

Упражнение 12. Диаграми на „кооперации” (или на „сътрудничество”, т.е „collaboration diagrams”)

Цел: Запознаване с предназначението, графичната нотация и процеса на построяване на UML collaboration диаграми. Collaboration Diagrams в UML 2.0 се наричат вече Communication Diagrams (Диаграми на Комуникации).

Предназначение на диаграмите на кооперации: От гледна точка на аналитика или архитекта на ПС, при проектиране на ПС, често е нужно да се представят структурните взаимовръзки между отделните обекти. Такова представяне на структурата на модела като **съвкупност от взаимодействащи си обекти, които изпращат и получават съобщения** се обезпечава от диаграмите на кооперациите. Диаграмата на кооперации е предназначена за описание на системата на ниво отделни обекти, които обменят съобщения (стимули) помежду си, за да се постигне някаква цел или за да се реализира някакъв случай на употреба. Има пет вида съобщения/стимули: Call, Send, Return, Create, Destroy и Send.

Кооперация (collaboration) — колекция от взаимодействащите си множества от обекти на отделните класове, групирани с цел реализация на отделен вариант на използване в общия контекст на моделираната система. Една и съща съвкупност от обекти може да участва в реализиране на различни кооперации. В зависимост от разглежданата кооперация се изменят както връзките между отделните обекти, така и потока на съобщенията между тях.

Взаимодействието между членовете на една инстанция на кооперация може да се покаже както чрез «collaboration diagram», така и чрез «sequence diagram». StarUML позволява автоматично конвертиране на диаграмата на кооперациите в диаграма на последователностите, и обратното. Collaboration Diagrams акцентират на връзката между обектите, докато диаграмите на последователност се фокусират на времето, в което събитията се случват.

Обектите и тяхното графично изобразяване в диаграмата на кооперации

Обект (object) — същност с добре определени граници и индивидуалност, която капсулира състояние и поведение.

В контекст на езика UML, всеки обект се явява екземпляр на класа, описан в модела и представен на диаграмата на класовете. Обект се създава на етапа на реализация на модела или изпълнение на програмата (тоест, това е run-time елемент). Той има собствено име и конкретни стойности на атрибутите. Следва да видим особеностите на семантиката и графическата нотация на обектите, от които се строят диаграми.

За диаграмите на кооперации, пълното име на обекта представлява текст, разделен с двоеточие и записан във формат:

<собствено име на обекта >'/'<Име на ролята на класа>:<Име на класа >

Името на ролята на класа се указва в този случай, когато съответстващия клас отсъства от модела или пък трябва да се акцентира вниманието на особеностите на неговото използване в контекста на моделиране на взаимодействието.

Името на класа – това е името от диаграмата на класовете.

Важно е да се отбележи, че целия запис на името на обекта се подчертава. Ако е указано собствено име на обекта, то трябва да започва с малка буква. В същото време <собствено име на обекта >'/'<Име на ролята на класа> могат да отсъстват. Двоеточие винаги има пред името на класа, а / - винаги има пред името на ролята.

Тоест, в диаграмите на кооперациите може да имаме следващите варианти на записи на пълното име на обект:



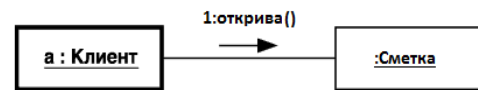
- $o : C$ – обект със собствено име o , екземпляр на класа C .
- $: C$ – анонимен обект, екземпляр на класа C .
- $o:($ или просто $o)$ — обект -сирак със собствено име o .
- $o / R : C$ — обект със собствено име o , екземпляр класа C , играещ роля R .
- $/ R : C$ — анонимен обект, екземпляр на класа C , играещ роля R .
- o / R — обект - сирак със собствено име o , играещ роля R .
- $/ R$ — анонимен обект и едновременно обект - сирак, играещ роля R .

В контекст на езика UML, всички обекти се делят на **активни и пасивни**:

Пасивният обект оперира само с данните и не може да инициира активност по управление на други обекти, но може да изпраща сигнали в процеса на изпълнение на заявки, които обработва. На диаграмата на кооперации, пасивният обект няма указан стереотип.

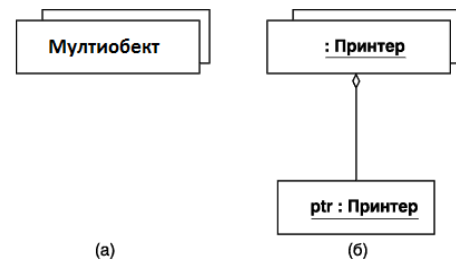
Активен обект (active object) - има собствен процес на управление и може да инициира действия по управление на други обекти, указва се като правоъгълник с удебелени черти.

Показан е фрагмент от диаграма на кооперациите: вижда се как активният обект $a : \text{Клиент}$ се явява инициатор за откриване на сметка, която е представена чрез анонимния обект $: \text{Сметка}$.



Мултиобект (multiobject) - представлява множество от анонимни обекти (тоест такива, които нямат собствено име), образувани са на базата на един клас. На диаграмата на кооперациите се използва мултиобект, за да покажат операции и сигнали, които са адресирани към множество от анонимни обекти.

Ето как се изобразява мултиобект (а) и (б). Отношение на агрегация (композиция) в (б) се отнася за цялото множество от обекти, представяно от мултиобекта (отношение на агрегация между мултиобекта и отделен обект от него:



Съставен обект (composite object) или обект-композит е предназначен за представяне на обект, имащ собствена структура и вътрешни потоци за управление. Този обект е екземпляр на клас - композит, свързан с отношение на композиция със своите части. На диаграмата на кооперациите се изобразява като обичаен обект, състоящ се от 2 секции: горна - със името на обекта и долна - неговите обекти=части вместо списък с атрибути. При което, в качеството на част може да имаме друг съставен обект:

Връзки в диаграмите на кооперациите

В диаграмите на кооперации, отношенията между обектите се описват с помощта на връзки, явяващи се екземпляри на съответстващите асоциации между класовете, на които принадлежат тези обекти.

Съобщения и тяхното графично изобразяване в диаграмите на кооперации

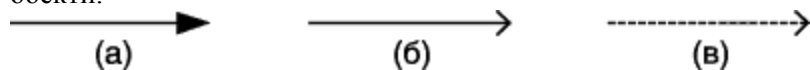
Съобщение (message) — спецификация (описание) на предаването на информация от един елемент на модела към друг, очаквайки приемащия елемент да предприеме определени действия.

В диаграмата на кооперациите, съобщението се явява причина или стимул за началото на изпълнение на някаква операция, на изпращане на сигнали, на създаване или унищожаване на обекти. Връзката между обектите обезпечава канал за насочено предаване на съобщението между обектите, като посоката е от обекта източник (от обекта, който инициира съобщението, понякога се нарича клиент), към обекта получател (поякога се нарича изпълнител на съобщението, също понякога - сървър).

Съобщението специфицира ролите, които играят обектите, съответно – източник и получател на съобщението. Съобщенията в диаграмите на кооперациите се изобразяват с допълнителна стрелка, успоредно със съответстващата връзка или роля на асоциацията, като стрелката сочи от източника към получателя.



В диаграмите на кооперации могат да се използват 3 типа стрелки за обозначаване на съобщения между обекти:



(a) – извикване на процедури (операции) или предаване на потока на управление. Съобщения от този тип могат да бъдат използвани при паралелно работещи обекти, когато един от тези обекти предава съобщение от този тип (извикване на процедури например) и чака, докато не завърши изпълнението на определена последователност от действия от втория обект. Обичайно, такива съобщения са синхронни, тоест инициират се - до завършване на някаква дейност или изпълнение на някакво условие.

(б) - обозначава асинхронно съобщение в прост поток на управление. В този случай, клиентът предава съобщение и продължава своята работа, без да очаква отговор от сървъра.

(в) – обозначава връщане от извикана процедура. Стрелките от този тип често отсъстват в диаграмите на кооперациите (сътрудничество), тъй като се подразбира тяхното съществуване след края на процеса на работа или дейност.

В UML са определени следните стереотипи на съобщения:

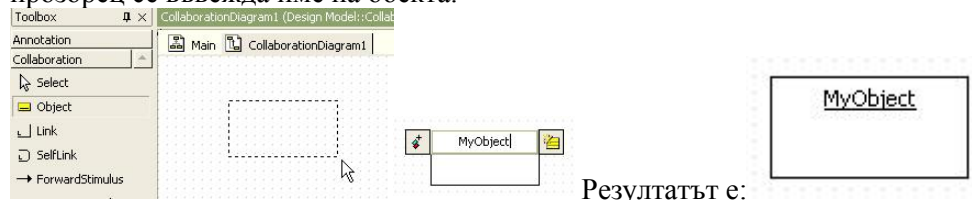
- <<call>> (извиква) – Съобщение, изискващо извикване на операции или процедури на обекта - получател. Ако съобщението с този стереотип е рефлексивно (Call to self – обект извиква себе си), то се иницира локално извикване на операция в обекта, изпращащ това съобщение.
- <<return>> (връща) – Съобщение, връщащо, стойност от изпълнение на операция или процедура към извикващия обект.
- <<create>> (създава) – създаване на обект.
- <<destroy>> (унищожава) – унищожаване на обект.
- <<send>> (предава) – обозначава изпращане на **сигнал** на друг обект, асинхронно. Разликата между **сигнал** и **съобщение** е това, че сигналът трябва явно да е описан в този клас, чийто обект иницира неговото предаване.

Виж. „Програмни спецификации . Ръководство за лабораторни упражнения” тема IX.

Приложение: StarUML и диаграми на кооперации

Създаване на обект:

Избира се [Toolbox] -> [Collaboration] -> [Object]. Посочва се позицията в [main window]. В диалоговия прозорец се въвежда име на обекта.



Резултатът е:

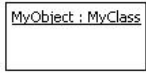
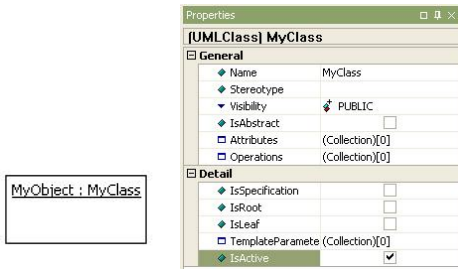
Създаване на изходящ стимул от обект:

Избира се обектът, от който ще излиза стимула и се посочва обекта, в който ще влиза с двоен клик се отваря диалогов прозорец, в който се въвежда след "->"името на стимула (за входящи - "<-", а за изходящи с връщане "<->").

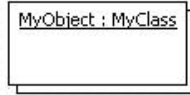


Резултатът е:

Превръщане на обект в активен обект: чеква се характеристиката на обекта [IsActive]. Ако не е присвоено class property, няма да може да се активира обекта.



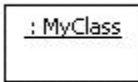
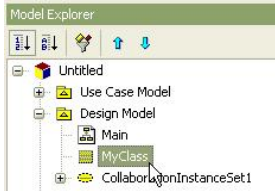
Превръщане на обект в мултиобект: чеква се характеристиката на обекта [IsMultiInstance].



Резултатът е:

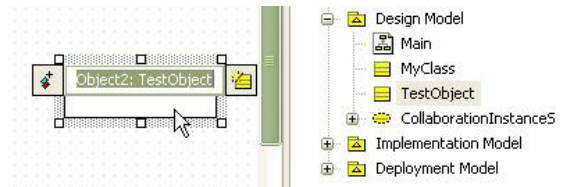
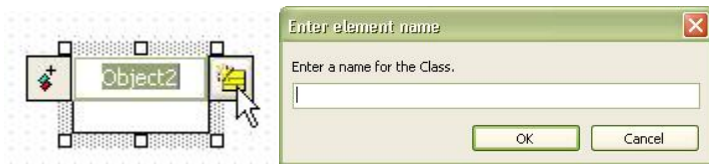
Създаване на обект от клас:

Избира се класа от [model explorer] и се провлачва в collaboration диаграмата:

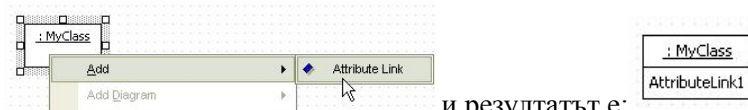


Резултатът е:

Създаване на клас от обект: Ако класът не е присвоен на обект, то се кликва двукратно върху обекта и се въвежда име на класа. Така новия клас е създаден и присвоен на обекта. Може да се присвоява и съществуващ клас чрез добавяне на AttributeLink към обекта.



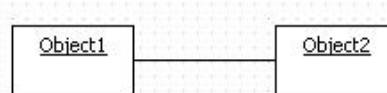
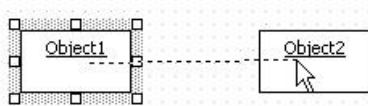
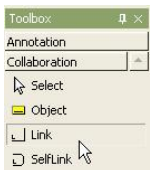
От [Add] -> [Attribute Link].



и резултатът е:

Създаване на връзка/автовръзка:

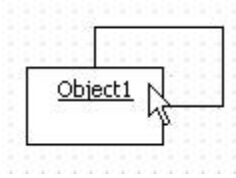
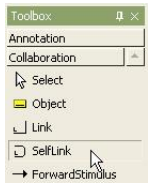
От [Toolbox] -> [Collaboration] -> [Link]. Посочва се обекта-начало и се провлачва до обекта-край в [main window].



Резултатът е:

От [Toolbox] -> [Collaboration] -> [SelfLink]. Посочва се обекта [main window].

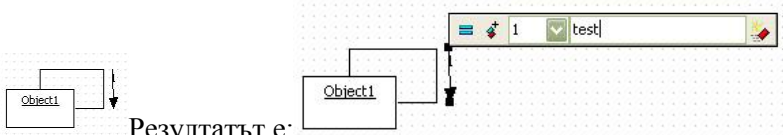
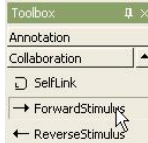
button.



Резултатът е:

Създаване на автостимул:

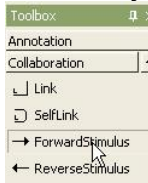
От [Toolbox] -> [Collaboration] -> [ForwardStimulus/ReverseStimulus]. Посочва се автовръзката в [main window] и се въвежда име в диалоговата кутия.



Резултатът е:

Създаване на стимули:

От [Toolbox] -> [Collaboration] -> [ForwardStimulus/ReverseStimulus]. Посочва се връзката, в която ще бъде стимула, в [main window] и се въвежда име в диалоговата кутия.



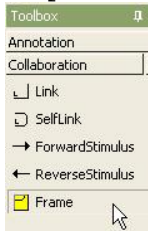
Резултатът е:

Характеристики ActionKind на стимула:

ActionKind	Shape
CALL	→
SEND	→
RETURN	--->
CREATE	<<create>> →
DESTROY	<<destroy>> →

Създаване на frame 생성 방법:

От [Toolbox] -> [Collaboration] -> [Frame]. Посочва се размера в [main window].



Резултатът е:

За да се визуализират поредните номера на стимулите:

