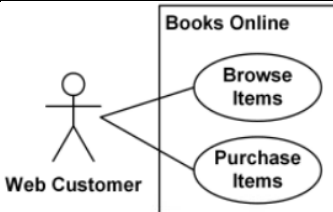
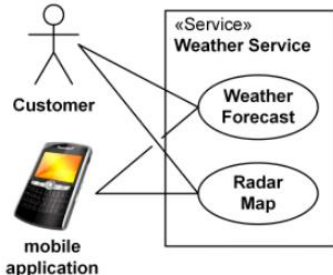


من هنگام ایجاد برخی نمودارهای مورد کاربرد و پاسخ به برخی از سؤالات دوستان، همواره با یک مشکل مشابه و بدون راه حل واضح در UML 2.4، روبرو هستم و گیج شدم. یک مثال به ظاهر ساده اما غیرقابل حل این است که روابط بین پرداخت (Payment)، پرداخت از طریق اعتبار (Payment by Credit)، پرداخت نقدی (Payment by Cash) و غیره را به گونه‌ای مشخص کنیم، که به مشتریان اجازه دهیم پرداخت خود را بخشی از طریق نقدی و بخشی را اعتباری و یا روش‌های دیگری، پرداخت کنند. با علم به اینکه این وضعیت کاملاً رایج است. تسویه حساب پایانه POS در سوپرمارکت‌ها. (نمونه مشابه دیگر عبارتند از برنامه‌ریزی سفر (Plan Trip)، کشف پرواز (Find Flight)، کشف هتل (Find Hotel) و کشف ماشین (Find Car) و غیره که در آن کاربر ممکن است بخواهد نه تنها برای یک بلکه برای چندین مورد در یک زمان جستجو کند).

اگرچه رابطه generalization موارد کاربرد برای پرداخت (Payment)، پرداخت از طریق اعتبار (Payment by Credit)، پرداخت با چک و غیره طبیعی به نظر می‌رسد، اما نمی‌توانیم از آن استفاده کنیم زیرا استفاده از یک روش پرداخت خاص را در هر زمان پیشنهاد می‌کند.



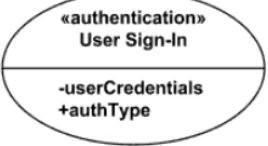

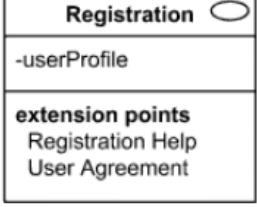
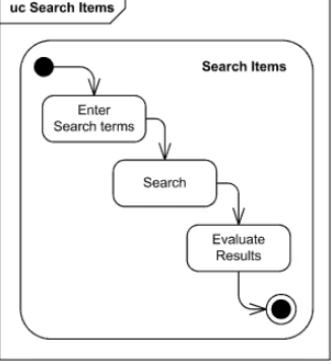
شاید فکر کنیم، رابطه Extend مناسب به نظر می‌رسد زیرا دارای نقاط توسعه با شرایط توسعه است، اما ما نمی‌توانیم از آن استفاده کنیم زیرا مورد کاربرد پایه Payment باید به خودی خود کامل باشد، که بدیهی است که موردی که ما داریم اینطور نیست.

شاید فکر کنیم، رابطه Include خوب به نظر می‌رسد زیرا مورد کاربرد پایه Payment به خودی خود کامل نیست و انواع مختلف پرداخت‌ها مکمل آن هستند، اما ما نمی‌توانیم از آن استفاده کنیم زیرا برای موارد مشمول شده، رابطه شمولیت، اختیاری نیست بلکه الزامی است (و هیچ شرط شمولیتی وجود ندارد) این بدان معناست که مشتری باید با استفاده از تمام روش‌های پرداخت، پرداخت کند؛ که کاملاً اشتباه است.

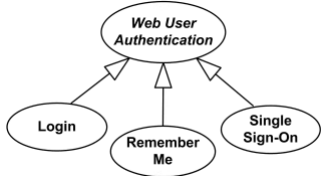
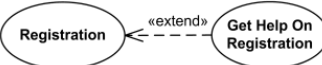
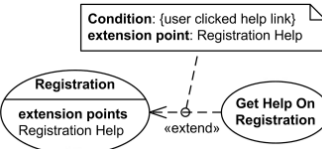

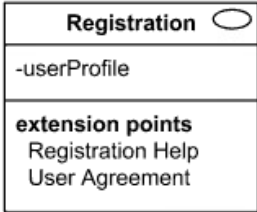
امیدوارم که متولیان این متودولوژی رابطه Include را به روز کنند تا شرایط و مکان‌های شمولیت را به روشی مشابه که برای رابطه Extend وجود دارد، مجاز بدانند، یا حداقل نمونه‌های نمودار مورد کاربرد بیشتری را در نسخه‌های آینده مشخصات UML ارائه کنند.

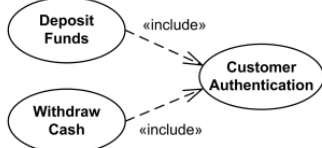
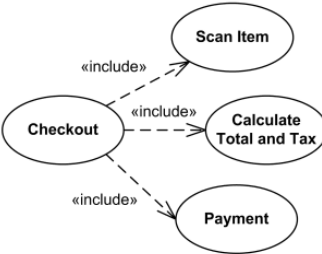
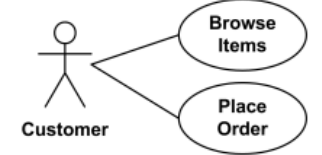

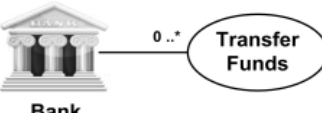
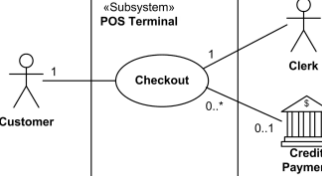
شرح	Notation
<b>Subject</b>	
<p>موضوع (موارد کاربرد)، سیستمی است که در حال طراحی یا بررسی است که مجموعه‌ای از موارد کاربرد برای آن قابل تصور است. موضوع، می‌تواند یک سیستم فیزیکی، برنامه نرم‌افزاری یا عنصر کوچکتر باشد که ممکن است رفتاری داشته باشد، به عنوان مثال، زیرسیستم، کامپوننت یا حتی کلاس.</p> <p>موضوع، توسط یک مستطیل با نام موضوع در گوشه بالایی با موارد کاربرد قابل اجرا در داخل مستطیل و کنشگران خارج از مرزهای آن ارائه می‌شود.</p>	 <p>Books Online</p> <p>Web Customer</p> <p>Browse Items</p> <p>Purchase Items</p> <p>کاربرد قابل اجرا و کنشگر Books Online موضوعی است که با موارد مرتبط است</p>
<p>کلیشه‌های استاندارد UML و کلمات کلیدی برای موضوع عبارتند از:</p> <p>«Subsystem» «Process» «Service» «Component»</p>	 <p>Customer</p> <p>mobile application</p> <p>«Service» Weather Service</p> <p>Weather Forecast</p> <p>Radar Map</p> <p>موضوع Weather Service کلیشه‌ای به عنوان «Service» دارد</p>
<b>Applicability of Use Cases</b>	
<p>موارد کاربرد که به صورت بصری در داخل مرزهای سیستم قرار دارند، موارد کاربرد قابل استفاده برای موضوع هستند (اما لزوماً متعلق به موضوع نیستند).</p>	 <p>Retail Website</p> <p>Browse Items</p> <p>Purchase Items</p> <p>تفسیر این نمودار: موارد کاربرد Browse Items و Buy Items برای موضوع Retail Website قابل استفاده هستند</p>
<b>Ownership of Use Cases</b>	
<p>تودرتو بودن (مالکیت) یک مورد کاربرد توسط یک طبقه‌بندی کننده با استفاده از نماد استاندارد برای طبقه‌بندی کننده‌های تودرتو نشان داده می‌شود.</p>	 <p>Retail Website</p> <p>Browse Items</p> <p>Purchase Items</p> <p>تفسیر این نمودار: موضوع Retail Website دارای موارد کاربرد Browse Items و Buy Items است</p>

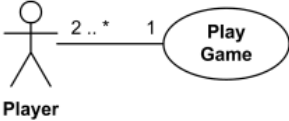
Actor	
<p>نماد استاندارد UML برای کنشگر، نماد "مرد چوبی" به همراه نام کنشگر در بالا یا پایین نماد است. نام کنشگران باید از دستورالعمل‌های حروف بزرگ و نقطه‌گذاری برای کلاس‌ها پیروی کند. نام کنشگران مفهومی باید به صورت مورب (ایتالیک) نشان داده شود. همه کنشگران باید نام داشته باشند.</p>	 <p><b>Student</b> کنشگری به نام Student</p>
<p>نمادهای سفارشی که نوع کنشگر را نشان می‌دهند ممکن است برای نشان دادن یک کنشگر نیز استفاده شوند، مانند استفاده از نماد(های) جداگانه برای کنشگران غیر انسانی.</p>	 <p><b>Web Client</b> نماد سفارشی برای کنشگر Web Client</p>
	 <p><b>Bank</b> نماد سفارشی برای کنشگر Bank</p>
<p>یک کنشگر کسبوکاری (که در RUP برای حمایت از مدل‌سازی کسبوکار معرفی شده است) نقشی را نشان می‌دهد که توسط شخص یا سیستمی، خارج از کسبوکار مدل‌شده، ایفا می‌شود. کنشگر کسبوکاری، بخشی از استاندارد UML نیست. کنشگر کسبوکاری به‌عنوان نماد «stick man» با خط موربی در قسمت سرش، نشان داده می‌شود.</p>	 <p><b>Passenger</b> کنشگر کسبوکار به نام Passenger</p>
<p>همچنین ممکن است یک کنشگر به صورت مستطیل که معرف یک کلاس است به همراه کلمه کلیدی «actor» نشان داده شود که در صورت لزوم دارای بخش‌های عمومی مانند کلاس است.</p>	<div style="border: 1px solid black; padding: 5px;"> <p><b>«actor» Customer</b></p> <p>+ name: Name + address: Address</p> </div> <p>کنشگر Customer در نمای کلاس</p>
Generalization between actors	
<p>رابطه Generalization یا تعمیم بین کنشگران به صورت یک خط مستقیم جهت‌دار با نوک پیکان بزرگ ارائه می‌شود (همانطور که برای رابطه Generalization یا تعمیم بین کلاس‌ها تعریف می‌شود).</p>	 <p>کنشگر Web Client یک سوپر کلاس مفهومی برای Administrator، Editor و Customer است</p>
Use Case	
<p>هر مورد کاربرد باید یک نام داشته باشد. Use case به صورت بیضی حاوی نام مورد کاربرد، نشان داده می‌شود.</p>	 <p>مورد کاربردی به نام User Registration</p>

<p>یک مورد کاربرد را می‌توان به صورت بیضی نشان داد که نام مورد کاربرد در زیر بیضی قرار داده شده است.</p>	 <p><b>Transfer Funds</b></p> <p>مورد کاربردی به نام User Registration</p>
<p>مورد کاربرد کسبوکاری در RUP برای پشتیبانی از مدل‌سازی کسبوکار - برای نشان دادن عملکرد، فرآیند یا فعالیت کسبوکار وقوع یافته در کسبوکار مدل‌شده، معرفی شد.</p> <p>مورد کاربرد کسبوکاری در RUP با نماد بیضی شکل و خط موربی که از سمت راست آن عبور می‌دهند، نشان داده می‌شود.</p>	 <p><b>Individual Check-In</b></p> <p>مورد کاربرد کسبوکار به نام Individual Check-In</p>
<p>ممکن است یک کلمه کلیدی کلیشه‌ای اختیاری در بالای نام و فهرستی از ویژگی‌ها شامل عملیات و فیلدها یا مشخصات در زیر نام در یک بیضی قرار داده شوند.</p>	 <p>استفاده از کلیشه ای به عنوان «authentication» برای مورد کاربرد User Sign-In</p>
<p>موارد کاربرد با نقاط توسعه (extension points) ممکن است در داخل بیضی مورد کاربرد با عنوانی به نام extension points برای فهرست آنها، نمایش داده شوند.</p>	 <p>مورد کاربردی به نام Registration که دارای نقاط توسعه Registration Help و User Agreement است</p>
<p>یک مورد کاربرد همچنین می‌تواند با استفاده از نماد مستطیل استاندارد برای طبقه‌بندی کننده‌ها با نماد بیضی در گوشه سمت راست بالای مستطیل و با محیط‌های لیست‌گونه جداگانه اختیاری برای ویژگی‌های آن نشان داده شود.</p>	 <p>مورد کاربردی به نام Registration با استفاده از نماد مستطیل استاندارد برای طبقه‌بندی کننده‌ها نشان داده شده است</p>
<p>Use Case را می‌توان در قاب با برچسب use case یا به صورت اختصاری UC ارائه کرد. ناحیه محتوای قاب می‌تواند شامل انواع مختلفی از نمودارهای UML باشد. به عنوان مثال، مورد کاربرد را می‌توان با activity diagram یا state machine توصیف کرد.</p>	 <p>مورد کاربردی Search Items که به شکل قاب ارائه شده است activity diagram</p>



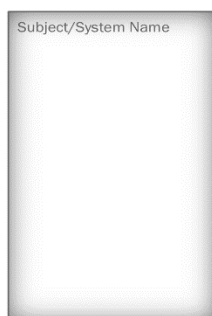
	مربوط به Search Items در آن گنجانده شده است
<b>Generalization between use cases</b>	
<p>رابطه Generalization یا تعمیم بین موارد کاربرد شبیه به رابطه Generalization یا تعمیم بین کلاس‌ها است. مورد کاربرد فرعی، ویژگی‌ها و رفتار مورد کاربرد والدین را به ارث می‌برد و ممکن است رفتار والدین را بازنویسی کند (override).</p> <p>این رابطه به شکل یک خط مستقیم با یک پیکان بزرگ ارائه می‌شود، مانند رابطه Generalization یا تعمیم بین طبقه‌بندی کننده‌ها.</p>	 <p>مورد کاربرد Web User Authentication یک مورد کاربرد مفهومی است و توسط موارد کاربرد Login، Remember Me و Single Sign-On تعریف می‌شود</p>
<b>Extend</b>	
<p>رابطه Extend یک رابطه جهت‌دار است که مشخص می‌کند چگونه و چه زمانی رفتار تعریف‌شده در موارد کاربرد توسعه‌دهنده مکمل (اختیاری) را می‌توان در رفتار تعریف‌شده در مورد کاربرد توسعه‌یافته درج کرد.</p> <p>مورد کاربرد توسعه‌یافته به خودی خود معنادار است، مستقل از مورد کاربرد توسعه‌دهنده است. مورد کاربرد توسعه‌دهنده معمولاً رفتار اختیاری را تعریف می‌کند که لزوماً به خودی خود معنی ندارد.</p> <p>رابطه Extend بین موارد کاربرد با یک فلش چین‌دار با سر پیکان باز از مورد کاربرد توسعه‌دهنده به مورد کاربرد توسعه‌یافته (پایه) نشان داده می‌شود. فلش با کلمه کلیدی «extend» برچسب‌گذاری شده است.</p>	 <p>مورد کاربرد Registration به تنهایی کامل و معنادار است. می‌توان آن را با موارد کاربرد اختیاری Get Help On Registration توسعه داد.</p>
<p>شرایط رابطه Extend و همچنین ارجاع به نقاط توسعه (extension points) به صورت اختیاری در یادداشت پیوست شده به رابطه Extend مربوطه نشان داده می‌شود.</p>	 <p>مورد کاربرد Registration به صورت مشروط توسط Get Help On Registration Use case در نقطه توسعه Registration Help توسعه داده می‌شود</p>
<b>Extension Point</b>	
<p>نقطه توسعه یا Extension Point یک ویژگی برای مورد کاربرد است که نقطه‌ای را در رفتار مورد کاربرد شناسایی می‌کند (ارجاع می‌دهد) که در آن نقطه، همانطور که در رابطه Extend مشخص شده است، آن رفتار می‌تواند توسط یک مورد کاربرد دیگر (توسعه دهنده) توسعه یابد.</p> <p>نقاط توسعه یا Extension Point ممکن است در محفظه‌ای از نماد بیضی شکل مورد کاربرد در زیر عنوان extension points نشان داده شوند. هر نقطه توسعه یا Extension Point باید یک نام داشته باشد که در یک مورد کاربرد منحصر به فرد باشد. نقاط توسعه یا Extension Point به صورت یک متن شبیه به نگارشی که در زیر نشان داده شده است، نمایش داده شوند:</p> <p><b>extension point ::= name [ : explanation ]</b></p>	 <p>مورد کاربرد Registration با نقاط توسعه User و Registration Help Agreement</p>
<p>نقاط توسعه یا Extension Point ممکن است در محفظه‌ای از مستطیل مورد کاربرد با نماد بیضی در زیر عنوان extension points نشان داده شوند.</p>	 <p>نقاط توسعه مورد کاربرد Registration که با استفاده از نماد مستطیل نشان داده شده است</p>
<b>Include</b>	

<p>یک رابطه Include یک رابطه مستقیم بین دو مورد کاربرد است، یک رفتار الزامی (نه رفتار اختیاری) مورد کاربرد مشمول شده در رفتار مورد کاربرد مشمول کننده (پایه) درج می‌شود.</p> <p>رابطه Include مشابه یک فراخوانی زیر روال یا ماکرو است و می‌تواند مورد استفاده قرار گیرد:</p> <ul style="list-style-type: none"> <li>• برای ساده کردن موارد کاربرد بزرگ با تقسیم آن به چندین مورد کاربرد</li> <li>• برای استخراج بخش‌های مشترک از رفتارهای دو یا چند مورد کاربرد</li> </ul> <p>یک رابطه Include بین موارد کاربرد با یک فلش چین‌دار با سر پیکان باز از مورد کاربرد پایه تا مورد کاربرد مشمول شده نشان داده می‌شود. فلش با کلمه کلیدی «include» برچسب‌گذاری می‌شود.</p>	 <p>موارد کاربرد Deposit Funds و Withdraw Cash در مورد کاربرد Customer Authentication قرار دارند</p>
<p>موارد کاربرد بزرگ و پیچیده را می‌توان با تقسیم آن به چند مورد کاربرد، ساده کرد که هر کدام واحد منطقی رفتار را توصیف می‌کنند. توجه داشته باشید که موارد کاربرد مشمول کننده به خودی خود ناقص هستند و نیاز به موارد کاربرد مشمول شده برای کامل شدن دارند.</p>	 <p>مورد کاربرد Checkout شامل چندین مورد کاربرد است یا می‌توان گفت که این مورد کاربرد، مشمول کننده چندین مورد کاربرد است؛ موارد کاربرد مشمول شده Scan Item، Calculate Total and Tax، و Payment این موارد کاربرد هستند</p>
<p style="text-align: center;"><b>Association</b></p>	
<p>ارتباط بین یک کنشگر و یک مورد کاربرد، نشان می‌دهد که کنشگر و مورد کاربرد با یکدیگر ارتباط برقرار می‌کنند. یک کنشگر می‌تواند به یک یا چند مورد کاربرد مرتبط باشد.</p>	 <p>کنشگر Customer با دو مورد کاربرد Browse Items و Place Order مرتبط است</p>
<p>مورد کاربرد، ممکن است یک یا چند کنشگر مرتبط داشته باشد. ممکن است از نمودار مورد کاربرد، مشخص نباشد که کدام کنشگر، شروع کننده مورد کاربرد است، یعنی «کنشگر اصلی» است. (در UML غیر استاندارد، کنشگران اصلی کسانی هستند که از خدمات سیستم استفاده می‌کنند و کنشگران پشتیبان کنشگرانی هستند که خدمات را به سیستم ارائه می‌دهند.)</p>	 <p>مورد کاربرد Manage Account با کنشگران Customer و Bank مرتبط است</p>
<p>هنگامی که یک کنشگر، یک ارتباطی چندگانه با یک مورد کاربرد دارد که در انتهای خط ارتباط مورد نظر، مورد کاربرد بیشتر از یک است، به این معنی است که یک کنشگر معین می‌تواند در چندین مورد کاربرد از آن نوع، دخالت داشته باشد. ماهیت خاص این درگیری چندگانه در UML 2.2 تعریف نشده است.</p> <p>تعدد موارد کاربرد می‌تواند به این معنی باشد که یک کنشگر، چندین مورد کاربرد را آغاز می‌کند:</p> <ul style="list-style-type: none"> <li>• به صورت موازی (concurrently)</li> <li>• در مقاطع مختلف زمانی</li> <li>• در زمان‌های مجزا</li> </ul>	 <p>کنشگر Bank در چندین مورد کاربرد از Transfer Funds (تبادلات صندوق‌ها) شرکت دارد</p>
<p>کنشگر الزامی، ممکن است به صراحت با عدد 1 یا بیشتر نشان داده شود. UML 2.5 همچنین اجازه می‌دهد تا کنشگر اختیاری (0) باشد. تعدد 0..1 کنشگر به این معنی است که کنشگر توسط هیچ یک از موارد کاربرد مرتبط، مورد نیاز نیست.</p>	

	<p>مورد کاربرد Checkout به 1 عدد کنشگر Customer نیاز دارد.</p> <p>مورد کاربرد مورد نظر ممکن است به خدمات Credit Payment نیاز نداشته باشد (مثلاً اگر پرداخت به صورت نقدی باشد).</p> <p>بنابراین ارتباط بین این مورد کاربرد با این خدمت در هر سمت می‌تواند عدد 0 را نیز داشته باشد.</p>
<p>هنگامی که یک مورد کاربرد با یک کنشگر، ارتباطی به حالت چندگانگی دارد که در سمت کنشگر بیشتر از یک است، به این معنی است که بیش از یک کنشگر در شروع مورد کاربرد، دخیل هستند. روشی که چندین کنشگر در مورد کاربرد، شرکت می‌کنند در UML 2.x تا UML 2.5 تعریف نشده است.</p> <p>به عنوان مثال، تعدد کنشگران می‌تواند به این معنی باشد که:</p> <ul style="list-style-type: none"> <li>• یک مورد کاربرد خاص، ممکن است مستلزم اقدام همزمان (concurrent) توسط دو کنشگر مجزا باشد (مثلاً در پرتاب یک موشک هسته‌ای)</li> <li>• ممکن است نیاز به اقدامات تکمیلی و متوالی توسط کنشگران داشته باشد (مثلاً یک کنشگر چیزی را شروع کند و دیگری آن را متوقف کند).</li> </ul>	 <p>دو یا چند کنشگر Player در مورد کاربرد Play Game دخیل هستند.</p> <p>هر کنشگر در یک بازی Play شرکت می‌کند</p>

**Subject:**

موضوع در یک نمودار مورد کاربرد، نمایی از یک حوزه، سیستم و یا زیرسیستم را معرفی می‌کند. موارد کاربرد در داخل آن و کنشگران در خارج آن قرار می‌گیرند.

**Stereotype:**

کلیشه یا عبارتی که در مراحل شناسایی و یا طراحی یک سیستم، معنی مشخصی دارد و نوعی توصیف از طبقه‌بندی خاصی است. عبارت مورد نظر به صورت تشریحی و به هم چسبیده (بدون فضای خالی) نوشته می‌شود و داخل علامت‌های «» قرار می‌گیرد. برای نمونه می‌توان به کلیشه‌های زیر اشاره کرد:

- «Business»
- «Scope»
- «Department»
- «Subsystem»
- «Process»
- «Service»
- «Component»

**Actor:**

کنشگر عنصری عملگرا است که در کسب‌وکار مورد بررسی، یک عامل فعال خارج از کسب‌وکار مورد نظر محسوب می‌شود. این عنصر از طریق رابط‌هایی با موضوع، موارد کاربرد یا کسب‌وکار مورد نظر در ارتباط خواهد بود.



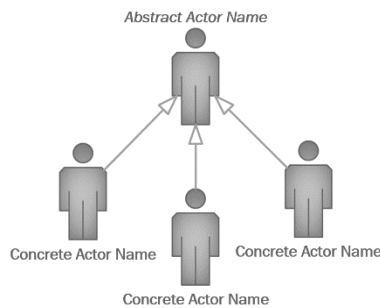
### Business Actor:

کنشگری که در معرف یک نقش کاربری خاص در کسب و کار است.



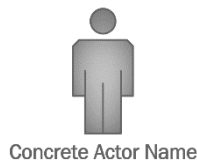
### Actor Generalization:

نوعی ارتباط ساختاری یا رفتاری بین کنشگران مفهومی و واقعی است به صورتیکه به واسطه آن مشخص می‌شود، کدام کنشگران واقعی، کنشگر مفهومی مورد نظر را توصیف می‌کنند.



### Concrete Actor:

معرف یک کنشگر واقعی است که به احتمال زیاد برای ایجاد ساختار و پیاده‌سازی رفتارهای آن از یک یا مجموعه‌ای از کلاس‌ها استفاده خواهد شد. به عبارت دیگر این کنشگر به صورت نرم‌افزاری، قابل پیاده‌سازی است.



### Abstract Actor:

معرف یک کنشگر مفهومی است که به صورت نرم‌افزاری، قابل پیاده‌سازی نمی‌باشد و از طرف دیگر به تنهایی نمایندگی کنشگرهای زیادی را به عهده خواهد داشت. توجه داشته باشید که نام این کنشگر به صورت ایتالیک نوشته خواهد شد.



## Use Case:

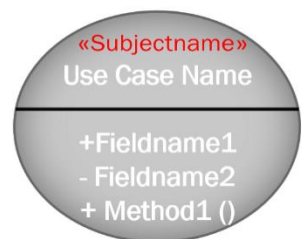
عنصری است که یک واحد رفتاری یا عملیاتی مشخص را که در یک موضوع، کسب‌وکار یا الگوریتم وجود دارد، معرفی می‌نماید.



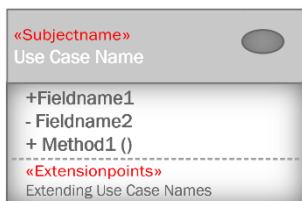
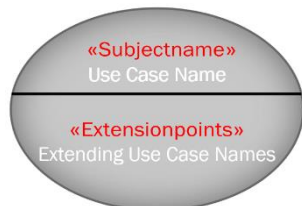
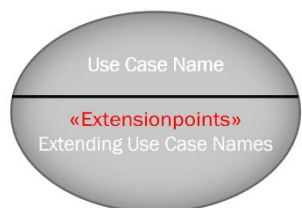
این عنصر می‌تواند در داخل یک سیستم یا موضوع مورد استفاده قرار گیرد. در این شرایط برخی مواقع ممکن است به واسطه یک کلیشه که توصیف‌کننده سیستم یا موضوع مورد نظر است تعریف شود که در این شرایط به شکل زیر خواهد بود.



این عنصر به دلیل اینکه می‌تواند نماینده یک کلاس باشد، می‌تواند فیلدها و متدهای آن کلاس را نیز در نمای خود نمایش بدهد.



این عناصر می‌توانند به صورت سلسله‌مراتبی همدیگر را تولید نمایند در این شرایط هر یک از عناصر رأس این سلسله‌مراتب، در هر نقطه دارای یک تعداد نقاط توسعه خواهند بود در این شرایط می‌توان یک مورد کاربرد را به اشکال زیر نمایش داد.



### Business Use Case:

مورد کاربرد کسب‌وکار، یک عملکرد، فرآیند یا فعالیت کسب‌وکاری را معرفی می‌کند. این موارد کاربرد می‌توانند پایه فعالیت‌ها، زیرفرآیندها و فرآیندهای کسب‌وکار در سایر نمودارهای سایر استانداردها مثل BPMN باشند.



### Abstract Use Case:

مورد کاربردی که قابل تقسیم به موارد کاربرد دیگری است که بتوانند در اجماع با هم کل عملیات و رفتار آن را معرفی کنند و مورد کاربرد مورد نظر به تنهایی و مستقل، معنی نداشته باشد و قابل تعریف نباشد، مورد کاربرد مفهومی نام دارد. توجه داشته باشید که نام این مورد کاربرد به شکل ایتالیک در نمای مورد کاربرد نمایش داده می‌شود.



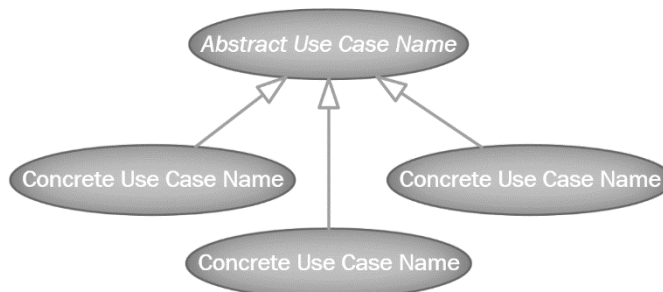
### Concrete Use Case:

مورد کاربردی که به تنهایی و مستقل، معنی داشته باشد و قابل تعریف باشد چه قابل تقسیم به موارد کاربرد دیگری باشد یا نباشد، مورد کاربرد واقعی نام دارد.



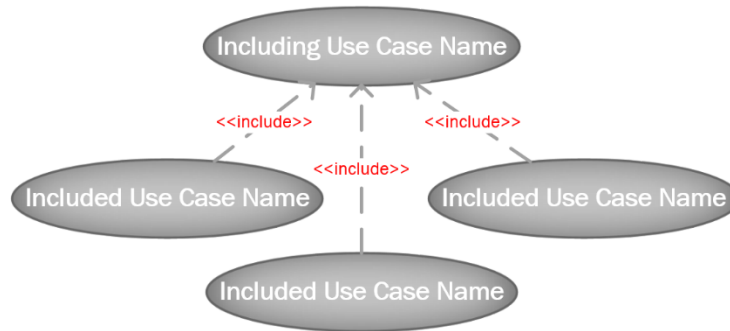
### Use Case Generalization:

معرف ارتباط خاص بین موارد کاربرد مفهومی و واقعی است. در این نوع ارتباط، موارد کاربرد واقعی تشکیل دهنده یا تولید کننده مورد کاربرد مفهومی هستند.



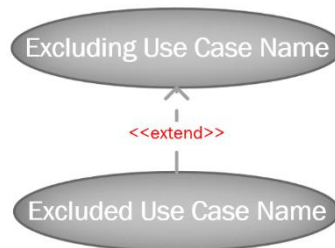
### Use Case Inclusion:

معرف ارتباط خاص بین موارد کاربرد پایه (شامل بر/تشکیل شونده) و پیرو (تشکیل دهنده) است. در این نوع ارتباط، موارد کاربرد پایه بخشی از رفتار خود را از مورد کاربرد پیرو دریافت می‌کنند.



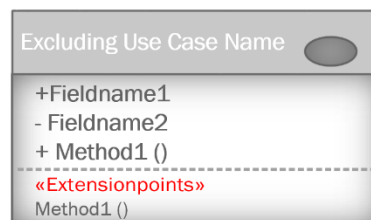
### Use Case Extension:

معرف ارتباط خاص بین موارد کاربرد پایه (توسعه پذیر) و پیرو (توسعه دهنده) است. در این نوع ارتباط، رفتار موارد کاربرد پایه توسط مورد کاربرد پیرو توسعه می‌یابد و شاید حالت خاص بگیرند.



### Extension Point:

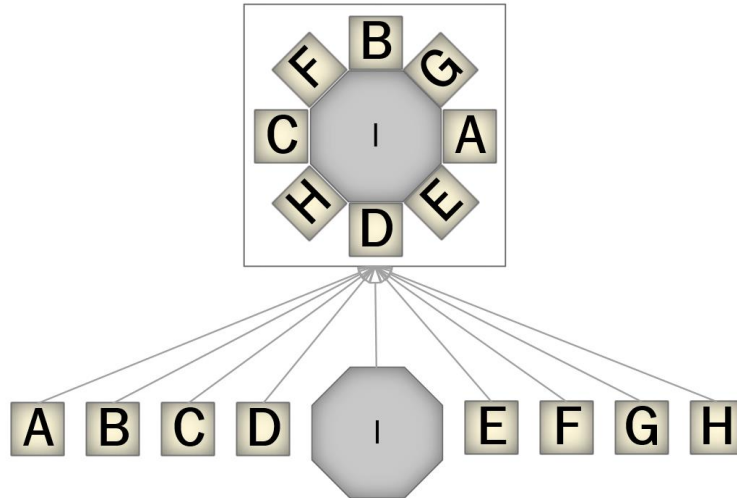
نقطه توسعه یک ویژگی از یک مورد کاربرد است که نقطه‌ای را در رفتار مورد کاربرد توسعه پذیر معرفی می‌کند (ارجاع می‌دهد) که در آن نقطه، آن رفتار می‌تواند توسط یک مورد کاربرد توسعه دهنده، بر اساس این رابطه، توسعه یابد.



### Generalization Relationship:

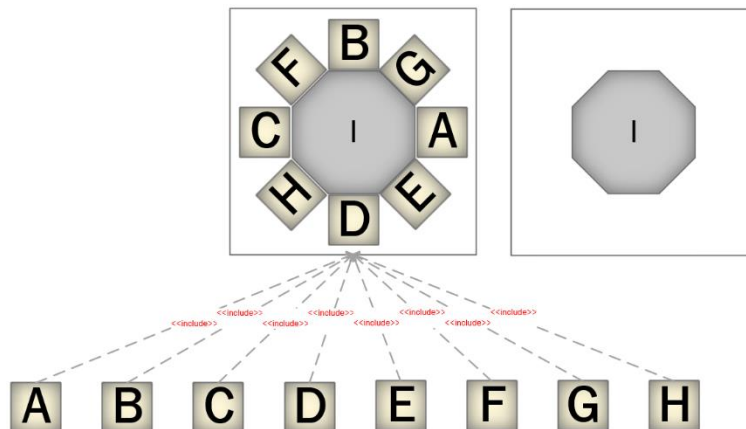
ارتباط ساخت یا تولید، نوع خاصی از ارتباط بین عناصر مختلف است که در آن عنصر ساخته شده، به صورت مستقل هیچ معنی ندارد و قابل تعریف نیست (عنصر مفهومی) و با مجتمع کردن عناصر سازنده در یک محیط، ماهیت پیدا کرده و قابل تعریف می‌شود. در واقع در این مدل، تمام عناصر سازنده به صورت مستقل معنی دارند و در ارتباطات دیگر مورد استفاده قرار می‌گیرند.





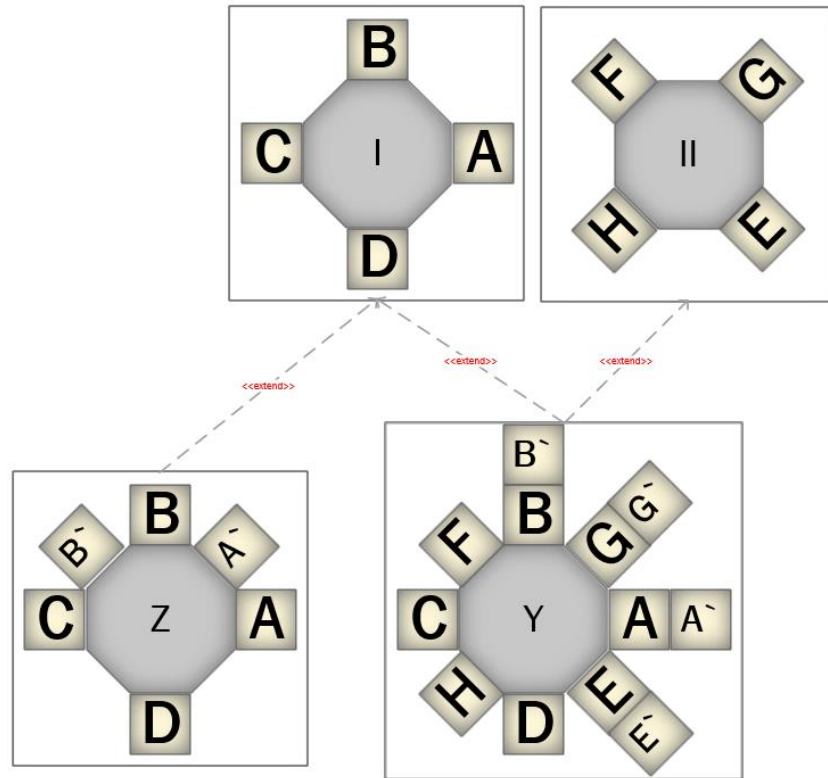
### Inclusion Relationship:

ارتباط شمولیت، نوع خاصی از ارتباط بین عناصر مختلف است که در آن عنصر مشترک پذیر، به صورت مستقل معنی دارد و قابل تعریف است (عنصر پایه) و با مجتمع کردن عناصر سازنده و خود در یک محیط، ماهیت پیدا کرده و قابل تعریف می‌شود. در واقع در این مدل، تمام عناصر سازنده به صورت مستقل معنی دارند و در ارتباطات دیگر مورد استفاده قرار می‌گیرند ولیکن عنصر پایه با وجود اینکه به صورت مستقل معنی دارد ولیکن در هیچ ارتباط دیگری قابل استفاده نیست.



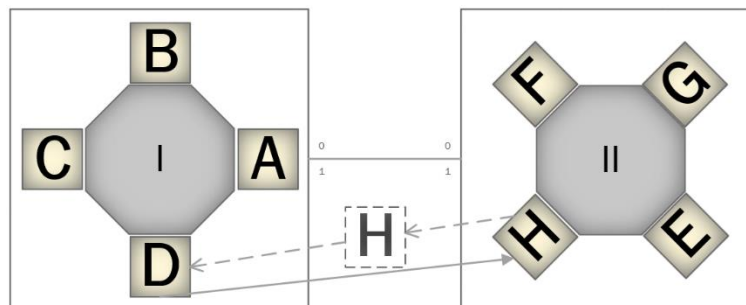
### Extension Relationship:

رابطه توسعه، نوع خاصی از ارتباط بین عناصر مختلف است که در آن عنصر توسعه پذیر، به صورت مستقل معنی دارد و قابل تعریف است (عنصر پایه). عناصر توسعه دهنده، کل یا بخشی از ماهیت عنصر توسعه پذیر را در داخل خود توسعه می‌دهند. ماهیت عناصر توسعه دهنده فارق از توسعه صورت گرفته، کاملاً مستقل از عنصر توسعه پذیر است. در واقع در این مدل، تمام عناصر توسعه پذیر و توسعه دهنده به صورت مستقل معنی دارند.



### Association Relationship:

رابطه وابستگی دو طرفه، نوع خاصی از ارتباط بین عناصر مختلف است که در آن عناصر دو طرف رابطه، به صورت مستقل معنی دارند و قابل تعریف هستند ولی ممکن است هر عنصر، ماهیتی/ماهیت‌هایی از عنصر دیگر را فراخوانی نماید و یا با آن تعامل نماید. عناصر دو طرف رابطه هیچ بخش مشترکی با هم نخواهند داشت و فقط از مشخصات همدیگر برای رسیدن به اهداف خود استفاده می‌کنند. اگر این بهره برداری برای رسیدن به هدف، فقط بر روی یک مشخصه باشد؛ یک رابطه ساده است و چنانچه بر روی چند مشخصه باشد رابطه چندگانه است.



### Dependency Relationship:

رابطه وابستگی یک طرفه، نوع خاصی از ارتباط بین عناصر مختلف است که در آن عناصر دو طرف رابطه، به صورت مستقل معنی دارند و قابل تعریف هستند ولی ممکن است عنصر یک طرف رابطه (عنصر پیرو)، ماهیتی از عنصر دیگر (عنصر پایه) را فراخوانی نماید و یا با آن تعامل نماید. عناصر دو طرف رابطه هیچ بخش مشترکی با هم نخواهند داشت و فقط عنصر پیرو از مشخصات عنصر پایه برای رسیدن به اهداف خود استفاده می‌کند. اگر این بهره برداری برای رسیدن به هدف، فقط بر روی یک مشخصه باشد؛ یک رابطه ساده است و چنانچه بر روی چند مشخصه باشد

رابطه چندگانه است. و به جهت اینکه ماهیت عناصر دو طرف این رابطه فرق دارد (یکی پایه و دیگری پیرو است)، چندگانگی در هر دو طرف معنی دارد.

