

3. Проектування програмного забезпечення засобами UML.

3.2. Діаграма станів

Головне призначення діаграми станів (**statechart diagram**) - описати можливі послідовності станів і переходів, які в сукупності характеризують поведінку системи, що моделюється, протягом усього її життєвого циклу. Діаграма станів представляє динамічну поведінку сутностей, на основі специфікації їхньої реакції на сприйняття деяких конкретних подій. Системи, які реагують на зовнішні дії інших систем або користувачів, іноді називають реактивними. Якщо такі дії ініціюються в довільні випадкові моменти часу, то говорять про асинхронну поведінку моделі.

Діаграми станів найчастіше використовуються для опису поведінки окремих систем і підсистем. Вони також можуть бути застосовані для специфікації функціональності екземплярів окремих класів, тобто для моделювання всіх можливих змін станів конкретних об'єктів. Власне кажучи діаграма станів є графом спеціального виду, що служить для подання кінцевого автомата.

Кінцевий автомат (state machine) - модель для специфікації поведінки об'єкта у формі послідовності його станів, які описують реакцію об'єкта на зовнішні події, виконання об'єктом дій, а також зміну його окремих властивостей.

У контексті мови UML поняття кінцевого автомата має додаткову семантику. Вершинами графу кінцевого автомата є стани й інші типи елементів моделі, які зображуються відповідними графічними символами. Дуги графа служать для позначення переходів зі стану в стан. Кінцевий автомат описує поведінку окремого об'єкта у формі послідовності станів, що охоплюють всі етапи його життєвого циклу, починаючи від створення об'єкта й закінчуючи його знищенням. Кожна діаграма станів являє собою кінцевий автомат.

Розглянемо приклад діаграми станів для об'єкту «двері». Двері протягом свого життєвого циклу можуть знаходитися у трьох станах: «Відчинені», «Зачинені» та «Замкнуті». Стани змінюються як результат реакції на події Відчинити, Зачинити, Відімкнути та Замкнути.

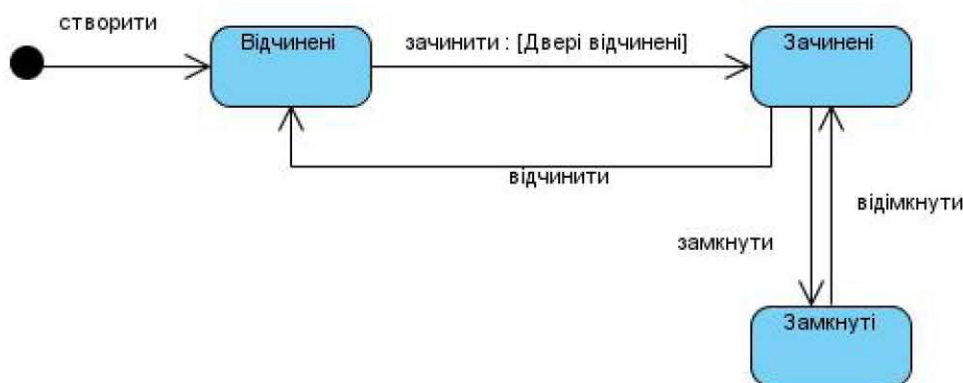


Рис. 3.2.1. Діаграма станів для об'єкту двері

Зауважимо, що не всі події можливі у кожному зі станів, наприклад, якщо двері відчинені неможливо замкнути їх, не зачинивши. Також зазначимо, що перехід із стану

«Відчинені» в стан «Зачинені» характеризується є умовою «чи вільний прохід» і можливий лише при виконанні цієї умови.

Розглянемо детально синтаксис UML діаграм станів.

Моделювання поведінки об'єктів і системи в цілому ґрунтується на понятті стану.

Стан (state) - умова або ситуація в ході життєвого циклу об'єкта, протягом якої він задовольняє логічній умові, виконує певну діяльність або очікує події.

Стан може бути представлено у вигляді набору конкретних значень атрибутів об'єкта деякого класу. Зміна окремих значень цих атрибутів буде відображати зміну стану об'єкта, що моделюється, або системи в цілому. Однак не кожен атрибут класу може характеризувати стан його об'єктів. Зазвичай, мають значення тільки ті властивості елементів системи, які відображають динамічний або функціональний аспект її поведінки. У цьому випадку стан буде характеризуватися деякою інваріантною умовою, що включає в себе тільки принципи для поведінки об'єкта або системи атрибути класів й їхнього значення.

Така умова може відповідати ситуації, коли об'єкт, що моделюється, перебуває в стані очікування виникнення зовнішньої події. У той же час знаходження об'єкта в деякому стані може бути пов'язане з виконанням певних дій. В останньому випадку відповідна діяльність починається в момент переходу елемента, що моделюється, в розглянутий стан, а після елемент може покинути даний стан у момент завершення цієї діяльності.

Стан на діаграмі зображується прямокутником з округленими вершинами (рис. 3.2.2). Цей прямокутник, у свою чергу, може бути розділений на дві секції горизонтальною лінією. Якщо зазначено лише одну секцію, то в ній записується тільки ім'я стану (рис. 3.2.2, зображення ліворуч). В іншому випадку в першій з них записується ім'я стану, а в другий - список певних внутрішніх дій або переходів у даному стані (рис. 3.2.2, зображення праворуч).

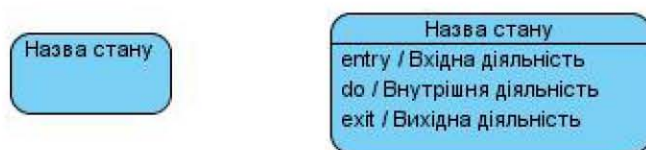


Рис. 3.2.2. Графічне зображення станів на діаграмі станів

Ім'я стану являє собою рядок тексту, що розкриває зміст або семантику даного стану. Оскільки стан системи є частиною процесу її функціонування, рекомендується як ім'я використовувати дієслова в теперішньому часі або відповідні прислівники. У деяких випадках ім'я в стану може бути відсутнім, тобто воно не є обов'язковим. Проте якщо на одній діаграмі кілька анонімних станів, то всі вони повинні розрізнятися між собою.

Внутрішня дія - певна атомарна операція, виконання якої приводить до зміни стану або поверненню *деякого значення*.

Зазвичай дія може бути реалізована за допомогою передачі повідомлення об'єкту, модифікацією зв'язку або значення атрибута.

Кожна дія записується у вигляді окремого рядка й має наступний формат:

Мітка дії вказує на обставини або умови, при яких буде виконуватися діяльність, означена виразом дії. Перелік міток дій у мові UML фіксований, причому ці мітки не можуть бути використані як імена подій:

Вхідна діяльність (entry action) - дія, що виконується в момент переходу в даний стан.

Позначається за допомогою ключового слова - мітки дії entry, що вказує на те, що вираз дії, вказаний за нею, повинен бути виконаним в момент входу в даний стан.

Вихідна діяльність (exit action) - дія, що виконується при виході з даного стану.

Позначається за допомогою ключового слова - мітки дії exit, що вказує на те, що вираз дії, вказаний за нею, повинен бути виконаним в момент виходу з даного стану.

Внутрішня діяльність (do activity) - виконання об'єктом операцій або процедур, які вимагають певного часу.

Позначається за допомогою ключового слова - мітки діяльності do, що специфікує так названу "До-діяльність", виконувану протягом усього часу, поки об'єкт перебуває в даному стані, або доти, поки її не буде перервано зовнішньою подією. При нормальному завершенні внутрішньої діяльності генерується відповідна подія.

Розглянемо як приклад стану аутентифікацію клієнта для доступу до банкомату (рис 3.2.3). Список внутрішніх дій у даному стані може включати наступні дії. Перша дія - вхідна, вона виконується при вході в цей стан і пов'язана з одержанням рядка символів, що відповідають PIN-коду клієнта. Далі виконується діяльність по перевірці введеного клієнтом PIN-коду. При успішному завершенні цієї перевірки виконується дія на виході, що відображає меню доступних для клієнта опцій.

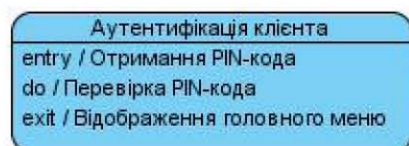


Рис. 3.2.3. Приклад стану з непустою секцією внутрішніх дій

Крім звичайних станів на діаграмі станів можуть розміщатися псевдостани.

Псевдостан (pseudo-state) - вершина в кінцевому автоматі, що має форму стану, але не має поведінки.

Прикладами псевдо станів є початковий і кінцевий стани.

Початковий стан (start state) - різновид псевдостану, що позначає початок виконання процесу зміни станів кінцевого автомата або знаходження об'єкта, що моделюється, в складеному стані.

У цьому стані об'єкт перебуває у початковий момент часу. Він служить для відображення на діаграмі станів графічної області, від якої починається процес зміни станів. Графічно початковий стан позначається у вигляді зафарбованого кружка (рис. 3.2.4), з якого може тільки виходити стрілка-перехід.

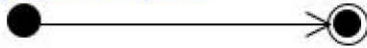


Рис. 3.2.4. Графічне зображення початкового й кінцевого станів на діаграмі станів

Кожна діаграма або під-діаграма станів повинна мати єдиний початковий стан.

Кінцевий стан (final state) - різновид псевдостану, що позначає припинення процесу зміни станів кінцевого автомата або знаходження об'єкта, що моделюється, в складеному стані.

У цьому стані повинен перебувати об'єкт, що моделюється, або система після завершення роботи кінцевого автомата. Він служить для вказівки на діаграмі станів графічної області, у якій завершується процес зміни станів або життєвий цикл даного об'єкта. Графічно кінцевий стан позначається у вигляді зафарбованого кружка, розташованого в колі (рис. 3.2.4), у яку може тільки входити стрілка-перехід.

Наведемо приклад діаграми стану існування об'єкту, що створюється, а потім знищується.

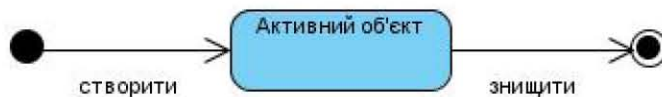


Рис. 3.2.5. Діаграма стану існування об'єкту

Кожна діаграма станів або підстанів може мати кілька кінцевих станів, при цьому всі вони вважаються еквівалентними на одному рівні вкладеності станів.

Перебування об'єкта, що моделюється або системи в першому стані може супроводжуватися виконанням деяких внутрішніх дій. При цьому зміна поточного стану об'єкта буде можлива або після завершення цих дій, або при виникненні деяких зовнішніх подій. В обох випадках говорять, що відбувається перехід об'єкта з одного стану в інший.

Перехід (transition) - відношення між двома станами, що вказує на те, що об'єкт у першому стані повинен виконати певні дії й перейти в другий стан.

Перехід здійснюється, коли настає деяка **подія**: закінчення виконання діяльності (do activity), одержання об'єктом повідомлення або прийом сигналу. На переході вказується ім'я події, а також дії, що виконуються об'єктом при переході з одного стану в інший.

Перехід може відбуватися у той же стан, з якого він виходить. У цьому випадку його називають переходом в себе. Вихідний і цільовий стани переходу в себе збігаються. Цей перехід зображується петлею зі стрілкою й відрізняється від внутрішнього переходу. При переході в себе об'єкт залишає вихідний стан, а потім знову входить у нього. При цьому щоразу виконуються внутрішні дії, специфіковані мітками entry й exit.

Спрацьовування <переходу> (fire) - виконання переходу з одного стану в інше.

Спрацює перехід чи ні залежить не тільки від настання події, але й від виконання певної умови, названої сторожовою ([guard]). Об'єкт перейде з одного стану в інше у тому випадку, якщо відбулася зазначена подія й сторожова умова прийняла значення

 <http://mmsa.kpi.ua>

"істина". До спрацьовування переходу об'єкт перебуває в стані, який будемо називати вихідним, або станом-джерелом (не плутати з початковим станом - це різні поняття), а після його спрацьовування – в цільовому стані.

На діаграмі станів перехід зображується суцільною лінією зі стрілкою, що виходить із вихідного стану й спрямована в цільовий стан. Кожен перехід може бути позначений рядком тексту, що має наступний загальний формат:

```
<ім'я події>' ('<список параметрів, розділених комами>')' '['<сторожова умова>']' <вираз дії>.
```

Подія (event) - специфікація істотних явищ у поведінці системи, які мають місце в часі й просторі.

У мові UML події відіграють роль стимулів, які ініціюють переходи з одних станів в інші. Як події можна розглядати сигнали, виклики, закінчення фіксованих проміжків часу або моменти закінчення виконання певних дій. Наприклад, після успішного ремонту комп'ютера відбувається подія - відновлення його працездатності.

Залежно від виду подій, що відбуваються, розрізняють два типи переходів: тригерні й нетригерні.

Тригерний перехід – перехід, який специфікує подія-тригер, пов'язана із зовнішніми умовами стосовно розглянутого стану.

У цьому випадку поруч зі стрілкою тригерного переходу обов'язково вказується ім'я події у формі рядка тексту, що починається з маленької літери. Приклад тригерного переходу зображено рис. 3.2.6, він є тригерним, оскільки з ним зв'язана конкретна подія-тригер, що відбувається асинхронно при здійсненні клієнтського виклику.

Нетригерний перехід – перехід, який відбувається по завершенню виконання до-діяльності в даному стані.

Для нетригерних переходів поруч зі стрілкою переходу не вказується ніякого імені події, а у вихідному стані повинна бути описана внутрішня до-діяльність, по закінченні якої відбудеться той або інший нетригерний перехід.

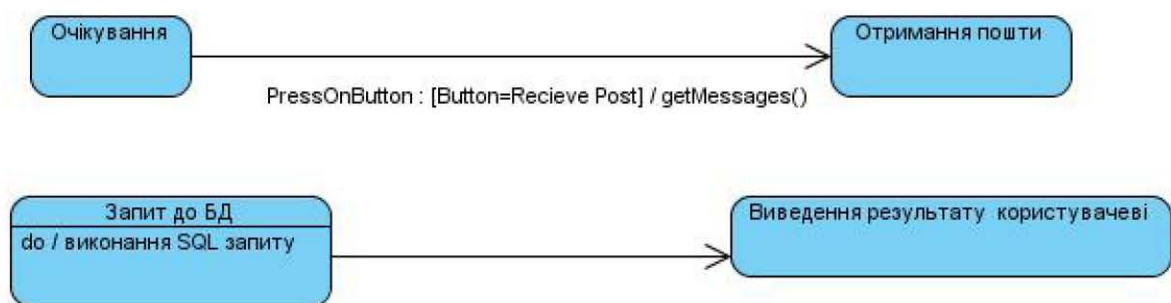


Рис. 3.2.6. Графічне зображення тригерного (згори) та нетригерного (внизу) переходів на діаграмі станів

Сторожова умова (guard condition) - логічна умова, що записується у прямих дужках і є булевим виразом.

Доповнення тригерних і нетригерних переходів сторожовими умовами дозволяє явно специфікувати семантику їх спрацьовування. Якщо сторожова умова приймає значення "істина", то відповідний перехід при настанні події-тригера або завершенні діяльності може спрацювати, у результаті чого об'єкт перейде в цільовий стан. Якщо ж сторожова умова приймає значення "неправда", то перехід не може спрацювати, навіть якщо відбулась подія-тригер або завершилася діяльність у вихідному стані. У випадку невиконання сторожової умови об'єкт або система, що моделюється, залишиться у вихідному стані. Однак обчислення істинності сторожової умови в моделі відбувається тільки після виникнення асоційованого з ним події-тригера або завершення діяльності, яка здійснювалася у стані-джерелі.

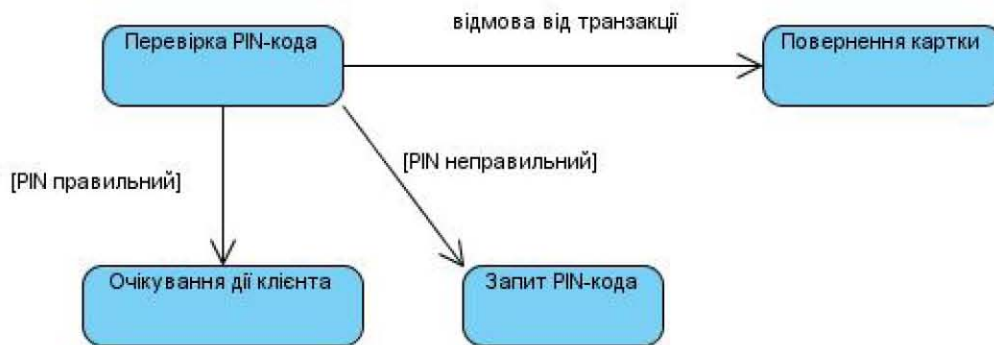


Рис. 3.2.7. Тригерні й нетригерні переходи на діаграмі станів

Зображений фрагмент діаграми станів (рис. 3.2.7) моделює зміну станів банкомата при перевірці PIN-коду. Нетригерні переходи на даній діаграмі позначені сторожовими умовами, які виключають конфлікт між ними. Що стосується тригерного переходу, позначеного подією скасування транзакції, то він відбувається незалежно від перевірки PIN-коду в тому випадку, коли клієнт вирішив відмовитися від введення PIN -коду.

Вираз дії (action expression) - це виклик операції або передача повідомлення, має атомарний характер і виконується відразу після спрацьовування відповідного переходу до початку дій у цільовому стані.

Вираз дії виконується в тому і тільки в тому випадку, коли перехід спрацьовує. Атомарність дії означає, що вона не може бути перервана ніякою іншою дією доти, поки не закінчиться її виконання. Дана дія може впливати як на сам об'єкт, так і на його оточення, якщо це треба з контексту моделі. Даний вираз записується після знака "/" у рядку тексту, приєднаного до відповідного переходу.

Прикладом виразу дії переходу (рис. 3.2.8) може служити відображення повідомлення на екрані банкомату в тому випадку, коли запитувана клієнтом сума перевершує залишок на його рахунку. У випадку якщо коштів вистачає, то відбувається перехід у стан одержання готівки.

Зауваження: отже, подія є приводом для початку переходу, сторожова умова перевіряє можливість його виконання, а дія є наслідком.



Рис. 3.2.8. Вираз дії переходу на діаграмі станів

Самовиклик

Стан може мати перехід, який веде з вихідного стану знову до нього, тобто здійснює само виклик, як це зображено на діаграмі (рис 3.2.9).



Рис. 3.2.9. Стан самовиклику

Моделювання складних об'єктів і систем, як правило, пов'язане з багаторівневим поданням їхніх станів. У цьому випадку виникає необхідність деталізувати окремі стани, зробивши їхніми складовими.

Складений стан (composite state) - складний стан, що складається з інших вкладених у нього станів.

Вкладені стани виступають стосовно складеного стану як підстани (substate). І хоча між ними має місце відношення композиції, графічно всі вершини діаграми, які відповідають вкладеним станам, зображуються усередині символу складеного стану (рис. 3.2.10.).

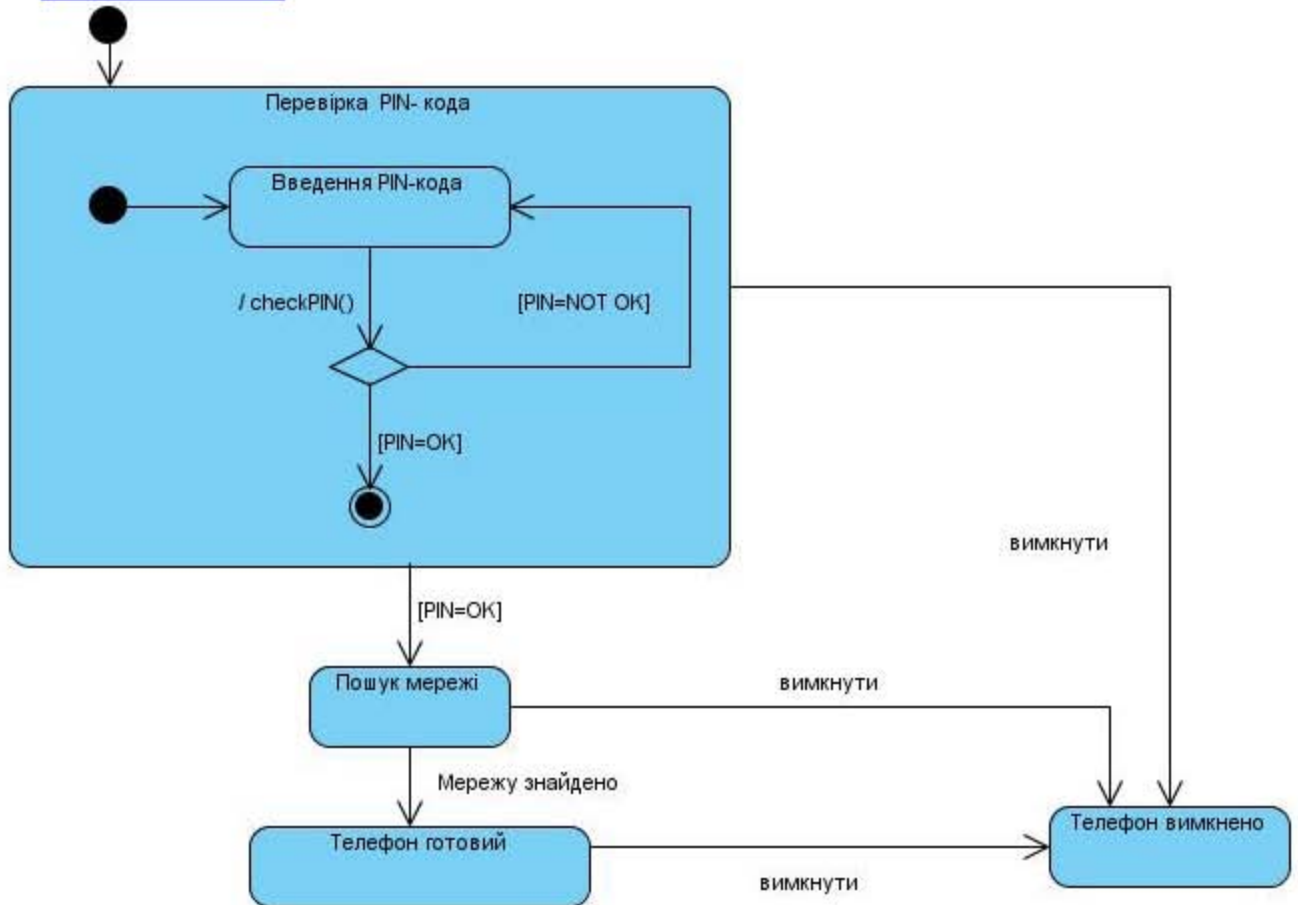


Рис. 3.2.10. Графічне зображення складеного стану

Якщо не потрібна така деталізація складеного стану, то діаграма може бути зображена наступним чином.

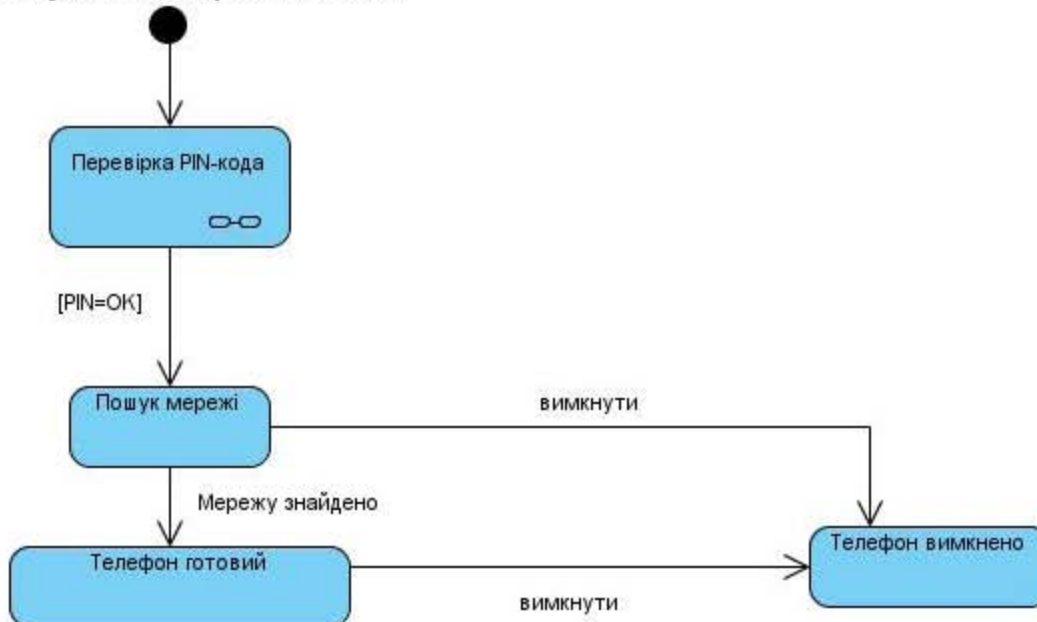


Рис. 3.2.11. Графічне зображення згорнутого складеного стану

Складений стан може містити або декілька послідовних підстанів, або декілька паралельних кінцевих підавтоматів. Кількість рівнів вкладеності складених станів не фіксована.

Послідовні підстани (sequential substates) - вкладені стани стану-композиції, у рамках якого в кожен момент часу об'єкт може перебувати в одному й тільки одному підстані.

Поведінка об'єкта в цьому випадку являє собою послідовну зміну підстанів, від початкового до кінцевого.

Приклад такої поведінки може надати звичайний телефонний апарат. Він може перебувати в різних станах, зокрема в стані додзвону до абонента. Для того щоб подзвонити, необхідно зняти слухавку, почути тоновий сигнал, після чого набрати потрібний телефонний номер. Таким чином, стан додзвону до абонента є складовим і складається із двох послідовних підстанів: Слухавка піднята й Набір телефонного номера (рис. 3.2.12).

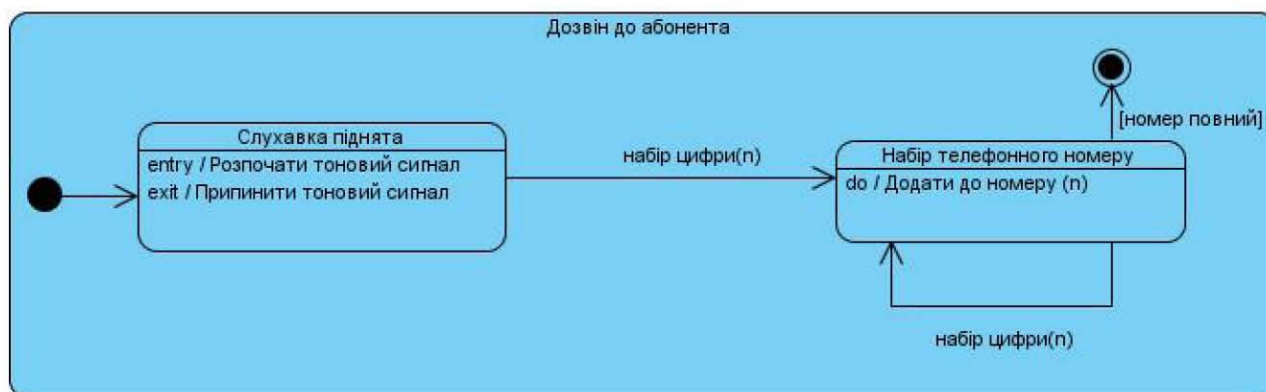


Рис. 3.2.12. Приклад складеного стану із двома вкладеними послідовними підстанами

Переходи можуть потребувати деяких пояснень. Два з них специфікують подію-тригер, що має ім'я: набір цифри(n) з параметром n. Параметром виступає окрема цифра на диску телефонного апарата. Перехід з початкового підстану не містить ніякого рядка тексту. Останній перехід у кінцевий підстан також не має події-тригера, але має сторожову умову, що перевіряє повноту набраного номера абонента. Тільки у випадку істинності цієї умови телефонний апарат може перейти в кінцевий стан для стану-композиції Додзвон до абонента.

Кожен складений стан повинне містити вкладені стани початковий і кінцевий стани.

Паралельні підстани (concurrent substates) - вкладені стани, що використовуються для специфікації двох і більше кінцевих підавтоматів, які можуть виконуватися паралельно усередині складеного стану.

Окремі паралельні підстани можуть, у свою чергу, складатися з декількох послідовних підстанів (рис. 3.2.13). У цьому випадку об'єкт, що моделюється, може перебувати тільки в одному з послідовних підстанів кожного підавтомату. Таким чином, для фрагмента діаграми станів (рис. 3.2.13) припустимо одночасне знаходження об'єкта тільки в наступних підстанах: (А, В, Г), (Б, В, Г), (А, В, Д), (Б, В, Д).

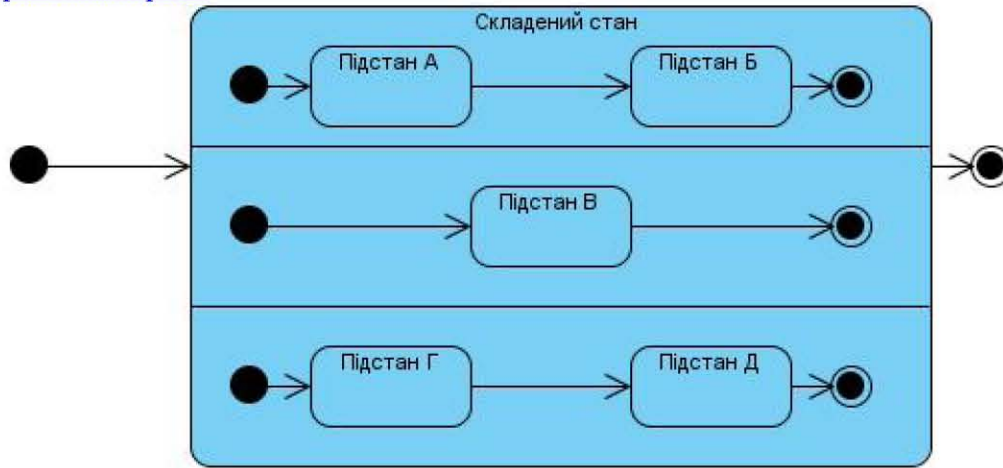


Рис. 3.2.13. Графічне зображення стану-композиції із вкладеними паралельними підстанами

Псевдостан вибору

Псевдостан вибору зображається ромбом, в який входить один перехід, а виходить два або більше. На наступній діаграмі показано, що створення повідомлення здійснюється відповідно до обраного під час знаходження у попередньому стані формату.

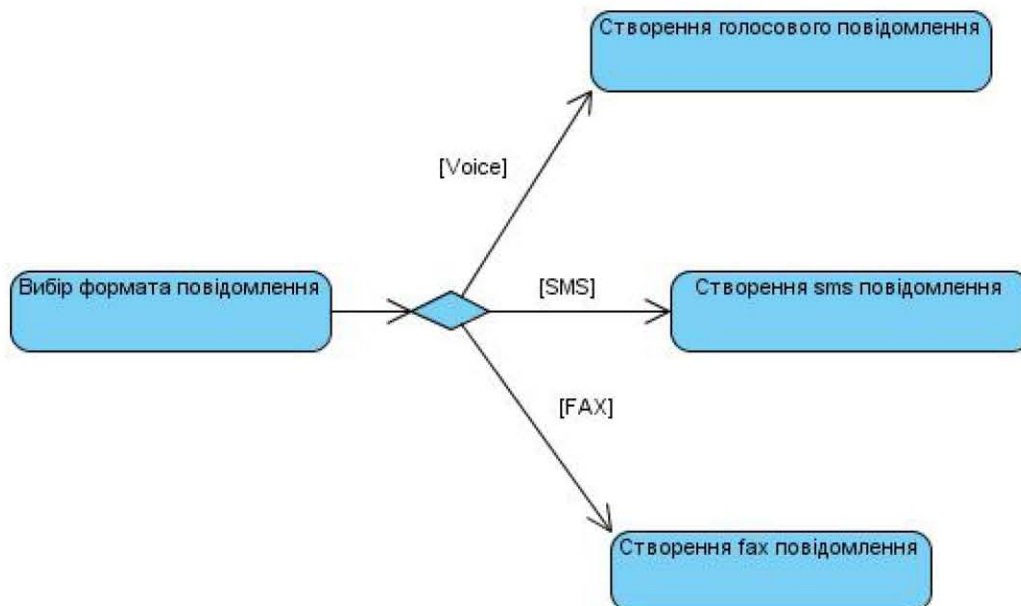


Рис. 3.2.14. Графічне зображення псевдостану вибору

Псевдостан перетину

Псевдо стан перетину використовується для об'єднання декількох переходів. Він може мати один або декілька вхідних переходів та один або декілька вихідних; сторожова умова може накладатися на кожний з них. Стан перетину не має додаткової семантики. На відміну від попереднього стану, який характеризує динамічний вибір, перетин є статичним.

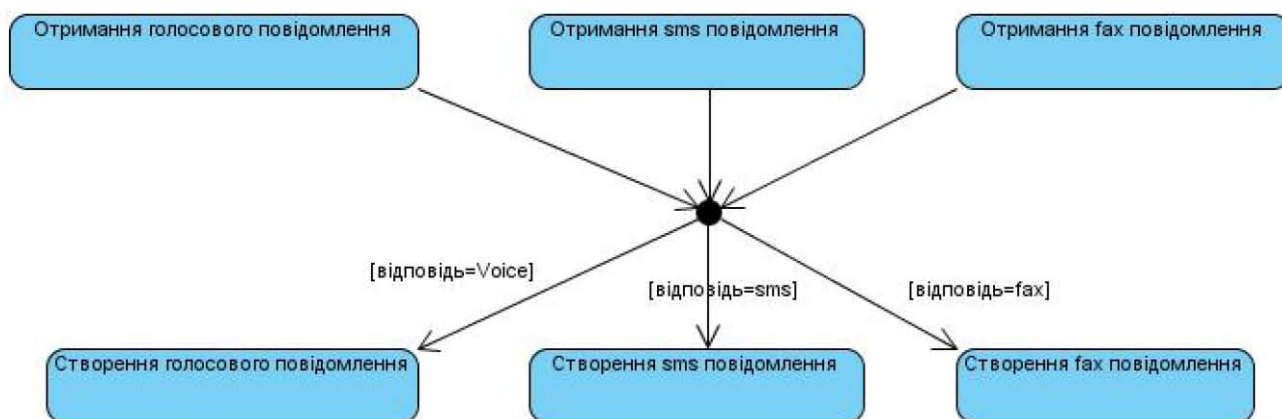


Рис. 3.2.15. Графічне зображення псевдостану перетину

Завершуючий псевдо-стан

У разі необхідності підкреслення того факту, що при виході із певного стану життєвий цикл об'єкту переривається, використовують завершуючий стан. Графічно він зображається як хрест.

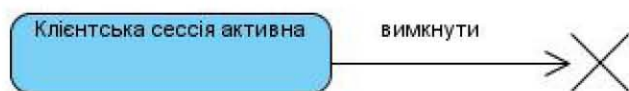


Рис. 3.2.16. Графічне зображення завершуючого псевдостану

Звичайний кінцевий автомат не дозволяє враховувати передісторію в процесі моделювання поведінки систем й об'єктів. Однак функціонування ряду систем засновано на можливості виходу з окремого стану-композиції з наступним поверненням у цей же стан. Може виникнути необхідність урахувати ту частину діяльності, що була виконана на момент виходу із цього стану-композиції, щоб не починати її виконання спочатку. Для цього існує історичний стан.

Історичний стан (history state) - псевдостан, що використовується для запам'ятовування того з послідовних підстанів, що був поточним у момент виходу зі складеного стану.

Історичний стан застосовується тільки в контексті складеного стану. При цьому існує два різновиди історичного стану: неглибокий або недавній й глибокий або давній (рис. 3.2.17).

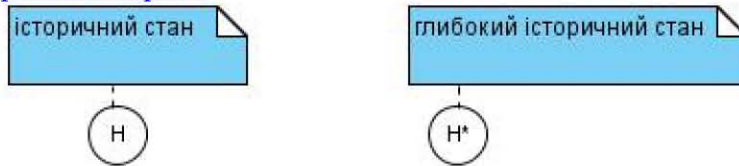


Рис. 3.2.17. Графічне зображення недавнього і давнього історичного стану

Неглибокий історичний стан (shallow history state) позначається у формі невеликого кола, в яке поміщена латинська буква "H" (рис. 3.2.17). Цей стан є першим підстаном у складеному стані, і перехід ззовні в розглянутий складений стан повинен вести безпосередньо в даний історичний стан. При першому переході в неглибокий історичний стан він не зберігає ніякої історії, тобто заміняє собою початковий стан відповідного кінцевого підавтомату.

Історичний стан втрачає свою історію в той момент, коли кінцевий підавтомат доходить до свого кінцевого стану. При цьому неглибокий історичний стан запам'ятовує історію тільки того кінцевого підавтомату, до якого він відноситься. Тобто цей тип псевдостану здатний запам'ятати історію тільки одного з ним рівня вкладеності.

На діаграмі наведено приклад функціонування пральної машини, яка під час своєї активності може перебувати у станах: «прання», «ополіскування», «віджим» (рис. 3.2.18). Якщо її вимкнути під час функціонування, а потім знов ввімкнути, то вона повернеться у той стан, в якому перебувала під час вимкнення і продовжить функціонування.

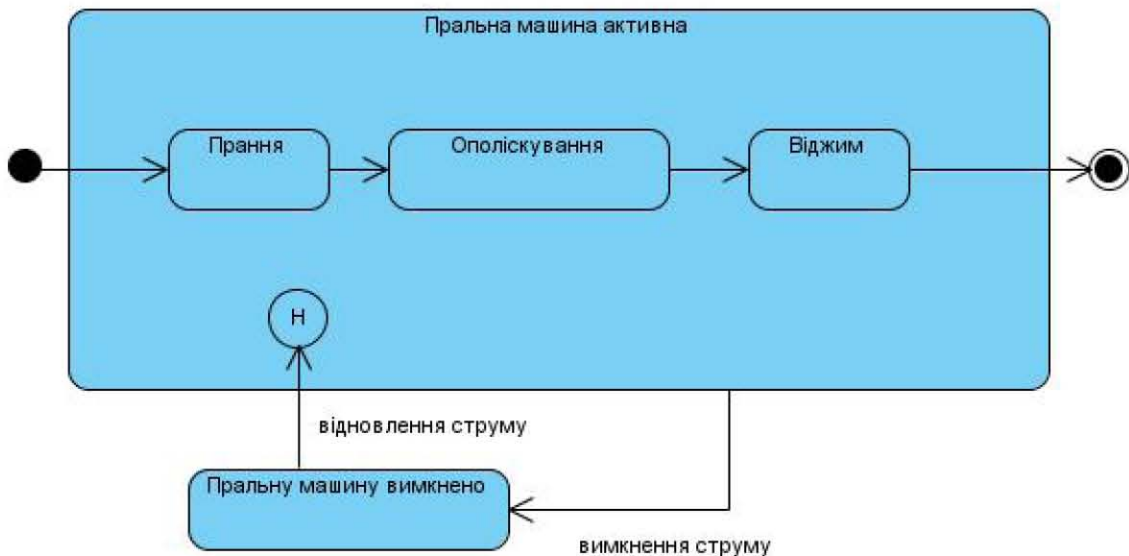


Рис. 3.2.18. Діаграма станів функціонування пральної машини

Глибокий історичний стан (deep history state або стан глибокої історії) також позначається у формі невеликої окружності, у яку поміщена латинська буква "H" з додатковим символом "*" (рис. 3.2.17) і служить для запам'ятовування всіх підстанів будь-якого рівня вкладеності для вихідного складеного стану.

Розглянуте вище поняття переходу цілком достатньо для більшості типових розрахунково-аналітичних завдань. Однак сучасні програмні системи можуть реалізовувати складну логіку поведінки своїх окремих компонентів. Іноді для адекватного відображення процесу зміни станів семантика звичайного переходу для них недостатня. Із

цією метою в мові UML специфіковані додаткові позначення й властивості, якими можуть володіти окремі переходи на діаграмі станів.

Паралельний перехід – перехід, який може мати кілька вихідних станів або цільових станів. Використання паралельних переходів може бути обумовлено необхідністю синхронізувати й/або розділити окремі процеси керування на паралельні гілки без специфікації додаткової синхронізації в паралельних кінцевих підавтоматах.

Графічно такий перехід зображується вертикальною рисою, аналогічно позначенню переходу у формалізмі мереж Петрі. Якщо паралельний перехід має дві або більше вихідних з нього дуг (рис. 3.2.19), то його називають поділом (fork). Якщо ж він має дві або більше вхідні дуги (рис. 3.2.19), то його називають злиттям (join). Текстовий рядок специфікації паралельного переходу записується поруч із рисою й відноситься до всіх вхідних або вихідних дуг.

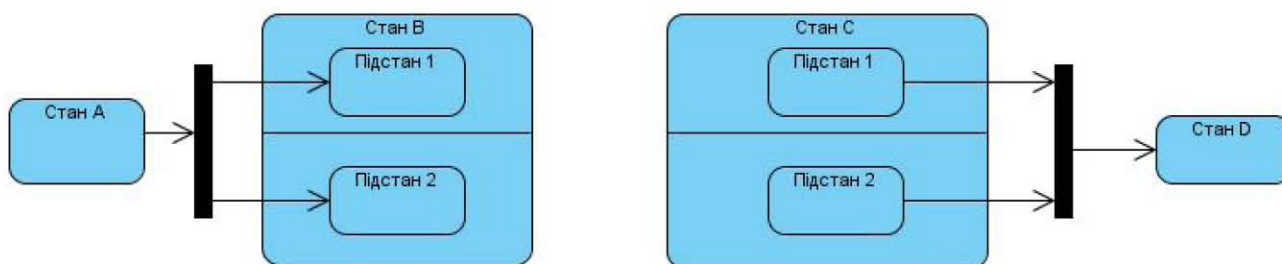


Рис. 3.2.19. Графічне зображення переходу-поділу в паралельні підстани і переходу-злиття з паралельних підстанів

Спрацьовування паралельного переходу відбувається в такий спосіб. У першому випадку відбувається поділ складеного кінцевого автомата на два кінцевих підавтомати, що утворюють паралельні гілки вкладених підпроцесів. При переході-поділі система або об'єкт, що моделюється, одночасно буде перебувати у всіх цільових підстанах цього паралельного переходу (підстани 1 й 2 стану В). Далі процес зміни станів буде протікати відповідно до раніше розглянутих правил для складених станів.

У другому випадку перехід-злиття спрацьовує, якщо має місце подія-тригер для всіх вихідних станів цього переходу, і виконані (при їх наявності) сторожових умов. При спрацьовуванні переходу-злиття одночасно лишаються всі вихідні підстани переходу (підстани 1 й 2 стану С) і відбувається перехід у цільовий стан. При цьому кожен з вихідних підстанів переходу повинен належати окремому кінцевому підавтомату, що входить у складений кінцевий автомат.