Activity diagram

Bc. et Bc. Martin Melis

Diagram aktivít

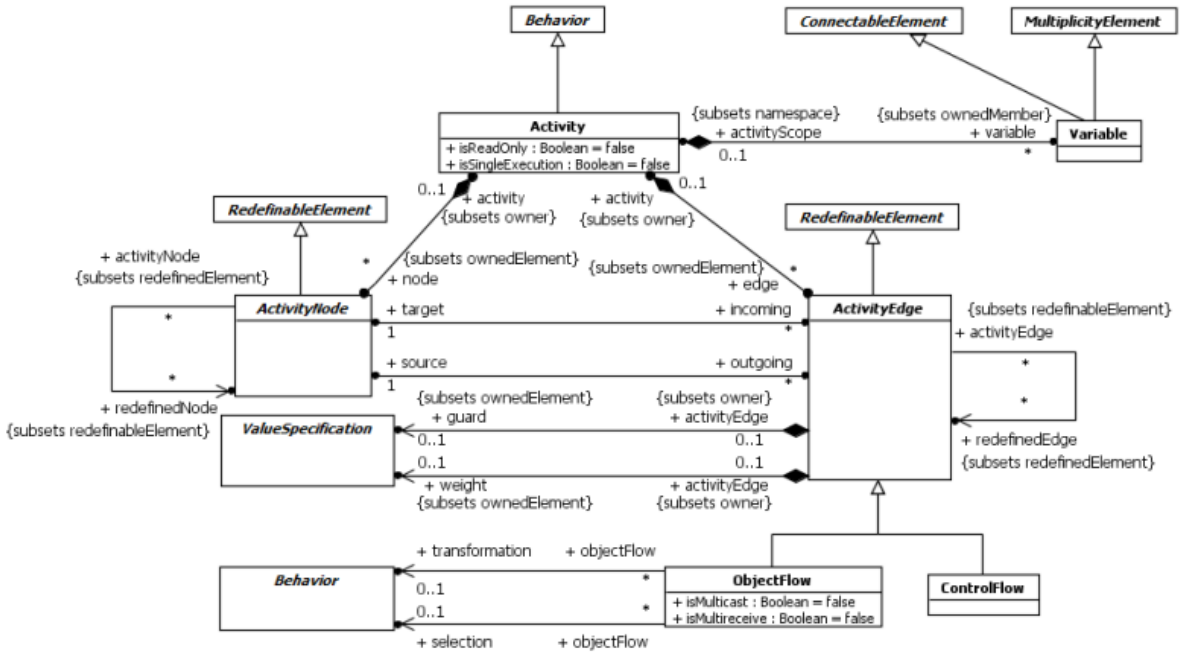
Základným komponentom diagramu aktivít je samotná entita *Activity*, vrámci ktorej modelujeme jej priebeh a tok (pomocou do nej vnorených entít). Každá z aktivít sa podľa metamodelu skladá z 2 základných entít a to hrán (ActivityEdge) a uzlov (ActivityNode).

Uzly delíme do 3 základných skupín:

- Kontrolné uzly (***ControlNodes***), ktoré zabezpečujú vetvenie a riadenie toku
- Objektové uzly (***ObjectNodes***), ktoré definujú manipulované objekty
- Vykonateľné uzly (***ExecutableNodes***), ktoré vykonávajú akciu, resp. manipulujú s objektami

Hrany rozlišujeme:

- Objektovo riadiace (***ObjectFlow***), ktoré znázorňujú pohyb objektov
- Riadiace tok (***ControlFlow***), ktoré znázorňujú logiku toku v diagrame



Základným princípom fungovania diagramu aktivít je monitorovanie toku, ktorý je reprezentovaný fingovaným posúvaním Tokenov medzi jednotlivými uzlami. Tokeny môžu byť objektové, teda de-facto reprezentujúce manipulované objekty (vrátane vlastností objektov), alebo kontrolné, ktorých úlohou je ovplyvňovať činnosť uzlov, no nenesú žiadne informácie a pohybujú sa iba po hranách riadiacich tok.

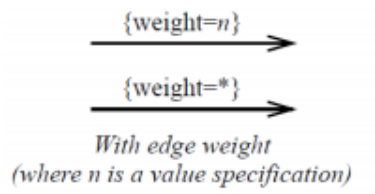
Každá z hrán (ObjectFlow/ControlFlow) môže byť nositeľom podmienky (Guard) v tvare „[need_to_be_met]“, až po ktorej naplnení môže posúvať tok k ďalšiemu z uzlov. Každá z hrán má taktiež vlasnosť *weight*, ktorá špecifikuje minimálny počet tokenov, ktoré musia cez danú hranu paralelne prechádzať. Ak cez danú hranu prechádza menej tokenov, daná hrana ich ďalej neprepustí.

Hrany riadiace tok objektov môžu obsahovať aj 2 základné funkcie a to *transform* a *select.* Transform mení vstupné tokeny na modifikované výstupné pre cieľový ActivityNode. Select aplikuje zvolený filter na vstupné tokeny a len tokeny ktoré prejdú testom, sú následne posunuté cieľovému ActivityNode-u.

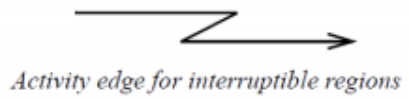
Samotný element *Activity* môže podľa metamodela obsahovať aj premenné, ktoré môžu byť počas vykonávania toku globálne modifikované. Zároveň môže mať aj predpoklady a dôsledky (*pre- and post-condtitions*) pre vykonanie. Aktivita ako behaviorálny element môže mať aj parametre, čo sú vstupné a výstupné štruktúry (reprezentované ako ActivityParameterNodes) očakávané pri inicializácií, resp. finalizácií danej aktivity. Tieto uzly potom posúvajú svoje tokeny ďalej k toku cez hrany. Jednou zo základných funkcií aktivity je funkcia *isSingleExecution()*, ktorá definuje, či počas vykonávania danej aktivity môžu do danej aktivity paralelne vstupovať ďalšie parametrické objekty (teda či môže byť aktivita vyvolaná druhý krát už počas vykonávania, alebo až po vykonaní predchádzajúceho volania).

Grafická notácia

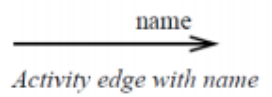
Hrany



Štandardná notácie pre hranu, ktorá ma definovanú *weight*, teda počet tokenov, ktoré môže prenášať.

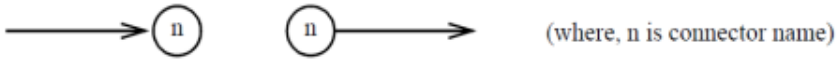


Hrana vyjadrujúca tok, ktorý sa aktivuje, ak dôjde k nekorektnému ukončeniu akcie.



Activity edge with name

Popis a meno hrany sa píšu nad hranu, môže obsahovať aj podmienku na prepustenie toku (v hranatých zátvorkách)

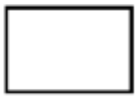


Verzia zápisu hrany cez konektor (len kvôli prehľadnosti)

Uzly


Action node

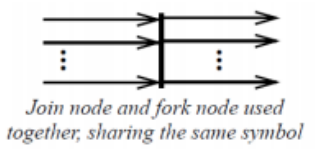
Štandardná akcia. Príklad akcie môže byť *odoslanie zásielky* a pod.


Object node

Objekt môže byť reprezentovaný napríklad triedou z Class diagramu, tu bude z pohľadu viacerých vrstiev v 3D UML treba vytvoriť možné prepojenie medzi rôznymi diagramami. Meno objektu môže byť priamo napísané v objekte, zvykne byť podčiarknuté. Do hranatých zátvoriek je potom možné znázorniť stav objektu.


Merge node and decision node used together, sharing the same symbol

Decision a Merge. Reprezentované budú ako 1 trieda, keďže môže existovať aj skombinovaný objekt podľa obrázka. To či bude objekt fungovať ako Decision, alebo ako Merge bude jasné podľa toho, koľko hrán doňho vstupuje a vystupuje. Merge funguje ako OR, teda prepúšťa tok, ak príde tok z aspoň jednej zo vstupných hrán.



Join node and fork node used together, sharing the same symbol

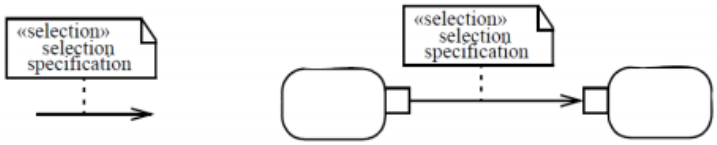
Podobne funguje aj Fork/Join. Fork a Join vyjadrujú paralelné vykonávanie tokov. Join prepúšťa tok, až keď doň vstúpia všetky vstupné hrany. Funguje teda ako AND.

Štart celého toku.

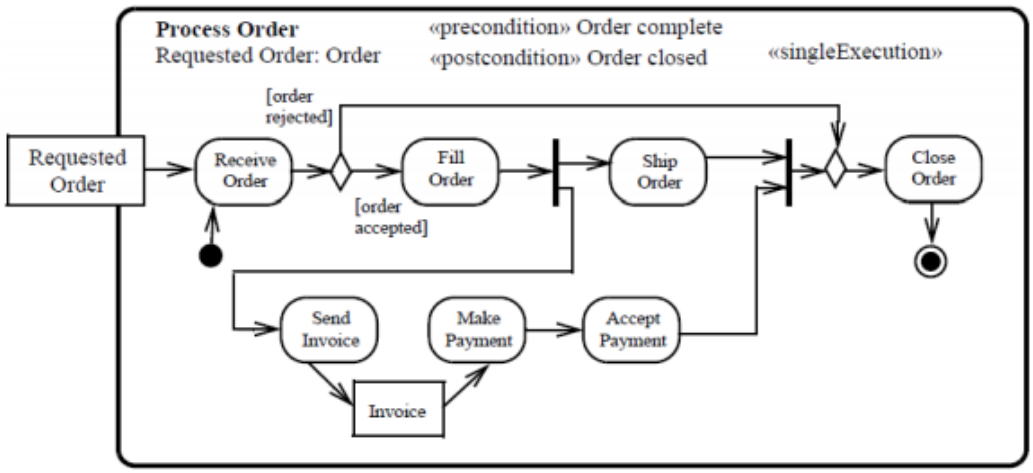
Ukončenie celej aktivity a paralelne aj ukončenie všetkých ostatných tokov.

Ukončenie danej vetvy toku. Ostatné vetvy pokračujú nezávisle.

Konektory (piny) sú iným zápisom pre prepojenie 2 akcií posielaním objektu. Výstupný pin z akcie definuje výstupný objekt a vstupný pin vstupný objekt. Medzi dvoma akciami môže byť vymieňaných aj viac objektov, preto môže byť aj viac paralelných pinových prepojení.

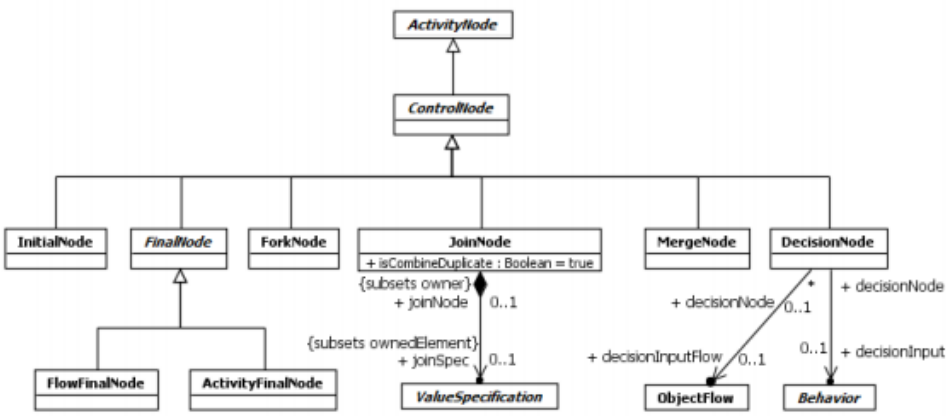


Popis konkrétnej selekcie, alebo transformácie môže byť zaznamenaný v anotácií v príslušnom stereotype.



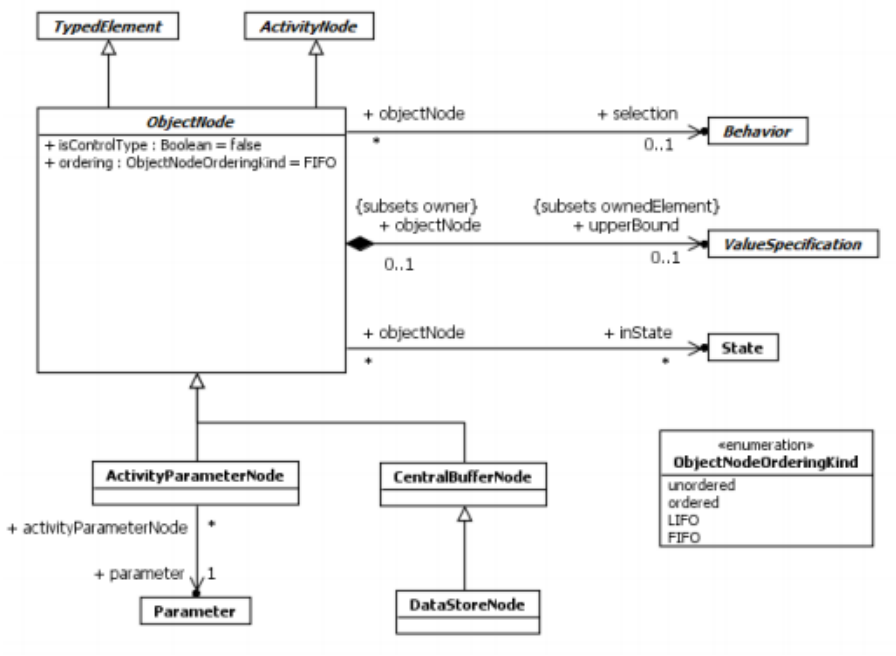
Príklad pre zápis modelovej aktivity.

Metamodel kontrolných uzlov

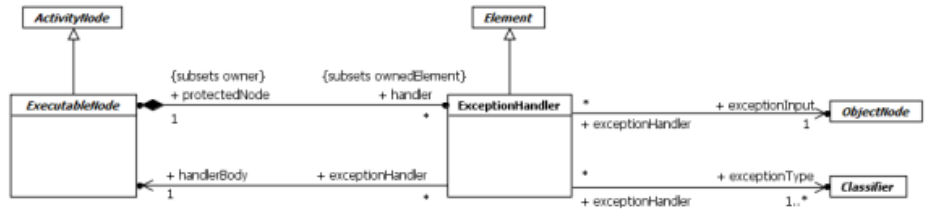


Funkcia isCombinesDuplicate definuje, či ak príde na Join elemtn viacero tokenov s rovnakým obsahom, tak má Join brána posunúť tieto duplikované tokeny ďalej všetky, alebo len 1.

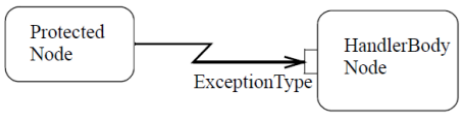
Metamodel objektových uzlov



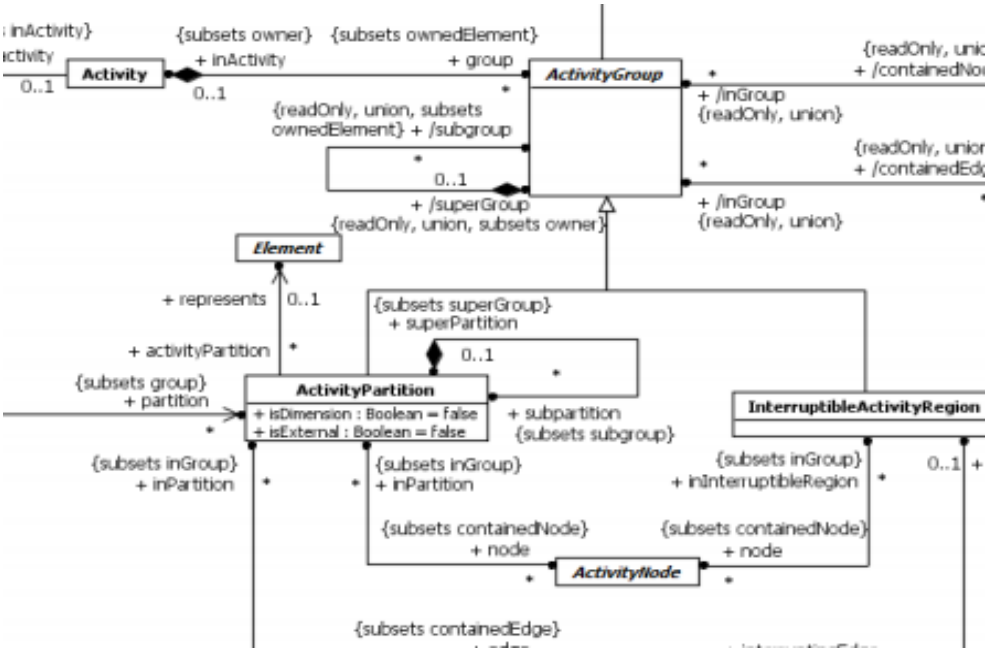
Metamodel výkonateľných uzlov



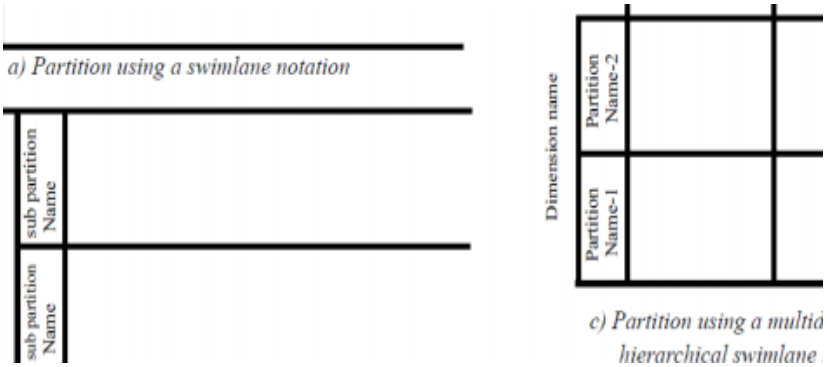
Typickým vykonateľným uzlom je akcia. V tele vykonateľného uzla budú andefinované funkcie ktoré bude vykonávať, pričom ExceptionHandler následne zabezpečuje vykonávanie v prípade ak dôjde k nepredvídanej chybe pri relaizácií pôvodnej funkcie (ExecuteNode vyvolá ExceptionHandler).



Metamodel združovacích entít



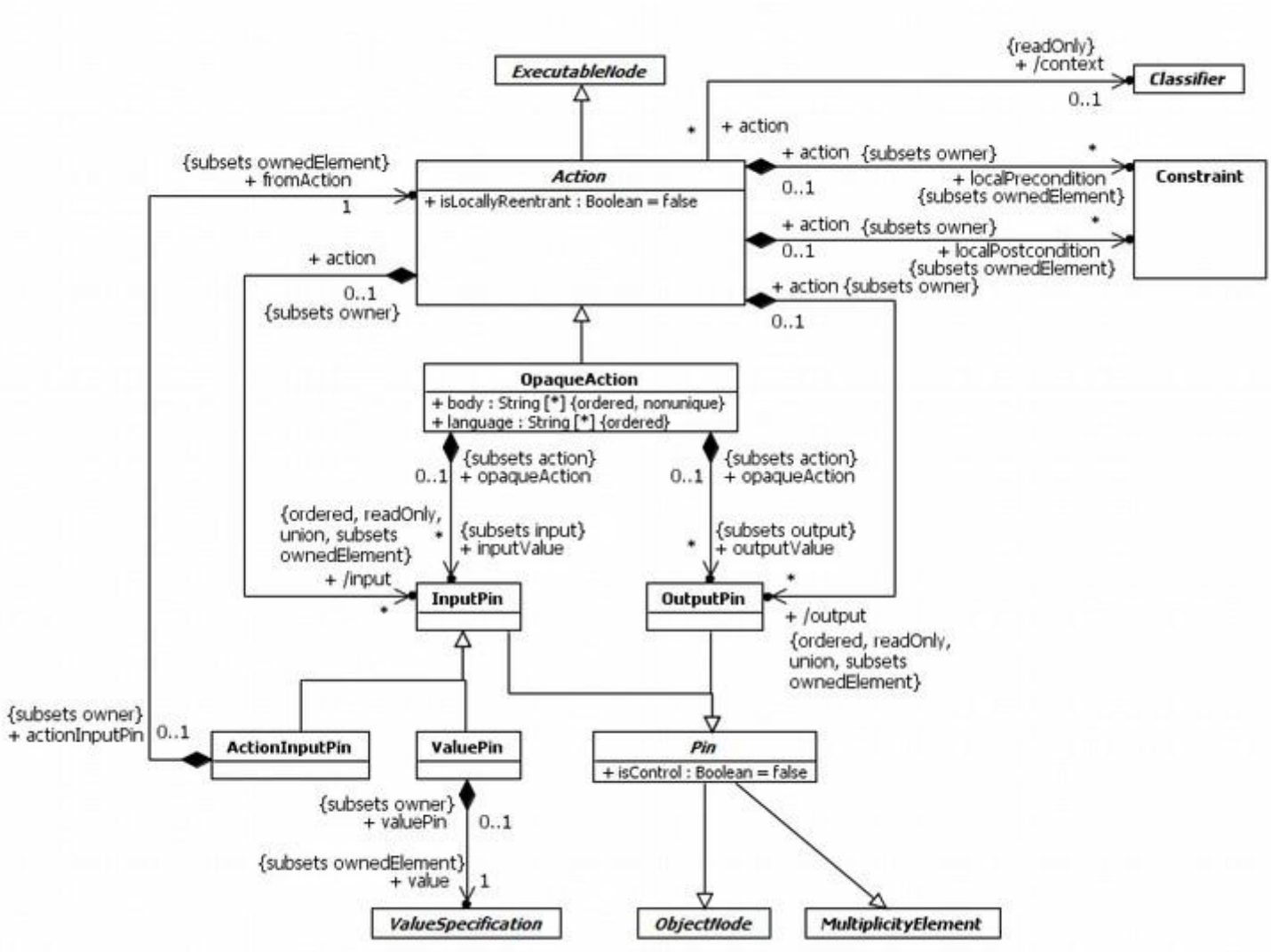
Úlohou združovacích entít (ActivityPartitions) je zoskupovať elementy na základe špecifickej vlastnosti. Vo všeobecnosti obsahujú funkcie isDimension, ktorá ak je nastavená na True hovorí, že daná skupina objektov (Partition) nemôže byť už vrámci danej aktivity vnorená do ďaľšej skupiny objektov. Všetky objekty vrámci danej skupiny musia mať priradený spoločný klasifikačný objekt (Classifier). Funkcia isExternal hovorí o objektoch, resp. subpartíciách, ktoré majú vrámci väčšej skupiny priradený iný klasifikačný objekt, no stále sa podieľajú na výkone danej skupiny akcií (ide o výnimku).



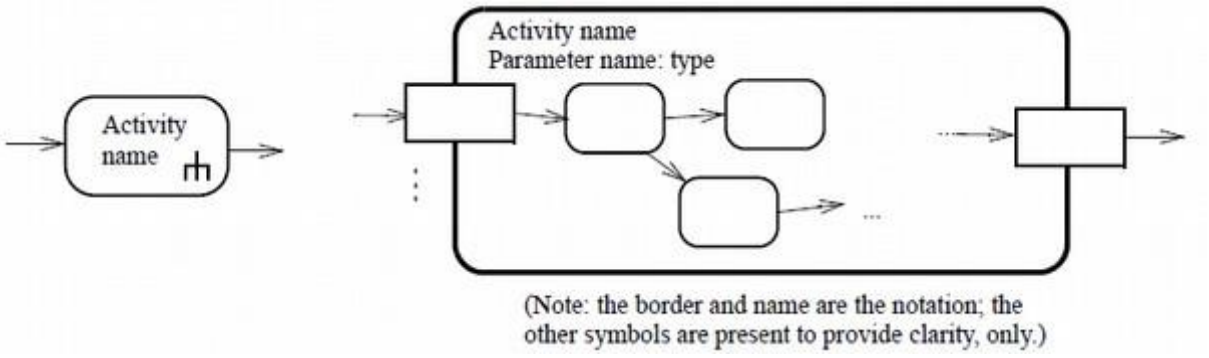
Najčastejšou formou vyjadrenia ActivityPartition je využitie notácie plaveckých dráh. V prípade zavedenia fragmentou zavedieme nové triedy, ktoré budú dediť od triedy ActivityPartition, budú mať vlastnú notáciu typickú pre fragmenty a vlastné vlastnosti.

Ďalšou z častí využívaných pri zoskupovaní sú prerušiteľné skupiny aktivít (*interruptable activity regions*)

Štruktúra akcie



Akciu môžeme vníma všeobecne ako spúšťač. Najčastejšie akcia spúšťa špecifické správanie sa, alebo funkciu, ktorá je v UML reprezentovaná triedou *Behaviour*. V špecifických prípadoch môže akcia spúšťať aj sled akcií v podobe druhej aktivity, vtedy využívame v notácií piktogram trojzubca



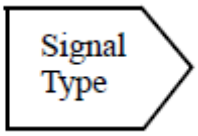
Akcie vo všeobecnosti delíme na 2 typy:

- Akcie ktoré vyvolávajú správanie
- Akcie ktoré preposielajú signály/objekty

Akcie vyvolávajúce správanie môžu vyvolávať priamo objekt reprezentujúci správanie (správanie sa systému), môžu vyslať správu spracúvanému objektu aby vyvolal správanie (systémové), alebo môžu inicializovať priamo správanie sa objektu. Toto budeme realizovať cez overloadig. Volanie môže by synchrónne alebo asynchrónne. Ak je volanie synchrónne, daná akcia bude ukonćená až vtedy, keď bude ukončené aj správanie sa, ktoré inicializovala.

Parameter isLocallyReentrant definuje, či môže by akcia znovu vyvolaná skôr, než sa ukončilo predošlé volanie a vykonávanie sa.

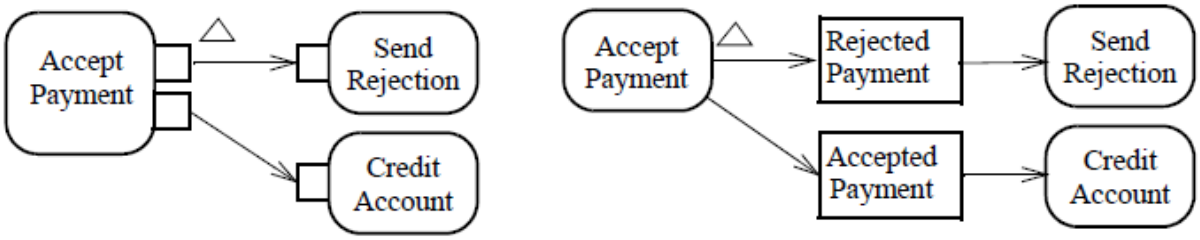
Akcie preposielajúce signály môžu vyslať signál na jeden odchodzí pin, môžu vysla signál na všetky odchodzie piny (broadcast), alebo môžu odoslať na pin špeciálny tip objektu (Ak odošlú objekt typu signál, ide o prvú zo spomenutých možností).



Notácia znázorňujúca akciu, ktorá vysiela objekt typu signál.

Jednotlivé vstupné a výstupné piny môžu by aj zoskupované do takzvaných parametrických skupín. Parametrická skupina vyjadruje, že akcia bude spustená (nad daným objektom) až po prijatí všetkých objektov patriacich do daného set-u a paralelne, objekty budú vyslané naraz, až keď budú pripravené všetky podľa príslušnosti.

V prípade, ak prenášaný objekt je výnimkou (exception) reprezentujúcou spustenie alternatívneho toku, tak v notácií využívame and hranou malý trojuholník.



Štruktúrované akcie

Štruktúrované akcie presne zodpovedajú fragmentom, ktoré je naším cieľom do modelu zaviesť. Špecifikácia vraví, že nie ej definovaný štandardná notaćia pre podmienky, slučky a sekvencie a preto zavedieme notáciu zo sekvenčného diagramu.

Slučka sa ako trieda skladá z 3 zákaldných častí:

- setupPart, kde sa
- test
- bodyPart, ktorý zahŕňa skupinu vykonateľných uzlov.

Pri vstupe do slučky sú automaticky povolené všetky inicializačné uzly (InitialNodes). Vykonateľné uzly sú povolené len v prípade, že sa povolí vstup do bodyPart.

setupPart je prediteračná fáza. Po nej môže nastať priamo fáza bodyPart, alebo test a to podľa toho, či je nastavené isTestFirst. Pri jednotlivých iteráciách si slučka posúva dáta pomocou

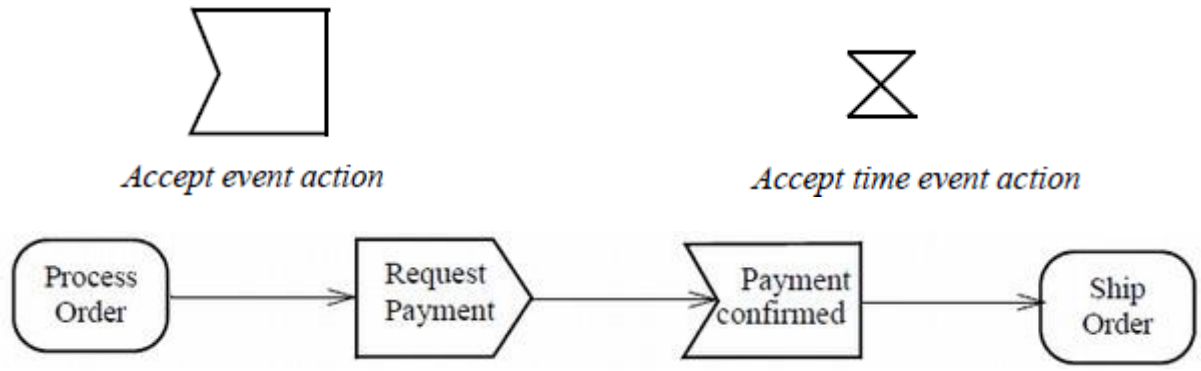
objektov na jednotlivých pinoch (loopVariable outpu pins, bodyOutput output pins, result output pins). Začiatok vykonávani slučky je zabezpečený presunov tokenov z loopVariableInput pinov na loopVariable outpu piny.

Podmienka (ConditionalNode) sa skladá z viac klauzúl, ktoré, reprezentujú jednotlivé vetvy toku. Každá z klauzúl sa skladá zo sekcie bodyPart a test a funguje podobne ako pri slučke.

Niekedy je vhodné, aby dáta spracované v slučke, alebo inej štrukturovanej akcií boli izolované a nedošlo tak k ich modifikácia z externého toku. Na to slúži značka mustIsolate.

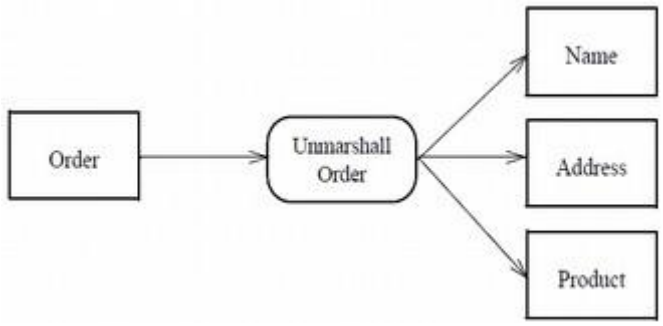
Akcie schvaľujúce udalosti

Takéto akcie vnímame ako spúštače (Triggers) jedného, alebo viacerých udalostí. Nie sú to, ale akcie ako také, sú to len prvky, ktoré vyčkávajú na vykonanie istej udalosti, resp. prijatí signálu a až následne spustia ďalšiu akciu. Sled akcií ktoré tieto spúšťače sledujú sa nachádza v tzv. Event poole.



Presýpacie hodiny znázorňujú časový spúšťač. Príkladom môže by odoslanie zásielky každý mesiac. Časový údaj sa k entite pripisuje ako text. Akcie schvaľujúce udalosti nemusia mať do seba primárne vchádzajúci kontrolný tok, v prípade, ak je ich spustenie inicializované napríklad udalosťou externou k danej aktivite.

Špecifické postavenie maju tzv. rozkaldné akcie (unmarshall actions), ktorých úlohou je z príchodzieho objektu vyparsovať špecifické dáta a podľa nastavených kritérií ich následne postúpiť na odchodzie objekty.


Order
Unmarshall
Order
Name
Address
Product