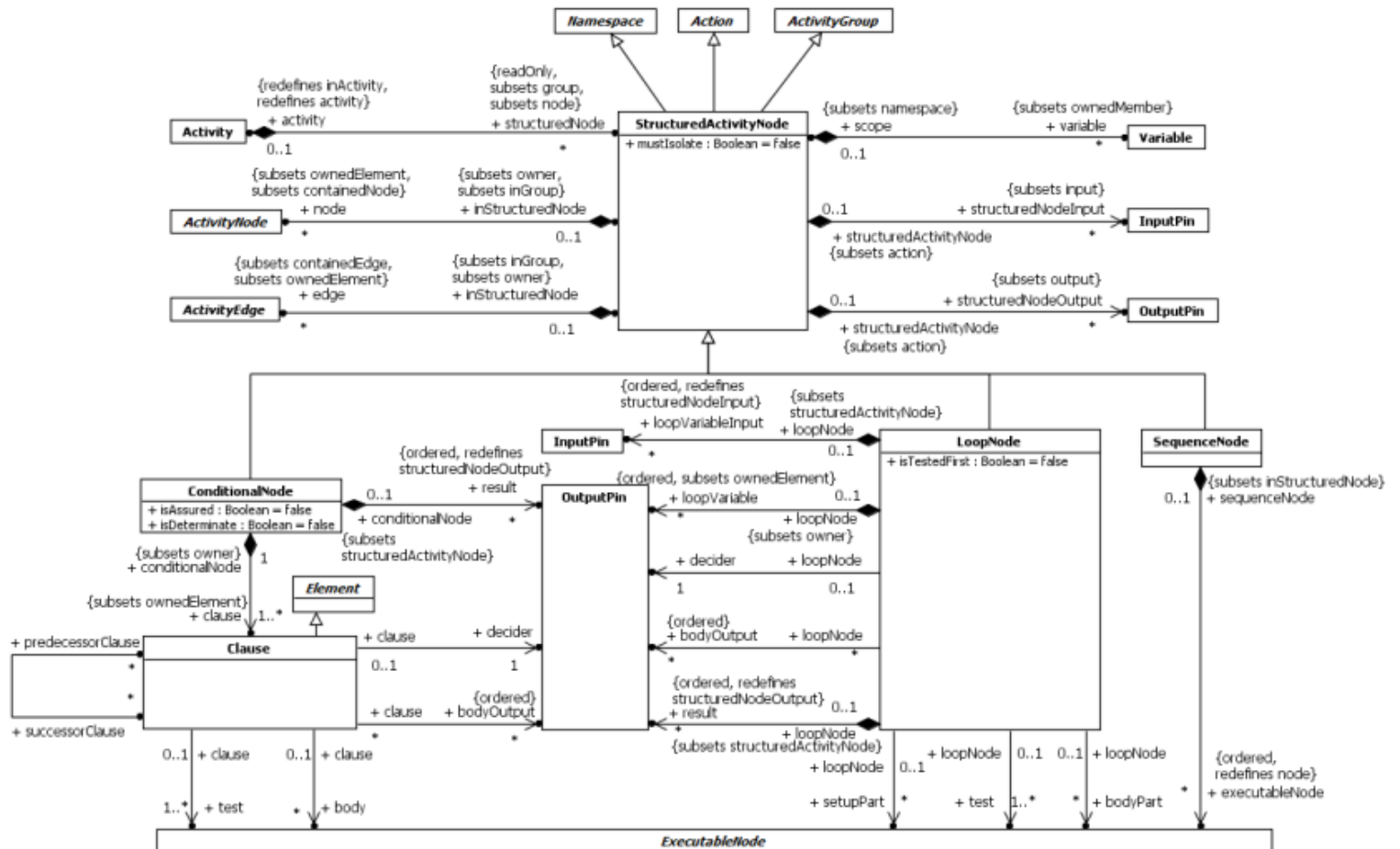


# **Prepojenie diagramu aktivít na fragmenty sekvenčného diagramu**

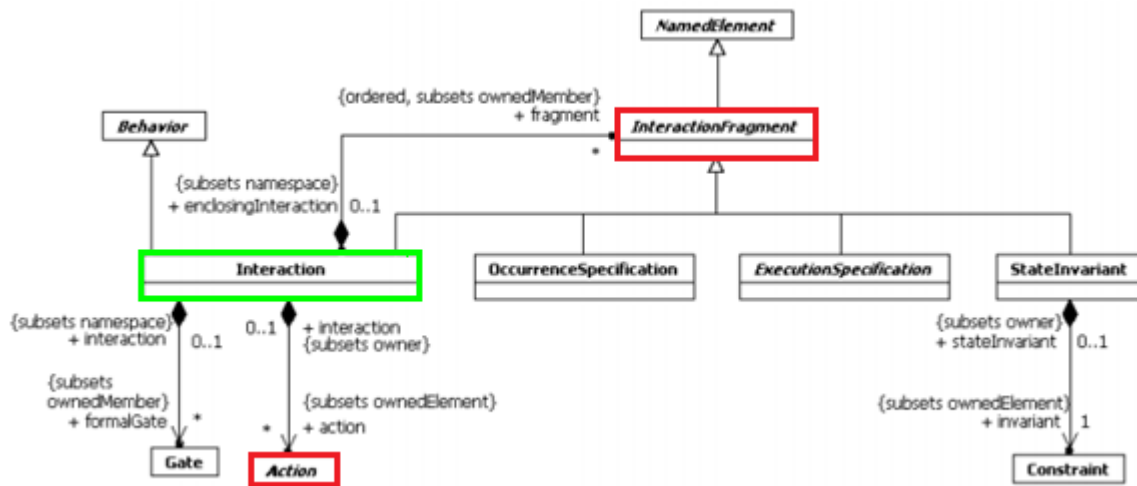
Bc. et Bc. Martin Melis

## Základná myšlienka

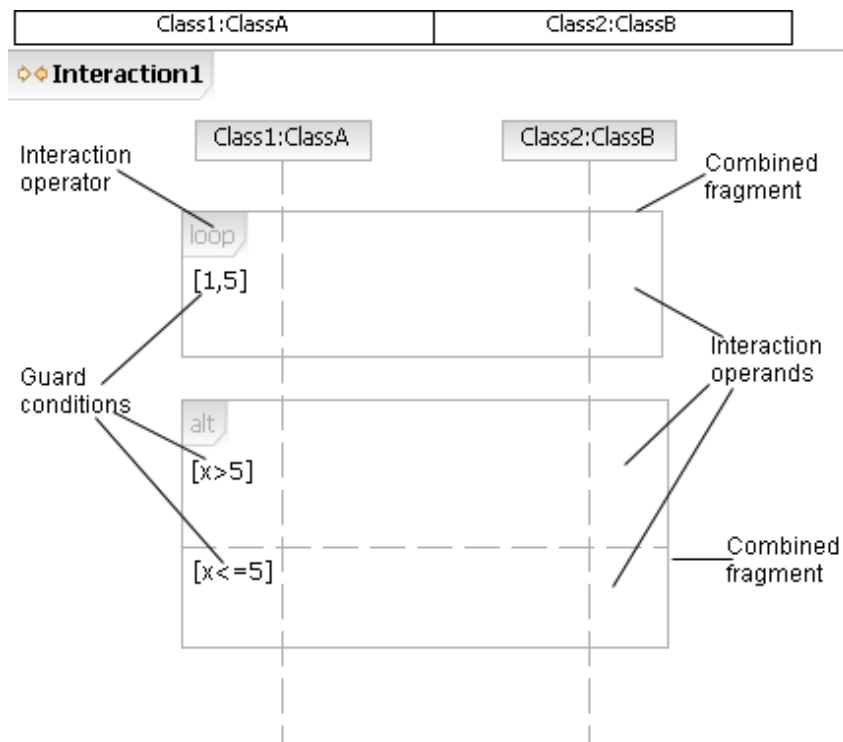
Diagram aktivít je podobne ako sekvenčný diagram, UML diagramom správania sa. Tento diagram samotný už obsahuje podľa špecifikácie štruktúrované entity, ktoré dovoľujú alternatívne vetvenie toku, prípadne tvorbu slučiek, problémom však je, že okrem limitovaného počtu akcií ktoré nám tieto štruktúrované entity poskytujú, nie je podľa metamodelu takto zabezpečené vnáranie.



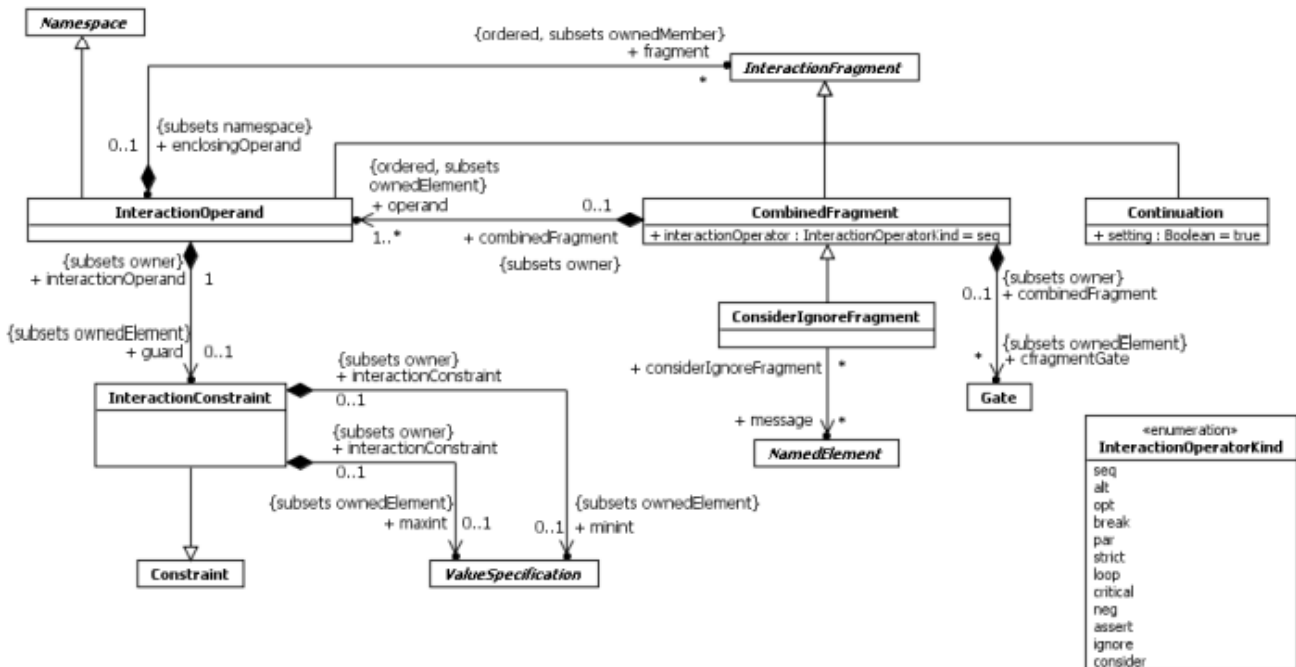
Našou úlohou, keďže chceme zachovať konzistenciu s metamodelom UML, je preto nájsť prepojenie medzi metamodelmi diagramu aktivít a sekvenčným diagramom, konkrétne fragmentami sekvenčného diagramu. Prepájacou entitou, ktorú vieme v tomto prípade identifikovať je entita interakcie. Interakciu vnímame ako špecializáciu triedy správania (Behavior), pričom každá z interakcií môže obsahovať n akcií, ale zároveň aj n interakčných fragmentov (toto zabezpečuje to spomínané prepojenie, ktoré hľadáme). Špecializáciou triedy **InteractionFragment** sú ďalej triedy **InteractionOperand** a **InteractionFragment**. Podľa špecifikácie je každý **InteractionFragment** súčasťou 0-1 **InteractionOperandu** a každý **InteractionOperand** je súčasťou z 0-1 **CombinedFragmentu**. Zjednodušene povedané, **Interaction fragment** vnímame ako región akcií, teda akoby element zoskupujúci akcie, no



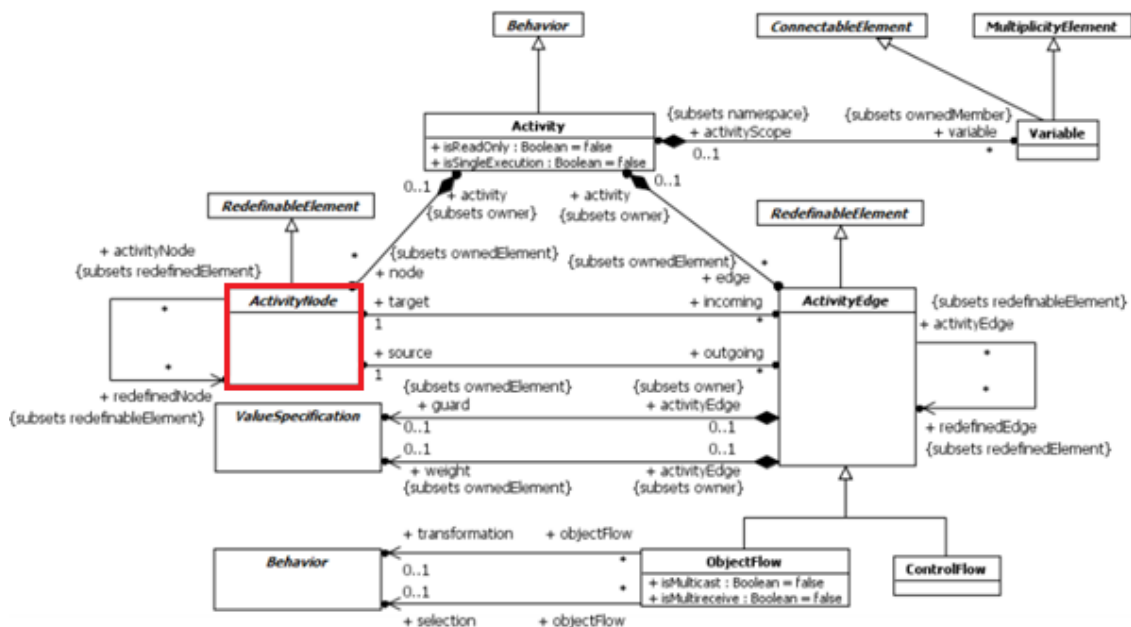
neobsahujúci žiadnu logiku (nevie urobiť loop, alt,...), táto trieda bude len niesť základné informácie o tom, aké elementy zoskupuje a pod. CombinedFragment je potom element, ktorý od neho dedí a teda ho špecifikuje. CombinedFragment, tiež zoskupuje akcie, no na rozdiel od InteractionFragmentu už zahŕňa aj logiku, pretože obsahuje funkciu interactionOperator (pozri obrázok nižšie pre list možných operácií) a zároveň sa skladá z jedného alebo viac InteractionOperandov, teda entít zahŕňajúcich celý tok akcií v danej vetve.



InteractionConstraint je trieda reprezentujúca podmienku vykonania, ktorá je priamo súčasťou zodpovedajúceho InteractionOperanda.

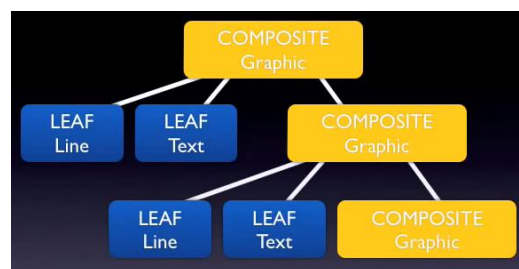
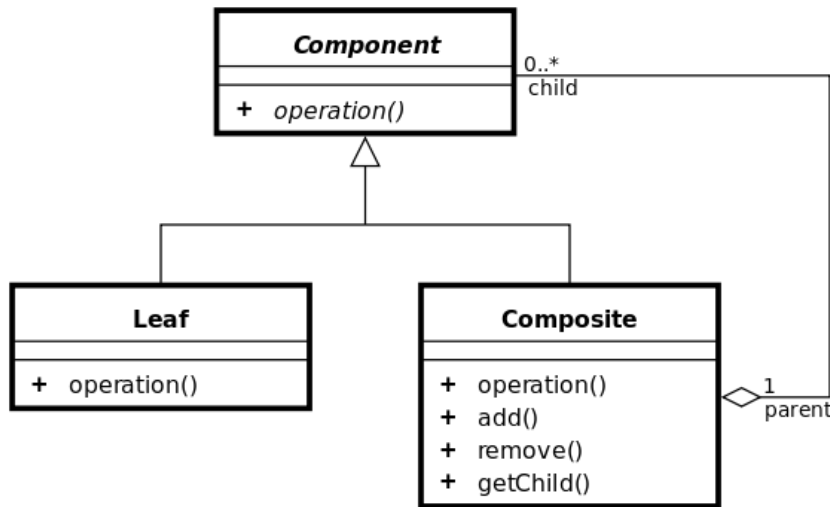


Ak sa pozrieme na metamodel Diagramu aktivít vidíme, že akcia je typu ExecutableNode, čo je podtyp ActivityNode-u. Jeho usporiadanie vrámci diagramu aktivít je nasledovné



Pri vnáraní sa jednotlivých fragmentov do seba využijeme návrhový vzor Composite. Composite funguje tak, že máme 2 typy uzlov (3, ale reálne 2 ktoré dedia od spoločného nadtypu) a to listový uzol (leaf) a composite uzol, čo je to isté, ako listový uzol, ibaže môže mať ďalších potomkov (v grafovej reprezentácii). Keďže môže mať ďalších potomkov, tak

obsahuje aj funkcie na vrátenie zoznamu potomkov, vymazanie konkrétneho potomka a pridanie potomka.



V našom prípade bude nadtriedou, od ktorej budeme dediť (Componentom) modifikovaná trieda InteractionFragment, teda najvšeobecnejšie zoskupenie akcií bez špecifických podmienok na vykonanie. Každý InteractionFragment je spojený s triedou Lifeline, ktorá špecifikuje poradie v akom sú vykonávané jednotlivé entity, ktoré do InteractionFragmentu spadajú. Funkcia execute pre vykonávanie v IF iba spustí vykonávanie a tok do náležitých InteractionOperandov, ktorých súčasťou bude IF (resp. spustí tok danej Lifeline). Compositom bude trieda CombinedFragment. V tomto prípade bude funkcia execute prekonávaná podľa toho, aký interactionOperator bude pridaný za parameter. Loop sa bude napríklad vykonávať prirodzene inak ako Alt. Leaf-om, resp. koncovým uzlom bude trieda InteractionOperand. Tu si treba uvedomiť podstatnú vec a to, že ak si užívateľ zvolí, že nakreslí napríklad fragment alt, systém nemôže vedieť, či je to už fragment finálny a v grafe ho môže reprezentovať ako leaf, alebo, či sa do samotného fragmentu (InteractionOperandu) nepokúsi užívateľ vnoriť ďalší fragment. Pridávanie nových leafov, cez Composite náležiaci danej úrovne zabezpečuje sekvenciu po sebe idúcich fragmentov, nezabezpečuje však vnáranie fragmentov. Pre vnáranie

sa fragmentov do seba, by mal každý z leaf-ov mať možnosť transformovať sa dodatočne na Composite (metamodelovo je to zabezpečené, pretože InteractionOperand môže obsahovať ďalší CombinedFragment).

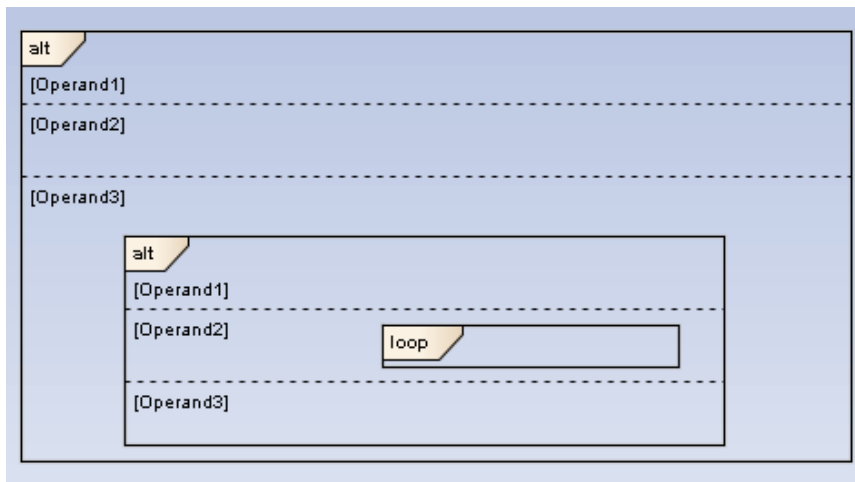
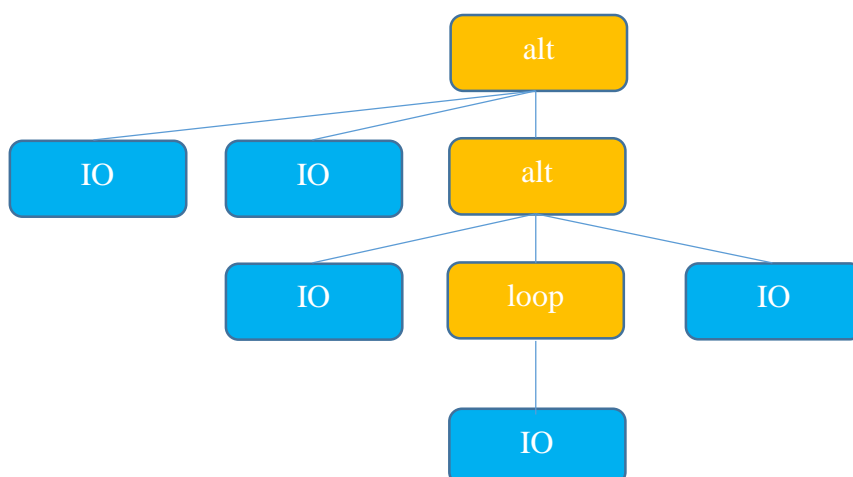
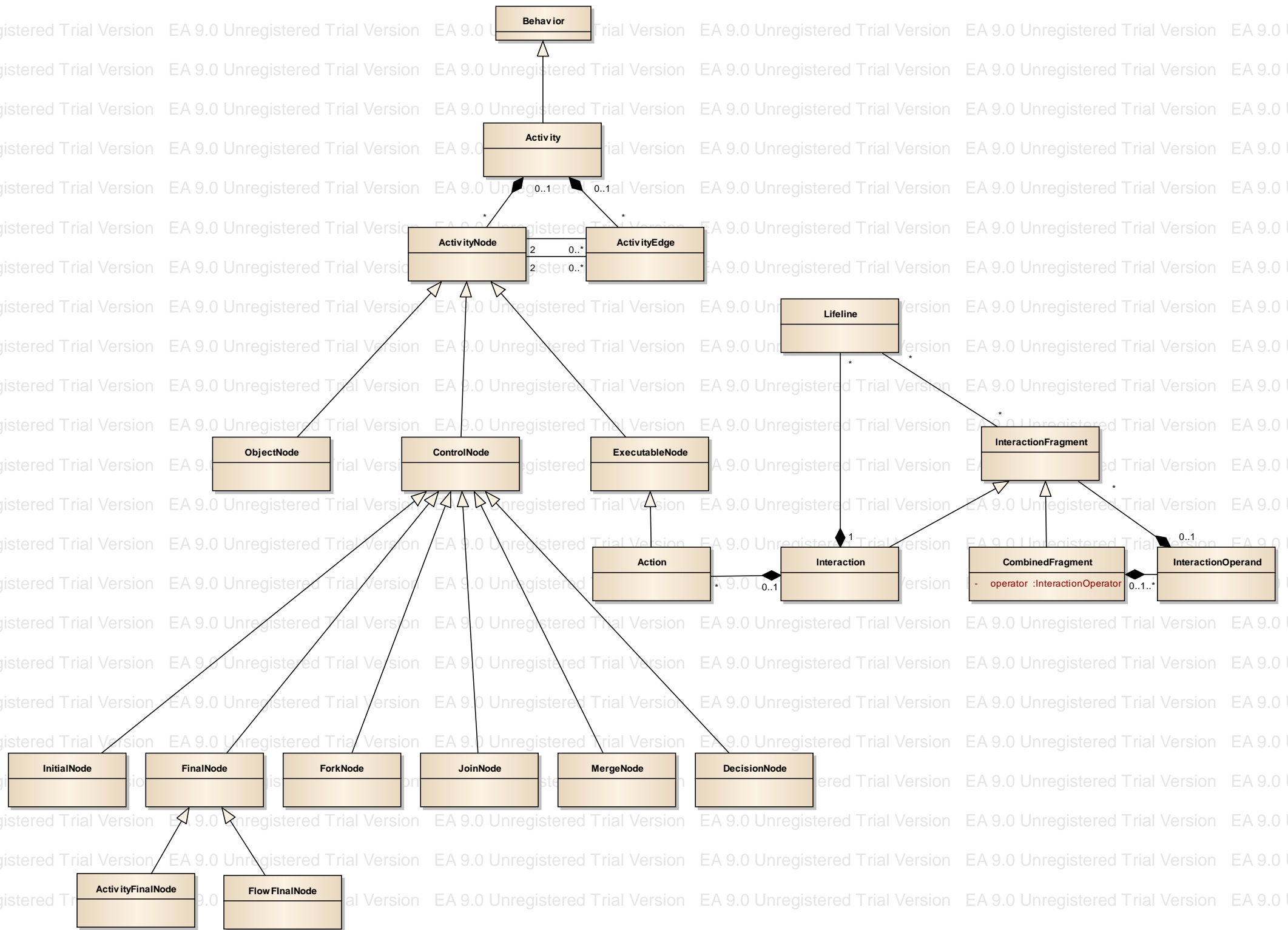


Diagram na hornom obrázku je reprezentovaný grafovou štruktúrou dole. Pri samotnom vykonávaní toku, bude program postupovať tak, že zanalyzuje objekt Composite na najvyššej úrovni (alt) a na základe vstupných parametrov a zvoleného typu execute (ide o alt nie napríklad loop) začne vykonávať akcie v príslušných operandoch. Ak podľa parametra aktivujeme Operand3 tak sa rekurzívne zopakuje proces vnárania. Až po tom, čo bude dovykonávaný posledný zo zvolených fragmentov, bude odoslaný návratový impulz kompozitu o level vyššie etc...Teda inak povedané horný alt sa dovykonáva, až keď dostane správu o dovykonaní vnoreného (teda všetkých vnorených) fragmentov.





Vyššie uvedený model kompletne popisuje základný model prepojenia fragmentov zo sekvenčného diagramu na akcie diagramu aktivít.