



Unified Modeling Language

ACTIVITY DIAGRAMS

Inleiding

Use case diagrammen laten zien wat het (informatie)systeem zou moeten doen. Activiteiten diagrammen laten zien hoe dat moet worden verwezenlijkt. Een activiteiten diagram toont het verloop van de processen binnen een enkele use case en vormen daarmee een specificatie van de processen binnen die use case. Een activiteiten diagram modelleert het dynamische gedrag van een (informatie)systeem, waarbij de focus ligt op de processen.

Voor elke use case wordt een activiteiten diagram gemodelleerd. Aan de basis van een activiteiten diagram ligt de specificatie van de use case.

Onderdelen van het activiteiten diagram

Een activiteiten diagram kan de volgende onderdelen bevatten:

- een start- en één of meer eindnodes (initial node, final node)
- pijlen die de stromen (flow, edge, path) weergeven
- één of meer processen (proces or action)
- één of meer beslissingsruiten (decision node)
- één of meer samenvoegruiten (merge node)
- één of meer vorken (fork)
- één of meer joins
- één of meer timers
- één of meer objecten
- eventueel een activiteiten frame

Startnode



De startnode geeft het begin van het diagram aan. Vanaf dit punt worden de activiteiten binnen een use case gestart.

Eindnode



De eindnode geeft het einde van het diagram aan. Aangekomen op dit punt zijn alle noodzakelijke stappen van de activiteit doorlopen. Er mag meer dan één eindnode in een activiteiten diagram worden opgenomen, maar het is “good practice” om het gebruik van de eindnode zo veel mogelijk te beperken tot één enkele eindnode.

Stroom



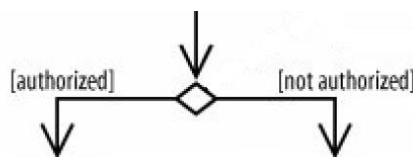
De stroom, flow, edge of path wordt weergegeven door een pijl en geeft de richting aan van berichten tussen de verschillende stappen in een activiteit. De hier afgebeelde pijl is naar beneden gericht, maar kan ook naar boven, naar links en naar rechts zijn gericht.

Proces



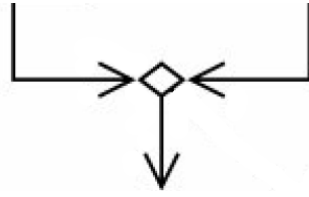
Een proces of actie is een enkele stap in een activiteit en wordt weergegeven door een rechthoek met afgeronde hoeken. Elke stap in een activiteit wordt in een afzonderlijke actie gemodelleerd.

Beslissingsruit



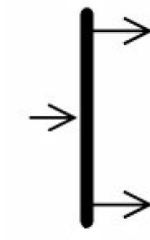
Zoals de naam al aangeeft, wordt bij een beslissingsruit een beslissing genomen. Een beslissingsruit heeft één ingaande en twee of meer uitgaande stromen. In de stap voor de beslissingsruit zal een vraag worden gesteld waarop twee antwoorden mogelijk zijn: “waar” of “niet waar”. Het is natuurlijk heel goed mogelijk dat in een stap drie of meer antwoorden mogelijk zijn, waarbij zelfs twee of meer antwoorden kunnen resulteren in “waar”. Er zullen dan meerdere beslissingsruiten moeten worden gebruikt, omdat er anders onvoorspelbare en ongewenste resultaten ontstaan. Een uitzondering hierop vormt de vraagstelling waarbij van alle antwoorden er altijd maar één waar kan zijn.

Samenvoegingsruit



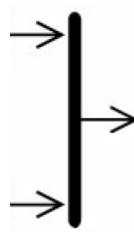
Bij een samenvoegingsruit worden twee of meer stromen samengevoegd tot één enkele stroom. In plaats van een samenvoegingsruit kan ook een join worden gebruikt. Het is zaak voorzichtig te zijn met een samenvoegingsruit of join. Een bepaalde stroom mag niet onbedoeld naar een onjuiste actie leiden.

Vork



Een vork wordt gebruikt om een enkele stroom te leiden naar twee of meer parallelle acties. Die vervolgacties worden gelijktijdig uitgevoerd met een identieke ingaande stroom, de uitgaande stromen van de vork. In tegenstelling tot de beslissingsruit worden bij de vork de beide uitgaande stromen gevolgd. Bij de beslissingsruit worden slechts één uitgaande stroom gevolgd, de stroom die het resultaat is van het antwoord “waar”.

Join

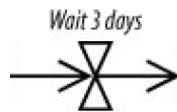


Bij de join worden, net als bij de samenvoegingsruit, twee of meer stromen samengevoegd tot een enkele stroom. Wanneer twee acties gelijktijdig worden uitgevoerd, betekent dit niet dat die twee acties gelijktijdig zijn afgerond. De join zorgt ervoor dat de uitgaande stroom niet eerder wordt gevolgd dan dat alle ingaande stromen bij de join zijn gearriveerd. Hetzelfde geldt voor de samenvoegingsruit.

Opdracht 1

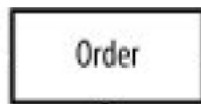
Maak met behulp van de hiervoor genoemde onderdelen een activiteiten diagram op basis van de uitwerking van opdracht 5 uit de reader “Use Case Diagrams”.

Timer



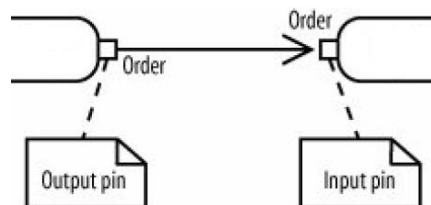
Een timer kan worden gebruikt om een vertraging in te stellen op de voortgang van een stroom. Bij de timer wordt de vertraging aangegeven in de gewenste tijdseenheid. Dit kunnen milliseconden, seconden, minuten, uren of zelfs dagen zijn. De timer heeft een ingaande en uitgaande stroom, die identiek aan elkaar zijn.

Object



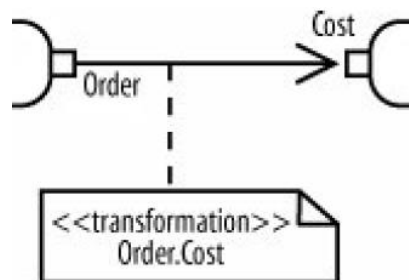
Een object kan worden gebruikt om aan te geven welke data van de ene stap in een activiteit worden overgedragen aan de andere stap. Een object wordt weergegeven door een rechthoek waarin de naam van het object wordt geplaatst. Het is niet strikt noodzakelijk om objecten op te nemen in een activiteiten diagram, maar kan handig zijn wanneer omliggende acties een object aanmaken, gebruiken of wijzigen.

Input en Output



In plaats van objecten, kan ook gebruik gemaakt worden van input en output pinnen, waarmee wordt aangegeven dat een object wordt overgedragen van de ene actie op de andere.

Transformatie



In de voorgaande voorbeelden met betrekking tot objecten, worden alle data van een object van de ene actie doorgegeven aan de andere actie. Wanneer slechts een deel van de data in de volgende actie nodig is, kan transformatie worden toegepast. Transformatie wordt weergegeven door middel van een notificatie die met een onderbroken lijn wordt verbonden met de stroom. In de notificatie wordt het prototype «transformation» geplaatst met daaronder de naam van het vereiste deel van de data van het object.

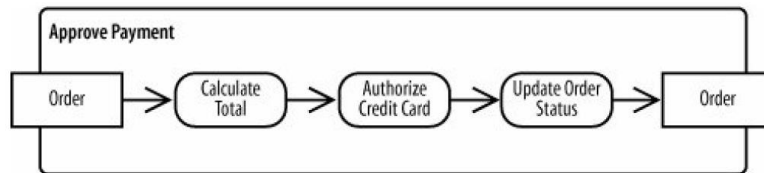
Hoe objecten veranderen tijdens een activiteit

Gedurende een activiteit kan de status van een object veranderen. Dit kan in het object worden aangegeven door de status van het object tussen blokhaken onder de naam van het object te plaatsen.



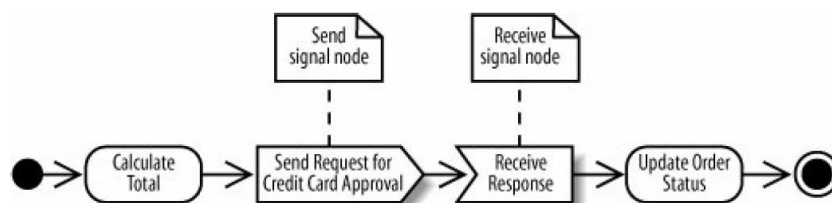
Objecten als input en output

In de hiervoor staande afbeelding is bij de actie “Approve Payment” een vorkje voor de naam van de actie geplaatst. Dit geeft aan dat deze actie in een ander diagram meer gedetailleerd is gemodelleerd. Het object “Order” wordt daarbij gebruikt als input en output.

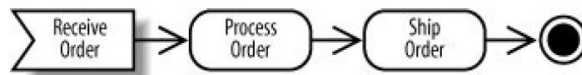


Signalen

Bij activiteiten kan sprake zijn van interactie met externe personen, systemen of processen. Wanneer bijvoorbeeld een betaling door middel van een creditcard moet worden geautoriseerd, zal interactie nodig zijn met de creditcard maatschappij. Om deze interactie met externen weer te geven worden signalen gebruikt. Er is sprake van inkomende en uitgaande signalen.



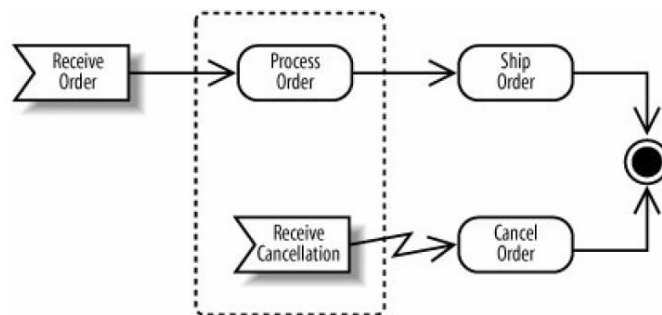
Een inkomend signaal kan gebruikt worden ter vervanging van de startnode. Dit houdt dan in dat de node *altijd* wacht op een signaal gedurende de periode dat de activiteit actief is.



In dit geval wacht de “Receive Order” node **altijd** in tegenstelling tot de “Receive Response” node in de vorige afbeelding, die alleen “begint met wachten” wanneer de voorgaande actie (“Calculate Total”) is afgerond.

Activiteiten onderbreken

In de diverse voorbeelden hiervoor is het afhandelen van een order vrij rechtlijnig en eenvoudig gemodelleerd. Een order kan worden geannuleerd. In het activiteiten diagram kan hiervoor een onderbrekingsgebied (interruption region) worden gemodelleerd, die wordt weergegeven door middel van een rechthoek met onderbroken lijn en afgeronde hoeken.



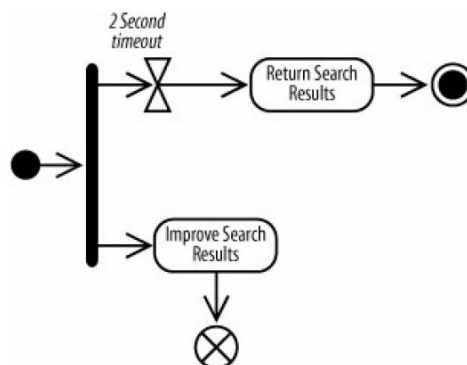
De uitgaande stroom na het inkomende signaal (de order wordt geannuleerd), wordt weergegeven als een bliksemschicht.

Opdracht 2

Pas dit diagram zodanig aan, dat een geplaatste order binnen drie dagen kan worden geannuleerd. Na drie dagen kan de order niet meer worden geannuleerd en wordt de order verzonden.

Activiteit stoppen

Een activiteit stopt normaal gesproken met de eindnode. Nieuw in UML 2.0 is de zogenaamde stroom-eindnode (flow final node).



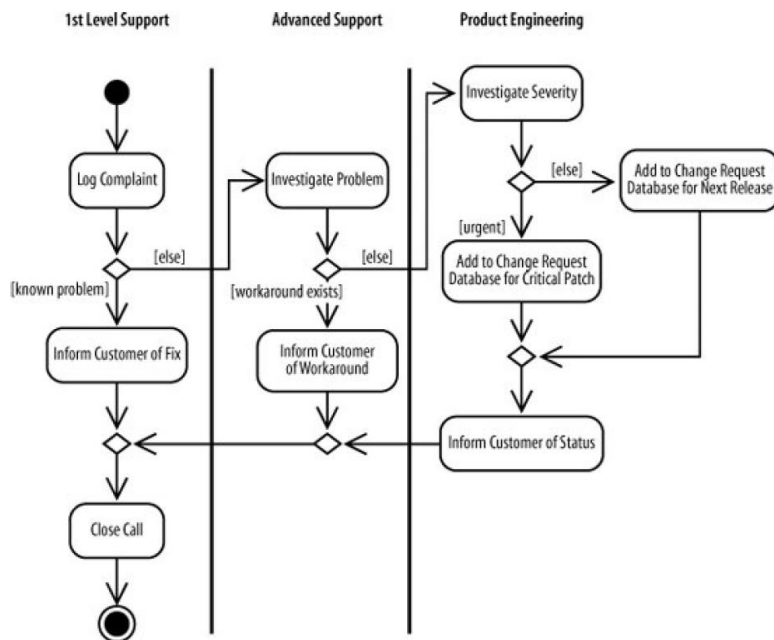
Een stroom-eindnode, weergegeven door een open rondje met daarin een kruis, stopt alleen de eigen stroom, maar niet de gehele activiteit.

Opdracht 3

Beschrijf in woorden de gebeurtenissen die worden weergegeven door dit laatste diagram.

Partities of zwembanen

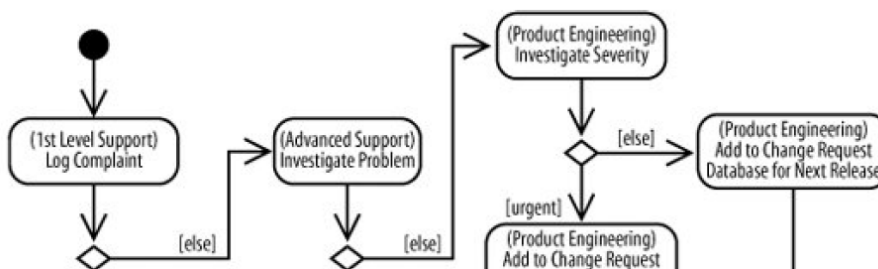
Bij een activiteit kunnen meer dan één rol betrokken zijn. Bepaalde stappen binnen een activiteit kunnen toebehoren aan een bepaalde rol, gespeeld door externe personen, systemen of processen. De stappen binnen een activiteit kunnen worden geordend naar rol, door gebruik te maken van partities, ook wel zwembanen genoemd (partitions of swimlanes).



In dit diagram is een ordening gemaakt naar “1st Level Support”, “Advanced Support” en “Product Engineering”. Elk van deze drie rollen heeft een eigen zwembaan, waarin de stappen voor die betreffende rol worden gemodelleerd.

Annotaties

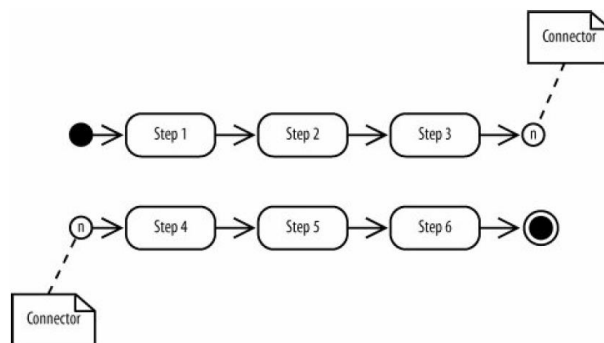
In plaats van partities kan ook gebruik gemaakt worden van annotaties (annotations).



Bij annotaties wordt de naam van de rol tussen haakjes boven de naam van de actie geplaatst.

Connectoren

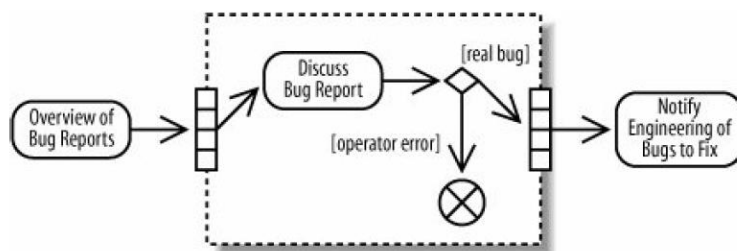
Wanneer een activiteit veel stappen bevat, kan het activiteiten diagram complex worden en daarmee vaak onleesbaar en moeilijk te begrijpen. Het verdient dan aanbeveling om het activiteiten diagram op te splitsen in kleinere diagrammen. Het moet echter wel duidelijk blijven welke stappen op elkaar volgen. Daar kunnen connectoren (connectors) voor worden gebruikt.



In de connectoren wordt een nummer geplaatst. In het hier gegeven voorbeeld diagram is in beide connectoren de letter “n” geplaatst. Er moet in plaats van “n” in beide gevallen “1” worden gebruikt, bij meerdere deeldiagrammen “2”, “3” enzovoorts, maar waar stroom uit stap 3 vervolgd wordt naar stap 4, krijgen beide connectoren hetzelfde nummer.

Uitbreidingsgebied

Het kan zijn dat bepaalde stappen in een activiteit bij herhaling moeten worden uitgevoerd, zoals bijvoorbeeld het doorlopen van een aantal bestanden op zoek naar een bepaald woord. In dergelijk gevallen kan een uitbreidingsgebied (expansion region) worden gebruikt.



Een uitbreidingsgebied wordt weergegeven door een rechthoek met onderbroken lijn. Aan de ingaande en de uitgaande kant van het uitbreidingsgebied worden collecties (collections) geplaatst. Aan de ingaande kant is dat een onbewerkte lijst met items en aan de uitgaande kant een lijst met bewerkte items.

In het hiervoor gegeven activiteiten diagram wordt een lijst met fouten doorlopen. Er wordt bepaald of een in de lijst opgenomen fout een echte fout is of een onjuiste handeling van de gebruiker. De echte fouten worden vervolgens doorgegeven in de uitgaande collectie.

Opdracht 4

Pas dit activiteiten diagram zodanig aan dat aan elke echte fout een prioriteit wordt toegekend: hoge prioriteit, gemiddelde prioriteit of lage prioriteit.