

Дәріс 3 3. UML тіліндегі ұсыныс. Жүйенің пайдаланушылық ұсынысын қалыптастыру. 3.1. UML тіліндегі ұсыныс

UML-де жұмыс жасау барлық уақытта қолданылып отыратын жалпы механизмдердің арқасында оңайланады. Олар: спецификациялар, толықтырулар, белгіленген бөлулер, кеңейту механизмдері.

Толықтырулардың ең маңызды түрі болып ескертулер табылады. Олар шектеулер мен түсіндірулерді көрсету үшін қолданылып, бір элемент немесе олардың жиынтығына тіркеледі. Ескертулер модельге қосымша материалдарды қосу үшін қолданылады. Мысалы: талаптар, бақылаулар және түсіндірулер. UML-дегі кеңейту механизмдері оның мүмкіндіктерін басқарулы түрде кеңейтуге мүмкіндік береді. Олардың ішіне мағыналар және шектеулермен белгіленген стереотиптер кіреді. Стереотиптер UML сөздігін **кеңейтуге**, берілген есепке көбірек сәйкес келетін құрылымдық блоктардың жаңа түрлерін құруға жол береді. Белгіленген шамалар UML құрылымдық блоктарының мүмкіндіктерін кеңейтеді және элементтің **спецификациясына** жаңа мәлімет енгізуге мүмкіндік береді. Шектеулер UML блоктарының қасиеттерін кеңейтеді, олардың көмегімен бар ережелерді өзгертіп, жаңаларын енгізуге болады. Бұл механизмдердің барлығы тілді пәндік кеңістіктің талаптарына және қолданушының таңдаған проекттік әдістемесіне сәйкес баптауға қолданылады.

Модельдеу-мәлімет алмасу тәсілі. UML-де әр түрлі бағдарламалық жүйелердің визуализациясын, специфицирленуін, құрылуын және құжаттау артефакттерін көрсету үшін қажетті барлық құралдар бар, бірақ кей жағдайларда оның мүмкіндіктерін дамыту және кеңейту қажеттілігі туады.

Ескертулер құрылған модельді түсіндіру үшін ескертулер мен түсіндірулер беруге мүмкіндік береді. Олар бағдарламалық қамтаманы жасаудың өмірлік циклінде маңызды рөл атқаратын артефакттарды, мысалы талаптар, бақылаулар, көрсете алады.

Сонымен, UML –де кездесетін 9 типті диаграмма бар:

- класстар диаграммасы;
- объектілер диаграммасы;
- прецеденттер диаграммасы;
- тізбекті диаграмма;
- кооперация диаграммасы;
- күйлер диаграммасы;
- әрекеттер диаграммасы;
- компоненттер диаграммасы;
- ашып қарау диаграммасы.

Класстар диаграммасында класстар, интерфейстер, объектілер мен кооперациялар және олардың арасындағы қатынастар көрсетіледі. Объектілі – бағытталған жүйені моделдеу кезінде бұл диаграмма түрі көп пайдаланылады. Класстар диаграммасы жобалау жағынан жүйенің статикалық түріне сәйкес

келеді. Активті класстардан тұратын, класстар диаграммасы, процесстер жағынан жүйенің статикалық түріне сәйкестендірілген.

Объектілер диграммасында объектілер және олардың арасындағы қатынастар көрсетіледі. Олар класстар диаграммасында көрсетілген мәндердің статикалық экземпляры "фотосы" болып табылады. Объектілер диаграммасы, класстар диаграммасы сияқты, жобалаушы және процесстер көзқарасымен жүйенің статикалық түріне жатады, бірақ макеттік тарату және нақты есептеумен бірге.

Прецеденттер диаграммасында прецеденттер және актерлар (класстардың жекеше түрлері), және олардың арасындағы қатынастар көрсетілген. Прецеденттер диаграммасы қолданушы прецеденттер көзқарасымен жүйенің статикалық түріне жатады. Олар негізінен жүйенің орындалу тәртібін моделдеу және ұйымдастыру кезінде ерекше керек.

Тізбекті диаграмма және кооперация диграммасы бұлар әрекеттесу диаграммаларының жекеше жағдайларында қолданылатын диаграммалар болып табылады.

Әрекеттер диаграммасы объектілер арасындағы байланыстар көрсетіледі; соның ішінде объектілер хабарласа алатын хабарламалар. Әрекеттер диаграммасы жүйенің динамикалық түріне жатады. Бұл кезде тізбекті диаграмма хабарламалардың уақытша тізбектеп орындалуын көрсетеді, ал кооперация диаграммасы – хабар алмасатын объектілердің құрылымдық ұйымдастырылуын көрсетеді. Бұл диаграммалар изоморфты деп аталады, яғни олар бір – біріне ауыса алады.

Күйлер диаграммасында (Statechart diagrams) әрекет түрі, уақиғасы, өтуі және күйі бар автоматты көрсетіледі. Бұл диграммада жүйенің динамикалық түріне жатады. Олар негізінен интерфейс, класс және кооперация тәртібін моделдеу кезінде маңызды. Олар объектінің жұмыс істеу тәртібін уақиғаларға байланысты орындауды қамтамасыздандырады.

Қызмет диаграммасы – бұл күйлер диаграммасының жекеше түрі; олардың арасында жүйе ішіндегі бір басқару ағымынан (деятельностан) екінші басқару ағымына (деятельностқа) өту тәртіптері көрсетіледі. Бұл диграммада жүйенің динамикалық түріне жатады;

Компоненттер диаграммасында компоненттер жиынын ұйымдастыру және олардың арасындағы қатынастар көрсетілген. Компоненттер диаграммасы тарату жағынан жүйенің динамикалық түріне жатады.

Ашып қарау диаграммасында жүйенің түйіндерін өңдейтін конфигурациямен және оларға орналастырылған компоненттермен көрсетіледі. Ашып қарау диаграммасы ашып қарау көзқарасы жағынан жүйе архитектурасының статикалық түріне жатады. Лар компоненттер диаграммасымен байланысқан, өйткені бір түйінге бір немесе бірнеше компонент орналаса алады.

Бұл жерде UML – де қолданылатын барлық диаграммалар берілмеген. Инструментальды құралдармен бұдан басқа да диаграммаларды салуға болады, бірақ олардың осы көрсетілген 9 маңызды және практикада көп қолданылатыны деп есептеледі.

UML тілінің тәртіптері

UML тілінің құрылыс блоктарын бір – бірімен өзбетінше біріктіруге болмайды. Сондықтан UML тәртіптер тізімінен тұрады, ол дұрыс анықталған, яғни семантикалық түрде басқа онымен байланысқан моделдермен сәйкестендірілген болуы керек.

UML семантикалық тәртіптердің келесі түрлері бар:

- *аттары*, мәндерге, қатынастарға және диаграммаларға беріледі;
- *әрекет жасау аймағы* (атқа берілетін кейбір мәндер, контекст);
- *көрнекілік* (аттардың көрініп тұруы және басқа элементтермен қолданылуы);
- *толықтылығы* (элементтердің бір-бірімен ақылдасып орындалу тәртібі);
- *орындалу* (орындау немесе кейбір динамикалық моделді импорттау).

UML тілінің жалпы механизмдері

Құрылыс тиімді және түсінікті түрде жүреді, егер кейбір келісушіліктер орындалса. Ал ол жұмысты келесі механизмдер арқылы жеңілдетуге болады:

- спецификациялар (спецификации) (Specifications);
- толықтырулар (дополнения) (Adornments);
- қабылданған бөлулер (принятые деления) (Common divisions);
- кеңейту механизмдері (механизмы расширения) (Extensibility mechanisms).

UML нотациясындағы күрделі жүйенің интегрирленген моделі төмендегі 3.1 суретте көрсетілген.



Сурет 3.1 - UML нотациясындағы күрделі жүйенің интегрирленген моделі

3.2. Қолдану вариант диаграммасы

Қолдану вариант диаграммасында (use case diagram) прецеденттер және актерлар (класстардың жекеше түрлері), және олардың арасындағы қатынастар көрсетілген. Қолдану вариант диаграммасы қолданушы прецеденттер көзқарасымен жүйенің статикалық түріне жатады. Олар негізінен жүйенің орындалу тәртібін моделдеу және ұйымдастыру кезінде ерекше керек.

Қолдану вариант диаграммасын құру келесі мақсаттардан тұру керек:

- Жүйені жобалаудың бастапқы кезеңінде модельден етін пәндік саланын шектері мен контекстін анықтау керек.

- Жобаланатын жүйенің функционалдық тәртібіне жалпы талаптарды қалыптастыру қажет.

- Бастапқы концептуалды модельді құру керек

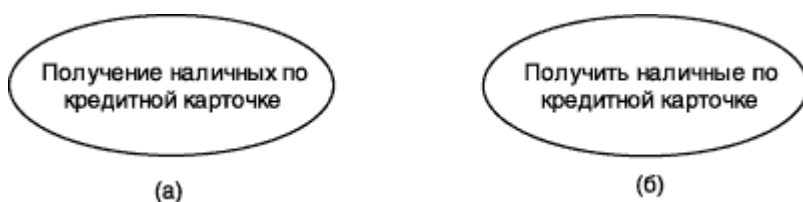
- Жүйе құрушылармен әрекеттесу үшін бастапқы құжаттаманы дайындау

Қолдану вариант диаграммасында негізгі элемент *қолдану вариант және актер*.

Қолдану варианты (use case) — актермен әрекеттесу процесінде жүйенің орындайтын әрекеттерінің тізбегі.

Жеке *қолдану вариант* диаграммада эллипс түрінде беріледі, оның шінде зат есім (сур 3.2.а) немесе етістікпен (сур 3.2б) қысқаша аты жазылады. Қолдану вариант аты бас әріппен басталады

Аты (name) — мәтін жолы, модельдің кез келген элементтің идентификациялау үшін қолданылады.



Сурет 3.2 - Қолдану варианттының графикалық көрсетілімі

Актер (actor) — қолдану варианттының өзара әрекеті кезіндегі сыртқы мәндердің атқаратын рольдері.

Диаграммада актерлердің графикалық белгісі "адам" түріндегі фигура, оның астына актер аты жазылады (сурет 3.3).



Сурет 3.3 Актердің графикалық белгіленуі

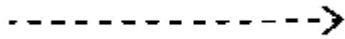
Қолдану вариант диаграммасындағы қатынас

UML тілінде 4 түрлі қатынастар бар:

- Тәуелділік (зависимость);
- Ассоциация (ассоциация);
- Толықтыру (обобщение);
- Тарату (реализация).

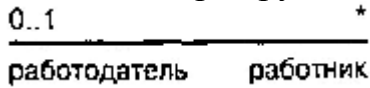
Бұл қатынастар UML –дегі негізгі құрылыс блоктары болып табылады және қалыпты модель құру үшін қолданылады.

Тәуелділік (Dependency) – бұл екі мән арасындағы семантикалық қатынас, бұл кезде екуінің бірі, тәуелді емесі екінші тәуелдінің семантикасын өзгерте алады. Графикалық түрде үзік сызықпен берілген бағдарша түрінде болады, онда таңба орналасу мүмкін.



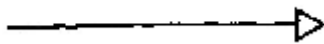
Тәуелділік

Ассоциация (Association) – құрылымдық қатынас, байланыстардың жиының сипаттайды. Байланыс дегеніміз – бұл объектілер арасындағы байланыс. Ассоциацияның басқа түрі бұл агрегация (Aggregation) – ол бүтін және оның бөліктері арасындағы қатынастарды көрсетеді. Графикалық түрде ассоциация түзу сызықпен берілген бағдаршамен беріледі, оның қасынан қосымша белгілер беруге болады, мысалы, қасқартулар немесе роль аттары.



Ассоциация

Толықтыру (Generalization) – бұл қатынас ("специализация/обобщение") «мамандандыру/толықтыру» деп аталады, бұл кезде мамандандырылған элемент объектісі (потомок) оны толықтырған элемент (родителя немесе предка) объектісі ретінде көрсетіледі. Соның арқасында, потомок (Child) өзінің толықтырушысының родителінің (Parent) құрылымын мұрагерленеді. Графикалық түрде сызық және боялмаған бағдарша түрінде беріледі.

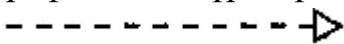


Толықтыру

Тарату (Realization) – бұл классификатор арасындағы семантикалық қатынас, бұл кезед бір классификатор "контракт" анықтайды, ал екіншісі оның орындалуын қамтамасыздандырады. **Тарату** қатынасы екі жағдайда кездеседі:

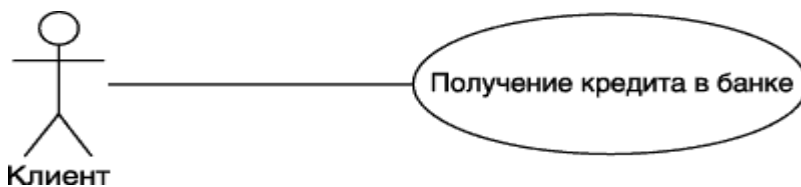
1. интерфейс және оларды таратушы класстар немесе компоненттер арасында;
2. прецедент немесе оны таратушы кооперация арасында

Графикалық түрде үзік сызықты басы боялмаған бағдарша түрінде.



Тарату

Ассоциация қатынасы – UML тіліндегі бір түбір түсінігі, оның көмегімен жүйенің басрлық графикалық модельдері тұрғызылады. Қолдану вариант диаграммасында да және басқа да диаграммаларда **ассоциация** қатынасы түзу сызықпен *актер* мен *қолдану вариантты* арасында беріледі. Бұл сызықтың қосымша белгілеулері болуы мүмкін (сурет 3.4).



Сурет 3.4 - Актер мен қолдану вариант арасындағы ассоциация қатынасының графикалық көрінісінің мысалы

Негізгі әдебиет: [1] – 121б.

Қосымша әдебиет: [2] – 37б.

Бақылау сұрақтары:

1. UML тілі дегеніміз не?
2. Тиянақты ұсыныс шегінде күрделі жүйе моделін иерархиялық тұрғызу принципін қандай деңгейде қарастыруға болады?
3. Диаграмма дегеніміз не?
4. Қандай диаграммаларды білесіз?
5. Қолдану вариант диаграммасы?